Extensible Hierarchical Method of Detecting Interactive Actions for Video Understanding

Jinyoung Moon, Junho Jin, Yongjin Kwon, Kyuchang Kang, Jongyoul Park, and Kyoung Park

For video understanding, namely analyzing who did what in a video, actions along with objects are primary elements. Most studies on actions have handled recognition problems for a well-trimmed video and focused on enhancing their classification performance. However, action detection, including localization as well as recognition, is required because, in general, actions intersect in time and space. In addition, most studies have not considered extensibility for a newly added action that has been previously trained. Therefore, proposed in this paper is an extensible hierarchical method for detecting generic actions, which combine object movements and spatial relations between two objects, and inherited actions, which are determined by the related objects through an ontology and rule based methodology. The hierarchical design of the method enables it to detect any interactive actions based on the spatial relations between two objects. The method using object information achieves an F-measure of 90.27%. Moreover, this paper describes the extensibility of the method for a new action contained in a video from a video domain that is different from the dataset used.

Keywords: Action detection, Hierarchical action composition, Generic action, Inherited action, Video understanding.

I. Introduction

Actions, in company with objects, are essential elements for understanding a video semantically, namely *who (objects)* did *what (actions)* in the video. Since 1992, most studies on actions have mainly dealt with the recognition problems found in different videos [1]–[16]. An action recognition method returns the type of a representative action found in a video. The above studies have targeted different types of actions, including gestures, single-person actions, human-human or human-object interactions, and group actions, and have mainly focused on increasing their classification accuracy.

To understand the actions appearing in a video, we need action detection methods supporting both spatial and temporal localization as well as action classification. A video action dataset includes both videos and annotations on the actions contained in the videos. Most video datasets that are widely used for action recognition include well-trimmed videos [17]. Each video usually contains a single representative action to which most frames within the video are related. However, frequently encountered videos, which are obtained from the Internet or produced by the general public through smartphones, are different from the videos collected in the video datasets. Such videos are lengthy and contain multiple actions intersecting temporally or spatially. Moreover, some recent studies on action detection support temporal

Manuscript received Aug. 11, 2016; revised Feb. 9, 2017; accepted Apr. 24, 2017. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0101-15-0266, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis).

Jinyoung Moon (jymoon@etri.re.kr), Yongjin Kwon (scocso@etri.re.kr), and Jongyoul Park (corresponding author, jongyoul@etri.re.kr) are with the SW & Contents Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Junho Jin (junho@etri.re.kr) is with the Hyper-connected Communication Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Kyuchang Kang (kc.kang@kunsan.ac.kr) is with the School of IT Information and Control Engineering, Kunsan National University, Gunsan, Rep. of Korea.

Kyoung Park (kyoung.park@sk.com) is with the Memory System Research Lab., SK Hynix, Ichoen, Rep. of Korea.

localization targeting for untrimmed videos [18], [19] or spatial localization for well-trimmed videos [20].

It is difficult to define all action types and train the models for them in advance before applying them to a specific video application. Existing action recognition and detection methods [1]–[16], [18]–[26] have targeted specific action types closely related to different video domains. The type of action supported by a recognition method is determined based on the video domain of the dataset used. For example, dramas and movies mainly contain human-human interactions and/or human-object interactions, whereas surveillance videos are focused on actions for monitoring objects of interest, such as persons, cars, and entrances. However, current videos include more heterogeneous actions than past ones, often produced by experts in a studio within a limited genre, because such videos are produced through various channels both commercially and personally. Therefore, we need an extensible approach to understanding any videos irrespective of their video domain.

To address these challenges, we propose a domain-extensible hierarchical method for detecting interactive actions. The proposed method extends a knowledge-driven approach whose feasibility was previously tested for six target actions [27]. The proposed method detects both generic composite actions based on dynamic spatial relations (DSR) between two objects and inherited composite actions determined by the related objects. As shown in Fig. 1, the method detects a generic composite action *GoIn(X,Y)* based on *Into(X,Y)* that consists of two atomic actions, *sr_closeTo(X,Y)* and *disappear(X)*, which means *X* is close to *Y*, and *X* then disappears in a moment. If Y is a car or an entrance, *GoIn(X, Y)* is embodied in *GetInto(X, Y)* or *Enter(X, Y)*, respectively.

Consequently, this paper makes the following three major contributions:

- For the extensibility, the proposed method detects a *generic composite action*, which can be an abstract description of a specific action, as well as an *inherited composite action* if defined. It is shown that the detection system detects a generic composite action corresponding to a new action and detects the new action by adding a rule for specializing the generic action with constraints on related objects.
- For representing and reasoning the composite actions associated with moving objects, the proposed method uses ontologies and rules, which are mostly utilized for static hierarchical knowledge. The application of such ontologies and rules enables the method to connect the generic and inherited actions semantically.
- For ten actions intersecting in time and space in untrimmed videos from the ActionNet-VE dataset, the detection method using object information achieves an F-measure of 90.27%.

II. Related Work

Studies on video actions can be divided into action recognition and detection. The goal of action recognition is to classify the type of a representative action contained in a segmented video. Action detection includes spatial and/or temporal localization as well as action recognition.

Studies on action recognition have made great strides in the complexity of recognizable actions and recognition rate. Initially, the KTH [28] and Weizmann [29] datasets

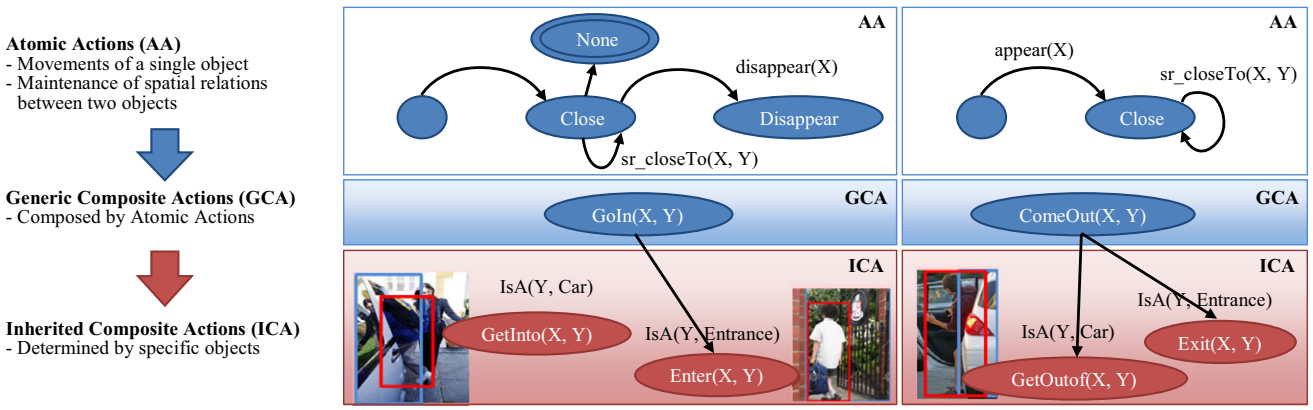


Fig. 1. Concept of extensible hierarchical detection method for interactive actions. Three types of detected actions are (1) an atomic action for describing movements of an object and maintenance of spatial relations between two objects, (2) a generic composite action composed by inference rules with multiple atomic actions, and (3) an inherited composite action from a generic composite action with constraints on the related objects.

were widely used. The videos contain simple single-person actions, such as walking and running, performed by students with a static background. A method using KTH [29] achieved an accuracy of 71.7%, whereas a method using spatio-temporal bag-of-features achieved an accuracy level of 91.8% four years later [5]. Since the development of the KTH and Weizmann datasets, new datasets have included real-world actions obtained from movies, sports, and YouTube videos. HMDB-51 [30] and UCF-101 [31] are middle-scale datasets, and Sports-1M [11] is a large-scale dataset. Accordingly, they are most frequently used nowadays. The method using hand-crafted improved dense trajectory (iDT) features [6], [7] and a method of hybrid representation [8] showed average levels of accuracy of 85.9% and 87.9% on the UCF-101 dataset, respectively. Although some methods using convolutional neural networks (CNNs) [9]–[11] have been proposed, the approaches of two-stream CNNs in [12], [13] started to slightly outperform approaches using hand-crafted features with accuracy levels of 88.0% and 88.6%, respectively. In two-stream CNNs, actions are represented based on appearance using RGB frames and motion through optical flows. Approaches using CNNs with RGB frames, optical flows, and additional iDT [14]–[16] have achieved levels of accuracy of 90.4%, 91.5%, and 93.5%, respectively. In terms of accuracy, they have surpassed approaches using only hand-crafted features. In contrast to the UCF-101 dataset, the best performances of the Sports-1M and HMDB-51 datasets in [13] and [16] are only 73.1% and 69.2%, respectively.

Although there have been fewer efforts on action localization [18]–[26], compared to those of action recognition, action detection has also made considerable progress recently by adopting CNNs and long short-term memories (LSTMs) [18]–[21]. Recent approaches for spatial and temporal action localization have mainly used the UCF-Sports [32], J-HMDB [33], and THUMOS [34] datasets. The UCF-Sports and J-HMDB datasets contain well-trimmed videos, which have also been used for action recognition, and annotations on the spatial region of an action for each frame. On the other hand, the THUMOS dataset, which has been used for temporal localization [18], [19], contains untrimmed videos and annotations on the time interval of different actions. One of the performance measures of action detection is the mean average precision (mAP) at an intersection-over-union (IoU) threshold of δ, which means that a detected action matches the ground truth action if the region of the detected action intersects with the union of regions of the two actions above δ%. For the UCF-Sports dataset, the action detection methods proposed in [22] and [23] achieved mAPs of 54% and 61.6% at an IoU of δ = 50%, respectively. The recent detection methods in [20] and [21] obtained considerable progress in terms of mAPs of 75.8% and 90.5%, respectively. For the J-HMDB dataset, the recent methods in [20] and [21] achieved mAPs of 50.3% and 60.7% at an IoU of δ = 50%, respectively.

Unlike data-driven vision-based studies on actions, sensor-based studies have been conducted on actions based on knowledge-driven approaches. By interpreting sensor data, these studies have aimed to recognize the activities of daily living (ADL) [35] in smart homes [36]–[43], such as *BrushTeeth*, *WashHands*, *MakeTea*, and *TakeMedicine*. The knowledge-driven methods for activity modeling and inference have used domain knowledge and heuristics with logic-based or ontology-based knowledge engineering methodologies. They show the strength in handling a cold start problem compared to data-driven approaches. However, they do not consider the extensibility for new actions because the ADL activities are fixed by sensors installed in a smart home before run-time.

There have been some efforts made to recognize video events [44]–[48] by making use of ontologies and rules represented by the World Wide Web (W3C) standards including the Ontology Web Language (OWL) [49] and Semantic Web Rule Language (SWRL) [50], or other ontology-based formalisms. Although the ontology-based studies have dealt with how to learn rules automatically for composite events in a video [46], [47], they have fixed atomic events in advance, which compose rules for composite events. However, it is difficult to define all necessary atomic events in advance in the real world.

III. Interactive Action Detection

The detection methods consist of temporal localization on atomic actions using a spatial database (DB) including object information, temporal localization on composite actions based on rule-based reasoning, and spatial localization on temporally localized composite actions using object information associated with the actions, each of which is explained in detail in this section.

Figure 2 shows an overview of the detection system that adopts the proposed method. The detection system obtains object information as input, which includes the object's ID, type, start and end frames, and object track with minimum bounding rectangles (MBRs) describing the location and size of the object in each frame. In the object information management module, the detection system parses the object information and then constructs a spatial

https://doi.org/10.4218/etrij.17.0116.0054

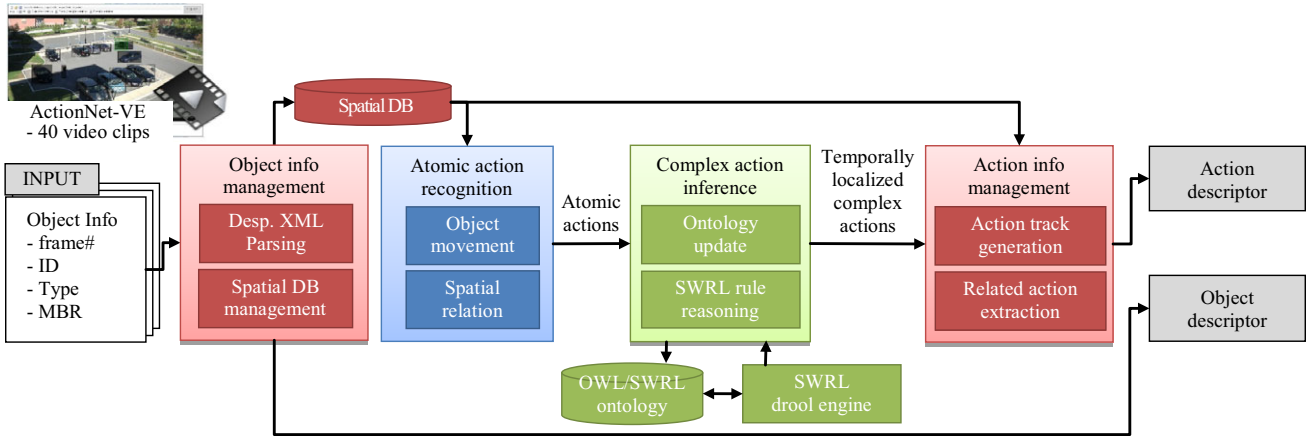


Fig. 2. Overview of the interactive action detection system.

DB including two tables, one for the static object information and the other for the dynamic object track. Using the spatial DB, the atomic action recognition module detects atomic actions describing object movements and the maintenance of spatial relations between two objects. The composite action inference module updates OWL-based assertions on both objects and detected atomic actions, and then applies reasoning with the defined SWRL rules. The composite action inference module returns the time intervals of detected composite actions with the start and end frames. The action information management module generates the action tracks including the MBRs of the action within the time interval for each action by using the MBRs of the related objects. Finally, the system returns two descriptors on the detected actions and their associated objects.

In this paper, the implemented system aims at detecting ten composite actions, as shown in Fig. 3. Among them, the four generic composite actions *GoIn*, *ComeOut*, *Carry*, and *Ride* are related with *into*, *out of*, *with*, and *on* spatial relations, respectively. The six inherited composite actions are *GetInto* and *Enter* (derived from *GoIn*), *GetOutof* and *Exit* (derived from *ComeOut*), and *RideBoard* and *RideBike* (derived from *Ride*).

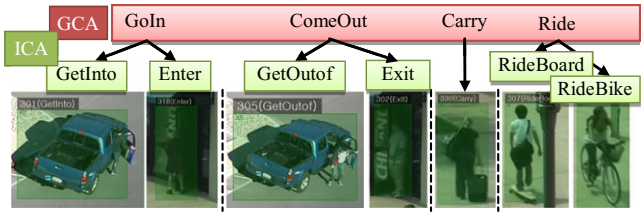


Fig. 3. Ten target actions: four generic composite actions (GCA), composed of object movements and spatial relations between two objects, and six inherited composite actions (ICA), determined by object type.

1. Hierarchical Actions

An action represents a visually meaningful state of a single object or between two objects. Each action has the time interval with the start and end frames and action track with MBRs within the interval. The proposed method handles the two types of actions, atomic and composite actions.

Atomic actions. An atomic action is the smallest component of action that cannot be separated into smaller ones. The atomic action represents a visual state maintained within its time interval. The atomic actions related to movements of an object are *appear*, *disappear, move,* and *stop.* The atomic actions based on the maintenance of spatial relation between two objects include *sr_closeTo*, *sr_with*, and *sr_on*.

Composite actions. In contrast to atomic actions, a composite action is composed of multiple atomic actions or another composite action with constraints on the related objects. By combining multiple atomic actions, a generic composite action represents a complex visual state. For example, a *carry* and *ride* composite action mean that a person is moving and the person holds a specific spatial relation with an object, such as *with* and *on*, respectively. A generic composite action can also represent the change of spatial state between the subject and object, such as a *GoIn* or *ComeOut action.* By putting constraints on the objects related to a generic composite action, an inherited composite action provides more specific visual information than its generic composite action.

2. Temporal Localization on Atomic Actions

The time intervals of *appear* and *disappear* actions related to an object are determined by the first and last frames in which the objects are seen in the screen.

For *move* and *stop* actions, the temporal localization method decides whether an object x moves between two consecutive sampled frames, k and $k+1$, as shown in (1) and (2) with a *move* threshold of δ_{move}. Similarly, a stop action is determined using a *stop* threshold of δ_{stop}. The time interval of a *move* action is calculated by aggregating consecutive sampled frames while the object moves.

$$\frac{\text{distance}\left(\text{centroid}(\text{MBR}(x,k)), \text{centroid}(\text{MBR}(x,k))\right)}{\text{average}\left(\text{width}(\text{MBR}(x,k)), \text{height}(\text{MBR}(x,k+1))\right)} > \delta_{\text{move,}} \quad (1)$$

$$\frac{\text{distance}\left(\text{centroid}(\text{MBR}(x,k)), \text{centroid}(\text{MBR}(x,k))\right)}{\text{average}\left(\text{width}(\text{MBR}(x,k)), \text{height}(\text{MBR}(x,k+1))\right)} < \delta_{\text{stop.}} \quad (2)$$

The other atomic actions are detected using the spatial relation functions on two MBRs of a single object between two consecutive sampled frames, which are used for atomic actions for describing movements of a single object, or on two MBRs of two objects in a frame, which are used for atomic actions based on the DSRs. The spatial relation functions between two geometric values *g1* and *g2* or for a geometric value *g*, such as *intersects(g1, g2), touches(g1, g2), contains(g1, g2), distance(g1, g2)* and *centroid(g)*, are provided by the database management system.

The time interval of a *sr_closeTo(X, Y)* action is calculated by aggregating consecutive sampled frames, where the two MBRs of *X* and *Y* intersect using the *intersects(g1, g2)* function.

The *sr_with* and *sr_on* actions are a special case of *sr_closeTo*. For *sr_with*, the temporal localization method decides whether the two MBRs of *X* and *Y* intersect in the frame, whether the area of the MBR of *X* is larger than the area of the MBR of *Y*, and whether the MBR of *X* is located higher than the MBR of *Y*. For *sr_on(X, Y)*, the temporal localization method decides whether the two MBRs of *X* and Y intersect in the fame and whether the centroid of the MBR of *X* is located higher than the centroid of the MBR of *Y*. The consecutive sampled frames that satisfy the defined conditions of *sr_with* and *sr_on* are aggregated for the time interval of each atomic action, respectively.

To remove false positive actions that result from occlusions between two objects briefly sliding by each other, the temporal localization method for an atomic action sets the minimum duration for each atomic action except for the *appear* and *disappear* actions. For example, a false positive *carry* action can be detected by an occlusion between one person and a bag owned by the other person briefly passing by the person if there is no restriction on the minimum duration for *sr_with* atomic action.

3. Temporal Localization on Composite Actions

As shown in Fig. 4, the temporal localization method for composite actions is divided into two parts, the part defining the terms and rules for detecting composite actions during the design period and the part automatically generating instances related to composite actions through rule-based reasoning during the execution time. A TBox contains terms representing objects and actions, which are defined with OWL classes and object properties in the OWL ontology file. Using the terms defined in the TBox, an RBox includes OWL/SWRL rules for detecting composite actions. At run-time, an ABox contains asserted facts represented by OWL instances, which are inferred from object information and detected atomic actions with their time intervals. The rule-based reasoning engines use TBox, RBox, and ABox for inferring instances of object properties representing composite actions. After rule-based reasoning, the temporal localization method updates the ABox with inferred values and then lists the temporally localized composite actions.

A. Defining Terms

There are two types of terms used for defining rules: terms representing objects defined with OWL classes and

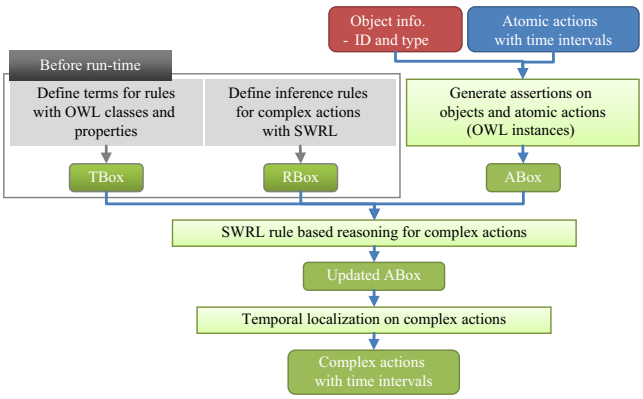


Fig. 4. Procedure of temporal localization on composite actions consisting of manually defining the terms and rules before run-time and generating assertions using object information and detected atomic actions with their time intervals, reasoning the rules for composite actions, and obtaining temporally localized composite actions with inferred instances.

https://doi.org/10.4218/etrij.17.0116.0054

terms representing actions defined with OWL object properties.

To represent the type of objects occurring in videos, we define the *Object* class as the subclass of *Thing*, because the top class in the OWL ontology is Thing. As shown in Fig. 5, the Object class has two disjoint subclasses, *Person* and *NotPerson* classes. The *NotPerson* class contains *Belongings*, *Entrance*, and *Vehicle* subclasses. The gray-colored leaf-node classes represent the types of objects occurring in a video.

The *Duplication* class is defined for representing an object for a specific period. Contrary to OWL classes, which are defined in typical ontologies for a text-based knowledge base, objects in a video have dynamic properties, such as their size and location at a particular time, their spatial relations with other objects, and related actions. The properties are changeable in time and space. Therefore, the Object class is not suitable for representing a moving object in a video. Because the objects perform a certain action for a specific period, this paper designs a Duplication class for describing an object during a particular period.

Corresponding to an action, this paper defines an OWL object property between the Object and Duplication classes or between two Duplication classes, as shown in Fig. 6. There is a *hasDuplication* object property between the *Object* and *Duplication* classes. Four atomic actions describing movements of a single object are represented by the object properties between the *Object* and *Duplication* classes. The three DSR-based atomic actions and ten composite actions performed by two objects are represented by the object properties between the two Duplication classes. The *startTime* and *endTime* data properties represent the time interval of the actions, which are the basic properties all actions should have.

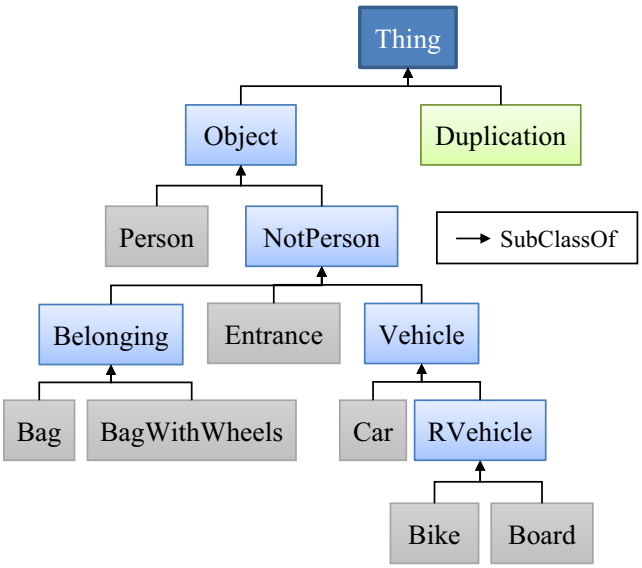


Fig. 5. Defined OWL classes including the Object class, for describing objects associated with actions in a video, and Duplication class, for describing an object during a certain period.

B. Defining Rules

This paper adopts OWL and SWRL for knowledge representation and inference because OWL is suitable for implementing restrictions on the type of related objects for an inherited composite action, and there are several inference engines supporting OWL and SWRL, the W3C standards.

Table 1 shows the OWL/SWRL based rules for ten composite actions. For example, a *GoIn* action is detected if there are the three intersecting atomic actions, *sr_closeTo(X,Y)*, *disappear(X)*, and *stop(Y)*, the interval of *stop(Y)* is longer than T_{stop}, and the difference between the end frame number of *sr_closeTo(X,Y)* and the frame number of *disappear(X)* is less than or equal to the sampling rate. The six rules for inherited composite actions defined by their generic composite actions and constraints on the type of object *Y*.

SWRL rules need multiple comparisons for determining whether there is an intersection between time intervals because OWL/SWRL provides binary predicates. To determine whether there is an intersection between two time intervals [*st1*, *et1*] and [*st2*, *et2*], two comparisons are required, as shown in (3). They are implemented using swrlb:lessThanOrEqual, which is a built-in SWRL function.

$$st_1 \le et_2 \;\&\&\; st_2 \le et_1. \tag{3}$$

4. Spatial Localization on Composite Actions

After rule-based reasoning with TBox, RBox, and ABox, the inferred instances representing composite actions are generated and stored in the OWL ontology. Using the time interval of a temporally localized composite action and the related object IDs obtained from the ontology and their object tracks stored in the spatial DB, the spatial localization method generates the action tracks including MBRs during the time interval of the action.

IV. Experiments and Results

For the experiments, this paper tested the detection method using 40 video clips and their annotation files in

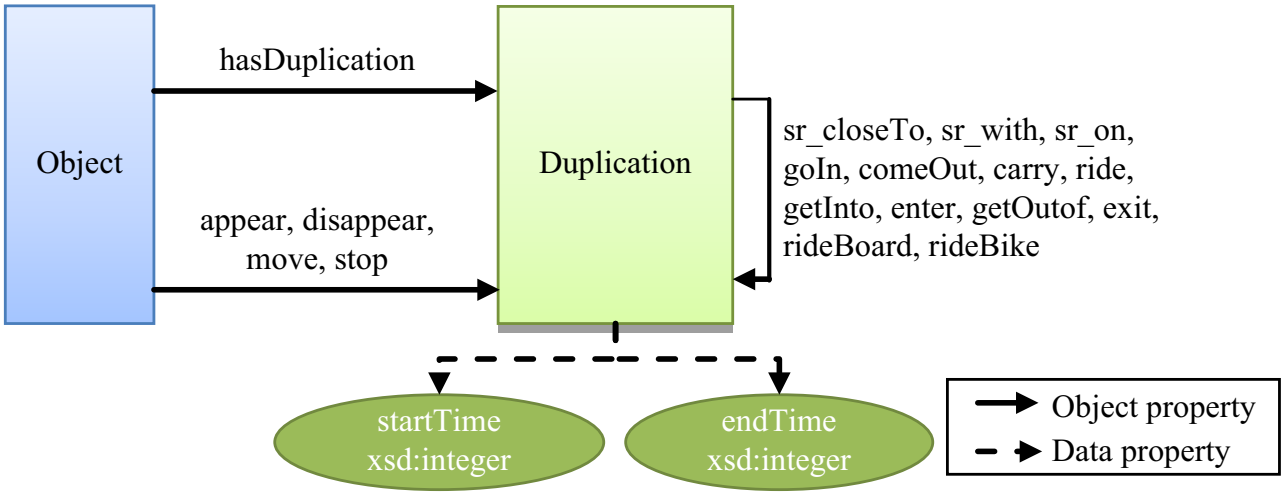


Fig. 6. Defined OWL object properties for representing atomic and composite actions and OWL data properties for describing basic properties of both actions and the start and end frames.

Table 1. OWL/SWRL based rules for ten composite actions.

Action	OWL/SWRL Rules
GoIn	Person(?*s1*)^ hasDuplication(?*s1*,?*d1*)^ hasDuplication(?*s2*,?*d2*)^ sr_closeTo(?*d1*,?*d2*)^ disappear(?*s1*,?*d3*)^ endTime(?*d1*,?*et1*)^ endTime(?*d3*,?*et3*)^ swrlb:subtract(?*mt1*,?*et3*,?*et1*)^ swrlb:(?*mt1*, samplingRate)^ startTime(?*d1*,?*st1*)^ stop(?*s2*,?*d4*)^ startTime(?*d4*,?*st4*)^ endTime(?*d4*,?*et4*)^ swrlb:subtract(?*inV*,?*et4*,?*st4*)^ swrlb:greaterThan(?*inV*, min_{stop}) ^ swrlb:lessThanOrEqual(?*st1*,?*et4*)^ swrlb:lessThanOrEqual(?*st4*,?*et1*) → GoIn(?*d1*,?*d2*)
GetInto	GoIn(?*d1*,?*d2*)^ hasDuplication(?*s2*,?*d2*)^ Car(?*s2*) → GetInto(?*d1*,?*d2*)
Enter	GoIn(?*d1*,?*d2*)^ hasDuplication(?*s2*,?*d2*)^ Entrance(?*s2*) → Enter(?*d1*,?*d2*)
ComeOut	Person(?*s1*)^ hasDuplication(?*s1*,?*d1*)^ hasDuplication(?*s2*,?*d2*)^ appear(?*s1*,?*d3*)^ stop(?*s2*,?*d4*)^ sr_closeTo(?*d1*,?*d2*)^ startTime(?*d1*,?*st1*)^ startTime(?*d3*,?*st3*)^ swrlb:equal(?*st1*,?*st3*)^ endTime(?*d1*,?*et1*)^ startTime(?*d4*,?*st4*)^ endTime(?*d4*,?*et4*)^ swrlb:lessThanOrEqual(?*st4*,?*st1*)^ swrlb:lessThanOrEqual(?*et1*,?*et4*) → ComeOut(?*d1*,?*d2*)
GetOutof	ComeOut(?*d1*,?*d2*)^ hasDuplication(?*s2*,?*d2*) ^ Car(?*s2*) → GetOutOf(?*d1*,?*d2*)
Exit	ComeOut(?*d1*,?*d2*)^ hasDuplication(?*s2*,?*d2*)^ Entrance(?*s2*) → Exit(?*d1*,?*d2*)
Carry	Person(?*s1*)^ hasDuplication(?*s1*,?*d1*)^ hasDuplication(?*s2*,?*d2*)^ sr_with(?*d1*,?*d2*)^ startTime(?*d1*,?*st1*)^ endTime(?*d1*,?*et1*)^ move(?*s1*,?*d3*)^ move(?*s2*,?*d4*)^ startTime(?*d3*,?*st3*)^ startTime(?*d4*,?*st4*)^ endTime(?*d3*,?*et3*)^ endTime(?*d4*,?*et4*) ^ swrlb:lessThanOrEqual(?*st1*,?*et4*)^ swrlb:lessThanOrEqual(?*st4*,?*et1*)^ swrlb:lessThanOrEqual(?*st1*,?*et3*)^ swrlb:lessThanOrEqual(?*st3*,?*et1*)^ swrlb:lessThanOrEqual(?*st3*,?*et4*)^ swrlb:lessThanOrEqual(?*st4*,?*et3*) → Carry(?*d3*,?*d4*)
Ride	Person(?*s1*)^ hasDuplication(?*s1*,?*d1*)^ hasDuplication(?*s2*,?*d2*)^ sr_on(?*d1*,?*d2*)^ move(?*s1*,?*d3*)^ startTime(?*d1*,?*st1*)^ endTime(?*d1*,?*et1*)^ startTime(?*d3*,?*st3*)^ endTime(?*d3*,?*et3*)^ swrlb:lessThanOrEqual(?*st3*,?*et1*)^ swrlb:lessThanOrEqual(?*st1*,?*et3*) → Ride(?*d3*,?*d2*)
RideBike	Ride(?*d1*,?*d2*)^ hasDuplication(?*s2*,?*d2*)^ Bike(?*s2*) → RideBike(?*d1*,?*d2*)
RideBoard	Ride(?*d1*,?*d2*)^ hasDuplication(?*s2*,?*d2*)^ Board(?*s2*) → RideBoard(?*d1*,?d2)

ActionNet-VE [51], which extended VIRAT Ground 2.0 [52] by adding generic and inherited composite actions and objects related to the actions. VIRAT ground 2.0 includes actions intersecting each other in time and space, as shown in Fig. 7. To reduce the computation time for detecting actions, object MBRs of frames sampled from every ten original frames are used in the implemented system. As shown in Table 2, the distribution of actions in ActionNet-VE is unbalanced because the videos were recorded in the real world. Therefore, the total precision, recall, and F-measure were calculated based on their weighted summation for the ten actions using the weight of each action proportional to its occurrences in the video clips, and not based on their average.

https://doi.org/10.4218/etrij.17.0116.0054

To evaluate the performance of the proposed detection system, this paper compared the ground truth with the system results using the matching criterion described in [41]. The matching criterion assumes that action *A* matches action *B* if the type of *A* is the same with that of *B*, the time intervals of A and B intersect with an IoU of δ = 10%, and two MBRs of *A* and *B* intersect each other with an IoU of δ = 10%.

As shown in Table 2, the proposed system achieved a precision of 88.30%, recall of 92.40%, and F-measure of 90.27% for all ten composite actions. The proposed system achieved higher precision, recall, and F-measure on person–vehicle interactions including *GetInto* and *GetOutof* than those on person–entrance interactions including *Enter* and *Exit*. Because some entrances were near some pavements, the proposed system considered going into or coming out of the pavements as entering or exiting the nearby entrances, which deteriorated the precision and recall on the person–entrance interactions. In addition, the proposed system had a higher precision and recall on *RideBike* than those on *RideBoard*. Because boards are smaller and thinner than bikes, occlusions decreasing the precision and recall occurring more frequently in the case of boards than bikes.

Table 3 shows that the proposed method outperformed the state-of-the-art methods on two typical human–vehicle interactions, *GetInto* and *GetOutof*, from the VIRAT Ground 2.0. Table 3 compares the precision and recall of the proposed method with the method using the context information that describes the state of the vehicle and person [53].

To show the extensibility of the proposed method, this paper conducted a test on a newly-added action, *DiveInto*, as shown in Fig. 8. When the related objects existed in the OWL ontology, the system successfully detected the *GoIn* action corresponding to the *DiveInto* action without any changes in the system. The system was able to detect both the *DiveInto* and *GoIn* actions after the system added a SWRL rule to specialize the *GoIn* action with the restriction of the type of related objects, which is similar to the rules for specializing generic actions listed in Table 1.

Fig. 7. Screenshots of ground truth and detected actions in a test video.

Table 2. Detection performance of the proposed method.

Action	Weight	Precision (%)	Recall (%)	F-measure (%)
GoIn	0.147	87.60	90.18	88.87
GetInto	0.037	94.44	100.00	97.14
Enter	0.110	84.71	86.91	85.79
ComeOut	0.102	84.21	92.31	88.07
GetOutof	0.029	87.88	95.46	91.51
Exit	0.073	82.26	91.07	86.44
Carry	0.389	89.18	94.61	91.82
Ride	0.056	95.24	90.70	92.91
RideBike	0.033	100.00	92.00	95.83
RideBoard	0.024	89.47	88.89	89.18
Total	1.00	**88.30**	**92.40**	**90.27**

Table 3. Comparison with state-of-the-art method for typical human–vehicle interactions on (precision/recall).

Action	[53]	Our method
GetInto	70.8/49.09	94.44/100.0
GetOutof	57.74/51.56	87.88/95.46

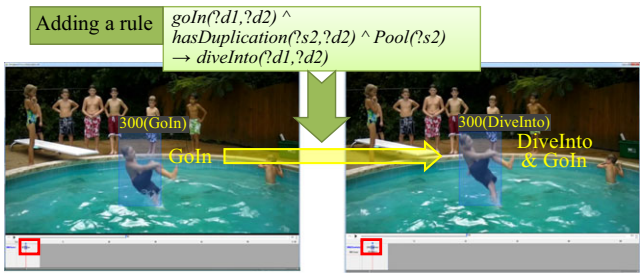


Fig. 8. Detection of a newly added action from a different domain, *DiveInto* using a generic and inherited composite action.

V. Discussion

Compared to most data-driven methods for detecting actions, the proposed method considers the generic properties between actions based on the dynamic spatial relations. The data-driven methods that can detect *Enter* and *GetInto* actions cannot detect *DiveInto* actions before learning their model on *DiveInto* actions. There is no relationship between *DiveInto*, *Enter*, and *GetInto* actions to the data-driven methods, whereas the proposed method recognizes them as actions inherited from the *GoIn* action. The precondition is the ontology should include all object types before run-time. The object types shown in Fig. 5 are enough only to detect the ten actions from ActionNet-VE. We plan to extend the ontology with 200 basic-level object categories provided by ImageNet for the object detection task in 2016 [54]. In addition, we can extend the detection system with a module that finds new object types from the object information and adds it to the ontology at run-time before action inference.

To understand interactions between two objects in all open-domain videos, the detection system needs to extend generic composite actions and their related atomic actions according to the dynamic spatial relations. We examined spatial prepositions and adverbs of English and identified 27 dynamic spatial relations. We are adding generic composite actions, such as *PickUp* and *PutDown* with new atomic actions, such as *move_up* and *move_down*.

Occlusions occurring in 2D videos reflecting the 3D real world distort the spatial relations. Although the detection system sets the minimum length of time interval for each atomic action for excluding false positive actions, as mentioned in Section III-2, the setting cannot resolve all troublesome cases caused by occlusions. For example, an occlusion between a person and a bag owned by a companion that is very close to the person can cause a false positive *carry* action if they move together and the occlusion continues for a longer period than the minimum duration for a *sr_with* atomic action. For resolving this problem, the additional method of modeling group interaction between multiple persons and objects is required.

VI. Conclusion

Actions performed by objects are fundamental elements for semantically understanding various video sequences. Studies do exist for targeting representative actions contained in well-segmented videos. Although some data-driven methods for action recognition and detection have achieved a superior performance of above 90% for fixed target actions, they do not consider the extensibility of their model for new actions. Therefore, a method for detecting heterogeneous actions contained in untrimmed videos is required for a semantic understanding of open-domain videos.

This paper proposed an extensible hierarchical method for detecting generic and inherited composite actions between two objects. A generic composite action consists of atomic actions describing movements of a single object and the state maintenance of spatial relations between two objects for a particular period. A generic composite action can be specialized by multiple inherited composite actions with constraints on objects associated with the actions. Because the generic actions represent spatio-temporal relations between two objects within a video, one of the generic actions can interpret any interactive actions between two objects, irrespective of the video domain.

Using the object information, the methods achieved an F-measure of 90.27% with an IoU of δ = 10% for ten composite actions in the ActionNet-VE dataset. In addition, it was shown that the method detected a new action, *DiveInto*, which was contained in a video whose domain differs from the domain of videos containing the ten actions above. The proposed system can detect a *DiveInto* action as a *GoIn* action without any change, and detect a *DiveInto* action after adding a new rule for the *DiveInto* action.

The extensibility of the proposed method is significant when it is adopted for video analysis in the field. Nowadays, actions are more heterogeneous than before because videos are also produced by the general public using their personal smartphones. In addition, it is impossible to define all actions in advance before applying the method to a particular video. Therefore, the proposed method, together with data-driven detection methods achieving a superior performance, can be used for semantic video understanding in a complementary manner.

References

[1] G. Lavee, E. Rivlin, and M. Rudzsky, "Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurrences in Video," *IEEE Trans. Syst., Man, Cybern., Syst. -Part C*, vol. 39, no. 5, Sept. 2009, pp. 489–504.

[2] R. Poppe, "A Survey on Vision-Based Human Action Recognition," *Image Vision Comput.*, vol. 28, no. 6, June 2010, pp. 976–990.

https://doi.org/10.4218/etrij.17.0116.0054

[3] J.K. Aggarwal and M.S. Ryoo, "Human Activity Analysis: A Review," *ACM Comput. Surv.*, vol. 43, no. 3, Apr. 2011, pp. 1–43.

[4] D. Weinland, R. Ronfard, and E. Boyer, "A Survey of Vision-based Methods for Action Representation, Segmentation, and Recognition," *Comput. Vis. Image Understanding.*, Feb. 2011, pp. 224–241.

[5] I. Laptev et al., "Learning Realistic Human Actions from Movies," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Anchorage, Alaska, June 23–28, 2008, pp. 1–8.

[6] H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," *IEEE Int. Conf. Comput. Vision*, Sydney, Australia, Dec. 1–8, 2013, pp. 3551–3558.

[7] H. Wang and C. Schmid, "LEAR-INRIA Submission for the THUMOS Workshop," *Int. Conf. Comput. Vision, Workshop Action Recogn. Large Number Classes*, Sydney, Australia, Dec. 1–8, 2013.

[8] X. Peng et al., "Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice," *Comput. Vis. Image Underst.*, vol. 150, Sept. 2016, pp. 109–125.

[9] M. Baccouche et al., "Sequential Deep Learning for Human Action Recognition," *Int. Workshop Human Behav. Underst.*, Amsterdam, Netherlands, Nov. 16, 2011, pp. 29–39.

[10] S. Ji et al., "3D Convolutional Neural Networks for Human Action Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no.1, Jan. 2013, pp. 221–231.

[11] A. Karpathy et al., "Large-Scale Video Classification with Convolutional Neural Networks." *IEEE Conf. Comput. Vis. Pattern Recogn.*, Columbus, USA, June 24–27, 2014, pp. 1725–1732.

[12] K. Simonyan, "Two-Stream Convolutional Networks for Action Recognition in Videos," *Int. Conf. Neural Inform. Process. Syst*., Montreal, Canada, Dec. 8–13, 2014, pp. 568–576.

[13] J. Ng et al., "Beyond Short Snippets: Deep Networks for Video Classification," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Boston, USA, June 7–12, 2015, pp. 4694–4702.

[14] L. Wang, Y. Qiao, and X. Tang, "Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Boston, USA, June 7–12, 2015, pp. 4305–4314.

[15] D. Tran et al., "Learning Spatiotemporal Features with 3D Convolutional Networks," *IEEE Int. Conf. Compt. Vis.*, Santiago, Chile, Dec. 13–16, 2015, pp. 4489–4497.

[16] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional Two-Stream Network Fusion for Video Action Recognition," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Las Vegas, USA, June 26–July1, 2016, pp. 1933–1941.

[17] J.M. Chaquet, E.J. Carmona, and A. Fernández-Caballero, "A Survey of Video Dataset for Human Action and Activity Recognition," *Comput. Vis. Image Understanding (CVIU)*, vol. 117, no. 6, June 2013, pp. 633–659.

[18] Z. Shou, D. Wang, and S.-F. Chang, "Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Las Vegas, USA, June 26–July 1, 2016, pp. 1049–1058.

[19] S. Yeung, "Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos," Preprint, submitted July 31, 2015. http://arxiv.org/abs/1507.05738v2.

[20] G. Gkioxari and J. Malik, "Finding Action Tubes," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Boston, USA, June 7–12, 2015, pp. 759–768.

[21] P. Weinzaepfel et al., "Learning to Track for Spatio-Temporal Action Localization," *IEEE Int. Conf. Compt. Vis.*, Santiago, Chile, Dec. 13–16, 2015, pp. 3164–3172.

[22] J. Gall et al., "Hough Forests for Object Detection, Tracking, and Action Recognition," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 33, no. 11, Nov. 2011, pp. 2188–2202.

[23] S.-C. Cheng, K.-Y. Cheng, and Y.-P. Chen, "GHT-Based Associative Memory Learning and Its Application to Human Action Detection and Classification," *Pattern Recogn.*, vol. 46, no. 11, Nov. 2013, pp. 3117–3128.

[24] S. Ma. et al., "Action Recognition and Localization by Hierarchical Space-time Segments," *IEEE Int. Conf. Compt. Vis.*, Sydney, Australia, Dec. 3–6, 2013, pp. 2744–2751.

[25] T. Lan, Y. Wang, and G. Mori, "Discriminative Figure-centric Models for Joint Action Localization and Recognition," *IEEE Int. Conf. Compt. Vis.*, Barcelona, Spain, Nov. 6–13, 2011, pp. 2003–2010.

[26] Y.S. Sefidgar et al., "Discriminative Key-component Models for Interaction Detection and Recognition," *Comput. Vis. Image Understanding.*, vol. 135, no. C, June 2015, pp. 16–30.

[27] J. Moon et al., "A Knowledge-Driven Approach to Interactive Event Recognition for Semantic Video Understanding," *Int. Conf. IT Convergence Security*, Prague, Czech Rep., Sept. 26–29, 2016, pp. 37–39.

[28] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: A Local SVM Approach," *Int. Conf. Pattern Recogn.*, Cambridge, UK, Aug. 23–26, 2004, pp. 32–36.

[29] M. Blank et al., "Actions as Space-Time Shapes," *IEEE Int. Conf. Compt. Vis.*, Beijing, China, Oct. 17–21, 2005. pp. 1395–1402.

[30] H. Kuehne et al., "HMDB: A Large Video Database for Human Motion Recognition," *IEEE Int. Conf. Compt. Vis.*, Barcelona, Spain, Nov. 6–13, 2011, pp. 2556–2563.

[31] K. Soomro, A.R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild," Center for Research in Computer Vision, UCF, Orlando, Tech. Prep. CRCV-TR-12-01, Nov. 2012.

[32] K. Soomro and A.R. Zamir, "Action Recognition in Realistic Sports Videos, Computer Vision in Sports," *Advances in Comput. Vis. Pattern Recogn.*, Springer International Publishing, Jan. 2015, pp. 181–208.

[33] H. Jhuang et al., "Towards Understanding Action Recognition," *IEEE Int. Conf. Compt. Vis*., Sydney, Australia, Dec. 3–6, 2013, pp. 3192–3199.

[34] Y.-G. Jiang et al., *THUMOS Challenge 2014*, Center for Research in Computer Vision, UCF, 2014, Accessed Aug. 8, 2016. http://crcv.ucf.edu/THUMOS14/

[35] A.B. James, "Activities of Daily Living and Instrumental Activities of Daily Living," in *Willard and Spackman's Occupational Therapy*, Philadelphia, USA: Wolters Kluwer Health/Lippincott Williams & Wilkins, 2014.

[36] L. Chen, C.D. Nugent, and H. Wang, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, June 2012, pp. 961–974.

[37] D. Riboni and C. Bettini, "OWL 2 Modeling and Reasoning with Complex Human Actions," *Pervasive Mobile Comput.*, vol. 7, no. 3, 2011, pp. 379–395.

[38] I.H. Bae "An Ontology-Based Approach to ADL Recognition in Smart Homes," *Future Gener. Comput. Syst.* vol. 33, Apr. 2014, pp. 32–41.

[39] G. Okeyo, L. Chen, and H. Wnag, "Combining Ontological and Temporal Formalisms for Composite Activity Modelling and Recognition in Smart Homes," *Future Gener. Comput. Syst.*, vol. 39, Oct. 2014, pp. 29–43.

[40] G. Meditskos, S. Dasiopoulou, and I. Kompatsiaris, "Meta-Q: A Knowledge-Driven Framework for Context-Aware Activity," *Pervasive Mobile Comput.*, vol. 25, Jan. 2016, pp. 104–124.

[41] S. Oh et al., *Instruction for VIRAT Video Dataset Release 2.0*, KITWARE, Sept. 30, 2011, Accessed Aug. 8, 2016. https://data.kitware.com/#collection/56f56db28d777f753209ba9f/folder/56f581c78d777f753209c9c2

[42] "Recognition Combining SPARQL and OWL2 Activity Patterns," *Pervasive Mob. Comput.*, vol. 25, Jan. 2016, pp. 104–124.

[43] G. Baryannis, P. Woznowski, and G. Antoniou, "Rule-Based Real-Time ADL Recognition in a Smart Home Environment," *Rule Technol., Res., Tools, Applicat., Int. Web Rule Symp.*, June 28, 2016, pp. 325–340.

[44] Y. Yildirim, A. Yazici, and T. Yilmaz, "Automatic Semantic Content Extraction in Videos Using a Fuzzy Ontology and Rule-Based Model," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, Jan. 2013, pp. 47–61.

[45] U. Akdemir, P. Turaga, and R. Chellappa, "An Ontology based Approach for Activity Recognition from Video," *ACM Int. Conf. Multimedia*, Vancouver, Canada, Oct. 27–Nov. 1, 2008, pp. 709–712.

[46] M. Bertini, A.D. Bimbo, and G. Serra, "Learning Ontology Rules for Semantic Video Annotation," *ACM Int. Conf. Multimed., Workshop Multimed. Semantics*, Vancouver, Canada, Oct. 26–31, 2008

[47] L. Ballan et al., "Video Annotation and Retrieval Using Ontologies and Rule Learning," *IEEE Trans. Multimedia*, vol. 17, no. 4, Oct. 2010, pp. 80–88.

[48] L. Ballan et al., "Event Detection and Recognition for Semantic Annotation of Video," *Multimed. Tools. Appl.*, vol. 51, no. 1, Jan. 2011, pp. 279–302.

[49] G. Antoniou and F.V. Harmelen, "Web Ontology Language: OWL," in *Handbook on Ontologies*, Heidelberg: Springer, 2004, pp. 67–92.

[50] W3C Std., *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, May 2004.

[51] J. Moon et al., "ActionNet-VE Dataset: A Dataset for Describing Visual Events by b ng VIRAT Ground 2.0," *Conf. Sign. Pro. Image Proc. Pattern Recogn*., Nov. 2015, pp. 1–4.

[52] S. Oh et al., "A Large-Scale Benchmark Dataset for Event Recognition in Surveillance Video," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Colorado, USA, June 20–25, 2011, pp. 3153–3160.

[53] X. Wang and Q. Ji, "Hierarchical Context Modeling for Video Event Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, Epub, Oct. 2016.

[54] O. Russakovsky and J. Deng, *ImageNet Large Scale Visual Recognition Challenge 2016 (ILSVRC2016)*, Accessed Feb. 1, 2017. http://image-net.org/challenges/LSVRC/2016/

Jinyoung Moon received her BS degree in computer engineering from Kyungpook National University, Daegu, Rep. of Korea, in 2000. She received her MS degree in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 2002. Since 2002, she has been working in the SW & Contents Research Laboratory at ETRI, Daejeon, Rep. of Korea. She is also a Ph.D. candidate in the Department of Industrial and Systems Engineering at the Korea Advanced Institute of Science and Technology. Her research interests include behavior understanding, human-computer interaction, and machine learning.

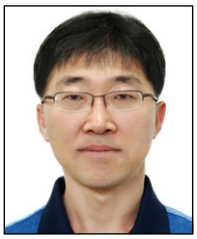
https://doi.org/10.4218/etrij.17.0116.0054

Junho Jin received his BS and MS degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 2012 and 2014, respectively. Since 2015, he has been working as a researcher at ETRI, Daejeon, Rep. of Korea. His research interests include programming languages, static analysis, vision-based deep learning and embedded software and programming.

Yongjin Kwon received his BS degree in computer science and engineering from Pohang University of Science and Technology, Rep. of Korea, in 2009, and his MS degree in computer science and engineering from Seoul National University, Rep. of Korea in 2012. Since 2012, he has been working with the SW & Contents Research Laboratory at ETRI, Daejeon, Rep. of Korea. His research interests include database systems, information retrieval, and machine learning.

Kyuchang Kang received his BS and MS degrees in electronic engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1994 and 1997 respectively. He also received his PhD in computer engineering from Chungnam National University, Daejeon, Rep. of Korea, in 2008. From 1997 to 2000, he was a researcher at the Agency for Defense Development, Daejeon, Rep. of Korea. From 2001 to 2016, he was a research engineering staff member of the Contents & Software Laboratory at ETRI, Daejeon, Rep. of Korea. Since 2017, he has been an assistant professor of the School of IT Information and Control Engineering in Kunsan National University, Rep. of Korea. His research interests include information analysis, machine learning, brain-cognitive engineering, and intelligent embedded devices.

Jongyoul Park received his BS degree in computer engineering from Chungnam National University, Daejeon, Rep. of Korea, in 1996, and his MS degree and PhD in information and communication engineering from the Gwangju Institute of Science and Technology, Rep. of Korea, in 1999 and 2004, respectively. From 2001 to 2002, he was a visiting researcher at the School of Computing, University of Utah, UT, USA. Since 2004, he has been working in the SW & Contents Research Laboratory at ETRI, Daejeon, Rep. of Korea. His research interests include visual intelligence, machine learning, and behavior and scene understanding.



Kyoung Park received his MS degree in computer engineering from Chonbuk National University, Jeonju, Rep. of Korea and PhD from Korea University, Seoul, Rep. of Korea, in 1993 and 2008, respectively. He joined ETRI, Daejeon, Seoul, Rep. of Korea, in 1993, and he has worked in high-performance computing systems, data-intensive computing, and massive scale machine learning for 24 years. Currently, he is working with SK Hynix in Icheon, Rep. of Korea, as a leader of system software research activities.