

Journal homepage: http://iieta.org/journals/ria

Combining Deep Learning and Ontology to Reveal Video Sequences Semantics

Jemai Bornia*, Frihida Ali

Check fo

Robotics, Computing and Complex Systems, National Engineering School of Tunis University Tunis El Manar, Tunis 1000, Tunisia

Corresponding Author Email: bornia.jemai@enit.utm.tn

https://doi.org/10.18280/ria.350204	ABSTRACT				
Received: 25 February 2021 Accepted: 22 March 2021	As a semantic technology, an ontology is a formal hierarchical taxonomy of lexical terms including their syntactic and semantic relationships employed to represent the meaning				
Keywords:	conveyed by a knowledge domain. The huge increase in video data volumes creates a growing demand for efficient methods to reveal and manage their semantic content through				

ontology, deep learning, video, semantic analysis, neural network

including their syntactic and semantic relationships employed to represent the meaning conveyed by a knowledge domain. The huge increase in video data volumes creates a growing demand for efficient methods to reveal and manage their semantic content through an ontology. Therefore, the need for a system that allows efficient representation of video content is obvious. This paper proposes a method to extract ontology from video streams. It is based on Deep Learning technology to identify objects and movements in video scenes. All extracted features are stored in an OWL file. The OWL file will be used afterward to exploit video documents in many ways such as modeling, indexing, querying, and feeding ontology editors to visualize, elucidate and reason on the semantic structure.

1. INTRODUCTION

The advent of the participative Web 2.0 and the available low-cost high-speed internet connections produce a huge volume of digital video data. The special properties of digital video data comparatively to alphanumeric data constitute a serious challenge for classical DBMS. To be searched and retrieved, video streams must be indexed. Many indexing ways are used such as 1) frames encoding and file format, name, duration and, physical footprint indexing, 2) content indexing by visual, audio, and some other usual properties description. The most promoting way for search and retrieval tasks is semantic indexing based on the video story depiction through a free or a controlled vocabulary (SKOS). In this paper, we introduce a new approach using Deep Learning technology (TensorFlow) that automatically extracts semantics contained into a video stream and use them to build a formal ontology.

This paper is organized as follows: In Section 2, we summarize the related work. Then, we describe the proposed methodology in Section 3. Afterward, some experimental results are presented in Section 4. Finally, we conclude and draw future works in Section 5.

2. RELATED WORK

Video stream indexing is an old research field. It has to deal with the main issues related to unconventional and unstructured data that challenge multimedia and information retrieval systems i.e. storage, search, and retrieval. Revealing and representing the semantics contained in video documents is the most promoting indexing method to fully and seamlessly exploit them. Ontology as a semantic technology helps in settling the video story semantics into a formally validated concepts/relationships graph representation. This formal representation may be used either by humans or by machines in query and reasoning tasks. In the following, we will describe the main technologies we used to develop our approach to automatically extract semantics (objects and movements) from video streams and organize them into a formal ontology.

2.1 General presentation of ontology

An ontology is a set of concepts and relationships between these concepts, it can include constraints (rules on their definitions allow deductions to be made). Each term of an ontology is clearly defined to ensure the meaning associated with it. Gruber proposed the most commonly accepted definition: "An ontology is an explicit and formal specification of a conceptualization of a domain of knowledge" [1].

The analysis of the knowledge domain conducts to a conceptualization that represents the abstraction of the world of this specific domain using a formal language such as first-order logic, descriptive logic, semantic network, etc. Explicit means that the model in question must be described unambiguously.

According to Ref. [2], an ontology is a five elements set.

$$O:=\{C, R, HC, rel, AO\}$$

where, *C* is a set of concepts specific to a knowledge domain; *R* is a set of relationships forged between concepts; *HC* is taxonomy delineating the *is-a* relations among concepts; *rel:* R->CxC is function that defines the relations on R and AO is the set of axioms asserted in a logical language.

In information systems, ontologies have been introduced to play the role of knowledge models, to provide definitions and accurate descriptions. They have been successful in different areas (semantic web, engineering, e-commerce, multimedia, artificial intelligence, library science, etc.). Hence, the amount of ontological data manipulated and generated has significantly increased. Three main factors can explain this increase: (i) the development of domain ontologies, (ii) the availability of software tools for building, editing, and deploying ontologies, and (iii) the existence of specific ontology model formalisms (OWL, PLIB, etc.) which have significantly contributed to the emergence of ontology-based applications [1].

Ontologies targeted video streams to achieve analysis tasks. In the following section, we will describe some experiments that used ontology as a tool to analyze video data.

2.2 Approaches to video analysis using ontology

This section summarizes some approaches using ontology as a tool for the semantic analysis of videos.

2.2.1 Ontology for Behavior detection in a surveillance video

The authors of this work proposed to use a generic ontology to reduce the semantic gap between linguistic concepts and visual concepts in a surveillance video [3]. The ontology is employed to assist in the detection of events and behaviors in a surveillance video. The authors used Protégé as an editor to define their ontology.

The ontology entities are divided into three groups [3]: content, system, and context.

-The contents are the events and the objects. The events are divided into personal events (standing, running, sitting, walking) and mobile events. The physical objects are moving objects and fixed objects.

- The system contains information used by the video surveillance system like reactions, areas.

- The context is or information concerned example weather (rain, sun, wind) or location (school bank, parking).

The properties are two groups: object properties and datatype properties.

2.2.2 Video Movement Ontology

The VMO [4] (Video Movement Ontology) allows annotating semantically the movements in a video document. VMO is defined using the Ontology Web Language (OWL). In fact, the concepts and their relations of this ontology are collected using the Benesh Movement Notation (BMN). The latter permits to describe of different kinds of dances or other human movements.

2.3 Definition of Deep Learning (D.L)

The D.L technology [5] is a derivative of Machine Learning, the techniques of which have evolved over the past decades. A Deep Learning algorithm allows the machine to go from an execution state to a self-learning state. The concept itself was inspired by the functioning of the human brain. It relies on a network of artificial neurons [5]. Deep Learning is made up of hundreds of layers of artificial neural networks, each with a different weight. Information flows through these neurons [6].

Each layer is analyzed after the one preceding it. Then the system will recognize the numbers before the date, and individual letters before the words. The first layer collects the details of the information presented to it and the response that the system produces from the last layer of neurons. Once the response is retrieved, the network compares it with the correct response provided by a human. If the answer is correct, the network will save and use it in other situations. If the answer is wrong, the network registers the error and readjusts the weight of each neuron [6]. This corrective action is repeated thousands of times. The goal is to produce the right answer in any circumstance. Of course, it will be easier to understand with a concrete example. Imagine a photo of a cat, if the system recognizes it, it simply records this success. If he identifies it as a dog, it will have to revisit the load exerted on each neuron. This will repeat itself with any error until the system can recognize a cat regardless of its color and the angle from which it was shot [7].

Deep learning technology is well used to process video documents; it is utilized in many research experiments such as The Deep Spatio-Temporal Fully Convolutional Networks (DST-FCN) approach [7]. This approach models the spatiotemporal dependency for the semantic segmentation of a video sequence. DST-FCN exploits spatial and temporal dependencies with 2D fully convolutional deep networks (FCN2D) [7] at the pixel level and 3D fully convolutional deep networks (FCN3D) at the voxel (3D-pixel) level. They are used to exploit convolutional long-term memory (ConvLSTM).

2.4 Analysis and research of information in a video document

Video processing and analysis are a necessity for many users who manage audio or video information in several domains. Many works have focused their efforts on technological aspects such as automatic analysis and extraction of audiovisual information.

At the content analysis level of a video, we can identify two levels of descriptions that are related to the video data [8]:

2.4.1 The signal level or low-level

This level is close to the digital representation of documents, it is concerned with describing the physical characteristics of the segments of a video such as color, texture, and shape. The information corresponding to this level is generally of numerical type, for example, tables of numbers, color histograms [8].

2.4.2 The semantic or high-level level

The semantic or high-level level: This is close to the way humans represent the content of documents, it allows describing conveyed concepts and relationships. The information linked to this level is generally of a symbolic type such as concepts, relations, and graphs. Some applications require both a signal and a symbolic description at the same time, so a semantic gap problem arises [8].

In the context of indexing [9] and video search by content, the query is often performed at the semantic level. Several methods have been proposed to reduce this semantic gap by integrating information describing the document structure.

2.5 Discussion

As the volume of videos produced today is very large, it is very difficult to manually and efficiently process the generated data. The major problem of existing systems for video content analysis is to exploit low-level content. Therefore, they usually work only on one track of the media present i.e. the image or the audio. They do not exploit the full semantic richness that is carried by the video different tracks. Whereas intelligent video analytics must be efficient and accurate, such as tracking moving objects, identifying anomalies, and generating alarms in the case of a video surveillance system.

In the next sections, we will describe our approach that

combines Deep Learning and ontology technologies to automatically unveil and process semantics carried by video documents.

3. PROPOSED METHODOLOGY

Our contribution is to combine Deep Learning and ontology for semantic video analysis. Specifically, we will create our ontological approach using the results collected from the Deep Learning processes. Figure 1 shows a global illustration of the approach. The first low level of the pyramid is related to image extraction, keyframes and object detection contained in keyframes. These are low-level class functionalities. The lowlevel functionalities are generic and can be reused in different applications. The intermediate level uses low-level results to detect movements and actions performed by objects. The goal of the semantic level is to map the mid-level results into ontological rules to enable semantic video analysis.



Figure 1. General diagram of the proposed method



Figure 2. Ontology rules extraction process overview

The main objective of the proposed approach is to design a technique to extract ontological rules from a video stream using Deep Learning technology. The proposed method starts by reducing the video to a number of frames called key frames. The next step is to detect the objects in each frame (moving objects or still objects). In this step the image names are assigned to the annotation module, if the file exists in this module a text file is assigned to each image. Then the actions and movements are followed between each two successive keyframes. As a result of the previous steps, an XML file containing information about the detected objects is automatically generated. As a last step, the ontology extraction will take place using the XML file.

To validate the consistency of the extracted ontology, a descriptive language reasoner like Fat++ or HermiT is used. Then we will test our ontology using a query language like SPARQL or DL Query.

The next figure (Figure 2) shows the first proof of the proposed approach. It starts by the segmentation of the video stream into still images and ends up by the generation of an ontology describing its semantics.

4. EXPERIMENTAL RESULTS

The proposed approach allows us to test any video and with an unlimited number, but we have tested here only six videos, with different sizes (4.38MB, 6.18MB, 15.4KB, 18MB, 13MB, 13.9MB). These videos contain fixed and moving objects, animals, and people.

Table 1 illustrates the tested video sizes, the number of images extracted, and the number of keyframes.

Table 1. Tested video's information

Video	Size	Number of pictures	Number of key frames
1	4.38 MO	100	2
2	6.18 MO	100	2
3	15.4 KO	109	20
4	18 MO	120	3
5	13 MO	105	3
6	13.9 MO	105	3

4.1 Technological choices

To execute the approach we used several tools and technologies we mainly use OpenCV-Python open-source computer vision library. This library can be used on Linux, Windows, and Mac OS X. Interfaces have been developed for Python, Ruby, Matlab, and other languages [10]. Additionally, we used Darknet, (an open-source Neural networks written in C) [11] and Pandas (a Python open-source library allowing the manipulation and analysis of data). The last offers data structures and functions for tables and time series manipulation [12]. Finally, we used Matplotlib (a Python open-source library) to plot and visualize data in the form of graphs. Matplotlib can also be combined with the two open-source Python scientific computing libraries NumPy and SciPy.

4.2 Results of experiments

4.2.1 Preparation for ontology extraction

The diagram (Figure 3) below presents the first phase of the algorithm. which consists of extracting the objects and their movements in a video sequence and the result of this phase will be written to an XML file which will be used in the second phase to encoding the ontology into an OWL file format, the standard Semantic Web technology based on descriptive logic [13].



Figure 3. Process of extracting objects and movement

To facilitate the execution of the approach we have created a graphical interface, which contains as a menu the stages of the proposed algorithm, and to start you have to identify yourself by giving the login and password.

<u>Video playback and segmentation</u>. To segment the video the process as follow in Figure 4 is very simple; we start by loading the video then authenticate and choose the "video segmentation" button The segmentation of the videos.

In this step gives a high number of images, which makes the analysis of the video sequences a bit difficult. Consequently, we must use the key frames that will be introduced in step 2 (next section).



Figure 4. Diagram of segmentation video

An example of the video we have tested is of size 13.9 MB, and lasts 20 seconds, we have obtained as segmentation results 99 images.

Key frame detection. This step minimizes the number of images extracted as a video contains many images so; we just take the images that contain more information i.e. the key frames. These images summarize accurately and compactly the video content.

A key frame is a representative image that includes all facts in the video collection. It is used for indexing, rating, and retrieving the video. Popular algorithms generate relevant key frames but with some redundancy. Consequently, we developed the approach to extract the key frames, which produces a less redundant key frames set (Figure 5).

To detect key frames the open CV is used to record the detected images (cv2). This method eliminates the redundant and inessential images in a video without affecting the visual content of semantic details.



Figure 5. The output of key frame

<u>Object detection</u>. Detecting the objects on the key frames is a very important step in tracing the movement of these objects. The objects are found, and we can calculate the percentage of appearance during the video sequence. In this sequence, we will present portions of the code and some images of the outputs.

The object detection phase identifies each object that is in the key frames. An XML file will be also generated automatically that describes the content of each key frame.

The following figure (Figure 6) shows the flowchart for detecting objects and saving them in an XML file.



Figure 6. Diagram of object detection

When the objects are detected, each one is recorded with its name and it is boxed on key frame as the following example (Figure 7) and (Figure 8).



Figure 7. Output example1: objects are framed in the key frame



Figure 8. Output example2: objects are framed in the key frame

object:==>person	object:==>horse
Accuracy==>9.09%	Accuracy==>7.79%
object:==>cumputer	
Accuracy==>7.79%	object:==>tree
object:==>mouse	Accuracy==>6.49%
Accuracy==>5.19%	
	object:==>train
object:==>mobile	100000000000000000000000000000000000000
Accuracy=>9.09%	Accuracy>10.39*
object:==>cup	object:==>cars
Accuracy==>7.79%	Accuracy==>9.09%
object:==>dog	
Accuracy==>11.69%	object:==>bus
object:==>house	Accuracy==>3.9%
Accuracy==>3.9%	
object:==>bicycle	
Accuracy==>3.9%	

Figure 9. Object detection of all tested videos

The proposed approach detects the objects of each video tested, and at the end, it displays all objects of all the videos under test with the appearance percentage too, this is demonstrated in the previous figure (Figure 9).

<u>Generation of matrices.</u> This step allows you to find the positions of the objects (X_max, X_minY_max, Y_min, Z_max, Z_min). The matrix contains the identity of the object, the identity of the key frame and the identity of the video, as well as the positioning of the pixels of each object or person.

<u>Movement tracking.</u> In this case, the detection was carried out by the algorithm that we developed which uses a DNN (Deep Learning).

According to the generated matrix, the graph of movement of each detected object is drawn, then a portion of code that makes it possible to detect the movement and two examples of the result.

According to the object id, the motion graph is drawn as it is in the following Figures 10 and 11.

The following two graphs show the movement of the object over a series of frames.

The two previous graphs show the movements of objects and people during a video sequence. In both figures, we have followed the movement of a person and a dog that appears in the video, a graph is obtained for each one, (the number of images as a function of time).



Figure 10. Graph of the movement of a person in a tested video



Figure 11. Graph of the movement of a dog in a tested video

4.2.2 Ontology extraction

In process 1(Preparation for ontology extraction), we prepared the train to extract the ontology that presents the semantic content of a video. We suggested resorting to Deep Learning to prepare for the construction of the ontology. What we need are the names of the objects, their types; mobile or fixed like a tree for example. However, the question is, how are we going to migrate from process 1 to process 2. Here, we have two methods: In the former, we developed an algorithm that automatically generates the owl file and, in the latter, we use the editor Protégé to load, validate and visualize the extracted ontology.



Figure 12. Process of ontology visualization

Running the algorithm defaults to an OWL file that describes all the objects in the video sequence as the result. A specific OWL file for each video and an OWL file for all the videos tested and, in our case, we tested six videos. Next, we will present some results of the generated OWL files and the ontologies that we extracted by Protégé.

<u>General presentation of Protégé</u> Protégé is a graphical ontology development environment developed by Stanford SMI [14]. It is an editor that allows to build an ontology for a given domain, to define data entry forms, and to acquire data using these forms in the form of instances of this ontology. Protégé is also a Java library that can be extended to create true knowledge base applications using an inference engine to reason and deduce new facts by applying inference rules to ontology instances and to the ontology itself (meta-reasoning).

In the context of the semantic web, plugins for RDF, DAML+OIL and OWL have been developed for Protégé. These plugins allow to use Protégé as an ontology editor for these different languages, to create instances and to save them in the respective formats.

Today, it brings together a large community of users and benefits from the very latest advances in ontological research (Figure 12): benchmark OWL compatibility, inferential services, knowledge base management, ontology visualization, alignment, and fusion, etc.

<u>Ontology generated from XML file.</u> Objects are defined as classes and object movements as properties of the ontology.

• The person is a class

• The dog is a class

• The tree is a class

People and dogs are defined as moving objects while trees are defined as fixed objects.

In the figure (Figure 13) below, the objects which are in test video one are displayed (animal, person, and trees) and the types of movements (walking, turning right, turning left, etc.).

To visualize the ontology embedded into the owl file, OntoGraf plugin is used.

The plugin OntoGraf supports interactive navigation in OWL ontology relationships. Different arrangements are supported to automatically organize the ontology structure. Different relationships are supported: subclass, individual, domain object properties/range and equivalence. Relationships and node types can be filtered to help you create the desired view.

In the following figure (Figure 14) we have the interface for defining the properties of the ontology layers, let's take for example jump, have_the_same_position, turn_lefet, and turn_right. Example the dog can take one of the three properties, jump, turn_lefet, and turn_right because it is set as a mobile object and the same for any mobile object in the ontology.



Figure 13. View of ontology loaded into Protégé using OntoGraf plugin



Figure 14. Objects properties definitions

The two following figures (Figure 15, Figure 16) show two examples of ontologies visualized with the OWLViz plugin of the Protégé. In this ontology, there is a thing class which is divided into two classes Fixed_objetc and Mobile_object which are also composed of subclasses such as animals, cars person...

The plug in OWLVizis conceived to be operated with the Protege -OWL editor. It allows incremental displation and navigation in the class hierarchies of an OWL ontology., which allows to compare the asserted class hierarchy and the inferred class hierarchy. OWLViz integrates with the Protege-OWL editor, using the same set of colors so that primitive and defined classes can be distinguished.



Figure 15. View of ontology loaded into Protégé using OWL-Viz plugin



Figure 16. A part of ontology loaded into Protégé using OWL-Viz

4.2.3 Ontology validation

<u>Validation with Fact++</u>. The quality and accuracy of ontologies play an essential rolefor semantically represent a video document. An invalid ontology tells us that there are gaps and conflicts in this ontology which means that some ontology concepts cannot be interpreted correctly. Consequenly, we used the reasoner FaCT ++ (Fast Classification of Terminologies) under Protégé to validate the extracted ontology.

FaCT++ is a reasoner for expressive description logics (DL). It covers OWL and OWL 2 based DL ontology languages. It can be used as a standalone DIG reasoner, or as a back-end reasoner for the OWL API-based application. It is now used as one of the default reasoners in the OWL editor of Protege 4.

The following figure shows the result of the ontology validation presented in (Figure 17).

We notice in the previous figure (Figure 17) that the supporting fact ++ takes 71 ms to verify the validity of the ontology.

It checks the hierarchy of classes, the hierarchy of object properties and the hierarchy of data properties. This means that the ontology is consistent and there are no conflicts in the classes or properties.

<u>Testing Ontology with DL Query language.</u> The DL-Query allows to query and search for an ontology. The ontology must be classified by a reasoner before it can be queried in the DL query tab.

In the following examples, we have executed three different queries after activation of the reasoner fact++.



Figure 17. Result of validation with Fact++

For example in the figure (Figure 18), the 'Fixed_Object' query is executed and as a result, the sub classes House, Tree and Trafic_sign are executed.

Active Ontology	Entities	Classes	Object Properties	Data Properties	Annotation Properties	Individuals	OWLViz	DL Query	OntoGraf	SPARQL Query	Ontology Differences
Cless hierarchy. III 2012			DL query.								
			- Query (class expression)								
▼- ● Thing			Fixed_Object								
V-• Fixed_Object											
-•t	Otrafic_sign OTRee VOMobile_Object Omobile_Object			Execute Add to ontology							
V- Mob											
►•• A				-Query results							
- Bus - Bus - Cars - Person			Sub classes (7) house								
			trafic_sig	n							
					TRee						

Figure 18. DL-Query request for the class Fixed_Object

In the following figure (Figure 19) we look for the class 'Animals', so the sub-classes Dog, Horse and cat appeared as a result of this request.

Active Ontology Entities Classes Object Properties Data Properties	Annotation Properties Individuals OWLViz DL Query OntoGraf SPARQL Query Ontology Differences					
Class hierarchy: IIEIBIO	1 query:					
😫 🖗 🕱	- Cuery (class expression)					
▼- ● Thing ▼- ● Fixed_Object	Animals					
- ● house - ● trafic_sign - ● TRee • ● Mobile_Object ▶ ● Animals	Execute Add to ortilogy Cuery results					
	Sub classes (7) cat					
Person	• dog					
	• horse					

Figure 19. DL-Query request for the class Animals

The third query (Figure 20) looks for the subclasses 'House' and 'Cars'.



Figure 20. DL-Query request sub-class House and Cars

5. CONCLUSIONS

In this paper, we proposed an approach to extract semantics carried by video streams using a combination of Deep Learning and ontology technologies. The features extracted by Deep Learning are organized into a formal ontology that is loaded and validated by Protégé open-source editor. However, we point out two limitations of our actual proof-of-the-concept approach that we are planning to overcome in our future work: 1) Our actual extracted ontology is declared formally errorfree by the Protégé editor plug-in Fact++ reasoner. We intend to check the semantic representation accuracy of our ontology by comparing it with a manually performed video semantic indexing. 2) Our approach voluntary ignores some others video stream properties such as the visual and audio ones. In the incoming version, these properties will be extracted in parallel with objects identities, characters relationships, verbal/gestural exchanges, time and location etc. Our goal is to build an ontology that accurately captures and represents the story content and tools used to convey it of the video documents and therefore facilitates its querying and sequences retrieval.

REFERENCES

- Ding, Y., Foo, S. (2002). Ontology research and development, Part 2 - A review of ontology mapping and evolving. Journal of Information Science, 28(5): 375-388. https://doi.org/10.1177/016555150202800503
- [2] Baglioni, M., Masserotti, M.V., Renso, C., Spinsanti, L. (2011). Improving geodatabase semantic querying exlpoiting ontologies. In: Claramunt C., Levashkin S., Bertolotto M. (eds) GeoSpatial Semantics. GeoS 2011. Lecture Notes in Computer Science, vol 6631. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20630-6_2
- Kazi Tani, M.Y., Lablack, A., Ghomari, A., Bilasco, I.M. (2015). Events detection using a video-surveillance ontology and a rule-based approach. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8926. Springer, Cham. https://doi.org/10.1007/978-3-319-16181-5_21
- [4] Aad, S.S., De Beul, D., Mahmoudi, S., Manneback, P. (2012). An ontology for video human movement representation based on Benesh notation. 2012 International Conference on Multimedia Computing and

Systems, pp. https://doi.org/10.1109/ICMCS.2012.6320129 77-82

- [5] Qiu, Z., Yao, T., Mei, T. (2017). Learning deep spatiotemporal dependence for semantic video segmentation. IEEE Transactions on Multimedia, 20(4): 939-949. https://doi.org/10.1109/TMM.2017.2759504
- [6] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61: 85-117. https://doi.org/10.1016/j.neunet.2014.09.003
- [7] Bhute, A.N., Meshram, B.B. (2011). Techniques for text, image, audio and video indexing and retrieval. In International Conference on Network Infrastructure Management Systems Interface-2011, VJTI, Matunga, Mumbai, pp. 90-96.
- [8] Snoek, C.G.M., Worring, M. (2005). Multimodal video indexing: A review of the state-of-the-art. Multimedia Tools and Applications, 25(1): 5-35. https://doi.org/10.1023/B:MTAP.0000046380.27575.a5
- Zelinsky, A. (2009). Learning OpenCV---Computer Vision with the OpenCV Library (Bradski, G.R. et al.; 2008) [On the Shelf]. IEEE Robotics & Automation Magazine, 16(3): 100. https://doi.org/10.1109/MRA.2009.933612
- [10] Joseph, R. (2017). Darknet: Open-source neural networks in c. Pjreddie. com. [Online]. Available: https://pjreddie.com/darknet/, accessed on 21 Jun. 2017.
- [11] McKINNEY, Wes. (2011). Pandas: A foundational Python library for data analysis and statistics. Python for High Performance and Scientific Computing 14.9.
- [12] http://www.w3.org/TR/owl2-overvoiew/, accessed on 21 Jun. 2020.
- [13] Noy, N.F., Fergerson, R.W., Musen, M.A. (2000). The knowledge model of Protégé-2000: Combining interoperability and flexibility. In: Dieng R., Corby O. (eds) Knowledge Engineering and Knowledge Management Methods, Models, and Tools. EKAW 2000. Lecture Notes in Computer Science, vol 1937. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-39967-4_2