EPiC Series in Computing

Volume 65, 2019, Pages 42–53

GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence

Visual Reasoning on Complex Events in Soccer Videos Using Answer Set Programming

Abdullah Khan[1,2,3], Loris Bozzato[1], Luciano Serafini[1], and Beatrice Lazzerini[3]

[1] Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy
[2] University of Florence, Via di Santa Marta, 3, 50139 Firenze, Italy
[3] University of Pisa, Via Largo L. Lazzarino 1, 56122 Pisa, Italy

{akhan,bozzato,serafini}@fbk.eu, b.lazzerini@iet.unipi.it

Abstract

In the context of computer vision, most of the traditional action recognition techniques assign a single label to a video after analyzing the whole video. We believe that understanding of the visual world is not limited to recognizing a specific action class or individual object instances, but also extends to how those objects interact in the scene, which implies recognizing events happening in the scene. In this paper we present an approach for identifying complex events in videos, starting from detection of objects and simple events using a state-of-the-art object detector (YOLO). We provide a logic based representation of events by using a realization of the *Event calculus* that allows us to define complex events in terms of logical rules. Axioms of the calculus are encoded in a logic program under Answer Set semantics in order to reason and formulate queries over the extracted events. The applicability of the framework is demonstrated over the scenario of recognizing different kinds of kick events in soccer videos.

Keywords: Event detection in video, Event Calculus, Answer Set Programming

1 Introduction

The increase in availability of data in both structured and unstructured formats is a common trend nowadays: on the other hand, information extraction for a meaningful use from this ocean of data is still a challenging task. The interpretation of these data need to be automated in order to be transformed into operational knowledge [3, 20]. In particular, *events* are mostly important pieces of such knowledge, as they represent activities and happenings in the represented scenario.

The goal of *event detection* from unstructured data formats (e.g. videos and images) consists in identifying and localizing specified spatio-temporal patterns in such data, where each pattern represents a significant event. Understanding of events taking place in videos is a challenging problem for the vision community due to factors such as, e.g. background clutter and variations in pose, illumination and camera point of view. Moreover, complex video sequences [9] may contain many activities and involve multiple interactions between objects: the recognition of such composite interactions is thus more demanding than the classification of single actions.

In fact, event recognition is considered to be a paragon for all computer vision tasks [2], also because of its wide applicability to different real world scenarios. Advances in deep convolutional neural networks (CNN) in recent times have mostly focused on developing end-to-end black box architectures that achieve a high accuracy in recognizing events. However, the major drawback of such approaches is the interpretability of the model [29]. For complex events, humans can analyze the properties of complex actions and inject some semantic knowledge to extract semantically meaningful events. Whereas, CNN based black box architectures often rely on high accuracy given the event is happening or not.

In this paper we approach the event recognition problem by aiming at bridging the gap between the methods based on deep learning, used in the extraction of events from the visual data, and logical reasoning, used for complex inferences on event conditions. Intuitively, the goal of such hybrid solution is the possibility to exploit the accuracy of learning and the rich semantic characterization of a logic based representation for complex events. To achieve this objective, in this work we propose a framework which combines both aspects: first, we make use of the state-of-the-art object detector *YOLO (You only look once)* [30] for extracting basic information (appearance and movement) about objects from video streams. Events are then represented inside the logical framework of the Event Calculus [22], which allows for the definition of complex events: the calculus and events representation are implemented as a logic program interpreted under Answer Set semantics in order to reason and formulate queries about the represented scenario.

In this paper we demonstrate and evaluate our approach over videos from the domain of sport activities: in particular, we consider clips extracted from soccer matches and we apply our framework to classify them on the base of a set of complex events (namely "corner kick" and "free kick" events) that can occur in these videos. We note that understanding such complex events from soccer videos is a very challenging task due to the dynamics and variation of video sequences (e.g. different camera angles and cuts, no fixed composition of video sequences).

The paper is organized as follows: in the following section we define the problem and the use-case scenario considered in our work. In Section 3 we briefly review the current approaches for object and event recognition in videos and discuss their limitation in recognizing complex events. We then present our proposed architecture (Section 4) an the components for event extraction (Section 5) and logic reasoning (Section 6), by showing their application in the soccer scenario. Finally, we experimentally evaluate the approach and discuss the results in Section 7.

2 Problem Description

Problem statement. The focus of our work is to recognize complex events from the scene starting from the simple facts that are detectable from the visual information in the input video frames. In particular, complex events can be characterized by conditions that may involve the validity of certain situational variables and temporal facts: thus, their recognition is not limited to verifying that certain sets of atomic events occur in the scene, but reasoning on different aspects of the represented situation is required in general.

Use-case: soccer kick events. To show the applicability of our approach, we consider the case of videos from soccer matches, in particular in the case of videos edited in the style of television or streaming broadcasting.

Several atomic and complex events related to the soccer match happenings can be recognized e.g. goals, substitutions, fouls, off sides etc. In this work we concentrate on recognizing different types of "kick" events, namely different situations in which the match has been stopped (e.g.

Figure 1: Time line definition for the "corner kick" complex event.

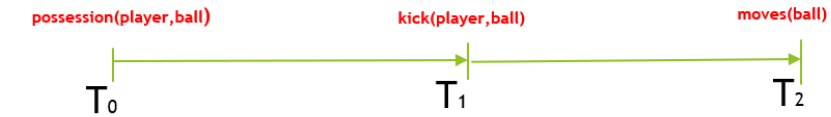

Figure 2: Time line definition for the "free kick" complex event.

after a foul) and the game is resumed by some player kicking the ball. In particular, we consider two common cases: *corner kicks* and *free kicks*.

A corner kick is an action occurring when the ball passes over the goal line by being hit from a player of the defending team: the ball is put into play by a player of the attacking team by taking the kick from the flag on a corner of the playing field. We recognize a corner kick event as a complex event, with visual conditions defined by the time line in Figure 1: we recognize the beginning of the event at time T_0 whenever the ball is seen near to a flag; then at a successive time T_1 a player should come near the position of the ball and kick the ball (i.e. cause the movement of the ball) at time T_2; after the ball has been kicked (usually towards the goal), we require that at T_3 the goal post is visible in the scene.

A free kick occurs when the game has been stopped due to a foul (or other soccer rules infringement): the game is resumed by a player kicking the ball from a position inside the field of play. The definition of the complex event for free kick is thus more relaxed than the one for corner kicks, as shown in Figure 2: at the beginning of the event at time T_0, a player comes in possession of the ball; then at time T_1 the ball is kicked (and thus the ball moves at the successive time point T_2). No further conditions on the visibility and nearness of other objects is required: clearly, we have also to constrain that event types are disjoint and no multiple kick events can happen simultaneously.

The experimental data at our disposal consists of short video clips extracted from the *Soccernet* dataset [15], where each clip is approximately 8 to 10 seconds, depicting the event of interest. We remark that, with respect to the "parking" use-case described in our preliminary work in [21], the task of (complex) event recognition over such data is more challenging due to the unpredictable position, movement and changes in viewpoint of the camera during the happening of events: the recognition of simple events from the visual information is thus complicated e.g. by the lack of fixed spatial references and possible incomplete visibility of an action.

3 Related Work

Object detection in videos aims to detect objects belonging to a pre-defined class and localize them with bounding boxes in a given video stream [30]. Object detectors based on bounding boxes have seen a steady improvement over the years. One of the pioneer CNN-based object detector was R-CNN [18], which involved a two-stage pipeline: one part of the system provides region proposals, then for each proposal CNN is used for classification. To reduce the computational cost Region of Interest Pooling is used in FAST R-CNN [17] leading to efficient results. Furthermore, the most recent object detectors [30, 25] combine the two tasks of region proposal and classification in one system. Single shot object detectors, YOLO, SSD (single shot multi-box detector) significantly improved the detection efficiency compared to prior object detection systems.

Event recognition in videos mostly focuses on understanding videos by classifying them according to predefined set of action classes. Traditional approaches for event recognition make use of the dense trajectories [35], Recurrent Neural Networks (RNN) [8] and handcrafted features [10]. They rely on aggregating local features and pooling them looking for a consensus of characteristics. Traditional approaches for event detection in videos such as Bag of words (BOW) [11] and Fisher Vector (FV) [28], look for a structure from the extracted features using clustering and then pool them in a way to improve classification. Currently, deep neural networks architectures such as NetVLAD [5] and ActionVLAD [16] caught the eye of the research community, which look for co-relations between a set of simple actions representations. More recently, considering dynamic nature of videos, encoding of temporal information as input became a trend: the concept involves extending the two-dimensional convolution to three dimensions, leading to 3D CNNs, which includes temporal information as a distinct input [34, 26]. Another approach to encode temporal information is through the use of Long-Short Term Memory (LSTM) networks [26]. One of the most successful frameworks for encoding both spatial and temporal information is the two-stream CNN [31]. The combination of 3D convolutions and the two-stream approach recently reported for video classification, received appreciation from the scientific community [12]. For event detection in soccer videos a recently proposed framework [24] combines temporal action localization using 3D convolutional networks and play-break (PB) rules for soccer video event detection achieving satisfactory results. In [19] authors construct a deep neural network for soccer video event detection combining CNN and RNN, taking advantage of Convolution Neural Network (CNN) in fully exploiting features and the ability of Recurrent Neural Network (RNN) in dealing with the temporal relation.

Most of the end-to-end black box architectures discussed above try to capture the pose, movements that are the part of actions known as atomic actions and interactions (*e.g., walking, running, surfing, riding* etc.), but such methods are not very successful in capturing semantically meaningful representation of actions. Injecting semantic definition and structural knowledge in these approaches is rather difficult: it is of great importance for the model to be interpretable, and this is a part where neural networks fall short.

Recently, some researchers have come up with hybrid solutions. For example, in [29] Human action recognition in videos is performed by distinguishing between simple and complex actions: to recognize simple actions, the solution take advantage of the 3D convolutional networks, while for complex actions that involve interaction between more than one individual, it uses the recognized simple human actions step to generate Event Calculus theories. A hybrid artificial intelligent system is proposed in [14] for human surveillance in wide open areas where objects are identified and localized by jointly employing cameras and RFID tags. Another approach [7] exploits the combination of data stream processing and rule based programming for recognition

of gestures.

In our work we exploit a classic logic-based event recognition framework (*Event Calculus*) for complex event representation based on logical rules and its Answer set program rendition to reason and formulate query over the events. The role of Answer set programming in conjunction with computer vision to formalize rules for visual scene understanding and answer queries about the event occurring in the scene is very recent [1, 33].

We note some similarities of our proposal to other rule based approaches for reasoning on events. For example, Cached Event Calculus reasoner (jREC) presented in [13] has been successfully used with the aim to monitor the health status of patients. Etalis [4] provides a rich set of operators for specifying how composite events are derived from primitive ones. A reactive and logic-based version of Event Calculus REC (Reactive Event Calculus) is presented in [6] for providing a solid formal background for monitoring declarative properties of events.

4 Proposed Architecture

Figure 3 describes the workflow for our proposed architecture. Our method follows two phases: (1) objects are detected and tracked from every single frame using YOLO, providing simple events such as appearance and disappearence of an object; (2) based on those candidate objects and simple events, complex events are represented in the logical framework of the *Event Calculus*. Reasoning on complex events is obtained by an encoding in logic programs under *Answer set* semantics (in particular, programs are run using DLV [23]). In the following sections, we detail the realization of the two steps and their results in the application to our experimental scenario.

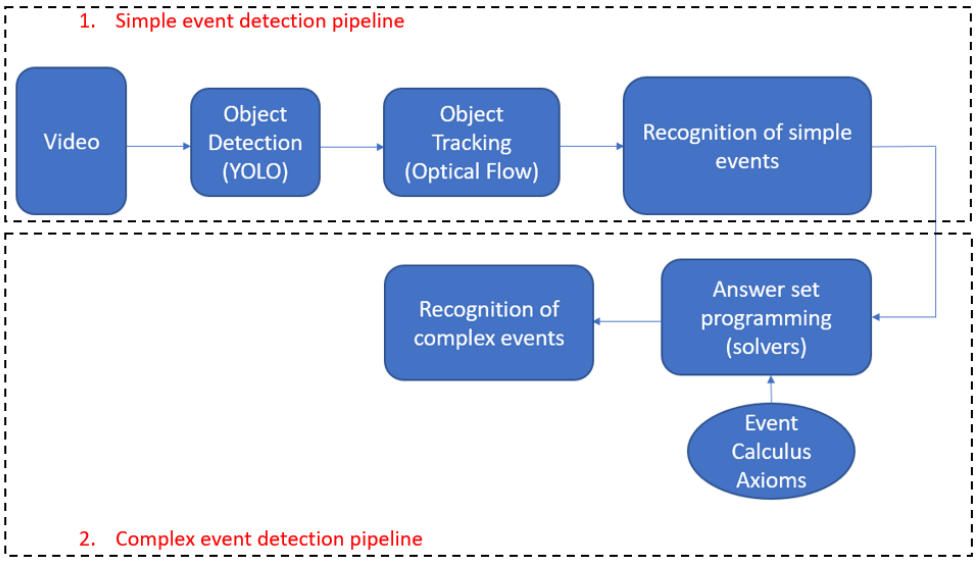

Figure 3: Block diagram of the proposed architecture

5 Objects and Simple Events Extraction from Video

Given a video as input, the task of object detector is: (1) determine whether an object exists or not in the image/video, (2) determine the location of the object by putting a bounding box around it. Most of the methods previously used for object detection have one thing in common: they have one part of their system dedicated to providing region proposals which includes re-sampling of pixels and features for each bounding box, followed by a classifier to classify those proposals. These methods are useful but are computationally expensive resulting in a low frame rate. Another simpler way of doing object detection is by using the YOLO system, which combines the two tasks of region proposal and classification in one system. The key idea behind YOLO is the use of small convolutional filters applied to feature maps of bounding boxes to predict the category scores, using separate predictors for different aspect ratios to perform detection on multiple scales. YOLO uses the *optical flow* method from OpenCV[1] to track objects by determining the pattern of motion of objects for two consecutive frames, which occurs due to the movement of the objects, helping in image segmentation and tracking. It works on the following assumptions. (1) Pixels grouped with similar motion, result in blob of pixels for all objects having different motion. (2) Intensities of pixels do not change between consecutive frames. (3) Neighbouring pixels have similar motion. We trained the system to detect four objects: player, ball, flag, goalpost.

The information extracted by YOLO consists of a unique frame identifier, class identifier, track identifier and bounding box co-ordinates for each object in the video. In our framework, this visual information is then post-processed by a Python script for the extraction of simple events. For the current use-case, such events include: the object *appears*, *disappears*, *moves* and two objects become *close*. The appearance and disappearance of the objects are simply identified from the track identifier of the objects. The closeness and movement of objects are determined on the base on the centroid distance between the objects.

The extracted information about objects and simple events derived from the visual data can be considered as the output of the first phase of our workflow: this knowledge will then be used as the input instance data of the logic based representation of events in the second step of the workflow, detailed in the following section.

6 Logical Reasoning on Complex Events

In this section we review the definition of Event Calculus as presented in [32]; then, we present the encoding of its axioms as datalog rules and we show how we can use it to reason over the events extracted from video in our scenario.

Event Calculus. Event Calculus (EC) was first introduced by Kowalski and Sergot in [22] as a logic framework for representing and reasoning about events and their effects. EC has been frequently used for event recognition as it provides a set of rich axioms for capturing the behavior of events and their effects. The EC language consists of (ordered) time-points, events and fluents. A *fluent* is a property whose truth value may change over time, such as the location of a physical object. The expressions referring to temporal entities that occur over some time interval are called *events*. After an event occurs, it may change the truth value of a fluent. It is assumed that the value of a fluent is preserved in successive time points, if no event changes its state. In particular, an event can *initiate* a fluent, meaning that the fluent is true after the

[1] see https://opencv.org/ and https://github.com/AlexeyAB/darknet

Basic Predicates	Description
$holdsAt(f,t)$	fluent f is true at time-point t
$happens(e,t)$	event e occurs at time-point t
$initiates(e,f,t)$	if event e occurs at time-point t, then fluent f will be true after t.
$terminates(e,f,t)$	if event e occurs at time-point t, then fluent f will be false after t

Table 1: Event Calculus predicates

happening of the event, or *terminate* a fluent, meaning that the occurrence of the event makes the fluent false.

The calculus makes use of the predicates listed in Table 1. The language provides predicates expressing the various states of an event occurrence: *happens* defines the occurrence of an event at a given time point, while *holdsAt* states that a fluent holds in a point in time. The predicates *initiates* and *terminates* specify under which circumstances a fluent is initiated or terminated by an event at a specific time point.

Event Calculus in Answer Set programs. An implementation of the Event Calculus into answer set programs is provided in [27]. The EC axioms determining the relation across fluents and events are defined by the rules that follow.[2]

$$initiated(F,T) \leftarrow happens(E,T), initiates(E,F,T). \quad (1)$$

$$terminated(F,T) \leftarrow happens(E,T), terminates(E,F,T). \quad (2)$$

$$holdsAt(F,T_1) \leftarrow holdsAt(F,T), \texttt{not}\ terminated(F,T), time(T), T_1 = T+1. \quad (3)$$

$$\leftarrow holdsAt(F,T_1), \texttt{not}\ holdsAt(F,T), \quad (4)$$
$$\texttt{not}\ initiated(F,T), time(T), T_1 = T+1.$$

$$holdsAt(F,T_1) \leftarrow happens(E,T), initiates(F,T), time(T), T_1 = T+1. \quad (5)$$

$$\leftarrow holdsAt(F,T_1), happens(E,T), terminates(F,T), \quad (6)$$
$$time(T), T_1 = T+1.$$

Axiom (1) and (2) state that a fluent is *initiated* with the occurrence of an event that *initiates* it, and that fluent will be *terminated* when another event occurs and *terminates* it. Axiom (3) states that if a fluent holds at time-point T and is not terminated in T, then the fluent is true at the next time-point T_1. Axiom (5) states that if a fluent is initiated by some event that occurs at time-point T, then the fluent is true at T_1. Constraint in Axioms (4) state that it can not be that a fluent F that is not initiated nor true at time T becomes true at time $T+1$. Similarly, by the constraint in axiom (6) it can not be that fluent F holds at time $T+1$ if an event happened at time T that terminated F.

Event reasoning on use-case scenario. We can now express our example scenario in terms of the presented ASP encoding of the Event Calculus. For explaining perceived dynamics of objects in the scene, we define the simple and complex EC events listed in Table 2. We base our event reasoning on appearance, disappearance, movement and nearness of objects in the scene: the fluents of our scenario are $visible(G)$, which is true if a goalpost G is currently visible in the scene, and $possession(P,B)$, which holds if a player P is currently in possession of the ball B. Table 2 provides the description of simple EC events. The occurrences of these events are

[2]We use the DLV syntax of rules (in particular, for the use of functors and number operations in rules).

Simple event	**Description**
$appears(G)$	The goalpost G enters the scene
$disappears(G)$	The goalpost G leaves the scene
$close(A, B)$	Object A and B are close to each other
$movesBall(B)$	The ball B moves in the scene
Complex event	**Description**
$kick(P, B)$	Player P kicks the ball B

Table 2: Description of simple and complex EC events

directly extracted from the output of the previous phase of the workflow: in other words, they will be compiled as facts in the final program. Given this information, complex EC events are then defined by combining simple events and conditions on fluents: in our scenario, we detect that a player *kicks* the ball if the player is in possession of the ball and the ball starts moving.

The definition for complex events for *corner kick* and *free kick* is then formulated as rules which body encode the composite conditions that define such events provided in Section 2 and in Figures 1 and 2. These conditions are given in terms of what can be derived from the result of reasoning on EC events and fluents together with further constraints e.g. on the sequence of occurrence of EC events. These rules can be seen as another program layer that reasons on the results of EC reasoning.

Assuming the previous rules defining the EC axioms, we encode our scenario in a logic program with the rules that follow. We first declare events and fluents of the scenario:

$$\begin{aligned}
event(appears(G)) &\leftarrow goalpost(G).\\
event(disappears(G)) &\leftarrow goalpost(G).\\
event(close(A,B)) &\leftarrow player(A), ball(B).\\
event(close(A,B)) &\leftarrow ball(A), flag(B).\\
event(movesBall(B)) &\leftarrow ball(B).\\
fluent(visible(G)) &\leftarrow goalpost(G).\\
fluent(possession(P,B)) &\leftarrow player(P), ball(B).
\end{aligned}$$

We can then specify the effects of events on fluents:

$$\begin{aligned}
initiates(appears(G), visible(G), T) &\leftarrow goalpost(G), time(T).\\
terminates(disappears(G), visible(G), T) &\leftarrow goalpost(G), time(T).\\
initiates(close(P,B), possession(P,B), T) &\leftarrow player(P), ball(B), time(T).\\
terminates(movesBall(B), possession(P,B), T) &\leftarrow player(P), ball(B), time(T).
\end{aligned}$$

Basically, the rules define that the appearance of an object initiates its visibility and its disappearance from the scene terminates the validity of the visibility fluent. For ball possession, the fluent starts its validity once a player is close to the ball and terminates once the ball starts moving (away from the player). Occurrences of complex events are derived from event calculus reasoning:

$$\begin{aligned}
happens(kick(P,B),T) \leftarrow\ & player(P), ball(B), time(T),\\
& holdsAt(possession(P,B),T),\\
& happens(movesBall(B),T).
\end{aligned}$$

By this rule, we recognize that a *kick* event occurs at a certain time slot if a player is in possession of the ball and the ball starts moving. By combining the information derived on

Class	Precision	Recall
Free kick	68.4 %	65 %
Corner kick	83.33 %	25 %

Table 3: Experimental Results

fluents, simple and complex EC events, we can define the conditions to recognize the different types of kick events. For example, for corner kicks:

$$\begin{aligned} \mathit{cornerkick}(T_1) \leftarrow\ & \mathit{player}(P), \mathit{ball}(B), \mathit{flag}(F), \mathit{goalpost}(G), \mathit{time}(T_1), \mathit{time}(T_2), \mathit{time}(T_3), \mathit{time}(T_4), \\ & \mathit{happens}(\mathit{close}(B,F), T_1), \mathit{holdsAt}(\mathit{possession}(P,B), T_2), \\ & \mathit{happens}(\mathit{kick}(P,B), T_3), \mathit{holdsAt}(\mathit{visible}(G), T_4), \\ & T_1 \leq T_2, T_2 \leq T_3, T_3 \leq T_4, \\ & \texttt{not}\ \mathit{exists\_prev\_event}(T_1). \end{aligned}$$

Namely, a corner kick is recognized if: at time T_1 the ball starts being close to a flag; then a player is in possession of the ball at T_2 and kicks the ball at T_3; after the ball has been kicked, a goal post has to be visible at time $T4$. A similar rule can be defined for recognizing free kicks:

$$\begin{aligned} \mathit{freekick}(T_1) \leftarrow\ & \mathit{player}(P), \mathit{ball}(B), \mathit{time}(T_1), \mathit{time}(T_2), \\ & \mathit{happens}(\mathit{close}(P,B), T_1), \mathit{happens}(\mathit{kick}(P,B), T_2), \\ & T_1 \leq T_2, \texttt{not}\ \mathit{cornerkick}(T_1), \\ & \texttt{not}\ \mathit{exists\_prev\_event}(T_1). \end{aligned}$$

The last condition in both definitions is a constraint to only consider the first kick event in the video, supported by the following rules:

$$\begin{aligned} \mathit{kickevent}(T_1) &\leftarrow \mathit{cornerkick}(T_1). \\ \mathit{kickevent}(T_1) &\leftarrow \mathit{freekick}(T_1). \\ \mathit{exists\_prev\_event}(T_2) &\leftarrow \mathit{kickevent}(T_1), \mathit{time}(T_2), T_1 < T_2. \end{aligned}$$

The final program, encoding the scenario of a clip, is obtained by combining these rules (together with the EC axiom rules) with the facts obtained from the tracker output. Let us consider an example instantiation:

$$\begin{aligned} & \mathit{happens}(\mathit{close}(\mathit{player}1, \mathit{ball}1), 1). \\ & \mathit{happens}(\mathit{close}(\mathit{ball}1, \mathit{flag}1), 1). \\ & \mathit{happens}(\mathit{movesBall}(\mathit{ball}1), 3). \\ & \mathit{happens}(\mathit{appears}(\mathit{goalpost}1), 4). \end{aligned}$$

According to the input evidence, first the player is recognized to be near the ball in frame 1. In the same frame, the ball is recognized to be near a flag. In the successive frame 3, the ball starts moving and in frame 4 a goalpost appears in the scene. Using the rules, we can thus derive the occurrence of complex event $\mathit{happens}(\mathit{kick}(\mathit{player}1, \mathit{ball}1), 3)$ and the validity of fluents $\mathit{holdsAt}(\mathit{possession}(\mathit{player}1, \mathit{ball}1), 1)$ and $\mathit{holdsAt}(\mathit{visible}(\mathit{goalpost}1), 4)$. This allow us to obtain the conditions to derive $\mathit{cornerkick}(1)$.

7 Evaluation on Complex Event Detection

In order to evaluate the performance of our system, we created a dataset of video clips trimmed from the *Soccernet* dataset [15]. The trimmed video clips are depicting a specific class of action.

		Actual	
		Free kick	Corner kick
PC	Free kick	26	12
	Corner kick	2	10

Table 4: Confusion matrix for complex event detection (PC = predicted class).

Our sample dataset consists of 80 clips, categorised as 40 free kicks and 40 corner kicks, where each clip is approximately 8 to 12 seconds. The clips have been processed using the two step architecture described in previous sections: after the extraction of simple events, we provide the data to the second part of the pipeline, where complex events for corner kick and free kick are derived by querying the resulting answer sets. Experiments related to the visual processing pipeline were performed on a GPU: we have used Nvidia Geforce GTX 1080 with 2560 cores running at 1733 MHz frequency on 8 GB on board GDDR5X memory. We have used CUDA[3] version 8.0 to compile the code. The information extracted from the visual processing pipeline is then post-processed using Python providing basic facts used as an input for the complex events detection pipeline. We use DLV as an Answer Set solver to reason and formulate queries about the presented soccer kicks.

The results of the detection of complex events are expressed in the form a confusion matrix in the Table 4. The overall performance of our experiments is shown in Table 3. The higher value on precision for corner kicks should not surprise, as we have strict rules for this class which gets fired only when several conditions on the event description are met, whereas for free kicks we have more relaxed rules. On the contrary, lower values on recall are partly justified because of the nature of the events we are taking into consideration: intuitively, the conditions for free kicks are relaxations of the ones for corner kicks and both includes a kick event as simple event in their definition. We remark that these results are strongly dependent on the (quite simplified) definition of rules for simple and complex events we provided in this example use-case: in other words, independently from the definition of our architecture, better results could be obtained e.g. by refining the algorithms used in the extraction of simple events and by imposing further conditions over other information in the definition of the datalog rules. Another limitation to the event detection stands in the ambiguities of the tracker output (e.g. multiple labelling of the same object, incorrect disappearance of objects) which produce unclean data at the end of the first step of the workflow. A solution to this problem would be to include a pre-processing step for data cleaning (possibly encoded as logical constraints based on the domain knowledge) which is able to resolve such ambiguities.

8 Conclusion and Future Work

In this paper we presented a method to derive complex events from simple facts which are extracted from the visual recognition techniques. The overall goal of this work is the integration of knowledge representation and computer vision: (1) Visual processing pipeline for simple event detection; (2) Answer set programming based reasoning to derive complex events. We demonstrated the applicability of the proposed architecture over the use-case of videos of soccer matches for the detection and classification of complex "kick events". As discussed above, the limitations of the object tracker result in noisy data affecting the performance of the complex event detection pipeline. As a future work, we aim to manage these inaccuracies by a (possibly

[3] https://developer.nvidia.com/cuda-gpus

logical based) data cleaning step. We also want to apply and evaluate the presented method in different scenarios, in order to understand the possibilities and limits of the approach in different visual contexts.

References

[1] Somak Aditya, Yezhou Yang, Chitta Baral, Cornelia Fermuller, and Yiannis Aloimonos. Visual commonsense for scene understanding using perception, semantic parsing and reasoning. In *2015 AAAI Spring Symposium Series*, 2015.

[2] Kashif Ahmad, Mohamed Lamine Mekhalfi, Nicola Conci, Farid Melgani, and Francesco De Natale. Ensemble of deep models for event recognition. *ACM TOMM*, 14(2):51, 2018.

[3] Adnan Akbar, Abdullah Khan, Francois Carrez, and Klaus Moessner. Predictive analytics for complex iot data streams. *IEEE Internet of Things*, 2017.

[4] Darko Anicic, Paul Fodor, Sebastian Rudolph, Roland Stühmer, Nenad Stojanovic, and Rudi Studer. Etalis: Rule-based reasoning in event processing. In *Reasoning in event-based distributed systems*, pages 99–124. Springer, 2011.

[5] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[6] Stefano Bragaglia, Federico Chesani, Paola Mello, Marco Montali, and Paolo Torroni. Reactive event calculus for monitoring global computing applications. In *Logic Programs, Norms and Action*, pages 123–146. Springer, 2012.

[7] Stefano Bragaglia, Paola Mello, and Davide Sottara. Towards an interactive personal care system driven by sensor data. In *PAI*, pages 54–59. Citeseer, 2012.

[8] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017.

[9] Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla. Label propagation in complex video sequences using semi-supervised learning. In *BMVC 2010*, pages 27.1–12, 2010.

[10] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1914–1923, 2016.

[11] Gabriela Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[12] Ali Diba, Ali Mohammad Pazandeh, and Luc Van Gool. Efficient two-stream motion and appearance 3d cnns for video classification. *arXiv preprint arXiv:1608.08851*, 2016.

[13] Nicola Falcionelli, Paolo Sernani, Albert Brugués, Dagmawi Neway Mekuria, Davide Calvaresi, Michael Schumacher, Aldo Franco Dragoni, and Stefano Bromuri. Event calculus agent minds applied to diabetes monitoring. In *Agents and Multi-Agent Systems for Health Care*, pages 40–56. Springer, 2017.

[14] Michele Fornaciari, Davide Sottara, Andrea Prati, Paola Mello, and Rita Cucchiara. An evidential fusion architecture for people surveillance in wide open areas. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 239–246. Springer, 2011.

[15] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1711–1721, 2018.

[16] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Con-*

ference on Computer Vision and Pattern Recognition, pages 971–980, 2017.

[17] Ross Girshick. Fast R-CNN. In *IEEE ICCV 2015*, pages 1440–1448, 2015.

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE CVPR 2014*, pages 580–587, 2014.

[19] Haohao Jiang, Yao Lu, and Jing Xue. 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)*, pages 490–494. IEEE, 2016.

[20] Abdullah Khan, Beatrice Lazzerini, Gaetano Calabrese, and Luciano Serafini. Soccer event detection. In *IPPR 2018*, pages 119–129, 2018.

[21] Abdullah Khan, Luciano Serafini, Loris Bozzato, and Beatrice Lazzerini. Event detection from video using answer set programing. In *CILC 2019*, volume 2396 of *CEUR Workshop Proceedings*, pages 48–58. CEUR-WS.org, 2019.

[22] Robert A. Kowalski and Marek J. Sergot. A logic-based calculus of events. *New Generation Comput.*, 4(1):67–95, 1986.

[23] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV system for knowledge representation and reasoning. *CoRR*, cs.AI/0211004, 2002.

[24] Tingxi Liu, Yao Lu, Xiaoyu Lei, Lijing Zhang, Haoyu Wang, Wei Huang, and Zijian Wang. Soccer video event detection using 3d convolutional networks and shot boundary detection via deep feature distance. In *International Conference on Neural Information Processing*, pages 440–449. Springer, 2017.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV 2016*, pages 21–37. Springer, 2016.

[26] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1942–1950, 2016.

[27] Erik T Mueller. *Commonsense reasoning: an event calculus based approach*. Morgan Kaufmann, 2014.

[28] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[29] Ioannis Prapas, Georgios Paliouras, Alexander Artikis, and Nicolas Baskiotis. Towards human activity reasoning with computational logic and deep learning. In *SETN 2018*, page 27. ACM, 2018.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE CVPR 2016*, pages 779–788, 2016.

[31] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[32] Anastasios Skarlatidis, Georgios Paliouras, Alexander Artikis, and George A Vouros. Probabilistic event calculus for event recognition. *ACM TOCL*, 16(2):11, 2015.

[33] Jakob Suchan, Mehul Bhatt, Przemysław Wałega, and Carl Schultz. Visual explanation by high-level abduction: on answer-set programming driven reasoning about moving objects. In *AAAI 2018*, 2018.

[34] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[35] Jan C Van Gemert, Mihir Jain, Ella Gati, Cees GM Snoek, et al. Apt: Action localization proposals from dense trajectories. In *BMVC*, volume 2, page 4, 2015.