Semantic Web based Satellite Image Classification

A Dissertation submitted in partial fulfillment of the requirement for the award of minor project of

Master of Engineering

In

Computer Technology and Application

By

Sonal Kumar

College Roll No. - 17/CTA/07

University Roll No. – 12214

Under the esteemed guidance of

Dr. Daya Gupta

Department of Computer Engineering
Delhi College of Engineering
University of Delhi
2008-2009

DELHI COLLEGE OF ENGINEERING

CERTIFICATE

DELHI COLLEGE OF ENGINEERING
(Govt. of National Capital Territory of Delhi)
BAWANA ROAD, DELHI – 110042

Date:___________

This is to certify that dissertation entitled "**Semantic Web Based Satellite Image Classification**" has been completed by Sonal Kumar in partial fulfillment of the requirement of **Master in Engineering** in **Computer Technology & Application**.

This is a record of her work carried out by her under my supervision and support during the academic session 2008 -2009. This is a beneficial work in field of Ranking Technique for creating better sequence of web pages to the user query.

(Dr. DAYA GUPTA)
HOD & PROJECT GUIDE
(Dept. of Computer Engineering)
DELHI COLLEGE OF ENGINEERING
BAWANA ROAD, DELHI - 110042

ACKNOWLEDGEMENT

It is a great pleasure to have the opportunity to extent my heartfelt gratitude to everybody who helped me throughout the course of this project.

It is distinct pleasure to express my deep sense of gratitude and indebtedness to my learned supervisor Dr. Daya Gupta, HOD, Department of Computer Engineering, Delhi College of Engineering, for her invaluable guidance, encouragement and patient review. Their continuous inspiration only has made me complete this dissertation. Without her help and guidance, this dissertation would have been impossible. She provided the conceptions and theoretical background for this study as well as suggested us the rational approach. She remained a pillar of help throughout the project. I would also like to Thanks the team of DTRL lab of DRDO i.e. Sci. F V.K. Panchal, Research associate Mr. Shashi Kumar, for their sincere guidance and assistance to me, in understanding the complex concepts of “Remote Sensing”. I would also like to take this opportunity to present my sincere regards to my teachers Mrs. Rajni Jindal, Dr. S.K. Saxena, Mr. Manoj Sethi and other staff members of Computer Engineering Department for providing me unconditional and anytime access to the resources and guidance.

Finally we are also thankful to my classmates for their unconditional support and motivation during this work. Last but not least, I special thanks to the crowd who are active in the field of Semantic Web and knowledge representation.

Sonal Kumar
College Roll no: 17/CTA/07
University Roll no: 12214
M.E. (Computer Technology and Application)
Department of Computer Engineering
Delhi College of Engineering, Delhi-110042.

ABSTRACT

It is a great pleasure to have the opportunity to extent my heartfelt gratitude to everybody who helped me throughout the course of this project.

In this study we have proposed a framework for semantic web based satellite image classification. Satellite images have several features embed in it like spatial, spectral and temporal properties etc. Through these features object detection has been performed. Semantic web has its application in every domain, so we have used it in Remote Sensing for classification of satellite images by semantic web technology i.e. Ontology. This work is extending the concept of building the description of the geographic domain in the form of its ontology i.e. Geo-Ontology. We are making the geo-ontology for several generic characteristics of the features in earth observation satellite (EOS) images. Geo-Ontology is the effective specification of concepts in geo domain. So it has been used by us as a means for image classification like other standard techniques as Minimum Distance to Mean and Maximum Likelihood.

We are also extending our framework to web based framework for satellite image classification and making it available in a format which can be easily accessed and shared by different departments of an organization. We are making the geo-ontology for several spectral features in earth observation satellite images, and making them available on GIS servers, which can further be used for semantic web based satellite image classification.

We have taken the satellite images of Alwar Region from the DRDO lab to show the rocky, urban, water, vegetation and barren classes in that region as our case study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

With the development of modern information society, especially the development of computer network, more and more information need to be processed in different software's, and can be released to the network. In the domain of remote sensing information processing, spatial resolution, spectral resolution and time resolution of remote sensing information have been developing constantly, and obtainable remote sensing data has been increasing at striking rate. Consequently, it becomes very important to know how to make these integrated data mutually operable in different systems, and to know how to acquire needed data from the heterogeneously -distributed databases.

The domain experts use the concepts and terminology specific to their respective field of expertise to translate the meaning of foreign set of concepts and terms to their own terminology. Software systems usually do not have any knowledge about the world and have to be told explicitly how to map a concept into meaning full information. This paper is concerned with semantic web based information retrieval of geographical information.

The Semantic Web is used as a tool that provides structure for content that is meaningful, and, a virtual (and eventually a real environmental application) in which agents carry out sophisticated tasks for human beings. Geospatial information is the key for effective classification of images in Remote sensing application domains. Ontology, which is the description of the concept, specifically domain-specific ontology, is at the heart of most semantic approaches to interoperability. Geo-Ontology has been acknowledged to be the core methodology for capturing and sharing semantics of geospatial information. For the description of geospatial web services we need geospatial domain ontology as common ground to which different regions of image can be classified.

1.1 Motivation

In recent years, remote sensing data collection and archiving [26] have significantly increased. Landsat data alone comprises 434 terabytes of archive (31 years of Landsat 1-5; 165 terabytes, 4 years of Landsat 7; 269 terabytes). Multiple Petabytes of data from Earth Observation Satellites (EOS) and Pre-EOS are archived by NASA DAACs. Availability of such a magnitude of data to the users has raised important challenges regarding its archiving, the ability to summarize and evaluate data, and convert the volumes of data into meaningful information that can be used for decision-making. Semantic enrichment of the existing data sources is an important first step that would enable to

• Package products based on the meaning and knowledge about the measurements and context of the information sources.

• Support interoperability between different formats within an archive and between them.

• Enable creation of machine understandable semantic metadata so that intelligent search engines / agents can automatically process and index the content.

• Provide content and semantics driven interactive user interfaces which supports advanced query that go beyond just keyword based searches.

• Explore and discover spatio- temporal patterns.

• Dissemination of information through current standards (e.g. OGC Web Map service (**WMS**), Web Feature Service (**WFS**), Web Coverage Service (**WCS**), Web Catalog Service (**CS-W**) etc.) driven web services oriented architectures and extended to the semantic web services vision. The Semantic Web is an extension of the current web in which information is given a well-defined meaning, better enabling computers and people to work in cooperation. It allows data to be shared and reused across application, enterprise, and community boundaries. Using web languages, such as RDF, DAML+OIL, and OWL, it is possible to create semantically rich data models. We build upon the emerging semantic web technology and combine them with remote sensing image classification with inferences and reasoning to generate classified image.

1.2 Related work:

Several image retrieval systems are developed such as text based and Content based Image Retrieval. In text based keyword search is used for image retrieval. They require considerable amount of effort in manual annotation of image. To overcome above disadvantage several CBIR techniques came forward where content (color, texture, shape, etc.) of the image is used for its retrieval. For retrieval user provides the query image and system converts them into their internal representation of feature vectors. The similarity between the query image and those in database are calculated and images are retrieved with the aid of some indexing scheme. Still there exist a sort of semantic gap between the retrieved images and the query image in the form of what the user is perceiving and what he/she is looking for. Some existing CBIR systems are Virage System [32], PhotoBook [33] etc.

To overcome the above said drawback new techniques like Relevant feedback [27], Semantic Interpretation [28], Machine learning [29] and Ontology learning came forward. In relevant feedback user is provided with the result and then user judges the result by accepting some of them. In Machine learning several networks have been proposed like Bayesian Network, Neural Network and Support Vector Machines (SVM) for image annotation. In ontology based approach [3] authors are defining high level concepts which describe each region of an image by its average color in lab color space.

As semantic web is "*meaning to the web*" it has been used in every field to make the domain data interoperable and available. It has been applied in Remote Sensing as well. Semantic Web has been used extensively for proposing the framework for building geospatial ontology's. Initially with the advent of web 2.0 several digital communities came into existence. Initially semantic web is used to search same interest people on social networks like Friend of a Friend (FOAF) [7, 8], Wikipedia. etc. Later on, it spread its reach to geology also where they implement several Online Map finders[9,10,11] searching for person provided with some address along with view on Google Earth/Map, Yahoo Map etc.

Classification of Remote sensing Information Processing has been done by combining the semantic web service with information processing [30]. Semantic web has several technologies which can be used to make application area more dynamic and sharable. So building a semantic service environment by combining the advantages of these two respects and conducting service-oriented research into semantic service description and reasoning can provide richer and better services for the users. Similarly it has been used by us to make use of its knowledge representation technique i.e. ontology [section 4.2] for extraction of features from 7-band image of a region. Currently the domain ontology engineers can be considered to be the only ones who really can commit to their ontology. As the creation of geospatial ontology is a critical task because it needs a lot of expertise and knowledge of deep concepts in a particular domain, so researcher **Eleni Tomai** in year 2005 have provided a tool [21] for creating ontology for different domains in geological world.

1.3 Problem Statement:

As here we are providing the usage of new technology in Satellite Image Classification. All existing techniques for classification are lacking in distributed environment due to there limitation. They have some limitations like more computation, incompatibility between different software's being used, time limitation, and more manual work is needed.

Here we are making use of Semantic web technology i.e Ontology in order to make the classification of the image on the web possible. Semantic web have various technologies in it. Out of them, that we have used here is RDF (Resource Description Framework) Schema for representation of the features in the image in the form of Geo-ontology. This technology is used because it has greater speed and accuracy in its computation. Through it, we can easily represent the features of an image on web in a web compatible format i.e. XML and use them when ever it is needed.

1.4 Scope of the Problem :

- ✓ To classify the satellite image in the form of representing water, vegetation, urban, rocky region with different colors.

- ✓ To increase the accuracy of the classification process as compared to other available process.
- ✓ To classify the image with minimum time and resources.
- ✓ To make possible the classification process on web.

1.5 Organization of Thesis :

The remainder of the thesis is organized as follows:

Section 2: The introduction of remote sensing concepts. It represents how the satellite image has been formed and how it is transformed into digital number representation. Some basic concepts and terms of the subjects are covered.

Chapter 3: Introduces the concept of Image Classification and its Procedure. Classification types and some of the traditional and existing image classification techniques are described in brief.

Chapter 4: This section provides description of Semantic web, what is its architecture and how is it better than current web. The different Technologies of Semantic Web. It describes the ontology design process along with some of its representation languages like OWL, RDF/XML.

Chapter 5: This section describes proposed framework of Image Classification in detail. It illustrates how Geo-Ontology is created, how training set has been generated. It also describes about Accuracy assessment and Error matrix for techniques. Some snap shots of different tools being used are also provided.

Chapter 6: This section describes the CASE STUDY taken from DRDO Labs. Section describes the Image Classification for ALWAR REGION by our framework. The training set values and the geo ontology produced by it according to our design process. Here we have shown that better results can be obtained by our framework as compared to other existing techniques. Accuracy of different classifiers has been compared.

Chapter 7: This section provides details of Conference in which this proposed work has been accepted.

Chapter 8: Conclusion and Future work

Chapter 9: References

Appendix A: Introduction to ERDAS IMAGINE

Appendix B: Introduction to Protégé Software

Appendix C: Introduction to JAVA API

- JENA Framework
- JAI API

Chapter 2

Introduction to fundamentals of Remote Sensing

2.1 Definition of Remote Sensing

The **Canada Center for Remote Sensing** has provided the definition of remote sensing in their Remote sensing tutorial as

"*Remote sensing is the science (and to some extent, art) of acquiring information about the Earth's surface without actually being in contact with it. This is done by sensing and recording reflected or emitted energy and processing, analyzing, and applying that information.*"

In much of remote sensing [1], **the process** involves an interaction between incident radiation and the targets of interest. This is exemplified in fig. 2.1 by the use of imaging systems where the following seven elements are involved. Note, however that remote sensing also involves the sensing of emitted energy and the use of non-imaging sensors.

- **Energy Source or Illumination (A)** – the first requirement for remote sensing is to have an energy source which illuminates or provides electromagnetic energy to the target of interest.

- **Radiation and the Atmosphere (B)** – as the energy travels from its source to the target, it will come in contact with and interact with the atmosphere it passes through. This interaction may take place a second time as the energy travels from the target to the sensor.

- **Interaction with the Target (H)** - once the energy makes its way to the target through the atmosphere, it interacts with the target depending on the properties of both the target and the radiation.

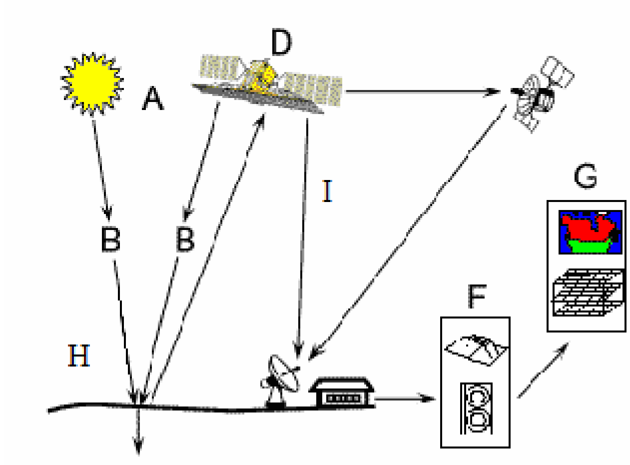


Fig 2.1: Capturing Image from Satellite

- **Recording of Energy by the Sensor (D)** - after the energy has been scattered by, or emitted from the target, we require a sensor (remote - not in contact with the target) to collect and record the electromagnetic radiation.

- **Transmission, Reception, and Processing (I)** - the energy recorded by the sensor has to be transmitted, often in electronic form, to a receiving and processing station where the data are processed into an image (hardcopy and/or digital).

- **Interpretation and Analysis (F)** - the processed image is interpreted, visually and/or digitally or electronically, to extract information about the target which was illuminated.

- **Application (G)** - the final element of the remote sensing process is achieved when we apply the information we have been able to extract from the imagery about the target in order to better understand it, reveal some new information, or assist in solving a particular problem.

2.2 Electromagnetic Radiation

As was noted in the previous section, the first requirement for remote sensing is to have an **energy source to illuminate the target** (unless the sensed energy is being emitted by the target). This energy is in the form of electromagnetic radiation. All electromagnetic radiation has fundamental properties and behaves in predictable ways according to the basics of wave theory.

Electromagnetic radiation consists of an electrical field (E) which varies in magnitude in a direction perpendicular to the direction in which the radiation is traveling, and a Magnetic field (M) oriented at right angles to the electrical field as depicted in fig 2.2. Both these fields travel at the speed of light (C). Two characteristics of electromagnetic radiation are particularly important for understanding remote sensing. These are the **wavelength and frequency.**

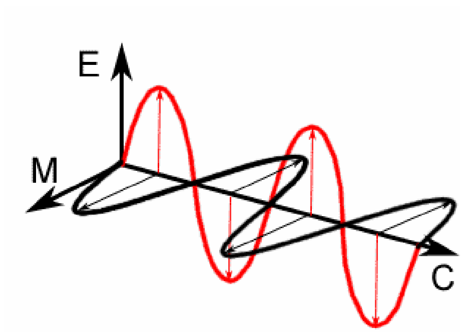
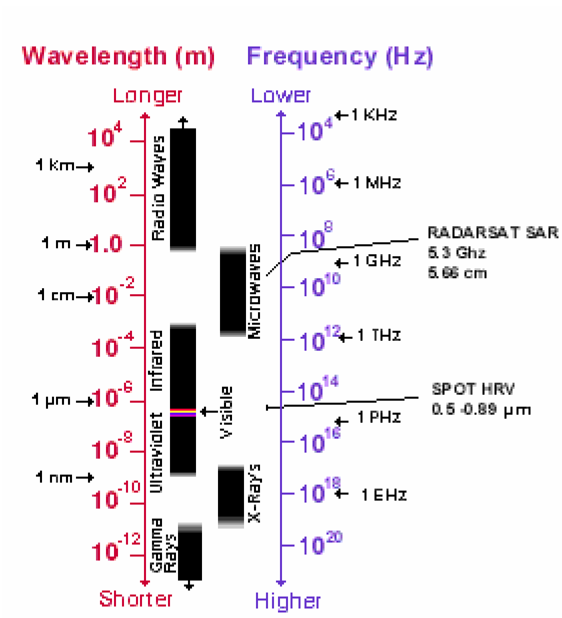


Fig 2.2: Electromagnetic Radiation

The wavelength is the length of one wave cycle, which can be measured as the distance between successive wave crests. It is measured in **lambda (λ)**. Frequency refers to the number of cycles of a wave passing a fixed point per unit of time. Frequency is normally measured in **hertz (Hz)**, equivalent to one cycle per second, and various multiples of hertz. The two are inversely related to each other. The shorter the wavelength, the higher the frequency. The longer the wavelength, the lower the frequency. Understanding the characteristics of electromagnetic radiation in terms of their wavelength and frequency is crucial to understanding the information to be extracted from remote sensing data.

2.3 The Electromagnetic Spectrum

The **electromagnetic spectrum** fig 2.3 ranges from the shorter wavelengths (including gamma and x-rays) to the longer wavelengths (including microwaves and broadcast radio waves). There are several regions of the electromagnetic spectrum which are useful for remote sensing. For most purposes, fig 2.4 the **ultraviolet or UV** portion of the spectrum has the shortest wavelengths which are practical for remote sensing. This radiation is just beyond the violet portion of the visible wavelengths, hence its name. Some Earth surface materials, primarily rocks and minerals, fluoresce or emit visible light when illuminated by UV radiation.



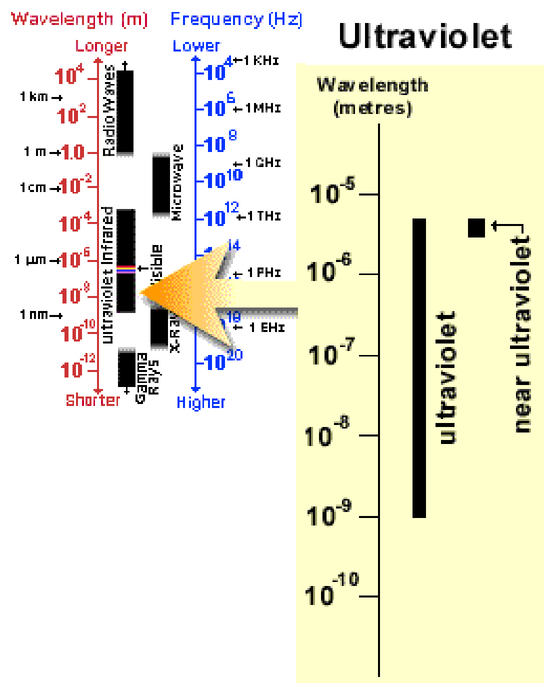


Fig 2.3: Electromagnetic Spectrum

Fig 2.4: Ultra Violet Portion of Spectrum

The light which our eyes - our "remote sensors" - can detect is part of the **visible spectrum** fig 2.5. It is important to recognize how small the visible portion is relative to the rest of the spectrum. There is a lot of radiation around us which is "invisible" to our eyes, but can be detected by other remote sensing instruments and used to our advantage. The visible wavelengths cover a range from approximately 0.4 to 0.7 μm. The longest visible wavelength is red and the shortest is violet. Common wavelengths of what we perceive as particular colors from the visible portion of the spectrum are listed below. It is important to note that this is the only portion of the spectrum we can associate with the concept of **colors**.

- **Violet:** 0.4 - 0.446 µm
- **Blue:** 0.446 - 0.500 µm
- **Green:** 0.500 - 0.578 µm
- **Yellow:** 0.578 - 0.592 µm
- **Orange:** 0.592 - 0.620 µm
- **Red:** 0.620 - 0.7 µm

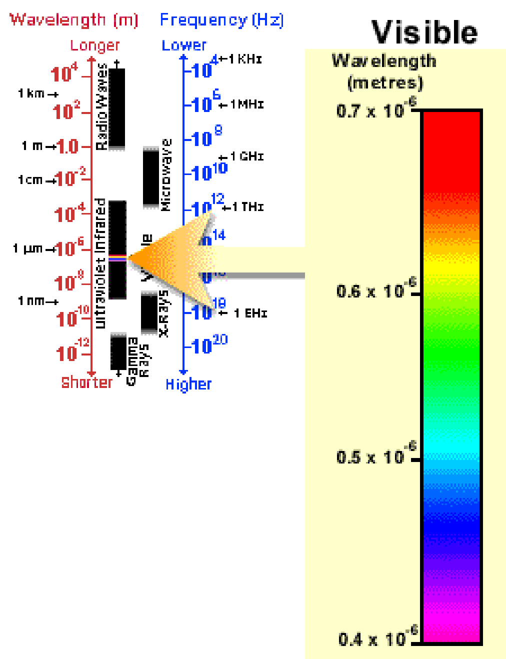


Fig 2.5: Visible Portion of Spectrum

2.4 Radiation - Target Interactions

Radiation that is not absorbed or scattered in the atmosphere can reach and interact with the Earth's surface. There are three (3) forms of interaction that can take place when energy strikes, or is **incident (I)** upon the surface. These are:

- **Absorption (A)**;

- **Transmission (T)**; and
- **Reflection (R)**.

The total incident energy will interact with the surface in one or more of these three ways. The proportions of each will depend on the wavelength of the energy and the material and condition of the feature.

Absorption (A) occurs when radiation (energy) is absorbed into the target while transmission (T) occurs when radiation passes through a target. Reflection (R) occurs when radiation "bounces" off the target and is redirected. In remote sensing, we are most interested in measuring the radiation reflected from targets. We refer to two types of reflection fig 2.6, which represent the two extreme ends of the way in which energy is reflected from a target: **specular reflection** and **diffuse reflection**.

When a surface is smooth we get **specular** or mirror-like reflection where all (or almost all) of the energy is directed away from the surface in a single direction. **Diffuse**

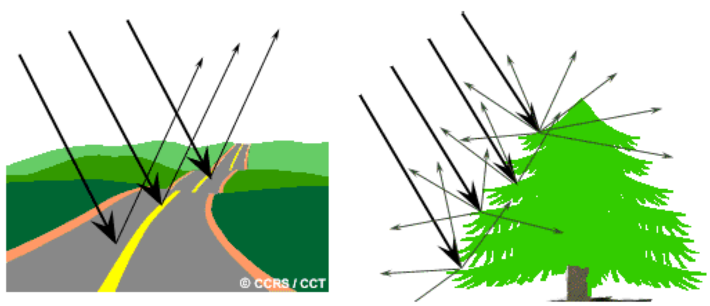


Fig 2.6: Specular and Diffuse Reflection

reflection occurs when the surface is rough and the energy is reflected almost uniformly in all directions. Most earth surface features lie somewhere between perfectly specular or perfectly diffuse reflectors. Whether a particular target reflects specularly or diffusely, or somewhere in between, depends on the surface roughness of the feature in comparison to the wavelength of the incoming radiation.

2.5 Passive vs. Active Sensing

So far, throughout this chapter, we have made various references to the sun as a source of energy or radiation. The sun provides a very convenient source of energy for remote sensing. The sun's energy is either **reflected**, as it is for visible wavelengths, or absorbed and then **reemitted**, as it is for thermal infrared wavelengths. Remote sensing systems which measure energy that is naturally available are called **passive sensors**. Passive sensors can only be used to detect energy when the naturally occurring energy is available. For all reflected energy, this can only take place during the time when the sun is illuminating the Earth. There is no reflected energy available from the sun at night. Energy that is naturally emitted (such as thermal infrared) can be detected day or night, as long as the amount of energy is large enough to be recorded.

Active sensors, on the other hand, provide their own energy source for illumination. The sensor emits radiation which is directed toward the target to be investigated. The radiation reflected from that target is detected and measured by the sensor. Advantages for active sensors include the ability to obtain measurements anytime, regardless of the time of day or season. Active sensors can be used for examining wavelengths that are not sufficiently provided by the sun, such as microwaves, or to better control the way a target is illuminated. However, active systems require the generation of a fairly large amount of energy to adequately illuminate targets. Some examples of active sensors are a laser fluorosensor and synthetic aperture radar (SAR).

2.6 Representation of Images

Electromagnetic energy may be detected either photographically or electronically. The photographic process uses chemical reactions on the surface of light- sensitive film to detect and record energy variations. It is important to distinguish between the terms **images** and **photographs** in remote sensing. An image refers to any pictorial representation, regardless of what wavelengths or remote sensing device has been used to detect and record the electromagnetic energy. A **photograph** refers specifically to images that have been detected as well as recorded on photographic film. Photos are normally recorded over the wavelength range from 0.3 μm to 0.9 μm - the visible and reflected infrared. Based on these definitions, we can say that all photographs are images, but not all images are photographs. Therefore, unless we are talking specifically about an image recorded photographically, we use the term image.

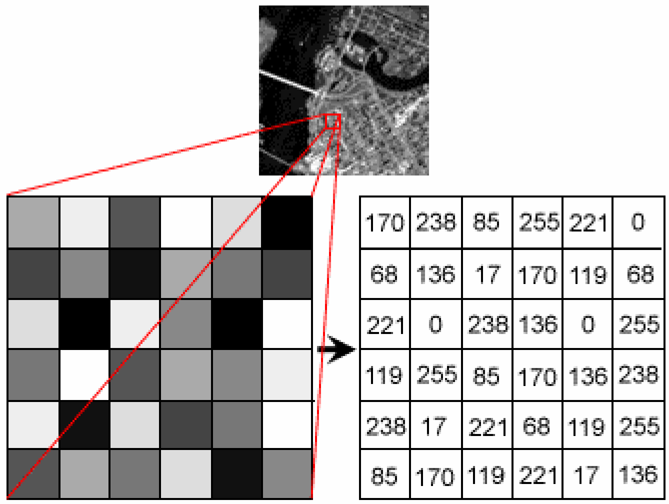


Fig 2.7: Digital Representation of an Image

A photograph could also be represented and displayed in a **digital** format by subdividing the image into small equal-sized and shaped areas, called picture elements or **pixels**, and representing the brightness of each area with a numeric value or **digital number** fig 2.7. Indeed, that is exactly what has been done to the photo on the top. The photograph was scanned and subdivided into pixels with each pixel assigned a digital number representing its relative brightness. The computer displays each digital value as different brightness levels. Sensors that record electromagnetic energy, electronically record the energy as an array of numbers in digital format right from the start. These two different ways of representing and displaying remote sensing data, either pictorially or digitally, are interchangeable as they convey the same information.

2.6.1 Digital Numbers (DN value)

It is a positive integer value, which represents the relative reflectance or emittance of an object in a digital image. For 8 bit images, the DN or digital number lies in the range of

0-255. In a most generalized way, a digital image is an array of numbers depicting spatial distribution of a certain field parameters (such as reflectivity of EM radiation, emissivity, temperature or some geophysical or topographical elevation. Digital image consists of discrete picture elements called pixels. Associated with each pixel is a number represented as DN (Digital Number) that depicts the average radiance of relatively small area within a scene. The range of DN values being normally 0 to 255. The size of this area effects the reproduction of details within the scene. As the pixel size is reduced more scene detail is preserved in digital representation.

Remote sensing images are recorded in digital forms and then processed by the computers to produce images for interpretation purposes. Images are available in two forms - photographic film form and digital form. Variations in the scene characteristics are represented as variations in brightness on photographic films. A particular part of scene reflecting more energy will appear bright while a different part of the same scene that reflecting less energy will appear black. Digital image consists of discrete picture elements called pixels. Associated with each pixel is a number represented as DN (Digital Number) that depicts the average radiance of relatively small area within a scene. The size of this area effects the reproduction of details within the scene. As the pixel size is reduced more scene detail is preserved in digital representation.

Chapter 3

Traditional Image Classification Techniques

3.1 Digital Image Processing

In today's world of advanced technology where most remote sensing data are recorded in digital format, virtually all image interpretation and analysis involves some element of digital processing. Digital image processing may involve numerous procedures including formatting and correcting of the data, digital enhancement to facilitate better visual interpretation, or even automated classification of targets and features entirely by computer. In order to process remote sensing imagery digitally, the data must be recorded and available in a digital form suitable for storage on a computer tape or disk. Obviously, the other requirement for digital image processing is a computer system, sometimes referred to as an **image analysis system** [1], with the appropriate hardware and software to process the data. Several commercially available software systems have been developed specifically for remote sensing image processing and analysis.

The common image processing functions available in image analysis systems can be categorized into the following four categories:

- **Preprocessing**
- **Image Enhancement**
- **Image Transformation**
- **Image Classification and Analysis**

- **Preprocessing:** functions involve those operations that are normally required prior to the main data analysis and extraction of information, and are generally grouped **as radiometric or geometric corrections**. Radiometric corrections include correcting the data for sensor irregularities and unwanted sensor or atmospheric noise, and converting the data so they accurately represent the reflected or emitted radiation measured by the sensor. Geometric corrections include correcting for geometric distortions due to sensor-Earth geometry variations, and conversion of the data to real world coordinates (e.g. latitude and longitude) on the Earth's surface.

- **Image Enhancement**: The objective of Image Enhancement is solely to **improve the appearance of the imagery** to assist in visual interpretation and analysis. Examples of enhancement functions include contrast stretching to increase the tonal distinction between various features in a scene, and **spatial filtering** to enhance (or suppress) specific spatial patterns in an image.

- **Image transformations** are operations similar in concept to those for image enhancement. However, unlike image enhancement operations which are normally applied only to a single channel of data at a time, image transformations usually involve combined processing of data from multiple spectral bands. Arithmetic operations (i.e. subtraction, addition, multiplication, division) are performed to combine and transform the original bands into "new" images which better display or highlight certain features in the scene. We will look at some of these operations including various methods of s**pectral or band** ratioing, and a procedure called **principal components analysis** which is used to more efficiently represent the information in multichannel imagery.

- **Image Classification:** Classification is the process of assigning classes to the pixels in a remotely sensed image. For example, in an agricultural map, each pixel could be assigned one of the classes' wheat, rice, barley, or fallow. The illustration in Fig 3.1(A) shows a picture captured by geographical satellites. The classification process yields the picture in Figure 3.1(B), in which water, highland regions, vegetation, and forest are classified and indicated with different colors. Our task is to identify, for each pixel in one image, which class should be assigned to that pixel. Since there are ways fewer classes than possible values for pixels we can consider classification as a process of simplification. If we assume that the image was classified correctly, we can easily do tasks such as area measurement and region extraction.

The classification technique is usually applied in the study of remote sensing to extract geographical information. Researchers may use that information to identify potential areas for agricultural development.

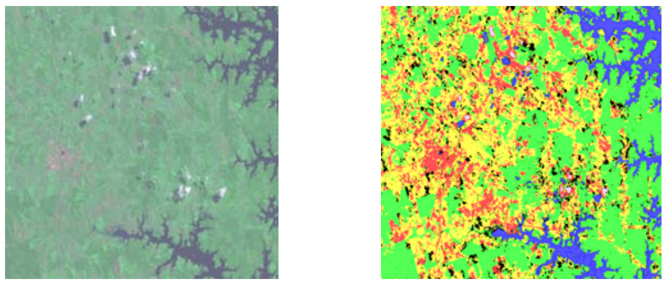


Fig 3.1: (A) Satellite Image (B) Classified Image

3.2 Image Classification Procedure

Image Classification procedure consists of three steps. These should be performed with accuracy so that the produced classified image will have better accuracy. The steps were defined as follows fig 3.2.

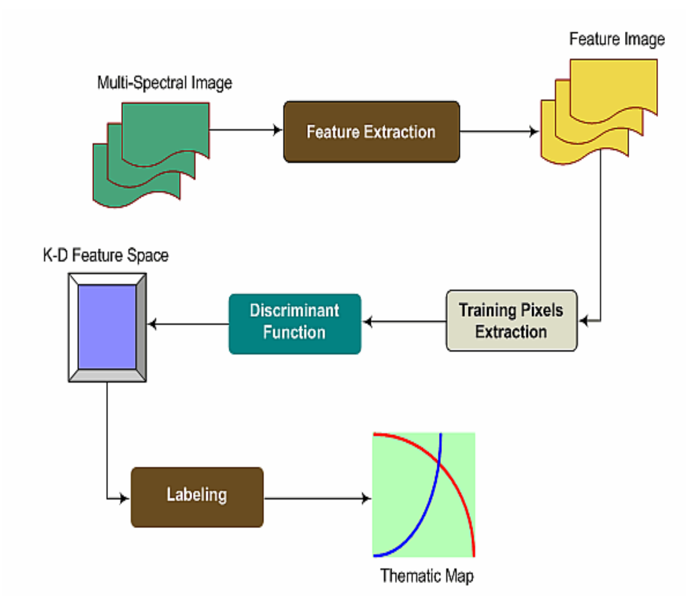


Fig 3.2: Image Classification Procedure [13]

3.2.1 Feature extraction.

This is an optional step of the classification process, which serves as a low-level pre-processing of the image to reduce its spectral or spatial dimensionality. It can be accomplished by using any type of spatial filter(s) or spectral transform(s) to reduce the data and/or enhance its multi-spectral features. In this stage, the multi-spectral image is transformed into a feature image.

3.2.2 Extraction of Training pixel.

In this step, pixels from the image are extracted to train the classifier to recognize patterns that help to differentiate the classes. Based on these patterns, the classifier creates discriminator functions to assign each pixel to a class in the feature space. The training of the pixels can be either Supervised or Unsupervised.

3.2.3 Classes labeling.

In this final stage of the image-classification process, the discriminated functions are used to label all the pixels in the entire feature image. If the training of the pixels was supervised, then a previous knowledge of the classes' spatial distribution allows the labeling of classes to be carried out upon the application of the discriminate functions to the feature space. If the training was unsupervised, the assignment of a label to each discriminated class is subject to the analyst's own labeling criteria.

3.3 Classification Techniques :

Several Classification techniques exist today. But two main types of classification are:

- **Supervised Classification**
- **Unsupervised Classification**

3.3.1 Supervised Classification :

In **supervised classification**, we must teach the classification algorithm how it can differentiate one class of other, usually by providing *samples* of pixels we know that should be assigned to a particular class fig 3.3. The algorithm will then use the information we provided to classify the other pixels on the image.

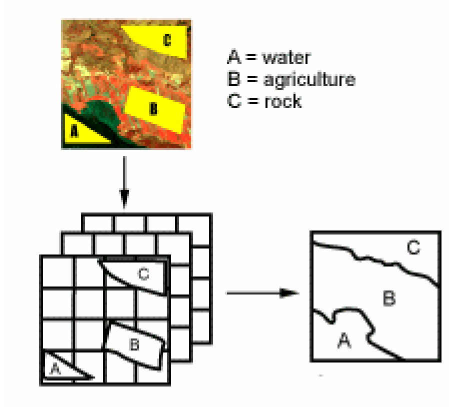


Fig 3.3: Supervised Classification

Some supervised classification techniques are :

- **Parellopiped Classifier**
- **Minimum distance to Mean Classifier**
- **Gaussian Maximum Likelihood**

3.3.1.1 Parallelopiped Classifier :

The Parallelopiped classifier is a very simple supervised classifier that uses intervals or bounded regions of pixels' values to determine whether a pixel belongs to a class or not. The intervals' bounding points are obtained from the values of the pixels of samples for the class. Since this classifier is supervised, there are two steps in its use: signature creation (training) and classification. Fig 3.4. depicts the general principle for it.

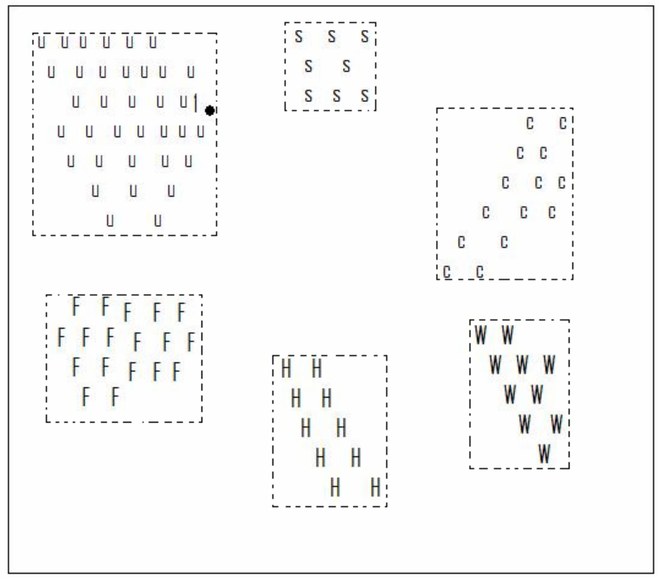


Fig 3.4: Parallelopiped Classifier

3.3.1.2 Minimum Distance to Mean Classifier

The minimum distance classifier [17] is a also very simple supervised classifier method that uses a central point (in feature space) to represent a class. The central point is calculated as the average of all pixels in all samples for that class. Classification is performed by calculating the distance (always in feature space) from a pixel with unknown class to the central points of each class and choosing the class which yields the smallest distance. Again, since it is a supervised classification algorithm, there are two steps in its use: signature creation and classification.

3.3.1.3 Gaussian Maximum Likelihood classifier

Maximum likelihood classifier [17] assumes that the statistics for each class in each band are normally distributed and calculates the probability that a given pixel belongs to a specific class. Unless a probability threshold is selected, all pixels are classified. Each pixel is assigned to the class that has the highest probability (i.e., the maximum likelihood).

3.3.2 Unsupervised Classification

In **unsupervised classification** we provide the algorithm with basic information on how many classes we expect to be present on the image, and the algorithm attempts to identify those classes fig 3.5. Some unsupervised algorithms are also known as *clustering* algorithms. Unsupervised classifiers do not utilize training data as the basis for classification, rather this family of classifiers involves algorithm that examine the unknown pixel in an image and aggregate them into a number of classes based on the natural grouping or clusters present in the image values. The basic premise is that values within a given cover type should be closed together in the measurement space whereas data in different classes should be comparatively well separated.

The classes that results from unsupervised classification are spectral classes because they are based solely on natural grouping in the image values, identity of the spectral classes will not be initially known. The analyst must compare the classified data within some form of referenced data to determine the identity and informational value of spectral classes. Some Unsupervised Techniques are: K-Means etc.

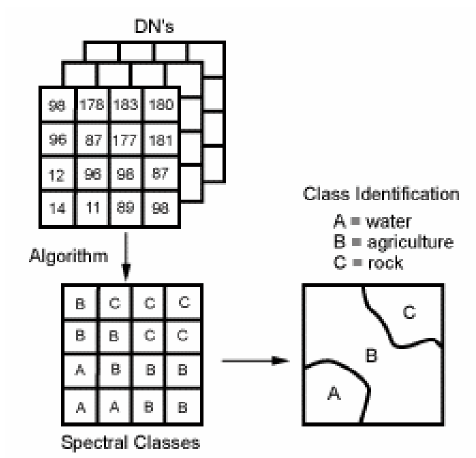


Fig 3.5: Unsupervised Classification

3.3.2.1 K-Means Classification

K-Means unsupervised classification calculates initial class means evenly distributed in the data space then iteratively clusters the pixels into the nearest class using a minimum distance technique. Each iteration recalculates class means and reclassifies pixels with respect to the new means. All pixels are classified to the nearest class unless a standard deviation or distance threshold is specified, in which case some pixels may be unclassified if they do not meet the selected criteria. This process continues until the number of pixels in each class changes by less than the selected pixel change threshold or the maximum number of iterations is reached.

3.4 Some Recent Classification Techniques.

As above we have described the traditional techniques of Image Classification. Here we are providing some recent techniques of it under research.

3.4.1 Fuzzy Classification

Fuzzy classification attempts to handle the mixed pixel problem by employing the fuzzy set concept, in which a given entity (a pixel) may have partial membership in more than 1 category. One approach to fuzzy classification is fuzzy clustering. This procedure is similar to “K-Means” unsupervised classification approach described earlier. The difference is that instead of having hard boundaries between classes in spectral measurement space, fuzzy regions are established. So instead of each unknown measurement vector being assigned solely to a single class, irrespective of how close that measurement may be a partition in measurement space, membership grade values are assigned that describes how close a pixel measurement is to the means of all classes.

Another approach to fuzzy classification is fuzzy supervised classification. This approach is similar to application of maximum likelihood classification, the difference being that fuzzy mean vector and covariance matrices are developed from statically waited training data. Instead of delineating training areas that are purely homogenous, a combination of pure and mixed training sites may be used. Known mixtures of various features types define the fuzzy training class weights. A classified pixel is then assigned a membership grade with respect to its membership in each information class. For example, a vegetation classification might include a pixel with grades of 0.68 for a class “forest”, 0.29 for “street” and 0.03 for “grass”.

3.4.2 Rough Set Classification

The rough set philosophy is founded on the assumption that with every object of the universe of discourse we associate some information (data, knowledge). For example, if objects are patients suffering from a certain disease, symptoms of the disease form information about patients. Objects characterized by the same information are indiscernible (similar) in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of rough set theory. Any set of all indiscernible (similar) objects is called an elementary set, and forms a basic granule (atom) of knowledge about the universe. Any union of some elementary sets is referred to as crisp (precise) set – otherwise the set is rough (imprecise, vague). Consequently each rough set has boundary-line cases, i.e., objects which cannot be with certainty classified neither as members of the set nor of its complement. Obviously crisp sets have no boundary-line elements at all. That means that boundary-line cases cannot be properly classified by employing the available knowledge. Thus, the assumption that objects can be" seen" only through the information available about them leads to the view that knowledge has granular structure. Due to the granularity of knowledge some objects of interest cannot be discerned and appear as the same (or similar). As a consequence vague concepts, in contrast to precise concepts, cannot be characterized in terms of information about their elements. The lower approximation consists of all objects that surely belong to the concept and the upper approximation contains all objects that possibly belong to the concept. Obviously, the difference between the upper and the lower approximation constitutes the boundary region of the vague concept. Approximations are two basic operations in rough set theory.

One of the main objectives of rough set data analysis is to reduce data size. Various notions such as *indiscernibility, rough set, reduct* are used to approximate inconsistent information and to exclude redundant data. They are used in analysis where there is complex variability in the spectral response pattern for individual cover types present. These conditions are quite common in such applications as vegetation mapping. Under these conditions, spectral variability within cover types normally comes about both from variations within cover types and from different site conditions.

Chapter 4
Introduction to Semantic Web and Web Ontology Language

4.1 Definition

The definition of *Semantic Web* according to **Tim Berners-Lee**, the inventor of *World Wide Web* is:

"*The extension of the current web in which information is given well-defined meaning, better enabling computers and humans to work in cooperation.*"

The majority of today's World Wide Web's content is designed for humans to read and understand, not for machines and computer programs to manipulate meaningfully. Computers can adeptly parse Web pages for layout and routine processing but, in general, machines have no reliable way to process the semantics. The *Semantic Web* will bring structure to the meaningful content of Web pages, where software agents roaming from page to page or from site to site can readily carry out automated sophisticated tasks for users.

4.2 Aims of Semantic Web

The aim of the Semantic Web is to allow much more advanced knowledge management systems:

- ✓ Knowledge will be organized in conceptual spaces according to its meaning.
- ✓ Automated tools will support maintenance by checking for inconsistencies and extracting new knowledge.
- ✓ Keyword-based search will be replaced by query answering: requested knowledge will be retrieved, extracted, and presented in a human friendly way.
- ✓ Query answering over several documents will be supported.

- ✓ Defining who may view certain parts of information (even parts of documents) will be possible.

4.3 From Current Web to the Semantic Web

The World Wide Web has changed the way people communicate with each other and the way business is conducted. It lies at the heart of a revolution that is currently transforming the developed world toward a knowledge economy and, more broadly speaking, to a knowledge society. This development has also changed the way we think of computers. Originally they were used for computing numerical calculations. Currently their predominant use is for information processing, typical applications being data bases, text processing, and games. At present there is a transition of focus towards the view of computers as entry points to the information highways.

Most of today's Web content [2] is suitable for human consumption. Even Web content that is generated automatically from databases is usually presented without the original structural information found in databases. Typical uses of the Web today involve people's seeking and making use of information, searching for and getting in touch with other people, reviewing catalogs of online stores and ordering products by filling out forms, and viewing adult material.

4.3.1 Semantic Web Solutions

The Semantic Web takes the solution further. It involves publishing in languages specifically designed for data: Resource Description Framework (RDF), Web Ontology Language (OWL), and Extensible Markup Language (XML). HTML describes documents and the links between them. RDF, OWL, and XML, by contrast, can describe arbitrary things such as people, meetings, or airplane parts. Tim Berners-Lee calls the resulting network of Linked Data the Giant Global Graph, in contrast to the HTML-based World Wide Web.

These technologies are combined in order to provide descriptions that supplement or replace the content of Web documents. Thus, content may manifest as descriptive data stored in Web-accessible databases, or as markup within documents (particularly, in Extensible HTML (XHTML) interspersed with XML, or, more often, purely in XML, with layout/rendering cues stored separately). The machine-readable descriptions enable content managers to add meaning to the content, i.e. to describe the structure of the knowledge we have about that content. In this way, a machine can process knowledge itself, instead of text, using processes similar to human deductive reasoning and inference, thereby obtaining more meaningful results and facilitating automated information gathering and research by computers.

An example of a tag that would be used in a non-semantic web page:

```
<item>cat</item>
```

Encoding similar information in a semantic web page might look like this:

```
<item rdf:about="http://dbpedia.org/resource/Cat">Cat</item>
```

4.4 A Layered Approach to Semantic Web

The development of the Semantic Web proceeds in steps, each step building a *layer* on top of another. In building one layer of the Semantic Web on top of another, two principles should be followed:

- ✓ **Downward compatibility.** Agents fully aware of a layer should also be able to interpret and use information written at lower levels. For example, agents aware of the semantics of OWL can take full advantage of information written in RDF and RDF Schema.

- ✓ **Upward partial understanding**. On the other hand, agents fully aware of a layer should take at least partial advantage of information at higher levels. For example, an agent aware only of the RDF and RDF Schema semantics an interpret knowledge written in OWL partly, by disregarding those elements that go beyond RDF and RDF Schema.

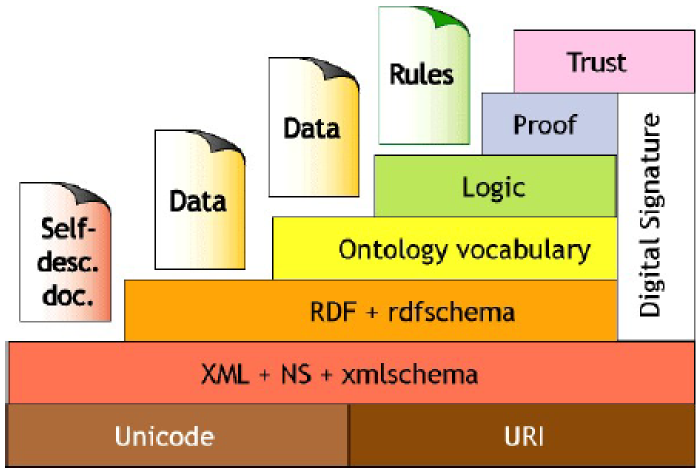


Fig 4.1: Layered Approach of Semantic Web

4.4.1 The Technologies

The common use of the term Semantic Web is to identify a set of technologies, tools and standards which form the basic building blocks of a system that could support the vision of a Web imbued with meaning. The Semantic Web has been developing a layered architecture, which is often represented using a fig 4.1 first proposed by Tim Berners-Lee, with many variations since. Figure 13 gives a typical representation of this diagram.

While necessarily a simplification which has to be used with some caution, it nevertheless gives reasonable conceptualizations of the various components of the Semantic Web. We describe briefly these layers.

- ✓ **Unicode and URI(Uniform Resource Locater)** : Unicode, the standard for computer character representation, and URIs, the standard for identifying and locating resources (such as pages on the Web), provide a baseline for representing characters used in most of the languages in the world, and for identifying resources.

✓ **XML (Extensible Markup Language):** A language that lets one write structured Web documents with a user-defined vocabulary fig 4.2. XML is particularly suitable for sending documents across the Web.

```
<class-def>
    <class name="plant"/>
    <subclass-of>
        <NOT><class name="animal"/></NOT>
    </subclass-of>
</class-def>
<class-def>
<class name="tree"/>
    <subclass-of>
        <class name="plant"/>
     </subclass-of>
</class-def>
<class-def>
     <class name="branch"/>
     <slot-constraint>
          <slot name="is-part-of"/>
          <has-value>
             <class name="tree"/>
          </has-value>
     </slot-constraint>
</class def>
```

Fig 4.2: Snap shot for XML code

✓ **RDF (Resource Description Framework)** is a basic data model, like the entity-relationship model, for writing simple statements about Web objects (resources) fig 4.3. The RDF data model does not rely on XML, but RDF has an XML-based syntax. Therefore, in figure it is located on top of the XML layer.

✓ **RDF Schema** provides modeling primitives for organizing Web objects into hierarchies. Key primitives are classes and properties, subclass and sub property relationships, and domain and range restrictions. RDF Schema is based on RDF. RDF Schema can be viewed as a primitive language for writing ontology's. But there is a need for more powerful *ontology languages* that expand RDF Schema and allow the representations of more complex relationships between Web objects.

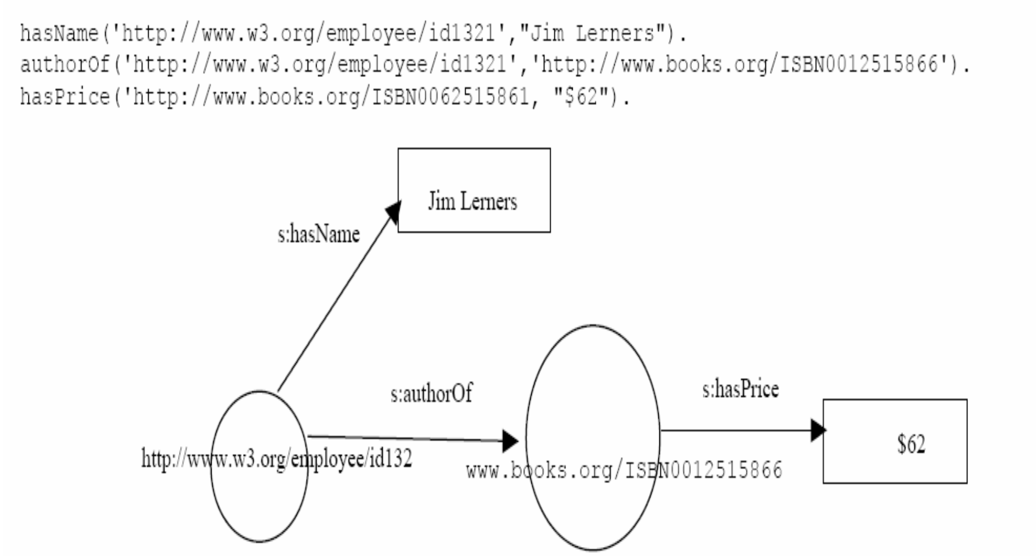


Fig 4.3: RDF Example

- ✓ **Ontology Vocabulary** explicit formal specifications of the terms in the domain and relations among them (Gruber 1993)—has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts.

- ✓ **Logic layer** is used to enhance the ontology language further and to allow the writing of application-specific declarative knowledge.

- ✓ **Proof layer** involves the actual deductive process as well as the representation of proofs in Web languages (from lower levels) and proof validation.

✓ **Trust layer** will emerge through the use of *digital signatures* and other kinds of knowledge, based on recommendations by trusted agents or on rating and certification agencies and consumer bodies. Sometimes "Web of Trust" is used to indicate that trust will be organized in the same distributed and chaotic way as the WWW itself. Being located at the top of the pyramid, trust is a high-level and crucial concept: the Web will only achieve its full potential when users have trust in its operations (security) and in the quality of information provided.

4.5 Ontology

4.5.1 Definition

Ontology [5] is defined as "explicit specification of conceptualization" or it can be a formal conceptualization of a domain that is shared and reused across domains, tasks and group of people. Ontology is a model of the world, represented as a tangled tree of linked concepts. Ontology is used to capture knowledge about some domain of interest. Ontology describes the concepts in the domain and also the relationships that hold between those concepts. Different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C).Basic structure of Ontology fig. 4.4 is formed by following components:

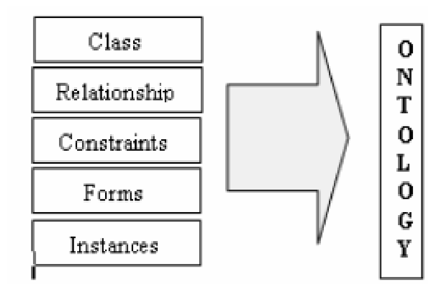


Fig. 4.4: Ontology Structure

- **Classes :** OWL classes are interpreted as sets that contain individuals. They are described using formal (mathematical) descriptions that state precisely the requirements for membership of the class fig 4.5. For example, the class Cat would contain all the individuals that are cats in our domain of interest. Classes may be organized into a super class-subclass hierarchy, which is also known as taxonomy. Subclasses specialize ('are subsumed by') their super classes. For example consider the classes Animal and Cat – Cat might be a subclass of Animal (so Animal is the super class of Cat). This says that, 'All cats are animals', 'All members of the class Cat are members of the class Animal', 'Being a Cat implies that you're an Animal', and 'Cat is subsumed by Animal'.

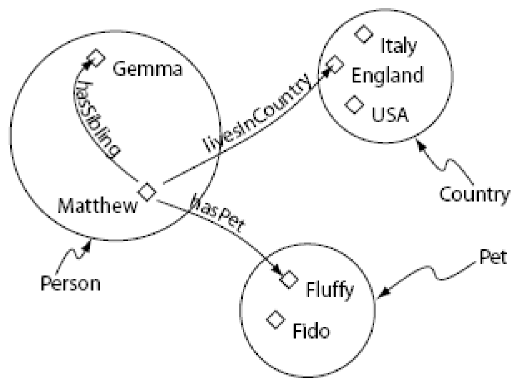


Fig. 4.5: Representation of Classes (Containing Individuals)

- **Instances/ Individuals**: Individuals, represent objects in the domain that we are interested in. An important difference between Prot´eg´e and OWL is that OWL does not use the Unique Name Assumption (UNA). This means that two different names could actually refer to the same individual. For example, "Queen Elizabeth", "The Queen" and "Elizabeth Windsor" might all refer to the same individual. In OWL, it must be explicitly stated that individuals are the same as each other, or different to each other — otherwise they might be the same as each other, or they might be different to each other. Fig. 4.6 shows a representation of some individuals in some domain.

Fig. 4.6: Representation of Individuals

- **Relationships/ Properties:** Properties are binary relations on individuals - i.e. properties link two individuals together. For example, the property hasSibling might link the individual Matthew to the individual Gemma, or the property hasChild might link the individual Peter to the individual Matthew. Properties can have inverses. For example, the inverse of hasOwner is isOwnedBy. Properties can be limited to having a single value –i.e. to being functional. They can also be either transitive or symmetric. Fig. 4.7 shows a representation of some properties linking some individuals together.

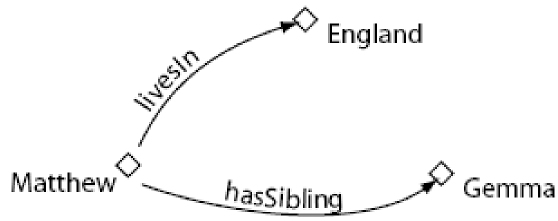


Fig. 4.7: Representation of Properties

- ***Forms*** are framework that is used to set the layout for the instances in ontology.

- ***Constraints*** are conditions that must be satisfied during the design. A property restriction is a special kind of class description. It defines an anonymous class, namely the set of class of all individuals that satisfy the restriction.In OWL properties are used to create restrictions. As the name may suggest, restrictions are used to restrict the individuals that belong to a class. Restrictions in OWL fall into three main categories:

  a. **Quantifier Restrictions**
  b. **Cardinality Restrictions**
  c. **hasValue Restrictions.**

We will initially use quantifier restrictions. These types of restrictions are composed of a quantifier, a property, and filler. The two quantifiers that may be used are:

• The existential quantifier ($\exists$), which can be read as at least one, or some.
• The universal quantifier ($\forall$), which can be read as only.

Using these attribute we have designed the ontology for Geology domain that can be used for the searching purpose and work as knowledge base for semantic web. It includes a collection of domain-specific concepts, and is a system description which includes class-subclass taxonomy, slots, forms, instances, relationships, constraints and performing query in knowledge base. The ontology design process is evolutionary in nature.

Ontology's are classified in four groups, according to their dependency on a specific domain or point of view (Guarino,1997):

i) **Top-level** ontology's describe very general concepts;
ii) **Upper level** ontology's describe the vocabulary related to a generic domain.
iii) **Domain** ontology's describe a domain or task.
iv) **Application** ontology's are at the lowest level in inheritance view combines, integrates, and extends all sub ontology's for the application.

4.5.2 Design Process of Ontology

The ontology has been designed by the process [31] depicted here. The various steps of process are shown in fig. 4.8

- **Expert Analysis/ Domain Analysis*:*** First step in ontology design process is to analysis the domain for which we are going to design ontology. For analysis we need an expert of the particular domain having the knowledge about the knowledge representation for that domain. The expert will cover the following main issues regarding ontology: Ontology scope and Knowledge source. In our study scope of our geo ontology is to classify a satellite image with maximum accuracy.

- **Tool and Languages/ Design Structure*:*** The ontology development tools such as Protégé [18], SWOOP and many others are freely available. Protégé is one of the best choices for a free software ontology development platform. Several ontology languages are available like RDF, RDFS, DAML+OIL, OWL [6]. OWL has three versions OWL lite, OWL DL, OWL Full. Each language have their own characteristics. We have made use of RDF/XML language for geo-ontology construction.

- **Ontology Design/ Creation*:*** Ontology design covers the design of framework for ontology by user, expert and designer to represent knowledge in efficient way.

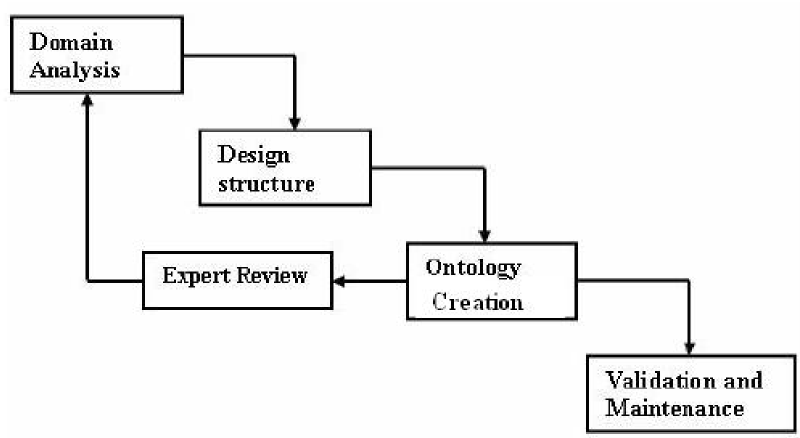


Fig. 4.8: Ontology Design Process

- **Validation and Maintenance**: Ontology is checked for validation. It is checked to determine if it is balanced or not.

4.5.3 Ontology Languages

Ontology can be expressed in various formats. Each format has its limitations.The various representation formats are RDF/XML, Notation 3, TURTLE, OWL. Out of which RDF/XML is used in this project for ontology creation as it has some more features as compared to OWL.

4.5.3.1 OWL Ontology

The **Web Ontology Language** (**OWL**) is a family of knowledge representation languages for authoring ontologies, and is endorsed by the World Wide Web Consortium[6]. This family of languages is based on two (largely, but not entirely, compatible) semantics: OWL DL and OWL Lite semantics are based on Description Logics, which have attractive and well-understood computational properties, while OWL Full uses a novel semantic model intended to provide compatibility with RDF Schema. OWL ontologies are most commonly serialized using RDF/XML syntax. OWL is considered one of the fundamental technologies underpinning the Semantic Web, and has attracted both academic and commercial interest.

Like Protégé OWL makes it possible to describe concepts but it also provides new facilities. It has a richer set of operators - e.g. and, or and negation. It is based on a different logical model which makes it possible for concepts to be defined as well as described. Complex concepts can therefore be built up in definitions out of simpler concepts. Furthermore, the logical model allows the use of a reasoner which can check whether or not all of the statements and definitions in the ontology are mutually consistent and can also recognize which concepts fit under which definitions. The reasoner can therefore help to maintain the hierarchy correctly. This is particularly useful when dealing with cases where classes can have more than one parent.

➢ **The species of OWL**

OWL ontology may be categorized into three species or sub-languages: OWL-Lite, OWL-DL and OWL- Full. A defining feature of each sub-language is its expressiveness. OWL-Lite is the least expressive sub-language. OWL-Full is the most expressive sub-language. The expressiveness of OWL-DL falls between that of OWL-Lite and OWL-Full. OWL-DL may be considered as an extension of OWL-Lite and OWL-Full an extension of OWL-DL.

- ***OWL Lite*** was originally intended to support those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It was hoped that it would be simpler to provide tool support for OWL Lite than its more expressive relatives, allowing quick migration path for systems utilizing thesauri and other taxonomies.

- ***OWL DL*** OWL-DL is much more expressive than OWL-Lite and is based on Description Logics (hence the suffix DL). Description Logics are a decidable fragment of First Order Logic2 and are therefore amenable to automated reasoning. It is therefore possible to automatically compute the classification hierarchy3 and check for inconsistencies in an ontology that conforms to OWL-DL.OWL DL was designed to provide the maximum expressiveness possible while retaining computational completeness (all conclusions are guaranteed to be computed), decidability (all computations will finish in finite time), and the availability of practical reasoning algorithms.

- ***OWL Full*** is based on a different semantics from OWL Lite or OWL DL, and was designed to preserve some compatibility with RDF Schema. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right; this is not permitted in OWL DL.

Each of these sublanguages is a syntactic extension of its simpler predecessor. The following set of relations hold. Their inverses do not.

- Every legal OWL Lite ontology is a legal OWL DL ontology.
- Every legal OWL DL ontology is a legal OWL Full ontology.
- Every valid OWL Lite conclusion is a valid OWL DL conclusion.
- Every valid OWL DL conclusion is a valid OWL Full conclusion.

4.5.3.2 RDF/XML:

The **Resource Description Framework (RDF)** is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources; using a variety of syntax formats.

Basically speaking, the RDF[4] data model is not different from classic conceptual modeling approaches such as Entity-Relationship or Class diagrams, as it is based upon the idea of making statements about resources, in particular, Web resources, in the form of subject-predicate-object expressions. These expressions are known as *triples* in RDF terminology fig 4.3. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object.

For example, one way to represent the notion "The sky has the color blue" in RDF is as the triple: a subject denoting "the sky", a predicate denoting "has the color", and an object denoting "blue". RDF is an abstract model with several serialization formats (i.e., file formats), and so the particular way in which a resource or triple is encoded varies from format to format.

This mechanism for describing resources is a major component in what is proposed by the W3C's Semantic Web activity: an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the Web, in turn enabling users to deal with the information with greater efficiency and certainty. RDF's simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity.

A collection of RDF statements intrinsically represents a labeled, directed multi-graph. As such, an RDF-based data model is more naturally suited to certain kinds of knowledge representation than the relational model and other ontological models traditionally used in computing today. However, in practice, RDF data is often persisted in relational database or native representations also called Triple stores, or Quad stores if context (i.e. the named graph) is also persisted for each RDF triple.As RDFS and OWL demonstrate, additional ontology languages can be built upon RDF.

- **RDF Basic Ideas**

The fundamental concepts [2] of RDF are resources, properties and statements.

- **Resources**

We can think of a resource as an object, a “thing” we want to talk about. Resources may be authors, books, publishers, places, people, hotels, rooms, search queries, and so on. Every resource has a URI, a Universal Resource Identifier. A URI can be a URL (Unified Resource Locator, or Web address) or some other kind of unique identifier; note that an identifier does not necessarily enable *access* to a resource. URI schemes have been defined not only for web-locations but also for such diverse objects as telephone numbers, ISBN numbers and geographic locations. There has been a long discussion about the nature of URIs, even touching philosophical questions (for example, what is an appropriate unique identifier for a person?), but we will not go into into detail here. In general, we assume that a URI is the identifier of a Web resource.

- **Properties**

Properties are a special kind of resources; they describe relations between resources, for example “written by”, “age”, “title”, and so on. Properties in RDF are also identified by URIs (and in practice by URLs). This idea of using URIs to identify “things” and the relations between is quite important. This choice gives us in one stroke a global, worldwide, unique naming scheme. The use of such a scheme greatly reduces the homonym problem that has plagued distributed data representation until now.

- **Statements**

Statements assert the properties of resources. A statement is an object attribute-value triple, consisting of a resource, a property, and a value. Values can either be resources or *literals*. Literals are atomic values (strings), the structure of which we do not discuss further. An example of a statement is

David Billington is the owner of the Web page http://www.cit.gu.edu.au/~db.

The simplest way of interpreting this statement is to use the definition and consider the triple

(“David Billington”, http://www.mydomain.org/site-owner, http://www.cit.gu.edu.au/~ db).

We can think of this triple (*x, P, y*) as a logical formula *P*(*x, y*), where the binary predicate *P* relates the object *x* to the object *y*. In fact, *RDF offers only binary predicates (properties)* fig 4.8. Note that the property “site-owner” and one of the two objects are identified by URLs, whereas the other object is simply identified by a string.

Fig. 4.9: Graph Representation of Triple

A second view is graph-based. fig. 4.9 shows the graph corresponding to the preceding statement. It is a directed graph with labeled nodes and arcs; the arcs are directed from the resource (the *subject* of the statement) to the value (the *object* of the statement). This kind of graph is known in the Artificial Intelligence community as a *semantic net* .

As we already said, the value of a statement may be a resource. Therefore, it may be linked to other resources. Consider the following triples:

(http://www.cit.gu.edu.au/ ~db, http://www.mydomain.org/siteowner,“David Billington”)

(“David Billington”, http://www.mydomain.org/phone, “3875507”)

(“David Billington”, http://www.mydomain.org/uses,http://www.cit.gu.edu.au/ ~ arock/defeasible/Defeasible.cgi)

(“www.cit.gu.edu.au/ ~arock/defeasible/Defeasible.cgi”,http://www.mydomain.org/site-owner, “Andrew Rock”)

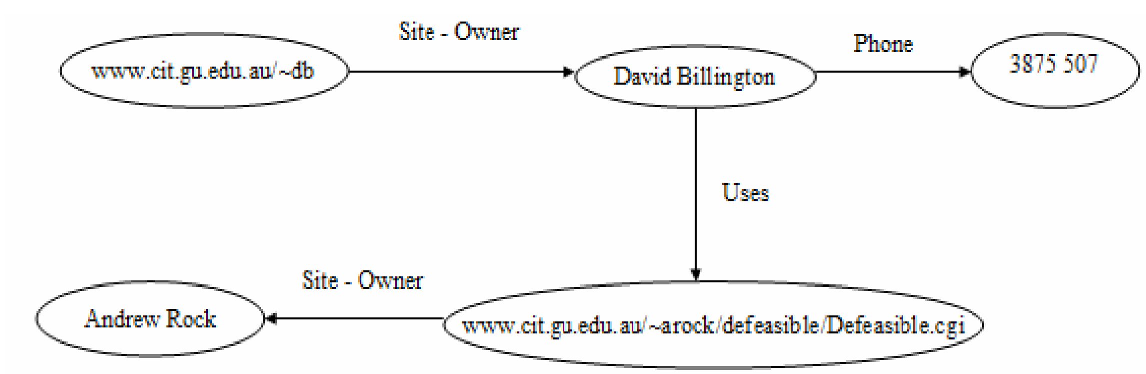


Fig 4.10: Semantic Net

The graphic representation is found in figure 4.10. Graphs are a powerful tool for human understanding. But the Semantic Web vision requires machine-accessible and machine-process able representations.

- **RDF/XML Syntax**

An RDF document [2] consists of an rdf:RDF element, the content of which is a number of descriptions. For example, consider the domain of university courses and lecturers at Griffith University in the year 2001.

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd="http://www.w3.org/2001/XLMSchema#"
xmlns:uni="http://www.mydomain.org/uni-ns#">

<rdf:Description rdf:about="949352">
<uni:name>Grigoris Antoniou</uni:name>
<uni:title>Professor</uni:title>
</rdf:Description>

<rdf:Description rdf:about="949318">
<uni:name>David Billington</uni:name>
<uni:title>Associate Professor</uni:title>
<uni:age rdf:datatype="&xsd;integer">27</uni:age>
</rdf:Description>

<rdf:Description rdf:about="CIT1111">
<uni:courseName>Discrete Mathematics</uni:courseName>
<uni:isTaughtBy>David Billington</uni:isTaughtBy>
</rdf:Description>

<rdf: Description rdf:about="CIT1112">
<uni:courseName>Concrète Mathematics</uni:courseName>
<uni:isTaughtBy>Grigoris Antoniou</uni:isTaughtBy>
</rdf: Description>
</rdf:RDF>
```

- **First,** the namespace mechanism of XML is used, but in an expanded way. In XML namespaces are only used for disambiguation purposes. In RDF external namespaces are expected to be RDF documents defining resources, which are then used in the importing RDF document. This mechanism allows the reuse of resources by other people who may decide to insert additional features into these resources. The result is the emergence of large, distributed collections of knowledge.

- **Second,** the rdf:about attribute of the element rdf:Description is strictly speaking equivalent meaning to that of an ID attribute, but it is often used to suggest that the object about which a statement is made has already been "defined" elsewhere. Formally speaking, a set of RDF statements together simply forms a large graph, relating things to other things through properties, and there is no such thing as "defining" an object in one place and referring to it elsewhere. Nevertheless, in the serialized XML syntax, it is sometimes useful (if only for human readability) to suggest that one location in the XML serialization is the "defining" location, while other locations state "additional" properties about an object that has been "defined" elsewhere. In fact the preceding example is slightly misleading. If we wanted to be absolutely correct, we should replace all occurrences of course and staff ID's, such as 949352 and CIT3112, by references to the external namespace, for example

```
<rdf:Description
rdf:about="http://www.mydomain.org/uni-ns/#CIT3112">
```

We have refrained from doing so to improve readability of our initial example because we are primarily interested here in the ideas of RDF. However, readers should be aware that this would be the precise way of writing a correct RDF document. The content of rdf:Description elements are called *property elements*. For example, in the description

```
<rdf:Description rdf:about="CIT3116">
<uni:courseName>Knowledge Representation</uni:courseName>
<uni:isTaughtBy>Grigoris Antoniou</uni:isTaughtBy>
</rdf:Description>
```

the two elements uni:courseName and uni:isTaughtBy both define property-value pairs for CIT3116. The preceding description corresponds to two RDF statements.

- **Third,** the attribute rdf:datatype="&xsd;integer" is used to indicate the data type of the value of the age property. Even though the age property has been defined to have "&xsd;integer" as its range, it is still required to indicate the type of the value of this property each time it is used.

- **The rdf : resource Attribute**

The preceding example was not satisfactory in one respect: the relationships between courses and lecturers were not formally defined but existed implicitly through the use of the same name. To a machine, the use of the same name may just be a coincidence: for example, the David Billington who teaches CIT3112 may not be the same person as the person with ID 949318 who happens to be called David Billington. What we need instead is a formal specification of the fact that, for example, the teacher of CIT1111 is the staff member with number 949318, whose name is David Billington. We can achieve this effect using an rdf:resource attribute:

```
<rdf:Description rdf:about="CIT1111">
<uni:courseName>Discrete Mathematics</uni:courseName>
<uni:isTaughtBy rdf:resource="949318"/>
</rdf:Description>
<rdf:Description rdf:about="949318">
<uni:name>David Billington</uni:name>
<uni:title>Associate Professor</uni:title>
</rdf:Description>
```

We note that in case we had *defined* the resource of the staff member with ID number 939318 in the RDF document using the ID attribute instead of the about attribute, we would have had to use a # symbol in front of 949318 in the value of rdf:resource:

```
<rdf:Description rdf:about="CIT1111">
<uni:courseName>Discrete Mathematics</uni:courseName>
<uni:isTaughtBy rdf:resource="#949318"/>
</rdf:Description>
<rdf:Description rdf:ID="#949318">
<uni:name>David Billington</uni:name>
<uni:title>Associate Professor</uni:title>
</rdf:Description>
```

The same is true for externally defined resources: For example, we refer to the externally defined resource CIT1111 by using http://www.mydomain.org/uni-ns/#CIT1111 as the value of rdf:about, where www.mydomain.org/uni-ns/ is the URI where the definition of CIT1111 is found. In other words, a description with an ID defines a fragment URI, which can be used to reference the defined description.

Chapter 5

Proposed Framework Description

5.1 Introduction

As here we are proposing a framework to classify a Satellite Image. We have studied that image classification is a procedure of detecting different classes in an image. These classes are formed by different land cover like water, vegetation, urban, barren and rocky etc. in an image. Here we are using a supervised technique to classify an image. We have seen several traditional techniques exist for classification but they are lacking in interoperability and reusability in the web. Some of their drawbacks are as follows:

- Currently available techniques are time consuming and require more manual power.
- Minimum Distance to mean and Maximum Likelihood are not having accuracy rate to the acceptable value.
- All the techniques available are having some floating point errors in it.
- All the rules reduction policy used will reduce the training set but also decreases the accuracy of the system.
- The available techniques are not sharable, interoperable and distributed in open environment like web.
- All the techniques used till now only made use of DN value in their classification. They don't use NDVI, SBI etc. values from the satellite image for classification.

We are here using a more mature and impressive technology of current web i.e. Semantic Web for classification of Satellite Images. As semantic web is more powerful for basic search and improves interoperability in a distributed environment. So we have taken its emerging Technology i.e Ontology for following purpose.

- Ontology provides such a structure to the description which can be easily distributed on web.
- This description of concept is easily compatible with different systems.
- With the use of it data can be easily shared and accessed on web.

- With the use of it, one description of an satellite image can be easily extended to classify another similar land cover image as well.

So, we have provided a description of the satellite image to the system through Geo-Ontology [19] and used the same for classification of the image.

Here we have made use of several different tools for the implementation of the system. As initially we have used ERDAS software [15] for the generation of training set corresponding to the satellite image. This training set has been used for the construction of geo-ontology through Protégé software. Several architecture was also proposed for the construction of geo ontology [20, 21]. But the ontology constructed through Protégé is more balanced and have less ambiguity in it. Then these XML statements and 7 band satellite images are called upon in a single platform for the implementation of a system. We have used here java as the platform. The JENA framework [16] have been used for the traversing Ontology and JAI framework [14] have been used for reading of Satellite images which are in TIFF format [22]. Finally the classified image has been produced by the system in JPEG format with distinct colors for water, vegetation, urban, barren and rocky class.

5.2 System Framework

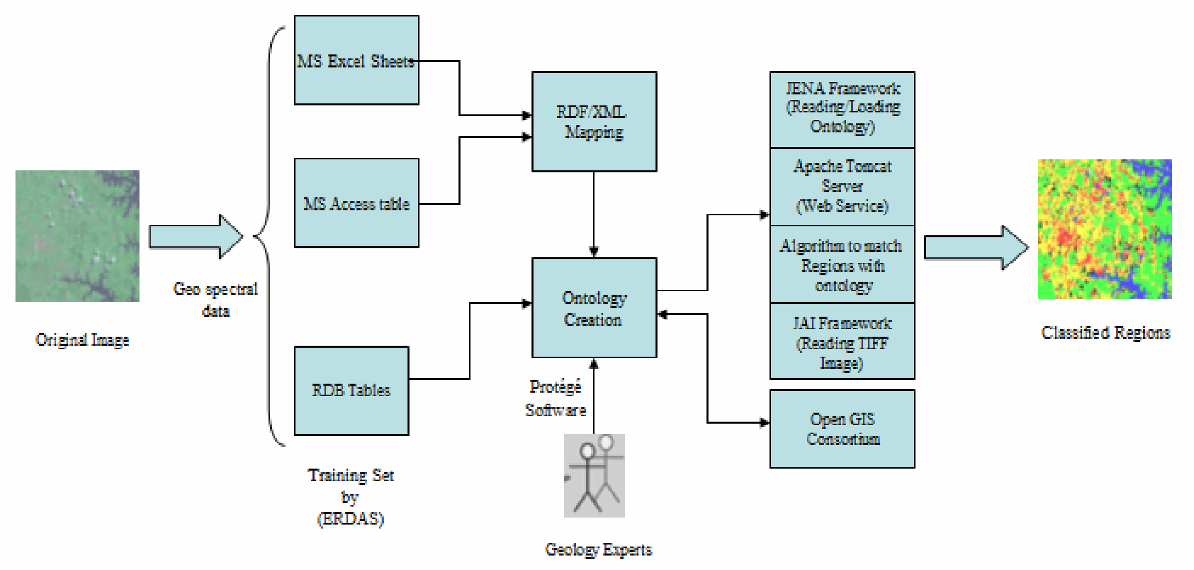


Fig 5.1: Framework for Semantic Classifier

Framework for the system has been built in such a way so that if original image is available in Excel sheet format rather than TIFF format, then also we can construct ontology for it by the use of Excel to RDF/XML mapper. If the data for the image is available in tabular format of a database, than also we can have build ontology for it by connecting database with protégé. Following components comprising the framework for the Semantic Classifier:

As we can see from the fig 5.1, the framework comprises of three basic steps and these are:

- Training Set Generation
- Geo-Ontology Construction
- Matching Region Detection/ Classification

5.2.1 Training Set Generation

Initially we have different bands satellite image of any region. The image comprises of several land covers like rocky, water, vegetation, open land, barren etc. In a satellite image the regions are not displayed appropriately. So, we need some system expert, to differentiate between these land covers. ERDAS software is used for expert knowledge.
In order to classify the image by different techniques and check their accuracy assessment each classified image has been compared with the image generated by expert system. Training set has been generated by this software manually fig 5.2. The individual classes pixel has been picked from an image and their proper class has been assigned.
In ERDAS first we have to open the 7 band image in View. Then we will select type of Classification i.e. Supervised Classification from classifier menu. We will open the tool box for Raster Image, it is shown in left bottom corner. Then we will select appropriate tool for the selection of the pixel/ pixels and merge all similar pixels into one class.

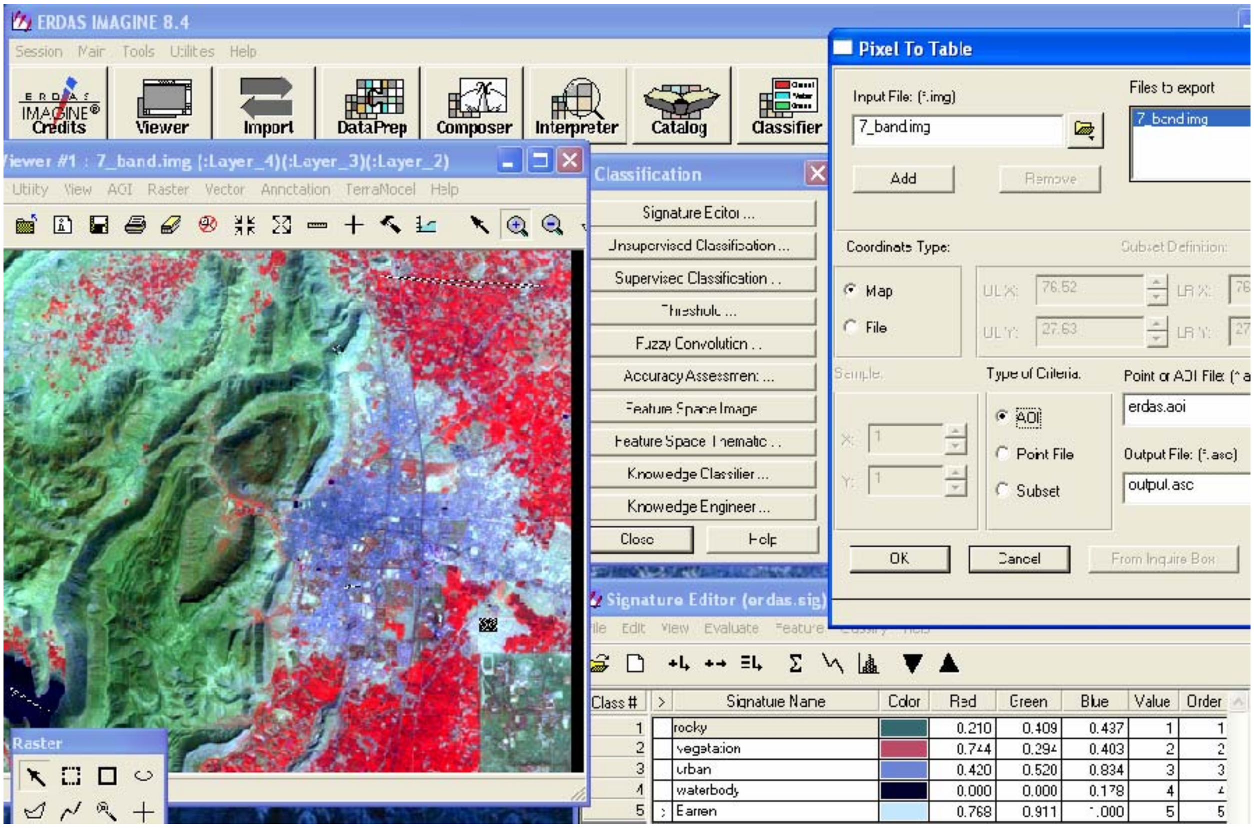


Fig 5.2: Snap shot for training set generation by ERDAS IMAGINE

Class creation has been done in Signature Editor where we merge the selected pixel of one class into one value and correspondingly one ASCII file has been created for it. In order to see the selected pixels we have to save them as a AOI layer as shown in fig 5.3.

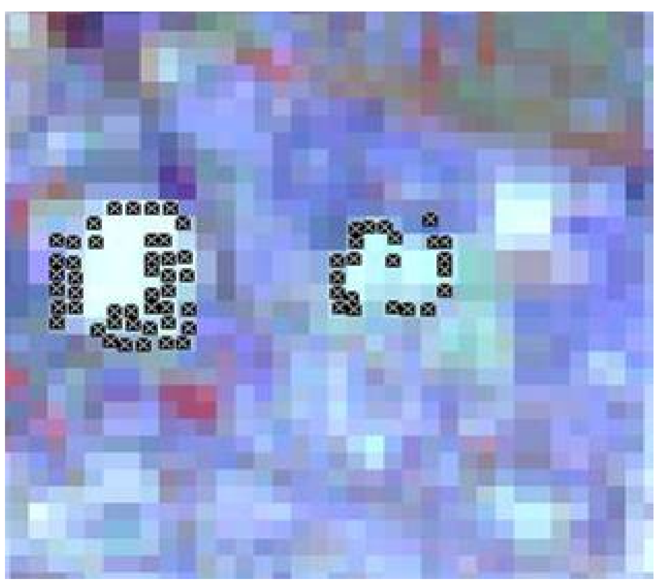


Fig 5.3: Showing the pixels selected for URBAN class

The training set generated for water, vegetation, urban, barren and rocky classes are shown as under:

Table 1: Training Set for ALWAR REGION IMAGE

RED	GREEN	NIR	MIR	RS1	RS2	DEM	DECISION
115	91	182	126	20	15	30	Barren
111	90	173	131	17	34	15	Barren
121	91	182	118	26	17	40	Barren
125	98	188	128	25	21	27	Barren
62	49	135	91	44	40	94	Rocky
84	64	160	102	20	25	165	Rocky
52	45	129	85	15	29	107	Rocky
91	69	171	106	10	46	123	Rocky
128	106	184	142	22	35	15	Urban
128	120	155	118	18	36	15	Urban
117	109	157	122	19	21	15	Urban
123	109	166	137	10	13	15	Urban
21	35	65	192	24	33	9	Vegetation
23	37	73	192	18	33	14	Vegetation
27	37	82	170	30	30	13	Vegetation
21	35	69	252	20	22	11	Vegetation
25	30	25	10	180	92	15	Waterbody
25	30	25	10	157	60	15	Waterbody
25	30	21	20	8	2	30	Waterbody
21	27	20	20	2	5	30	Waterbody

Here as we have taken the 7 band image i.e. Red band , Green Band, NIR (Near Infrared), MIR (Micro Infrared), DEM(Digital Elevation Model), RS1(Radar Sat 1) and RS2 (Radar sat 2). In the above the last column showed the decision of the expert system for the different band values of the pixel. Through this excel sheet geo-Ontology is created in which we generate a description of the training set. Along with it some constraints are assigned to these properties in the form of rule.

5.2.2 Geo-Ontology Construction

As we have studied earlier about ontology, which is simply the description of any concept in any domain. Similarly, Geo-Ontology is Ontology for Geology Domain. **Geo ontology is a description of geographical information in digital satellite imag**e. Combining geology knowledge with semantic concept, we can derive a relation for geo ontology comprising of the following:

Geo Ontology: {C, R, P, V, I, X}

C: Concepts	**R**: Relations among concepts
P: Properties of concepts.	**V:** Value of a property.
X : Axiomes/Constraints on concepts.	**I**: Instances of Concept.

The geo-ontology concept [24] belongs to the fundamental geographic information. The domain ontology of a concept has been created with the consult of geology experts. We have created the geographic ontology (geo-ontology) [23, 25] which provides a model of the terminology and structure of the geographic space.

5.2.2.1 Design Process of Geo- Ontology Construction:

Using this technology of Semantic web the ontology for geographic remote sensing classification domain has been built. We have used the process defined in section 4 for reaching this ontology.

We have made use of Protégé OWL plug in, for implementing the guidelines for building Geo- ontology (Fig 5.4 depicts the visualization of the geo-ontology). We have explained the steps of Geo-ontology construction:

Step 1: As, firstly with the help of experts we have determined the scope of the ontology i.e. Image classification. Then the classes have been constructed according to the application domain like barren, urban, water, vegetation and rocky by different bands of image Red, Green, NIR(Near Infrared), MIR, RS1(RadarSat 1), RS2, DEM(Digital Elevation Model). Several class and property relationships have been established to define the spectral property of the image in each individual class.

Each of the individuals being assigned to these classes must have a value within the range from each of the specified band.

Step 2: Here we will create our own knowledge base by coding fundamental geographic information like spatial and spectral concerns and various cartographic processes in RDF/XML[4] format. The geographic ontology (geo-ontology) provides a model of the terminology and structure of the geographic space.

The developed geo-ontology are OGC complaint as they are following Geo-Owl and Geo Concepts. The geo ontology has been constructed by the use of Protégé in RDF/XML format.

Step 3: As in figure 5.5 (A) and (B) we can see the property hasRedbvalue has been assigned to the red band as the value of barren region in Red band, and is defined with the range of maximum and minimum DN value. Similarly each region has its range in Red, Green, NIR, MIR, RS1, RS2, and DEM band. Finally there are some of the restrictions which have been assigned to Barren, Rocky, Urban, and Vegetation and Water class as shown in Table 2. All elements of geo-ontology are conceptualized (i.e. Bands, Regions, pixels) as classes.

- In the Bands class we have subclass of different bands in an image. In our case of Alwar Region Image we have 7 band image so we have seven subclasses of Bands class as: Red_Band, Green_Band, RS1_Band, RS2_Band, NIR_BAND, MIR_BAND, DEM_BAND
- In the Regions class we have 5 subclasses as Barren, Rocky, Urban, Vegetation, as Water.
- In each band subclass we have added some properties to it like Red_Band has properties hasRedbvalue which says that the DN value of the pixel in barren region in Red band image. Similarly it has hasReduvalue, hasRedvegvalue, hasRedwvalue, hasRedrvalue as shown in fig 5.5
- Each property assigned to a class has constraints applied like any pixel will fall in DEM_Band class, if it satisfies the following condition:

Table 2: Constraints on DEM_Band class

Property Name	**Cardinality**	**Type**	**Other Facets**
hasDEMbvalue	Single	Integer	minimum=15, maximum=164
hasDEMrvalue	Single	Integer	minimum=45, maximum=252
hasDEMuvalue	Single	Integer	minimum=0, maximum=15
hasDEMvegvalue	Single	Integer	minimum=0, maximum=110
hasDEMwvalue	Single	Integer	minimum=15, maximum=97

Some of the snap shots of the tool for class/ subclass creation, properties declaration and Constraints are shown in fig 5.6, 5.7, 5.8 correspondingly.

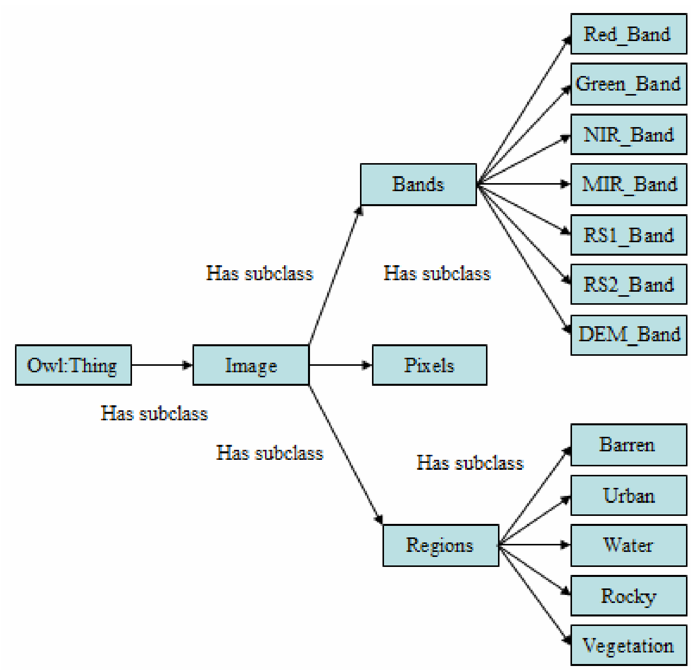


Fig 5.4: Geo- Ontology for a Satellite Image

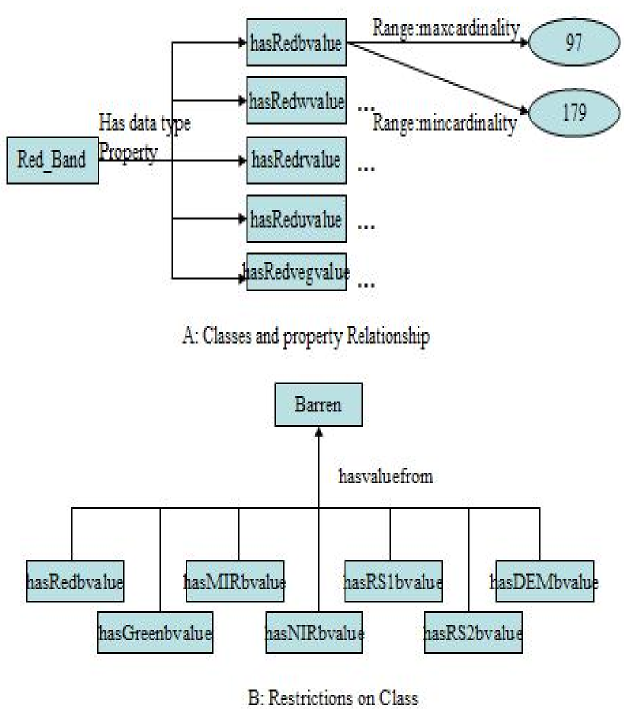


Fig 5.5: (A) Classes and Property Relationship (B) Restriction on Classes

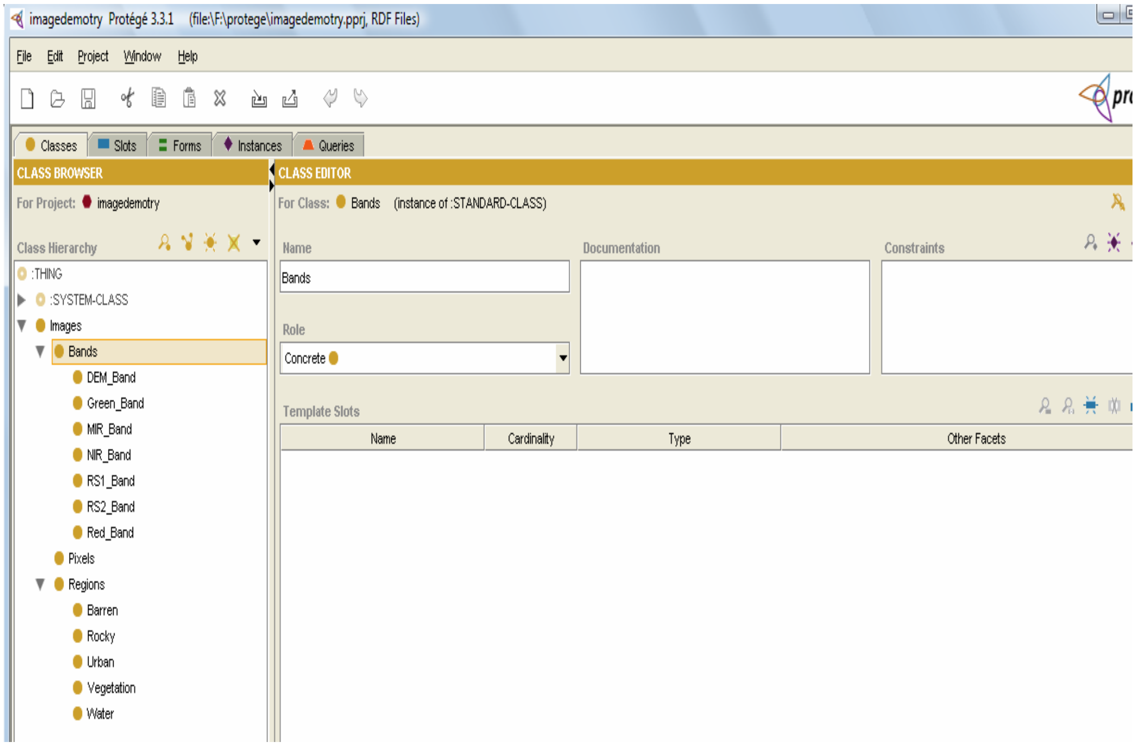


Fig 5.6: Class/ Sub class Editor window of Protégé

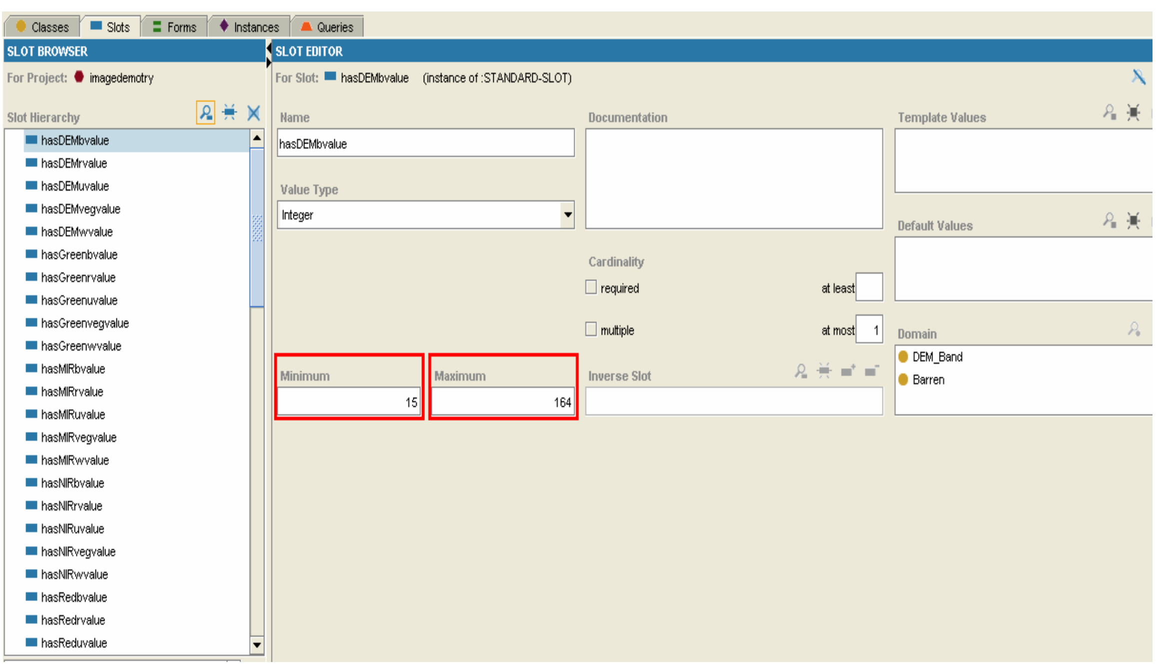


Fig 5.7: Properties Editor Window of Protégé

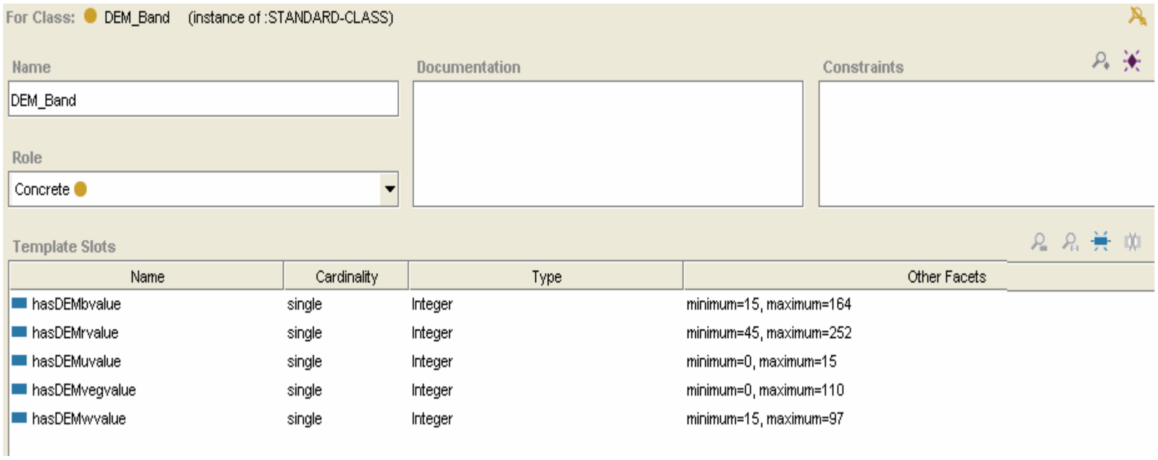


Fig 5.8: Constraints Editor on DEM_Band class

5.2.3 Matching Region Detection/ Classification

Classification is the process of assigning classes to the pixels in a remotely sensed image. These classes are Barren land, Vegetation, Water, rocky, Urban etc. Our task is to identify, for each pixel in one image, which class should be assigned to that pixel. Since there are very fewer classes than possible values for pixels we can consider classification as a process of simplification. If we assume that the image was classified correctly, we can easily do tasks such as area measurement and region extraction. The algorithm used for matching the individual pixel of the image with description of an image in ontology is done as follows:

5.2.3.1 *Algorithm for the Semantic Classifier*

1. Reading the image in an format (.tiff , .bmp, .jpg, .gif) etc.

2. Construction of Geo-ontology which defines different features as a class for classification i.e. water, rocky, vegetation, sand etc.

3. Making the Semantic classifier that will move iteratively for each feature

4. for each pixel on an image

If pixel validates all properties of a particular feature in the Bands selected for that feature

Then

Final image will contain that feature for a validated region.

Else

Move to another feature ontology.(step 3)

5. Finally we will have the classified image of the region.
6 .Visual representation can made available on Google map/Earth through
7. GoogleEarthtoolBox by specifying the latitude, Longitude and altitude of the region.
8. Overlay the classified image in googleearth so that better decision making and analysis can be done at critical time.

As the ontology construction for the geology domain has been completed, we need to call the ontology and satellite image on the single platform in order to compare them. As ERDAS and Protégé alone are not having such ability to call the image and ontology together so we need to build a middleware for them. The middleware has been built on JAVA platform with JENA and JAI API. Java is having the capability to call the ontology as well as satellite image on one program by the use above API's.
We need to compare the description of a satellite image i.e. geo ontology with the 7 band satellite image entered by the user. We have to compare the each pixel DN value with the attribute value pair of geo-ontology for better specification.

Through this algorithm we will compare the user entered satellite image with the geo-ontology. Input to the system will be the 7 band satellite image of the region with some properties of the image like DN value, NDVI, SBI etc. These values for the region are compared against the description of image generated through geo-ontology with the help of training set, to generate the classified image. This query image has been compared with domain ontology to generate the result.

5.3 Accuracy Assessment K coefficients

Another area that is continuing to receive increased attention by remote sensing specialists is that of classification accuracy assessment. Historically, the ability to produce digital land cover classifications far exceeded the ability to meaningfully quantify their accuracy. In fact, this problem sometimes precluded the application of automated land cover classification techniques even when their cost compared favorably with more traditional means of data collection. The lesson to be learned here is embodied in the expression “A classification is not complete until its accuracy is assessed."

Accuracy assessment is a general term for comparing the classification to geographical data that are assumed to be true, in order to determine the accuracy of the classification process. Usually, the assumed-true data are derived from ground truth data. It is usually not practical to ground truth or otherwise test every pixel of a classified image. Therefore, a set of reference pixels is usually used. Reference pixels are points on the classified image for which actual data are (or will be) known. The reference pixels are randomly selected.

5.3.1 Classification Error Matrix

One of the most common means of expressing classification accuracy preparation of a classification *error matrix* (sometimes called a confusion *matrix* or a *contingency table).* Error matrices compare, on a category-by category basis, the relationship between known reference data (ground truth) and the corresponding results of an automated classification. Such matrices are square, with the number of rows and columns equal to the number of categories whose classification accuracy is being assessed.

This matrix stems from classifying the sampled training set pixels and listing the known cover types used for training (columns) versus the pixels actually classified into each land cover category by the classifier (rows). Several characteristics about classification performance are expressed by an error matrix. For example, one can study the various classification errors of omission (exclusion) and commission (inclusion).

Once accuracy data are collected (either in the form of pixels, cluster of pixels, or polygons) and summarized in an error matrix, they are normally subject to detailed interpretation and further statistical analysis. A further point to be made about interpreting classification accuracies is the fact that even a completely random assignment of pixels to classes will produce percentage correct values in the error matrix. In fact, such a random assignment could result in a surprisingly good apparent classification result. The k ("KHAT") statistic is a measure of the difference between the actual agreement between reference data and an automated classifier and the chance agreement between the reference data and a random classifier. Conceptually, k can be defined as

$$k = \frac{\text{observed accuracy - chance agreement}}{\text{1 - Chance agreement}}$$

This statistic serves as an indicator of the extent to which the percentage correct values of an error matrix are due to "true" agreement versus "chance" agreement. As true agreement (observed) approaches 1 and chance agreement approaches 0, k approaches 1. This is the ideal case. In reality, k usually ranges between 0 and 1. For example, a k value of 0.67 can be thought of as an indication that an observed classification is 67 percent better than one resulting from chance. A k with the value of 0 suggests that a given classification is no better then a random assignment of pixels. In cases where chance agreement is large enough, k can take on negative values-an indication of very classification performance. (Because the possible range of negative values depends on specific matrix, the magnitude of negative values should not be interpreted as an indication of relative classification performance).

The KHAT statistic is computed as

$$\hat{k} = \frac{N \sum_{i=1}^{r} x_{ii} - \sum_{i=1}^{r} (x_{i+} \cdot x_{+i})}{N^2 - \sum_{i=1}^{r} (x_{i+} \cdot x_{+i})}$$

Where:

r = number of rows in the error matrix

x_{ii} = the number of observations in row i and column i (on the major diagonal)

x_{i+} = total of observations in row i (shown as marginal total to right of the matrix)

x_{+i} = total of observations in column i (shown as marginal total at bottom of the matrix)

N = total number of observations included in matrix

This K co-efficient has been calculated for our technique to compare the accuracy of the results produced by Semantic Classifier with other traditional and existing techniques.

Chapter 6

Case Study

6.1 Case Study: Image Classification for ALWAR REGION

In this we illustrate how Image Classification is done using our approach defined in section 5. This case study is based on the remote sensing image obtained from DRDO lab i.e. Defense Terrain Research Lab (DTRL). They have provided us the satellite image of the region. We have classified them using our ontology approach. We have to differentiate between the regions/classes i.e. Water, Vegetation, rocky, barren, urban in ALWAR REGION image. In the final image these classes are depicted in different colors. The accuracy of the system has been measured on the basis of the Kappa coefficient.

Initially we have 7 bands Satellite Image of The ALWAR REGION as shown in figure 6.1. The three steps of our system are applied on these images as follows:

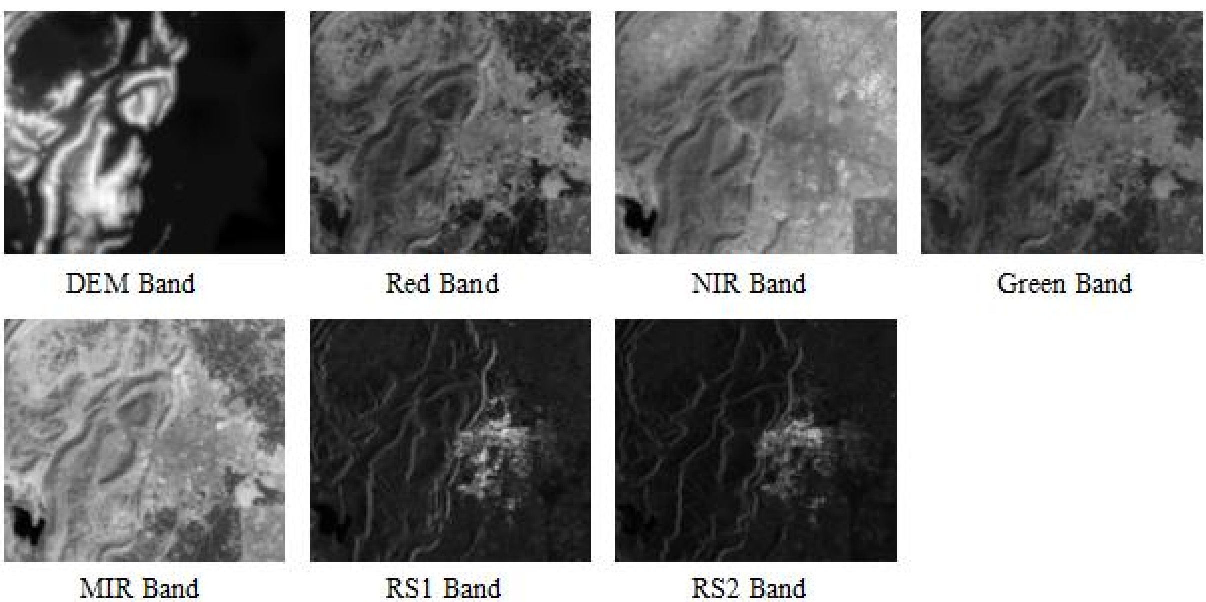


Fig 6.1: Seven band Satellite Image of ALWAR REGION

6.1.1 Training Set Generation for ALWAR REGION

We will open these entire 7 band image in the ERDAS IMAGINE. We will manually select pixels in the image depicting different classes and assign some color to similar class pixels. The class definition is done in Signature Editor of ERDAS. The training set for each class has been constructed by selecting 500 pixels of each class from the image. The snap shot of training set generated for 5 different classes are depicted in Table 3, 4,5,6,7.

Table 3: Snap shot of Training set for Barren Class

RED	GREEN	NIR	MIR	RS1	RS2	DEM	DECISION
115	91	182	126	20	15	30	Barren
111	90	173	131	17	34	15	Barren
121	91	182	118	26	17	40	Barren
125	98	188	128	25	21	27	Barren
130	94	186	128	24	33	25	Barren
138	106	212	137	32	22	115	Barren
170	127	234	159	37	45	100	Barren
156	123	204	142	27	57	78	Barren

Table 4: Snap shot of Training set for Rocky Class

RED	GREEN	NIR	MIR	RS1	RS2	DEM	DECISION
62	49	135	91	44	40	94	Rocky
84	64	160	102	20	25	165	Rocky
52	45	129	85	15	29	107	Rocky
91	69	171	106	10	46	123	Rocky
87	67	168	104	8	21	157	Rocky
76	59	157	95	9	47	114	Rocky
70	51	159	95	11	46	127	Rocky
82	59	159	100	7	9	173	Rocky

Table 5: Snap shot of Training set for Urban Class

RED	GREEN	NIR	MIR	RS1	RS2	DEM	DECISION
128	106	184	142	22	35	15	Urban
128	120	155	118	18	36	15	Urban
117	109	157	122	19	21	15	Urban
123	109	166	137	10	13	15	Urban
109	103	160	131	20	26	15	Urban
109	99	162	133	27	33	15	Urban
117	101	166	115	22	13	15	Urban
115	101	164	137	17	35	15	Urban

Table 6: Snap shot of Training set for Vegetation Class

RED	GREEN	NIR	MIR	RS1	RS2	DEM	DECISION
21	35	65	192	24	33	9	Vegetation
23	37	73	192	18	33	14	Vegetation
27	37	82	170	30	30	13	Vegetation
21	35	69	252	20	22	11	Vegetation

19	37	85	234	29	37	10	Vegetation
17	32	69	246	25	37	10	Vegetation
29	40	76	186	23	21	14	Vegetation
15	25	60	237	13	26	12	Vegetation

Table 7: Snap shot of Training set for Water Class

RED	GREEN	NIR	MIR	RS1	RS2	DEM	DECISION
25	30	25	10	180	92	15	Water body
25	30	25	10	157	60	15	Water body
25	30	21	20	8	2	30	Water body
21	27	20	20	2	5	30	Water body
23	30	16	14	3	4	30	Water body
21	24	14	10	1	2	30	Water body
23	25	14	12	1	1	30	Water body
23	27	12	10	2	1	30	Water body

6.1.2 Geo- Ontology Construction

With the above generated training set the ontology is constructed for the system as shown in fig. 5.4 which comprises of Classes like Bands, regions, pixels. Subclasses of these are Red, Green, DEM, RS1, RS2, NIR, MIR similarly region sub classes are water, barren, urban, rocky, vegetation. Each subclass has some property been assigned like hasRedbarrenvalue which indicates the DN value in red band for barren class. The values corresponding to these properties are taken from training set of particular class. Like we

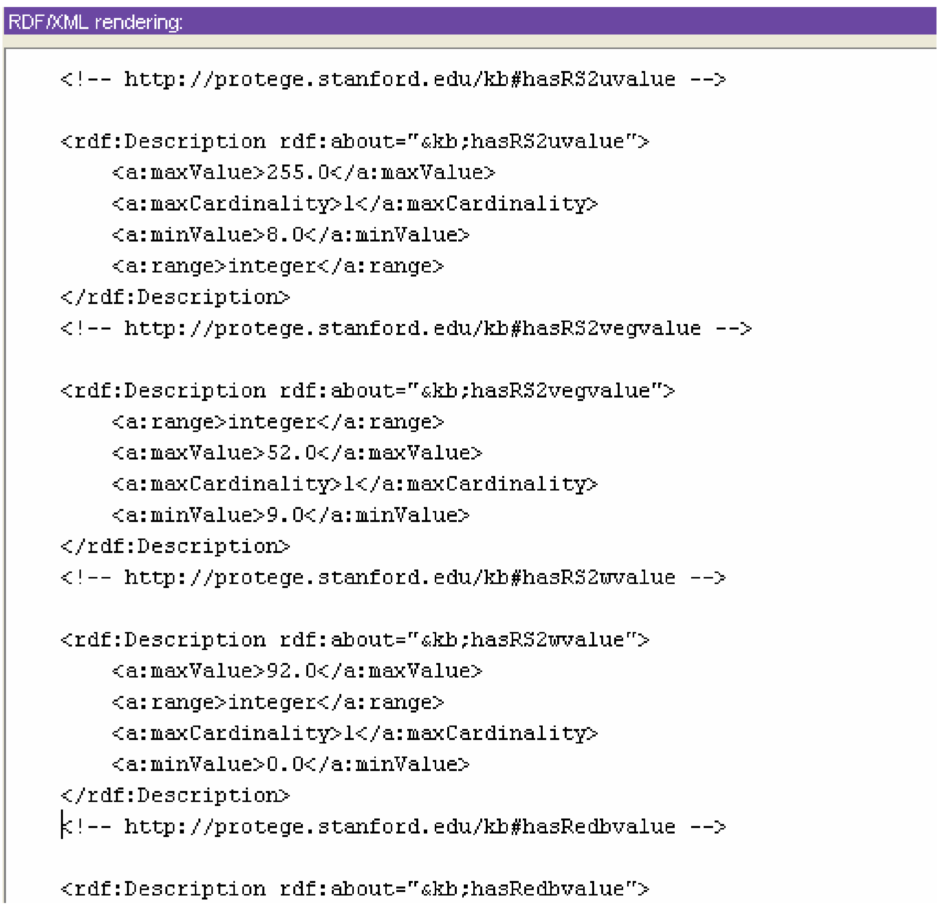
Table 8: Training Set for Geo-Ontology Construction

Regions/ Bands	Barren		Rocky		Urban		Water		Vegetation	
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
Red	97	179	25	127	62	185	13	25	3	39
Green	70	143	20	99	62	146	22	40	17	51
NIR	160	248	74	202	100	228	0	25	45	120
MIR	98	173	42	138	76	173	0	20	137	255
RS1	8	111	4	87	8	252	0	180	9	87
RS2	14	140	8	100	8	255	0	92	9	52
DEM	15	164	45	252	0	15	15	97	0	110

have taken the minimum and maximum value of the training set for particular class in particular band as shown in table 8. These minimum and maximum values are then used to construct the property relationship like hasRedbvalue, hasGreenbvalue etc. as shown in table 9. The snap shot for the RDF/XML code corresponding to generated Geo-Ontology is shown in fig. 6.2.

Table 9: Snap shot of Properties assigned to bands and region classes

S.No	Class Name	Attribute	Value	Description
1.	Barren	hasRedbarrenvalue	97-179	DN value of Barren region in Red Band
		hasGreenbarrenvalue	70-143	DN value of Barren region in Green Band
		hasNIRbarrenvalue	160-248	DN value of Barren region in NIR Band
		hasMIRbarrenvalue	98-173	DN value of Barren region in MIR Band
		hasRS1barrenvalue	8-111	DN value of Barren region in RS1 Band
		hasDEMbarrenvalue	15-164	DN value of Barren region in DEM Band
		hasRS2barrenvalue	14-140	DN value of Barren region in RS2Band
2.	Red Band	hasRedbarrenvalue	97-179	DN value of Barren region in Red Band
		hasRedrockyvalue	25-127	DN value of Rocky region in Red Band
		hasRedurbanvalue	62-185	DN value of Urban region in Red Band
		hasRedwatervalue	13-25	DN value of Water region in Red Band
		hasRedvegetationvalue	3-39	DN value of Vegetation region in Red Band


RDF/XML rendering:

```
<!-- http://protege.stanford.edu/kb#hasRS2uvalue -->

<rdf:Description rdf:about="&kb;hasRS2uvalue">
    <a:maxValue>255.0</a:maxValue>
    <a:maxCardinality>1</a:maxCardinality>
    <a:minValue>8.0</a:minValue>
    <a:range>integer</a:range>
</rdf:Description>
<!-- http://protege.stanford.edu/kb#hasRS2vegvalue -->

<rdf:Description rdf:about="&kb;hasRS2vegvalue">
    <a:range>integer</a:range>
    <a:maxValue>52.0</a:maxValue>
    <a:maxCardinality>1</a:maxCardinality>
    <a:minValue>9.0</a:minValue>
</rdf:Description>
<!-- http://protege.stanford.edu/kb#hasRS2wvalue -->

<rdf:Description rdf:about="&kb;hasRS2wvalue">
    <a:maxValue>92.0</a:maxValue>
    <a:range>integer</a:range>
    <a:maxCardinality>1</a:maxCardinality>
    <a:minValue>0.0</a:minValue>
</rdf:Description>
<!-- http://protege.stanford.edu/kb#hasRedbvalue -->

<rdf:Description rdf:about="&kb;hasRedbvalue">
```

Fig. 6.2: Snap shot for RDF/XML rendering for Geo-Ontology

The Geo-Ontology for the above generated training set has been constructed in RDF/ XML format. The constraints are applied in the manner as shown in fig. 5.5(A) and (B).

6.1.3 Matching Region Detection

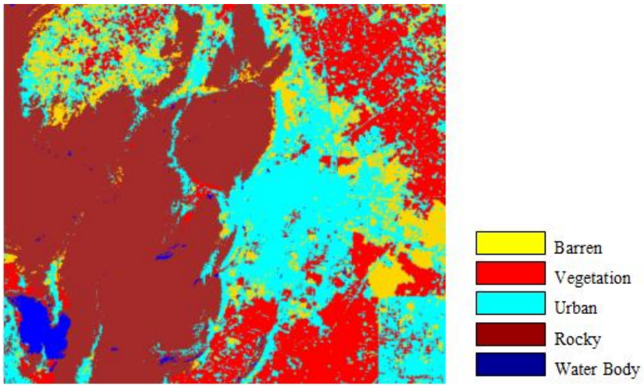
The matching algorithm sated above is coded in JAVA language. Here we have made use of Java API i.e. JENA API for traversal of Ontology and JAI API for Satellite image. Following snap shot fig. 6.3 for java code can be viewed as checking for Urban land condition on pixels.

```
161
162             public static boolean urbancheck(Model m, int w, int h)
163            {
164             int i,offset;
165             offset= h*width+w;
166             boolean flag=true;
166             for(i=0;i<num;i++)
168             {
169                 Resource iter1 = m.getResource(res+urban[i]);
171                  Property propmin=m.getProperty(minvalue);
172                  Property propmax=m.getProperty(maxvalue);
173
174                  Statement stmmin=m.getProperty(iter1, propmin);
175                  Statement stmmax=m.getProperty(iter1, propmax);
176                  float min=Float.parseFloat(stmmin.getObject().toString());
177                  float max=Float.parseFloat(stmmax.getObject().toString());
178                  if(pixel[i][offset]>=(int)min && pixel[i][offset]<=max)
179                      flag=true;
180                  else
181                  { flag=false;
182                    return flag;}
183             }
184             return flag;
185            }
186
```

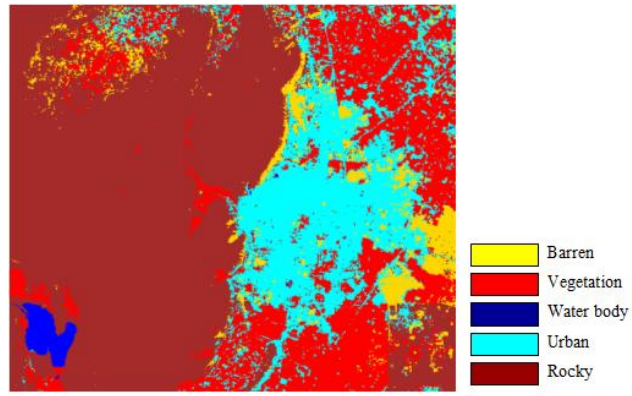
Fig. 6.3: Showing part of Java code for urban land cover check

As here in the above code we can see, a function having arguments the Model i.e. Geo ontology of the Satellite Image along with it the x and y location of the pixel in the image. Firstly we have to move to that particular statement of the RDF/XML code where we have provided description for the Urban land cover as shown in line no. 169. Then we have to move to property of the urban class specifying for minimum and maximum DN value of the pixel as shown in line no. 171 and 172. Then we have compared this minimum and maximum value with the original value of the pixel as shown in line no. 178. If the pixel DN value validates the condition then the pixel falls in urban class else pixel will move for other classes check.

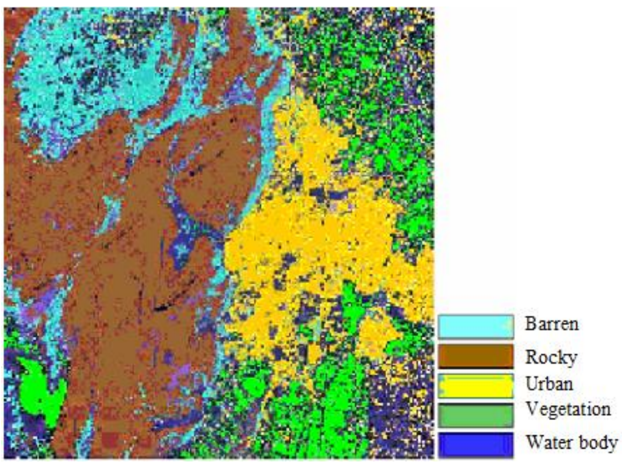
6.1.4 Generated Results



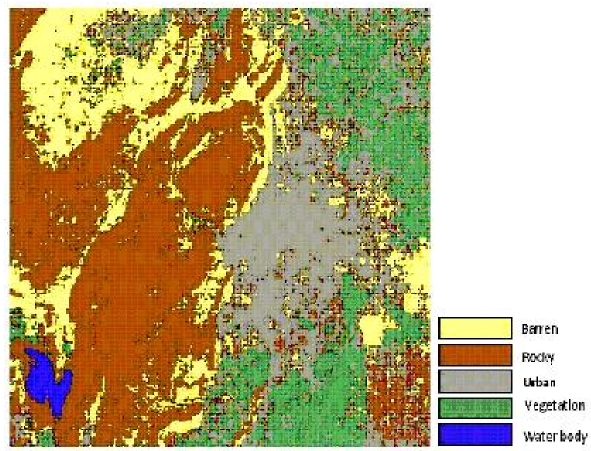
Minimum Distance Classifier



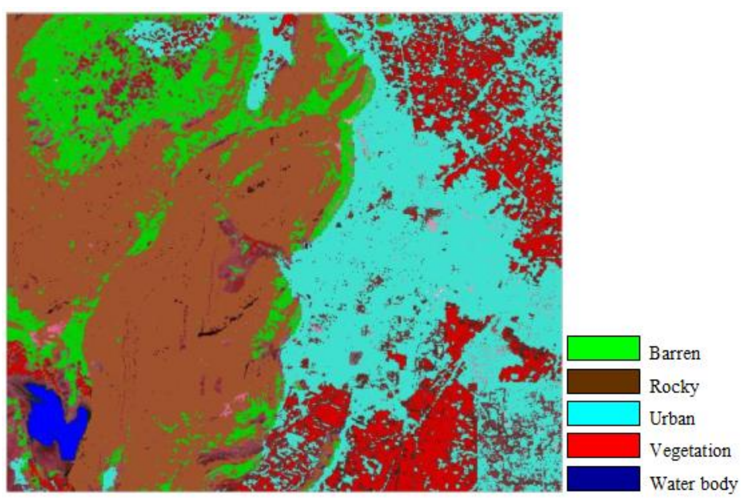
Maximum likelihood Classifier



Fuzzy classifier



Rough Set Classifier



Semantic Classifier

Fig 6.4: Classified Images of five Different Classifier

As here fig 6.4, in generated results we can see that Maximum Likelihood is showing very less barren land cover which is indicated by yellow color. The barren land cover is distinctly and completely shown in Semantic Classifier. Similarly some water is shown in Minimum Distance on Rocky portion where water is not possible. In semantic classifier exact location of water is depicted which is correct as well.

6.2 Accuracy Assessment (K coefficient) and Error Matrix

The Ꝁ -coefficients of the classification by using Semantic Classifier is illustrated in Table-10. Ꝁ-coefficient of the classification results from MLC, MDC, Fuzzy, Rough Set and Semantic Classifier is given in Table-11.

The KHAT statistic is computed as

$$\hat{k} = \frac{N \sum_{i=1}^{r} x_{ii} - \sum_{i=1}^{r} (x_{i+} \cdot x_{+i})}{N^2 - \sum_{i=1}^{r} (x_{i+} \cdot x_{+i})}$$

After implementing this formula in MATLAB code the k-coefficient of the Semantic Classifier comes out to be 0.9881.

Table 10: Accuracy Table for Semantic Classifier for ALWAR Region

Regions	Barren	Urban	Rocky	Vegetation	Water	Total
Barren	176	15	0	0	0	191
Urban	0	417	0	0	0	417
Rocky	2	0	286	0	0	288
Vegetation	0	0	0	329	0	329
Water	0	0	0	0	206	206
Total	176	417	286	329	206	1431

(K)Kappa co-efficient = 176+417+286+329+206=1414/1431 = **98.81%**
1431

Table 11: K- coefficient of MDC, MLC , Fuzzy, Rough Set and Semantic Classifier

Minimum Distance Classifier	Maximum-Likelihood Classifier	Fuzzy Classifier	Rough Set Classifier	Semantic Classifier
$\kappa = 0.7364$	$\kappa = 0.7525$	$\kappa = 0.9134$	$\kappa = 0.9525$	$\kappa = 0.9881$

6.2.1 Matlab Code for Accuracy assessment

For accuracy assessment of any technique we need to calculate the number of training pixels which are following the rules. Like the number of pixels following the barren class rules. Following code has been used for accuracy assessment:

```
clear all;
clc;

red=xlsread('F:\imageclassifier\alwar classified image\Alwar training sets.xls','Sheet1','a1227:a1432');
green=xlsread('F:\imageclassifier\alwar classified image\Alwar training sets.xls','Sheet1','b1227:b1432');
NIR=xlsread ('F: \imageclassifier\alwar classified image\Alwar training sets.xls','Sheet1','c1227:c1432');
MIR=xlsread('F:\imageclassifier\alwar classified image\Alwar training sets.xls','Sheet1','d1227:d1432');
rs1=xlsread('F:\imageclassifier\alwar classified image\Alwar training sets.xls','Sheet1','e1227:e1432');
rs2=xlsread('F:\imageclassifier\alwar classified image\Alwar training sets.xls','Sheet1','f1227:f1432');
DEM=xlsread('F:\imageclassifier\alwar classified image\Alwar training sets.xls','Sheet1','g1227:g1432');
[m,n]=size(green);
%figure; imshow(green)
%[n,m,o]=size(A);

barren=0;
vegetation=0;
water=0;
urban=0;
rocky=0;
for i=1:m
    %i
    for j=1:n

     if((red(i,j)>=62) & (red(i,j)<=185) & (DEM(i,j) >= 0)&
(DEM(i,j)<=15)&(green(i,j)>=62) & (green(i,j)<=146) & (NIR(i,j) >=
100)& (NIR(i,j)<=228)&(MIR(i,j)>=76) & (MIR(i,j)<=173) & (rs1(i,j) >=
8)& (rs1(i,j)<=252)& (rs2(i,j) >= 8)&(rs2(i,j)<=255)) %THEN Urban
            M(i,j,1)= 64;
            M(i,j,2) = 224;
            M(i,j,3) = 208;
            Urban=urban+1;

elseif((red(i,j)>=97) & (red(i,j)<=179) & (DEM(i,j) >= 15)&
(DEM(i,j)<=164)&(green(i,j)>=70) & (green(i,j)<=143) & (NIR(i,j) >=
160)& (NIR(i,j)<=248)&(MIR(i,j)>=98) & (MIR(i,j)<=173) & (rs1(i,j) >=
8)& (rs1(i,j)<=111) &(rs2(i,j) >= 14)&(rs2(i,j)<=140)) % THEN Barren
    M(i,j,1)= 2;
            M(i,j,2) = 208;
            M(i,j,3) = 2;
```

```
            barren=barren+1;

elseif((red(i,j)>=3) & (red(i,j)<=39) & (DEM(i,j) >= 0)&
(DEM(i,j)<=110)&(green(i,j)>=17) & (green(i,j)<=51) & (NIR(i,j) >= 45)&
(NIR(i,j)<=120)&(MIR(i,j)>=137) & (MIR(i,j)<=255) & (rs1(i,j) >= 9)&
(rs1(i,j)<=87)& (rs2(i,j) >= 9)&(rs2(i,j)<=52))% THEN Vegetation
    M(i,j,1)= 208;
            M(i,j,2) = 2;
            M(i,j,3) = 2;
            vegetation=vegetation+1;

elseif((red(i,j)>=13) & (red(i,j)<=25) & (DEM(i,j) >= 15)&
(DEM(i,j)<=97)&(green(i,j)>=22) & (green(i,j)<=40) & (NIR(i,j) >= 0)&
(NIR(i,j)<=25)&(MIR(i,j)>=0) & (MIR(i,j)<=20) & (rs1(i,j) >= 0)&
(rs1(i,j)<=180)& (rs2(i,j) >= 0)&(rs2(i,j)<=92))% THEN Waterbody
    M (i, j, 1) =0;
             M (i, j, 2) =0;
             M(i,j,3)=255;
             Water=water+1;

elseif((red(i,j)>=25) & (red(i,j)<=127) & (DEM(i,j) >= 45)&
(DEM(i,j)<=252)&(green(i,j)>=20) & (green(i,j)<=99) & (NIR(i,j) >= 74)&
(NIR(i,j)<=202)&(MIR(i,j)>=42) & (MIR(i,j)<=138) & (rs1(i,j) >= 4)&
(rs1(i,j)<=87)& (rs2(i,j) >= 8)&(rs2(i,j)<=100))% THEN Rocky
    M(i,j,1) = 160;
            M(i,j,2) = 82;
            M(i,j,3) = 45;
            rocky=rocky+1;

%IF <empty> THEN Rocky

             End
   End
End
    barren
    vegetation
    urban
    water
    rocky
```

Chapter 7

Publication from Thesis

During the period of working over this project we interacted with International community working on web technologies. We discussed our approach for representing knowledge with them and collected the reviews and worked over the suggestion send to us. One Research papers have been accepted in International conference for presentation and will be published in their proceedings.

This paper presents the concept of knowledge representation with Ontology in Geology domain and with the use of that knowledge satellite image classification has been done. Image classification is the main domain of geology in remote sensing. This Project has been undertaken with collaboration of DTRL(Defence Terrain research laboratory) lab of DRDO.

7.1 The details of Conference publications :

Conference Name: *International Conference on Semantic Web and Web Services (SWWS-09), Las Vegas, USA.*

URL: http://www.world-academy-of-science.org/

Paper Title: *"Enabling Web Services for Semantic web based Satellite Image Classification"*

Authors: Dr. Daya Gupta, Sonal Kumar, Shashi Kumar, V.K. Panchal

Location: Las Vegas, USA.

Publishers/ proceedings: The accepted papers will be included in the conference proceedings, which has an ISBN number. The proceedings will be made available during the conference. The proceedings will also be submitted for several database indexes. The previous conferences are submitted for several reputed database indexes. Paper will be indexed at **DBLP** Bibliography Server.

Chapter 8

Conclusion and Future Work

This project has proposed a Framework for a tool as a Semantic Web Image Classifier which will classify the EOS images in a cheaper and easily accessed way. As we know that Ontology is the description of the concept and here we are using that concept for the geology domain in satellite image classification. We have made use of ontology concept in describing the features of satellite image in Geo-Ontology. This description is then used further for Classification of Satellite images. As ontology are generated by considering several factors of an image like DN values, NDVI, SBI etc. so they can be used as a generic tool for classification of image. The implementation of the complete Tool is done with the use of JENA Framework and JAI API for Traversing of ontology and loading and creating classified image correspondingly. These results will be compared against the result produced by the Minimum Distance to mean , maximum Likelihood algorithm, fuzzy Set and Rough Set to detect the level of accuracy of the classified image. The accuracy for the classified image is compared on the value of kappa coefficient. The geo-ontology is easily available online because they are building in a RDF/XML format which is web enabled format. By the help of this online available geo-ontology any satellite image comprising of those land cover can be easily classified.

The geo-ontology can be constructed in other technologies of semantic web like Notation3, Turtle, etc. And better rules can be used to accurately classify the image. This geo-ontology can be placed on net as they are in a web compatible format i.e. XML. The implementation of the Automatic Tool is proposed as a future work with the creation of such complex geo-ontology. Several experts and domain people will contribute their experience to develop well specified ontology for Semantic Web Image Classifier. As here in this project we have to create different ontology for different satellite image. No generic ontology is available till now which can be used as it is. So several work need to be done in this field to make some generic geo-ontology through which any image can be classified in minutes instead of in a day.

References

[1] Canada Center for Remote Sensing, "Fundamentals of Remote sensing", pg no. 1-164.

[2] Grigoris Antoniou and Frank van Harmelen, "A Semantic Web Primer ", pg no. 3-16, 2002.

[3] Wassim Messaoudi, Imed Riadh Farah, Karim sahib ettabaa, SEMANTIC STRATEGIC SATELLITE IMAGE RETRIEVAL, Laboratories de Recherche en Informatique Arabisee et Documentique Integree (R.I.A.D.I.) Ecole Nationale des Sciencesde 'Informatique .(http://ieeexplore.ieee.org/iel5/4520396/4529902/04529997.pdf ?arnumber=4529997).

[4] RDF/XML—http://jena.sourceforge.net/tutorial/RDF_API/index.html

[5] Matthew Horridge, Holger Knublauch , Alan Rector, Robert Stevens, Chris Wroe," A Practical Guide To Building OWL Ontologies Using The Prot´eg´e OWL Plugin and CO-ODE Tools Edition 1.0", pg no. 1-118, August 27, 2004

[6] Smith, Michael K.; Chris Welty, Deborah L. McGuinness (2004-02-10). "OWL Web Ontology Language Guide". W3C. http://www.w3.org/TR/owl-guide/. Retrieved on 2008-07-15.

[7] Alexandre Passant, FOAFMAP: Web 2.0 meets the semantic Web.

[8] "The friend of a friend(foaf) project," http://www.foaf-project.org/.

[9] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. C. Doshi, , and J. Sachs, "Swoogle: A search and metadata engine for the semantic web," in *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, 2004.

[10] Li Ding, Lina Zhou, Tim Finin, Anupam Joshi, "How the Semantic Web is Being Used: An Analysis of FOAF Documents", Department of Computer Science and Electrical Engineering, Department of Information Systems University of Maryland, Baltimore County , *Proceedings of the 38th Hawaii International Conference on System Sciences – 2005.*

[11] Ontogeo , http://ontogeo.ntua.gr/.

[12] Ontospace, http://134.102.58.154/.

[13] "A J2SE Application Enhances and Classifies Map Data", http://software.intel.com/en-us/articles/

[14] JAI API- http://java.sun.com/developer/onlineTraining/javaai/jai/index.html

[15] ERDAS Software -- http://www.erdas.com/

[16] JENA – www.jena.soureforge.net/

[17] T.M. Lillesand and R.W. kiefer "Remote Sensing & Image Interpretation " , 3rd edition, 1994.

[18] Protégé : http://protege.stanford.edu/doc/users.html

[19] Frederico Fonseca, Gilberto Câmara, Antonio Miguel Monteiro, "A Framework for Measuring the Interoperability of Geo-Ontologies," (http://www.dpi.inpe.br/gilberto/papers/scc_interoperability.pdf)

[20] OntoGeo -- (http://ontogeo.ntua.gr/)

[21] Eleni Tomai Maria Spanaki, "From ontology design to ontology implementation: A web tool for building geographic ontologies" (http://ontogeo.ntua.gr/publications/agile2005tomaispanaki.pdf)

[22] TIFF format: (http://en.wikipedia.org/wiki/Tagged_Image_File_Format)

[23] Mei Kun, Bian Fuling, An Ontology-Based Approach for Geographic Information Retrieval on the Web , 1-4244-1312-5/07 IEEE.

[24] Hong Wang, Lin Li, Ping-chao Song, "DESIGN OF GEO ONTOLOGY BASED ON CONCEPT LATTICE," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B4. Beijing 2008.*

[25] Qian Lia, *, Haigang Suia, Yuanyuan Fenga, Qin Zhanb, Chuan Xua , RESEARCH ON REMOTE SENSING INFORMATION PROCESSING SERVICES BASED ON SEMANTIC WEB SERVICES, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B4. Beijing 2008.*

[26] S. S. Durbha, R.L. King, and N. H. Younan , "Leveraging the Semantic Web Technologies for Knowledge Discovery from Earth Observation Data Archives" , http://earth.esa.int/rtd/Events/ESA-EUSC_2005/Ab42_Durbha.pdf

[27] Y. Rui, T.S. Huang, "Optimizing learning in image retrieval", *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, June 2000, pp.1236-1243.*

[28] Y. Rui, T.S. Huang, S.-F. Chang, Image retrieval: current techniques, promising directions, and open issues, J. Visual Commun. Image Representation, 1999, pp.39–62.

[29] A. Vailaya, M.A.T. Figueiredo, A.K. Jain, H.J. Zhang, Image classification for content-based indexing, IEEE Trans. Image Process,2001, pp.117–130.

[30] N. Ruan, N. Huang, W. Hong, “Semantic-Based Image Retrieval in Remote Sensing Archive: An Ontology Approach”, Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006, pp. 2903-2906.

[31] Daya Gupta, Kartar Jat, “ Knowledge Representation with ontology and Relational database to RDF converter”, *In the proceedings of EISWT-08, ISBN : 978-1-60651-003-02, 2008, pp 141-46.*

[32] A. Pentland, R. W. Picard, and S. Sclaroff, “Photobook: Content-based manipulation for image databases,” *Int. J. Comput. Vis.*, vol. 18, no. 3,pp. 233–254, 1996.

[33] A. Gupta and R. Jain, “Visual information retrieval,” *Commun. ACM*,vol. 40, no. 5, pp. 70–79, May 1997.

Appendix A

An Introduction to ERDAS IMAGINE

ERDAS Software:

Overview

ERDAS® is pleased to provide ERDAS IMAGINE® version 8.4[15]. Many private and commercial users who need to extract and interpret information from imagery recognize ERDAS IMAGINE as a must have. With ERDAS IMAGINE 8.4, ERDAS' latest, most advanced release of ERDAS IMAGINE, production workflows are enhanced and simplified like never before. As an example, the Batch Wizard streamlines repetitive procedures such as importing, re-projecting, and exporting large numbers of files at once, using a wizard approach to record and "re-play" commonly used procedures.

Also featured is the IMAGINE Expert Classifier™ – a tool for graphically building and executing geographically aware, rules-based expert systems. This tool can be used to build decision support systems, classifiers for high-resolution imagery, GIS analysis techniques, etc. These can then be distributed to other users for use with their own data. The software consists of two parts — the Knowledge Engineer and the Knowledge Classifier:

Knowledge Engineer:

This program provides a graphical user interface for the "expert" to build a knowledge base. The knowledge base is represented as a tree diagram consisting of final and intermediate class definitions (hypotheses), rules (conditional statements concerning variables), and variables (raster, vector, or scalar). Hypotheses are evaluated by the use of rules – if one or more rules are true, then that hypothesis may be true at that particular location. A rule is evaluated based on input variables to determine if it is true. For instance, a rule could be that slopes must be gentle (less than 5 degrees) – to evaluate this, a variable is required determining the slope at every location. This could be in the form of an existing image specifying slope angles, it could come from a spatial model calculating slope on-the-fly from an input DEM, or it could even be an external program. Variables can also be defined from vectors and scalars. If the variables' value indicates that the rule is correct, this (combined with other correct rules) indicates that the hypothesis (class allocation) is true.

Key features:

- Graphical drag-and-drop tool for building the knowledge tree.
- Confidence value definition and propagation, or the ability to handle uncertainty, is of vital importance to the knowledge base. The expert places confidence in each rule, and as multiple rules are triggered within a tree, the Knowledge Classifier combines the confidences.
- Several rules could be true at a particular location – the one with the highest confidence is most likely to be the class for that pixel.
- Variables can be from various sources – images, vectors, scalars, graphical models, and even user-defined programs.
- The ability to include prompts for particular data files and variables enables the creation of portable knowledge bases
- Use spatial operators (as opposed to traditional per-pixel classifiers) via Model Maker.
- Enables the multiple AND'ing or OR'ing of rules through the construction of the tree branches horizontally or vertically.
- Pathway cursor enables quick feedback on the results of a classification to aid in developing and fine-tuning a knowledge base.
- Access to existing ERDAS IMAGINE tools, such as Model Maker for defining spectral/spatial operators, shortens the learning curve.

Knowledge Classifier:

With a previously created expert knowledge base, a less experienced user may use the Knowledge Classifier is to apply the knowledge base to data and perform a classification.

Key features:

- Wizard interface allows non-experts to apply the knowledge base to their own data.
- Evaluate all possible classification classes, or only consider a subset of rules.
- Identify missing files and prompt user to find them automatically.
- Options to output fuzzy sets and confidence layers, as well as a classification.
- Operator only requires an IMAGINE Advantage™ license.

The Knowledge Engineer is a standard part of IMAGINE Professional 8.4. The Knowledge Classifier is a standard part of IMAGINE Advantage 8.4. Consequently expert users can design their knowledge bases using IMAGINE Professional, but then these knowledge bases can be distributed to the thousands of IMAGINE Advantage users around the world to apply the knowledge-based classification process to their own data. This portability of knowledge bases is one of the keys to the strength of the expert systems approach.

Appendix B

An Introduction to Protégé Software

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 ontology. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontology in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. Further, Protégé can be extended by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications.

Ontology describes the concepts and relationships that are important in a particular domain, providing a vocabulary for that domain as well as a computerized specification of the meaning of terms used in the vocabulary. Ontology ranges from taxonomies and classifications, database schemas, to fully axiomatized theories. In recent years, ontology has been adopted in many business and scientific communities as a way to share, reuse and process domain knowledge. Ontology are now central to many applications such as scientific knowledge portals, information management and integration systems, electronic commerce, and semantic web services.

The Protégé platform supports two main ways of modeling ontology:

- The **Protégé-Frames** editor enables users to build and populate ontology that are *frame-based*, in accordance with the Open Knowledge Base Connectivity protocol (OKBC). In this model, ontology consists of a set of classes organized in a sub assumption hierarchy to represent a domain's salient concepts, a set of slots associated to classes to describe their properties and relationships, and a set of instances of those classes - individual exemplars of the concepts that hold specific values for their properties.
- The **Protégé-OWL** editor enables users to build ontology’s for the *Semantic Web*, in particular in the W3C's Web Ontology Language (OWL). "OWL ontology may

include descriptions of classes, properties and their instances. Given such an 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" (see the OWL Web Ontology Language Guide).

The Protégé-OWL editor is an extension of Protégé that supports the Web Ontology Language (OWL). OWL is the most recent development in standard ontology languages, endorsed by the World Wide Web Consortium (W3C) to promote the *Semantic Web* vision. "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" (see the OWL Web Ontology Language Guide).

The Protégé-OWL editor enables users to:

- Load and save OWL and RDF ontology.
- Edit and visualize classes, properties, and SWRL rules.
- Define logical class characteristics as OWL expressions.
- Execute reasoners such as description logic classifiers.
- Edit OWL individuals for Semantic Web markup.

Protégé-OWL's flexible architecture makes it easy to configure and extend the tool. Protégé-OWL is tightly integrated with Jena and has an open-source Java API for the development of custom-tailored user interface components or arbitrary Semantic Web services.

Appendix C

JAVA API (Application Programming Interface)

Application Programming Interface, a language and message format used by an application program to communicate with the operating system or some other control program such as a database management system (DBMS) or communications protocol. APIs are implemented by writing function calls in the program, which provide the linkage to the required subroutine for execution. Thus, an API implies that some program module is available in the computer to perform the operation or that it must be linked into the existing program to perform the tasks. The API is designed to facilitate the user and provide a better way of interaction. There are multiple real life problems where it is not possible that each user can understand a problem in a way a designer handle, API provide an interaction so that each end user can handle a problem in an easy way and perform the desired task with an ease.

One of the well known API is the packages are written in java since that runs well on all platforms. These should include:

- Be as uniform as possible in their interface.
- Be as interactive as possible -- with no "hard wired" input.
- Be as graphical as possible to provide graphical insight so that it will provide a better way of interaction with the problem.

API basically provides the way of interaction to user. There are multiple languages which will use the different API according to their need and facilities provided by these APIs.

JAI (Java Advanced Imaging) API

Introduction:

The Java Advanced Imaging API[14] extends the Java platform (including the Java 2D API) by allowing sophisticated, high-performance image processing to be incorporated into Java programs. Java Advanced Imaging is a set of classes which provide imaging functionality beyond that of Java 2D and the Java Foundation classes, although it is compatible with those APIs.

The Java Advanced Imaging API implements a set of core image processing capabilities including image tiling, regions of interest, threading and deferred execution. JAI also offers a set of core image processing operators including many common point, area and frequency-domain operators.

Java Advanced Imaging encapsulates image data formats and remote method invocations within a re-usable image data object, allowing an image file, a network image object or a real-time data stream to be processed identically. JAI follows the Java run time library model, providing platform independence with the "write once, run anywhere" paradigm. Client-server imaging is supported by way of the Java platform's networking architecture and remote execution technologies. Remote execution is based on Java RMI (remote method invocation). This allows Java code on a client to invoke method calls on objects that reside on another computer without having to move that object to the client.

Java Advanced Imaging follows an object model where both images and image operators are defined as objects sub classed off of a common parent. An operator object is instantiated with one or more image sources and other parameters. This operator object may then become an image source for the next operator object. The connections between the objects define the flow of processed data. The resulting editable graphs of image processing operations may be defined and instantiated as needed. JAI also provides an extensible framework that allows customized solutions to be added to the core API.

Data representation

Images in JAI may be multidimensional (i.e., several values associated to a single pixel) and may have pixels with integer or floating point values. Pixels may be packed in different ways or unpacked in the image data array. Also, different color models can be used.

In order to be able to represent a variety of image data, you must deal with an assortment of classes. Before I present examples of the classes in action, here are descriptions of the basic classes for image data representation.

- **Planar Image:** This is the basic class for image representation in JAI; it allows the representation of images with more flexibility than the Java class BufferedImage. (BufferedImage and PlanarImage use several different classes for flexible image data representation.) Image pixels are stored in an instance of Raster object, which contains a concrete subclass of DataBuffer filled accordingly to the rules described by one of the subclasses of SampleModel. A PlanarImage is read-only (i.e., it may be created), and pixels values may be read in several different ways, but there are no methods that allow the modification of pixels values.

- **Tiled Image:** This subclass of PlanarImage can be used for reading and writing image data.
- **RenderedOp:** This subclass of PlanarImage represents a node in a rendered imaging chain. A rendered imaging chain is a powerful and interesting concept in JAI that allows the processing of an image to be specified as a series of steps (operators and parameters) that are applied to one or more image.

Another interesting concept in JAI is tiled images. Tiles can be considered subsets of the images that may be processed independently. Large images can be processed in JAI with reasonable performance (even through rendered imaging chains), since there is no need to load the whole image data in memory at once. If the image is tiled, all its tiles must have the same width and height.

Benefits of JAI

The JAI API offers application developers various advantages in comparison to other imaging solutions. These advantages include the following:

- It's a cross-platform imaging platform with the support of distributed calculations.
- It has an object-oriented API that is extensible and flexible.
- It supports Java's Remote Method Invocation and the Internet Imaging Protocol for its network-based imaging; this allows for scalable solutions from PDAs, laptops, desktops, and high-end servers.

JENA API

Jena [16] is a Java API for semantic web applications. The key RDF package for the application developer is `com.hp.hpl.jena.rdf.model`. The API has been defined in

terms of interfaces so that application code can work with different implementations without change. This package contains interfaces for representing models, resources, properties, literals, statements and all the other key concepts of RDF, and a ModelFactory for creating models. So that application code remains independent of the implementation, it is best if it uses interfaces wherever possible, not specific class implementations.
The `com.hp.hpl.jena...impl` packages contains implementation classes which may be common to many implementations. For example, they defines classes `ResourceImpl`, `PropertyImpl`, and `LiteralImpl` which may be used directly or subclassed by different implementations. Applications should rarely, if ever, use these classes directly. For example, rather than creating a new instance of `ResourceImpl`, it is better to use the `createResource` method of whatever model is being used. That way, if the model implementation has used an optimized implementation of `Resource`, then no conversions between the two types will be necessary.

The Jena Ontology API is language-neutral: the Java class names do not mention the underlying language. For example, the OntClass Java class can represent an OWL class, RDFS class, or DAML class. To represent the differences between the various representations, each of the ontology languages has a *profile*, which lists the permitted constructs and the names of the classes and properties. Thus in the DAML profile, the URI for object property is daml:ObjectProperty (short for http://www.daml.org/2001/03/daml+oil#ObjectProperty), in the OWL profile is it owl:ObjectProperty (short for http://www.w3.org/2002/07/owl#ObjectProperty) and in the RDFS profile it is null since RDFS does not define object properties.

The profile is bound to an *ontology model*, which is an extended version of Jena's Model class. The base Model allows access to the statements in a collection of RDF data. OntModel extends this by adding support for the kinds of objects expected to be in an ontology: classes (in a class hierarchy), properties (in a property hierarchy) and individuals. When you're working with an ontology in Jena, all of the state information remains encoded as RDF triples (which Jena calls Statements) stored in the RDF model. The ontology API *doesn't change the RDF representation of ontologies*. What it does do is add a set of convenience classes and methods that make it easier for you to write programs that manipulate the RDF statements.

The predicate names defined in the ontology language correspond to the accessor methods on the Java classes in the API. For example, an OntClass has a method to list its super-classes, which corresponds to the values of the subClassOf property in the RDF representation. This point is worth re-emphasising: no information is stored in the OntClass object itself. When you call the OntClass listSuperClasses() method, Jena will retrieve the information from the underlying RDF statements. Similarly adding a subclass to an OntClass asserts an additional RDF statement into the model.