

Ontology Based Image Retrieval in the Compressed Domain

Suhendro. Irianto¹, Bobby Bachry²

^{1,2}Department of Informatics, Darmajaya Informatics and Business Institute, Bandar Lampung, Lampung, Indonesia

Abstract: Image Ontology based image retrieval has been presented in this paper, the work also present visualization approach that aims to building effective image retrieval with high accuracy. In order to calculate the accuracy of image retrieval, the work used a method called accuracy classified. Accuracy classified can be defined as item classified correctly divided by multiply by 100%. In this research DataGenie used ODBC to connect to a MySQL database. The research demonstrates that accuracy for each class greater than 50 % except for cat and emu classes which have of Life Object Super Class. In additional to demonstrating potential for high accuracy, the Image ontology proposed can also be used to visualize the relationship of each instance with its slot. The useful feature of protégé demonstrated in this work was its ability to migrate any relational database into it. Furthermore the image ontology can also be saved as XML or HTML files so they can be used by any web application

Keywords: Image ontology, Protégé, jambalaya, DataGenie

1. Introduction

According to Wikipedia, ontology is a study of conceptions of reality and the nature of being. In philosophy, ontology is the study of being or existence, which forms the basic subject matter of metaphysics. It seeks to describe or posit the basic categories and relationships of being or existence to define entity types within its framework. It also stated by [1] that Ontology is a specification of an abstract, simplified view of the world. Ontology is defined as a set of representational terms called concepts and ontologies can be constructed in two ways: domain-dependent or generic. Generic ontologies are definitions of concepts in general such as WordNet (<http://wordnet.princeton.edu>) which defines the meaning and interrelationships of English words. A domain-dependent ontology generally provides concepts in a specific domain, which focuses on the knowledge in the limited area, while generic ontologies provide concepts more comprehensively.

Ontology-based image retrieval has two components: semantic image annotation and semantic image retrieval. Semantic image annotation focuses mainly on the description of image content, and tries to describe image content as fully as possible. Using the resulting semantic content description, semantic image retrieval allows searching and retrieval based on image content. Compared with text-based metadata image retrieval and content-based image retrieval, ontology-based image retrieval is more focused on capturing semantic content, which has the potential to satisfy user requirements better.

Through semantic annotation, both images and retrieval queries can be formalized as XML files. In semantic annotation, the semantic meanings of images and queries are described based on a combination of concepts defined in ontology. In image retrieval, the goal is to determine the similarity between images and a retrieval query. To achieve this objective in ontology-based image retrieval, similarity is implemented by comparing in two steps: extraction of combined concept entities and similarity comparison between images and a retrieval query.

There are three kinds of inter-relationships generally represented in an ontology: "Is-a", "Instance-Of", and "Part-Of". These relationships correspond to key abstraction primitives in object-based and semantic data models. Furthermore [1] stated that "Instance-Of" relation shows membership between concepts, while "Part-Of" shows composition relationships. When a concept has an "Is-a" relation to another concept, this means that the second concept is more general than the first concept. If concept X has relation "Is-a" to concept Y, concept X is called a sub concept and concept Y is called a super concept. For instance, the concept "People" has "Is-a" relation to concept "Life Object", which is a more general concept compared with "People". Thus "Life Object" is superconcept to the concept "People".

One characteristic of the "Is-a" relation is that all the attributes of a super-concept can be inherited by its sub-concepts. Sub-concepts normally have more attributes than super-concepts and as a result, sub-concepts are more specific. The "Is-a" relationship is very important in similarity comparisons between a retrieval query and images. When trying to find an image including a super-concept, the image with sub-concepts should also be considered similar in image retrieval.

[2] stated that if image search is based on keywords then the search method does not solve the answer quality problem of keyword-based search, which is due to the fact that a set of keywords does not describe accurately the contents of images. For more accurate descriptions, semantically richer ontology-based annotations can be employed. Such annotations are not atomic keywords but can be more detailed structured descriptions that are linked with other resources in a semantic graph. With the help of ontologies, the user can also express the queries more precisely and unambiguously, which leads to better accuracy.

The main difficulty with the ontology based approach is the extra work needed in creating the ontology and the detailed annotations. However, in many applications this cost is justified due to the better accuracy obtained in information

retrieval and to the new semantic browsing facilities offered to the end-user. The trade-off between annotation work and quality of information retrieval can be balanced by using less detailed ontologies and annotations, if necessary.

Ontologies are central to the architecture of the Semantic Web, the next generation Web that is currently under active development [3,4]. Further more [4] stated that In the context of computer and information sciences ontology can be defined as a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (sets), attributes (properties), and relationships (relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application. In the context of database systems, ontology can be viewed as a level of abstraction of data models, analogous to hierarchical and relational models, but intended for modeling knowledge about individuals, their attributes, and their relationships to other individuals.

Undeniably, images will be major constituents of the Semantic Web, and how to share, search and retrieve images on the Semantic Web is an important and challenging research problem. Unlike other resources, the semantics of an image is implicit in the content of an image. Although this is not a problem for human recognition, it poses a challenge for image searching and retrieval based on the semantics of image content. Manual annotation of images provides an opportunity to make the semantics of an image explicit and richer. As [3] mentioned that annotation is the process of describing images and retrieval stands for the process of finding images. There are two major approaches to image retrieval. First is content based image retrieval which analyzes the actual image data and the second is a metadata-based approach that retrieves images based on human-annotated metadata. Also relevance feedback has been used in image retrieval complementing text-based systems. In this thesis the retrieval is done by using the annotated metadata and the content-based analysis.

However, different annotators might use different vocabularies to annotate images, which could lower relevancy in retrieval. This work proposes to build an ontology-based annotation which is created for a particular domain so that the terms and their relationships are formally defined. In this way, annotators of a particular image domain, the "THING" domain will use the same vocabulary to annotate images, and users will search images guided by the ontology with greater ACCURACY.

2. Ontology and protégé

For years, there has been the development of ontology tools that let experts develop ontologies and populate them with knowledge. With more registered users, Protégé has become the most widely used, freely available, platform-independent open-source technology for developing and managing large terminologies, ontologies, and knowledge bases. The Protégé system was designed as an open, modular platform upon which developers can build custom-tailored functionality.

Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data.

Keizer and Abu-Hanna [5,6], explained that many projects have used Protégé as their primary development environment. These include Corner's Clinical Bioinformatics Ontology, the DICE TS (Diagnoses for Intensive Care Evaluation Terminological System) MGED Ontology, and verification and identification of errors and inconsistencies in the Gene Ontology [7]

However, Protégé is not only an ontology-development and knowledge-acquisition tool, but also a platform for developing knowledge-based applications and, more specifically, semantic web applications. Its knowledge model, based on the Open Knowledge Base Connectivity protocol [8] (supports a flexible meta-modelling mechanism. Also mentioned by [5], the Protégé-OWL enables users to build ontologies for the *Semantic Web*, particularly in the Web Ontology Language (OWL). OWL ontology may include descriptions of classes, properties and their instances. Given such ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms.

Ontology plays a central role in the Semantic Web and it provides formal models of domain knowledge that can be exploited by intelligent agents. As a result, a development tool for Semantic Web applications should provide services to access, visualize, edit, and use ontologies. In this work I use Protégé – OWL to create image ontology.

Mean while [9] described that this mechanism lets developers build editors for different ontology languages. Protégé plug-ins support ontology editing in both RDF Schema and OWL. In fact, the Protégé OWL plug-in is arguably the most widely used editor for OWL ontologies. Protégé's architecture also lets developers extend the environment with a wide range of plug-ins, which perform various types of inference, provide visualization mechanisms, support queries, and enable access to and integration with standard terminologies, such as UML. Finally, since the Semantic Web is open to so many potential application areas, a tool should be customizable and extensible. The OWL Plug-in, an extension of Protégé, Protégé [7,9] is an open platform for ontology modeling and knowledge acquisition. The OWL Plug-in can be used to edit OWL ontologies, to access description logic (DL) reasoner, and to acquire instances for semantic mark-up. As an extension of Protégé, the OWL Plug-in profits from the benefits of a large user community case of a library of reusable components, and a flexible architecture. The OWL Plug-in therefore has the potential to become a standard

infrastructure for building ontology-based Semantic Web applications.

PROMPT is a suite of ontology management tools, which supports semi automated ontology merging and mapping. The ontology-versioning support includes a structural comparison of ontology versions and a mechanism to accept and reject changes. A view mechanism lets you extract views from large ontologies. Many of these functionalities can be wrapped in Web Services, becoming a value-added set of services that the ontology repositories provide. [10] stated that a life-science researcher could access a repository, determine ontologies that are potentially interesting; compare them to each other and to other standard terminologies and follow development across different versions. In this work, the latest version of Protégé, Protégé 3.2.1, was used. This version has more than 15 plug-ins but the work described here only utilized the four of them which are the most relevant for building image ontology, namely: jambalaya, DataGenie, OntoViz and TGVizTab.

Jambalaya is a plug-in created for Protégé which uses “shrimp” to visualize the knowledge bases the user has created. The Jambalaya Tab is composed of the Protégé class and instance panels as used in the rest of Protégé, and the Jambalaya Main View. Jambalaya has several ways of saving state. It provides the option of exporting an image (in .jpg or .png file formats). It takes a picture of the current screen. Jambalaya also saves the current layout upon exit, and offers the choice of revisiting that layout upon re-opening that ontology. There are also snapshots which can be saved on a filmstrip and HTML bookmarks containing SVG (Scalable Vector Graphics) files.

In Jambalaya Protégé, concepts are mapped as nodes in the graph, and are represented as complex objects, typically shaped as rectangles. Slots are mapped as graph edges, and join nodes. They can be followed like hyperlinks to navigate between nodes. Nodes and arcs can be clustered. Nodes are clustered using the abstraction inherent in the hierarchical structure such as an “is-a” relationship. Arcs are clustered using a feature that is referred to as *composites*, which are groups of different slot types. Composite arcs can also show where relationships between nodes exist at lower levels in the hierarchy than currently shown. The elements of Jambalaya also support the notion of filtering, which implies that certain types of each element can be turned ON or OFF in the graph view.

Jambalaya also provides many different tools that can be used to describe the functionality such as: Main View (used to display the different layouts and views of the ontology), Hierarchical View (uses default Protégé “is-a” relationship to structure the tree), Treemap view (uses a Treemap algorithm to lay out nodes in a space-filling manner), Search (used to search functionality, using the plain-text name of nodes and arcs) and Script Manager (used to allow users to write scripts to access functionality).

DataGenie is a tab plug-in that allows Protégé to read from arbitrary databases. In this research DataGenie used ODBC to connect to a MySQL database, and allowed movement of portions (or all) of the database into Protégé classes, where

each table became a class and each attribute became a slot. In addition, if the relational database table has foreign key references to other tables, these can be replaced by Protégé instance pointers when the database is converted into a knowledge base.

The DataGenie plug-in is not a database back-end and is used when there is legacy data that one wants to transfer into Protégé before doing additional knowledge acquisition or knowledge modeling. This plug-in (as written) does not include any capability for moving data in the opposite direction (from Protégé classes and instances into a relational database). The DataGenie plug-in might be used as a database viewer. For efficiency, a database might be stored as a set of custom-designed database tables, but then the DataGenie could be used to view portions of this schema in the Protégé frame-based user interface.

TGVizTab is a Protégé plug-in which allows ontologies to be visualized using the TouchGraph library (<http://www.touchgraph.com>). Touch Graph provides a java library for rendering networks as interactive graphs. This library has been modified and integrated with Protégé as a tab plug-in. TGVizTab plug-in has features such as: visualizing classes and instances, network depth control, change graph and slot colours, hide/show individual slots and nodes, hyperbolic zooms, graph rotation, node search, and save or load graphs and settings. Hyperbolic zooms uses hyperbolic geometry to lay out ontology and it uses a 3D representation to allow more elements to be fit into the limited space. Figure 1, shows a TGViz snapshot of the Image Ontology proposed.

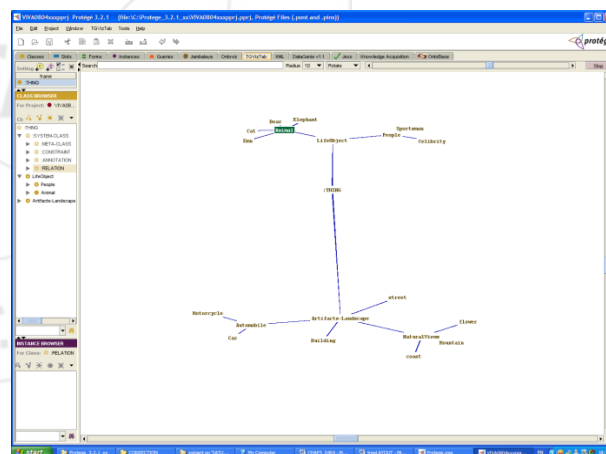


Figure 1: Visualization of Life Object and Artefact Landscape Classes using the TGViz plug-in

The OntoViz Tab allows users to visualize Protégé ontologies with the help of the highly sophisticated graph visualization software called Graphviz (<http://www.graphviz.org>) from AT&T. The types of visualizations are highly configurable and include:

- Picking a set of classes or instances to visualize part of ontology.
- Displaying slots and slot edges.
- Specifying Colours for nodes and edges.
- When picking only a few classes or instances, users can apply various closure operators (e.g., subclasses, super classes) to visualize their vicinity.

The Graph visualization of image ontology proposed using OntoViz plug-in for subclass, super class and slot extension is displayed in Fig 1 whilst its inverse slot extension, slot and slot edges is shown in Figure 2. Image ontology system visualization can be examined in Figure 3.

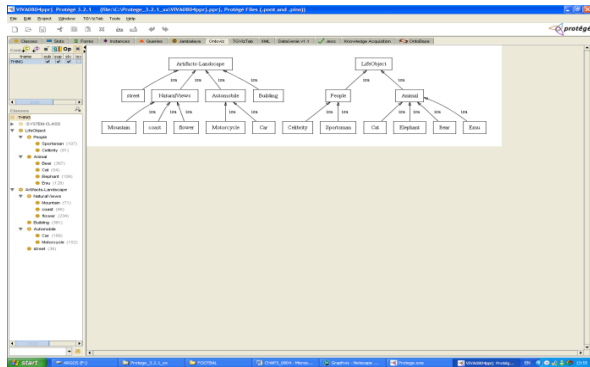


Figure 2: Graph visualization of subclass, super class and slot extension generated by OntoViz plug-in

3. Building Image Ontology

The research was carried out by constructing an Image Ontology which consists of two super classes Life-Object and Artefact-Landscape. Life-Object has *People* and *Animal* classes, whilst the Artefact-Landscape has *Automobile*, *Building* and *NatureView* sub-classes. Further details of the super classes, classes, and sub-classes are shown in Figure 4 and Figure 5. Seven slots were constructed in this image ontology, namely: *category*, *description*, *event*, *idname*, *imagename*, *keyword*, and *image*.

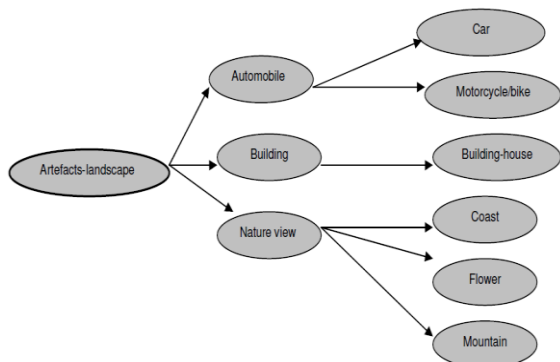


Figure 3: Super class Artefacts and its classes and sub-classes

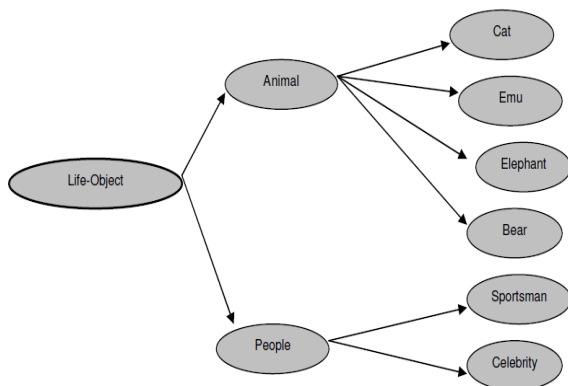


Figure 4: Super class Life Object and its classes and sub-classes

Table 1: An example of slot which is saved as HTML file of elephant instance and generated by Protégé

Own Slots		
	Slot Name	Value
■	category	Animal
■	description	Elephant (Elephantidae) is a family of pachyderm, and the only remaining family in the order Proboscidea in the class Mammalian. Elephantidae has three living species. The African Bush Elephant, the African Forest Elephant, and the Asian Elephant. Other species have become extinct since the last ice age, which ended about 10,000 year ago, the Mammoth being the famous of these. Kingdom: Animalia, Phylum: Chordata, Subphylum: Vertebrate, Class: Mammalia, Order: Proboscidea, Superfamily: Elephantidae.
■	event	African's elephant in the National zoo
■	idname	elep011
■	image	1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009, 1010
■	imagename	elephant0011.jpg
■	keyword	Elephant, animal, wildlife, mammoth, gigantic, jungle, school, training, zoo, Thailand, Indonesia, Africa. Elephant in the African Jungle.

In this work, approximately 3000 instances were created, each with its properties or attributes for 12 classes. Every instance has attributes which explain the description, category, event, image identification, image name, image keyword and image content. For each instance there are seven slots which enhance its image identification: *category* (i.e. people, animal, automotive etc), *description* (describes what the image is about), *event* (what event is about), *image file name*, *image keyword* and *image content* which has an associated value embedded in it.

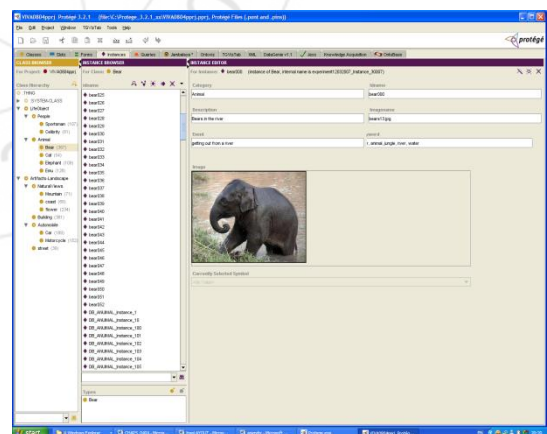


Figure 5: Example of elephant class

The Jambalaya visualization plug-in represents concepts and relationships using the graph metaphor as shown in Figure 2. Classes and instances are represented as nodes in a graph. The arcs are used to show the relationships between concepts and instances and "a" relationship between classes, in the concept hierarchy. The general purposed language for representing information in the Web (Resource Description Framework or RDFs structure) of image slot using protégé described as follows:

```
<class>
<name>LifeObject</name>
<type>:STANDARD-CLASS</type>
<own_slot_value>
  <slot_reference>:ROLE</slot_reference>
  <value value_type="string">Abstract</value>
</own_slot_value>
<superclass>:THING</superclass>
<template_slot>category</template_slot>
<template_slot>description</template_slot>
<template_slot>event</template_slot>
<template_slot>idname</template_slot>
<template_slot>image</template_slot>
<template_slot>imagename</template_slot>
<template_slot>keyword</template_slot>
</class>
```

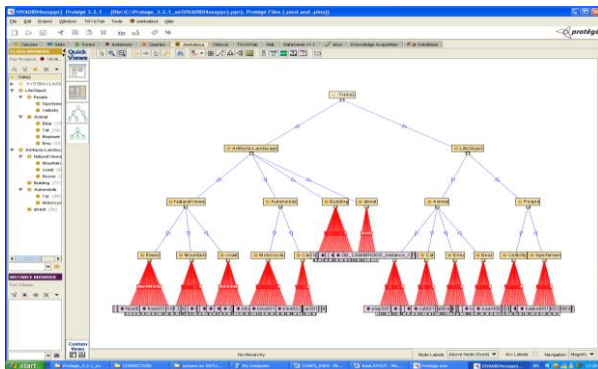


Figure 5: The Jambalaya plug-in for Protégé (vertical tree visualization of image classes and instances) using Jambalaya

Jambalaya Protégé, concepts are mapped as nodes in the graph, and are represented as complex objects, typically shaped as rectangles. Slots are mapped as graph edges, and join nodes. They can be followed like hyperlinks to navigate between nodes. Nodes and arcs can be clustered. Nodes are clustered using the abstraction inherent in the hierarchical structure such as an “is-a” relationship. Arcs are clustered using a feature that is referred to as *composites*, which are groups of different slot types. Composite arcs can also show where relationships between nodes exist at lower levels in the hierarchy than currently shown. The elements of Jambalaya also support the notion of filtering, which implies that certain types of each element can be turned ON or OFF in the graph view.

Jambalaya also provides many different tools that can be used to describe the functionality such as: Main View (used to display the different layouts and views of the ontology), Hierarchical View (uses default Protégé “is-a” relationship to structure the tree), Tree map view (uses a Tree map algorithm to lay out nodes in a space-filling manner), Search (used to search functionality, using the plain-text name of nodes and arcs) and Script Manager (used to allow users to write scripts to access functionality). The different layouts and views of the proposed ontology are shown in Figure 10 to 15.

Fig 10. Spring layout visualization of classes and instances using Jambalaya

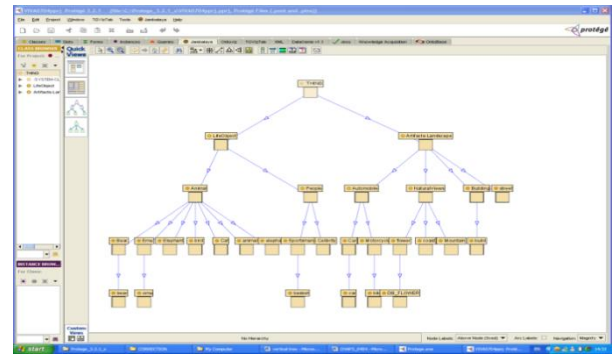


Figure 6: Vertical layout visualization of classes using Jambalaya

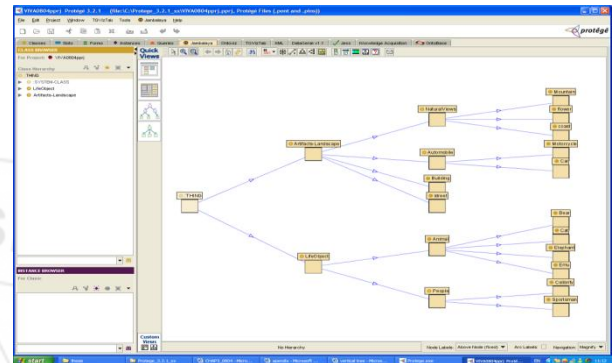


Figure 7: Horizontal layout visualization of classes using Jambalaya

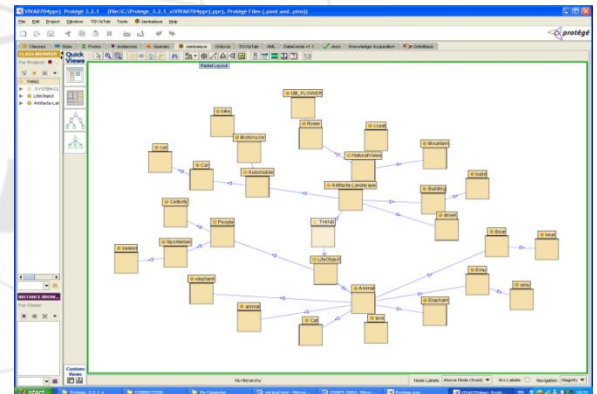


Figure 8: Radial layout visualization of classes using Jambalaya

4. Evaluations

In the experiment two super classes were introduced, Life-Object and Artefacts-Landscape. There are seven slots in the image ontology proposed which are *category*, *description*, *event*, *idname*, *imagename*, *keyword* and *image*. An example of instance for each category can be shown in Appendix A. When making a query, a filter is applied using “CONTAINS” a certain value associated with that image query, an example of a query result is shown in Fig 5.18. In order to calculate the accuracy of image retrieval, the work used a method that is widely used by many authors (Beretti et al. 2001; Keysers et al. 2003; Liu et al. 2000; Veropoulos et al. 1998; Constantinidis et al. 2001). The method is defined as follows:

$$\text{accuracy} = \frac{\text{items classified correctly}}{\text{all items classified}} \times 100\%$$

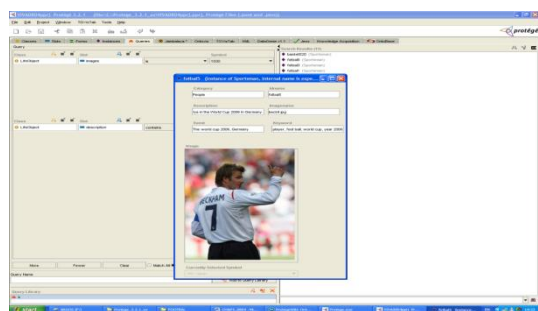


Figure 9: An example of a query result with value of 1001 and “World Cup 2006” description

In order to make instances accessible, slots should be assigned to a certain type of Widget. There are nine widgets in protégé 3.2.1, however only two were used in this work as they are the most relevance, namely: *TextFieldWidget* and *ImageMapWidget*.

Table 2: The accuracy of 12 classes for image ontology based retrieval of a query by entering “CONTAIN” an associate value of the image and its description

Category	Class	Accuracy (%)
Artefact-Landscape	Bike	95
	Car	96
	Building	89
	Coast	73
	Mountain	51
	Flower	70
Life Object	Bear	62
	Cat	47
	Emu	42
	Elephant	53
	Celebrity	79
	Sportsman	84

In protégé, the widget type menu allows you to choose the widget type for the currently selected widget. The choices for type of widget depend on the value type of the slot. Each value type has a default widget for the slot, and a preset list of possible widget types based on the slot's value type. The *TextFieldWidget* allows the user to enter text description about the slot. The *SymbolListWidget* provides a drop-down list that allows the end-user to enter a list of values (in this case integer values), and the *ImageMapWidget* allows the user to enter an image in the slot or instance.

As the queries were made for an image ontology consisting of more than 3000 instances with an image in every instance, the work demonstrates some high accuracy for some classes. *Bike, car, building, celebrity, general, coast and flower* demonstrate accuracy greater than 70% of instances retrieved. On the other hand *cat* and *emu* had accuracy less than 50%. Furthermore the accuracy of the system can be examined by viewing Table 5.1 and Figs 5.19, 5.20, 5.21 and 5.22.

From Table 5.1 it can be seen that ten of twelve classes achieved an accuracy of image retrieval greater than 50% except for *cat* and *emu* classes which have accuracies 47% and 42%, respectively. On the other hand *bike* and *car* have accuracy greater than 90%. The work used five image associated values and a description for each class to query

the proposed ontology. The values associated with image content uses are 1001, 1002, 1003, 1004 and 1005 and so on whilst the description can be one or more words such as “celebrity” or “an academy award ceremony”. These values were assigned previously to each image when the slots were defined.

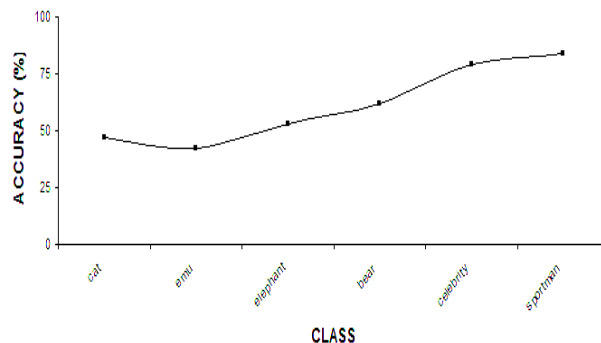


Figure 10: The accuracy of Life-Object Super Class

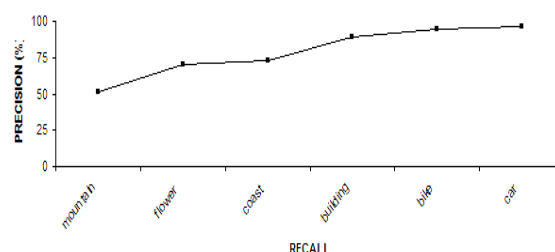


Figure 11: The accuracy of Artefact-Landscape Super Class

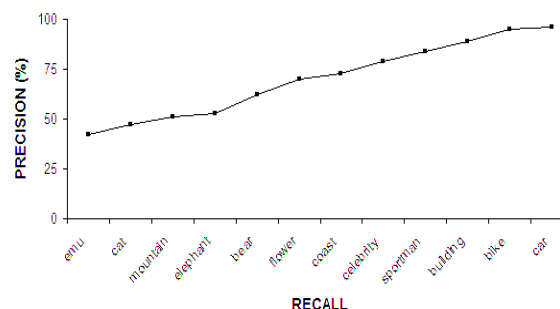


Figure 12: The accuracy of Artefact-Landscape and Life-Object Super Classes

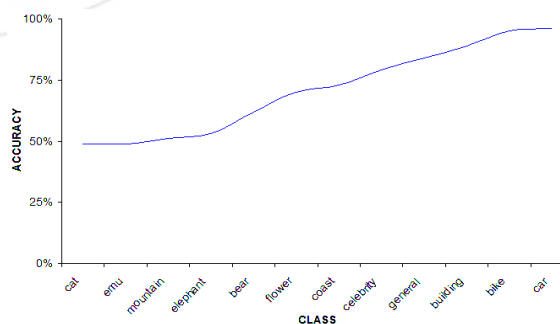


Figure 13: The accuracy of twelve instances

The experimental results are also calculated for the precision and recall besides their accuracy to evaluate the effectiveness of image ontology based retrieval. As the experiment was carried out, Artifact-Landscape super class show 76% in average precision and 16% for recall. Amongst instances in the Artifact-Landscape super class *car* instance performs the

highest precision at 96% whilst the lowest precision is 45% for *flower* instance. Meanwhile, the highest precision of 59% has shown by *general* instance and the lowest is 24% for *cat* instance. Moreover the average precision and recall for Artifact-landscape are 44% and 28%, respectively. Table 2000 shows a complete result of precision and recall of Image ontology based retrieval.

Table 3:The precision and recall of twelve instances

Super Class	Sub Class	Precision	Recall
Artefact-Landscape	Bike	0.95	0.01
	Car	0.96	0.07
	Mountain	0.76	0.11
	Coast	0.68	0.21
	Building	0.79	0.23
	Flower	0.45	0.35
Life Object	General	0.59	0.14
	Celebrity	0.54	0.16
	Bears	0.46	0.22
	Elephant	0.55	0.34
	Emu	0.24	0.34
	Cat	0.24	0.48

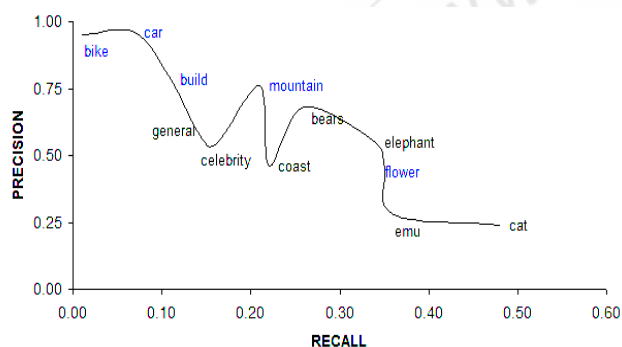


Figure 14: The precision and recall of twelve instances of Image Ontology image retrieval

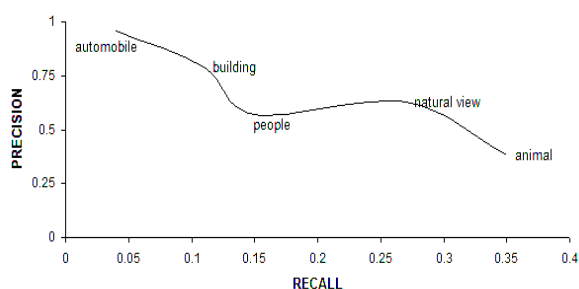


Figure 15: The precision and recall of six classes of Image Ontology image retrieval

Evaluations

Ontology based image retrieval was presented in the paper, an image ontology was constructed as a tool to improve the image retrieval accuracy. The image ontology was also used as a tool to visualize the relationships of images in the database.

Image Ontology based image retrieval has been presented in this paper, the work also present visualization approach that aims to building effective image retrieval with high accuracy. The research demonstrates that accuracy for each

class greater than 50 % except for cat and emu classes which have of Life Object Super Class. In additional to demonstrating potential for high accuracy, the Image ontology proposed can also be used to visualize the relationship of each instance with its slot. The useful feature of protégé demonstrated in this work was its ability to migrate any relational database into it. Furthermore the image ontology can also be saved as XML or HTML files so they can be used by any web application.

5. Acknowledgments

Our thanks to the Department of Higher Education, The Republic of Indonesia for their funding to this research, also thanks to Informatics and Business Institute who support the Laboratory facilities in carrying out the research.

References

- [1] Brickley. D and R. V. Guha, "Resource description framework (RDF) schema specification 1.0", In W3C Candidate Recommendation, 27 March, 2000.
- [2] Wan Y., Li P., Li R., Zhou M., Ye Y., Wang B. (2016) Towards Understanding URL Resources in Recent SinaWeibo. In: Cellary W., Mokbel M., Wang J., Wang H., Zhou R., Zhang Y. (eds) Web Information Systems Engineering – WISE 2016. WISE 2016. Lecture Notes in Computer Science, vol 10041. Springer, Cham
- [3] Yan Ma, Deokjai Choi, Shingo Ata. Challenges for Next Generation Network Operations and Service Management: 11th Asia-Pacific Network Operations and Management Symposium, APNOMS 2008, Beijing, China, October 22-24, 2008. Proceedings.
- [4] H. et al. An Ontology of Soil Properties and Processes. In: Groth P. et al. (eds) The Semantic Web – ISWC 2016. ISWC 2016. Lecture Notes in Computer Science, vol 9982. Springer, Cham
- [5] K. Setharaman, and Kamarasan. tistical framework for image retrieval based on multiresolution features and similarity method, Juornal Multimedia and Tools Application , Vol.73. Issue. 3,2014.pp.1943-1962
- [6] Abu-Hanna, A.; Cornet, R.; de Keizer, N.Crubezy, M and Tu,"Protégé as a vehicle for developing medical terminological systems. Elsevier on International Journal Of Human-Computer, Vol. 62, 2005, Pp. 639–663.
- [7] Mark A. Musen and the Protégé Team, The Protégé Project: A Look Back and a Look Forward, AI MATTERS, VOLUME 1, ISSUE 4, 2015, Pp.4-1
- [8] Chaudhri. V, A. Farquhar, R. Fikes,P. Karp and J. Rice,"OKBC: A programmatic foundation for knowledge base interoperability", In fifteenth national conference on artificial intelligence (AAAI-98), 1998, pp. 600–607, AAAI Press/The MIT Press.
- [9] Noy. N.F and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology", Technical Report SMI-2001-0880, Stanford Medical Informatics, 2001.
- [10] Noy.N.F and M.A. Musen, "The PROMPT suite: Interactive tools for ontology merging and mapping", Elsevier on International Journal of Human-Computer Studies, Vol.59, 2003, pp. 983-1024.

- [11] Guarino, N., and Musen, M.A. (2015). Applied Ontology: The next decade begins. Applied Ontology, 10(1):1–4.

Author Profile

Suhendro Y. Irianto received his Master in computers science from The University of Indonesia, Jakarta Indonesia. He got Ph.D degree image Retrieval from the University of Bradford, United Kingdom in 2008. Currently he works as a researcher at Informatics Engineering department, Darmajaya Informatics and Business Institute, Indonesia. His interests are image retrieval, biometrics, pattern recognition, and multimedia database.

Bobby Bahcry received his Master in computers science from The University of Gunadarma, Jakarta Indonesia. Currently he works as a researcher at InformationSystem department, Darmajaya Informatics and Business Institute, Indonesia. His interests are image retrieval, Network Security, and multimedia database.

