EDCAR: A Knowledge Representation Framework to Enhance Automatic Video Surveillance

Loredana Caruccio[a,*], Giuseppe Polese[a], Genoveffa Tortora[a], Daniele Iannone[b]

[a] *Department of Computer Science, Univesity of Salerno, Via Giovanni Paolo II 132, Fisciano (SA), Italy*
[b] *Datonix S.p.A., Via Francesco De Sanctis 2, Avella (AV), Italy*

Abstract

The main purpose of video-based event recognition is to interpret activities or behaviors within video sequences, in order to detect and isolate specific events, which have to be readily recognized and prompted to the people responsible for their monitoring. In this paper, we present a knowledge representation framework and a system for automatic video surveillance, which analyzes record scenes in order to detect the occurrence of specific events defined as targets. The framework, named Elements and Descriptors of Context and Action Representations (EDCAR), enables the representation of relevant elements, general descriptors of the context, and actions that have to be captured, including the definition of action compositions and sequences, in order to monitor and recognize abnormal situations. EDCAR and the associated system also support video summarization of relevant scenes, providing an inference engine to handle complex queries. They have been used experimentally on several video surveillance scenarios, which enabled us to prove their effectiveness with respect to similar solutions described in the literature.

Keywords: Knowledge Representation Framework, Video Scenario, Action

*Corresponding author

Email addresses: lcaruccio@unisa.it (Loredana Caruccio), gpolese@unisa.it (Giuseppe Polese), tortora@unisa.it (Genoveffa Tortora), daniele.iannone@datonix.it (Daniele Iannone)

This is a post-peer-review, pre-copyedit version published in Expert Systems with Applications, Elsevier. 131:190-207 (2019). The final authenticated version is available online at: https://doi.org/10.1016/j.eswa.2019.04.031

Composition, Event Recognition, Video Surveillance.

1. Introduction

Video-based Event Recognition is an important research field of Artificial Intelligence, which concerns the automatic analysis of video files, in order to recognize and classify events (Hongeng et al., 2004). There can be simple events, involving single entities, like for example a person walking, and/or complex events, characterized by more entities that may or may not interact among them to achieve common goals. Complex events might be compositions of single events or sequences of unrelated events across groups of entities that may not be related, completely disjoint, and/or included in the scene at different times.

In many practical applications, ranging from video surveillance in airports or other critical environments, to real-time monitoring of patients in hospitals, or even children in a nursery school, it is necessary to recognize and interpret complex events occurring in a video sequence. This is mainly due to the fact that it is difficult to keep surveillance personnel highly concentrated, especially when they have to simultaneously watch several videos on different monitors. Thus, it is necessary to devise a system capable of recognizing complex scenarios on the basis of semantic models representing the monitored situations. To this end, among the proposals provided in the literature, there are knowledge-based approaches (Nevatia et al., 2003; Ghanem et al., 2004; Castro et al., 2011; Tani et al., 2014), and model specific approaches (Fine et al., 1998; Pavlovic et al., 1999; Joo & Chellappa, 2006; Guo et al., 2016), which are based on mathematical, statistical, or grammar-based models. Among the formers we can find approaches enabling the definition of either sequences or compositions of actions, but not the modeling of the context, and vice versa. Concerning model-based approaches, they are efficient for some specific scenarios, but not for general-purpose ones, since some of them might not be suitably modeled through the primitives of the model underlying the given approach.

To tackle these limitations, in this paper we propose a knowledge repre-

sentation framework, named Elements and Descriptors of Context and Action Representations (EDCAR), which enhances the capabilities of existing video surveillance approaches and current knowledge representation methods, by enabling the representation of a context, and the potential events that might occur in it in terms of sequences and/or composition of actions, making it possible to progressively understand situations occurring in a video sequence. In particular, EDCAR permits to: (1) define elements of the context that are relevant to a specific monitored environment; (2) define general descriptors of the context characterizing additional information that could be useful in the monitoring of a specific environment; (3) define actions that have to be captured; and, (4) define action compositions and sequences, in order to observe and recognize abnormal situations. In this way, it is possible to precisely describe events that can occur, and to make predictions on what might be the consequences, by performing inferences on the actions, the instantiated elements, and the descriptors of context.

Based on EDCAR, we have implemented IVIST (Intelligent VIdeo SurveillanT), a system prototype providing automated support for the specification of knowledge with EDCAR, and implementing the associated inference procedures, including the triggering of alarms. IVIST relies on some external modules, such as a tracker and an object detector (Redmon et al., 2016; Kalal et al., 2012).

We have performed several experiments on public datasets, particularly focusing on video sequences containing typical video surveillance scenarios. A comparative evaluation with respect to similar solutions proposed in the literature proved the effectiveness of IVIST and the underlying framework EDCAR.

The paper is organized as follows: Section 2 surveys the related work, and presents the improvements offered by the EDCAR approach. Section 3 describes the proposed framework by illustrating the *Element of Context Representation (ECR)* (Section 3.1), the *Action Representation (AR)* (Section 3.2), the *General Context Descriptors (GCD)* (Section 3.3), and the modeling of scenarios (Section 3.4). Some application scenarios modeled through the EDCAR framework are presented in Section 4. The IVIST system is presented in Section 5, whereas

experimental results are discussed in Section 6. Finally, Section 7 provides the conclusions and discusses future developments.

2. Related Work

In the literature there are several approaches focusing on human activity recognition (Afsar et al., 2015; Zhang et al., 2017, 2018). Based on their structure, they can be divided into two categories: *single-layer* and *hierarchical* approaches (Aggarwal & Ryoo, 2011). Those in the first category are mainly targeted at simple movement and action recognition, whereas those in the second category are characterized by multiple layers of processing, and can potentially recognize interactions and group activities.

Single-layer approaches mainly employ mathematical models with a single-layer structure, which directly acts on the video frames. These approaches deem an activity as a particular class of images, and the recognition is carried out through the association of a sequence of unknown images to a known class (Sheikh et al., 2005; Rodriguez et al., 2008; Jiang et al., 2006). Recent advances on hardware technologies have been exploited to improve some algorithms of human behavior recognition. As an example, by exploiting advances in 3D sensors, algorithms exploiting deep information have been proposed for human joint estimation and behavior recognition (Kim et al., 2016).

Since the target of our framework is the recognition of complex video surveillance scenarios, we analyze hierarchical approaches in more details, with particular emphasis on knowledge-based ones.

Hierarchical approaches deal with the recognition of complex actions through the identification and the correlation of simple actions. In this category we find statistical or syntactical approaches. The formers define systems composed of more than one elaboration layer to enable the recognition of more complex actions. System layers are implemented by means of state-based models as a basis for the activity recognition (Brand et al., 1997; Fine et al., 1998). Syntactical approaches model human activities as strings of symbols, each representing an

atomic action. These techniques have been initially used mainly for pattern recognition within static images, whereas recently they have also been used for human behavior recognition (Joo & Chellappa, 2006).

Recently, many deep learning-based approaches have been developed in the context of visual understanding applications (see Wang & Sng (2015); Guo et al. (2016) for surveys). As an example, in Xue et al. (2016) a convolutional neural network (CNN) is used to classify and recognize people in RGBD videos. The authors combine motion information with the CNN classifier into a probabilistic tracking algorithm, in order to train the classifier offline, and then run the tracking procedure online, yielding a semi-automated approach for surveillance. Moreover, in Park et al. (2016) different sources of knowledge are combined in deep learning, in order to recognize actions in a video sequence. Other approaches propose specific classification models in order to recognize events or correctly distinguish wide sets of human-activities (Zhang et al., 2017; Wang et al., 2018; Sultani et al., 2018).

The class of approaches more similar to our proposal is represented by knowledge-based approaches. They exploit repositories in which the human activities to be detected are explicitly described. The recognition activity consists of matching what the system detects and what is described in the knowledge base. The hierarchy existing among specific actions of the application context is directly modeled within the knowledge base. Thus, the analysis process is simplified and more understandable.

The first proposal in this category is the one from Ghanem et al. (2004), who have developed a system for event recognition using Petri Nets (Peterson, 1981) as a knowledge representation method. The system provides a graphical user interface (GUI) to formulate requests, which are automatically associated to a set of Petri Nets representing the instance components. The system also uses some video management modules in order to extract the tracks identifying primitive events. Such scenes are properly filtered by the Petri Nets in order to recognize composite events matching the formulated requests.

Other proposals in this category rely on Ontologies. For instance, Nevatia

et al. (2003) use an event ontology to represent high-level events as compositions of single events linked by means of spatial, temporal, and logic relationships, possibly involving more than one actor. The ontology-based approach proposed in (Tani et al., 2014) aims to detect single/multiple object events through a set of Semantic Web Rule Language (SWRL) rules (O'connor et al., 2005).The latter are used to classify the different bounding boxes based on their semantics (Group Of Person/ Person), and to associate an appropriate video event class concerning the behavior of its objects. Moreover, the approach introduced in (SanMiguel et al., 2009) integrates two types of knowledge: the scene and the system, where the former describes simple or complex events in terms of objects, their relations (events), spatial context, and so on, whereas system knowledge is processed in order to determine the best configuration of the processing schemas. In particular, system knowledge helps detecting the object/event capabilities, their reactions to specific events, and different analysis schemas. Finally, Castro et al. (2011) have developed a system that permits to customize the intrusion detection scenario according to the specific contexts to which it is applied. The system collects multi-sensor data and integrates them through a generic ontology in a homogeneous way.

A knowledge-based video surveillance system using first-order logic in defined in (Tran & Davis, 2008). It consists of a network of grounded atoms (atomic assertions) describing event occurrences in a video stream. Event detection is accomplished by associating a confidence value, based on the goodness of the detection. The detected ground atoms are involved in an inference process, aiming to deduce new logic formulas for describing events or behaviors. Elhamod & Levine (2012) introduced a semantic behaviour-based approach, which relies on object and inter-object motion features. The approach provides a mathematical and logical description of certain common behaviours, through which it detects behaviours based on the features of recently recorded objects. Such features describe the motion and the spatial relations among objects involved in the scene.

Finally, Lim et al. (2014) proposed an intelligent framework for the detection

of multiple events. The authors modularize the surveillance problems into a set of variables comprising regions-of-interest, classes (i.e. human, vehicle), attributes (i.e. speed, locality), and a set of notions (i.e. rules) associated to each of the attributes, in order to derive knowledge concerning the environment.

Among the approaches surveyed above, the ones more directly comparable with EDCAR are those using a knowledge base to automatically analyze the recorded scenes. With respect to them, EDCAR enables the definition of semantic models that are based on both events (simple or complex actions) and the surrounding context (elements and general information), which permit the sequencing of actions and/or their composition, enhancing the capability to interpret different scenarios based on the context in which they occur. To this end, most of the surveyed knowledge-based approaches have some limitations, since some of them enable the composition of actions but neither their sequencing, nor the modeling of the context. On the other hand, those enabling the modeling of the context do not allow to define sequences or compositions of actions. The only knowledge-based approach that allows to define sequences of actions is the one from (Ghanem et al., 2004), but it does not allow to compose them, nor to define the context. In addition, EDCAR permits to simultaneously analyze multiple scenarios, enabling the identification of the current scenario based on the occurred events.

3. The Knowledge Representation Framework EDCAR

Video event understanding is the translation process of low-level video contents into high-level semantic concepts (Lavee et al., 2009). In this context, the concept of video sequence represents the key point for classifying low-level contents into their semantic interpretation. Indeed, it might happen that contents appearing in different video sequences yield different semantic interpretations. Moreover, the definition of *action* in the context of human activity recognition provides the second key point in the event recognition process. Such a definition considers an action as a set of gestures, where the term "set" does not impose

particular restrictions on how the action is interpreted, how its gestures are related, or how actions depend on each other.

A fundamental challenge in the real-time contexts is the capability to recognize actions when they occur. Moreover, in order to support the recognition in complex scenarios, contextualized into specific environments, there is the need of customizing what has to be detected. One way to do this is to enable the modeling of knowledge concerning target scenarios. However, general-purpose knowledge representation frameworks provide too basic mechanisms, which make it difficult to model complex target scenarios through simple rules. This led us to define a new knowledge representation framework, named EDCAR, which enables the modeling of objects or actors, context, and actions of the scenario to be detected by means of *ECR (Elements of Context Representations)*, *GCD (General Context Descriptors)* , and *AR (Action Representations)*, respectively. In particular, in order to define such scenarios, it has been necessary to address the concept of an action composed of more elementary ones, and the definition of action sequences. A scene instance is described by *relevant elements of context*, which collect data about the points of interest in the recorded scene, including their correlations. GCDs can be related to ECRs and describe the whole environment, such as geographical, historical, emotional or biological information.

By defining target scenarios through the EDCAR framework it is possible to (1) understand current events through the stored knowledge, by reducing events to simple ones, and by instantiating elements, descriptors of context, and actions, with actual data (filling the slots); and (2) reason about events and supply missing information in occurred events by making inferences on the instantiated data.

The EDCAR framework is shown in Figure 1. It is composed of seven layers as described in the following.

- *Enviromental Layer*, collects data from a set of video surveillance devices, such as cameras, microphones, thermal sensors, and so forth;
- *Frame Layer*, analyzes frame sequences in order to identify elements within

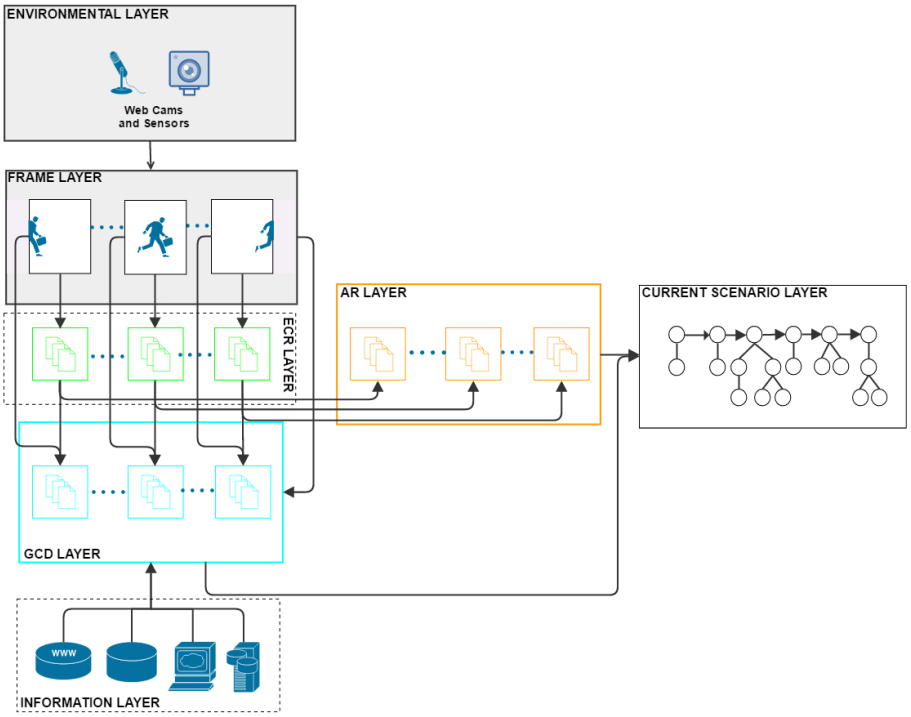


Figure 1: The EDCAR framework.

them, and to extract useful information instantiating ECR, GCD, and AR forms.

- *ECR Layer*, collects all the relevant elements of context generated through the analysis on frames in the previous layer and the ECRs.
- *AR Layer*, collects all the actions involving ECRs, generated according to the scene interpretation and the ARs.
- *GCD Layer*, collects all the contextual information according to the GCDs. Such information is generated by the analysis of both frames in the frame layer, which can also involve ECRs, and the data collected through the Information Layer, described next.
- *Information Layer*, collects all the useful information to characterize the environment, such as statistical, geographical, and/or historical data, by

using external information sources in terms of Open Data, statistics, or sensor data.

- *Current Scenario Layer*, relates information concerning ARs and GCDs instantiated in the previous layers, and determines a current scenario instance. In particular, the information is interpreted according to their sequentiality and composition.

By summarizing, the whole information flow is based on several internal and/or external information sources, and on the frame-based models defined within target scenarios, in terms of ECR, AR, or GCD. In particular, the first two layers characterize peculiarities that are common to other existing video analysis approaches. The last five layers enable the interpretation of the occurring scenario by only focusing on relevant events, in terms of elements, actions, and context information. This represents the strength point of the EDCAR framework, since it permits to define complex scenarios to be monitored by composing and sequencing actions, elements of context, and contextual information.

In the following subsections the ECR, AR, and GCD forms are described in details (3.1, 3.2, 3.3, respectively). In 3.4 it is described how to define target scenarios to be detected, in terms of action sequences.

3.1. Elements of Context Representation

The representation of elements of context (ECR) is a significant part of the proposed knowledge representation framework. In fact, such a representation enables the definition of knowledge concerning specific elements that have to be caught and monitored in order to derive a correct interpretation of a video sequence. In particular, an element of context can be formally described through the following features that must occur in order for the element to be recognized:

- *Type*. A scene may contain different kinds of elements. Each element of context may assume as value of the type feature one of the basic types *Actor* or *Object*, or a specialization of them. A specialization hierarchy

can be constructed for both basic types. As an example, in a railway station environment, a *TRAVELER* can be defined as a specialization of *PERSON* (which is itself a specialization of *ACTOR*). This is done by specifying *PERSON* as type of *TRAVELER*. Moreover, it is possible to add some constraints to the *Type* specification, by detailing specifications such as *.CONSTANT*, which expresses the fact that an element will always participate to the scene and that its properties (defined below) will never change. As an example, we can consider the *GROUND* constant as an element of context, and can define it by adding *.CONSTANT* to its type specification.

- *Properties*. An element holds specific properties. The specification of properties in the model refers to technical information that can be caught through devices and/or feature extraction algorithms (Laptev, 2005; Hongeng & Nevatia, 2001). An example of property is *location*, which is composed of the spatial coordinates (XValue and YValue) of a specific point of the element (e.g. its centre). Moreover, additional properties can be determined by means of rules defining relationships between pairs of elements in the scene. As an example, in the railway station environment a *ownership* property between the *BAGGAGE* and its owner can be defined.
- *Entry condition*. Some elements have to satisfy an *Entry condition* in order to be considered relevant in the scene, and/or to be distinguished from others. The *Entry condition* can be specified in terms of both rules and actions. As an example, a *TRAVELER* specification can be distinguished from the specification of a generic *PERSON* if it presents an entry condition specifying a "*baggage holding*" property.

It is worth to notice that for the definition of a target scenario it is necessary to characterize all and only the elements of context that appear to be relevant in the situations to be described. As an example, in the context of a railway station the recognition of a timetable is not as relevant as the recognition of travelers, baggages, and so forth.

TRAVELER

Features	**Description**
Type	PERSON
Properties	Location: XValue, YValue Area: dim
Entry Condition	(HAS, baggage)

Table 1: An ECR describing the concept of TRAVELER.

An example of ECR representation is shown in Table 1, which describes the features of a *TRAVELER*. This element of context has been specified as a specialization of *PERSON*; it holds location and area *Properties* in the scene, and it has to satisfy the *Entry Condition* concerning the baggage holding.

3.2. Action Representation

In order to recognize and interpret a scene, it is important to recognize single actions occurring in the scene that can altogether be interpreted as an event. Moreover, an action can produce a side effect, since it might yield a change of context.

Action representation (AR) models define knowledge concerning the correct recognition of an action. In particular, an AR model allows us to define the features that an action should exhibit in order to be interpreted as a given one. It can be described through the following features that must occur in order for the action to be recognized:

- *Type*. An action can be *simple* or *composite*. A simple action represents individual events that can be detected by analyzing the properties of elements occurring in it. As an example, an action determining a *change of position* (*CHANGE_LOC*) of an actor or an object can be considered as simple, because it can be recognized by analyzing the *location* property. Instead, a composite action depends on the recognition of other actions. As an example, the *getting close* action (*GET_CLOSE*) performed by a

person with respect to another element depends on the recognition of simpler actions, such as the *CHANGE_LOC* action.

- *Elements*. An action involves one o more elements of context. For this reason, through the elements specification it is possible to define the elements on which the action applies. They can be *Actor*, *Object*, a specialized type (such as PERSON, TRAVELER, and so forth), or *ActorObject*. The latter is specified when the action does not depend on the specific type of element on which it is applied. As an example, the *CHANGE_LOC* action could involve an actor or an object. In this case, the value of the feature *Elements* is *ActorObject*.
- *Rule*. Independently from its type, the recognition of an action depends on the rules characterizing it. A rule can represent (i) an arithmetic/logic formula on elements properties (simple action), (ii) composition of actions (composite action), (iii) a call to a function/module dedicated to the recognition of the action, or (iv) a combination of (i)-(iii). Some examples of AR rules are shown in Table 2. For instance, the rule for the *CHANGE_LOC* action expresses the condition of the property location change by the elements of context to which it is applied.
- *Effect*. An action can produce an effect on the context. The effect yields the context update that must be carried out when the action occurs. In other words, through this definition it is possible to express the context update deriving from an occurring action. As an example, the effect of the *CHANGE_LOC* action is to modify the *location* property of the involved element.

It is worth to notice that for the scenario specification it is necessary to characterize all and only the actions defining the target situation. As an example, in a railway station the action of a person that looks up a timetable will not be as relevant as the movements of travelers around the railway station, their mutual interactions, their actions on baggages, and so forth.

An example of AR representation is shown in Table 3, which describes the

Action	**Rule**
CHANGE_LOC	*ActorObject*.Location ! = (GET_LOC, *ActorObject*)
DIFF_LOC	|(GET_LOC, *ActorObject1*) - (GET_LOC, *ActorObject2*)|
FALL_DOWN	(DIFF_LOC, *ActorObject*, *Ground*) < thr1
PICK_UP	(DIFF_LOC, *Actor*, *Object*) < thr1 ∧ (DIFF_LOC, *Object*, *Ground*) > thr2
SET_DOWN	(DIFF_LOC, *Actor*, *Object*) < thr1 ∧ (DIFF_LOC, *Object*, *Ground*) < thr2
GRASP	(PICK_UP, *Actor*, *Object*) ∧ !(CHECK_OWNER, *Actor*, *Object*)
LEAVE_OBJECT	(CHANGE_LOC, *Actor*) ∧ (DIFF_LOC, *Actor*, *Object*) > thr
CHECK_OWNER	*Object*.Owner == *Actor*? true : false
RUN_AWAY	(CHANGE_LOC, *Actor*) ∧ *Time*.rapid

Table 2: Examples of AR rules.

CHANGE_LOC

Features	**Description**
Type	Simple
Elements	ActorObject
Rule	ActorObject.Location ! = (GET_LOC, ActorObject)
Effect	(SET_LOC, ActorObject)

Table 3: An example of AR.

features of the *CHANGE_LOC* action. In particular, this action is of *Type* Simple; the *Elements* feature is ActorObject, since the action does not depend on the specific type of element to which it is applied; it will be recognized under the *Rule* ActorObject.Location ! = (GET_LOC, ActorObject), which verifies whether the ActorObject location is different from the previously stored one; the *Effect* feature yields the update of the ActorObject location property.

3.3. General Descriptors of Context

The General Context Descriptor (GCD) models permit to better characterize one or more elements identified in the scene, or to provide generic information on the environment. This allows to precisely understand what is occurring in a

scene, since the interpretation could refer not only to elements and actions, but also to general information related to the context.

A GCD model allows us to define general context information that can be integrated with actions in order to interpret a scenario. It can be described through the following features that must occur in order for the context to be recognized:

- *Type*. Several context information can be collected according to sources managed in the information layer. For this reason, it is possible to collect several kinds of data, such as biometric, geographic, or statistic data. Thus, the feature *Type* refers to the type of data that the GCD manages. As an example, information on *expressions* of an actor (e.g., a *FACIAL_EXPRESSION* GCD) could manage data classified as biometric ones.
- *Properties*. A context information holds specific properties. The specification of properties in the model refers to data that can be caught through tools or modules working on external sources and/or on the identified ECR. An example of property could be a *confidence* indicator, whose value indicates the reliability degree of the collected information.
- *Tool/Module*. Independently from its type, context information depends on tools or modules to extract it. As an example, the module extracting *FACIAL_EXPRESSION* information could be the "FE_Learning", which classifies facial expressions of persons detected in a video sequence (Cohen et al., 2003; Zeng et al., 2009; Shan et al., 2009).
- *ECR Index*. A context information can be connected to at most one ECR. Such feature indicates a specific class of ECRs, whereas a NULL value indicates that the information concerns the general environment. As an example, a *FACIAL_EXPRESSION* GCD can be related to ECRs, which are actors in the scene.

It is worth to notice that for the scenario specification it is necessary to characterize all and only the general context information useful to better interpret

FACIAL_EXPRESSION

Features	**Description**
Type	Biometric emotions
Properties	Classification: {Anxiety, Anger, Tranquility, Relaxation} Confidence: value
Tool/Module	FE_Learning
ECR Index	Actor

Table 4: An example of GCD.

the situation.

An example of GCD representation is shown in Table 4, which describes the features of the *FACIAL_EXPRESSION* information. In particular, this information has been specified to be of *Type* Biometric data; it holds the classification of expression and the confidence degree as properties, which can be extracted through the external module FE_Learning, and it has to be connected to an ECR of type Actor through the *ECR Index*.

3.4. Event Modeling

In order to define target events, an implementation can be used in which the target scenario is represented through the definition of (1) the context, and (2) the composition and sequencing of actions that should yield an alarm raise.

In other words, the EDCAR framework allows us to define the structure of the knowledge useful to represent the elements and descriptors of context, and the actions that can be involved in a specific scenario. However, a scenario represents a sequence/composition of actions on the relevant elements of context. In particular, the sequence of actions can be defined with a different occurrence type. The possible occurrence types that can be specified are:

- *Mandatory*. A scenario cannot be recognized if a mandatory action does not occur. Thus, it is strictly required.

- *Optional*. Some actions can be optional. They strengthen the recognition of a scenario, but they are not strictly required.
- *One of*. An action could be optional if one among a set of other actions occurs.
- *Repeatable*. An action can be repeated one or more times.

More than one occurrence type specification can define a single action.

3.4.1. Visual Representation

The definition of action composition/sequence can be accomplished by using the visual language *Pinco*, which depicts actions as circles, whereas relations between actions as directed links. The way in which these links connect action circles represents the instantiation modality of an action in the scenario. The instantiation modality of an action can be accomplished by using the icons shown in Figure 2.

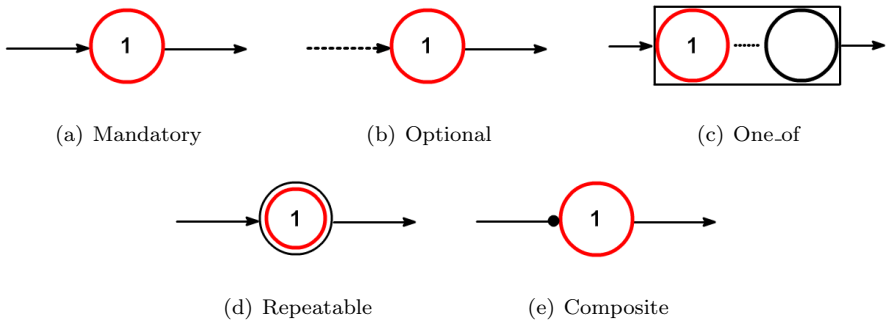
A *mandatory* action is represented in Pinco by a circle linked through a solid directed arrow; whereas an *optional* action is represented by a circle linked through a dashed directed arrow. An action included in a *One_of* set is represented by including the action circles in a rectangle shape. Moreover, *repeatable* actions are modeled by double circles.

A *composite* action is automatically retrieved when an action is added to the visual representation of a scenario. It is represented by an action circle linked through a directed arrow ending with a tiny circle.

A target scenario can be modeled in Pinco by composing its graphical icons. Some examples of complete representations for analyzed scenarios are presented in the next Section.

4. Usage Scenarios

In this section we present some target scenarios modeled through the EDCAR framework. In particular, in the following subsections we present the scenarios “Steal of Baggage”, “Crowd Activity”, “Unattended Baggage”, and



(a) Mandatory (b) Optional (c) One_of

(d) Repeatable (e) Composite

Figure 2: Visual representation types of sequence occurrence and action composition.

"Fighting". Since EDCAR can also be used without contextual information, we have provided both examples using and not using GCDs.

4.1. Steal of Baggage

This scenario describes a possible situation in which a thief steals the baggage of a traveler. In particular, it is composed of the following five main actions: *SET_DOWN*, *APPEAR_ACTOR*, *GET_CLOSE*, *GRASP*, *RUN_AWAY*, on a context composed of the following three relevant elements: *TRAVELER*, *BAGGAGE*, and *GROUND*. The main actions define the target situation in terms of the top-level sequence of actions; other defined actions represent those composing the main ones. The visual representation of the model of this scenario is shown in Figure 3.

The scenario starts with the detection of a TRAVELER with his/her BAGGAGE, that sets the latter down on the GROUND (SET_DOWN). Successively, another PERSON appears in the scene (APPEAR_ACTOR), approaches the baggage (GET_CLOSE), and grasps it (GRASP) by raising it from the ground (PICK_UP); then, the second actor flees from the scene with the baggage (RUN_AWAY).

4.2. Crowd Activity

This example models the scenario "Crowd Activity" that is important to detect panic situations. In particular, the scenario describes many actors that

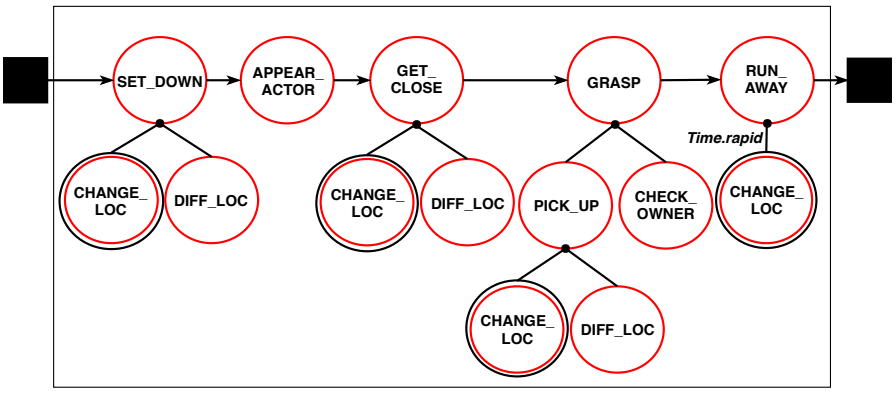


Figure 3: Visual representation of the model for the "Steal of Baggage" scenario.

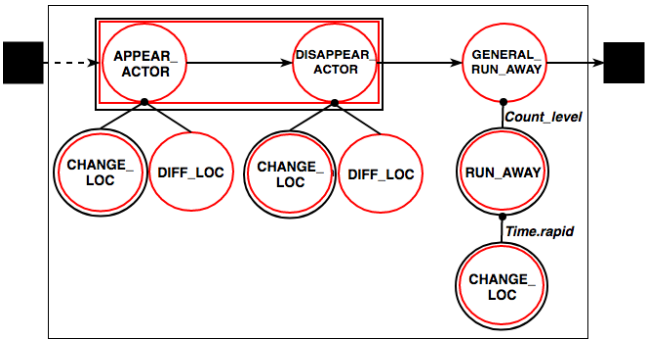


Figure 4: Visual representation of the model for the"Crowd Activity" scenario.

suddenly run away. It is composed of three main actions, two of which are optional, namely *APPEAR_ACTOR* and *DISAPPEAR_ACTOR*, and one mandatory, namely *GENERAL_RUN_AWAY*. The context is composed of one relevant element, namely *PERSON*, which can be instantiated several times. The visual representation of the model of this scenario is shown in Figure 4.

The scenario starts with a set of PERSONS that could be modified according to the occurrence of the actions APPEAR_ACTOR and/or DISAPPEAR_ACTOR, whereas the alarm activation depends on the percentage of persons for which the action RUN_AWAY occurs.

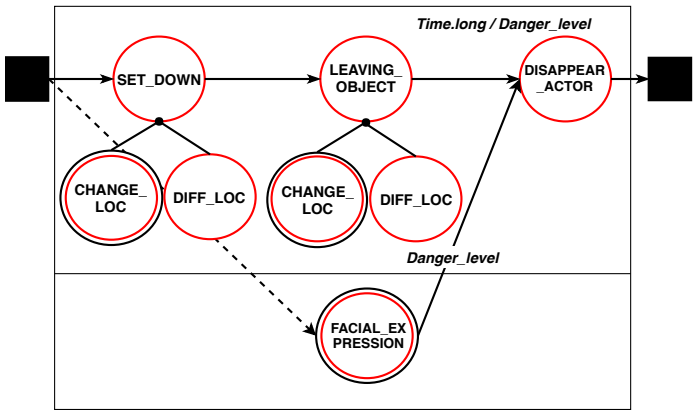


Figure 5: Visual representation of the model for the "Unattended Baggage" scenario.

4.3. Unattended Baggage

This scenario describes a traveler setting down his/her baggage in a railway station and running away. It shows how it is possible to distinguish an involuntary abandonment of baggage from a possible terrorist attack, also by using GCDs. In particular, the scenario is composed of three main actions: *SET_DOWN*, *LEAVING_OBJECT*, and *DISAPPEAR_ACTOR*, in a context composed of three relevant elements: *TRAVELER*, *BAGGAGE*, and *GROUND*, and to which the *FACIAL_EXPRESSION* descriptor is associated, in order to collect emotional data connected to the traveler (Ekman & Friesen, 1978; Pantic & Rothkrantz, 2000). The visual representation of the model of this scenario is shown in Figure 5.

The scenario starts with a TRAVELER that sets his/her BAGGAGE on the GROUND (SET_DOWN), leaves it there (LEAVING_OBJECT), and disappears from the scene (DISAPPEAR_ACTOR). Associated to this scenario there is also a GCD producing a *Danger_level* indicator, according to the facial expression of the traveler (FACIAL_EXPRESSION), which will be used to distinguish the above mentioned two cases:

- *Involuntary baggage abandonment*, in which the traveler has not assumed any expression related to the anxiety and/or anger. In this case, a warning

of baggage abandonment will be raised.

- *Possible terrorist attack*, in which the traveler has assumed a facial expression related to anxiety and/or anger. In this case, a possible terrorist attack alarm will be raised.

In other words, in order to activate the alarm/warning, it is necessary that the actor leaves the scene. Our assumption here is that in case of terrorist attack the terrorist runs away, because in case of suicide attacks the terrorist usually does not abandon the baggage. Thus, the alarm/warning is triggered upon an actor leaving the scene for a given time interval that is tuned based on the recognized facial expression.

4.4. Fighting

This example models the "Fighting" target scenario, that is, the possibility of recognizing a situation in which persons are fighting. Through this example it is possible to understand how to model more complex scenarios in terms of composition of elements of context, actions, and general context descriptors.

In particular, the scenario is composed of four main actions: *APP_ACTOR*, *GET_CLOSE*, *HITTING_PERSON*, and *ANOMALOUS_BEHAVIOR*, on a context composed of only one type of relevant elements: *PERSON*, and to which the *FACIAL_EXPRESSION* and the *DANG_EVENTS* descriptors are associated. The visual representation of the model of this scenario is shown in Figure 6.

The scenario starts with a PERSON, and another one that appears on the scene (APPEAR_ACTOR). Then, they get close (GET_CLOSE), start hitting each other (HITTING_PERSON), and having an abnormal behavior (ANOMALOUS_BEHAVIOR), described as a number of dodges by the person receiving shots. However, associated to this scenario there are also two GCDs producing a *Danger_level* indicator, according to the facial expression of the persons (FACIAL_EXPRESSION), and the known likelihood that dangerous events

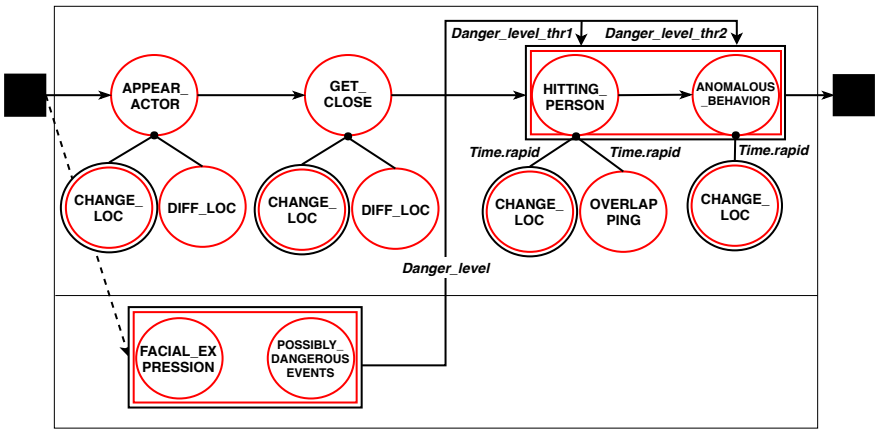


Figure 6: Visual representation of the model for the "Fighting" scenario.

(DANG_EVENTS) occur in that area. Such descriptors will be used to distinguish two cases:

- *Possible joke between friends*, in which persons have not assumed any expression related to anxiety and/or anger, and/or the environment is not signaled as a dangerous one. In this case, no alarm will be raised.
- *Fighting*, in which the persons have assumed facial expressions related to anxiety and/or anger, and/or the environment is signaled as a dangerous one. In this case, an alarm will be raised.

5. The IVIST System

The EDCAR knowledge representation framework described above can be used in the real-time video event understanding system IVIST, which enables the recognition of target events modeled with EDCAR, raising specific alarms.

In order to detect target events, IVIST tries to interpret scenes based on its knowledge of "abnormal" situations. It accomplishes this by using Elements of Context Representations (ECR), Action Representations (AR), and General Context Descriptors (GCD) to describe individual elements, actions, and

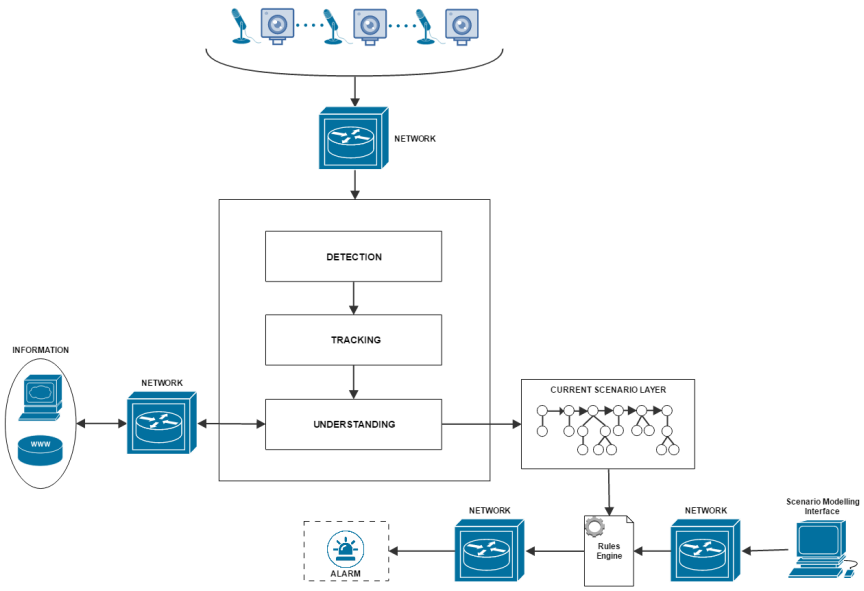


Figure 7: IVIST architecture.

context information, including their compositions and sequential definitions to recognize complex situations.

In Section 5.1 we describe the architecture of the IVIST system, whereas in Section 5.2 we provide implementation details.

5.1. System Architecture

The architecture of the IVIST system is shown in Figure 7. The *input* module shown on the top represents a set of cameras and sensors, which can be used for transmitting input information through a network. The core of the system is composed of three main modules: *Detection*, *Tracking*, and *Understanding*. The latter is the highest level task in computer vision. It needs efficient solutions to handle many lower-level tasks, such as edge detection, optical flow estimation, object recognition, object classification, and tracking. The maturity of many solutions to these low-level problems has spurred additional interest in utilizing them for higher level video understanding tasks (Lavee et al., 2009).

- *Detection.* The goal of the detection module is to acquire the data, to

detect "atomic" events, and to identify elements of context.

- *Tracking.* Object tracking is the process of tracking an object over time by locating its position in every frame of the system (Joshi & Thakore, 2012). Thus, through the tracking module the system tracks moving elements of context in the images produced by the Detection module. The information on the tracked objects are synthesized by means of simple actions, which are given in output to the *Understanding* module.

- *Understanding.* The use of sequences/compositions of actions representing relevant situations allows us to effectively analyze different video scenes, and to derive logic consequences to correlate them, also determining whether they contain target events. In particular, this module concerns the understanding at semantic level of the findings from the previous module, involving the elements of the context, through the instance modeling of the current scenario.

The above described modules interact with the following four modules: i) *Information*, which collects information provided by external sources, in order to enrich the current scenario with contextual information; ii) *Scenario modeling interface*, which permits the visual modeling of target scenarios through a Web-based interface; iii) *Rules engine*, which classifies the current scenario as a target one; iv) *Alarm*, which raises specific alarms.

More details on the implementation of IVIST modules are provided in the next Section.

5.2. Implementation Details

IVIST has been implemented in Python[2], since it permits to produce prototypical systems in a short time, and it permits to exploit powerful Machine Learning[3] and Computer Vision[4] libraries.

[2] `www.python.org`

[3] As an example, scikit-learn: `http://scikit-learn.org/`

[4] As an example, OpenCV: `http://opencv.org/`

In what follows, we provide a detailed description of the IVIST modules.

Detection. The detection module exploits a recent approach to object detection: You Only Look Once (YOLO) (Redmon et al., 2016), which permits to predict bounding boxes and class probabilities directly from full images in a one-step evaluation. The methodology underlying YOLO models the object detection problem as a regression one, and implements the model through a convolutional neural network. After partitioning a frame in an $S \times S$ grid, YOLO verifies whether the center of an object falls into a grid cell, in which case the grid cell becomes responsible for detecting that object and its bounding box. For each bounding box that a cell predicts, a confidence score is computed to quantify how confident the model is that the box contains an object and how accurate is the box. Moreover, a set of class probabilities is predicted for the grid cell containing an object, regardless of the number of boxes that the cell predicts. Thus, an $S \times S$ grid in which each cell predicts B bounding boxes and is assigned C class probabilities is encoded through an $S \times S \times (B * 5 + C)$ convolutional neural network. The initial convolutional layers of the neural network extract features from the image, while the fully connected layers predict the output probabilities and coordinates.

Tracking. The aim of the tracking module is to track moving elements that have been detected in the detection module. More specifically, the main task of the tracking module is to assign an ID to each element in the scene, and to ensure that elements will remain correctly identified upon their movements in the scene.

The current implementation of IVIST is based on the tracker Tracking-Learning-Detection (TLD) (Kalal et al., 2012). TLD is a framework that decomposes the tracking task into three phases: tracking, learning, and detection. During the tracking phase an object is monitored from frame to frame, whereas during the detection phase TLD treats every frame as independent, and scans the image to localize all the appearances that have been observed and learned in the past. Successively, TLD performs the learning phase, whose aim is to

detect possible ID assignment errors of the previous phases, trying to correct them. The learning phase represents the main novelty of TLD, since it is based on a new learning methodology named P-N learning. The latter identifies errors through two types of "experts": the P-expert that identifies false negatives, and the N-expert that identifies false positives. Both the experts make errors themselves, which are mutually compensated thanks to their independence.

Understanding. This is the core module of the IVIST system. It exploits the EDCAR framework to understand what is occurring in the scene. In particular, for each identified and tracked element, and for each additional information on the context extracted from the *Information* module, the *Understanding* module instantiates the defined ECR and GCD models. Moreover, whenever elements interact with each other and/or with the environment, the *Understanding* module cooperates with the *Rules engine* module in order to instantiate the defined AR models. Finally, it performs inferences in order to construct the current scenario and to decide if an alarm has to be raised, according to the defined target scenarios.

Information. The *Information* module collects data from the Open Data databases included in the system, and/or from external Geographic Information Systems (GIS). In particular, the module uses several tools to parse and clean the collected data, so that they can be used by specific algorithms synthesizing them in terms of GCD models.

Rules Engine. This module collects instances of ECR and GCD (provided by the understanding module), and triggers several threads to interpret the action that is occurring, according to the defined target scenarios. The interpretation is made by using simple rules or dedicated modules.

Alarm. When a target scenario is detected, the *Understanding* module will notify the *Alarm* module that an alarm has to be raised. However, according to the simultaneous control of several scenarios carried out by the *Rules engine* module, there is the possibility that more than one alarm should be raised at

the same time. For this reason, the *Alarm* module will decide to whom send the specific alarm signal, by using a binding table. Moreover, the *Alarm* module also maintains a system log storying all the occurred target scenarios.

Scenario Modeling Interface. This module enables domain experts to define a target scenario that IVIST has to check. The module implements both a textual and a visual (Pinco) interface.

6. Experimental Results

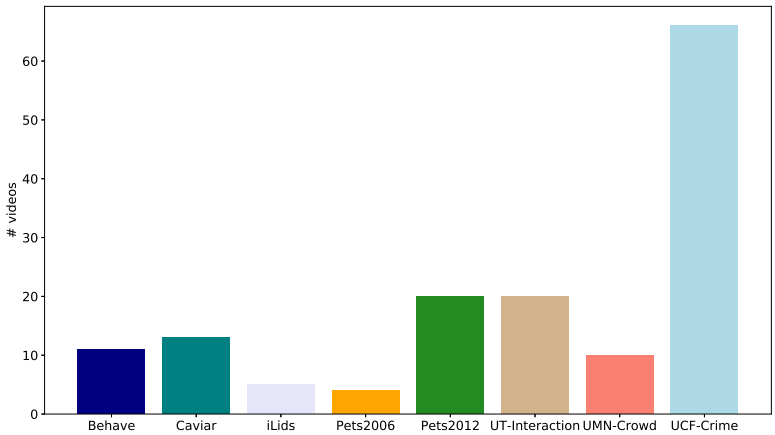
We have experimentally evaluated the IVIST system and its underlying EDCAR framework on a set of video sequences concerning the four target scenarios described in Section 4.

The experiments were performed on a computer equipped with an Intel i6700HQ CPU, 16 GB DDR4 RAM, and a NVIDIA GeForce GTX 970M vid card.

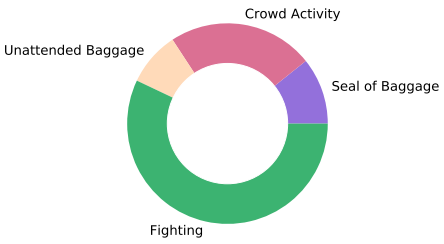
6.1. The Datasets

In recent years, many datasets dedicated to human action and activity recognition have been created (Chaquet et al., 2013). Generally, they refer to different kinds of scenarios, and are useful for the recognition of different actions/activities. Thus, we evaluated the IVIST system and its underlying EDCAR framework on several datasets provided in the literature, and whose characteristics are shown in Table 5. In particular, given the focus of the paper, we selected datasets from the literature that contained video surveillance scenarios.

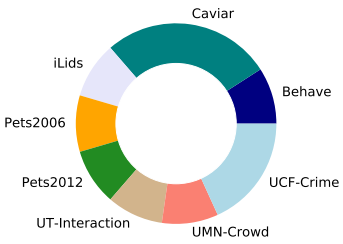
Figure 8 shows several characteristics of the datasets used in our experiments. We selected datasets containing videos lasting from few minutes (e.g. UCF-Crime) up to some hours (e.g. Behave), focusing on videos instantiating the usage scenarios described in Section 4.



(a) Number of analyzed videos for each dataset.



(b) Amount of test videos instantiating the defined scenarios.



(c) Amount of scenarios instantiated within selected datasets.

Figure 8: Number of analyzed videos per dataset and their incidence on the scenarios.

Name	Scenes	No. views	References
Behave	Outdoors	2	(Fisher, 2007)
Caviar	In/Outdoors	1,2	(Fisher et al., 2005)
UMN-Crowd	In/Outdoors	1	(UMN-Crowd, 2009)
iLids	Indoors	1	(Valenzise et al., 2007)
Pets2006	Outdoors	1	(PETS2006, 2006)
Pets2012	Outdoors	4	(PETS2012, 2012)
UT-Interaction	Outdoors	1	(Ryoo & Aggarwal, 2010)
UCF-Crime	In/Outdoors	1	(Sultani et al., 2018)

Table 5: Characteristics of datasets used for the evaluation of IVIST effectiveness.

6.2. Evaluation metrics

We analyzed experimental results by means of the following metrics: the *Positive Predictive Value (PPV)*, the *Detection Rate*, the *Accuracy*, and the *F1-score*. All of them consider *true positives (TP)*, *true negatives (TN)*, *false positives (FP)*, and *false negatives (FN)*. In our context, *true positives* and *true negatives* represent correctly raised and not-raised alarms, respectively, *false positives* represent raised alarms for non-dangerous scenarios, and *false negatives* represent non-raised alarms for dangerous scenarios.

Positive Predictive Value (PPV). PPV denotes the fraction of correctly raised alarms. In other words, this value permits to evaluate how precise IVIST is in raising alarms with respect to the total number of raised alarms.

$$PPV = \frac{TP}{TP + FP} \tag{1}$$

Detection Rate. The detection rate denotes the fraction of correctly raised alarms over the number of alarms to be raised. In other words, it permits to evaluate the completeness of IVIST in raising alarms.

$$Detection\ Rate = \frac{TP}{TP + FN} \tag{2}$$

Accuracy. The accuracy denotes the ability of the system to correctly raise or not raise alarms over the total number of tested videos.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

F1-score. The F1-score measures the balance between PPV and the Detection Rate.

$$Accuracy = 2 \times \frac{PPV \ * \ Detection\ Rate}{PPV \ + \ Detection\ Rate} \tag{4}$$

6.3. Analysis of results

As said above, we have experimentally evaluated the IVIST system and its underlying EDCAR framework on a set of video sequences instantiating the four target scenarios described in Section 4, namely "Steal of Baggage", "Crowd Activity", "Unattended Baggage", and "Fighting".

Steal of Baggage. As said in Section 4, the "Steal of Baggage" scenario involves two people, and one baggage. Initially, a man, named *Person1*, appears in the scene; he is pulling his baggage, named *Bag2* (Figure 9(a)). Eventually, *Person1* leaves *Bag2* and goes away (Figure 9(b)-9(c)). Then, another man, named *Person2*, grasps *Bag2* (Figure 9(e)) and runs away (Figure 9(f)).

Notice that, the detection module detects the elements appearing in the scene, and classifies persons and baggages (Figure 9(a)-9(e)). Moreover, the tracking module has been capable of tracking the elements (Figure 9(a)-9(e)) by detecting the movements of the persons w.r.t. the objects within the scene. Finally, the understanding module has been able to recognize the suspicious scenario, and to correctly trigger the alarm (Figure9(f)).

The recognition of the Steal of Baggage scenario has been evaluated by using the UCF-Crime dataset. As shown in Figure 10, IVIST obtained the best results in terms of PPV. This is due to the fact that no false positives have been produced. However, even though several false negatives occurred, causing a reduced detection rate, IVIST obtained an accuracy and a F1-score over 70%. Such results can be considered good w.r.t. quality of test videos, since they

have a low-resolution, which deteriorates performances of the external module used for tracking.

Crowd Activity. The "Crowd Activity" scenario involves many people that could initially appear and/or disappear in/from the scene (Figure 11(a)), and then suddenly run away (Figure 11(b)).

Figure 11 shows two frames from a video on which the detection has detected people in the scene, and the tracking module has tracked elements of the context (Figure 11(a)). Finally, the understanding module has been able to recognize

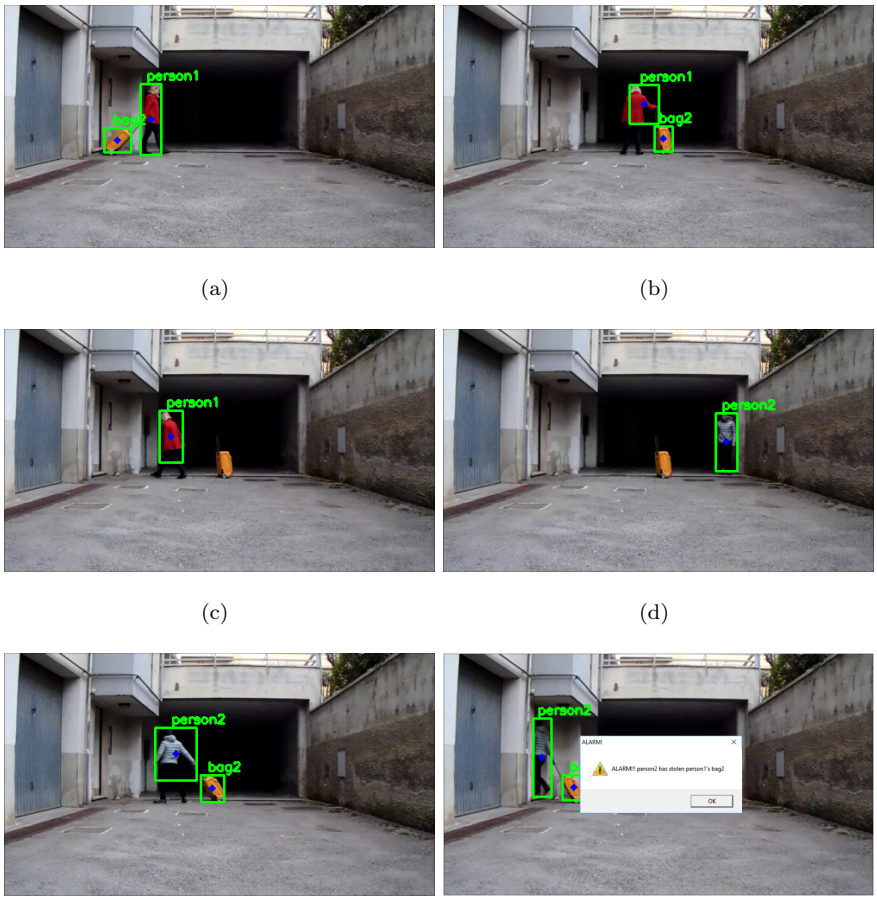


Figure 9: Selected frames from the IVIST execution on the "Steal of Baggage" scenario.

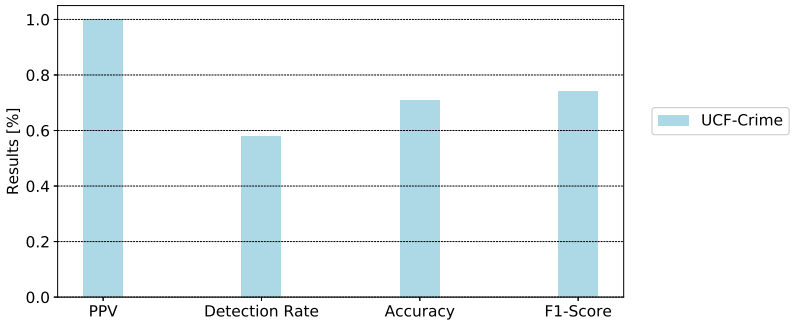


Figure 10: Results on the Steal of Baggage scenario.

(a) (b)

Figure 11: Selected frames from the IVIST execution on the “Crowd Activity” scenario.

the anomalous scenario, and to correctly trigger the alarm (Figure11(b)).

The recognition of the Crowd Activity scenario has been evaluated by using three datasets: Pets2012, UMN-Crowd, and Caviar. As shown in Figure 12, IVIST obtained the best results on the Caviar dataset. This is due to the fact that neither false positives nor false negatives were produced. The performances are good also on the other two datasets. In fact, a maximum detection rate is obtained on the datasets UMN-Crowd and Caviar. With respect to Pets2012, IVIST obtained slightly worse results, since for some videos the framing made the recognition of people particularly complex for the detection and tracking modules. As an example, in one video people walking on a street were framed from behind and from long distance, which tricked the detection module when they suddenly escaped.

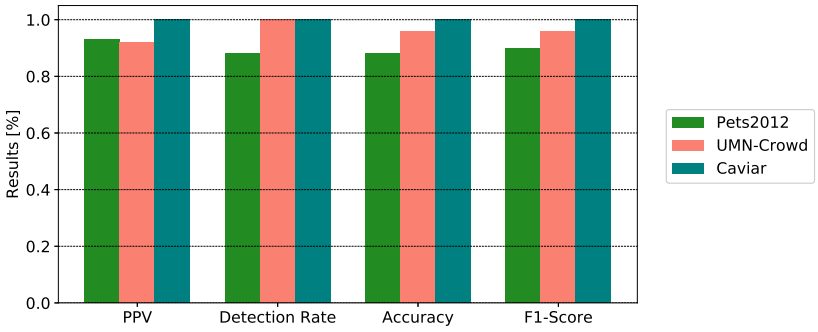


Figure 12: Results on the Crowd Activity scenario.

Unattended Baggage. The "Unattended Baggage" scenario involves one person, and one baggage. Initially, a man, named *Person1*, appears in the scene; he is pulling his baggage, named *Bag2* (Figure 13(c)). Then, *Person1* leaves *Bag2* and goes away (Figure 13(d)-13(e)). Afterwards, an alarm or warning is raised according to a timer setting, which is based on the danger level (Figure 13(f)).

The understanding module is able to distinguish between the two follow-

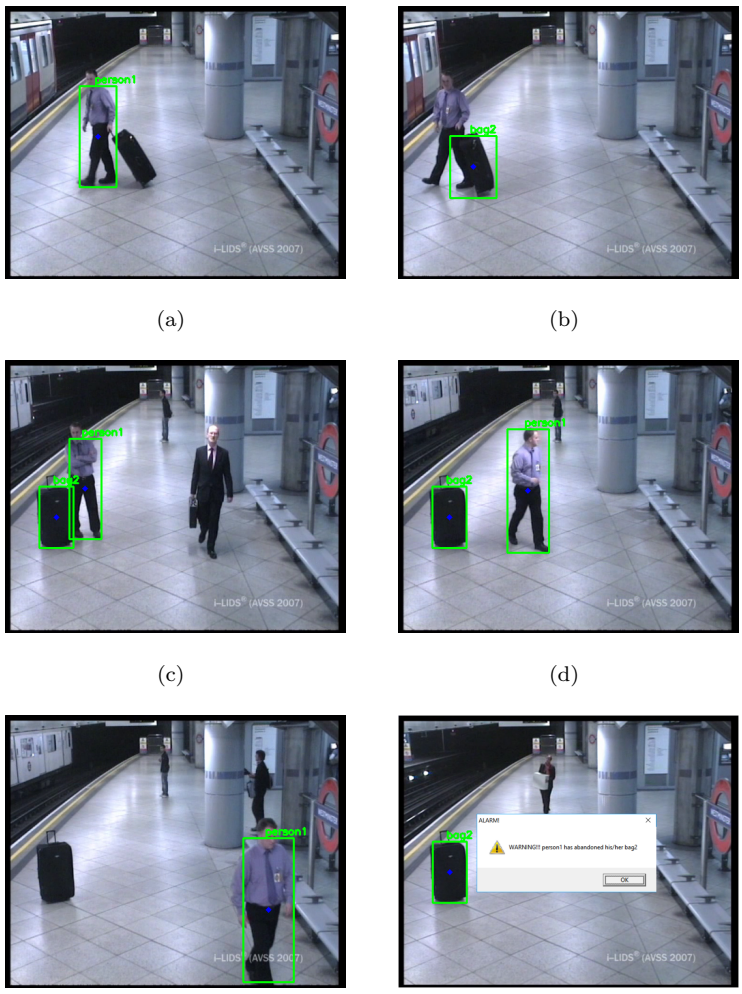


Figure 13: Selected frames from the IVIST execution on the "Unattended Baggage" scenario.

ing specific scenarios: (1) *involuntary baggage abandonment*, and (2) *possible terrorist attack*, by analyzing facial expressions.

It is worth to notice that the module for facial recognition is external to our architecture. It has turned out to be useful in disambiguating scenarios with few persons, whose face happens to be focused in some video frames. The level of confidence has been achieved by performing a pre-processing phase, during which a sampling of video frames has been accomplished, and then computing the ratio of frames in which a given facial expression is recognized.

The recognition of the Unattended Baggage scenario has been evaluated by using three datasets: Pets2006, iLids, and Caviar. As shown in Figure 14, IVIST obtained the best results on the Pets2006 and iLids datasets. This is due to the fact that neither false positives nor false negatives are produced. Since the dataset contains only four videos, even one false negative reduced the performances in terms of detection rate by 33% for the Caviar dataset.

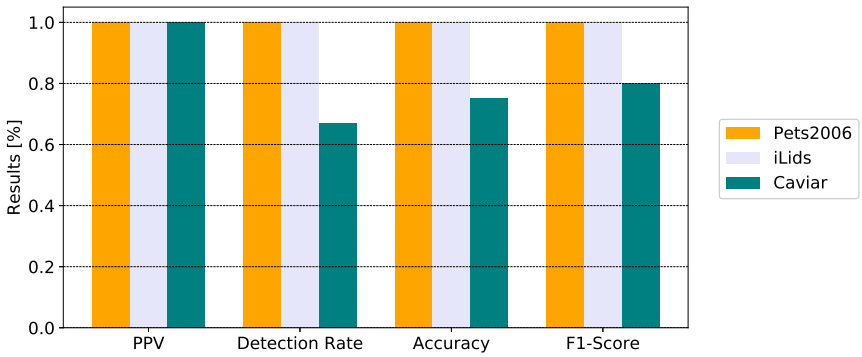
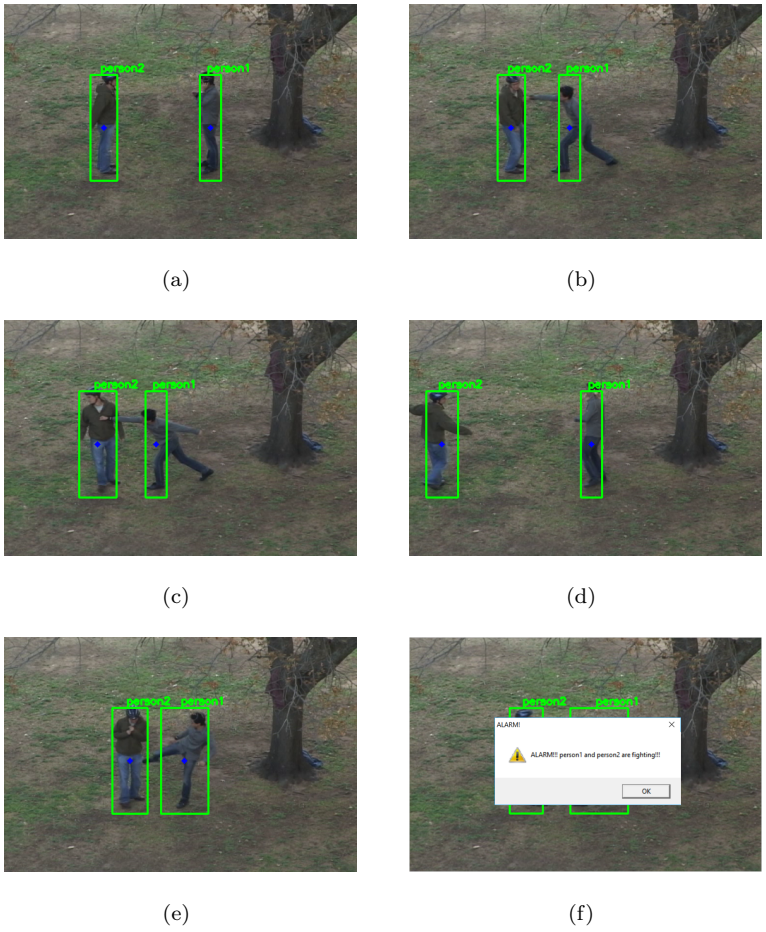


Figure 14: Results on the Unattended Baggage scenario .

Fighting. As said in Section 4, the "Fighting" scenario involves two persons, named *Person1* and *Person2*. They approach each other (Figure 15(a)), and then start hitting and behaving anomalously (Figure 15(b)-15(e)). Also in this case, an alarm will be raised according to the danger level setting (Figure 15(f)).

The understanding module is able to distinguish between the two following



(a) (b) (c) (d) (e) (f)

Figure 15: Selected frames from the IVIST execution on the "Fighting" scenario.

specific scenarios: (1) *possible joke between friends*, and (2) *fighting*, by analyzing facial expressions and the possible presence of dangerous events. Only in the second case an alarm is raised.

The Unattended Baggage is the scenario that has been evaluated with the greatest number of datasets: Behave, UT-Interaction, UCF-Crime, and Caviar. As shown in Figure 16, IVIST obtained the best results with the Behave and UT-Interaction datasets, even if some false negatives were produced on them. Moreover, also for this scenario the UCF-Crime dataset resulted the most critical

one. In general, results show that IVIST achieved good performances in terms of accuracy and F1-score with respect to the scenario complexity, due to necessity to recognize many human-interactions.

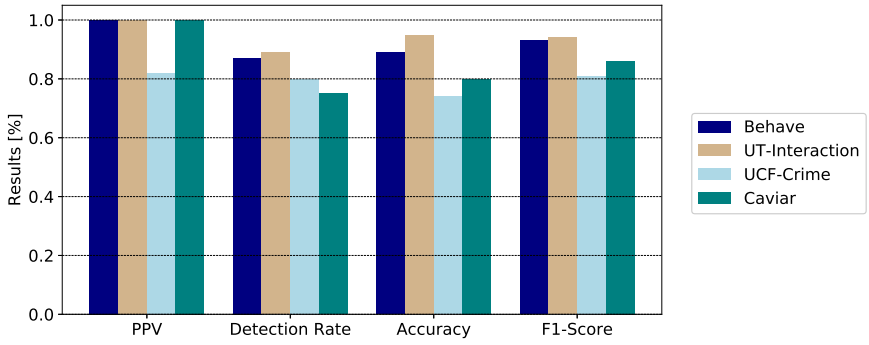


Figure 16: Results on the Fighting scenario.

6.4. Comparison with state-of-the-art solutions

Table 6 compares IVIST and the underlying EDCAR framework with analogous solutions defined in the literature, based on the analyzed scenarios. In particular, in order to prove the effectiveness of our proposal, we considered approaches evaluated on at least one of the datasets we analyzed in our evaluation. In general, it can be observed that the most frequently analyzed scenario is the "Unattended Baggage". Moreover, we can notice that none of the compared solutions have been evaluated against the "Steal of Baggage" scenario.

Experimental results described in the previous section show that IVIST achieves good performances in terms of accuracy and balanced value (F1-score). However, the comparisons with the approaches recalled in Table 6 were each accomplished based on one of these two performance metrics, since none of the compared approaches was evaluated against both metrics. Moreover, from what said above, no comparison was possible on the "Steal of Baggage" scenario. Notice that, although Elhamod & Levine (2012) analyzed this scenario (they called it "Theft of Luggage"), achieving a F1-score of 1, they evaluated their solution

Methodology	Steal of Baggage	Crowd Activity	Unattended Baggage	Fighting
SanMiguel et al. (2009)	✗	✗	✓	✗
Elhamod & Levine (2012)	✗	✗	✓	✓
Lim et al. (2014)	✗	✓	✓	✗
Wang et al. (2018)	✗	✓	✗	✗
IVIST	✓	✓	✓	✓

Table 6: A comparison between our methodology and other related works according to the analyzed video surveillance scenarios.

only on the Caviar dataset, which does not contain positive videos. Even achieving the same result with IVIST, we thought that such a comparison would not provide valuable insights.

Results of comparison are shown in Figure 17. They show an improvement of IVIST with respect to other approaches. In particular, they show better performances on the "Unattended Baggage" scenario with respect to SanMiguel et al. (2009) (Figure 17(a)), Lim et al. (2014) (Figure 17(c)), and Elhamod & Levine (2012) (Figure 17(b)). With respect to the latter, IVIST achieved better performances also on the "Fighting" scenario (Figure 17(b)). Finally, for the "Crowd Activity" scenario IVIST achieved better performances with respect to Lim et al. (2014) (Figure 17(c)), and similar performances with respect to Wang et al. (2018) (Figure 17(d)).

It is worth noting that we have compared with the approaches shown in Table 6, since the first three are similar to our proposal (e.g they are knowledge-based), whereas although the fourth one faces the event recognition as a classification problem, we could somehow compare with it, because it shared the "Crowd Activity" scenario and the UMN-Crowd dataset with respect to our evaluation. In the literature, there are many recent approaches facing the event recognition as a classification problem Zhang et al. (2017). However, they typically stress their evaluations on the recognition of one among a predefined set of actions within each test video, after a training session performed on a big training

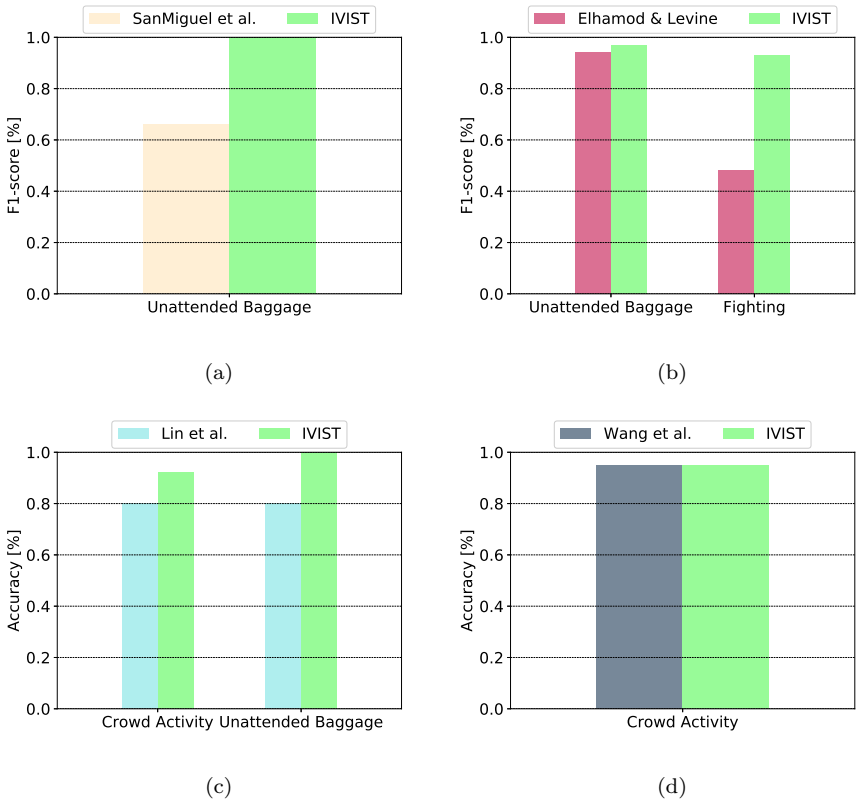


Figure 17: A comparative results based on the F1-score or the Accuracy metrics.

dataset. This kind of evaluation is out of the scope of our proposal that is scenario-based and it prescribes the specification of knowledge concerning target scenarios rather than a training step, which would not be suitable for video-surveillance and other complex scenarios.

6.5. Discussion

As analyzed in the previous section, IVIST generally outperforms other analyzed approaches on the considered scenarios. Such results have been achieved despite the limits of current detection and tracking technologies. In fact, although deep learning is contributing to make detection technologies more precise, they might still miss to immediately detect some objects when they appear

in the scene. To this end, the EDCAR framework has turned out to be robust with respect to this problem, since it provides a scenario modeling paradigm that acts at more an abstract level, making it tolerant with respect to delayed object detections. Moreover, trackers tend to lose person/object IDs in certain situations, such as when subjects temporarily overlap or disappear from the scene. To this end, the modeling mechanisms of EDCAR can limit the impact of tracking failures, though entailing an increased modeling effort.

In conclusion, we can affirm that contextual information can enhance the automatic interpretation of complex video surveillance scenarios, and that the EDCAR framework is able to represent and characterize several types of information in terms of ECR, AR, and GCD, in order to achieve good performances, also in presence of some detection or tracking errors. To this end, since the detection and tracking modules are external to the proposed framework, future improvements to their underlying technologies from the research community can contribute to reduce the complexity of the scenario modeling phase.

7. Conclusions

We have presented EDCAR, a knowledge representation framework capable of describing patterns of knowledge in a video sequence. Based on the proposed framework, we have devised a hierarchical approach that allows to summarize complex video surveillance scenarios. Moreover, we have described the IVIST system, which implements the framework and its underlying approach, by also interacting with some external modules, such as an object detector, a tracker, and a facial expression analyzer.

EDCAR and IVIST have been evaluated on public video datasets, which enabled us to prove their effectiveness with respect to similar solutions presented in the literature. Such results have been achieved also thanks to an additional modeling effort to handle possible errors from external modules for object detection and tracking (YOLO and TLD). To reduce such modeling complexity, in the future we plan to simplify the action rules modeling activity by exploiting

visual interfaces, such as gesture-based interfaces (Deufemia et al., 2011), policy specification interfaces (Giordano & Polese, 2013), and user intent understanding techniques (Caruccio et al., 2015).

As a further future development, we would like to apply the EDCAR framework and the IVIST system to new application domains. As an example, we are currently using EDCAR and IVIST experimentally on the medical domain, and in particular in the monitoring of actions of patients in emergency rooms of hospitals, in order to match the results of video interpretation mechanisms with those of other clinic tests, and promptly detect possible dangerous cases, which might possibly be underrated, especially in busy emergency settings. Another interesting application domain is sport analytics, where proper video summarization mechanisms can support coaches in the detection of gaming strategies, and in the evaluation of errors in the application of gaming strategies.

References

Afsar, P., Cortez, P., & Santos, H. (2015). Automatic visual detection of human behavior: A review from 2000 to 2014. *Expert Systems with Applications*, *42*, 6935–6956.

Aggarwal, J. K., & Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, *43*, 16.

Brand, M., Oliver, N., & Pentland, A. (1997). Coupled hidden markov models for complex action recognition. In *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR)* (pp. 994–999). IEEE.

Caruccio, L., Deufemia, V., & Polese, G. (2015). Understanding user intent on the web through interaction mining. *Journal of Visual Languages & Computing*, *31*, 230–236.

Castro, J., Delgado, M., Medina, J., & Ruiz-Lozano, M. (2011). Intelligent

surveillance system with integration of heterogeneous information for intrusion detection. *Expert Systems with Applications*, *38*, 11182–11192.

Chaquet, J. M., Carmona, E. J., & Fernández-Caballero, A. (2013). A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, *117*, 633–659.

Cohen, I., Sebe, N., Garg, A., Chen, L. S., & Huang, T. S. (2003). Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and image understanding*, *91*, 160–187.

Deufemia, V., Giordano, M., Polese, G., & Tortora, G. (2011). Dialogue-driven search in surveillance videos. In *Proceedings of the 17th International Conference on Distributed Multimedia Systems, (DMS)* (pp. 134–139).

D'Souza, C., Deufemia, V., Ginige, A., & Polese, G. (2018). Enabling the generation of web applications from mockups. *Software: Practice and Experience*, *48*, 945–973.

Ekman, P., & Friesen, W. V. (1978). *Facial action coding system: Investigator's guide*. Consulting Psychologists Press.

Elhamod, M., & Levine, M. D. (2012). Real-time semantics-based detection of suspicious activities in public spaces. In *2012 Ninth Conference on Computer and Robot Vision* (pp. 268–275). IEEE.

Fine, S., Singer, Y., & Tishby, N. (1998). The hierarchical hidden markov model: Analysis and applications. *Machine learning*, *32*, 41–62.

Fisher, R. (2007). BEHAVE: Computer-assisted prescreening of video streams for unusual activities. *The EPSRC project GR S*, *98146*. URL: `http://homepages.inf.ed.ac.uk/rbf/BEHAVE/`.

Fisher, R., Santos-Victor, J., & Crowley, J. (2005). CAVIAR: Context aware vision using image-based active recognition. URL: `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/caviar.htm`.

Ghanem, N., DeMenthon, D., Doermann, D., & Davis, L. (2004). Representation and recognition of events in surveillance video using petri nets. In *Conference on Computer Vision and Pattern Recognition Workshop, (CVPRW)* (pp. 112–112). IEEE.

Giordano, M., & Polese, G. (2013). Visual computer-managed security: A framework for developing access control in enterprise applications. *IEEE Software*, *30*, 62–69.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48.

Hongeng, S., & Nevatia, R. (2001). Multi-agent event recognition. In *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV)* (pp. 84–91). IEEE volume 2.

Hongeng, S., Nevatia, R., & Bremond, F. (2004). Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, *96*, 129–162.

Jiang, H., Drew, M. S., & Li, Z. (2006). Successive convex matching for action detection. In *Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR)* (pp. 1646–1653). IEEE.

Joo, S.-W., & Chellappa, R. (2006). Attribute grammar-based event recognition and anomaly detection. In *Conference on Computer Vision and Pattern Recognition Workshop, (CVPRW)* (pp. 107–107). IEEE.

Joshi, K. A., & Thakore, D. G. (2012). A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, *2*, 44–48.

Kalal, Z., Mikolajczyk, K., Matas, J. et al. (2012). Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, *34*, 1409.

Kim, H., Lee, S., Kim, Y., Lee, S., Lee, D., Ju, J., & Myung, H. (2016). Weighted joint-based human behavior recognition algorithm using only depth information for low-cost intelligent video-surveillance system. *Expert Systems with Applications*, *45*, 131–141.

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*, 107–123.

Lavee, G., Rivlin, E., & Rudzsky, M. (2009). Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *39*, 489–504.

Lim, M. K., Tang, S., & Chan, C. S. (2014). iSurveillance: Intelligent framework for multiple events detection in surveillance videos. *Expert Systems with Applications*, *41*, 4704–4715.

Nevatia, R., Zhao, T., & Hongeng, S. (2003). Hierarchical language-based representation of events in video streams. In *Conference on Computer Vision and Pattern Recognition Workshop, CVPRW* (pp. 39–39). IEEE volume 4.

O'connor, M., Knublauch, H., Tu, S., Grosof, B., Dean, M., Grosso, W., & Musen, M. (2005). Supporting rule system interoperability on the semantic web with swrl. In *International Semantic Web Conference* (pp. 974–986). Springer.

Pantic, M., & Rothkrantz, L. J. M. (2000). Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on pattern analysis and machine intelligence*, *22*, 1424–1445.

Park, E., Han, X., Berg, T. L., & Berg, A. C. (2016). Combining multiple sources of knowledge in deep cnns for action recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision, (WACV)* (pp. 1–8). IEEE.

Pavlovic, V., Rehg, J. M., Cham, T.-J., & Murphy, K. P. (1999). A dynamic bayesian network approach to figure tracking using learned dynamic models. In *Proceedings of the 7th IEEE International Conference on Computer Vision, (ICCV)* (pp. 94–101). IEEE volume 1.

Peterson, J. L. (1981). *Petri net theory and the modeling of systems*. Prentice Hall PTR.

PETS2006 (2006). Pets2006 challenge. In *Proceedings of the 9th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006)* (pp. 47–50). URL: `http://www.cvg.reading.ac.uk/PETS2006/data.html`.

PETS2012 (2012). Pets2012 challenge. URL: `http://www.cvg.reading.ac.uk/PETS2012/a.html`.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

Rodriguez, M. D., Ahmed, J., & Shah, M. (2008). Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *2008 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Ryoo, M. S., & Aggarwal, J. K. (2010). UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). `http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html`. Last accessed: 2018-08-23.

SanMiguel, J. C., Martinez, J. M., & Garcia, Á. (2009). An ontology for event detection and its application in surveillance video. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on* (pp. 220–225). IEEE.

Shan, C., Gong, S., & McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. *Image and vision Computing*, *27*, 803–816.

Sheikh, Y., Sheikh, M., & Shah, M. (2005). Exploring the space of a human action. In *Proceedings of the 10th International Conference on Computer Vision (ICCV)* (pp. 144–149). IEEE.

Sultani, W., Chen, C., & Shah, M. (2018). Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6479–6488). URL: `http://crcv.ucf.edu/cchen/`.

Tani, M. Y. K., Lablack, A., Ghomari, A., & Bilasco, I. M. (2014). Events detection using a video-surveillance ontology and a rule-based approach. In *European Conference on Computer Vision* (pp. 299–308). Springer.

Tran, S. D., & Davis, L. S. (2008). Event modeling and recognition using markov logic networks. In *Computer Vision–ECCV 2008* (pp. 610–623). Springer.

UMN-Crowd (2009). Unusual crowd activity dataset of University of Minnesota. `http://mha.cs.umn.edu/movies/crowdactivity-all.avi`. Last accessed: 2018-08-23.

Valenzise, G., Gerosa, L., Tagliasacchi, M., Antonacci, F., & Sarti, A. (2007). Advanced video and signal based surveillance. *AVSS 2007*, *2*, 21–26. URL: `http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html`.

Wang, L., & Sng, D. (2015). Deep learning algorithms with applications to video analytics for A smart city: A survey. *CoRR*, *abs/1512.03131*.

Wang, T., Qiao, M., Deng, Y., Zhou, Y., Wang, H., Lyu, Q., & Snoussi, H. (2018). Abnormal event detection based on analysis of movement information of video sequence. *Optik*, *152*, 50–60.

Xue, H., Liu, Y., Cai, D., & He, X. (2016). Tracking people in rgbd videos using deep learning and motion clues. *Neurocomputing*, *204*, 70–76.

Zeng, Z., Pantic, M., Roisman, G. I., & Huang, T. S. (2009). A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*, *31*, 39–58.

Zhang, J., Shum, H. P., Han, J., & Shao, L. (2018). Action recognition from arbitrary views using transferable dictionary learning. *IEEE Transactions on Image Processing*, *27*, 4709–4723.

Zhang, S., Wei, Z., Nie, J., Huang, L., Wang, S., & Li, Z. (2017). A review on human activity recognition using vision-based method. *Journal of healthcare engineering*, *2017*.