Visual Domain Ontology using OWL Lite for Semantic Image Processing

Ahmad Adel Abu-Shareha, Ali Alshahrani

ITC Department, Arab Open university, Riyadh, Saudi Arabia

Abstract **– In this paper, a visual domain ontology (VDO) is constructed using OWL-Lite Language. The VDO passes through two execution phases, namely, construction and inferring phases. In the construction phase, OWL classes are initialized, with reference to annotated scenes, and connected by hierarchical, spatial, and content-based relationships (presence/absence of some objects depends on other objects). In the inferring phase, the VDO is used to infer knowledge about an unknown scene. This paper aims to use a standard language, namely, OWL, to represent non-standard visual knowledge; facilitate straightforward ontology enrichment; and define the rules for inferring based on the constructed ontology. The OWL standardizes the constructed knowledge and facilitates advanced inferring because it is built on top of the first-order logic and description logic. The VDO then allows an efficient representation and reasoning of complex visual knowledge. In addition to representation, the VDO enables easy extension, sharing, and reuse of the represented visual knowledge.**

Keywords **– Semantic Image Processing, Image Knowledge Representation, Visual Domain Ontology, OWL.**

DOI: 10.18421/TEM82-08
https://dx.doi.org/10.18421/TEM82-08

Corresponding author: Ahmad Adel Abu-Shareha,
ITC Department, Arab Open university, Riyadh, Saudi Arabia
Email: a.abushareha@arabou.edu.sa

Received: 04 March 2019.
Accepted: 25 April 2019.
Published: 27 May 2019.

1. Introduction

Several image-processing applications, such as image classification, object recognition, and image disambiguation, are based on a supervised or semi-supervised learning process, which involves training and predicting phases. The training phase used training data to construct a model that usually handles low-level features and/or image labels. Predicting employs an input with unknown or ambiguous labels and uses the constructed model to predict output labels. These simple applications handle minimum visual information provided by the constructed model. The new trends in computer vision and image processing move toward semantic technique for image classification, object recognition, segmentation, and disambiguation. Implementing the complex recognition and predicting tasks of semantic techniques requires an extensive and expressive knowledge source to replace the temporary models constructed on-the-fly in the training phase. Ontology is one of the well-known sources of knowledge. It is a formal knowledge representation that handles a set of concepts within a domain, their relationships, and properties [1].

Constructing an ontology to handle visual knowledge is not a trivial task. Visual knowledge is complex and heterogeneous. For example, features may be represented as a vector of numerical values with uncertainties. Spatial information may be represented as direction in 2D space, and content information may be represented as tuple of two or more entries. Ontology for the visual domain should represent such heterogeneity and enable high-level prediction and recognition. Moreover, ontology should be general-purpose for various applications and for facilitating the links between image applications toward a semantic image processing pipeline[1].

Several existing approaches have been proposed to construct ontologies for semantic image-processing applications, such as using ontologies for semantic feature extractions [2],[3],[4], semantic image segmentations [5],[6], and semantic object recognition [7],[8]. These approaches are pioneers in using ontologies, and their constructed and utilized ontologies are task oriented but suffer from the following drawbacks: storing low-level features and their object identities only; having a weak structure; and limited ability to fit the task at hand because of the utilization of simple encoding languages and techniques. Moreover, these techniques cannot construct a consistent complex visual knowledge source.

Scholars must construct visual knowledge ontology by using standard knowledge representation languages, such as RDF and Web Ontology Language (OWL). This paper introduces visual domain ontology (VDO), a formal representation of visual domain knowledge, by using OWL Lite. The well-defined syntax of OWL represented in RDF triples and RDF graph, its formal semantics, and its sufficient expressive power can efficiently represent complex knowledge in a restricted visual domain [9]. This paper is organized as follows. Section 2 provides a brief description of the ontology. Section 3 highlights some of the previous works in the ontologization of image knowledge. Section 4 describes the structure and components of the proposed VDO, its implementation using OWL Lite, and the reasoning mechanisms that must be incorporated with the ontology. Section 5 presents the implication of the proposed work. Section 6 concludes the study.

2. Ontology

Ontology is a conceptual knowledge representation that can be interpreted by both human and computer. Ontology of a given domain should cover the abstract information for that domain, that is, **domain components**. Ontology of visual knowledge should represent visual information, which can be summarized as follows: low-level features that are directly extracted from the images and images' objects; the association of the low-level features to object identity (feature-to-object) under uncertainty values; content information that represents the presence/absence of several objects together in the scene; and objects' spatial relationships and context information represented by non-visual domain-knowledge [10]. For **ontology components**, ontology involves concepts and hierarchical relationships, which are used to represent and categorize the *domain components*. In an ontology, each category is represented by a *concept* and different categories are connected to each other through hierarchical relationships. In addition, ontology may involve additional components, such as properties and other relationships, which enrich the ontology and enable efficient knowledge representation. Concepts are connected to each other via hierarchical and non-hierarchical relationships to create the **ontology structure**, which allows the ontology to be represented broadly by general-purpose data structures of graphs and trees. The stated relationships between the underlying ontology concepts are guaranteed through the relationships (represented by edges) among these concepts. Moreover, these edges infer relationships in the so-called **reasoning**. The ontology specification can be represented using general-purpose programming languages or tools; this representation is denoted as general **representation** of ontology. However, this representation cannot construct complex knowledge because the ontology grows; this phenomenon leads to high chance for inconsistency and connectivity loss because of the lack of rules to control the construction. Figure 1. shows a simple visual ontology that represents visual information by using a general-purpose graph. "Outdoor" is a concept connected through a hierarchy relationship to "Everything" in high levels and "Animal" in low levels. Non-hierarchy relationship "on-top" connects concepts to one another. Properties such as "has-Texture" and "has-Color" associate property values with a specific concept.

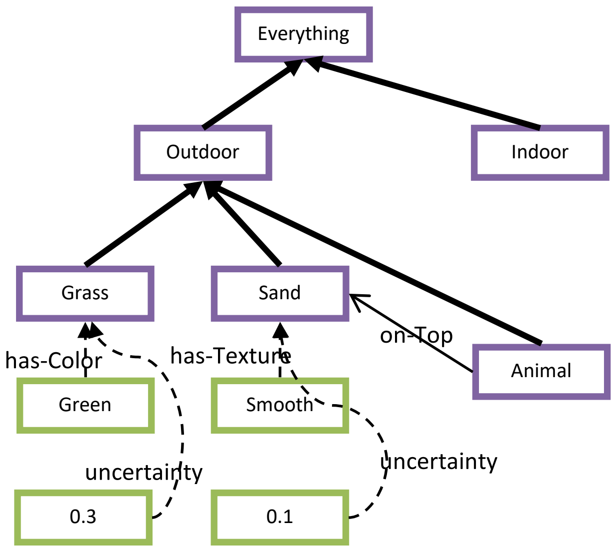

Figure 1. Graph Representation of Visual Knowledge Ontology

Advances in ontology representation have been associated with the semantic web and the emergence of standard ontology languages, such as RDF and OWL. OWL prevents inconsistency and looseness and is considered an inference technique because it is built on top of the first-order and description logic. Ontology concepts are defined using such languages, besides existing hierarchical relationships and properties, as ontology **classes**; actual objects, which belong to a specific ontology class, are represented as **individuals**. The OWL syntax is represented in RDF triples. The triple structure is *{subject – relationship*

– object}. Figure 2. shows an example triple. However, visual knowledge requires a tuple of more than three elements. For example, a five-tuple representation is required, structured as *{Object – hasFeature – Feature – hasUncertainty – Uncertainty}*, to encode low-level features with associated uncertainty. A triple can encode the first three elements by using RDF triples to represent five tuple, and the last two elements should be represented in another triple and connected to one of the elements of the first triple (subject, object or relation). This construction is not trivial and a real problem in visual knowledge ontologization.

```
<owl:Class rdf:ID="Animal">
        <rdfs:subClassOf>
                <owl:Class rdf:ID="Outdoor"/>
        </rdfs:subClassOf>
</owl:Class>
```

Means:

Animal ***sub-Class-Of*** ***Outdoor***

Figure 2. Example of RDF triples

To dwell on this issue, scholars represent visual knowledge as: {*Grass – hasColor – Green – hasUncertaintyValue – 0.7*}, {*Apple – hasColor – Green – hasUncertaintyValue – 0.6*}, {*Apple – hasColor – Red – hasUncertaintyValue – 0.4*}. These relationships are represented in five tuples and should be divided into triples to be represented using OWL. The first triples out of each tuple can be constructed easily as follows: {*Grass – hasColor – Green*}, {*Apple – hasColor – Green*}, {*Apple – hasColor – Red*}. The second set of triple contains two elements, and the third element should intuitively be used as the connector between the first and second set of triples. If the second triple is associated with the subject of the first triple set, then a disassociation problem arises. For example, if the following two triples exist in a single ontology {*Grass – hasColor – Green*}, {*Apple – hasColor – Green*}, then adding uncertainty values to the object (Green) in a new triple will then lead to the following two triples {*Green – value – 0.7*} and {*Green – value – 0.6*}. The four triples will lead to the situation where the association between the objects (Apple, Grass) and their green color uncertainty values has been lost. In this example (Green), each concept in the ontology will have multiple relationships with other concepts. Adding the uncertainty value to the objects of the triples (Apple, Grass) will not solve the problem because these concepts might have relationships to other elements in the ontology. For example, in the triples {*Apple – hasColor – Green*}, {*Apple – hasColor – Red*}, {*Grass – hasColor – Green*}, the uncertainty triples associated with the (Apple, Grass) are {*Apple – value – 0.6*} and {*Grass – value – 0.7*}. These five triples will also lead to disassociation. The same convention applies if the uncertainty triples are associated with the relationships (e.g. "hasColor").

The disassociation problem can be tuned if the actual represented objects (Apple, Green, and so on) are encoded as classes and not as individuals and by using constructors, such as "intersectionOf" and 'UnionOf' with these classes. The demonstration of this problem is shown in Figure 1. The objects in the figure cannot be represented as classes connected via hierarchical sub-super class relationships because the sub-super relationships carry out the classification task; however, in the visual domain, the hierarchy conducts the encapsulation task. The problem with such representation will arise when the visual ontology is integrated into the domain knowledge that intuitively will represent the objects as individuals according to the definition of ontology. This representation spoils the inferring process [11].

The utilization of triples as the basics of the ontology languages renders the construction of complex visual knowledge as non-trivial. Such difficulties have forced previous ontologization approaches to limit the represented knowledge to one feature to avoid the ***association problem***. The association problem occurs when multiple features are used in the same ontology. Another problem, ***encoding problem***, appears when classes are used in place of individuals in modern ontology languages.

3. Related Work

Visual ontologization approach can be evaluated based on the following five criteria to solve the association and encoding problems: included **visual components**, utilized **ontology components**, **ontology structure**, utilized **ontology representation**, and utilized **ontology reasoning**. The **visual components** are the forms of visual knowledge that are represented by the ontology; these components include low-level features, object identities, content, and context information. The **ontology structure** enables the connectivity between the visual components and can be simple or advanced. The **ontology structure** has a major rule in the reasoning process. The included **ontology components** play a major rule in the quality and amount of the represented visual components. These components might be limited to classes and hierarchical relationships or might be extended to involve many components. The **ontology representation** can be general with or without restrictions using formal representation, which exhibits the advantages of preventing knowledge conflicts, allowing complex knowledge representation, and supporting reasoning. **Ontology reasoning** is used to predict knowledge from stated facts. These criteria affect the efficiency of the constructed ontology in two ways: the richness of visual knowledge and the capability to implement efficient reasoning.

Existing visual knowledge ontology can be categorized into two groups: general and task-oriented. These approaches are discussed with emphasis on ontology-discriminating criteria and capability to solve common problems. In the task-oriented approach, Liu has developed a visual ontology for bird recognition, in which shape feature is used to represent the birds' visual information [12]. The **visual components** of the developed ontology refer to the shape feature along with the birds' identities. **The ontology components** of the developed ontology are classes that represent features, identities, and hierarchical relationships that connect the classes at various levels. **The ontology structure** is built by connecting the shape features encoded in classes with higher classes that represent the birds' identities. **The representation** used to encode the ontology is a general-purpose graph. **Reasoning** makes use of the constructed hierarchical relationships. This ontology does not come across the association and encoding problem. Overall, the constructed ontology represents only a single low-level feature, single object, and specific dataset for a given task [13]. Khan and Wang [14] and Torresani et al. [15] developed and used similar ontologies, which all fall under the task-oriented category and represent features, object identities, and specific dataset for a task.

Penta et al. [11] constructed refined visual ontology based on a set of **visual components**, namely, spatial relationship, uncertainty of the concepts, association of low-level features, and object identity. **The ontology components** of the developed ontology are classes that represent both features and identities and the hierarchical relationships that connect classes at various levels. **The ontology structure** is built by connecting features that are encoded in classes with object identities that are represented by high-level classes. The hierarchical structure of the developed ontology is rich to some extent and facilitates **reasoning**. **The representation** used to encode the ontology is a general-purpose graph. This ontology does not come across the association and encoding problem. A similar approach was developed by Iakovidis et al. [16] and Minu et al. [17]

Overall, task-oriented leaks standard representation (e.g., does not use standard representation languages) and does not enable general-purpose semantic-based processing and knowledge enrichment by which the knowledge can be realistic and rational. Moreover, encoding heterogeneous visual knowledge is difficult using graphs. Such ontologies have been used for specific applications.

The general-purpose approach aims to construct an ontology that can be applied with a wide range of semantic-based techniques. A major contribution in this field is the ontologization of MPEG-7 [18]. MPEG-7 is an XML-based standard for describing the multimedia content. As an XML-based standard, MPEG-7 can establish a well-defined standard format for human understanding and machine manageability. Ontologization of MPEG-7 has produced a rich source for describing multimedia contents. The **visual components** included are the spatial relationships, uncertainty of the concepts, association of the low-level features, and object identities. The **ontology components** used are classes that represent the features, objects, and relationships which create the **ontology structure**. The disadvantages of using MPEG-7 ontologies are the inability to include domain-specific knowledge and enable knowledge enrichment as XML does not facilitate **reasoning**.

Multimedia OWL (M-OWL), a formalism ontology language that supports the description of media contents based on MPEG-7, was developed. For the visual components, M-OWL considers the following different layers of knowledge abstraction in the multimedia data: the observable features (local and global) which are represented by MPEG-7 and the abstract knowledge (concept) which is represented by OWL language. Moreover, M-OWL considers Feature-to-Concept association, wherein features are associated with the corresponding concepts. Concept-to-Concept probability based on Bayesian theory has been also addressed [19],[20]. M-OWL sets a solid base for ontologizing the multimedia but exhibits the following disadvantages: low-level features are embedded in MPEG-7, which require a special mechanism that deals with MPEG-7 structure; and the actual visual knowledge is encoded in the MPEG-7 and not in OWL, which gives a secondary role to OWL itself. Thus, M-OWL is bounded by the limited reasoning support of MPEG-7 that is equal to the reasoning supported by a graph [21].

In contrast to task-oriented ontologies, general visual ontologies enable general-purpose semantic-based processing but do not fully enable knowledge enrichment and knowledge reasoning to make the knowledge realistic and rational. This limitation could be due to the fact that these ontologies do not use standard languages, such as RDF and OWL, which enable full reasoning capabilities. Table 1. shows the comparison between general and task-oriented approaches for ontologizing of visual knowledge. In conclusion, scholars must develop an ontology that optimizes the representation of the visual knowledge, allows efficient reasoning and enrichment, solves the association and encoding problems, and maximizes the usability.

Table 1. Comparison between task-oriented and general purpose approaches for ontologizing the visual knowledge

	Task Oriented	General Purpose
Visual Components	Low-level features and object identities. Uncertainty relationships in few cases.	Low-level features, object identities spatial relationships and content relationships.
Ontology Structure	Hierarchical with mostly two levels (features and objects).	Hierarchical with mostly two levels (features and objects)
Ontology Components	Concepts (classes) and hierarchical relationships.	Concepts (classes), hierarchical relationships and spatial relationships.
Ontology Representations	Graph-based.	Standard ontology representation languages.
Ontology Reasoning	Non-standard (hierarchical only with uncertainty in few cases.	Limited standard (hierarchical with uncertainty).

4. Proposed Work

This work aims to construct general-purpose ontology for solving the association and encoding problems while considering the five criteria discussed earlier. This paper proposes VDO, which is represented using OWL Lite. The considerations and goals of the VDO are listed as follows:

- To ease the problem of visual recognition by dividing this problem into sub-problems. Subsequently, multiple-scale encoding of the visual information will be used and connected via a hierarchy as equivalent to the region merging mechanism used in image segmentation [22].
- To control predicting by uncertainty values, which makes the predicting and the analysis process more accurate with an element of risk.
- To solve the disassociation problem and to avoid the encoding problem discussed earlier.

The components of the constructed VDO are:

- **Metadata**: identifies the ontology components (classes, relationships, and properties) and creates the structure of the underlying ontology.
- **Visual knowledge:** the facts in the visual domain, obtaining, and encoding using the pre-defined metadata.
- **Enrichment procedure**: the process of integrating the domain knowledge to the visual knowledge at the individual's level
- **Rules:** the reasoning mechanism.

4.1. VDO Metadata

The metadata for the VDO are built using the following OWL-Lite ontology components: classes, individuals, relationship, properties, constructors, and axioms. The core elements of the metadata are made up of classes. The utilized classes are of *scale variety*, which are ordered from low-scale to high scale, as follows: *Feature*, *Blob*, *Object*, *Region*, *Scene*, and *Domain*. A VDO class of low-order indicates single real-object or part of an object in a visual domain, whereas a class in the high-order refers to a region or a scene which contains a group of objects that share the characteristic of being in the same spatial space (e.g. complete scene). These classes are represented in **OWL-Lite** by using the *owl:class* tag. Only *Feature* class is associated with sub-classes; each represents a single feature type, such as shape, texture, and so on. The *rdfs:subClassOf* tag connects a sub-class to its super-class.

Table 2. presents the metadata class components of VDO. Individuals are used to represent the actual objects that belong to a class. Figure 3. shows an example of the VDO classes in OWL-Lite with enriched facts represented in individuals.

```
<owl:Class rdf:ID="Object"/>
<owl:Class rdf:ID="Shape">
      <rdfs:subClassOf>                          Metadat
            <owl:Class rdf:ID="Feature"/>
      </rdfs:subClassOf>
 </owl:Class>
<Object rdf:ID="Apple">                          Fact
```

Figure 3. Example of Object and Feature Classes with Enriched Facts Represent using an "Object" Individual

Table 2. The class elements of the VDO Metadata

Name/ Description	Usability	OWL Element
Feature/ low-level features	It is used to describe blobs, objects, regions and scenes. Examples: Shape, Color.. etc.	***owl:class rdf:ID="Feature"***
Blob/ part of object.	It is used to describe another blob or an object. However, several blobs may combine together to represent another blob. Blobs are mainly described by low level features.	***owl:class rdf:ID=" blob "***
Object/ real-object	Represents a real object. Multiple objects represent a scene and can be located in a specific region. Object is described by low-level features or blobs, or both.	***owl:class rdf:ID="Object"***
Region/ meaningful region in the space.	Region in the space, several regions combined to represent a scene. Region is described by low-level global features, and objects.	***owl:class rdf:ID="Region"***
Scene	Scene is described by low-level Global features, and items.	***owl:class rdf:ID="Scene"***
Domain/ domain of interest.	Domain is described using one or more scene.	***owl:class rdf:ID="Domain"***

The *scale variety classes* are connected to each other to create a hierarchy of features-blobs-objects-scenes, which allows smooth prediction and recognition by fact flooding. Transitive relationships with the name *Located-in* have been identified to create a hierarchical structure in the VDO. A relationship *Contain* is created as an inverse to the relationship *Located-in*. The relationship *Contain* allows propagation in the inverse direction. Intuitively, the inverse of the transitive is also transitive. The relationship *Located-in* is associated with several sub-relations; each relation connects two classes of different scales. These sub-relationships are as follows: *Feature-Located-in* relationship connects feature to higher-scale classes, more specifically blobs. A *blob-Located-in relationship* connects blobs to objects and other high-order classes. *Object-Located-in* relationship connects object to regions and scene. The sub-relationships are identified in OWL using *rdfs:subPropertyOf* tag. The inverse relationship of *Located-in*, *Contain* relationship is also associated with sub-relationships inverse to the sub-relationships of *Located in*, which are as follows: *Contain-Feature*, *Contain-Blob*, and *Contain-Object*. Using OWL, the hierarchical relationships are identified using *owl:ObjectProperty* tag and the inverse relationships are identified using *owl:inverseOf tag*. These relationships are summarized in Table 3. Figure 4. shows an example of the VDO classes and their hierarchical relationships in OWL-Lite with enriched facts.

The content relations connect individuals belonging to classes of scales identical (seldom of different scales) to each other. In the constructed VDO, the content relationships are captured by the spatial relations, which encode the matrix directions of left, right, bottom, and top. The relationship *Adjacent-to* is identified in VDO with four sub-relations that represent four directions, and their inverse relationships are also identified and used in VDO, as follows: *Top-of* and its inverse *Bottom-of*, the *Left-of* relationship and its inverse *Right-of*. In OWL these relationships are identified using *owl:ObjectProperty* tag. The sub-relationships are identified in OWL using *rdfs:subPropertyOf* tag. The inverse relationships are identified in OWL using *owl:inverseOf* tag. These relationships are summarized in Table 4.

```
<owl:ObjectProperty rdf:ID="Contain-Feature">
    <rdfs:subPropertyOf>
        <owl:TransitiveProperty rdf:about="#Contain"/>
    </rdfs:subPropertyOf>
    <rdfs:range rdf:resource="#Feature"/>
    <owl:inverseOf>
        <owl:ObjectProperty rdf:ID="Feature-Located-in"/>
    </owl:inverseOf>
</owl:ObjectProperty>
<Object. rdf:ID="Apple">
    <Contain-Feature>
        < Shape rdf:resource="#Circle"/>        Fact
    </Contain-Feature>
</Object.>
```

Figure 4: Example of Object and Feature Classes with Enriched Facts Connected with Hierarchical Relationships

Table 3. The hierarchical relationships elements of the VDO Metadata

Name/ Description	Usability	OWL Element
Located-in/ hierarchy relation	Relationship that connects the scale variety classes.	***owl:ObjectProperty rdf:ID="Located-in"***
Contain/ hierarchy relation	Inverse relationship of 'Located-in'	***owl:ObjectProperty rdf:ID="Contain"***

Table 4. The content relationships elements of the VDO Metadata

Name/ Description	Usability	OWL Element
Adjacent/ content relation	Relationship that connects the classes of similar scale.	***owl:ObjectProperty rdf:ID="Adjacent-to"***

Multiple types of low-level features are utilized in the visual ontology (such as color and texture); each feature is a sub-class of the Feature class. The ***values*** of these features, which will be connected to visual objects, are encoded as properties. The association between the values of the low-level feature and the real objects must be tied with an uncertainty value. Similarly, the association of objects to each other using the *Located-in* and *Adjacent-to* relationships should also be tied with uncertainty value. The uncertainty value describes the degree of belief associated with some relationships. To solve the association problem and avoid the encoding problem, the VDO uses a complex class, a notion of anonymous individuals, and complex relationships.

In a single triple, the subject and relationship are defined typically; the object of the triple is a complex class, which encapsulates the object itself and the uncertainty using another triple. Thus, two triples are overlapped to allow the five tuples to be represented using the RDF triples. Consider the following example: the two following five tuples {*Apple hasColor Green hasUncertainty 0.6*}, {*Apple hasColor Red hasUncertainty 0.6*} and {*Grass hasColor Green hasUncertainty 0.7*} are represented in VDO using the following triples for the first tuple: {*Apple hasComplexClass UnnamedClass1*}, {*UnnamedClass1 hasColor Green*}, {*UnnamedClass1 hasUncertainty 0.6*} The following triples for the second tuple: {*Apple hasComplexClass UnnamedClass2*}, {*UnnamedClass2 hasColor Red*} and {*UnnamedClass2 hasUncertainty 0.4*}. The following triples for the third tuple: {*Grass hasComplexClass UnnamedClass3*}, {*UnnamedClass3 hasColor Green*} and {*UnnamedClass3 hasUncertainty 0.7*}. The "UnnamedClass" is a typical class in OWL that is defined using *owl:class* tag. In VDO, various complex classes are defined, as follows: *Complex-Feature*, *Complex-Blob*, *Complex-Object,* and *Complex-Region.* Similarly, the "hasComplexClass" is a typical relationship in OWL that is defined using *owl:ObjectProperty* tag. The complex relationships in VDO are: *Contain-Complex-Feature*, *Contain-Complex-Blob,* and *Contain-Complex-Object.*

Overall, this representation enlarges the ontology extensively and increases the unnecessary naming for the complex classes. To solve this problem, the VDO makes use of the notion of *Anonymous instance,* which is provided by the ontology language OWL to eliminate the unnecessary naming. Therefore, each instance of the complex class is an anonymous instance which has no name as represented in the previous triples using 'UnnamedClass1' and 'UnnamedClass2'. The uncertainty property is given in Table 5. The structure of the complex classes is illustrated in Figure 5.

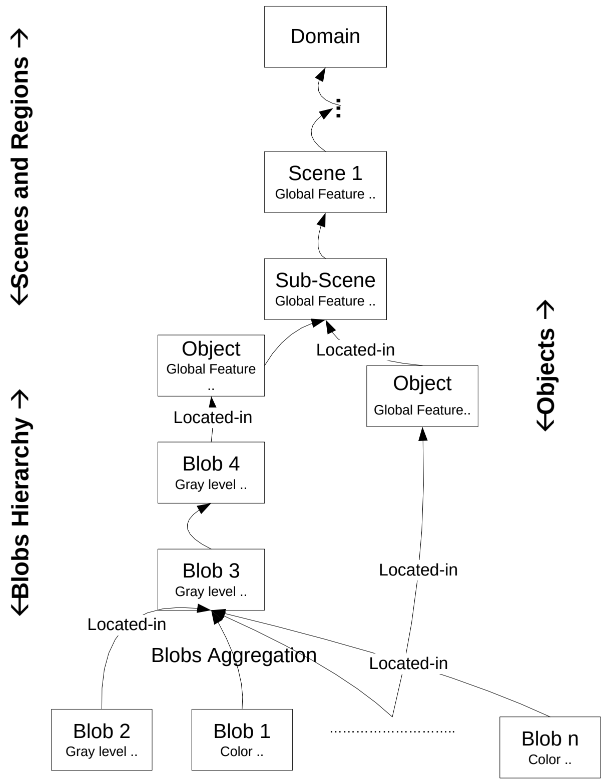

Figure 5. Structure of classes' metadata

Table 5. The uncertainty element of the VDO Metadata

Name/ Description	Usability	OWL Element
Uncertainty	Property that is combined with other properties and relationships.	***owl:DatatypeProperty rdf:ID="Uncertainty"***

Figure 6. illustrates a part of VDO metadata with facts. Figure 7. illustrates a part of the VDO for fruit domain visualized using TGViz.

```
<Object rdf:ID="Apple">
  <Contain-Complex-Feature>
    <Complex-Feature>
       <Contain-Feature>
          <Color rdf:ID="Red"/>
       </Contain-Feature>
       <Uncertainty
       rdf:datatype="http://www.w3.org/2001/XMLSchema
       #float"> 0.3 </Uncertainty>
   </Complex-Feature>
  </Contain-Complex-Feature>
  <Contain-Complex-Feature>
   <Complex-Feature>
     <Uncertainty rdf:datatype="http://www.w3.org
     /2001/XMLSchema #float">0.6</Uncertainty>
   <Contain-Feature rdf:resource="#Green"/>
  </Complex-Feature>
 </Contain-Complex-Feature>
 <rdf:type rdf:resource="#Fruit"/>
</Object.>
```

Figure 6: Part of VDO using OWL

[1] http://users.ecs.soton.ac.uk/ha/TGVizTab/

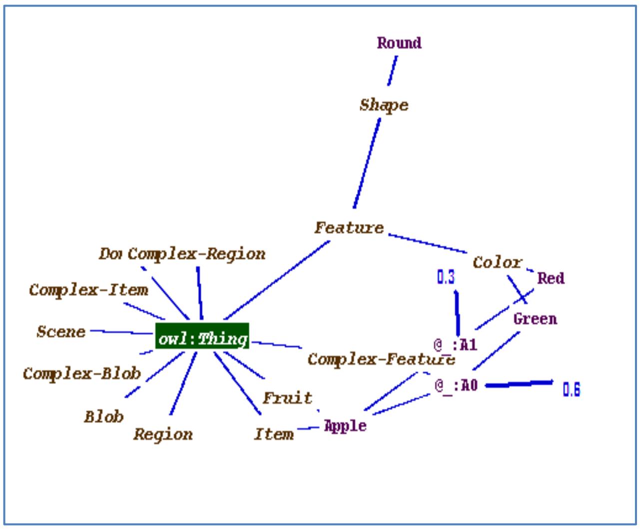

Figure 7. Visualization of VDO Metadata and Facts using TGViz

4.2. VDO Overall Structure with Example

Figure 8. illustrates a simple VDO example that represents visual facts acquired from a single image (image acquired from pascal [23]). The hierarchical structure is built using the relationships *Located-in*. The lower classes are parts of the higher one. Moreover, some of the classes can be *Located-in* other components with the same scale, especially for blobs. As such, several blobs can be combined to form another blob, and so on. Using such ontology in recognition process, a blob is recognized/predicted through a given color. A higher-scale object that is connected to that region can be recognized/predicted sequentially. As shown in Figure 8., the *leg* in the structure, which attaches to the human and animal, can infer the *nature-scene* in a Bottom-up flooding mechanism.

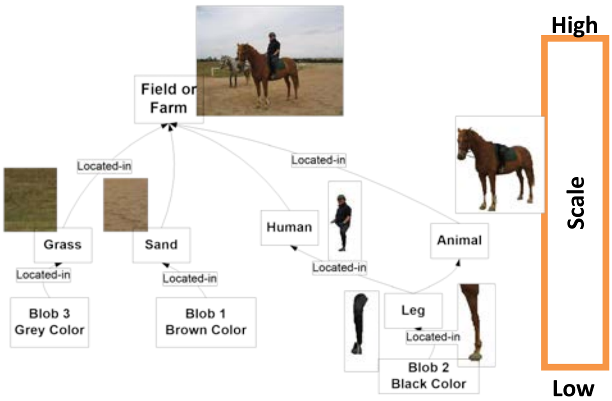

Figure 8. Structure of classes' metadata for specific image

4.3. Ontology Enrichment and Domain Knowledge

Semantic image processing through the extracted visual data (low-level features, spatial relationships, and content information) remains inaccurate in a general sense. The context knowledge can aid in disambiguating objects from one another. For example, if a soccer field is recognized, then the ambiguity of an object can be resolved to a *person* not a *scarecrow*; having recognized a *person* in a soccer field, the *person* may be further categorized as a *player* or *referee*, and so on. The context knowledge is represented in domain-specific ontology. The individuals in the domain knowledge, according to the ontology formalism, are the real objects which are also the same as in the case of the developed visual ontology. Thus, the VDO integrates the domain knowledge with the visual knowledge at the individual level.

The set of the classes in the domain-specific ontology is referred to as the *abstract type*, comparing to the *scale variety classes* for the visual ontology. The hierarchy of the abstract classes is carried out in the classification relationships. For example, the triple {*Pets – subClassOf – Animal*} represents a relationship between pets and animals. If the triple {*Cat – instanceOf – Pets*} exists in the domain ontology, and the triple {*Cat – instanceOf – Object}* exists in the visual ontology, then the integration of the visual and domain-specific ontologies occurred at 'Cat' node. Thus, the VDO will have Cat as an object, and cat is a pet, and pet is a subclass of animal.

4.4. Reasoning

Reasoning is used to extract knowledge that is not explicitly stated in the ontology or verify some assertions. OWL and most of the semantic web ontology languages are based on description logic and first-order logic. Thus, OWL uses a set of first-order rules/formula to reason about the elements of the ontology. For the VDO, most common reasoning rules that are required are presented in Table 6. These rules are simplified to allow easy implementation and avoid conflictions. Reasoning about the context is based on the 'InstanceOf' and 'subClassOf' Axioms, which allow transitive propagation between the physical objects of the *scale variety classes* and the knowledge in the *abstract type*. Reasoning about the spatially-connected classes is based on the transitive relationship '*Located-in'* and the developed notion of complex classes. This reasoning is combined with a probability calculation. However, the theory of probability-based reasoning is not addressed in this work and left to be determined based on the task at hand. Bayesian theory and Markov chain can be used for this purpose. Moreover, more reasoning rules may be added subsequently as needed.

Table 6. Reasoning Rules

Name/ Description	OWL Element	Usability
Transitive Property	(Pro.Rdf:type owl:TransitiveProperty)^ (X Pro. Y) ^ (Y Pro. Z) → (X Pro. Z)	Recognize round shape lead to recognize car. (Located-in Rdf:type owl:TransitiveProperty)^ (feature:Round Located-in Wheel) ^ (Wheel Located-in Car) → (Round Located-in Car)
Inverse Property	(Pro. InverseOf Pro.2)^ (X Pro. Y) → (Y Pro.2 X)	(Located-in inverseOf Contain) ^ (Round Located-in Car) → (Car Contain Round)
subClassOf	(X subClassOf Y) ^ (Y subClassOf Z) → (X subClassOF Z)	(Color subClassOf feature) ^ (Feature subClassOF Descriptor) → (Color subClassOf Descriptor)
Instance of	(I instanceOf X) ^ (X subClassOf Y) → (I instanceOf Y)	(farmer instanceOf Human) ^ (Human subClassOF Creature) → (farmer sinstanceOf Creature)

5. Implication

The practical utilization of the proposed VDO is subject to the availability of annotated image dataset and domain ontology. First, the metadata, classes, and other components are represented using OWL-support tools and APIs such as JENA [24] or Protégé [25]. Second, the labels in the image dataset are utilized with the OWL metadata as follows: The top-level labels in the images (e.g. those representing complete sense or sub-scene names) are added as *individuals* to the *class* ***scene***. The individuals of the scene and sub-scene classes are connected via the established relationships ***located-in***. Next, the labels of the objects in the images are added as *individuals* of the *class* ***object***. The same process is then applied for the object-parts which are added as *individuals* of the *class* ***blob***. Third, the image features for each label are extracted and initially saved in a table, and an uncertainty value is mined for each. The features and the uncertainty values are then attached to the ontology as explained earlier. Fourth, the spatial and content relationships are extracted and saved, associated with or without uncertainty value, and then attached to the ontology. Finally, the ontology is linked to the domain ontology; WordNet [26] can be used for this purpose.

The prediction/recognition of image labels is implemented by an over-segment of an input image with unknown labels. Features are extracted from each segment. The recognition/predicting process is performed over the VDO by using the flooding mechanism over the *scale variety classes*. The flooding process is started by matching the extracted low-level features with the features in the VDO. If the matched features are associated with blobs, the recognized blobs are led to recognize individuals of classes at higher scale. Otherwise, if the matched features are associated with objects and scenes, then a flooding is not needed. As individuals of different scales classes are recognized, a repeated flooding process can be implemented to extract the final individuals and use them to label the input image. Figures 9. and 10. illustrate the flow chart for construction and predicting phases respectively.

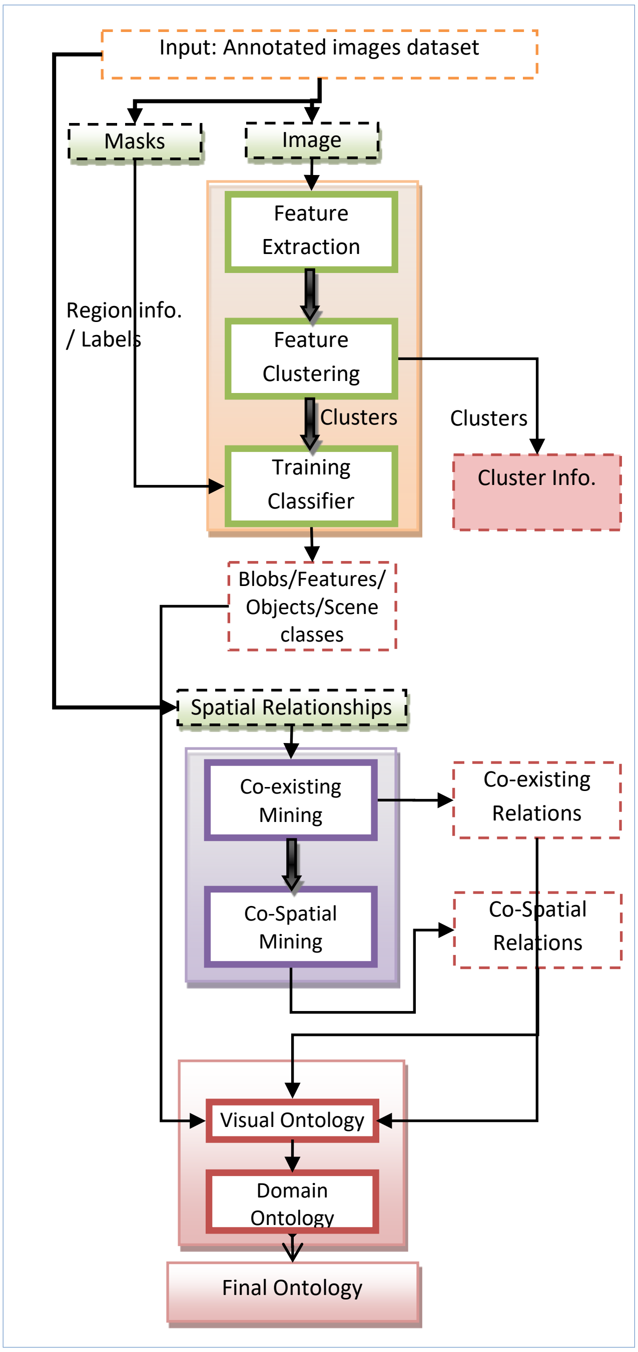

Figure 9. Construction phase of the VDO

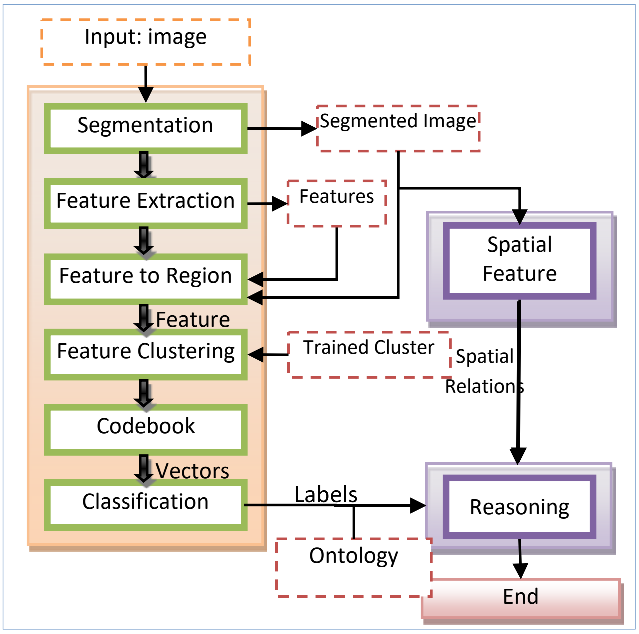

Figure 10. Predicting phase in VDO

6. Conclusion

Existing ontologies in the visual domain use general-purpose forms for ontology representation that do not allow formal representation, robust reasoning or enrichment for the represented knowledge. For an efficient representation and reasoning of complex visual knowledge, and to facilitate knowledge extension, sharing, and reuse, a formal ontology language such as OWL should be used. However, the simplicity of the OWL encoding, embodied in triples, renders the utilization of OWL in representing complex visual knowledge as non-trivial.

This paper proposes VDO using OWL-Lite. The VDO represents visual facts in a specific domain using the standard **ontology representation** OWL. The VDO passes through two execution phases: construction phase and inferring phase. In the construction phase, the **visual components** contain scale variety chain of classes that represent feature, object, and scene. As such, the low-level features are represented and linked to blobs; parts of an object are linked to their object identities; and objects of a scene are linked to the scene identity through hierarchical relationships. As for the **ontology components**, the visual components of features-blobs-objects-scenes are represented in OWL classes. These classes relate to hierarchical relationships that create the **ontology structure**. The components are also linked to each other through non-hierarchical content and contextual relationships.

Generally, the proposed VDO solved the problems of the previous works by enabling general purpose semantic-based processing, knowledge enrichment, and knowledge reasoning by which the knowledge can be more realistic and rational. The advantages of the VDO over existing ontologies are as follows: this ontology provides a standard representation of the visual knowledge using OWL and enables the representation of complex relationships (uncertainty, spatial, and contextual) in a single structure, thereby enabling advanced inferring and predicting process. VDO enables general-purpose semantic-based image processing and fully enables knowledge enrichment and knowledge reasoning by which the knowledge can be more realistic and rational. VDO provides a standard manner to represent visual knowledge using OWL, a standard language that is built on top of RDF and has well-defined syntax and semantics that allows for easy representation and reasoning process. Using a standard representation also eases the interpretability of complex information (uncertainty, spatial, and contextual information). Such representation of complex knowledge in a single source enables efficient inferring and predicting process. Subsequently, the contributions of this paper encompass using a standard language, OWL, to represent non-standard visual knowledge and defining the rules for inferring from such representation.

References

[1]. Abu-Shareha, A. A., & Rajeswari, M. (2015). A Review on Ontology-Based Label Extraction from Image Data. *Journal of Theoretical & Applied Information Technology*, *71*(2), 268-280.

[2]. Kohnen, M., Vogelsang, F., Wein, B. B., Kilbinger, M. W., Guenther, R. W., Weiler, F., ... & Dahmen, J. (2000, June). Knowledge-based automated feature extraction to categorize secondary digitized radiographs. In *Medical Imaging 2000: Image Processing* (Vol. 3979, pp. 709-718). International Society for Optics and Photonics.

[3]. Lehmann, T. M., Güld, M. O., Deselaers, T., Keysers, D., Schubert, H., Spitzer, K., ... & Wein, B. B. (2005). Automatic categorization of medical images for content-based retrieval and data mining. *Computerized Medical Imaging and Graphics*, *29*(2-3), 143-155.

[4]. Rahman, M. M., Antani, S. K., Demner-Fushman, D., & Thoma, G. R. (2014, March). Biomedical image representation and classification using an entropy weighted probabilistic concept feature space. In *Medical Imaging 2014: PACS and Imaging Informatics: Next Generation and Innovations* (Vol. 9039, p. 903908). International Society for Optics and Photonics.

[5]. Berka, P., Athanasiadis, T., & Avrithis, Y. S. (2006, December). Rule-based Reasoning for Semantic Image Segmentation and Interpretation. In *SAMT (Posters and Demos)*.

[6]. Bauer, T., & Strauss, P. (2014). A rule-based image analysis approach for calculating residues and vegetation cover under field conditions. *Catena*, *113*, 363-369.

[7]. Maillot, N. E., & Thonnat, M. (2008). Ontology based complex object recognition. *Image and Vision Computing*, *26*(1), 102-113.

[8]. Kanimozhi, T. and A. Christy.(2014). Applications of Ontology and Semantic Web in Image Retrieval and Research Issues. *International Journal of Computer Science and Information Technologies*, *5*(1), 763-769.

[9]. Abu-Shareha, A. A., & Mandava, R. (2011, June). Semantics Extraction in Visual Domain Based on WordNet. In *2011 Fifth FTRA International Conference on Multimedia and Ubiquitous Engineering* (pp. 212-219). IEEE.

[10]. Hanbury, A. (2008). A survey of methods for image annotation. *Journal of Visual Languages & Computing*, *19*(5), 617-627.

[11]. Penta, A., Picariello, A., & Tanca, L. (2007, September). Towards a definition of an Image Ontology. In *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)* (pp. 74-78). IEEE.

[12]. Liu, Y., Zhang, J., Tjondronegoro, D., & Geve, S. (2007, December). A shape ontology framework for bird classification. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)* (pp. 478-484). IEEE.

[13]. Lorenzatti, A., Abel, M., Nunes, B. R., & Scherer, C. M. (2009, November). Ontology for imagistic domains: Combining textual and pictorial primitives. In *International Conference on Conceptual Modeling* (pp. 169-178). Springer, Berlin, Heidelberg.

[14]. Khan, L., & Wang, L. (2002, October). Automatic Ontology Derivation Using Clustering for Image Classification. In *Multimedia Information Systems* (Vol. 56, p. 65).

[15]. Torresani, L., Szummer, M., & Fitzgibbon, A. (2014). Classemes: A compact image descriptor for efficient novel-class recognition and search. In *Registration and Recognition in Images and Videos* (pp. 95-111). Springer, Berlin, Heidelberg.

[16]. Iakovidis, D. K., Schober, D., Boeker, M., & Schulz, S. (2009, November). An ontology of image representations for medical image mining. In *2009 9th International Conference on Information Technology and Applications in Biomedicine* (pp. 1-4). IEEE.

[17]. Minu, R. I., & Thyagarajan, K. K. (2013). A novel approach to build image ontology using texton. In *Intelligent Informatics*(pp. 333-339). Springer, Berlin, Heidelberg.

[18]. Agius, H. (2008). Mpeg-7: Multimedia content description interface. *Encyclopedia of Multimedia*, 475-483.

[19]. Ghosh, H., Chaudhury, S., Kashyap, K., & Maiti, B. (2007). Ontology specification and integration for multimedia applications. In *Ontologies* (pp. 265-296). Springer, Boston, MA.

[20]. Harit, G., Chaudhury, S., & Ghosh, H. (2006, December). Using multimedia ontology for generating conceptual annotations and hyperlinks in video collections. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 211-217). IEEE Computer Society.

[21]. Abdelmoty, A. I., Smart, P. D., Jones, C. B., Fu, G., & Finch, D. (2005). A critical evaluation of ontology languages for geographic information retrieval on the Internet. *Journal of Visual Languages & Computing*, *16*(4), 331-358.

[22]. Fauqueur, J., & Boujemaa, N. (2004). Region-based image retrieval: Fast coarse segmentation and fine color description. *Journal of Visual Languages & Computing*, *15*(1), 69-95.

[23]. Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, *88*(2), 303-338.

[24]. Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., & Wilkinson, K. (2004, May). Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters* (pp. 74-83). ACM.

[25]. Knublauch, H., Fergerson, R. W., Noy, N. F., & Musen, M. A. (2004, November). The Protégé OWL plugin: An open development environment for semantic web applications. In *International Semantic Web Conference* (pp. 229-243). Springer, Berlin, Heidelberg.

[26]. Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39-41.