

1 INTRODUCTION

“Those who wish to succeed must ask the right preliminary questions”

Aristotle

In recent years various novel paradigms have been proposed for solving computational problem. Traditional computing methods fail to provide solution to certain real life problems. Natural computation combines all the features and disciplines of nature to solve computationally unsolvable problem that is hardly solved by our traditional computing science.

Thus we have applied Cuckoo search, a new emerging field of Natural Computation in Groundwater detection. Cuckoo search is integrated with Case based reasoning for our application. Case based reasoning is a newly emerging artificial intelligence technology; CBR provides a new approach to learn from the successful cases in the past and can imitate the human operation experiences [2]. Generally groundwater occurs below the surface of earth, where it occupies all or part of the void spaces in soils or geologic strata [3]. To find groundwater possibility manually is quite impossible thus we devised a system which outputs groundwater probability without manual intervention.

Satellite remote sensing has been recognized as a valuable tool for viewing, analysing, characterizing and making decisions about our environment [4][5]. Image classification is applied to remotely sensed landuse data to generate a map for image analysis. Thus we have applied another branch of Natural Computation i.e. Membrane Computing in Image Classification. Image classification is one of the important approaches for recognizing different terrain features. Images contain various topographic traits and it is the specific task of the analysts to identify these features contained in an image.

1.1 Motivation

Groundwater possibility detection is covered in our research work. Water covers 70.9% of the Earth's surface and is vital for all forms of life. The existence of human society cannot be overemphasized without conservation of water. Groundwater is generally obtained from beneath the earth surface. This groundwater is recycled to obtain purified form of water for drinking and other work. Groundwater constitutes only 0.6% of all the water on this earth planet, 97.4% accounts for seawater and 2% for snow and ice on the poles [6]. Hence groundwater is an important commodity which we use for various purposes such as domestic, industrial and agricultural use but with the increase in population its resources are depleting. Thus we need a mechanism to preserve water as well as a methodology which analyzes the percentage of water at certain areas. Geoscientist needs a system to understand the potentiality of water at certain regions. Though some system exists which finds out the potentiality of groundwater but their efficiency is very low. Moreover it is not possible for scientist to migrate from one position to another to find out the existence of water. Thus we have developed a mechanism which finds out the existence of groundwater without any drilling mechanism. We have developed software which answers the query of geoscientist. Thus we have applied one of the new techniques of natural computation to this optimization problem.

Cuckoo search is a new metaheuristic algorithm comes under natural computation technique. It was introduced by Xin-She Yang and Suash Deb in the year 2009 in there research paper titled "Cuckoo Search via Lévy Flights" [7]. This algorithm is based on the obligate brood parasitic behaviour of cuckoo species in combination with the Lévy flight behaviour of some birds and fruit flies. We have applied this new metaheuristic algorithm in combination with case based reasoning to detect the presence of groundwater in certain regions. Cuckoo search

is combined with case based reasoning in this application. One of the important features of cuckoo search is that it has a huge search space as compared with other metaheuristic algorithm. As we have huge database, as provided by experts we need a technique which efficiently searches all possible cases. Thus we used the strength of cuckoo search in groundwater potentiality.

As results of Natural computation were very promising in the above optimization problem thus we even applied one branch of Natural Computation i.e. Membrane Computing in Image Classification. It has lower cost and higher degree of classification accuracy thus it is able to replace high resolution high cost satellite imageries. Most of natural computation technique like Biogeography Based Optimization (BBO), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) take an image in the form of clusters or group of pixels. Image does not contain each feature distinguishly, some features are intertwined with other features. Due to mixture of feature at some portion of images the concept of mixed pixels came into existence. Mixed pixels are hardly resolved by these computational techniques. But our natural computation technique i.e. Membrane Computing can identify this heterogeneous portion in images and can classify each portion perfectly.

Our proposed work is motivated from bio-inspired technique called Membrane Computing technique and bases its fundamentals from the research paper titled “An Introduction to and an Overview of Membrane Computing” by Gheorghe Păun and Grzegorz Rozenberg [8]. The main characteristics of Membrane Computing for satellite image classification are that this technique is flexible to classify each feature more efficiently than other techniques. Moreover MC works with single unit of object rather than group of objects. Even MC has an important feature i.e. maximal parallelism thus decreasing the time complexity of many computationally unsolvable problems.

1.2 Related Work

Detection of Groundwater is a field of research which is being solved by geoscientist through their traditional mechanism such as Ground Penetrating Radar (GPR) [9]. But it is a new field of research for computer scientists. Few works are being carried out in this area. Particle Swarm Optimization (PSO) [10] and Biogeography Based Optimization [11] have already been applied for detection method. For prediction mechanism they had used case based reasoning. In some cases the output shows discrepant result.

To produce correct output in all cases we have applied Cuckoo search along with some propositional logic. As cuckoo tries to find out the most suitable host nest to lay their eggs similarly our algorithm always find the best old case that mostly matches with recent new case. In almost all cases our algorithm produced correct output. The efficiency of metaheuristic algorithm is that they imitate the best features in nature, especially the selection of the fittest in biological systems have evolved by natural selection over millions of years [12].

In remote sensing the problem of satellite image classification has been solved by using the traditional classical approaches like Parallelopiped Classification [5], Minimum Distance to Mean Classification [5], Maximum Likelihood Classification [5] etc. However these techniques show limited accuracy in information retrieval and high resolution image is needed. Also these techniques are insensitive to different degrees of variance in the spectral response data.

To provide a solution to the above problems, soft computing techniques were introduced in remote sensing for image classification. The principal constituents of soft computing techniques are fuzzy logic [13], rough set theory [14], neural network theory, and

probabilistic reasoning and Swarm Intelligence techniques [15] with the latter subsuming belief networks, genetic algorithms, chaos theory and parts of learning theory.

However the soft computing techniques like the fuzzy classifier [13], and the rough set classifier [14] were not able to provide good result in case of ambiguity sine the mail goal of these techniques was to synthesize approximation of concepts from the acquired data. Hence these techniques did not provide very much accurate results with low spatial resolution images. Also these techniques were not able to handle the crisp and continuous data separately.

The solution to the above drawbacks was provided by recently introduced concept of swarm intelligence [15] [16] which come under natural computation. A hybrid BBO/ACO2/PSO was formulized [17]. This technique improves the classification of multi-spectral satellite images. Though this classifier identifies the features efficiently on the experimental image but it does not give accurate result if the same algorithm is applied to different experimental data. Moreover due to existence of BBO in the algorithm heterogeneous region would hardly get classified.

To overcome the above mentioned difficulties we have proposed Membrane computing in image classification. The proposed algorithm once applied to different dataset the result is satisfactory. As it works on single unit object thus heterogeneity factor can easily be detected and identified in an image.

1.3 Problem Statement

The aim of this work is to propose a branch of natural computation i.e. Cuckoo Search in Case Based Reasoning for detection of groundwater possibility and another branch of natural computation i.e. Membrane Computing is used in terrain feature extraction. The images are generally high resolution satellite multi-spectral images. Cuckoo search is a new technique which is recently added in natural computation. Two techniques are integrated together i.e. Cuckoo search and Case based reasoning and then both of them are applied in groundwater detection mechanism. Membrane computing is basically a model of computation that is motivated by the structure and functioning of living cells- more specifically, by the role of membranes in compartmentalization of living cells into “protected reactors” [8]. Basically most of the natural computation technique proposed in land cover feature extraction method uses clusters for each feature but in Membrane computing we did not use clusters rather we have worked on single unit of an image i.e. pixels. Thus our problem statement can be proposed as follows-

“Our research work is two folded. The first work aimed at detecting groundwater possibility at various geological portions using Cuckoo search along with Case based reasoning and the second work adapted Membrane computing technique for a more focussed satellite image classification”

1.4 Scope of the work

Our approach is to use *Cuckoo Search in Groundwater Detection as well as Membrane Computing in Image classification* to able to perform some very important targets successfully completed

The proposed detector used in groundwater possibility is applied to data as given by user. Expert has given us data containing the main attributes which are necessary to know about the groundwater possibility. The attributes are generally geology, landform, landuse, soil, lineament, and slope. Depending on their value experts say the presence of groundwater in that region. Even our detector detected the presence of groundwater in a region as per given by user. Thus our detector can be used by geoscientist in their real life applications. It is a bio-inspired groundwater potentiality detector that detects the presence of groundwater without even manual intervention. Geologist, by using our system can understand the percentage of groundwater of a remote place without even going to such places.

The proposed algorithm has been applied to 7 band Indian Resourcesat Satellite image of dimension 472X576 of Alwar Region in Rajasthan (since it contains a variety of land cover features) to classify the area into 5 features namely water, urban, barren vegetation and rocky region and assign colour codes to them hence solving the problem of land cover feature extraction of this region. Our classifier is a universal classifier. It has been validated with other images too. We have been applied our classifier to 8 band Indian Resoucesat Satellite image of dimension 916X569 of Saharanpur region in Uttar Pradesh. It has even classified this image into barren, denseurban, urban, vegetation and water efficiently. Thus we can claim our classifier as a universal one. In fact, our classifier not only classifies the images properly but even with high efficiency rate and even consuming minimum computation and resources.

1.5 Organization of thesis

The remainder part of this thesis is organized in the following sections:-

Section 2: This section gives a detailed description of Case based reasoning along with Cuckoo search. Both are used in groundwater detection system. Thus this chapter is introduced in thesis work. Then the components and features of case based reasoning along with its life cycle are discussed. The different techniques of case based reasoning are discussed. Cuckoo search definition along with its generic algorithm is shown in the next portion of this chapter. Cuckoo search is integrated with other swarm intelligence techniques in recent years. It's modelling and its efficiency in such environment is shown.

Section 3: This section gives brief introduction to remote sensing. And then studies the basic introduction to image classification. Then it describes all the techniques that had been used till date in image classification. Both traditional and soft computing methods are described. The remaining part deals with membrane computing introduction. This is the technique that we have applied in image classification. It is a new addition to this image classification technique.

Section 4: This section provides the proposed algorithm of Cuckoo search in groundwater detection possibility. Some groundwater features are described in detail and then the proposed architecture is given. How cuckoo search is used in case based reasoning is also described.

Section 5: This section gives the proposed framework of membrane computing for image classification from multi-spectral satellite image. It describes the membrane computing algorithm in respect to feature extraction in detail along with some definition. It also presents the overall mathematical formulation of the proposed algorithm.

Section 6: The first section provides the result of the case study of groundwater detection. It even shows some layout of our software that is developed for groundwater potentiality in some region and our software has shown correct results in almost all the cases as provided by users. This second section presents the result of the case study which is based on the experiments performed at DTRL Lab-DRDO for the classification of satellite image of Alwar area in Rajasthan and even in Saharanpur region in Uttar Pradesh, using our proposed algorithm. Also it is compared with other soft computing technique and it has been shown that membrane computing is at par with other technique.

Section 7: This section gives the details of publication from the thesis and the conference details.

Section 8: In this section the conclusion of the thesis work and the future scope of the work are presented.

Section 9: This section gives the reference details of the thesis.

Appendix A: Abbreviations used.

Appendix B: Introduction to MATLAB software.

Appendix C: Introduction to ERDAS.

2 THE OMNIPRESENCE OF CASE BASED REASONING AND CUCKOO SEARCH

“When you strike at a king, you must kill him”

Ralph Waldo Emerson

Preview

This first portion of this chapter describe the concepts, components and features of CBR. CBR is authored by prominent information scientists, Professors S.K.Pal and S.Shiu. The second portion of this chapter gives a detailed introduction to cuckoo search along with its performance analysis.

2.1 Case Based Reasoning (CBR) :

The classic definition of CBR was coined by Riesbeck and Schank [18]:

“A case-based reasnor solves problems by using or adapting solutions to old problems”

Thus case based reasoning is technique which solves problem case based on the cases stored in the case base. This technique matches the problem case with case base and find out a solution.

2.2 Components and Features of Case-Based Reasoning

The structure of a case-based reasoning system is usually devised in a manner that reflects separate stages: retrieval and adaptation. However at the highest level CBR can be viewed as black boxes as depicted in figure 2.1 that incorporates the reasoning mechanism and the following external facets:

- The input specification or problem case.
- The output that defines a suggested solution to the problem.
- The memory of past cases, the case base, that are referred by the reasoning method.

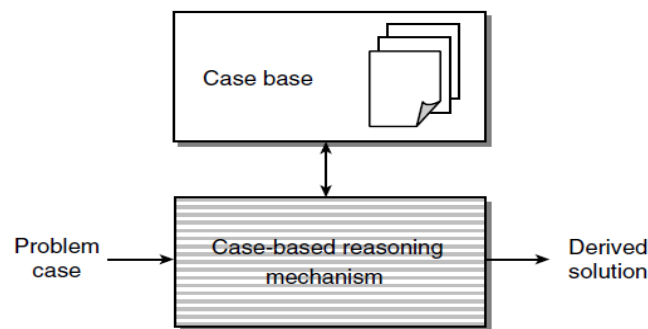


Figure 2.1 CBR system

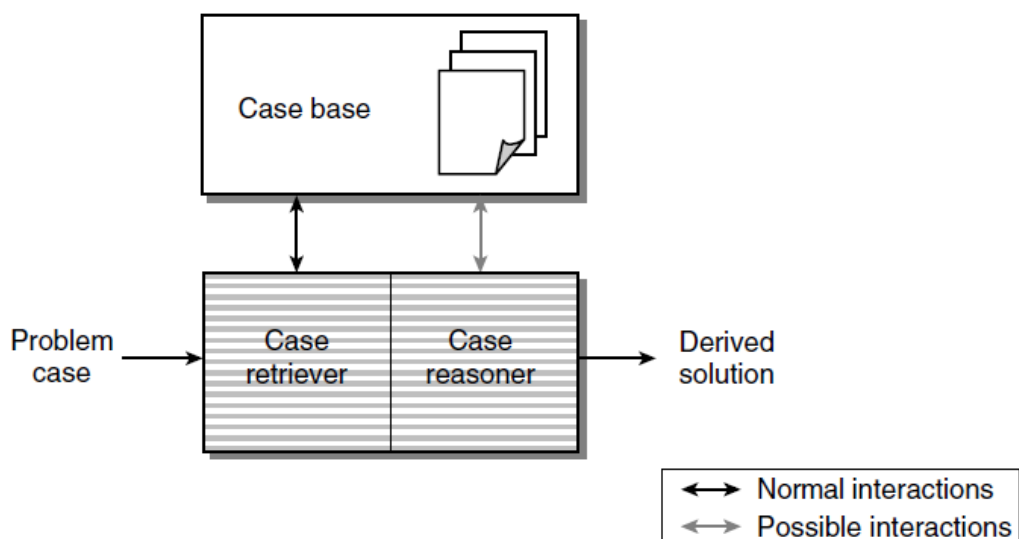


Figure 2.2 Two major Components of CBR

In most CBR systems, the case-based reasoning mechanism, alternatively referred to as the problem solver or reasoner, has an internal structure divided into two major parts: the case retriever and case reasoner. The case retriever's task is to find the appropriate cases in the case base while the case reasoner uses the case retrieved to find a solution to the problem description given. This reasoning process generally involves both determining the differences between the cases retrieved and the current case, and modifying the solution to reflect these differences appropriately. The reasoning process may or may not involve retrieving additional cases or portion of cases from the case base.

2.3 CBR Life Cycle

The problem-solving life cycle in a CBR system consists essentially of the following four parts:

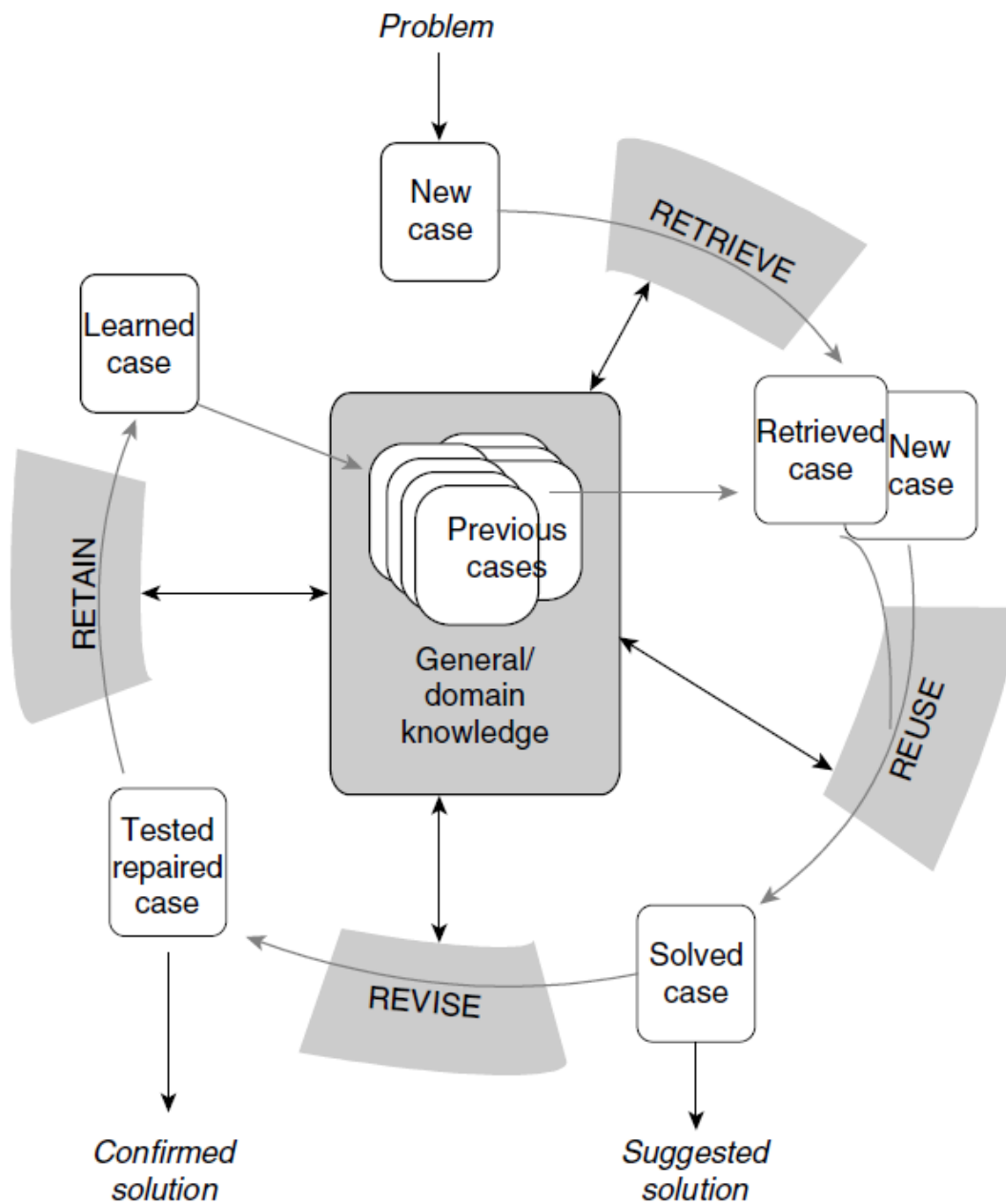


Figure 2.3 CBR cycle

- **Retrieving** similar previously experienced cases (e.g., problem-solution-outcome triples) whose problem is judged to be similar.
- **Reusing** the cases by copying or integrating the solutions from the cases retrieved.
- **Revising** or adapting the solution(s) retrieved in an attempt to solve the new problem.
- **Retaining** the new solution once it has been confirmed or validated.

In many practical applications, the reuse and revise stages are sometimes difficult to distinguish, and several researchers use a single adaptation stage that replaces and combines them. However, adaptation in CBR systems is still an open question because it is a complicated process that tries to manipulate case solutions. Usually, this requires the development of a causal model between the problem space (i.e., the problem specification) and the solution space (i.e., the solution features) of the related cases.

2.4 Applications of CBR

CBR solves real-world problems. Thus this section provides an overview how CBR solves real life problems using four different techniques.

2.4.1 CBR using nearest neighbour

Nearest neighbour techniques are perhaps the most widely used technology in CBR since it is provided by the majority of CBR tools [19]. Nearest neighbour algorithms all work in a similar fashion. The similarity of the problem (target) case to a case in the case-library for each case attribute is determined. This measure may be multiplied by a weighting factor. Then the sum of the similarity of all attributes is calculated to provide a measure of the similarity of that case in the library to the target case. This can be represented by the equation:

$$\text{Similarity (T, S)} = \sum_{i=1}^n f(T_i, S_i) \times w_i$$

where T is the target case; S the source case; n the number of attributes in each case; i an individual attribute from 1 to n ; f a similarity function for attribute i in cases T and S ; and w the importance weighting of attribute i . This calculation is repeated for every case in the case library to rank cases by similarity to the target. Algorithms similar to this are used by most CBR tools to perform nearest neighbour retrieval. Similarities are usually normalised to fall within a range of zero to one (where zero is totally dissimilar and one is an exact match) or as a percentage similarity where 100% is an exact match. The use of nearest neighbour is well illustrated by the Wayland system [20].

2.4.2 CBR using induction

Induction techniques are commonly used in CBR since many of the more powerful commercially available CBR tools provide this facility (e.g. Kate from AcknoSoft; ReCall from ISoft; CBR-Works from TecInno; and ReMind from Cognitive Systems) [19]. Induction algorithms, such as ID3, build decision trees from case histories. The induction algorithms identify patterns amongst cases and partition the cases into clusters. Each cluster contains cases that are similar. A requirement of induction is that one target case feature is defined (i.e. the feature that the algorithm will induce). Essentially the induction algorithms are being used as classifiers to cluster similar cases together. It is assumed (usually correctly) that cases with similar problem descriptions will refer to similar problems and hence similar solutions. It is used in Troubleshooting CFM 56-3 engines on Boeing 737s [21]

2.4.3 CBR using fuzzy logic

Fuzzy logics are a way of formalising the symbolic processing of fuzzy linguistic terms, such as *excellent*, *good*, *fair* and *poor*, which are associated with differences in an attribute describing a feature [22]. Any number of linguistic terms can be used. Fuzzy logics intrinsically represent notions of similarity, since good is closer (more similar) to excellent than it is to poor. For CBR, a fuzzy preference function can be used to calculate the similarity

of a single attribute of a case with the corresponding attribute of the target. For example, in Fig. 2.4, a difference of 1 unit in the values of an attribute would be considered excellent, a difference of 2 would be good, 3 would be fair and 4 would be poor. The result of using fuzzy preference functions is a vector, called the fuzzy preference vector. The vector contains a fuzzy preference value for each attribute. The values in this vector can be combined, through weighted aggregation, to produce a robust similarity value. The use of fuzzy preference functions allows for smooth changes in the result when an attribute is changed unlike the large changes that are possible when step functions are used. A fuzzy preference function is used to transform a quantifiable value for each attribute into a qualitative description of the attribute that can be compared with the qualitative description of other attributes.

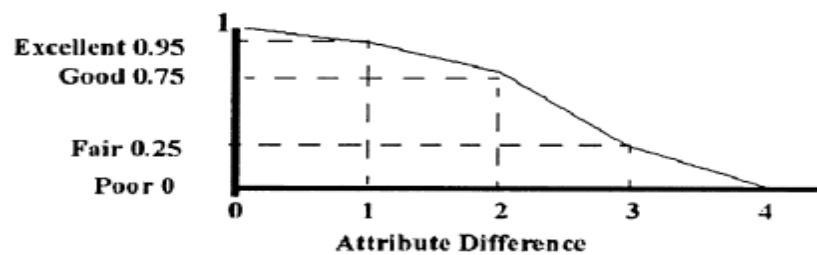


Figure 2.4 A fuzzy preference function [23]

Thus, a fuzzy preference function allows a comparison of properties that are based on entirely different scales such as cost measured in cents per pound and spectral curve match measured in reflection units.

2.4.4 CBR using database technology

At its simplest form, CBR could be implemented using database technology. Databases are efficient means of storing and retrieving large volumes of data. If problem descriptions could make well-formed queries it would be straightforward to retrieve cases with matching descriptions. The problem with using database technology for CBR is that databases retrieve using exact matches to the queries. This is commonly augmented by using wild cards, such as

“WESTp” matching on “WESTMINSTER” and “WESTON” or by specifying ranges such as “ < 1965”. The use of wildcards, Boolean terms and other operators within queries may make a query more general, and thus more likely to retrieve a suitable case, but it is not a measure of similarity. However, by augmenting a database with explicit knowledge of the relationship between concepts in a problem domain, it is possible to use SQL queries and measure similarity [23].

2.5 Cuckoo Search

Cuckoos are different kind of birds because they have aggressive reproduction strategy [7]. Cuckoo lay their eggs in communal nests, and they remove others’ eggs to increase the hatching probability of their own eggs. It is a metaheuristic algorithm. Various other metaheuristic algorithms exists i.e. PSO, GA. Two crucial characteristics of the modern metaheuristics: intensification and diversification [24]. Intensification intends to search around the current best solutions and select the best candidates or solutions, while diversification makes sure the algorithm can explore the search space efficiently.

To cover the efficiency of metaheuristic algorithm Cuckoo search was developed [7]. It outperforms the already existing soft computing algorithms. Even it has shown good results in design optimization problem.

2.5.1 Cuckoo’s behaviour

Cuckoos are fascinating birds, not only because of the beautiful sounds they can make, but also because of their aggressive reproduction strategy. Some species such as the *ani* and *Guira* cuckoos lay their eggs in communal nests, though they may remove others’ eggs to increase the hatching probability of their own eggs. Quite a number of species engage the obligate brood parasitism by laying their eggs in the nests of other host birds (often other species). There are three basic types of brood parasitism:

- intraspecific brood parasitism,
- cooperative breeding, and
- nest takeover.

Some host birds can engage direct conflict with the intruding cuckoos. If a host bird discovers the eggs are not its own, it will either throw these alien eggs away or simply abandons its nest and builds a new nest elsewhere. Some cuckoo species such as the New World brood-parasitic *Tapera* have evolved in such a way that female parasitic cuckoos are often very specialized in the mimicry in colour and pattern of the eggs of a few chosen host species. This reduces the probability of their eggs being abandoned and thus increases their reproductivity. Furthermore, the timing of egg-laying of some species is also amazing. Parasitic cuckoos often choose a nest where the host bird just laid its own eggs. In general, the cuckoo eggs hatch slightly earlier than their host eggs. Once the first cuckoo chick is hatched, the first instinct action it will take is to evict the host eggs by blindly propelling the eggs out of the nest, which increases the cuckoo chick's share of food provided by its host bird.

2.5.2 Rules in Cuckoo Search

In cuckoo search three idealized rules have been formalized.

1. Each cuckoo lays one egg at a time, and dump its egg in randomly chosen nest;
2. The best nests with high quality of eggs will carry over to the next generation;
3. The number of available host nests is fixed and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

The fraction p_a of the n nests being replaced by new nests. For a maximization problem, the quality or fitness of a solution can simply be proportional to the objective function. Based on

the above three rules, a general pseudo code of Cuckoo search had been given by **Yang and Deb**.

1. Objective function $f(x)$, $x = (x_1, \dots, x_d)^T$.
2. Initial a population of n host nests x_i ($i = 1, 2, \dots, n$).
3. While ($t < \text{MaxGeneration}$) or (stop criterion);
 - i. Get a cuckoo (say i) randomly by Lévy flights;
 - ii. Evaluate its quality/fitness F_i ;
 - iii. Choose a nest among n (say j) randomly;
 - iv. If ($F_i > F_j$)
Replace j by the new solution;
End
 - v. Abandon a fraction (p_a) of worse nests
 - vi. Keep the best solutions
 - vii. Rank the solutions and find the current best;
- End while
4. Postprocess results and visualisation.

2.5.3 Differences of Cuckoo search with PSO and GA

Though Cuckoo search has some similarities with hill-climbing in combination with some large scale randomization but there are some significant differences.

- CS is a population based algorithm as similar to GA and PSO but the selection mechanism is different and it is more similar to harmony search.
- The randomization is more efficient as the step length is heavy-tailed and any large step is possible.

- The number of parameters to be used is less than GA and PSO and thus it is more general and it can be adapted to wider class of optimization problems. CS can even be extended to meta-population algorithm.

2.5.4 Performance Analysis on Michaelwicz function

Yang and Deb validated their algorithm in Michaelwicz function. The function looks like:

$$F(x, y) = -\sin(x) \sin^{2m} \left(\frac{x^2}{\Pi} \right) - \sin(y) \sin^{2m} \left(\frac{2y^2}{\Pi} \right)$$

Where $m = 10$ and $(x, y) \in [0, 5] \times [0, 5]$. This function has a global minimum $f_* \approx -1.8013$ at $(2.20319, 1.57049)$. The landscape of this function is shown in figure 2.5. The global optimum has been found by cuckoo search and the results are shown in figure 2.6 where the final locations of the nests are also marked \diamond within the figure. Here $n = 15$ nests, $\alpha = 1$ and $p_a = 0.25$. In the experiment $n = 15$ to 50.

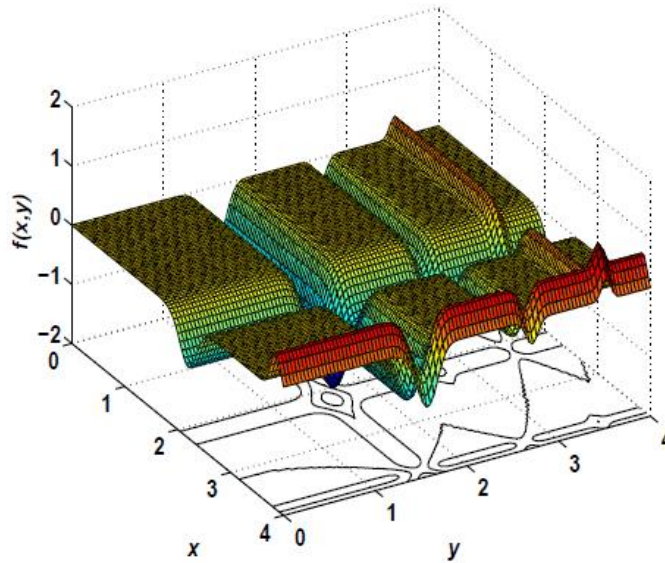


Figure 2.5 The landscape of Michaelwicz's function

From the figure 2.6, the optimum is approaching, most nests aggregate towards the global optimum. We also notice that the nests are also distributed at different (local) optima in the case of multimodal functions. This means that CS can find all the optima simultaneously if the number of nests is much higher than the number of local optima. This advantage may

become more significant when dealing with multimodal and multiobjective optimization problems.

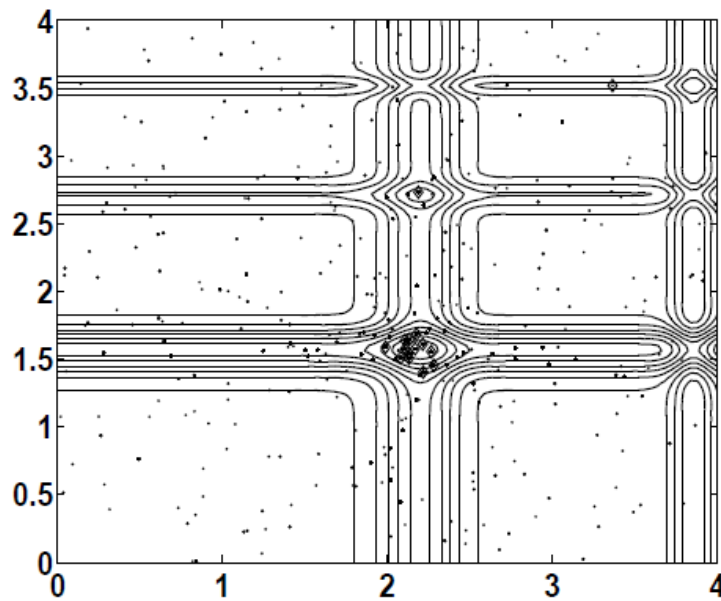


Figure 2.6 Search paths of nests using Cuckoo search. The final locations of the nests are marked with ◊ in the figure

2.5.4.1 Comparison of CS with PSO and GA

To carry out the comparison some benchmark functions have been used. The benchmark functions are used to test the performance of optimization algorithms. The following test functions have been taken to carry out the comparative study:

- De Jong's first function is a sphere function

$$F(x) = \sum_{i=1}^d x_i^2 \quad x_i \in [-5.12, 5.12]$$

Whose global minimum $f^* = 0$ occurs at $x^* = (0, 0 \dots 0)$. Here d is the dimension.

- Easom's test function is unimodal

$$F(x, y) = -\cos(x)\cos(y)\exp[-(x-\Pi)^2 - (y-\Pi)^2],$$

Where

$$(x, y) \in [-100, 100] \times [-100, 100].$$

It has a global minimum of $f^* = -1$ at (Π, Π) in a very small region.

- Shubert's bivariate function

$$F(x, y) = -\sum_{i=1}^5 i \cos [(i + 1)x + 1] \sum_{i=1}^5 \cos[(i + 1)y + 1],$$

Has 18 global minima in the region $(x, y) \in [-10, 10] \times [-10, 10]$. The value of global minima is $f_* = -186.7309$

- Griewangk's test function has many local minima

$$F(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

But a single global minimum $f_* = 0$ at $(0, 0, \dots, 0)$ for all $-600 \leq x_i \leq 600$ where $i = 1, 2, \dots, d$.

- The generalized Rosenbrock's function is given as ,

$$F(x) = \sum_{i=1}^{d-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2]$$

Which has a minimum $f(x_*) = 0$ at $x_* = (1, 1, \dots, 1)$.

- Schwefel's test function is multimodal

$$F(x) = \sum_{i=1}^d [-x_i \sin(\sqrt{|x_i|})], \quad -500 \leq x_i \leq 500,$$

With a global minimum of $f_* = -418.9829d$ at $x_i^* = 420.9687$ ($i = 1, 2, \dots, d$).

- Rastrigin's test function

$$F(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$$

A global minimum $f_* = 0$ at $(0, 0, \dots, 0)$ in a hypercube $-5.12 \leq x_i \leq 5.12$ where $i = 1, 2, \dots, d$.

Table 2.1 Comparison of CS with PSO and GA

FUNCTION/ ALGORITHM	GENETIC ALGORITHM	PARTICLE SWARM OPTIMIZATION	CUCKOO SEARCH
De Jong (d = 256)	25412±1237 (100%)	17040±1123(100%)	4971±754(100%)
Eastom's	19239±3307(92%)	17273±2929(100%)	6751±1902(100%)
Shubert's (18 minima)	54077±4997(89%)	23992±3755(92%)	9770±3592(100%)
Rosenbrock's (d = 16)	55723±8901(90%)	32756±5325(98%)	5923±1937(100%)
Schwefel's (d = 128)	227329±7572(95%)	14522±1275(97%)	8829±625(100%)
Rastrigin's	110523±5199(77%)	79491±3715(90%)	10354±3755(100%)
Griewank's	70925±7652(90%)	55970±4223(92%)	10912±4050(100%)

We can see that the CS is much more efficient in finding the global optima with higher success rates. Each function evaluation is virtually instantaneous on modern personal computer. For example, the computing time for 10,000 evaluations on a 3GHz desktop is about 5 seconds. For all the test functions, CS has outperformed both GA and PSO. The primary reasons are two folds: A fine balance of randomization and intensification, and less number of control parameters. As for any metaheuristic algorithm, a good balance of intensive local search strategy and an efficient exploration of the whole search space will usually lead to a more efficient algorithm. On the other hand, there are only two parameters in this algorithm, the population size n , and p_a . Once n is fixed, p_a essentially controls the elitism and the balance of the randomization and local search. Few parameters make an algorithm less complex and thus potentially more generic.

3 *APPLYING IMAGE CLASSIFICATION IN REMOTE SENSING*

“We opened every potential flood gate we could; we were at least expecting a lot of rain”

David Lindsay

Preview

It starts with a brief introduction to remote sensing and then describes image classification. This chapter mainly focuses on different techniques used so far in image classification along with the introduction to a technique that had been adapted for our research work. Mainly all traditional and soft computing techniques are described. Then an introduction to membrane computing is described.

3.1 Introduction

Remote Sensing is the science of obtaining and interpreting information from a distance, using sensors that are not in physical contact with the object being observed [25]. It is used for the knowledge elicitation of earth’s surface and atmosphere obtained from **Earth Observation Satellite (EOS)**. A good amount of vagueness, uncertainty and ambiguity comes into existence when categorizing geographic objects based on the geospatial information thus received.

3.2 Image Classification

Image classification is formally defined as the process where a received pattern/signal is assigned to one of a prescribed number of classes. The main objective of image classification procedures is to categorize all pixels in an image into land cover features. The image that generally obtained from remote sensors is multi-spectral images and the spectral pattern present within the data for each pixel is used as the numerical basis for categorize spectral pattern refers to the set of radiance measurements obtained in various wavelength bands of each pixel.

Spectral pattern reorganization refers to the family of classification procedures that utilizes spectral information of pixels as the basis for automated land cover classification. Commonly used techniques like Rough set, Rough-Fuzzy theory, artificial neural network etc. are used to classify images. Some of the techniques are described below.

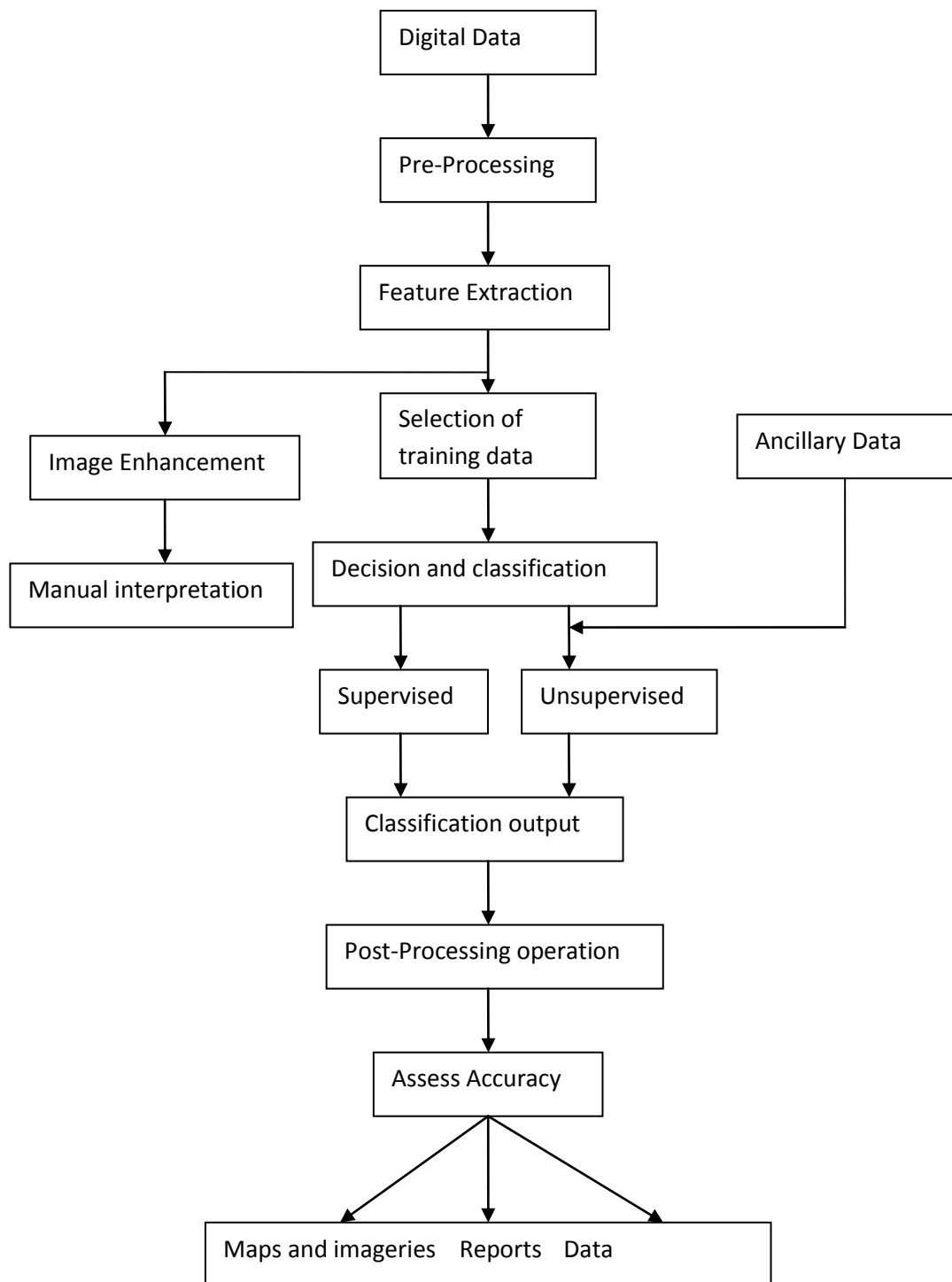


Figure 3.1 Image Classification Process

3.3 Traditional Image Classification Methods

Some of the traditional classical approaches like Parallelopiped Classification, Minimum Distance to Mean Classification, Maximum Likelihood Classification etc.

3.3.1 Parallelopiped Classification

The Parallelopiped classifier is a supervised classifier that uses intervals of pixels values of bounded region of an area to determine whether a pixel belongs to a class or not. The values of the pixels of samples for the class give the intervals' bounding points. It is supervised classifier and thus it uses signature creation and classification for image classification.

Although parallelopiped classification is an accurate classification technique, it is not widely used because it has several disadvantages. The most important disadvantage is that it can leave many unclassified pixels. Another disadvantage of this classification method is that it can have overlap between training pixels.

3.3.2 Minimum Distance to Mean Classification

In this classification first of all mean or average or spectral in each band for each category is determined. For each category there is a mean vector. Those pixels which are unidentified can be classified by computing the distance between the value of unknown pixel and each of category mean. The unknown pixel is assigned to the class to which it closely belongs to.

The minimum-distance-to-means strategy is mathematically simple and computationally efficient, but it has certain limitations. Most importantly, it is insensitive to different degrees of variance in the spectral response data. However this classifier is not widely used in applications where spectral classes are close to one another in the measurement space and have high variance.

3.3.3 Gaussian Maximum Likelihood Classification

The maximum likelihood classifier quantitatively evaluates both the variance and covariance of the category spectral response patterns when classifying an unknown pixel. To do this, an assumption is made that the distribution of the cloud of points forming the category training data is Gaussian (normally distributed). This assumption of normality is generally reasonable for common spectral response distributions. In the Dimensional graph the vertical axis is associated with the probability of a pixel value being a member of one of the classes. The resulting bell-shaped surfaces are called probability density functions, and there is one such function for each spectral category. The probability density functions are used to classify an unidentified pixel by computing the probability of the pixel value belonging to each category.

The major disadvantage of this technique is that the maximum likelihood decision rule is based on the probability that a pixel belongs to a particular class. The basic equation assumes that these probabilities are equal for all classes, and that the input bands have normal distributions. The maximum likelihood algorithm assumes that the histograms of the bands of data have normal distributions, however this not true and hence not applicable in all applications.

3.4 Soft Computing Techniques

Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty and partial truth. In effect, the role model for soft computing is the human mind. The guiding principle of soft computing is: Exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost. The principal constituents of soft computing are fuzzy logic [13], rough set theory [14], neural network theory and probabilistic reasoning, and Swarm Intelligence [15]

with the latter subsuming belief networks, genetic algorithms, chaos theory and parts of learning theory [16].

3.4.1 Rough Set Theory

The rough set philosophy is founded on the assumption that with every object of the universe of discourse we associate some information (data, knowledge) [14]. 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).

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.

3.4.2 Semantic Web Based Classification (Ontology):

The ontology design process is provided by Gupta 2008 [26] [27] has mainly the following step:

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

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, then also we can have built ontology for it by connecting database with protégé. The figure 3.2 describes the necessary steps of this classifier.

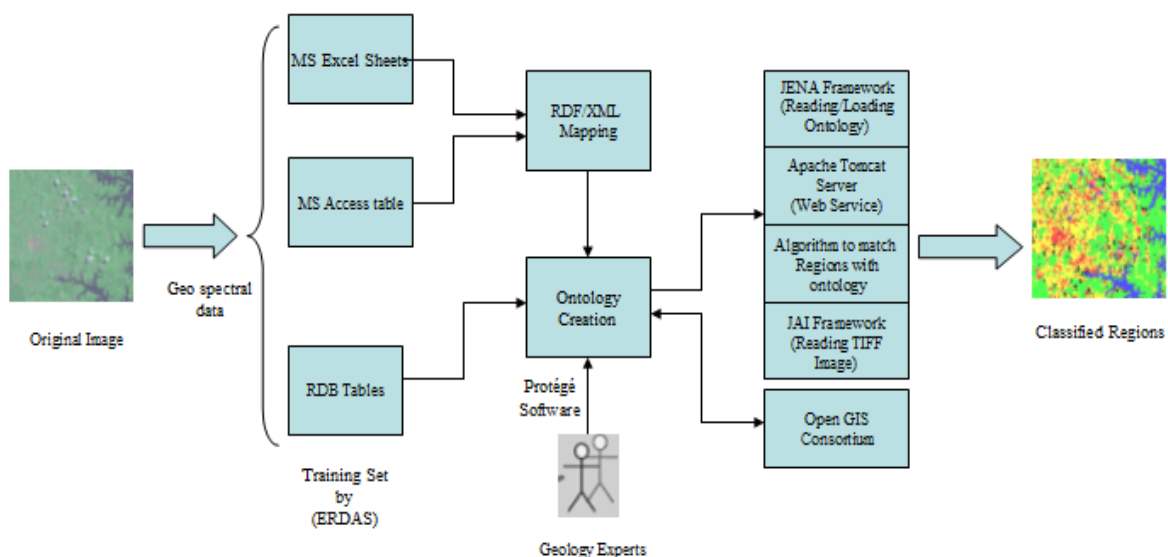


Fig 3.2 Framework for Semantic Classifier

3.4.3 Swarm Intelligence Technique:

Swarm intelligence, also referred as collective intelligence, is defined as the problem solving behavior that emerges from the interaction of such a collection of individual agents, [28]. These interactions contribute to distributive collective problem solving.

3.4.3.1 Hybrid ACO/PSO:

This swarm intelligence technique has been proposed by Gupta 2009 [29]. Basically work on Ant Colony Optimization for image classification has been proposed by Omkar [16]. The technique uses Swarm Intelligence. It uses cAnt miner and modified hybrid PSO/ACO is used for rule generation. There are training set which is fed as input to rule generation. The training set generated from original unclassified image can be in any tabular format as excel-sheet or access sheet or simple text file. Before this the training set is done unified by changing it to the Attribute Relation File Format. The cAnt miner algorithm is applied by using the MYRA tool and the Modified Hybrid PSO/ACO algorithm is applied by using the poen source tool for Hybrid PSO/ACO. These algorithms provide rule set for the different regions. Then the regions are classified according to different rules.

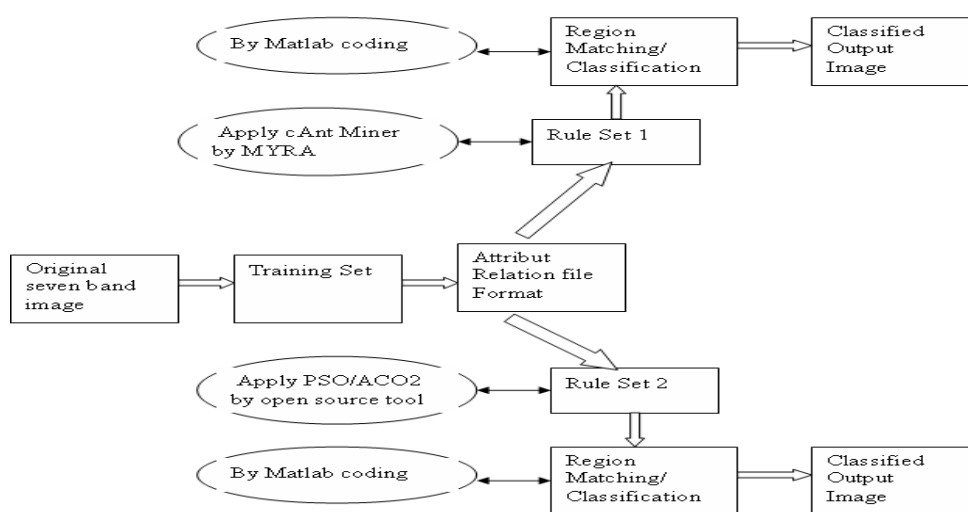


Figure 3.3 Diagrammatic representation of Swarm Intelligence Classifier

3.4.3.2 Biogeography Based Optimization

In the year 2010 Gupta [17] proposed a hybrid method of BBO and ACO/PSO. The previous swarm intelligence technique described ACO/PSO method in image classification. This section only describes how BBO is applied to image classification. Basically BBO has mainly 2 steps for classification: Selective cluster BBO Classification Applier layer and Feature extraction analysis layer. The main steps are described below:

BBO Classifier Layer:

(i) Selective Cluster BBO Classification Applier Layer: Hence, we apply BBO Technique on those clusters of the satellite image which show the maximum classification efficiency which is due to the fact that these are the clusters which predominantly show the presence of the feature that is most efficiently classified by the BBO Algorithm. In our illustration, BBO will be able to extract the water pixels most efficiently since we choose the bands namely NIR and MIR for partitioning. And therefore, we apply BBO on the kth cluster of the satellite image since this is the cluster which gives the maximum classification efficiency because it predominantly shows the presence of water body in the image.

(ii) BBO Feature Extraction Inefficiency Analysis Layer: However, it is also observed that BBO shows a wide range of efficiencies in classifying the rest of the features. In other words, BBO is not able to classify all the features with good classification efficiency and shows poor performance on some of the remaining features. In fact, in our illustration, it shows the poorest performance in classifying the urban pixels. Hence a proceed towards ACO2/PSO classification to improve the image classified by BBO.

3.4.4 Recently Used Technique: Membrane Computing

Basically our research work had applied the following technique in image classification. Membrane computing is a currently used method and it is used because the above mentioned technique never taken heterogeneity factor into consideration. Thus we have applied this technique in image classification. The following section starts with an introduction the membrane computing and in later chapter we have described how MC is used in image classification.

Based on many years of research on DNA computing and inspired by biological cells, Gheorghe Păun [5] proposed membrane computing model in 1998. The research area of membrane computing originated as an attempt to formulate a model of computation motivated by the structure and functioning of a living cell— more specifically, by the role of membranes in compartmentalization of living cells into “protected reactors”. Therefore, initial models were based on a cell-like (hence hierarchical) arrangement of *membranes* delimiting compartments where *multisets* of chemicals (called *objects*) evolve according to given *evolution rules*. These rules were either modelling chemical reactions and had the form of (multiset) rewriting rules, or they were inspired by other biological processes, such as passing objects through membranes (either in symport or antiport fashion), and had the form of communication rules. All classes of computing devices considered in membrane computing are now generically called *P systems*. Application areas include biology and bio-medicine as well as optimization, economics, and computer science.

3.4.4.1 Membranous Structure

The central notion of membrane computing is the notion of *membrane* understood as a three-dimensional vesicle, which geometrically can be considered as a ball in the Euclidean space. Thus a membrane delimits a space, separating in this way “inside” from “outside”. The

(inside) space delimited by a membrane serves as a “protected reactor”—a space where specific reactions take place, using molecules in this space.

Membrane m' is *included* in membrane m then m is called the *upper neighbour* of m' , and m' called a *lower neighbour* of m . A membrane m with no lower neighbours' is called *elementary*, and a membrane m with no upper neighbour is called a *skin membrane*. The number of membranes is called the *degree* of a membrane structure. A *region* is either a space delimited by an elementary membrane or a space delimited by a non-elementary membrane and its entire lower neighbour's. We use membrane labels to label regions in an obvious way: regions inherit labels of their (upper) delimiting membranes. Finally, we say that different membranes m and m' are *sibling* membranes if there is a membrane m'' which is the upper neighbour for both m and m' .

Graphically, we will represent membrane structures in the two-dimensional plane using Euler-Venn diagrams representing the inclusion relationships between the regions delimited by membranes. Figure 3.4, a sort of logo of membrane computing, is an example of such a representation. It also illustrates some of the basic terminology introduced above. Note, e.g. that region 8 is elementary and it is a lower neighbour of region 6, which is the upper neighbour for both region 8 and region 9, which therefore are siblings. Also, regions 2, 3, and 4 are siblings, and so are regions 5, 6, and 7.

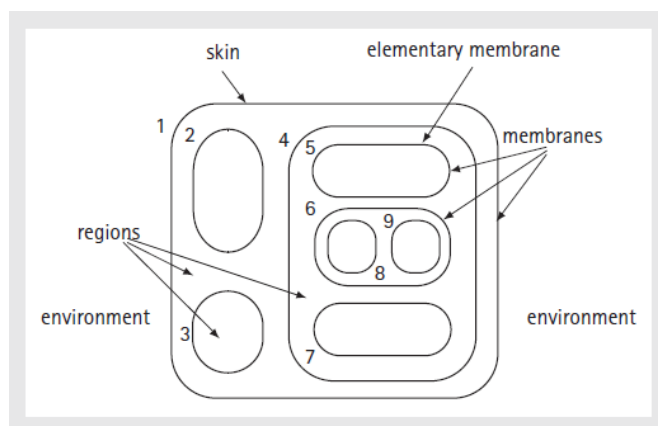


Figure 3.4 A membrane structure

For a given membrane structure S , let l_1, \dots, l_n be the labels of membranes in S , where l_1 is the label of the skin membrane. The membrane labeled by l_i , $1 \leq i \leq n$, is represented by an ordered pair of parentheses $[]_{l_i}$, and a parentheses expression for μ is constructed as follows:

1. $[]_{l_1}$ is a *correct form* for S .
2. If $z = z_1 []_{l_i} z_2$ is a correct form for S , for some $i \in \{1, 2, \dots, n\}$, and membranes labelled by $k_1, \dots, k_p \in \{2, \dots, n\}$ are all lower neighbours of the membrane labelled by l_i , then for each string $x = []_{k_{j_1}} []_{k_{j_2}} \dots []_{k_{j_p}}$, where j_1, \dots, j_p is a permutation of $1, 2, \dots, p$, $z_1 [x]_{l_i} z_2$ also a correct form for S .
3. A *parentheses expression* for μ is a correct form for S which includes the right hand parenthesis $]_{l_i}$ for each $i \in \{1, 2, \dots, n\}$.

Clearly, a parentheses expression μ for a membrane structure S contains all the information about the inclusion relationships between the regions delimited by membranes of M , and hence all the neighbourhood information about M (regions, upper and lower neighbours, etc.). Thus, e.g. for the membrane structure S the parenthesis expression is given below from Fig. 3.4,

$$[[]_2 [[]_7 []_5 [[]_8 []_9]_6]_4 []_3]_1$$

A mathematically natural way to formalize membrane structures is to use rooted unordered node-labelled trees. For example, the membrane structure from Fig. 3.4 corresponds to the tree given in Fig. 3.5.

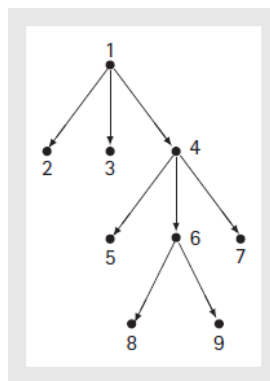


Figure 3.5 A tree describing the membrane structure from Figure 3.4

3.4.4.2 The formation of Rules

For a rule $u \rightarrow v$ we say that $|u|$ (the weight of the multiset on the left hand side) is the *weight* of the rule; a rule with the weight at least two is said to be *cooperative*. A rule of weight 1 is called *noncooperative*. *Catalytic rules* are cooperative rules of the form $ca \rightarrow cv$, where c, a are objects with $c \neq a$, v a multiset, and c does not appear in v (c only assists a in evolving into v , but c itself is not undergoing any transformation).

The rules are of the form $u \rightarrow v$, where u is a string over O , and v is a string over $O \times \{here, out, in\}$. Thus, each element of v is of the form (a, tar) , where a is an object from O and tar is either *here* or *out* or *in*. The target indications indicate as follows.

If $tar = here$, then a remains in the region where the rule resides.

If $tar = out$, then a is sent to the upper neighbour of the region where the rule resides (recall that the upper neighbour of the skin region is the environment).

If $tar = in$, then a is sent non-deterministically to one of the lower neighbours.

The application of a single rule proceeds as follows. First of all, we assume the existence of a global clock that measures time by marking time units. When a rule $u \rightarrow v$ residing in region h is used, it acts only on objects placed in h (i.e. objects specified in u are available in h in required multiplicities). Then the following happens within one time unit.

1. Objects present in u in the specified multiplicities are removed from h .
2. Objects present in v in the specified multiplicities are produced.
3. The produced objects are placed either in h or in neighbours of h according to the target indications in v .

Hence, e.g. if h is not elementary, the rule $a^2bc^3 \rightarrow ba^2c(da, out)(ca, in)$ is applicable if h contains at least two objects a , one object b , and three objects c . The meaning of this rule:

1. Removes from h two objects a , one object b , and three objects c ;
2. produces one object b , four objects a , two objects c , and one object d ;

3. the produced object b remains in h , two produced objects a remain in h , one produced object c remains in h , the produced object d and one produced object a are sent to the upper neighbour, one produced object a is sent to a (non-deterministically chosen) lower neighbour, and one produced object c is sent to a (non-deterministically chosen) lower neighbour.

3.4.4.3 Maximizing Parallelism

The maximal parallelism means that rules should be used in parallel to the maximum degree possible. For example, assume that currently the multiset of objects present in region h is $w = a^3 b^2 c^2$ and the set of rules R residing in h is $R = \{r_1, \dots, r_5\}$, with

$$r_1 : ab \rightarrow v_1, r_2 : c \rightarrow v_2, r_3 : bc \rightarrow v_3, r_4 : a^3 c^2 \rightarrow v_4, r_5 : ad \rightarrow v_5,$$

for some v_1, \dots, v_5 .

Note that applying r_1 removes (“consumes”) the multiset ab , while two parallel applications of r_1 remove the multiset $a^2 b^2$. The remaining “still available” multiset ac^2 does not allow one more parallel application of r_1 , but it allows for a parallel application of r_2 , it even allows two parallel applications of r_2 which remove the multiset c^2 . The remaining multiset a does not allow an application of any rule at all, and therefore we say that the parallel application of the *multiset of rules* $\{(r_1, 2), (r_2, 2)\}$ is maximally parallel. It has been checked that application of $\{(r_3, 2)\}$ is maximally parallel (because no rule can be applied to the remaining multiset of objects a^3), and the application of $\{(r_1, 1), (r_2, 1), (r_3, 1)\}$ is maximally parallel (because no rule can be applied to the remaining multiset of objects a^2). On the other hand, the application of $\{(r_1, 2), (r_2, 1)\}$ is not maximally parallel, because r_2 can be still applied to the remaining multiset of objects ac .

3.4.4.4 The formal definition of Membrane Computing

Thus, a P system (with multiset rewriting rules) of degree $m \geq 1$ is a construct

$$\Pi = (O, H, \mu, w_1, \dots, w_m, R_1, \dots, R_m, i_0),$$

where:

1. O is the alphabet of objects;
2. H is the alphabet of membrane labels;
3. μ is a membrane structure of degree m ;
4. $w_1, \dots, w_m \in O^*$ are the multisets of objects associated with the m regions of μ ;
5. $R_i, 1 \leq i \leq m$, are the finite sets of multiset rewriting rules associated with the m regions of μ .
6. $i_0 \in H \cup \{e\}$ specifies the input/output region of Π , where e is a reserved symbol not in H .

Now an example of a P system Π with multiset rewriting rules specified as follows:

$$\Pi = (O, \mu, w_1, w_2, R_1, R_2, i_0), \text{ where}$$

$$O = \{a, b, c, d, e, f\},$$

$$\mu = [[]_2]_1,$$

$$w_1 = aa,$$

$$w_2 = d,$$

$$R_1 = \{r_1 : a \rightarrow (a, out), r_2 : a \rightarrow (c, in)\},$$

$$R_2 = \{r_3 : d \rightarrow e, r_4 : d \rightarrow f, r_5 : ce \rightarrow cee, r_6 : cf \rightarrow c(b, out)\}, \text{ and}$$

$$i_0 = 2.$$

3.4.4.5 Performance Analysis of Membrane Computing

The overall quality of a computing platform depends on the extent to which it possesses certain positive attributes, including usability, performance, flexibility, maintainability and scalability. Performance, flexibility and scalability are three of the most important quality attributes for a computing platform for membrane computing applications. In this section, we

define the attributes of performance, flexibility and scalability [30] in the context of a computing platform for membrane computing applications, explain the significance of these attributes, and indicate the connections that exist between them.

3.4.4.5.1 Performance

By the performance of a computing platform for membrane computing applications we mean the speed at which it executes P systems; that is, the amount of useful processing it performs per unit time. A suitable measure of the amount of useful processing performed is the number of reaction rule applications performed. Thus the performance of a computing platform for membrane computing applications can be measured in reaction rule applications per unit time.

3.4.4.5.2 Flexibility

By the flexibility of a computing platform for membrane computing applications we mean the extent to which it can support the execution of a wide range of P systems. Thus a flexible computing platform for membrane computing applications must be able to adapt to the specific properties of the P system to be executed. The greater the flexibility of the computing platform, the greater the diversity among the P systems in the class of P systems that the computing platform is able to execute.

3.4.4.5.3 Scalability

By the scalability of a computing platform for membrane computing applications we mean the extent to which increases in the size of the P system to be executed do not lead to a reduction in the ability of the computing platform to perform its functions or a reduction in the performance of the computing platform. We take the size of a P system as being largely determined by the number of regions and the number of reaction rules it contains.

4 COMBINING CUCKOO SEARCH WITH CASE BASED REASONING FOR DETECTION OF GROUNDWATER POSSIBILITY

“If you don’t think every day is good day, just try missing one”

Cavett Robert

Preview

In this chapter deals with our proposed algorithm for detection of groundwater feasibility in any particular area. Our algorithm has shown almost 99% accuracy in detection mechanism. Our experts had given some areas values and our proposed algorithm detected the presence/absence of groundwater in that area.

4.1 Introduction

Groundwater is located beneath the ground surface in soil spores spaces and in fractures of rock formation. Groundwater is basically soil moisture, permafrost immobile water in very low permeability bedrock and deep geothermal or oil formation water. Groundwater existence depends on natural resources. Existence of various resources is necessary to find the potentiality of groundwater. In recent findings a lot of swarm intelligence techniques have been devised for detection of groundwater. Both Biogeography Based Optimization [11] and Particle Swarm Intelligence [10] had been used in these detection process but there some pitfalls in those algorithms. Cuckoo search is able to remove all those pitfalls and came with almost full efficiency in detection process. Cuckoo search being a metaheuristic algorithm

have a balanced layout in both randomization and intensification. It maintains a good balance of intensive local search and then exploration of full search which leads to find the problem in most optimized way. But in case of BBO and PSO both searching space is narrowed down and thus the searching is concentrated on local optima.

Basically groundwater possibility varies with natural resources. The detection is quite variable thus several conditions are needed to be considered during their detection mechanism. In some cases geology (one of the most important attribute for groundwater detection) has high significance in detection whereas in some cases it has least importance. Thus any combinations are not encountered in previously discussed swarm intelligence technique as they have loopholes in their searching mechanism.

But Cuckoo search discussed almost all possible conditions where possibility of groundwater can be high, low and moderate. Our proposed algorithm combined both the strength of Case based reasoning and even Cuckoo search. Basically for the initial search we have used Case based using nearest neighbour to match the user input data with the stored database. If a high match is found then in the second search we have used another matching pattern and even propositional logic to go through all possible conditions.

Thus proposed work presents a biologically inspired technique called Cuckoo search technique that can be adapted according to the database of expert knowledge for a more focussed groundwater detection possibility. The detector explores all the characteristic of Cuckoo search and thus it is flexible and can be adapted according to the application. The proposed algorithm is an improved version than the previous detector due to the metaheuristic characteristics of Cuckoo search. We have named our classifier as **bio-inspired groundwater potentiality detector**. The proposed algorithm has been applied to a database as given by our geoscience expert. It is a database having all the attributes values along with the possibility of groundwater in those conditions.

4.2 Dataset

The dataset used in our experiment consists of groundwater observations made by domain experts to predict the groundwater possibility. Experts have given us values of attributes of various regions. It is not concentrated to any particular area.

Table 4.1 Six attributes used in dataset

Attributes	Values
Geology	Sedimentary, Younger alluvium, Older alluvium, Igneous, Metamorphic
Landform	Floodplain, Intermontane valley, Pediment, Alluvial fans, Bajada, Pediplain, Buried pediment, Alluvial Plain, Deltaic Plain, Wadi, River terraces, Old meander etc.
Soil	Sandy loam, Sandy gravel, Coarse sand, Clay loam, Alluvial sand, Gravel sand, Gravel Sand Pebbles, Sand, Rocky etc.
Land use	Agricultural land, Forest, Cultivated land, Fallow land, Waterbody, Wasteland, Swampy land, Buildup, Urban, Grass, Shrubs, mixed vegetation etc.
Slope	Gentle, Steep
Lineament	Absent, Present

From the above table it is noticed that generally 6 attributes are considered. Namely they are geology, landform, soil, land use, slope and lineament. Each attribute has their significance in detection. Our expert has ranked the attribute. The ranking denotes which attributes are most important which less is. Rank 1 indicates this attribute existence or value has a great impact on the existence of groundwater in that region whereas rank 6 indicates that this attribute has least significance in groundwater potentiality. Thus the ranking is as follows:

Table 4.2 Ranking of attributes

Rank	Attributes
1	Geology
2	Lineament
3	Landform
4	Slope
5	Landuse
6	Soil

Each of the attributes is described in the following section:

- **Geology:** is the science that comprises the study of the solid Earth and the processes by which it is shaped and changed.
- **Lineament:** A lineament is a linear feature in a landscape which is an expression of an underlying geological structure such as a fault. Typically a lineament will comprise a fault-aligned valley, a series of fault or fold-aligned hills, a straight coastline or indeed a combination of these features. Fracture zones, shear zones and igneous intrusions such as dykes can also give rise to lineaments.
- **Landform:** A landform or physical feature of the terrain. Landform elements also include seascape and oceanic water body interface features such as bays, peninsulas, seas and so forth, including sub-aqueous terrain features such as submersed mountain ranges, volcanoes, and the great ocean basins.
- **Land use:** Land use is the human use of land. Land use is the modification of natural environment or building something new in areas like fields, pastures and settlements.
- **Soil:** Soil contains various mineral constituents. It is a natural body consisting of various layers. Basically their morphological, physical, chemical and mineralogical characteristics vary from their parent materials.
- **Slope:** In geographical term slope is known as gradient. Gradient tells you how steep the route is. One common way to describe the average steepness or gradient between two points is to state the difference in elevation (ED) divided by the horizontal distance (HD):

$$\text{GRADIENT} = \text{ED/HD}$$

4.3 Integration of Case Based Reasoning

In our application CBR has immense importance for retrieval of similar cases from the stored database. In this CBR system, geographical parameters and their corresponding solutions i.e. the possibility of groundwater (High, Moderate and Low) are stored as cases in the case base. By the attribute importance, six attributes are selected as inputs. The possibility of groundwater is selected as the output decision attribute of the case. The cases are collected from expert knowledge. The case is represented by the ordered pair $C = \langle F, D \rangle$ where F is the geographical parameter of case C i.e. $F = \{ \text{Geology, Landform, Soil, Lineament, Slope, Land type} \}$ and D is the decision attribute i.e. the possibility of groundwater in the given region. The case searching and matching is a key step in case retrieving and it directly influences the retrieval efficiency and accuracy. The case retrieval in essence is to find the most similar case in the case base to target case. The nearest neighbour is widely used for its advantage of clear physical concept and simple calculation for case searching. A sum of similarities is calculated according to the similarity between each feature in problem case and the cases in case base. Evaluate the fitness of each nest for case retrieval according to the following equation:

$$\text{Similarity (T, S)} = \sum_{i=1}^n f(T_i, S_i) \times w_i$$

$$\text{if } T_i = S_i$$

$$\text{then } f(T_i, S_i) = 1$$

Otherwise

$$f(T_i, S_i) = 0$$

where Similarity (T, S) represents the similarity degree of problems case (P) and case (C) stored in case base, $i =$ attribute of case, $n =$ number of attributes, $T_i =$ problem case, $S_i =$ cases stored in case base and $w_i =$ weight of attribute i , the more important attributes should be assigned larger weights than less important ones. The larger the weighted sum the more

similar the two cases will be. The view of our case base is shown in the following figure. It contains 7 columns. The first 6 are for attributes and the last column tells about the potentiality of groundwater in such region.

GEOLOGY	LAND_FORM	SOIL	LAND_USE	SLOPE	LINAMENT	SOLUTION
SEDIMENTARY	FLOODPLAIN	SANDYLOAM	AGRICULTURALLAND	GENTLE	ABSENT	HIGH
SEDIMENTARY	INTERMOUNTAINVALLEY	SANDYGRAVEL	FOREST	STEEP	PRESENT	HIGH
SEDIMENTARY	PEDIMENT	COARSESAND	CULTIVATED	GENTLE	PRESENT	MODERATE
SEDIMENTARY	FLOODPLAIN	SANDYLOAM	AGRICULTURALLAND	GENTLE	PRESENT	HIGH
YOUNGERALLUVIUM	FLOODPLAIN	SANDYLOAM	AGRICULTURALLAND	GENTLE	PRESENT	HIGH

Figure 4.1 Cases from the Case Base

4.4 Algorithm for Cuckoo search in Groundwater potentiality

Begin

1. Generate objective function of each stored cases in the case base $f(x)$, $x = \{x_1, x_2, \dots, x_d\}$. // Expert has provided us 140 cases. Thus our application $d=140$.
2. Take problem case input from the user (say i).
3. Initialize Count_case_base to 1
4. Initialize total_cases to 115. //Out of 140 some cases are taken for validation. Thus we have taken a total of 115 for our database.
5. While (count_case_base \leq total_cases)
 - i) Consider the current case (nest) from the case base (say j).
 - ii) Evaluate Similarity(F_i, F_j) = $\sum_{i=1}^n f(F_i, F_j) \times w_i$
 - iii) Store the similarity value of F_j in another database named Temp_case_base
 - iv) Count_case_base = count_case_base + 1

- End while
6. Store the cases in final_Case_base having the maximum values from temp_case_base
 7. Keep all these maximum values (best solutions) // nests with quality solutions.
 8. Ignore_Cases = Total_cases – final_case_base
 9. A fraction (ignore_cases) which are worst nests are abandoned.
 10. Rank the solutions / cases in final_case_base
 - i) Find a correlation between the problem cases and the cases in final_Case_base
 - ii) The maximum value of correlation function is ranked as best solution.
 11. The case with highest correlation value is considered.
 12. Postprocess results
 - i) Apply all possible propositional logic condition on the cases
 - ii) Depending on the condition the outcome can be
High, Moderate or Low.

End

4.4.1 The Detailed working of the algorithm

Step 1: In this step basically we have to find out number of cases need to be considered. Though objective function had to be found out for each case in the case base but the real evaluation is done in step 5(ii). Thus we have considered a total of 115 cases in our case base. The left out cases are needed for validation process.

Step 2: In this step a user input is taken. Users are required to give their problem case to the system. Thus user has to enter the values of 6 attributes. A pop up menu appears for each attribute and in that menu all the attribute values are displayed. (say i)

Step 3: A counter is required for keeping track of cases in the case base. Thus we initialize a counter name count_case_base to 1. This counter basically points to the current case in the case base.

Step 4: In this step we consider the total cases needed for our processing purpose. Thus we initialize a counter named total_case to 115.

Step 5: A while condition is used. This loop is terminated until all the cases in the case base are not encountered.

Step 5.1: Point the counter count_case_base to the first case in the database (say j).

Step 5.2: Evaluate the objective function $\text{Sim}(F_i, F_j) = \sum_{i=1}^n f(F_i, F_j) \times w_i$

Where F_i is the problem case, F_j is the case stored in case base. N = number of attributes, in our application it is 6. The function f returns the 1 if $F_i = F_j$ otherwise 0 is returned and w_i is the weight of the attribute. Figure 4.2 shows the weight of each attribute.

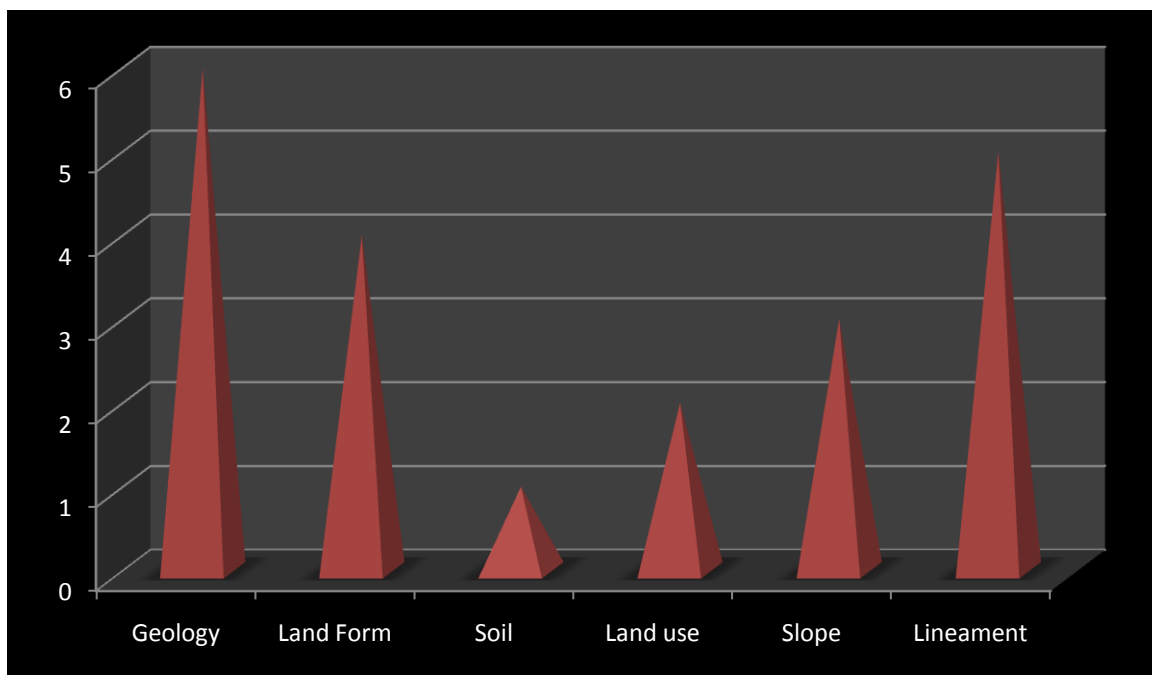


Figure 4.2 Weight of each attribute

Y axis represents the weight whereas X-axis represents the attributes. It shows geology has a weight of 6 whereas soil has weight 1. It means geology has immense significance in groundwater possibility whereas soil has least importance.

In the following figures shows the distribution of attribute values. Figure 4.3 has shown geology values. X-axis represents the attribute values and Y-axis gives their numerical values. Figure 4.3 shows F_{geology} . The attribute geology is scaled in the range of 5 to 2.3. And other attributes like landform, landuse and soil is scale in the range of 5 to 0. If attribute value is 5 it means it has highest importance in that attribute and 0 means least importance.

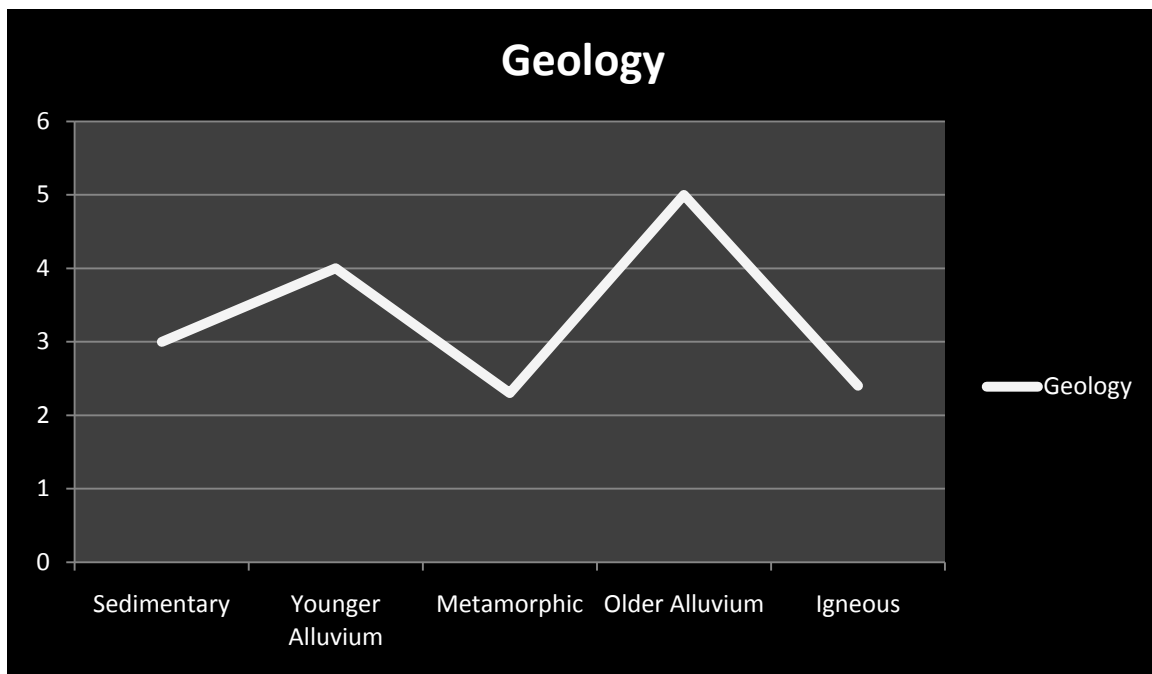


Figure 4.3 Geology attribute values

Figure 4.4 shows for landform attribute. It has huge values thus the range is scaled from 5 to 0. In our case base we have used almost 37 different values but in the diagram only few are shown. Thus it depicts F_{landform} .

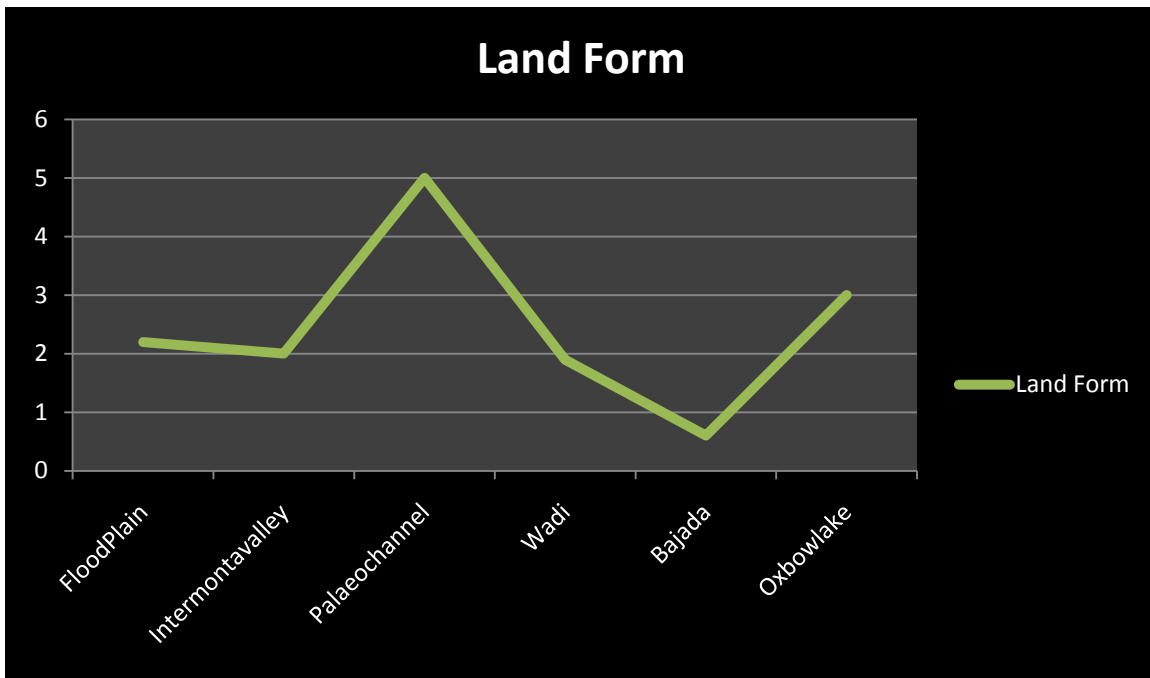


Figure 4.4 Land form attribute values

Figure 4.5, 4.6, 4.7 and 4.8 shows attribute values for soil, land use, slope and lineament. In all cases the range is scaled from 5 to 0 except slope and lineament. Figure 4.5 depicts F_{soil} . In both these cases they have 2 attribute values either high or low. Thus they both have values 2 or 1.

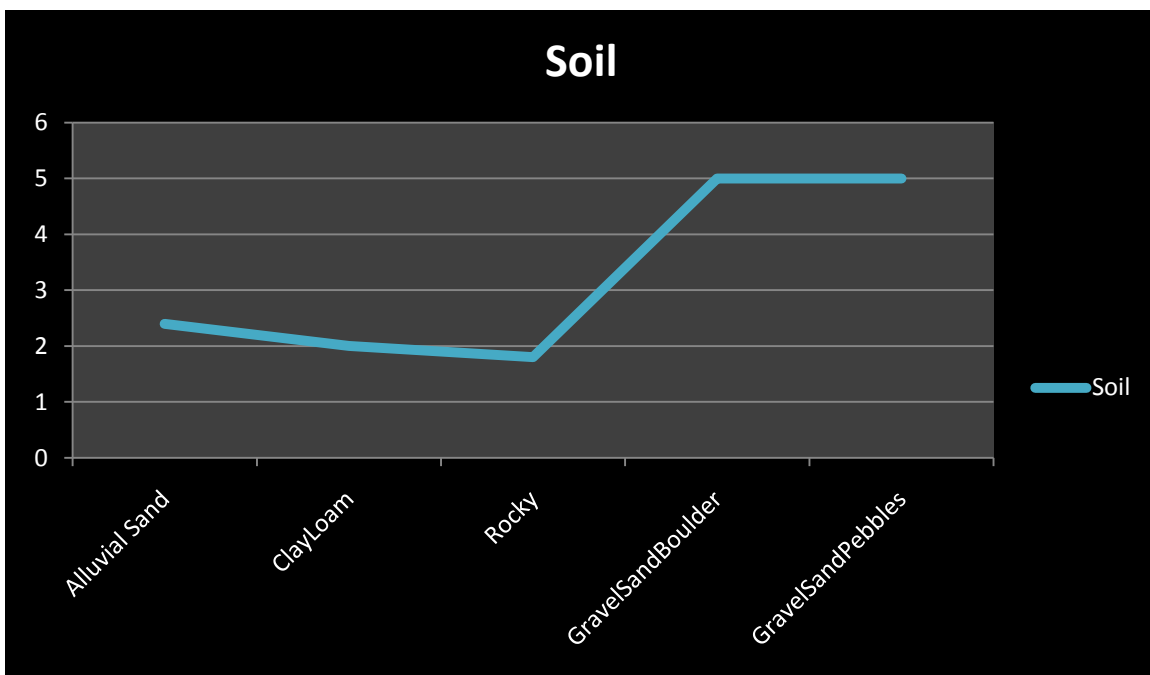


Figure 4.5 Soil attribute values

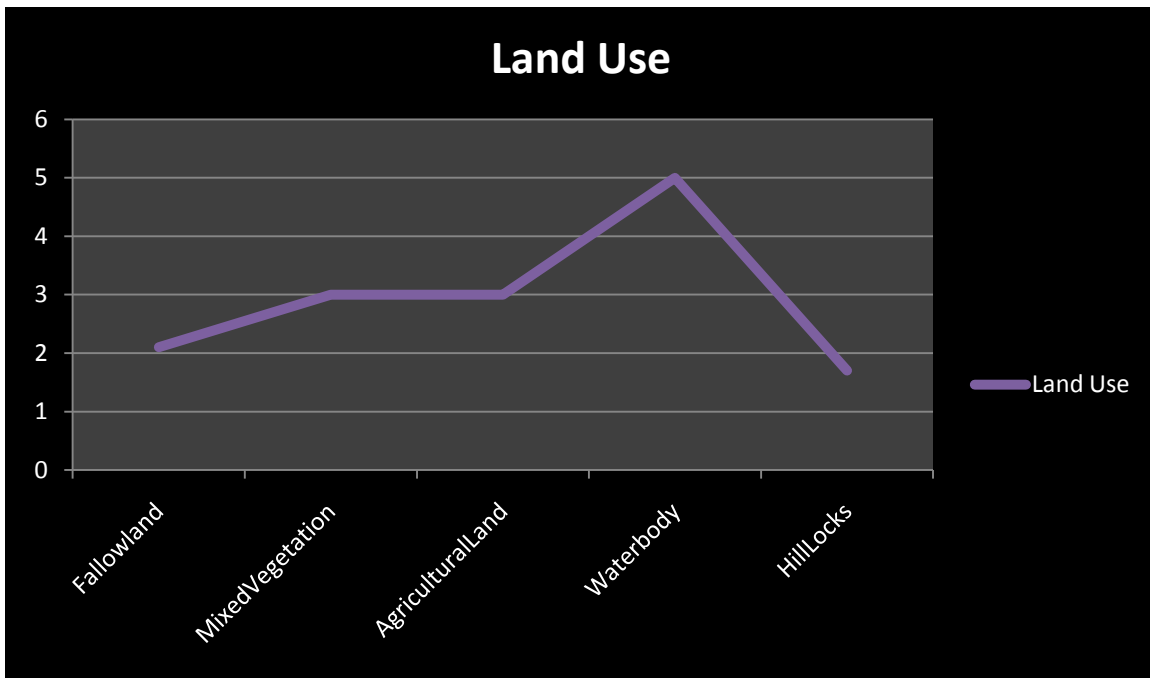


Figure 4.6 Land Use attribute values

Figure 4.7 shows for slope F_{slope} . Slope can be either gentle or steep. In case of gentle slope groundwater resistivity is low thus more water is accumulated in such region whereas in case of steep slope no water is accumulated. Thus gentle is assigned high value whereas steep has low value.

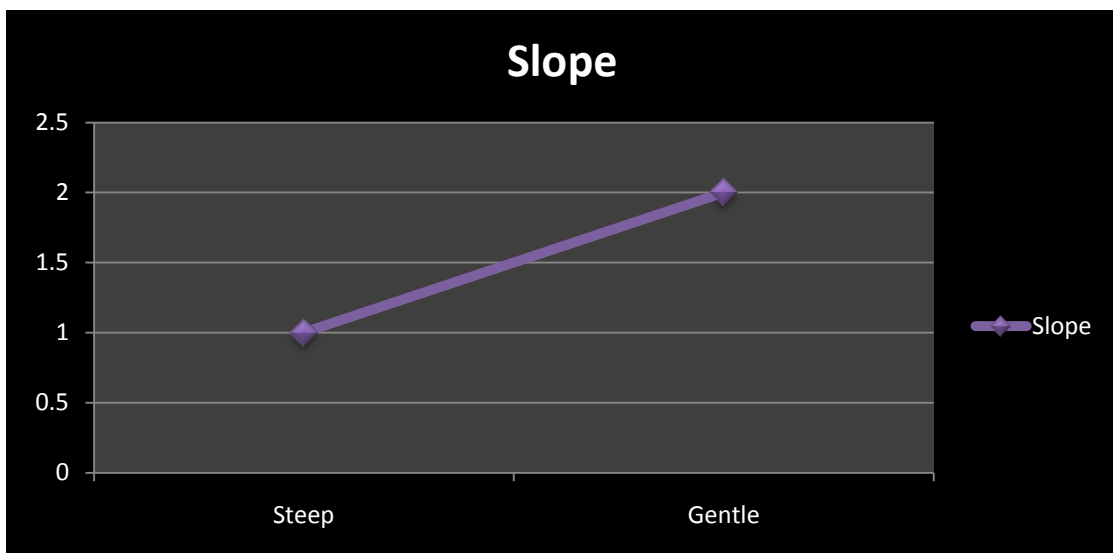


Figure 4.7 Slope attribute values

The following figure shows the attribute value for lineament. It means a sort of ridge in the rock. Thus if lineament is present high possibility of water otherwise low. It depicts $F_{\text{lineament}}$.

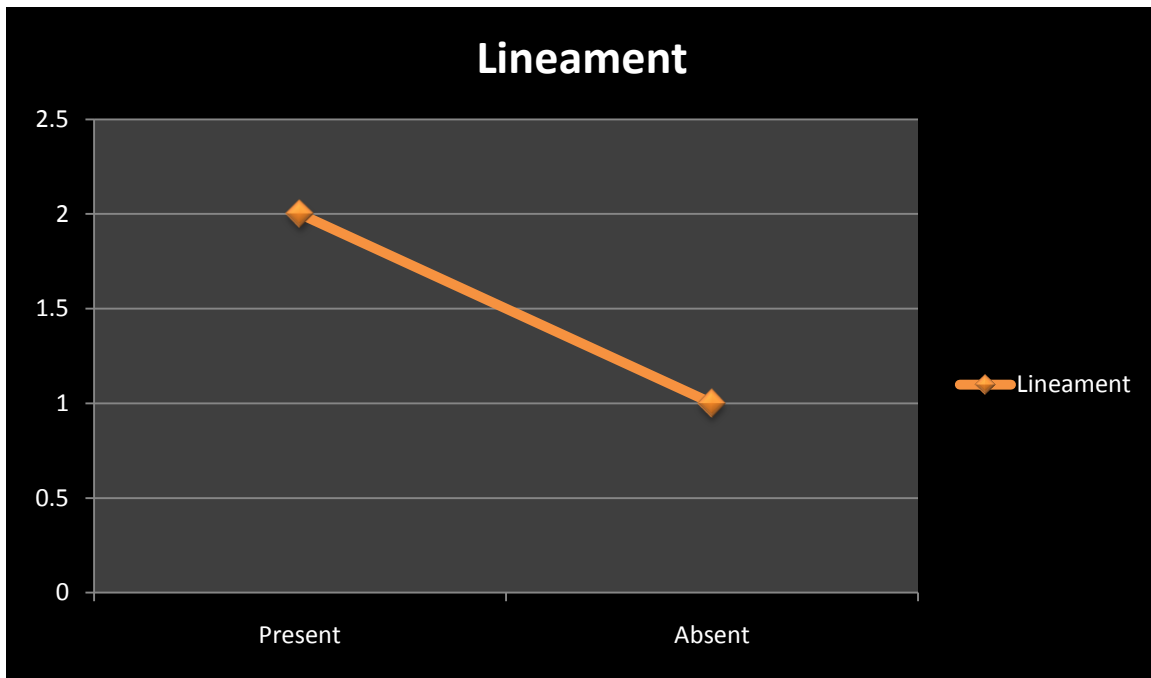


Figure 4.8 Lineament attribute values

Step 5.3: As each case of the case base is encountered and their similarity function is evaluated their function values is stored in a smaller database. This database is named as temp_case_base. It contains all the function value when compared between problem case and cases of case base.

Step 5.4: Increment Count_case_base by 1.

Step 6: The database named as temp_case_base contains all cases similarity values. Now we need to find out all those nests which are most suitable for a cuckoo. Depending on their value cuckoo finds out which nests are having maximum value. All the maximum valued nests are stored in final case base named final_case_base. Among these final nests cuckoo has to find out the nest which matches it requirements.

Step 7: The nests having maximum value are considered as best solutions. All these best solutions/quality solutions are stored.

Step 8: Find out all those cases are not kept in best solutions. Those ignored cases are found out by subtracting total_case-final_Case_base.

Step 9: The ignore cases are basically worst nests. They are abandoned for later uses. They are ignored for the current cuckoo. If some other cuckoo finds out nests they will be considered again. Here we find a fraction of worsts nests $p_a = \frac{\text{ignore_case}}{\text{total_case}}$

Step 10: Rank the best solution. In these step we need to find out the nest that mostly suits with the current cuckoo search. From the final_case_base we need to find out the nest that matches perfectly with the cuckoo search. It is considered as current best. To find out the current best following steps are performed

Step 10.1: Take each cases stored in final_case_base. Correlate the problem case i.e. current cuckoo search with the maximized nests. The nest from final_case_base which correlates maximum with the problem case is considered.

Step 10.2: The maximum correlated nests is again chosen as current best.

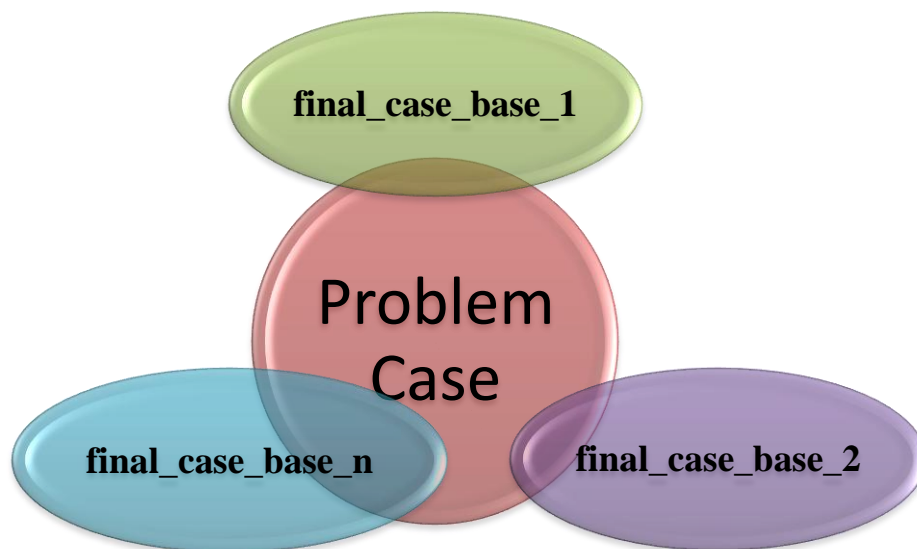


Figure 4.9 Current cuckoo searches for the best nest among quality solution

Step 11: Among highest correlation value cuckoo recognizes its best nest. From this nest it does some post processing.

Step 12: This is the final step of cuckoo search. Here post processing is performed. In post processing the best nest undergo through some propositional logic condition. Based on the outcome of those conditions groundwater potentiality is detected.

Step 12.1: Depending on the case some propositional logic formula is derived. The logic derives the outcome is formalized. All the conditions are discussed later.

Step 12.2: If condition is met following outcome is possible.

- i) High
- ii) Moderate
- iii) Medium.

If the percentage of groundwater is in the range of 0-64% then low probability, 65-84% then moderate potentiality otherwise high probability of groundwater.

Step 13: End of program.

4.4.2 Flowchart of the working model of Cuckoo search

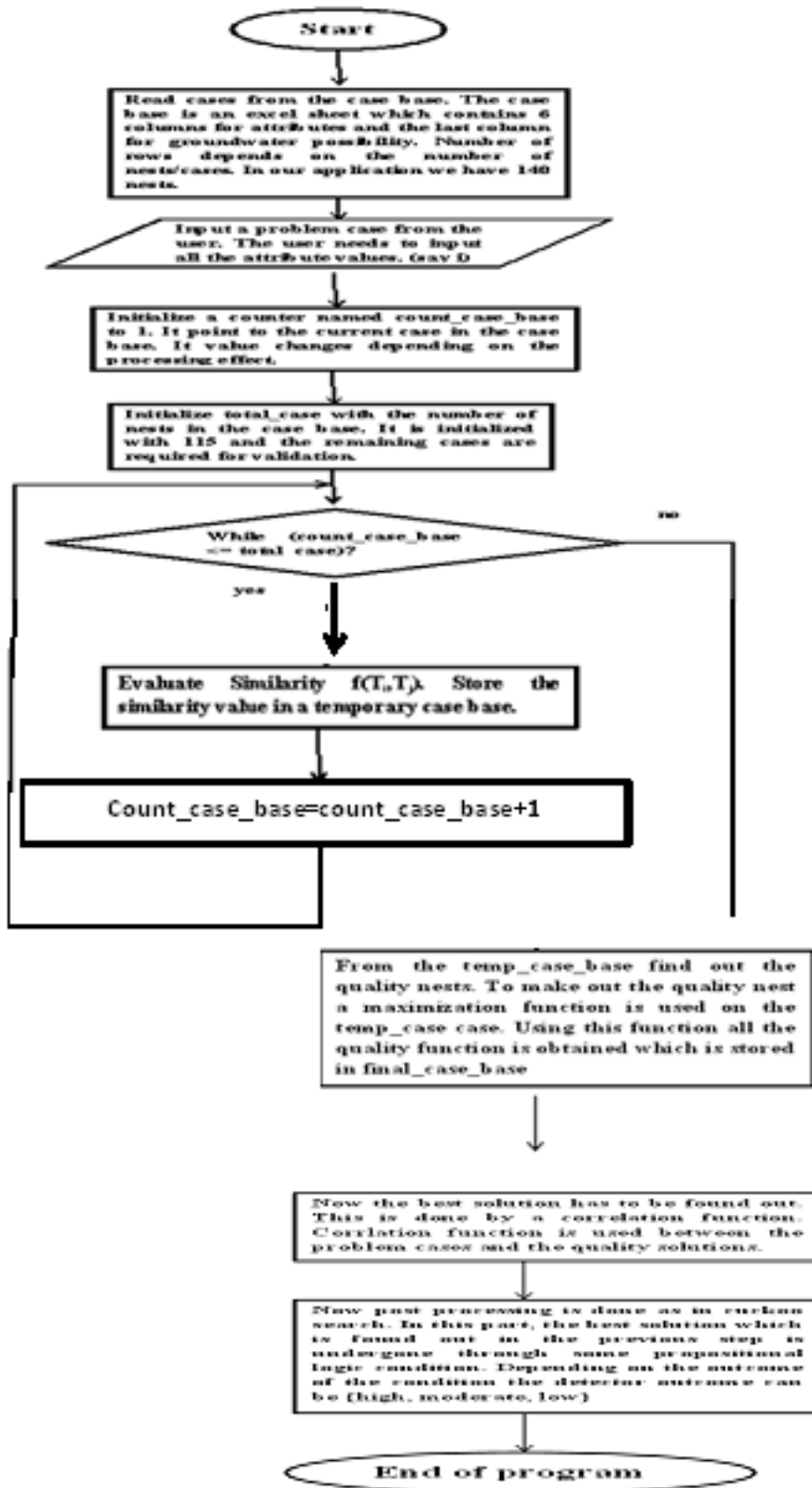


Figure 4.10 Detailed architecture of cuckoo search in groundwater potential

4.5 Usage of Propositional Logic

In mathematical logic, a propositional calculus or logic (also called sentential calculus or sentential logic) is a formal system in which formulas of a formal language may be interpreted as representing propositions. Thus we have defined several conditions. Only few of them are shown below:

i) $P = \text{“Geology is older alluvium”}$

$Q = \text{“High groundwater”}$.

$$P \rightarrow Q$$

ii) $P = \text{“geology is younger alluvium”}$

$Q = \text{“Landform is floodplain”}$

$R = \text{“High groundwater”}$

$$(P \wedge Q) \rightarrow R$$

iii) $P = \text{“geology is younger alluvium”}$

$Q = \text{“Landform is bajada”}$

$R = \text{“Moderate groundwater”}$

$$(P \wedge Q) \rightarrow R$$

iv) P= “geology is sedimentary”

Q= “landform is pediplain”

R= “Soil is clay”

S= “Land use is built-up”

T= “Low groundwater”

$$(P \wedge Q \wedge R \wedge S) \rightarrow T$$

v) P= “geology is igneous”

Q = “low groundwater”

$$P \rightarrow Q$$

vi) P= “slope is gentle”

Q= “lineament is absent”

R= “high groundwater”

$$(P \wedge Q) \leftrightarrow R$$

4.6 Unified Modelling Language of Groundwater Detection

UML diagrams have been used in our software to better understand the system. Basically some diagrams are depicted through which user better visualize the software. We have used IBM rational rose tools for drawing all the diagrams. The proposed bio-inspired groundwater potentiality detector software has following UML diagrams.

- i) **Use case Diagram:** In our software the user is the actor and all the necessary work that needed to be done by user are use case. The user has to input all the values of the attributes, reset the system, and exit it when finished with the software.

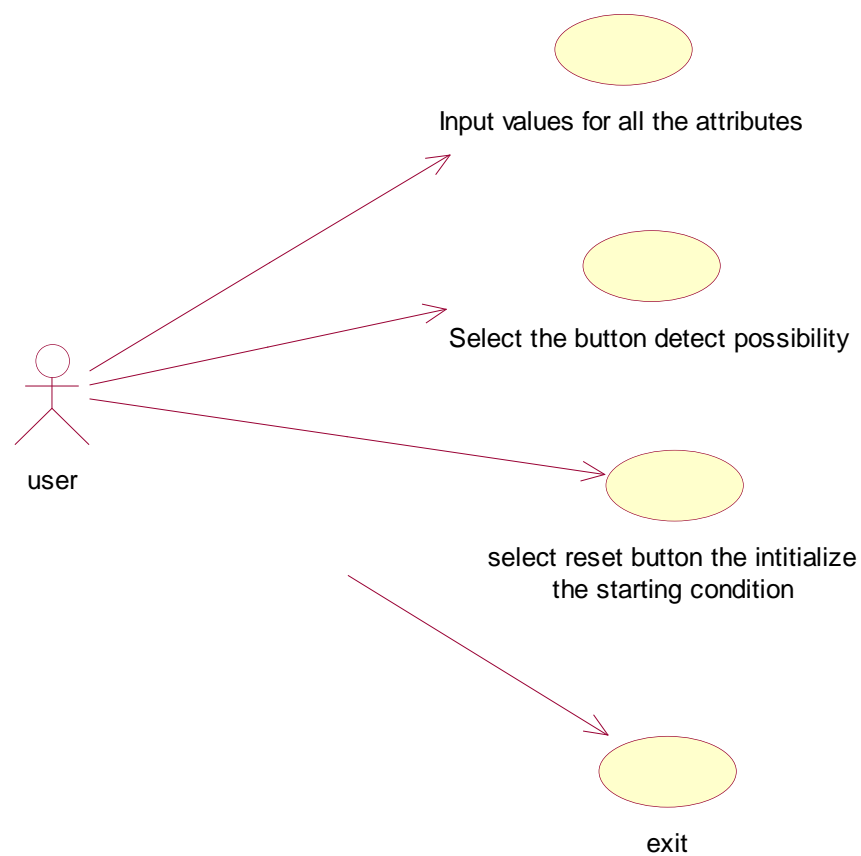


Figure 4.11 Use case diagram

- ii) **State Diagram:** It describes the behaviour of a system. State diagram describe all the possible states the system has. Depending on the event the system views different states. Thus in our system it has the following state. One input state, running state one the user click for detection and back to the initial state.

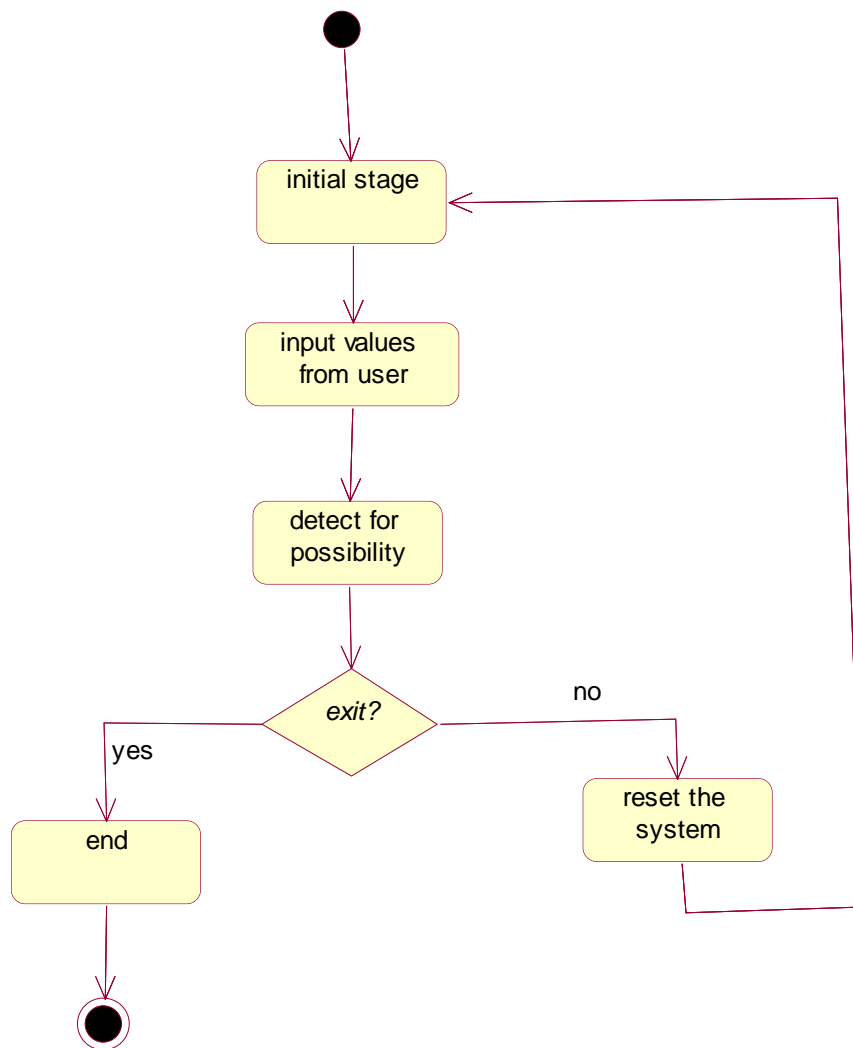


Figure 4.12 State Diagram

iii) **Interaction Diagram:** Interaction diagram model the behaviour of use cases by describing the way group of objects interact to complete the task. Thus the interaction diagram of the system.

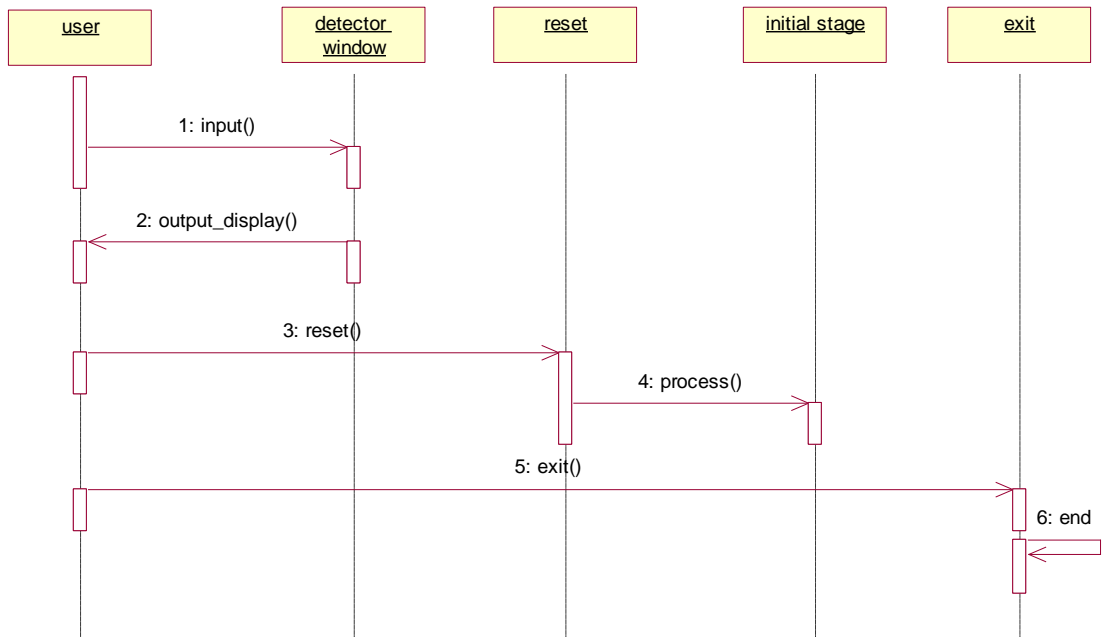


Figure 4.13 Interaction Diagram

5 *APPLYING MEMBRANE COMPUTING IN IMAGE CLASSIFICATION*

“A discovery is said to be an accident meeting a prepared mind”

Albert Szent-Gyorgyi

Preview

This chapter deals with the proposed framework in image classification. Image classification is a technique which is randomly used in remote sensing. Lot of hard and soft computing techniques had already been used but membrane computing had never shown its efficiency in this field. Thus the chapter gives an overview of our work.

5.1 Introduction

Though membrane computing had already been used to different fields of engineering but it has never shown its footprints in image classification. Thus we have proposed a framework of membrane computing in land cover feature extraction. It has proved itself a good classifier and even a refined one. Membrane computing deals with computing model abstracted from the structure of living cells and even it can be thought as the study of organization of cells in tissues thus even in our work each feature is recognized as a cell. The proposed work called Membrane computing in Image Classification can be adapted according to the database of expert knowledge for a more focussed satellite image classification.

With the help of satellite image various features of our nature are identified. The general features on which scientists generally work are water, barren, urban and vegetation areas.

Indeed other features can be recognized but these are commonly available in any kind of land areas. During feature extraction we attempt to identify all those features that land is consisting of.

The proposed work tried to focus on extracting the natural terrain features from satellite image using membrane computing. Membrane computing works on single unit of object thus it focuses on heterogeneous issues more prominently. Our till date soft computing technique classified image with high accuracy but they have little focus on sparse regions. Images generally has sparse region where lots of features can be identified. This heterogeneous region has been easily captured by membrane computing. Here lies a significant difference than other computational intelligence techniques.

Our proposed classifier is flexible enough to classify any land cover very efficiently based on 7-band data and hence can be adapted according to the application. The proposed algorithm is applied to two different images and in both of them it has shown efficient and good results. Thus it can be named as a Universal Classifier. The algorithm had been applied to the 7-band Indian Resourcesat Satellite Image of dimension 472X576 of the Alwar region in Rajasthan and even in 8-band multi-spectral, multi-resolution and multi-sensor satellite image of dimension 916X569 of Saharanpur region in Uttar Pradesh.

As it classified both images efficiently we can even use this classifier in other images too. It will even classify those images with powerful results. We also present a comparative study of the results of applying our proposed classifier and the other recent soft computing classifiers such as ACO, Fuzzy sets, Rough-Fuzzy Tie up and semantic web based classifiers when compared with the traditional probabilistic classifiers such as Minimum Distance to Mean Classifier and the Maximum Likelihood Classifier on the 7-band Alwar Image.

The proposed Universal Classifier is described in the following section:

5.2 Dataset used

Our objective is to use our universal classifier as an efficient land cover classifier for satellite image. As we have applied our classifier in two different dataset thus we have described both dataset here. We have taken multi-spectral, multi resolution and multi sensor image of size 472X576 of Alwar area in Rajasthan and 916X569 of Saharanpur region in Uttar Pradesh. For the first case 7 different bands and for the second one 8 different bands are taken. These bands are green, red, Near-Infra Red(NIR), Middle Infra Red(MIR), Radarsat 1, Radarsat 2 and Digital Elevation model for the first image and blue, red, green, NIR, MIR, Thermal 1, Thermal 2 and FIR. The expert has the attribute set P from the Indian Remote Sensing (IRS-P6) satellite optical band image set i.e. Green(G), Red(R), Near-Infrared and Middle Infrared bands. The ground resolution of these images is 23.5m and is taken from LISS-III, sensor and is provided to us courtesy of DTRL (Remote sensing) Lab, DRDO, Delhi.

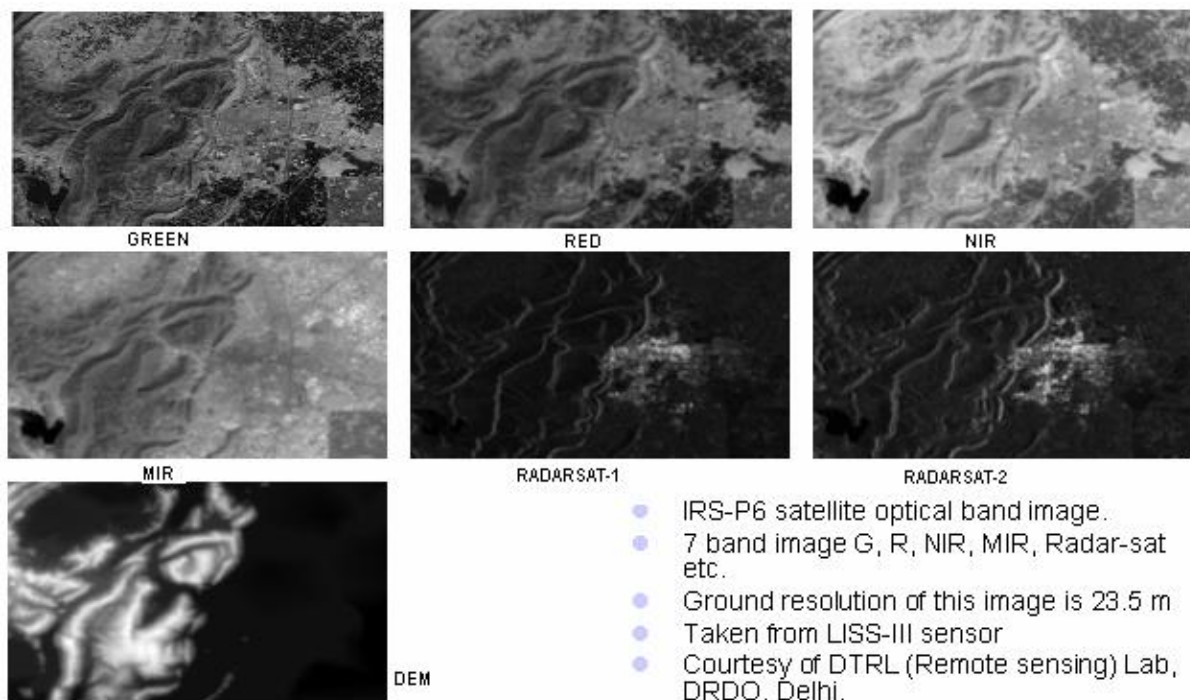


Figure 5.1 7 bands satellite image of Alwar region in Rajasthan.

The land cover classification with independent attributes set x consists of two sets of radarsat microwave image] 3s radarsat-1 and radarsat-2 and digital elevation model data. The r1 dataset is from low incidence angle between 20° - 27° and ground resolution of 10m. The high incidence angle for r2 dataset is between 45° - 49° in the c-band. The dem i.e. D dataset is also generated from SAR interferometry using r1 and r2 and have 25 meter resolution. These are chosen because the characteristics / observed behaviour of the visual bands and of the microwave bands are typically not related.

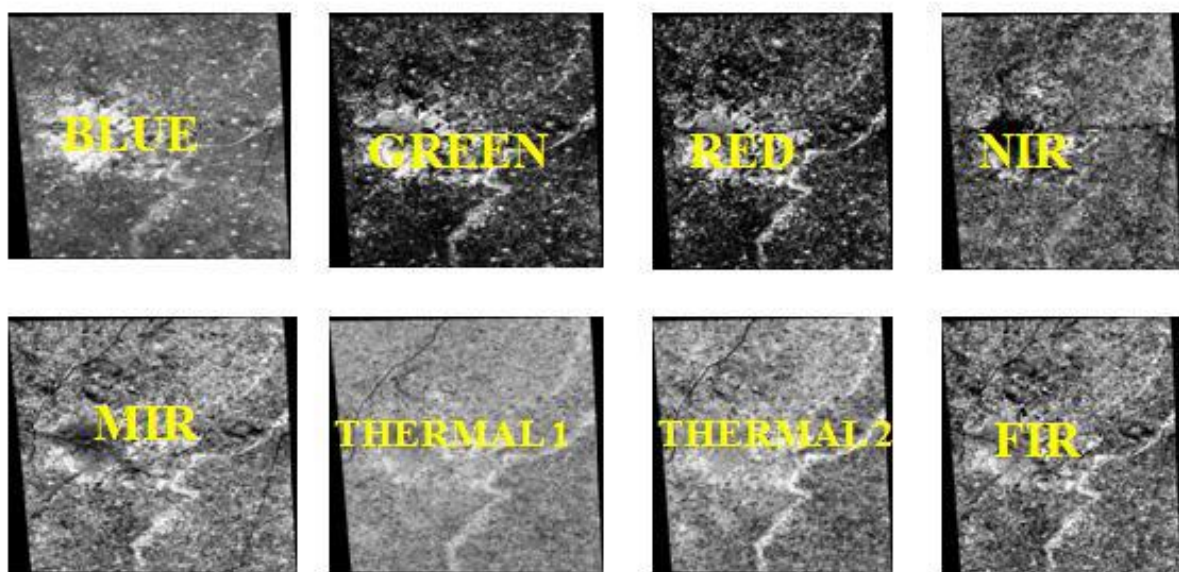


Figure 5.2 8 bands Satellite image of Saharanpur region in Uttar Pradesh

5.3 A brief description of the Image that validates our algorithm

An image is nothing but collection of pixels. Thus the image of Alwar taken by us for validating our algorithm has 2,55,136 pixels. Alwar region has mainly the following features:

- Water,
- Urban
- Vegetation

- Rocky,
- Barren.

Our geoscience experts recognized all these features in Alwar region. Thus our aim is to extract all these features from Alwar image. The data provided to us by our experts are collection of excel sheets. Each excel sheet is each feature. Thus number of rows in each sheet specifies number of pixels having that feature in the image. Depending on the data we have identified following observation.

- i) Water⁵⁴⁸ : It means excel sheet of water contains 548 rows .
- ii) Barren³¹⁵⁷ : It means excel sheet of barren contains 3157 rows .
- iii) Vegetation³⁰⁴⁹ : It means excel sheet of vegetation contains 3049 rows .
- iv) Urban²⁶⁰⁹ : It means excel sheet of urban contains 2609 rows .
- v) Rocky⁵¹⁶⁴ : It means excel sheet of rocky contains 5164 rows .

After applying our algorithm we have concluded what our image consists of. Our original image has following data:

- Barren¹¹²⁴⁹⁶ : It means our image has 1,12,496 number of pixels which is identified as barren region by our classifier.
- Rocky⁸³²³⁹ : It means our image has 83,239 number of pixels which is identified as rocky region by our classifier.
- Vegetation⁴⁹⁰⁴⁹ : It means our image has 49,049 number of pixels which is identified as vegetation region by our classifier.
- Urban¹⁰³⁹⁵ : It means our image has 10,395 number of pixels which is identified as urban region by our classifier.
- Water²⁵⁴⁰ : It means our image has 2,540 number of pixels which is identified as water region by our classifier.

5.4 Topographic Facets Extraction Interpreted in Membrane Computing Terms

The aim of this section is to provide an idea how components of membrane computing has been easily interpreted in terms of land cover features. Though it tries to give some light in these areas but a more systematic and detailed study remains to be elaborated. The membrane computing parameters [31] for land cover feature extraction algorithm are defined as follows:-

- **Object:** The multi-spectral band of image consist huge number of pixels. Once a rule is applied to object it moves between adjacent/neighbouring cells. Thus pixel focuses on objects of membrane computing. Thus objects $\in C$ is an integer and $C \in [0, 255136]$.
- **Membrane:** Each pixel can move from one cell to another which is being compartmentalized through membrane. Stated otherwise, membrane delimited inside from outside and this separation means that each membrane represents 5 specific features i.e. rocky, barren, water, urban, vegetation that are to be extracted from the image. Each membrane has its own object processing rule.
- **Membrane structure:** The framework of hierarchical structure where Region 1 is has 5 sub-membranes. It contains region 2,3,4,5 and 6 which represents each identified features of the Alwar region, i.e. region 2 is for urban, 3 for water, 4 for vegetation, 5 for barren and 6 represents rocky features. Each of the sub-membrane is elementary membrane. Figure 5.3 shows depicted the hierarchical structure.

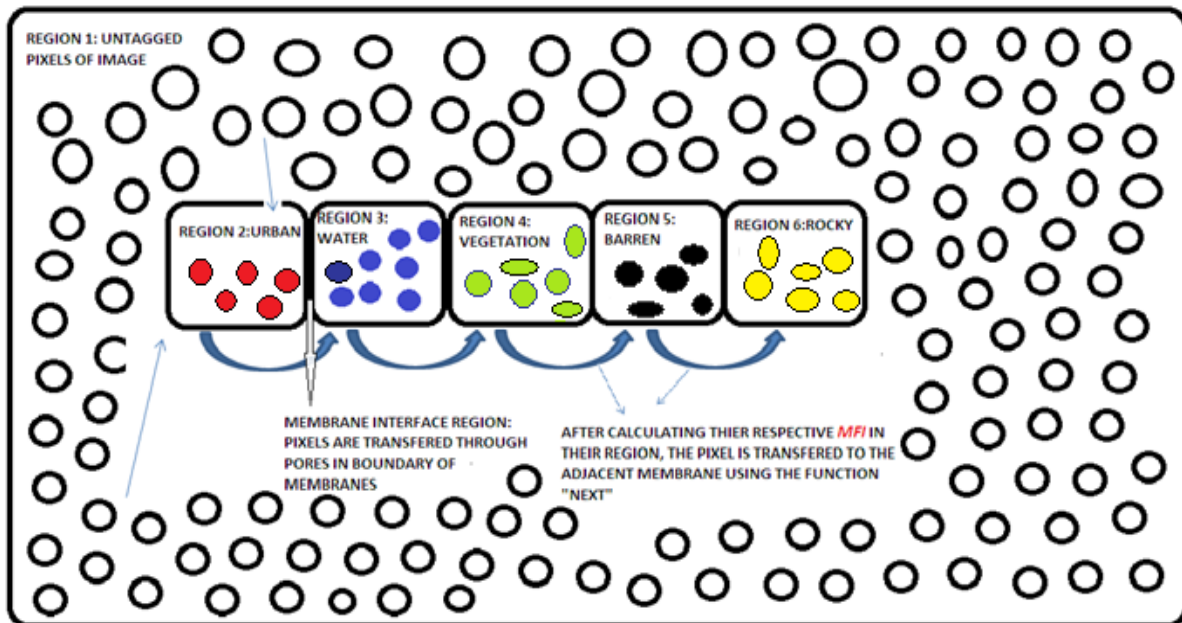


Figure 5.3 Membranous structure of Alwar Region [32]

- **Environment:** The outside of the skin membrane is considered as empty. Thus only objects which are unclassified in nature are thrown outside. The pixels which cannot be classified are thrown in the environment.
- **Multiset Rewriting Rules:** Depending on the evolution rule which is cooperative in nature, pixels can be absorbed by different membrane. The Region 1 of figure 5 consists of multisets of objects as $a^{112496}b^{83239}c^{49049}d^{10395}e^{2540}$ where a represents barren pixels, b represents rocky, c represents vegetation, d represents urban and e represent water pixels. Thus a^{112496} means the image has 112496 pixels which are barren in nature and so forth. All these values are computed after classification. Since each feature membrane i.e region 2, 3, 4, 5, 6 consists homogeneous pixels thus multiset of each feature membrane can be represented as :

- vi) Water⁵⁴⁸ (Region 3)
- vii) Barren³¹⁵⁷ (Region 5)
- viii) Vegetation³⁰⁴⁹ (Region 4)

ix) Urban²⁶⁰⁹ (Region 2)

x) Rocky⁵¹⁶⁴ (Region 6)

- **Symport rule:** They can represent input into a membrane or ejection out of the membrane of pixels. If the rule condition is satisfied it takes the form (pixel, in_k) where k is the region (water, rocky, barren, vegetation, urban) otherwise it is of the form (pixel, out).
- **Promoter/inhibitor:** The rules can only be applied under certain conditions. Depicting the barren condition for promoter case which can be applied to other feature membrane also can be written as:-

$$\begin{array}{l} (\text{barren})^p \rightarrow (\text{barren})^{p+1} \\ \left| \right. \\ \text{Max}(\text{corr}(\text{barren}, \text{pixel_value})) \end{array}$$

While the inhibitor case is as follows:

$$\begin{array}{l} (\text{barren})^p \rightarrow (\text{barren})^{p+1} \\ \left| \right. \\ \text{Min}(\text{corr}(\text{barren}, \text{pixel_value})) \end{array}$$

In either case we take the maximum or minimum value of correlation between barren value and each pixel value. Here p represents the number of pixels which are classified as barren till now and if the promoter condition is met then the pixel is absorbed in barren region.

- **Maximal Parallelism:** It means the rule should be applied in parallel to the maximum degree. The multiset of objects present in Region 1 as in figure 5.3 is $w = a^{112496} b^{83239} c^{49049} d^{10395} e^{2540}$ and the set of rules R residing in the same region is $R = \{r_1, r_2, r_3, r_4, r_5\}$. The definition of these rules is given at Proposed Architecture.

Thus land cover feature extraction P system transition system can be defined as follows:

$$\Pi = (O, H, \mu, w_1, r_1, r_2, r_3, r_4, r_5, i_0),$$

where:

1. $O = \{a,b,c,d,e\}$.
2. $H = \{1,2,3,4,5,6\}$
3. μ is membrane structure represented in parenthesis form as:

$$[[]_2 []_3 []_4 []_5 []_6]_1$$

4. $w_1 = a^{112496} b^{83239} c^{49049} d^{10395} e^{2540}$
5. $r_i, i=1$ to 5 ,
6. $i_0 = \{2, 3, 4, 5, 6\}$.

5.5 Formulation of Evolution Rules

Evolution rule is applied to objects and depending on the conditions of those rules i.e. promoter/inhibitor condition object change its position in a membrane. Thus in our application i.e. in Image classification we have taken Alwar area and Saharanpur region for validating our algorithm and depending on our application we have formulated some following rules. These rules [33] are applied parallelly in the image and we have obtained efficient results due to maximal parallelism. To follow all rules we need to go through figure 5.3. The formulation of all rules is provided in figure 5.4. Following is the description of all the symbols used in our rules and after that a brief description of all rules.

- i) a means barren region.
- ii) b means rocky region.
- iii) c means vegetation region.
- iv) d means urban region and
- v) e means water region.

- vi) PV means pixel_value which will be described during description of algorithm.
- vii) C means correlation function used for region matching and it is described in section 5.7.

$$\begin{array}{l}
 r_1: a^5 b^3 d \rightarrow v_1 \\
 \text{where } v_1: \\
 \{(d, in_k), (a^5 b^3, here)\} \quad \left| \begin{array}{l} 5 \\ k=\{\forall \max\{C(i,PV)\}\} \& \\ i=1 \quad k=urban \end{array} \right. \\
 \\
 r_2: a^4 e c^4 \rightarrow v_2 \\
 \text{where } v_2: \\
 \{(e, in_k), (a^4 c^4, here)\} \quad \left| \begin{array}{l} 5 \\ k=\{\forall \max\{C(i,PV)\}\} \& \\ i=1 \quad k=water \end{array} \right. \\
 \\
 r_3: c d^5 \rightarrow v_3 \\
 \text{where } v_3: \\
 \{(c, in_k), (d^5, here)\} \quad \left| \begin{array}{l} 5 \\ k=\{\forall \max\{C(i,PV)\}\} \& \\ i=1 \quad k=vegetation \end{array} \right. \\
 \\
 r_4: a d^9 e^2 \rightarrow v_4 \\
 \text{where } v_4: \\
 \{(a, in_k), (d^9 e^2, here)\} \quad \left| \begin{array}{l} 5 \\ k=\{\forall \max\{C(i,PV)\}\} \& \\ i=1 \quad k=barren \end{array} \right. \\
 \\
 r_5: b c^7 \rightarrow v_5 \\
 \text{where } v_5: \\
 \{(b, in_k), (c^7, here)\} \quad \left| \begin{array}{l} 5 \\ k=\{\forall \max\{C(i,PV)\}\} \& \\ i=1 \quad k=rocky \end{array} \right.
 \end{array}$$

Figure 5.4 Evolution rule applied to Alwar region

Following gives a brief description of all the rules:

- Rule 1 (r_1): It means while we apply this rule we have taken 5 objects of barren region, 3 objects of rocky region and 1 object of urban region. Thus total 9 objects of the main image are taken. Thus we have taken randomly 9 objects of the image with close approximation where each object matching the above said condition. Now we apply correlation function taking one object at a time and then matching each object with each feature. If the considered object matches any of the features then it is absorbed in the matching feature. Thus if 'd' tagged objects match with urban then it is absorbed in that region i.e. in_k where k represents urban and objects that does not match urban region are placed in region 1 only.
- Rule 2 (r_2): It means while we apply this rule we have taken 4 objects of barren region, 4 objects of vegetation region and 1 object of water region. Thus total 9 objects of the main image are taken. Thus we have taken randomly 9 objects of the image with close approximation where each object matching the above said condition. Now we apply correlation function taking one object at a time and then matching each object with each feature. If the considered object matches any of the features then it is absorbed in the matching feature. Thus if 'e' tagged objects match with water then it is absorbed in that region i.e. in_k where k represents water and objects that does not match water region are placed in region 1 only.
- Rule 3 (r_3): It means while we apply this rule we have taken 5 objects of urban region, and 1 object of vegetation region. Thus total 5 objects of the main image are taken. Thus we have taken randomly 5 objects of the image with close approximation where each object matching the above said condition. Now we apply correlation function taking one object at a time and then matching each object with each feature. If the considered object matches any of the features then it is absorbed in the matching

feature. Thus if 'c' tagged objects match with vegetation then it is absorbed in that region i.e. in_k where k represents vegetation and objects that does not match vegetation region are placed in region 1 only.

- Rule 4 (r_4): It means while we apply this rule we have taken 9 objects of urban region, 2 objects of water region and 1 object of barren region. Thus total 12 objects of the main image are taken. Thus we have taken randomly 12 objects of the image with close approximation where each object matching the above said condition. Now we apply correlation function taking one object at a time and then matching each object with each feature. If the considered object matches any of the features then it is absorbed in the matching feature. Thus if 'a' tagged objects match with barren then it is absorbed in that region i.e. in_k where k represents barren and objects that does not match barren region are placed in region 1 only.
- Rule 5 (r_5): It means while we apply this rule we have taken 7 objects of vegetation region and 1 object of rocky region. Thus total 8 objects of the main image are taken. Thus we have taken randomly 8 objects of the image with close approximation where each object matching the above said condition. Now we apply correlation function taking one object at a time and then matching each object with each feature. If the considered object matches any of the features then it is absorbed in the matching feature. Thus if 'b' tagged objects match with rocky then it is absorbed in that region i.e. in_k where k represents rocky and objects that does not match rocky region are placed in region 1 only.

5.6 Algorithm of Membrane Computing in Image Classification

The following algorithm has been proposed for extracting land features from the satellite image [33].

Input: Multi-spectral satellite image

Output: Extracted features from the image

Algorithm:

1. Place all the unclassified objects in one class (Region 1).
2. Create sub-membranes for each land cover features (for water, barren, vegetation, rocky, urban).
3. Find out the mean of each sub-membrane/training set i.e. Barren_Mean, Urban_Mean, Water_Mean, Vegetation_Mean, Rocky_Mean.
4. while(no_of_unclassified_objects \neq NULL)
 - {
 - i) Take any combinations of evolution rules (e.g. r_1r_4 , r_3r_5 , $r_2r_5r_1$)
[To achieve maximal parallelism no. of evolution rules taken must be greater than or equal to 2]
 - ii) Find out object (object_value) depending on considered evolution rules. [If r_1 and r_3 are to be applied parallel then find out the value of any randomly taken 15 objects. r_1 has a^5b^3d and r_3 has cd^5 multisets of objects. Thus r_1 contributes 9 objects and r_3 6 objects. Total of 15 (9+6) objects]
 - iii) Apply all the considered evolution rules parallel on randomly taken objects/objects.
 - iv) If (evolution rule is satisfied)
 - Then absorb the object in the specified feature
 - Else
 - Leave the objects in the same region i.e. Region 1.

End If.

v) No_of_unclassified_objects - -

}

5. End of program.

Description of Algorithm is provided in detail:

Step 1 Expert has provided us with information that most features are extracted in NIR and MIR band. Thus we create a matlab file .m that has data of the entire image. It is an excel sheet with 9 columns. The first and second column indicates the x and y co-ordinate and the rest 7 columns contain the band values data of all the objects. We recognize this excel sheet as one class.

Step 2. Now we know there are 5 features in Alwar region. Accordingly we create 5 sub-membranes within region 1 to identify all the features. The sub-membranes are water, vegetation, barren, rocky and urban region. This sub-membrane contains DN values as provided by geoscientists.

Step 3. The expert has given us the training set (i.e. water, barren, urban, rocky vegetation). From the training set we compute the mean of each attribute of each land cover feature. For example we get the mean of water(i.e. a land cover feature) as water_mean =[M₁,M₂,M₃,M₄,M₅,M₆,M₇] where M₁ is the mean of red band values, M₂ mean of NIR band value, M₃ mean of MIR band value, M₄ mean of RS1 band value, M₅ mean of RS2 band value, M₆ mean of green band value, M₇ mean of DEM band value. Below diagram shows training dataset of water.

x coordinate	y coordinate	RED	GREEN	NIR	MIR	RS1	RS2	DEM
0	0	134	119	186	133	39	47	99
0	0	144	120	192	137	67	52	99
0	0	132	111	186	133	37	46	103
0	0	132	112	188	135	45	34	94
0	0	130	112	181	135	68	27	94
0	0	117	99	173	115	69	47	98
0	0	123	99	181	124	72	33	98
0	0	121	104	181	131	51	21	91
0	0	132	112	179	131	86	69	109

Step 4. All objects that contained in region 1 are untagged objects. Thus we need process each object of this membrane. Until all objects of this membrane are not processed we continue in a loop.

Step 4.1. In membrane computing maximal parallelism is achieved if we process two or more evolution at the same time. Depending on the processor we take any combination of evolution rules. There is random function and it decides which rule to take and which rule not to take at that instant of time.

Step 4.2. When processor had chosen which rules to take it return with evolution number. [If r_1 and r_3 are to be applied parallel then find out the value of any randomly taken 15 objects. r_1 has a^5b^3d and r_3 has cd^5 multisets of objects. Thus r_1 contributes 9 objects and r_3 6 objects. Total of 15 (9+6) objects].

Step 4.3. We formulate number of objects that needed to be calculated. Find out all those objects object_value (based on evolution rule). Object_value = $[P_1 P_2 P_3 P_4 P_5 P_6 P_7]$ where P_1 is the value of digital numbers of red band, P_2 is the value of digital numbers of green band, P_3 is the value of digital numbers of NIR band, P_4 is the value of digital numbers of MIR band, P_5 is the value of digital numbers of RS1 band, P_6 is the value of digital numbers of RS2 band, P_7 is the value of digital numbers of DEM band. After find their values all those values are stored in an excel sheet.

Step 4.4. Now we apply promoter/inhibitor condition depending on the evolution rule. We formulate a correlation function between each object and each membrane. The object which match mostly with a sub-membrane that object is intake in that sub-membrane otherwise it remain in the same outer membrane i.e. region 1 only. At the end they are tagged indication that they are already processed and no further processing is required for them.

Step 4.5. The counter which counts number of objects processed is decreased by one is updated. If counter is 0 then processing stop.

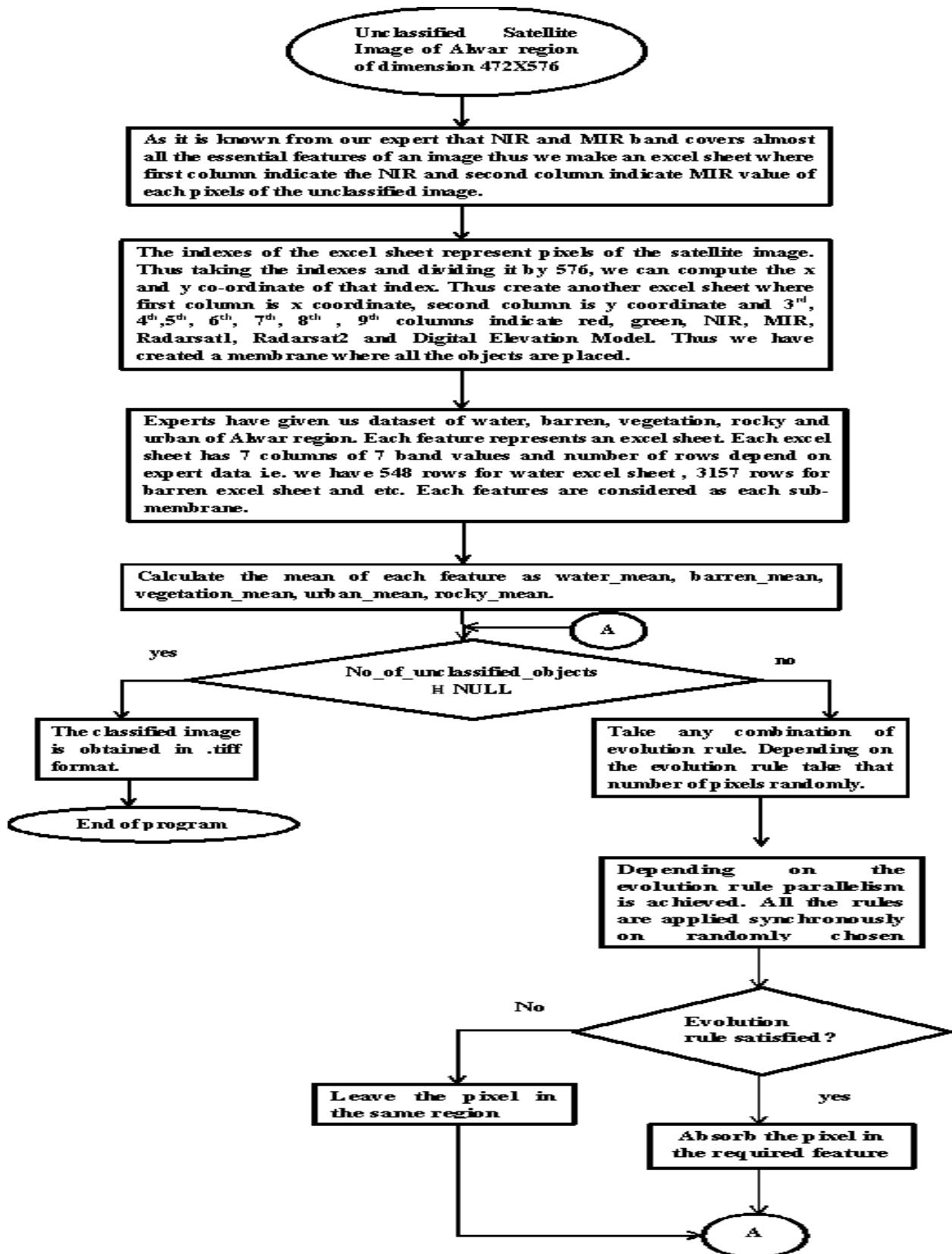


Figure 5.5 Flowchart of Membrane computing applied in Image Classification

5.7 Description of Mathematical formula used in Membrane Computing feature extraction

Step 4.4 in our proposed algorithm has applied population correlation function for region matching. Thus a brief description is laid down below for population correlation.

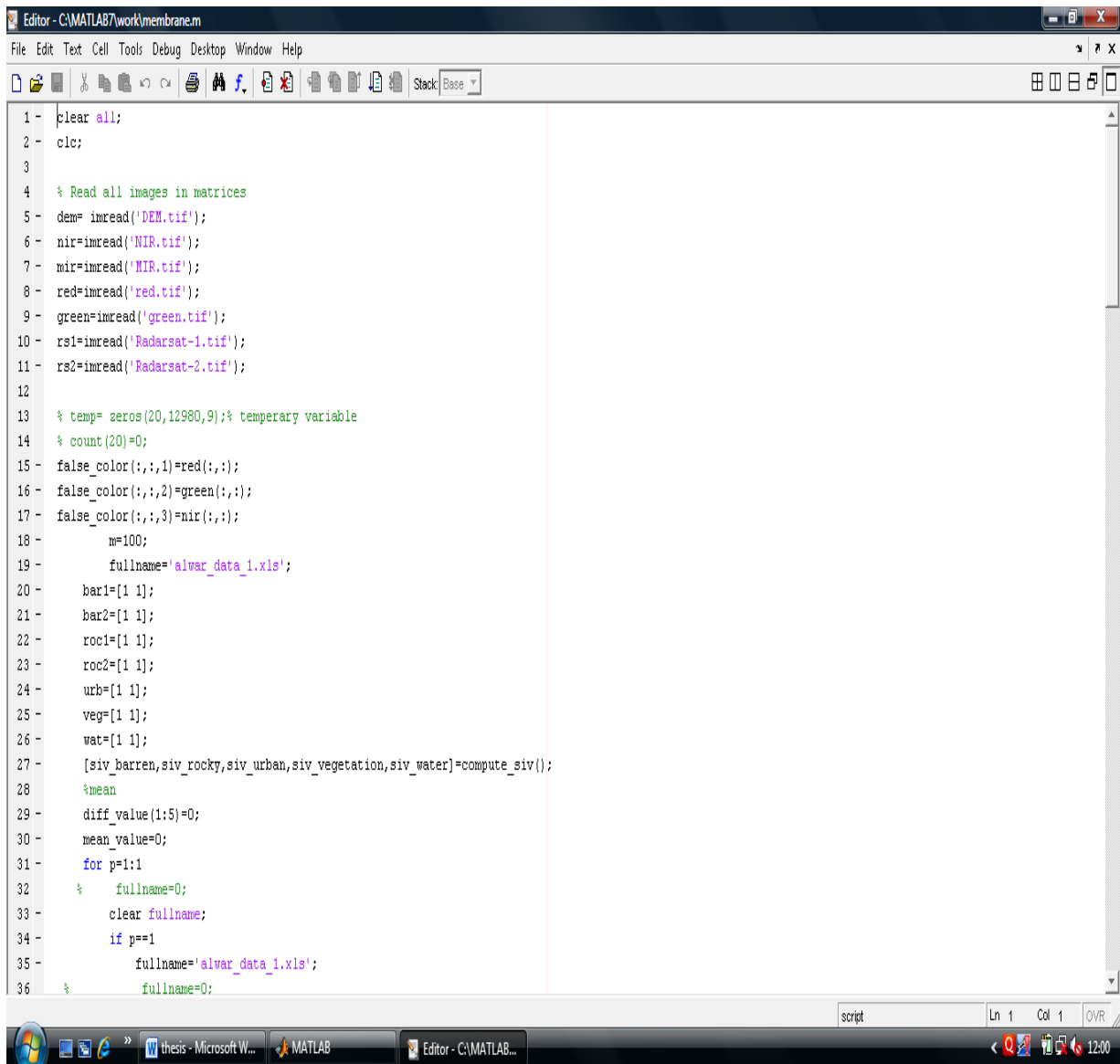
Correlation focuses on departure of two or more random variables from independence. We have worked on Pearson correlation. When we divide the covariance of the two variables by the product of their standard deviations then Pearson's correlation is obtained. The population correlation coefficient $\rho_{X,Y}$ between two random variables X and Y with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y is defined as:

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y},$$

where E is the expected value operator, *cov* means covariance, and, *corr* a widely used alternative notation for Pearson's correlation. The Pearson correlation is valid only if both of the standard deviations are finite and nonzero. The formula has been given in equation 1.

$$\rho(x, y) = \frac{E[xy]}{\sigma(x)\sigma(y)} \quad (1)$$

where x and y are two zero-mean real-valued random variable, $E[xy]$ is the cross-correlation between x and y , and $\sigma_x^2 = E[x^2]$ and $\sigma_y^2 = E[y^2]$ are the variance of the two variables x , y respectively. In our approach x represents the value of the attributes of the pixel and y represents the mean value of the attribute of the training set.



```
1 - clear all;
2 - clc;
3
4 - % Read all images in matrices
5 - dem= imread('DEM.tif');
6 - nir=imread('NIR.tif');
7 - mir=imread('MIR.tif');
8 - red=imread('red.tif');
9 - green=imread('green.tif');
10 - rs1=imread('Radersat-1.tif');
11 - rs2=imread('Radersat-2.tif');
12
13 - % temp= zeros(20,12980,9);% temporary variable
14 - % count(20)=0;
15 - false_color(:,:,1)=red(:,:,);
16 - false_color(:,:,2)=green(:,:,);
17 - false_color(:,:,3)=nir(:,:,);
18 -     m=100;
19 -     fullname='alwar_data_1.xls';
20 -     bar1=[1 1];
21 -     bar2=[1 1];
22 -     roc1=[1 1];
23 -     roc2=[1 1];
24 -     urb=[1 1];
25 -     veg=[1 1];
26 -     wat=[1 1];
27 -     [siv_barren,siv_rocky,siv_urban,siv_vegetation,siv_water]=compute_siv();
28 -     %mean
29 -     diff_value(1:5)=0;
30 -     mean_value=0;
31 -     for p=1:1
32 -         %     fullname=0;
33 -         clear fullname;
34 -         if p==1
35 -             fullname='alwar_data_1.xls';
36 -         %     fullname=0;
```

Figure 5.5 MATLAB window containing the code for Universal Classifier

5.8 Structural Design of Universal Classifier

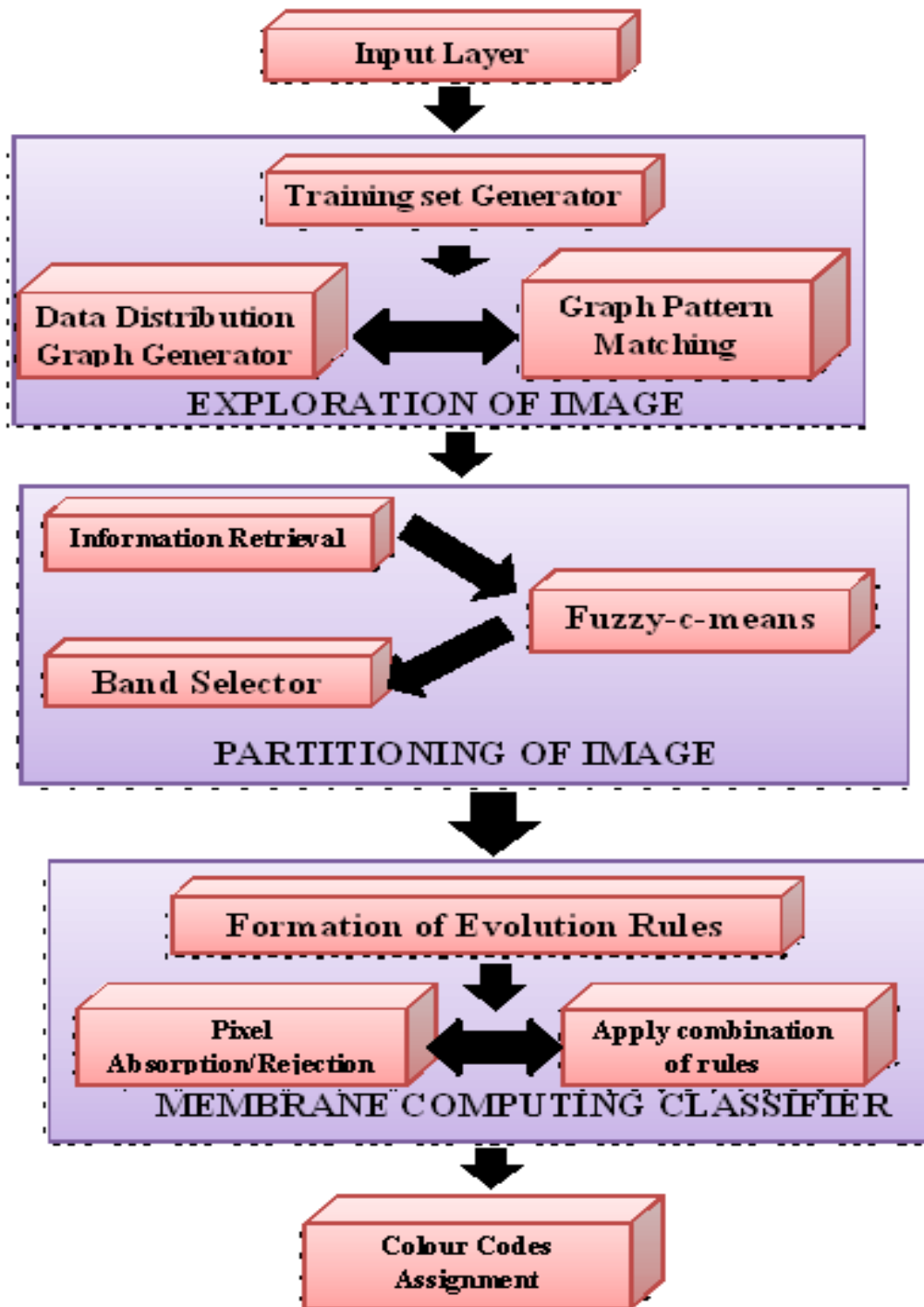


Figure 5.7 Structural Design of Universal Classifier

The above layers of our universal classifier are described below in the following section:

Input Layer: A high resolution multi-spectral satellite image is taken as input. For our illustration, the images that we have used are 7 bands Indian Resourcesat satellite image of size 472 X 576 of Alwar region of Rajasthan and 8 bands of same satellite of size 916 X 569 of Saharanpur region of Uttar Pradesh.

Exploration of Image: This layer is responsible for analysing the input satellite image to exploit the data distribution of the training set and feed its output to the next layer i.e. partitioning of image.

- (i) **Training set generation:** In this layer, we generate the training set using ERDAS software, according to different classes by experts.
- (ii) **Data distribution graph generation:** Now, using this training set, we plot the data distribution graph between the average of the Standard Deviations of each land cover feature viz water, urban, rocky, vegetation and barren (plotted on the y-axis) for each of the 7-Bands of the image i.e. Red, Green, NIR, MIR, RS1, RS2 and DEM (plotted as the x-axis) as shown in figure 5.8. On the x-axis, each of the 7-bands of the image are represented as integers from 4-10 for Red, Green, NIR, MIR, RS1, RS2 and DEM bands respectively. We are not considering integers 1-3 in the graph plot on the x-axis since these represent the other attributes of the image namely pixel no. and x and y coordinates which are not required at this stage.

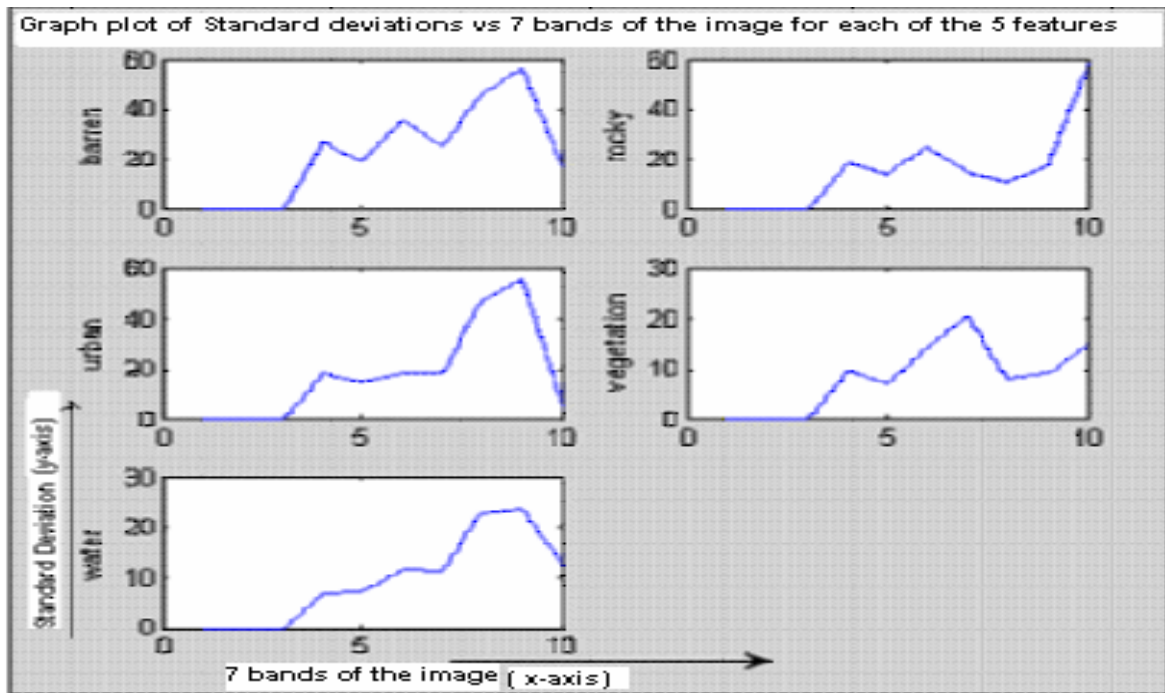


Figure 5.8 Graph plot of the standard deviations of each land cover feature v/s each of the 7 bands [13]

- (iii) **Graph Pattern Similarity Analysis Layer:** From the graph, it can be observed that the minimum difference between the average standard deviations of the NIR and the MIR bands of the Alwar Image is achieved in particularly two land cover features, those of water and urban area, both of which exhibit the same graph pattern in the NIR and the MIR bands i.e. | average of standard deviation of NIR band ~ average of standard deviation of the MIR band |_{lowest} = {water, urban}. Thus our classifier had perfectly classified water region but urban region can be further explored.

Partitioning of Image Layer: As we have used huge data image nearly more than 2 lakhs pixels we needed to partition the image into smaller portion. Moreover the version of Matlab 7.0 can handle only 65,356 pixels at a time. Thus due to software leakage we had to use this layer.

- (i) **Image Grid Division:** In this step basically we retrieved the whole image information in two columns in an excel sheet where first column represent the

NIR band values and the second column represents MIR band values. These two bands are considered only during the initial stage because according to our experts these two bands extract most of the features in the image.

- (ii) **Fuzzy-c-means:** In this stage we group the pixels. This stage is required due shortage of software. Basically our software can handle only 65,536 objects at a time thus to group the pixels we have used this mechanism. It divided the image into 4X4 matrix. Though in our final processing we have used single unit of object only.
- (iii) **Band Selector:** Now in this layer we have produced an excel sheet which contains the whole image information. Figure 5.9 illustrates this layer. Thus depending on the pixel values we have retrieved 7 bands values from the image. The first and second column has x and y co-ordinate values and the next 7 columns contain all the 7 band values.

x co-ordinate	y co-ordinate	RED	GREEN	NIR	MIR	RS1	RS2	DEM
244	478	162	133	199	153	16	24	14
244	486	136	117	201	148	46	33	12
244	490	146	125	199	151	29	33	11
245	486	148	123	202	144	39	60	12
245	491	138	115	199	153	18	28	11
246	485	170	144	202	153	63	38	11
247	536	130	111	199	146	9	13	6
250	453	152	130	199	148	18	67	13
250	472	154	127	204	153	40	53	15
250	473	152	127	199	149	27	49	15
250	540	134	112	199	146	9	44	6
250	541	130	109	197	149	9	55	6

Figure 5.9 Band selector of Alwar image

Membrane Computing Classifier:

- (i) **Formation of Evolution rules:** Now the main layer of this classifier comes. All the rules that needed to be applied are formed here. Here all the rules are defined before starting of the main step i.e. applying all these evolution rules.
- (ii) **Applying the rules:** This layer basically takes any combination of rules and apply rules promoter/inhibitor condition. Number of objects taken depends on the combination of rules. The rules are applied and depending on its symport the object is absorbed accordingly to that sub-membrane.
- (iii) **Pixel absorption/rejection:** If the rule is satisfied then object get absorbed in a sub-membrane that is implied in the rule otherwise it remains in the membrane from where it was taken.

Colour code assignment: Finally colour codes are assigned for each pixel of the image corresponding to the classified feature of each pixel. The colour codes for water, urban, vegetation, rocky and barren features are assigned as blue, red, green, yellow and black colour respectively.

Output Layer: Final classified image is obtained in .jpeg or .tiff or any other image format.

6 RESULTS AND DISCUSSION

“By the time we’ve made it, we’ve had it”

Malcolm Forbes

Preview

This chapter gives all the detailed description of the results obtained. The first section discusses the result of groundwater detection. It has achieved 99% efficiency and thus it can be named as Universal Detector. It is applied to various regions and in almost all regions it has generated correct output. The second section deals with membrane computing in image classification. It even proves that the classifier which we proposed is a universal one. It gives almost correct output on a specific query. Even we have discussed our results and even found out efficiency of our result through Kappa coefficient. The algorithm proposed in the previous chapter has been implemented in Matlab 7.0 [34]. The algorithm has been applied to 7-band image of Alwar city (in Rajasthan) and even in 8-band image of Saharanpur area (Uttar Pradesh).

6.1 Cuckoo Search : Results and Discussion

We have applied our proposed algorithm in possibility of groundwater detection. In our discussed algorithm many regions are considered by our expert. As user input its requirement our detector detects whether such region will have {high, moderate, low} groundwater. Thus software is a total solution for groundwater detection possibility provided all cases are stored in the case base. Already the dataset used is shown in the previous chapter. From the case base provided by our expert we have separated 15 cases for our validation purpose. Our

detector has achieved almost 99% accuracy. Almost all cases are giving accurate results. Cuckoo is a recent technique and it has not shown its footprints in detector mechanism. Thus our proposed algorithm combines all the strength of this metaheuristic algorithm. Its searching is more intensified than other swarm intelligence thus it shows more accurate results than other methods.

The validation dataset has same configuration as the case base. Figure 6.1 shows the validation dataset.

GEOLOGY	LANDFORM	SOIL	LANDUSE	SLOPE	LINEAMENT	POSSIBILITY
2.3	0.1	5	4	1	2	MODERATE
2.3	1	5	4	1	2	MODERATE
3	2.2	2.1	2.4	1	1	HIGH
3	2.2	2.1	2.4	2	1	HIGH
3	2.2	2.1	3	2	1	HIGH
3	2.2	2.1	5	2	1	HIGH
2.4	1	5	2.2	1	1	LOW

Figure 6.1 Validation dataset of groundwater

The values are basically attribute values. In case of the attribute value they are represented in numerical form. The values are already discussed in the previous chapter.

6.1.1 User Interface

The development of graphical user interface is accomplished using Matlab ver 7.0. User is required to enter all the parameters in order to predict the groundwater possibility. Upon execution of our system the first window which appears is shown in figure 6.2

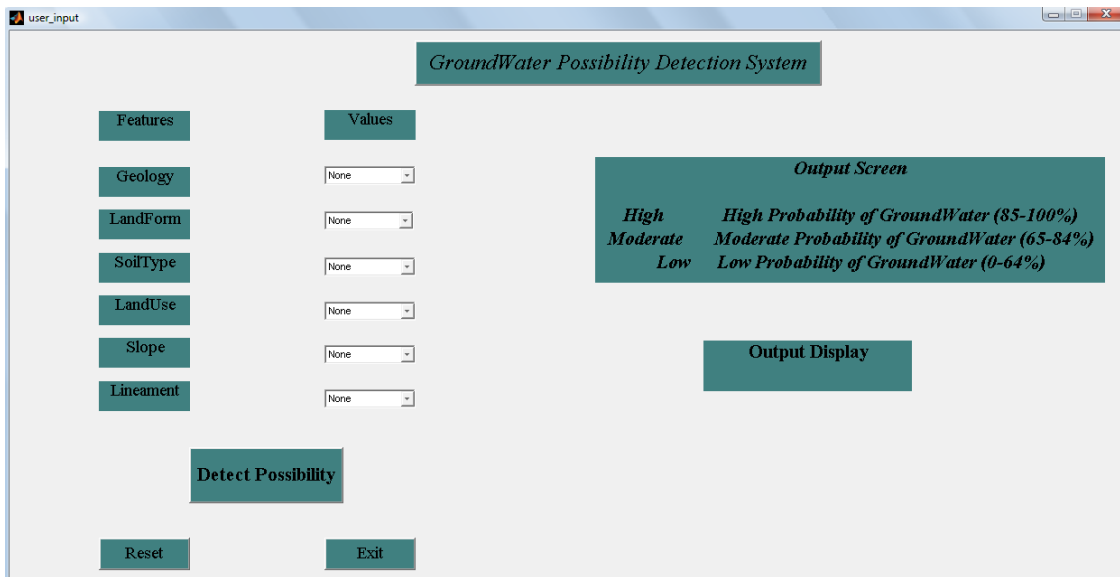


Figure 6.2 First window appear upon execution of the software

It shows all the attributes along with their values. When the button “detect possibility” is pressed then in the text bar “output display” the value {high, moderate, low} is displayed. Thus in the following figure 6.3 it shows attribute along with their values selected by user.

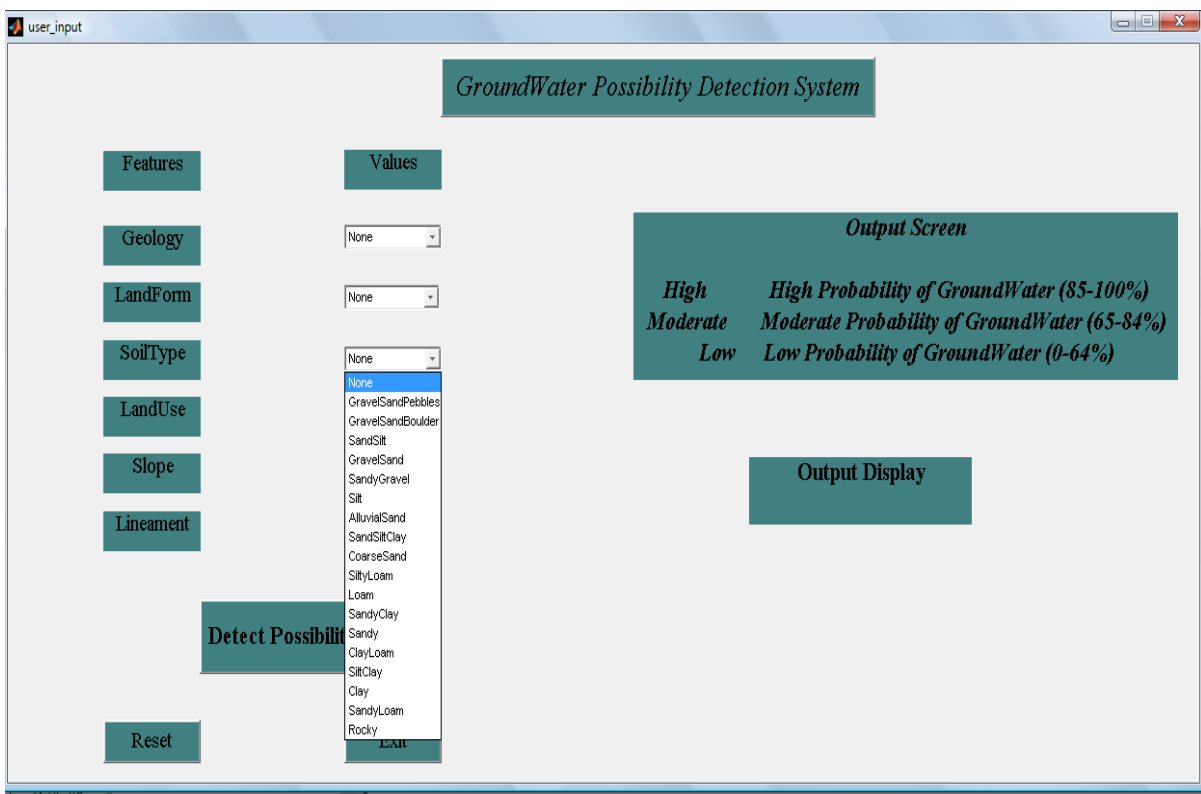


Figure 6.3 Window with attribute value

6.1.2 Results

In GUI, the user has to give all the parameters values to detect groundwater possibility.

Example query entered:

Table 6.1 Query Table

Geology	Younger Alluvium
Landform	Intermontav alley
Soil	Coarse Sand
Land use	Cultivated Land
Slope	Gentle
Lineament	Present
Detect Possibility	?

The figure 6.4 shows the window when user enter all these values to the parameters.

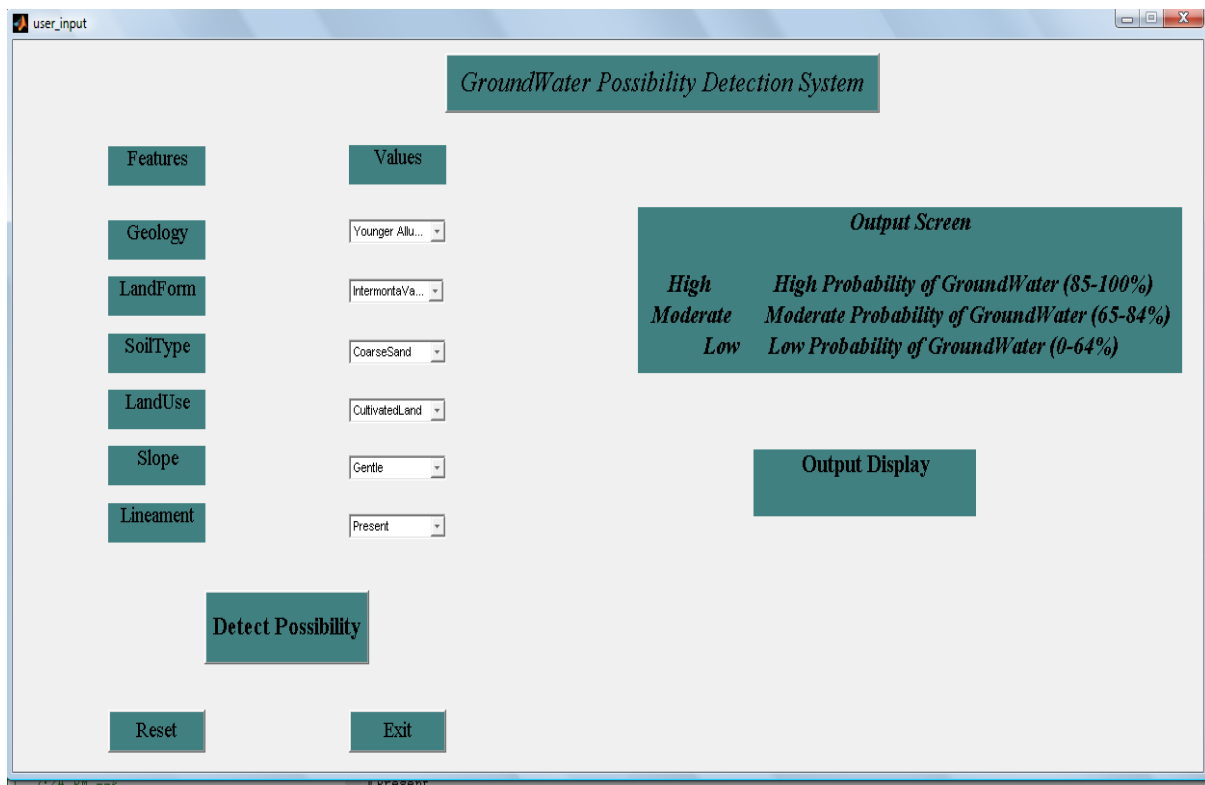


Figure 6.4 Window after values entered by user.

Figure 6.5 shows when user clicked detect possibility button. After clicking output display text bar changes into the possibility value of “High”.

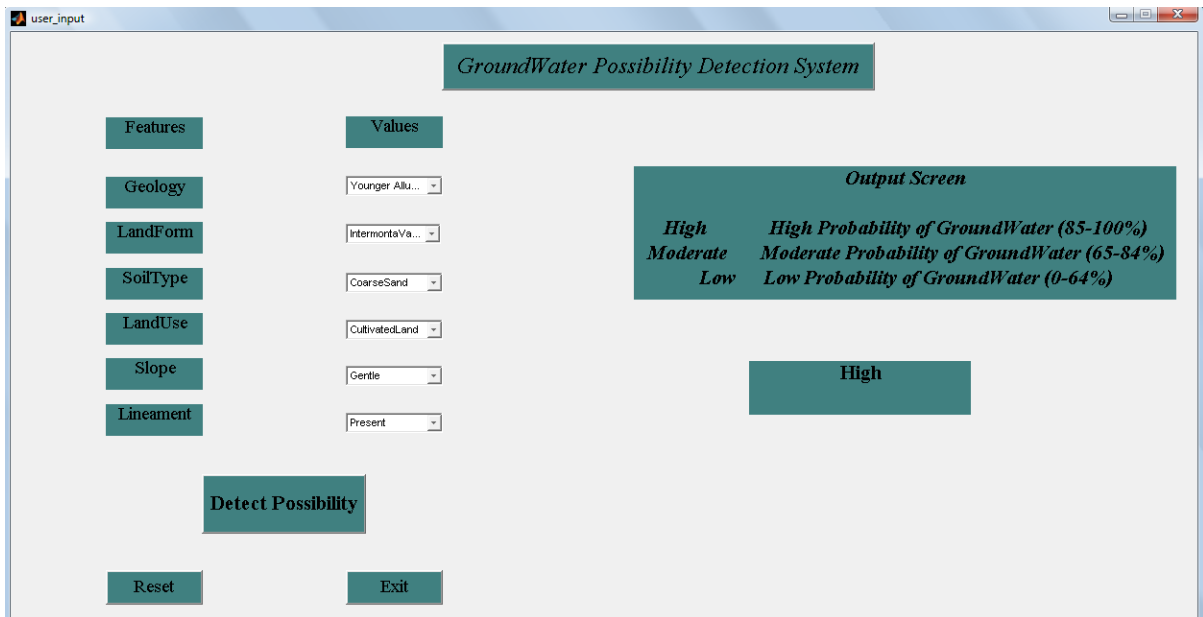


Figure 6.5 Output display changes

Now in the next two figures the purpose of reset button and exit button is described. Figure 6.6 shows the function of reset button. Though all parameter values are reset to initial condition but the text box "output display" is not initialized same. Rather it contains the previous value only.

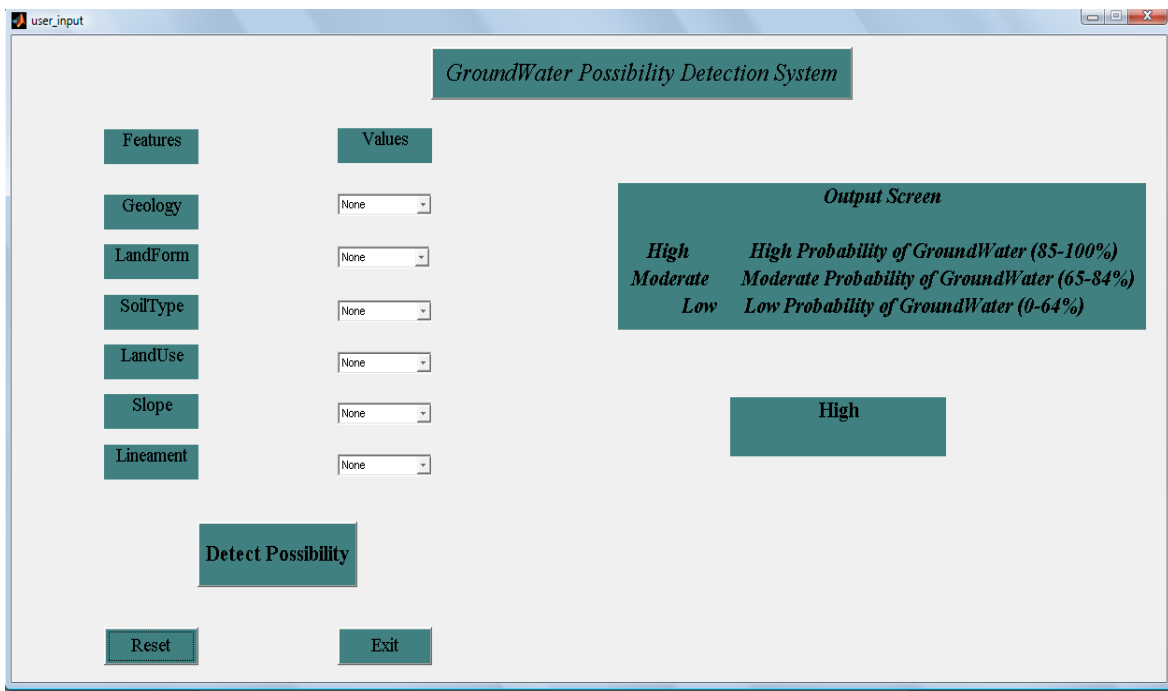


Figure 6.6 Purpose of reset button

When exit button is clicked then the GUI is closed and it goes back to matlab window. Figure 6.7 depicts its function

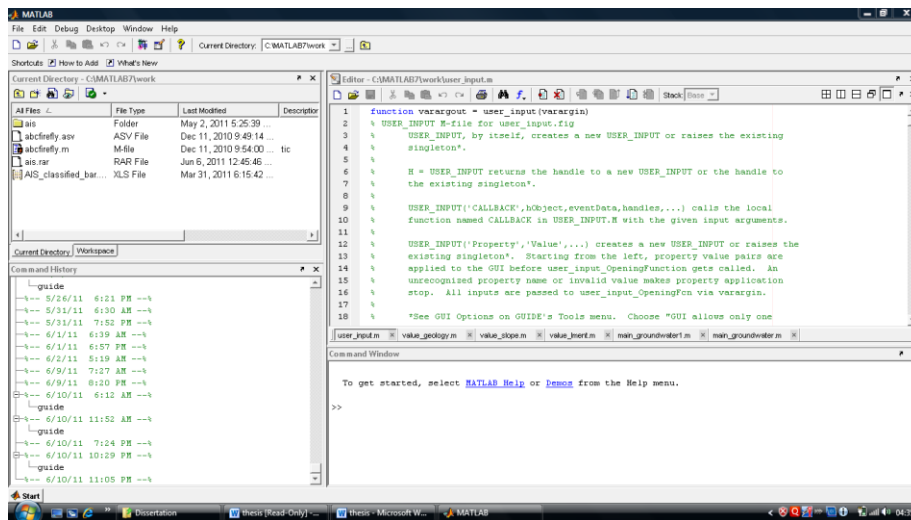


Figure 6.7 Exit button feature

6.1.3 Graphical representation of Cuckoo search in groundwater potentiality

Different query is taken and depending upon the query we will be generating the graphs of cuckoo nests. The output for such query is shown in figure 6.8.

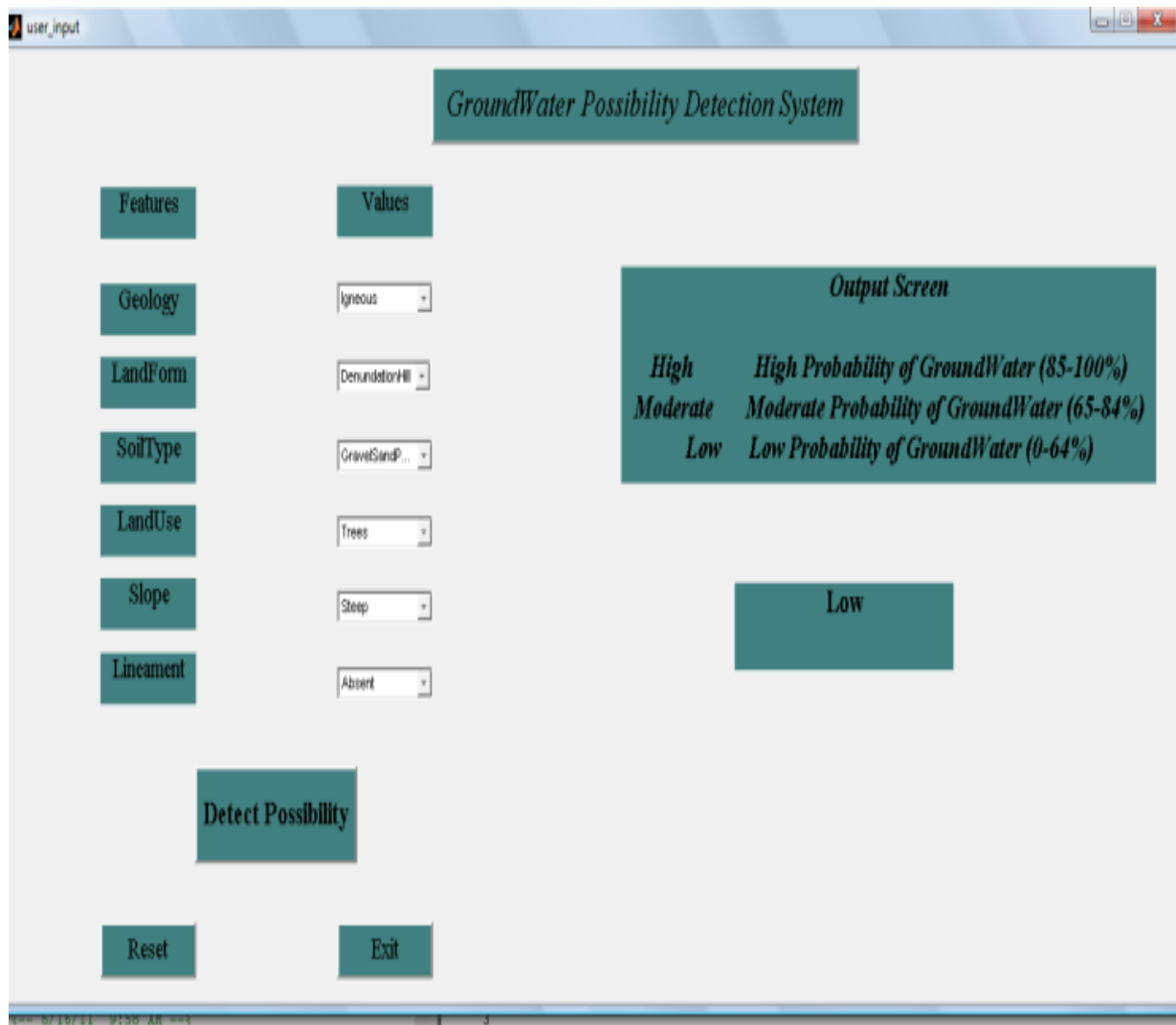


Figure 6.8 Query input by user

When the query is provided by user then the first objective function is evaluated and all the initial nests values are depicted. The X axis shows all the possible nests that cuckoo will search and Y-axis represent their objective function value. They are plotted on the graph according to their values. It is illustrated in figure 6.9.

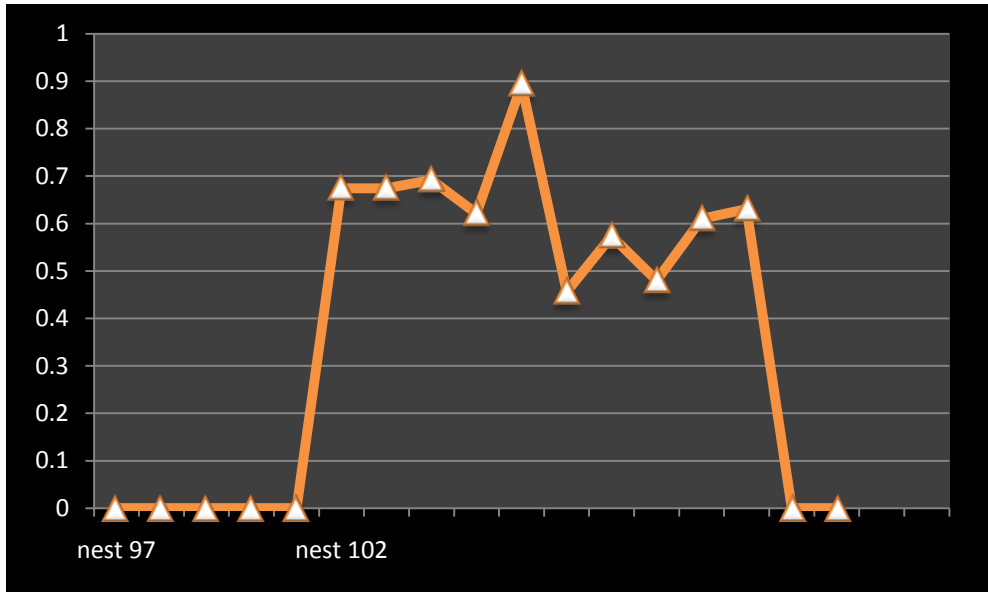


Figure 6.10 Plot of best solutions

It shows that nest number 106 is the best solution. Thus taking the most relevant one we have apply the propositional logic stored in our database. It follows the logic,

P= “geology is igneous”

Q= “lineament is absent”

R= “slope is steep”

S= “groundwater is low”

$$(P \wedge Q \wedge R) \rightarrow S$$

Thus the result is indeed low. Thus our detector has proved almost 99% accurate and gives most efficient output in all cases. Now in Cuckoo search one step is identified as the fraction of worst nests p_a . The formula is already defined in the previous section. Thus from our current query $ignore_case=115-10=105$ and $total_case=115$,

$$P_a = \frac{ignore_case}{total_case}$$

$$P_a = \frac{105}{115} \times 100 = 91.3\%$$

Thus almost 91.3 % are considered as worst nests and their condition hardly matched with cuckoo search. From remaining 8.7% cuckoo has to find the best possible case.

6.2 Membrane Computing Classification: Results and Discussion

We have applied our classification process on different dataset and it had proved that it is a universal classifier. First we have shown our result on Alwar region and then on Saharanpur region. Figure 6.11 shows the original images and classified image of Alwar region. Figure 6.12 shows the original image and classified image of Saharanpur area. Our objective is to use the proposed natural computation algorithm as an efficient landcover classifier for satellite image. Based on the results obtained on applying membrane computing to the 7-band image of Alwar region and 8-band image of Saharanpur region for image classification, we observe that we are able to classify water pixels with the highest efficiency i.e. in case of Alwar almost 99% and in Saharanpur 100% efficiency is achieved and these pixels are best viewed in the NIR and MIR bands.

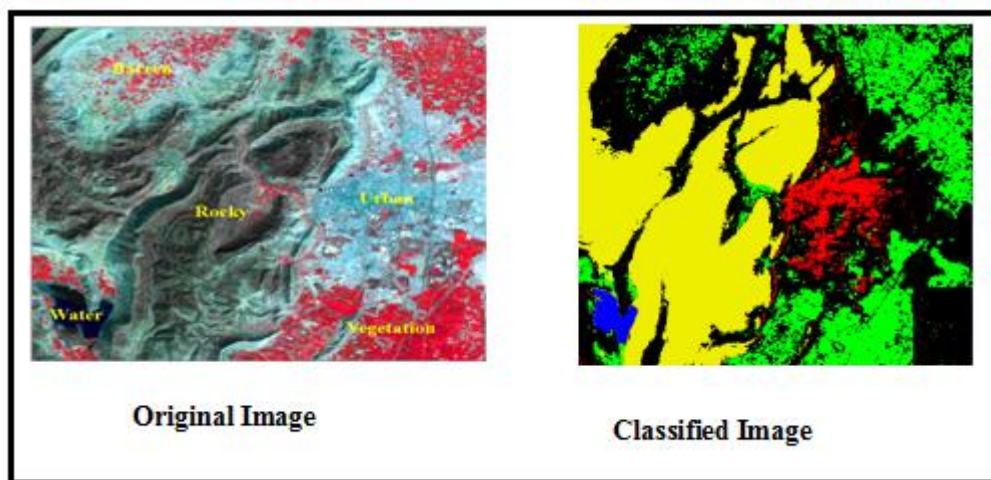


Figure 6.11 Images of Alwar region

Original image depicted the image which is obtained after classification by our geoscientist and classified image is obtained after processing our Universal Classifier. The image obtained after classification has different colour. The yellow, black, blue, green, red colour represents rocky, barren, water, vegetation and urban region respectively. The second image

is of Saharanpur region. Saharanpur has different features. It lies in the state of Uttar Pradesh. It didn't have any rocky region as in the case of Alwar whereas a new kind of feature appeared here i.e. denseurban. The colour codes for Saharanpur region are different. The yellow, green, red, pink and blue represents barren, vegetation, urban, denseurban and water region.

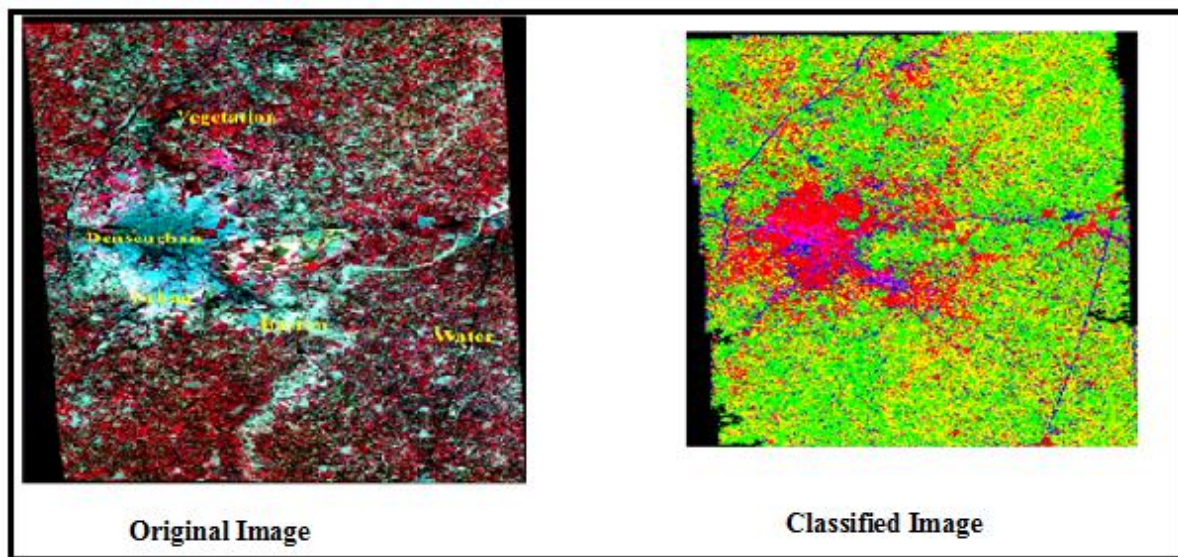


Figure 6.12 Images of Saharanpur region

6.3 Accuracy Assessment of the proposed Algorithm

A classification is incomplete till accuracy assessment is analyzed. Accuracy assessment is used for comparing the classification to geographical data that are assumed to be true, in order to determine the accuracy of the classification process. From ground truth data we get assumed data. Practically it is not feasible to test every pixel of a classified image. A set of reference pixels is used. Reference pixels are points on the classified image for which actual features are (or will be) known. The reference pixels are randomly selected. Accuracy assessment is an important step in the classification process. The goal is to quantitatively

determine how effectively pixels were grouped into the correct feature classes in the area under investigation.

6.3.1 Error Matrix

The aim of this step is to determine quantitatively that how pixels are correctly absorbed in their required features. To determine our accuracy assessment we need to find the error matrix. Error matrices compare, on category-by category basis, the relationship between known reference data (ground truth) and the corresponding results of an automated classification. From our experiments we have come with following number of pixels in each region i.e.

- i) 150 vegetation pixels.
- ii) 190 urban pixels.
- iii) 200 rocky pixels
- iv) 70 water pixels,
- v) 170 barren pixels

The bar graph in figure 6.13 shows that the region is highly concentrated with rocky region, whereas density decreases in case of vegetation, urban and for barren region. Water is sparsely populated in Alwar region.

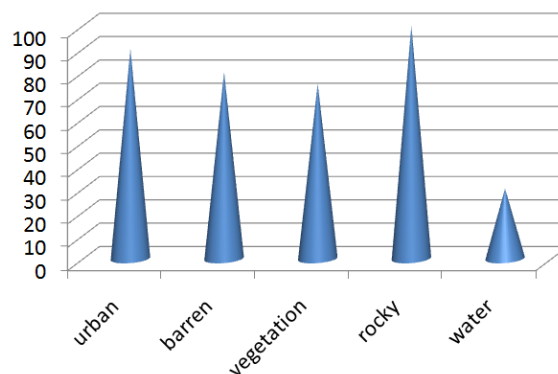


Figure 6.13 Density of each feature in Alwar region

From the training set and the error matrix which is obtained after implementing membrane computing is shown in Table 6.2. The error matrix's interpretation along column suggests how many pixels are classified correctly by our algorithm. The diagonal elements mean no. of properly classified pixels in that category. The error matrix which we have obtained shows high values for vegetation and rocky but urban values need to be explored. Water is almost perfectly classified which can be inferred from our experiment.

Table 6.2 Error Matrix when membrane computing is applied to Alwar region

	Vegetation	Urban	Rocky	Water	Barren	total
Vegetation	146	24	1	0	10	181
Urban	0	24	0	0	2	26
Rocky	0	0	120	0	33	153
Water	0	0	0	69	0	69
Barren	0	55	19	0	89	163
Total	146	103	140	69	134	592

The table shows it has classified the Water pixels with almost 99% efficiency (with minimal omission errors), in case of vegetation we have achieved with minimal omission error (non-ideal classification) thus we can say membrane computing is a good classifier for water and vegetation pixels. For water a zero commission error is achieved whereas vegetation has got commission error of 31 in 181. The graph in figure 6.14 is plotted which represents the percentage on the number of pixels of the image which are classified as water, barren, vegetation, rocky and urban respectively.

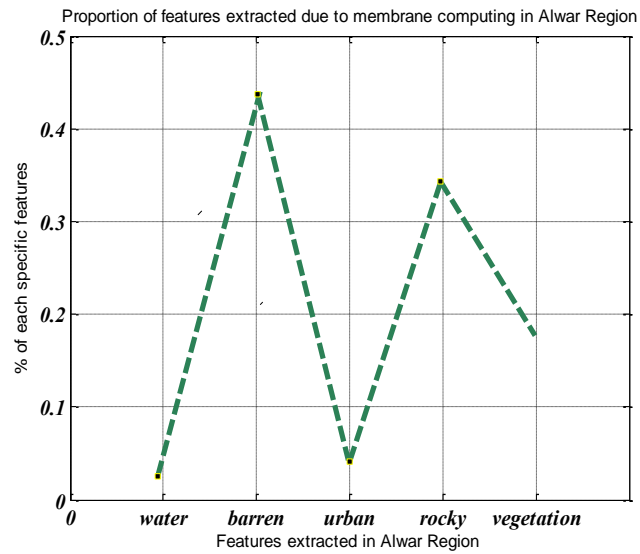


Figure 6.14 Plot of terrain features of Alwar region

In case of Saharanpur region we took 30 barren pixels, 30 denseurban pixels, 30 urban pixels, 118 vegetation pixels, 20 water pixels from the training set and the error matrix obtained shown in Table 6.3.

The error matrix along the diagonal shows that how many pixels are correctly classified. Table 6.3 shows that whole 20 out of 20 pixels are correctly classified as water. Thus we achieved zero omission error (ideal classification).

Table 6.3 Error matrix when Membrane computing is applied to Saharanpur region

	Barren	Denseurban	Urban	Vegetation	Water	Total
Barren	30	0	0	0	0	30
Denseurban	0	6	0	0	24	30
Urban	0	6	24	0	0	30
Vegetation	0	0	0	111	7	118
Water	0	0	0	0	20	20
Total	30	12	24	111	51	228

Though table 6.3 shows how our classifier classifies each feature perfectly whereas figure 6.15 tried to give it a graphical view. As Saharanpur region have 5 features thus we had shown all features along x-axis and number of pixels had been classified along y-axis. The blue bar represents number of pixels classified correctly by our classifier whereas the red bar depicts the sample training set that had been given to us by our geology expert.

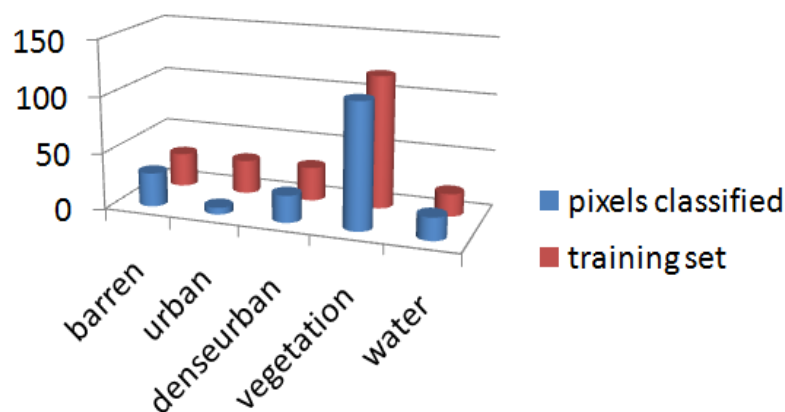


Figure 6.15 Comparison study between data given by expert and result obtained through our classifier

6.3.1.1 Producer's Accuracy

Several other descriptive measures can be obtained from error matrix. The accuracy of individual category can be calculated by dividing the number of correctly classified pixels in each category by either the total number of pixels in corresponding row or column. Producer's accuracies(as shown in table 6.4) result from dividing the number of correctly classified pixels in each category (on the major diagonal) by the number of training set pixels used for that category(the column total). This figure indicates how well the training pixels of a given cover type are classified. The results in the table 6.4 show that water have been extracted perfectly followed by rocky pixels which show 99% efficiency.

Table 6.4 Producer's Accuracy for Alwar area

Feature	Accuracy Calculation	Producer's Accuracy
Vegetation	146/150	97%
Urban	24/190	12%
Rocky	120/200	60%
Water	69/70	98.5%
Barren	89/170	52%

The result shows that our classifier is perfectly good for vegetation and water body but needs to be improved for urban and barren area. Thus to check whether our classifier gives approximate result for other area we have applied it on Saharanpur region and its producer's matrix quite approachable.

Table 6.5 Producer's Accuracy of Saharanpur area

Feature	Accuracy Calculation	Producer's Accuracy
Barren	30/30	100%
Denseurban	6/30	20%
Urban	24/30	80%
Vegetation	111/118	94%
Water	20/20	100%

The result shows again that even in water region highest efficiency followed by vegetation achieving almost 99% accuracy is reached in these case even barren has good efficiency but denseurban needed to be considered for further analysis.

6.3.1.2 User's Accuracy

User's Accuracies (as shown in following table) is computed by dividing the number of correctly classified pixels in each category by the total number of pixels that were classified in that category (the row total). This figure is a measure of commission error and indicates the probability that a pixel classified into a given category actually represents that category on the ground.

Table 6.6 User's Accuracy for Alwar region

Feature	Accuracy Calculation	User's Accuracy
Vegetation	146/181	80%
Urban	24/26	92.3%
Rocky	120/153	78.4%
Water	69/69	100%
Barren	89/163	54.6%

It shows that commission error is high in case of barren region. Thus it can be further considered for our future work. Next we show our accuracy matrix for Saharanpur region.

Rather we have achieved low commission error in this area.

Table 6.7 User's Accuracy for Saharanpur region

Feature	Accuracy Calculation	User's Accuracy
Barren	30/30	100%
Denseurban	6/30	20%
Urban	24/30	80%
Vegetation	111/118	94%
Water	20/20	100%

6.3.2 KHAT Statistics

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 \hat{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, \hat{k} can be defined as

$$\hat{k} = \frac{\text{observed accuracy} - \text{chance agreement}}{1 - \text{chance agreement}} \quad \dots(6.1)$$

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, its value approaches 1. This is the ideal case. In reality, its value usually ranges between 0 and 1. For example, a value of 0.98 of KHAT statistics can be thought of as an indication that an observed classification is 98 per cent better than one resulting from chance. A KHAT coefficient with the value of 0 suggests that a given classification is no better than a random assignment of pixels. In cases where chance agreement is large enough, KHAT coefficient can take on negative values-an indication of very poor 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 principle advantage of computing KHAT coefficient is the ability to use this value as a basis for determining the statistical significance of any matrix or the differences among matrices. The KHAT coefficient incorporates the non-diagonal elements of error matrix (and hence error of omission and commission) as a product of the row and column marginal.

The KHAT statistic is computed as:

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

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 observation in row i

x_{+i} = total of observation in column i

N = total number of observations included in matrix

For Alwar region

To illustrate the computation of KHAT for the error matrix in table 6.2,

$$\sum_{i=1}^r x_{ii} = 146 + 24 + 120 + 69 + 89 = 448$$

$$\sum_{i=1}^r x_{i+} \cdot x_{+i} = (181 \cdot 146) + (26 \cdot 103) + (153 \cdot 140) + (69 \cdot 69) + (163 \cdot 134) = 77,127$$

$$\text{'k' kappa} = \frac{592 \cdot 448 - 77,127}{592 \cdot 592 - 77,127}$$

$$= 0.68812.$$

Hence the KHAT statistics of the proposed membrane computing based land cover feature extractor (or the universal classifier) is 0.68812. This value is a substantial agreement between the ideal classifier and the proposed algorithm. The Kappa (K) coefficient of the Alwar image is 0.68812 which indicates that an observed classification is 68.812% better than one resulting from chance.

For Saharanpur region

To illustrate the computation of KHAT for the error matrix in table 6.3,

$$\sum_{i=1}^r x_{ii} = 30 + 6 + 24 + 111 + 20 = 191$$

$$\sum_{i=1}^r x_{i+} \cdot x_{+i} = (30*30) + (12*30) + (30*24) + (111*118) + (51*20) = 16,098$$

$$\text{'k' kappa} = \frac{228*191 - 16,098}{228*228 - 16,098}$$

$$= 0.7649.$$

Hence the KHAT statistics of the proposed membrane computing based land cover feature extractor (or the universal classifier) is 0.7649. This value is a substantial agreement between the ideal classifier and the proposed algorithm. The Kappa (K) coefficient of the Saharanpur image is 0.7649 which indicates that an observed classification is 76.49% better than one resulting from chance.

6.4 A Comparative Study with other classifiers

One of the important characteristic features of membrane computing is that it works on single unit of object rather than group of objects. Consequently the clarity observed using membrane computing is much clearer than other classifiers because most of them use certain clustering mechanism. Like BBO uses fuzzy-c-means cluster to group all the similar pixels together. Though our classifier has not reached highest efficiency as other soft computing techniques but it has achieved another wider area of research i.e. recognition of heterogeneity in an image and even analysing the feature that heterogeneous region is possessing. Till date all the classifiers only classifies a given image and achieves efficiency but they do not focuses on region where mixture of many features are possible. Though they classify homogeneous region perfectly but they do not have any impact on heterogeneous portion of the image. In our Alwar image lot of area has heterogeneous region. The following figure 6.16 shows a portion where both probability of water and vegetation area is high.

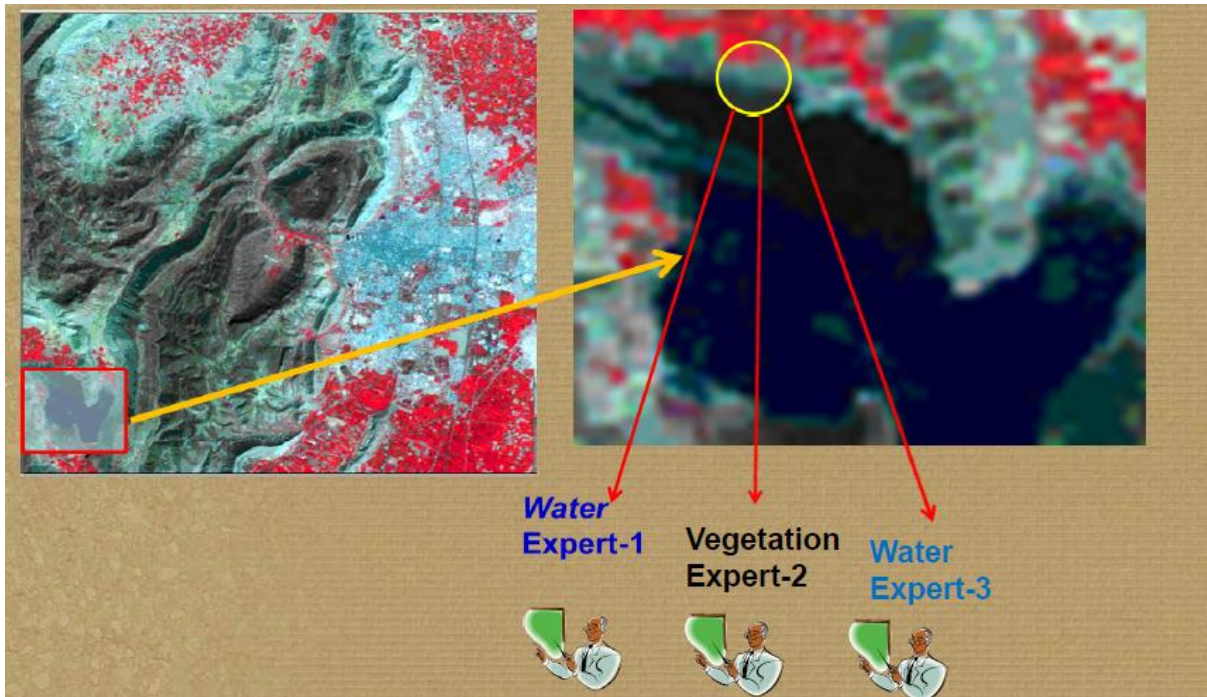


Figure 6.16 Conflict portion of the image

This image has a portion near the water body region where at the edge of water body both probability of water and vegetation is mixed and experts have conflicts to decide about the assignment of this heterogeneous portion. Basically this conflict arises when spectral signatures become similar for different features as shown in figure 6.17.

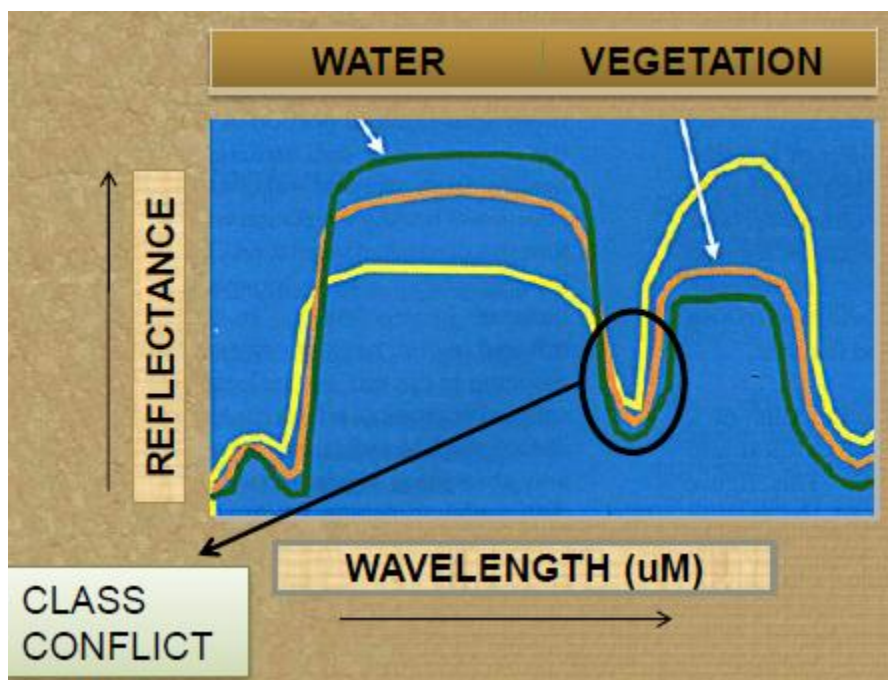


Figure 6.17 Mixture of spectral signature

We have gone through all other classifiers and had seen that some classifiers assigned this portion rocky and some purely vegetation. But the image shows it does not lie in any of the above mentioned feature.

Our classifier has observed these region minutely and seen that it is neither water nor rocky nor vegetation area. Rather it is the conflict dataset match has shown a strong correlation with barren dataset. Thus our Universal Classifier has identified this heterogeneous portion and classified as barren.

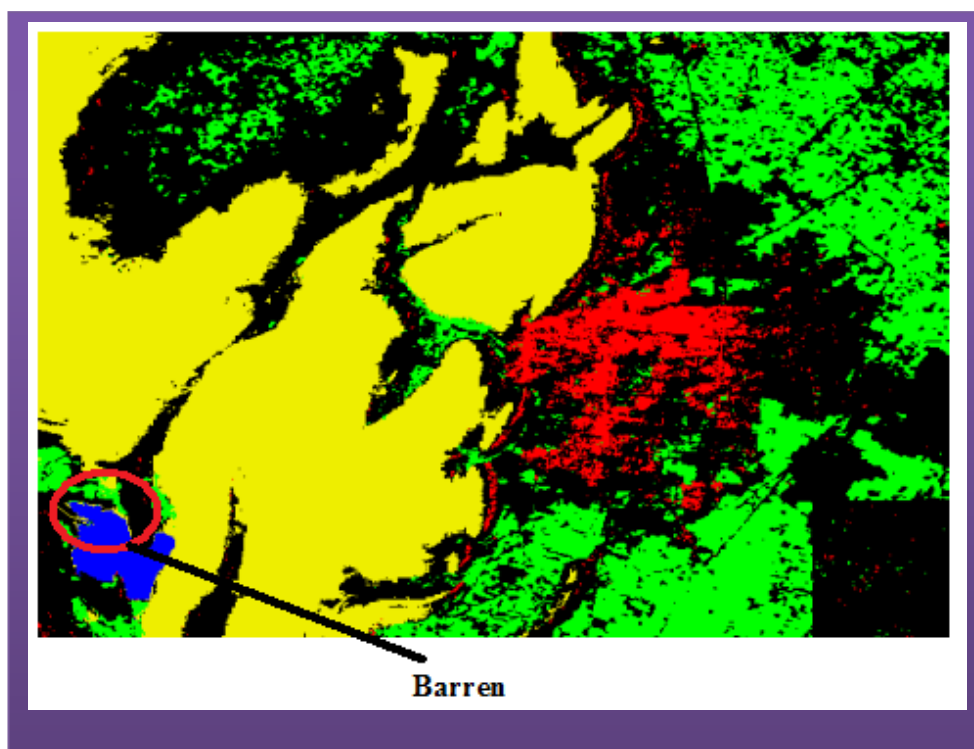
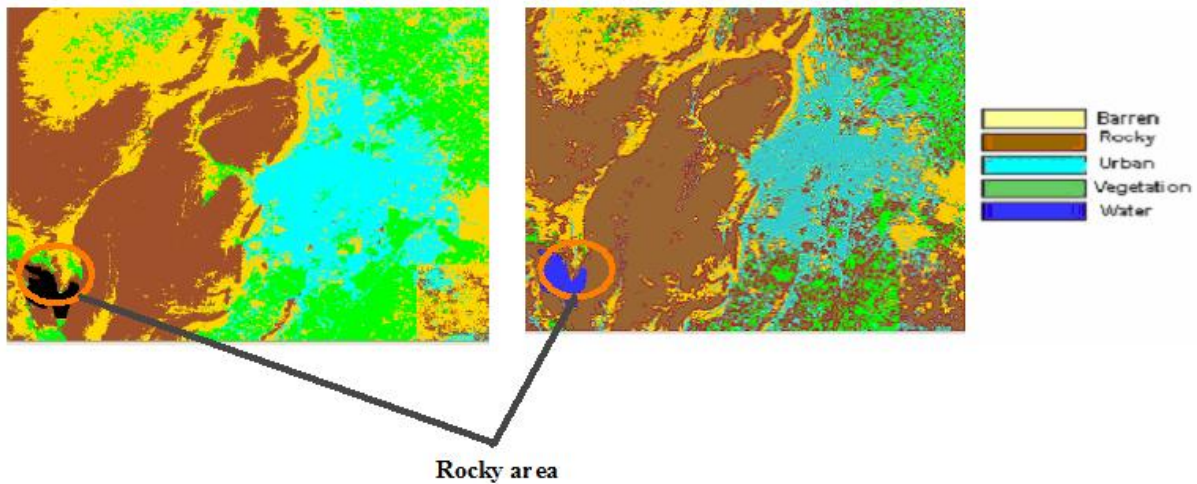


Figure 6.18 Identified heterogeneity and assigning proper feature

Considering our traditional classifier, Minimum distance classifier and Maximum Likelihood classifier.

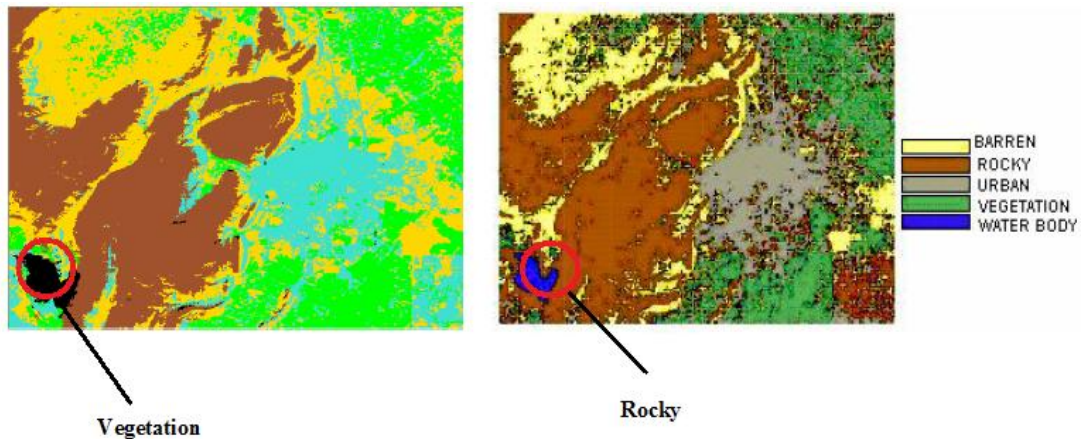


(a) Minimum distance classifier (b) Maximum Likelihood classifier

Figure 6.19 A view of traditional classifier

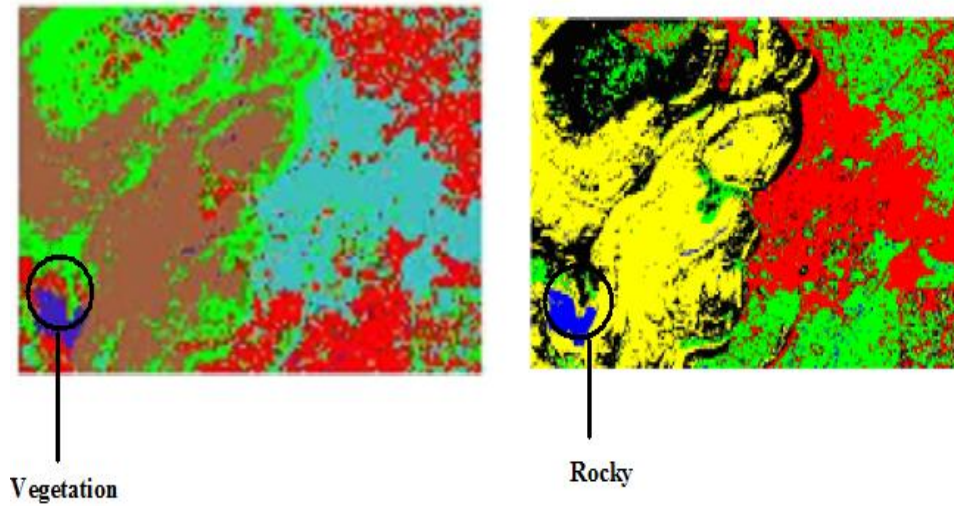
If original image is considered then it can be inferred that this both encircled portion is obviously not rocky area. Though both of the classifier has high Kappa coefficient i.e. for Minimum distance it is 0.7364 and for Maximum likelihood it is 0.7525 but though these heterogeneous region is wrongly classified. However, it shows our Universal classifier has shown comparable results compared to other soft computing technique. The Satellite image classification results of the recent soft computing classifiers when applied on the same Alwar Image, are shown below. For the purpose, Rough Sets, Fuzzy Sets, Rough-Fuzzy Tie-up, Ant Colony Optimization, Particle Swarm Optimization and Biogeography Based Optimization methods and their hybrids such as the hybrid ACO-BBO Classifier, Rough –Fuzzy Tie Up and hybrid ACO2/PSO Classifier are analysed. Also, semantic-based image classification is added, as a special instance. Fig 6.20(a) shows the Fuzzy Classification of Alwar region which has a Kappa –Coefficient of 0.9134. Fig 6.20 (b) presents the results of an integrated Rough –Fuzzy Tie Up Approach which has a Kappa Coefficient of 0.9700. Fig 6.20 (c) applies the cAntMiner Algorithm on the Alwar Region which has a Kappa Coefficient of 0.964. Fig 6.20(d) shows the result of applying the hybrid ACO-BBO Technique on the Alwar Image which has a Kappa-Coefficient of 0.96699. Fig 6.20 (e) applies the Hybrid

ACO2/PSO Classifier which has a Kappa Coefficient of 0.975. Fig 6.20 (f) presents the results of the Semantic Web Based Classifier on the image with a Kappa Coefficient of 0.9881 and fig 6.20 (g) shows the result after applying hybrid ACO2/BBO/PSO with Kappa coefficient of 0.9812.



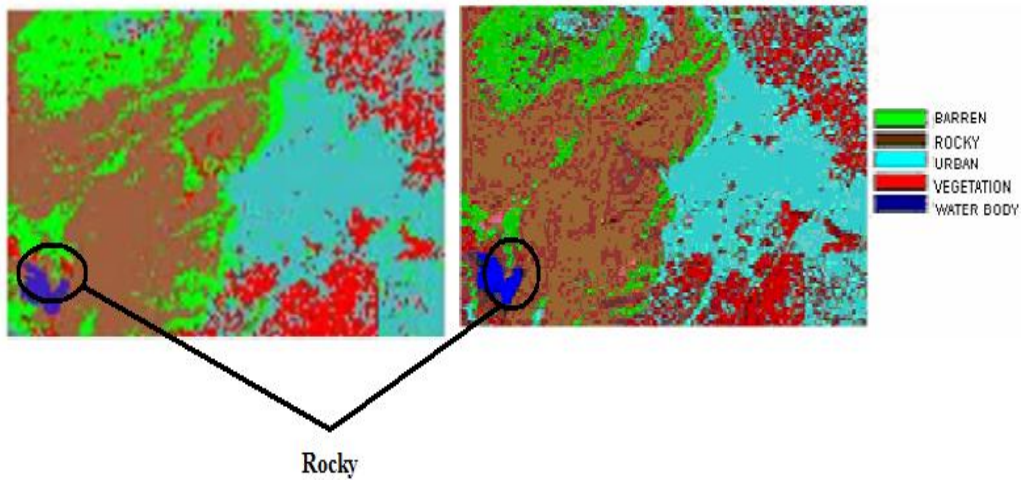
(a) Fuzzy Classification

(b) Rough-Fuzzy tie up



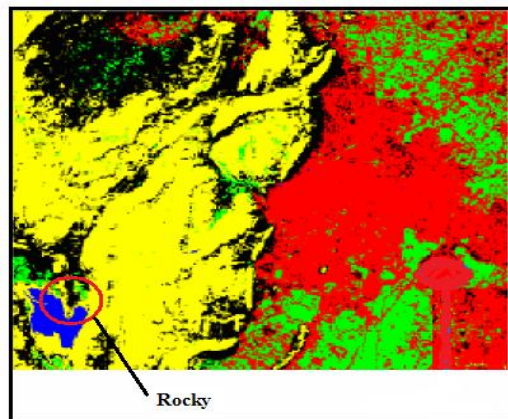
(c) cAntMiner Algorithm

(d) Hybrid ACO-BBO Algorithm



(e) Hybrid ACO2/PSO

(f) Semantic Web based classifier



(g) Hybrid ACO/PSO/BBO Algorithm

Figure 6.20 Classified alwar image after applying various soft computing techniques

Thus our universal classifier has proved that it observed the heterogeneity portion of the image and even recognized the correct feature of that portion. Though its Kappa coefficient is less than other soft computing technique but it recognizes all the dissimilar portion in the image.

7 PUBLICATION FROM THESIS

“If you can dream it, you can do it”

Walt Disney

During the period of working over this project we interacted with International community working on Artificial Intelligence , in particular the founder of the key technique addressed in the thesis, (called the Membrane Computing) Gheorghe Păun (Professor, Cleveland State University). We discussed our approach for representing knowledge with him and collected reviews and worked over the suggestions sent to us. Two research papers have been accepted in International conferences/journals for presentation and also published in their proceedings. First paper focuses on proposing a Membrane Computing in Image Classification only in water region. It is blueprint to see whether membrane computing can be applied to feature extraction and we came across positive result. This paper can be of major benefit to the all the geoscientist because they basically need to have a knowledge of the area on which they are working and is also under practical implementation at DTRL, DRDO. The second paper embeds a methodical study of the same P system in satellite image classification. This paper can provide a new direction to membrane computing as hardly any image classification has any impact in P system. Even a Ph.d study is studying in DRDO and she goes through this paper for her research work.

7.1 Details of Conference:

1. Conference Name: 5th International Multi Conference on Intelligent Systems & Nanotechnology (IISN-2010) (**Published**).

URL: <http://www.istk.org>

Paper Title: “Membrane computing for water body area extraction in satellite image”

Authors: Bidisha Das, V. K. Panchal, Nitish Gupta, Vartika Singh.

Location: Klawad, Haryana.

Conference date: February 18-19, 2011

Publisher/Proceedings: The paper is included in the conference proceedings, which has an ISBN number. The proceedings will also be submitted for several database indexes.

2. Conference Name: The 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing (**Accepted**), The 2011 International Conference on Genetic and Evolutionary Methods (GEM'11).

URL: <http://www.world-academy-of-science.org/worldcomp11/ws>

Paper Title: “A methodical study for the extraction of landscape traits using membrane computing technique”

Authors: Daya Gupta, Bidisha Das, V. K. Panchal.

Location: Las Vegas, Nevada, USA.

Conference date: July 18-21, 2011

Publisher/Proceedings: The paper is included in the conference proceedings, which has an ISBN number. The proceedings will also be submitted for several database indexes.

8 CONCLUSION AND FUTURE SCOPE

“A fool thinks himself to be wise, but a wise man knows himself to be a fool”

William Shakespeare

Preview

This chapter is the last part of this thesis thus it provides an overall overview of our proposed work in different field of geoscience. Basically in both conclusion and future scope section first part deals or discusses bio-inspired groundwater potentiality detector and about our Universal Classifier for images.

8.1 Conclusion

The proposed framework of groundwater detection is another area of natural computation that we have implemented. It has been proved as a groundwater detector with 99% efficiency. The swarm intelligence that currently emerged known as cuckoo search is considered along with case based reasoning. This metaheuristic algorithm had been proved to be better searching algorithm than other swarm intelligence technique. Thus our proposed methodology is able to detect the presence of groundwater at inaccessible areas through remote sensing. Thus traditional methods difficulties and disadvantages are overcome. Thus here the host nests are considered as single case in a case base. We are provided by our experts with huge case base. Our software predicted the presence of groundwater in a region as provided by user

query. The system aimed at answering whether the queried region is having low, high or moderate groundwater. Case retrieval method is an intelligent method which involves retrieving the most similar previous cases to provide a solution to a new decision problem. The Case base is designed with the help of a groundwater monitoring experts. Using the system, it is now possible to undertake a regional hydro geological survey to map the distribution of groundwater reserves both in plan and in depth with high efficiency, thus ensuring a reliable choice of the most productive areas for water supply bores. The system has been developed which is capable of determining groundwater content at various depths without drilling wells. The system developed is of great significance as the presence or absence of ground water in a given area has a direct effect on its real estate values. So the system can play a crucial role in economic sector of a nation. Moreover, it is very useful in military applications during the time of combats to find groundwater possibility in inaccessible areas like areas across the border of a nation. Movement of troops depends largely on such information as water is a basic need for survival

A large number of soft computing techniques have been already applied to remote sensing satellite image. All these techniques have efficiently classified the geospatial features of the terrain despite its discrepant uncertainties. Membrane computing is an emerging area which had hardly has any impact on image classification. Thus terrain features classification is taken as a case study. In this proposed work algorithm we represent an efficient land cover classifier for the satellite image. Many of the land cover features are identified much more clearly when proposed algorithm is used. By using this approach we are able to classify the satellite image according to different areas like water, urban, vegetation, barren and rocky region with different colors assigned to each feature's pixels. It is perceived that the Kappa coefficient can be considered as a well-founded metric for assessing accuracy of classification in remote sensing. A novel approach for feature extraction from high resolution

multi- spectral satellite image is presented in this paper. From the Kappa coefficient it has been observed that vegetation region has been classified with almost zero omission error (columnwise) which lacks by only 4 pixels. Membrane computing has achieved almost 99% omission error in water region and it has been proved that it is equally at par with other soft computing technique. As membrane computing follows maximal parallelism, even the time complexity is reduced than other soft computing algorithm. Membrane computing not only identified homogeneous region but also considered heterogeneous regions. Thus it is classifier that not only classifies images but also identifies region with heterogeneity.

8.2 Future Scope

Groundwater detection mechanism can improve its searching mechanism to 100% efficiency if huge amount of case base is provided by experts. However, one difficult part of this research is to find sufficient amount of cases. Most of the domain experts are not willing to provide their own know-how since this may cause losing their jobs. But, fewer cases will result in poor accuracy. If the mechanism of this study can be applied in practice to deal with more than thousand cases, the efficiency can be improved. Future work can be done by experimenting with other categorical datasets to explore the fruitful results that can be applicable in other fields.

The future scope of the research includes proposing certain modification to the algorithm so that the Kappa coefficient can be improved further. The current system is implemented using a simple heuristic technique namely correlation function; the system performance can be increased by using other heuristic functions. The system performance can be further increased by using better unsupervised classifications and better training sets. Moreover it had identified most of the features but urban needed to be identified properly. Lot of urban areas are identified as barren thus lowering its Kappa coefficient. Moreover only one part of

the whole image is identified as heterogeneous region. The image also has other parts also that possess heterogeneity. It can be combined with other bio-inspired algorithm to better classify the image. Saharanpur is an urban growth area. Lot of research work is going on in this area. Researchers are predicting the direction of urban growth. Thus comparing this image with some previous image having same dimension and same latitude, longitude values we can predict the direction. Though neural network has already been applied for urban growth prediction but membrane computing is a new technology in this field.

9 REFERENCES

- [1] L. Adleman, “Computing with DNA”, *Scientific American*, vol. 279, no. 2, August 1998.
- [2] Chunhua Yang, Hongqui Zhu, Weihua Gui, “Permeability prediction model for imperial smelting furnace based on improved case based reasoning”, *IEEE Proceeding of the 9th World Congress on Intelligent control & Automation*”, Chongqing, June 25-27, 2008.
- [3] www.britannica.com/EBchecked/topic/247019/groundwater.
- [4] Zhou, Wei, “Survey and analysis of land satellite remote sensing applied in highway Transportation, Infrastructure and System Engineering”, *IEEE International Geoscience and Remote Sensing Symposium*, 2008.
- [5] T.M. Lillesand, R.W. Kiefer & J.W. Chipman, “Remote Sensing & Image Interpretations” 5th edition, Wiley & Sons Pvt. Ltd., pp 586-592, 2005, England.
- [6] I. Pufford, H. Flowers, “Environmental chemistry at a glance”, Great Britain, Wiley Blackwell, pp 46-47, 2006
- [7] Xin-She Yang, Suash Deb, “Cuckoo Search via Lévy Flight”, In *Proceedings of World Congress on Nature & Biologically Inspired Computing, NaBIC*, IEEE Publisher (USA), pp 210-214, Dec. 2009, India.
- [8] Gheorghe Păun, Grzegorz Rozenberg, “An Introduction to & an Overview of Membrane Computing”, Chapter 1.

- [9] Sabbar Abdallah Salih, "Application of Groundwater penetrating Radar (GPR) in detection of groundwater table", Proceedings of 10th International Conference on Ground Penetrating Radar, pp 221-224, Netherland, June, 2004.
- [10] V.K.Panchal, Harish Kundra, Navpreet Kaur, "A novel Approach to Integration of waves of swarms with Case Based Reasoning to detect Groundwater potential", 8th Annual Asian Conference & Exhibition of geospatial Information Technology & Application, Map Asia, Singapore, 2009.
- [11] V.K.Panchal, Er. Harish Kundra, Amanpreet Kaur, "Biogeography Based Groundwater Exploration", International Journal of Computer Application, (IJCA), vol. 1, no. 8, 2010.
- [12] Xin-She Yang, Suash Deb, "Engineering Optimization by Cuckoo search", International Journal Mathematical Modelling & Numerical Optimization, vol. 1, no.4, pp 330-343, 2010.
- [13] L.A. Zadeh, "Fuzzy sets", Information & Control, vol.8, no.3, pp 338-353, June,1965.
- [14] Z. Pawlak, "Rough Set Theory & its Application to Data Analysis", Cybernetics & Systems, vol. 29, no. 7, pp 661-688, 1998
- [15] E. Bonabeau, M. Dorigo, G. Theraulaz, "Swarm Intelligence from Nature to Artificial System", First Edition, Oxford University Press, pp 1-24, USA, 1999.
- [16] S.N.Omkar, K.M. Manoj, D. Mudigree & D. Muley, "Urban Satellite Image Classification using Biologically Inspired techniques", In Proceedings of IEEE International Symposium on Industrial Electronics, Vigo, Spain, pp. 1767-1772, 2007.
- [17] Daya Gupta, Lavika Goel, V.K.Panchal, "Embedding Expert Knowledge to Hybrid Bio-Inspired techniques- An Adaptive Strategy towards Focussed Land Cover feature Extraction", International Journal of Computer Science & Information Security, (IJCSI), vol.8, no.2, 2010.

- [18] C.K. Riesbeck, R. C. Schank, "Inside case based reasoning", L. Erlbaum Associates Inc. Hillsdale, 1989, New Jersey, ISBN: 0898597676.
- [19] I. Watson, "Applying case based reasoning: Techniques for enterprise system", The Morgan Kaufmann Series in Artificial Intelligence, 1997, California, USA.
- [20] C.J.Price, I. Pegler, "Deciding parameters values with case based reasoning", In I. Watson (eds), Progress in Case Based Reasoning, Lecture Notes in Artificial Intelligence,1020 Springer, 1995, Berlin.
- [21] R. Heider, "Troubleshooting CFM 56-3 engines for the Boeing 737 using case based reasoning and data mining", In I.Smith, B. Faltings(eds), Advances in case based reasoning, lecture notes in Artificial Intelligence,1168 Springer, pp 513, 1996, Berling.
- [22] J. Mendel, "Fuzzy logic systems for engineering: a tutorial", Proceedings of IEEE, vol. 83, no. 3, pp. 345-377, 1995.
- [23] I.Watson, "Case based reasoning is a methodology not a technology", Journal of Elsevier, Knowledge based systems, vol.12, pp 303-308, 1999.
- [24] C. Blum, A. Roli, "Metaheuristics in combinatorial optimization: Overview & conceptual comparison", ACM Computational Survey, vol. 35, pp. 268-308, 2003.
- [25] "An Introduction to Remote Sensing & Geographic Information System", Yale University, Genocide Studies Project, Remote Sensing & Geographic Information System Research.
- [26] Sonal Kumar, Daya Gupta, V.K.Panchal, Shashi Kumar, "Enabling web services for classification of satellite images", In International Conference on Semantic Web & Web Services, SWWS, July 13-16, 2009, Orlando.
- [27] Daya Gupta, J. Kartar, "Knowledge Representation with Ontology & Relational Database to RDF Converter", In Proceedings of Enterprise Information Systems & Web Technologies, pp 141-146, 2008.

- [28] Lila Kari, Grzegorz Rozenberg, “The Many Facets of Natural Computing”, Communication of the ACM, vol.51, no.10,Oct. 2008.
- [29] Shelly Bansal, Daya Gupta, V.K.Panchal, Shashi Kumar, “Swarm Intelligence Inspired Classifiers in Comparison with Fuzzy & Rough Classifiers – A Remote Sensing Perspective”, International Conference on Contemporary Computing, IC3,2009, Noida,India.
- [30] Van Nguyen, David Kearney, Gianpaolo Geosia, “Balancing Performance, Flexibility & Scalability in a Parallel Computing Platform for Membrane Computing Application”, In: Eleftherakis, G. , Kefalas, P., Păun, G., Rozenberg, G., Salomaa, A. (eds) WMC, 2007, LNCS, vol. 4860, pp. 385-413, Springer, Heidelberg, 2007.
- [31] Gheoghe Păun, Radu A. Păun, “Membrane Computing as a Framework for modelling economic processes”, IEEE Proceedings of the 7th International Symposium on Symbolic & Numeric Algorithm for Scientific Computing, SYNASC, 2005.
- [32] Bidisha Das, Nitish Gupta, V.K.Panchal, Vartika Singh, “Membrane Computing for water body area extraction in satellite image”, 5th International Multi-Conference on Intelligent Systems & Nanotechnology, IISN, February, 2011, Klawad, India.
- [33] Daya Gupta, Bidisha Das, V.K.Panchal, “A methodical study for the extraction of Landscape Traits using Membrane Computing”, The 2011 International Conference on Genetic & Evolutionary Methods, (GEM), 2011, Las Vegas, USA, (accepted).
- [34] MATLAB, <http://www.mathworks.com/>.

APPENDIX A

Abbreviations

EM – Electromagnetic

DN – Digital Number

BBO – Biogeography Based Optimization

MDC – Minimum Distance Classifier

MLC – Maximum Likelihood Classifier

SC – Soft Computing

RS – Rough set theory

SVM – Support Vector Machine

MIR – Middle Infra-Red

RS1 – Radarsat-1

DEM – Digital Elevation Model

RS2 – Radarsat2

LS-SVM – Least square support vector machine

MCPSO – Membrane Computing Particle Swarm Optimization

MC – Membrane Computing

ACO – Ant Colony Optimization

PSO – Particle Swarm Optimization

CS – Cuckoo Search

SI – Swarm Intelligence

FL –Fuzzy Logic

NIR – Near Infra-Red

CBR – Case Based Reasoning

GUI – Graphical User Interface

APPENDIX B

An Introduction to MATLAB

URL: <http://mathworks.com/>

MATLAB is a high performance language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

1. Math and computation
2. Algorithm, simulation and prototyping
3. Modeling, simulation and prototyping
4. Data analysis, exploration and visualization
5. Scientific and engineering and visualization
6. Application development, including graphical user interface building

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for Matrix Laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today MATLAB uses software developed by the LAPACK and ARPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering and science. In industry, MATLAB is the tool of choice for high-productivity research, development and analysis.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control system neural networks, fuzzy logic, wavelets, simulation and many others.

FILE TYPES

MATLAB can read and write several types of files. There are mainly five different types of files used in MATLAB which is used for storing data or programs.

- **M-FILES-** They are the standard ASCII files, with a .m extension to the file name. There are basically two types of files and they are SCRIPT and FUNCTION file. In general, mostly MATLAB files are saved as M-FILES.
- **MAT-FILES-** They are the binary data-files, with a .mat extension to the filename. These files are created when you save the MATLAB data with the save command. The data which you save in MATLAB can only be read by mat lab as it save in a special format.
- **FIG-FILES-** They are the binary figure-file, with a .fig extension to the filename. Such files are created by saving a figure in this format by using the save and saveas option in it. These files basically create all kind of information which is used for again recreating a figure and can be opened by filename.fig.

- **P-FILES-** These are the compiled M-File , with a .p extension to the filename. These file can be executed directly without using any compiler and parsed in it. These files are created with the P-CODE command.
- **MEX-FILES-** These are MATLAB-callable Fortran and C programme , with the .mex extension to the filename. Use of these file require some experience in MATLAB and lot of patience in it .

APPENDIX C

An Introduction to ERDAS

URL: <http://www.ERDAS.com/>

Overview

ERDAS is pleased to provide ERDAS IMAGINE® version 8.4. Many private and commercial users who need to extract and interpret information from imagery recognize ERDAS IMAGINE as a musthave. 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; reprojecting, 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.

Key Features Summary

- IMAGINE Expert Classifier
- Direct read and edit of ESRI's Shapefiles
- Direct read of ESRI's SDE data
- Enhanced and expanded native raster file handling
- Re-projection of raster data on-the-fly
- Batch processing wizard

- Enhanced Viewer functionality
- Improved print versatility on Windows NT
- International 2-byte font support in Annotation layers
- Support for ERDAS IMAGINE .img files larger than 2 GB