CHAPTER 1

INTRODUCTION

1.1 Motivation

Since time immemorial, remote sensing data have been used for extraction of information in a number of applications ranging from civilian applications such environmental engineering, forestry, ecology, agriculture to military applications. We have indeed excelled in data acquisition capabilities as can be seen from the availability of remote sensing data at many different spectral, spatial and temporal resolutions. Nonetheless, we still lag behind in development of efficient image processing tools to process such a large variety of data to cater for the needs of different applications which also vary. For example, per-pixel classification algorithms are now fairly well established for deriving land parcel information from medium resolution remote sensing data. Similarly, in recent years, sub-pixel classification algorithms are now in the process of cementing their place in land parcel extraction from coarse spatial resolution remote sensing data.

However, most of the per-pixel and sub-pixel algorithms are based on exploiting the spectral characteristics of remote sensing data for information extraction. As the spatial resolution becomes higher or finer, more details within a land parcel are resolved in the images. As a result, both per-pixel and sub-pixel algorithms fail miserably in extracting information from high spatial resolution data, now available at 40cm resolution and at 25cm resolution in future. We, therefore, need algorithms which may treat land parcel areas as single objects. This can be possible, if the algorithms are capable of including various other attributes of an object such as texture, context, shape, size, shadow etc. along with the spectral attributes of remote sensing data in the classification process.

OBIA can be considered as an emerging discipline that can be readily applied to partition remote sensing images into meaningful image objects based on their spatial and spectral characteristics. OBIA incorporates knowledge from a vast array of disciplines, such as

Geomatics Engineering, *Image & Signal Processing*, *Machine Learning*, *Computer Vision*, *Data Base Management Systems* to name a few. In addition, advancements in remote sensing technology have also led to generation of enormous amounts of data which poses new challenges in their storage, search, scalability, sharing, analysis and visualization. These challenges need to be tackled at a fundamental level of computer science and information technology through development of efficient and intelligent methods in order to build highly reliable and scalable systems.

Recently, the OBIA has been implemented in a number of commercial software packages such as Definiens eCognition Developer, Objective etc. However, most of these software provide overly complicated options which may not be directly applied for a customised application such as cadastral mapping. Significant research is therefore required in this relatively new field in remote sensing to develop OBIA as a well-established paradigm.

1.2 Objectives of Dissertation

The main objective of this dissertation is to investigate the potential of Object Based Image Analysis (OBIA) for processing and classification of remote sensing data and subsequently developing an operational tool using the concepts of OBIA for extraction of land cover information map from multispectral images. The specific objectives of this dissertation include the development of following algorithms:

i) Multi-scale image segmentation algorithm based on region growing and merging technique for generation of initial image objects.

ii) Attribute selection method for identifying useful image object features to be used in the third and final classification stage.

iii) Supervised classification method for extracting different land cover classes from an image using the image objects.

1.3 Organization of Dissertation

Chapter 1 gives the overview of recent trends and challenges in the field of remote sensing and finally describes the problem at hand.

Chapter 2 gives details of national and international status of research in the field of geo-spatial data processing, limitations of existing methods being used and the potential of Object Based Image Analysis (OBIA) as a emerging new technique. The technical concepts of OBIA and its different stages namely image segmentation, feature selection and classification have been discussed.

Chapter 3 gives details of the proposed methodology for extraction of land cover information from high resolution remote sensing images. It also discuss the implementation details and the tool & technologies used.

Chapter 4 contains analysis of the results of the proposed methodology over different real and synthetic datasets and discusses the nature of results.

Chapter 5 concludes the work of this dissertation and gives directions for future advancement.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Current Status of Research

Due to advancements in computer technology, the OBIA has emerged as an alternative to the traditional pixel-based image analysis methods for processing high spatial resolution remote sensing data. It has been recognized that traditional pixel-based image analysis techniques are limited due to following reasons [1]:

i) Image pixels do not represent true geographical objects.
ii) Spatial attributes like texture, context and shape can not be used in pixel based image analysis.
iii) Traditional pixel-based classification methods can't work with high resolution images due to the presence of high level details present.

Remote sensing images are generally acquired in raster form i.e. matrix organized into rows and columns where each cell represents some information. Majority of data processing techniques in remote sensing are applied on per pixel basis and do not take into account contextual information. Information extraction from high resolution remote sensing images can effectively accomplished only by using neighbourhood information and the context of image objects [2].

At its most rudimentary level, OBIA has three distinct stages, as can be seen from its generic flow in Figure 2.1:

i) Image segmentation
ii) Attributes selection
iii) Object classification

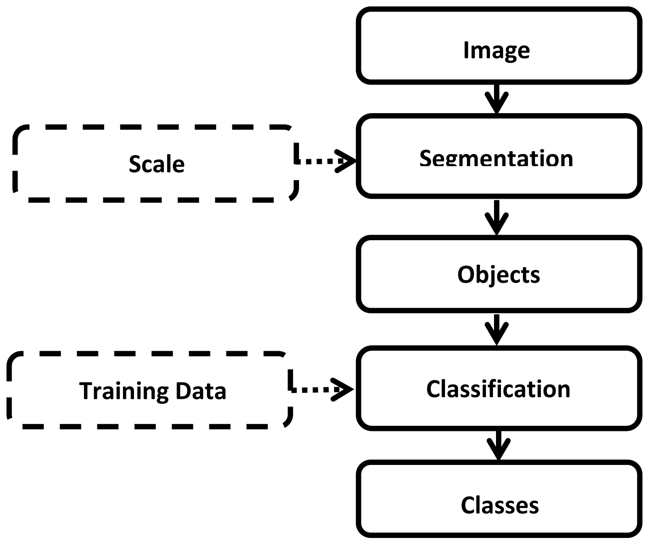


Figure 2.1 Generic flow of a typical OBIA process

Object based techniques have been used in a variety of application in the field of remote sensing like,

The advantages and strengths of OBIA, as given below, have been very well summarized by [1].

i) Image analysis technique involving breaking an image into objects is similar to the way humans conceptually organize a geographic landscape to interpret and analyse it.
ii) The computational during the classification stage is reduced due to the use of image objects as basic units in OBIA.
iii) Image objects can be characterized using a large number of attributes e.g. spectral, shape, texture, size etc.
iv) Image objects can be more easily converted to vector formats and integrated with Geographic Information Systems than pixel-wise classified raster maps.

2.2 Research Gaps

The previous sections have described the application of OBIA in different areas of geo-spatial data processing with the conclusion that OBIA is the future in deriving useful information from high spatial resolution remote sensing data. There is no argument that OBIA is superior but there are still many questions that need to be answered to consider OBIA as a fully developed technology. Some of the barriers in using OBIA may be listed as,

i) OBIA is not an operationally established paradigm.

ii) Large number of remote sensing vendors have started developing commercial software for information extraction from remote sensing image using the concepts of object based image analysis. These developments have facilitated research in the field of OBIA but have also lead to more confusion to build a general consensus on what OBIA is about.

iii) Most works on image segmentation focus just on improving the quality of segmentation results, however, an equal emphasis needs to be given on runtime efficiency keeping in the large amounts of data to be processed in remote sensing.

iv) Image segmentation has no unique solution as there is a strong lack of consensus in the field of OBIA on different concepts.

2.3 Object Based Image Analysis

Object Based Image Analysis (OBIA) may be regarded as the sub-discipline of remote sensing and geosciences used for processing and analysis of remote sensing data. The basic processing units of OBIA are image objects, which are contiguous pixel groups having real world connotation exhibiting useful attributes like shape, texture which single pixels do not contain [2] [3].

The image segmentation corresponds to division of image into segments of contiguous pixels based on the intensity values. These attributes such as shape (measured as smoothness, compactness, length, width, area, and border length etc.), texture (measured as contrast, entropy etc.), and topology (context relation and neighbourhood) together with the spectral attributes become the input data to the final stage of object classification [4].

Figure 2.2 illustrates the generic flow of an OBIA process to extract land cover information mao from a high resolution remote sensing image. Firstly, the image in Figure 2.2(a) is segmented to generate image objects, shown in Figure 2.2(b). Then the user provides training data for the classes to be extracted from the image, in this case, three classes (i) House (ii) Grassland (iii) Trees have been considered. The final classified image is shown in Figure 2.2(c).

Figure 2.2 (a) Remote Sensing Image

Figure 2.2 (b) Segmented Image

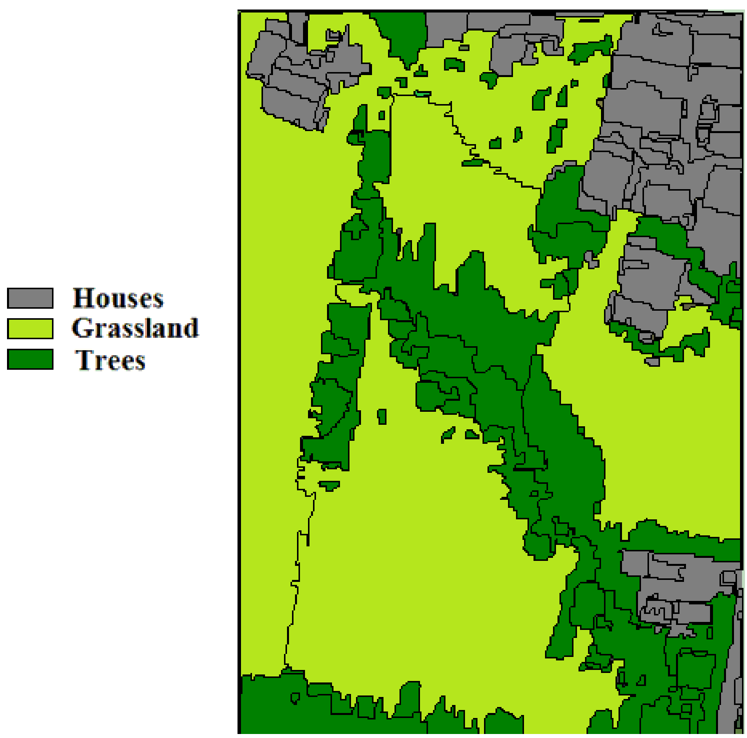


Figure 2.2 (c) Classified Image

2.4 Image Segmentation

The aim of image segmentation is to break an image into disjoint groups of pixels in order to simplify the representation of an image. Segmentation is, however, regarded as an ill poised problem as the segmentation results obtained by the users may differ from one application to the other. For instance, a face recognition system needs a segmentation algorithm which can extract faces from an image even in the presence of skin colour pixels in the background.

The image may be segmented into groups of pixels via a number of segmentation methods. These include histogram based methods, edge detection, region growing and clustering based methods. Most of the algorithms regard the problem of image segmentation as an optimization problem where the objective break the image into multiple segments based upon a homogeneity criterion for groups of pixels or image segments. The homogeneity can be based on,

1. intensity of pixels
2. statistical measures of texture such as variance and entropy etc.
3. spatial parameters like shape and size of segments.

The image segmentation has recently become major focus of research in the areas of computer vision and medical imaging. As a result, many advanced segmentation methods have been developed in recent years. Each method has its advantages and limitations. Selection of a suitable segmentation method depends on many factors, such as,

1. aim of study
2. type of image (e.g. grayscale, coloured or multi-spectral)
3. computational constraints (e.g. real time requirements in surveillance systems)
4. spatial resolution of image.

2.4.1 Overview of Existing Methods

Image segmentation methods can be broadly divided into three groups; pixel based, region based and edge based methods, as illustrated in Figure 2.3

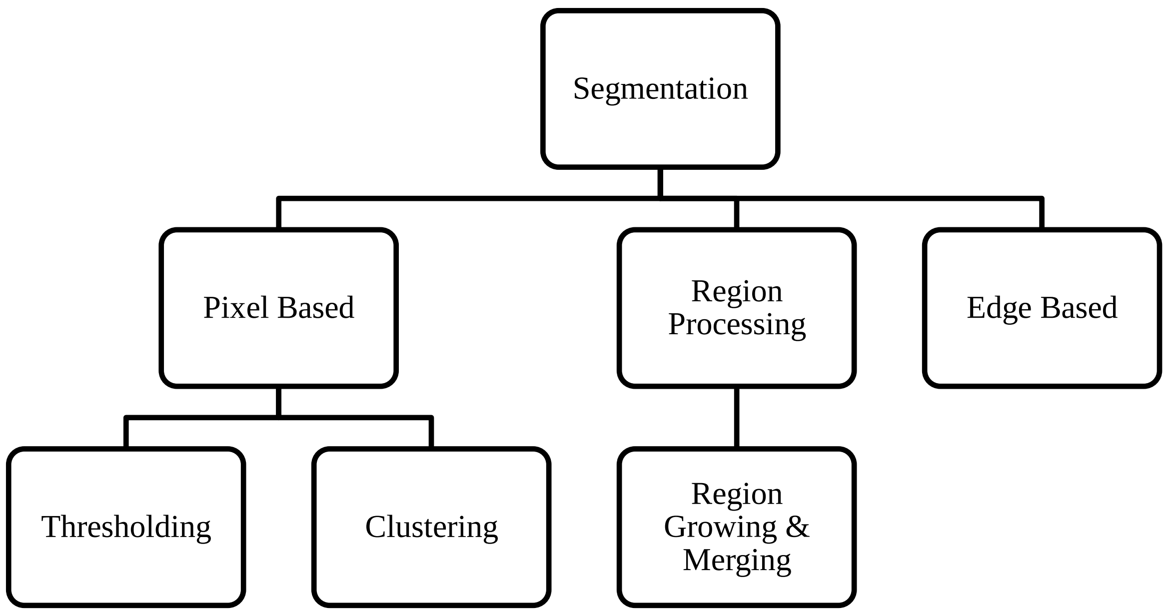


Figure 2.3 Taxonomy of image segmentation methods

i) Pixel based

Thresholding: Thresholding is used for segmentation of gray scale images. Each pixel is marked as background or foreground pixel based on its comparison with a threshold value. Key question in thresholding is the judicious selection of a threshold value, which can be determined in two ways:

1. Manual selection of a threshold value by trial or by image histogram inspection for a value which divides two groups.

2. Using thresholding algorithms like Otsu's method [5], can be used to compute a value automatically.

The advantages of thresholding methods are,

1. Easy to implement.
2. Computationally inexpensive.
3. Given excellent results for high contrast images.

However, these methods work only on single bands grayscale images and lack the sensitivity and specificity needed for accurate segmentation.

Clustering: Clustering algorithms such as the conventional K-means algorithm consider the image pixels as data points in a attributes space formed by the colour model (e.g. RGB, HSI etc.) of the image and try to find natural clusters in them. The pixels are clustered into groups having centroids $\mu_i \mid i = 1 \ldots k$. These centroid are obtained by minimizing the objective function [6].

$$V = \sum_{i=1}^{k} \sum_{xj \in Si} (x_j - \mu_j)^2 \qquad 2.1$$

where k is the number of clusters, $S_i \mid i = 1 .. k$, and μ_i is the mean of all the pixels $x_j \in S_i$.

These algorithms have the advantage in enabling the user to get fixed number of image segments using the input parameter k.

However, clustering has many limitations, such as,

1. An inappropriate choice of k (the number of clusters) may yield poor results.
2. Converges to local optimum may produce wrong results.
3. Computationally very expensive (NP hard), may not be feasible for large image.

ii) Edge Based

The aim is to detect edges between the regions and to combine these edges to form borders describing the regions. The most common algorithm of detecting edges in an image is based on some edge detection masks like Sobel, Prewitt [5] etc.

The edge based methods have the following advantages,

1. Enable the user to perform multiscale segmentation [7] by using edge information from multiple scales.
2. An approach similar to how humans segment images.
3. Works well in images with good contrast between object and background.

Whereas, the limitations of these methods are,

1. Detecting correct edges of a region may be hard due to noise and occlusions.
2. Does not work well on images with smooth transitions and low contrast.
3. Robust edge linking is not trivial.

iii) Region Growing and Merging

Region growing image segmentation starts with a seed point as input from the user corresponding to the region in the image to be extracted. The method iteratively adds neighboring pixels that are connected and similar to the seed point. The region grows until no more pixels can be added. Region merging is an extension of region growing process where multiple seed points are given as inputs which grow simultaneously by addition of neighbouring pixels and merging of adjacent regions. The algorithm works by examining neighbour regions of all regions and decides whether to merge a given pair of region or not based on a merging predicate. Merging is done if the predicate evaluates to true for a given pair of neighbouring regions. The merging predicate works by measures the similarity between the neighboring regions. Varied results can be obtained using region growing and merging depending on the choice of seed points, similarity criteria used for merging adjacent regions and the merging order.

The advantages of region based methods are,

1. Generally works better in a noisy image in comparison to other segmentation approaches.
2. Various image attributes like colour, intensity, texture can be used which might not be possible in edge based and pixel based techniques.

However, these also have some limitations such as,

1. May lead to over segmentation in textured image.
2. Selection of seed points is critical results may vary drastically for different sets of seed points.

The selection of seed points may be done in the following ways [5] [8]:

1. Seed points are given as input by the user corresponding to the region of interest.
2. All pixels are used as seed points i.e. each pixel in the image start as separate segments and the process in continued until there is pair of adjacent region fulfilling the merging criteria.
3. Seed points are randomly selected and region growing is continued until all the pixels in the image are belong to a region.

The similarity criteria for merging neighbouring regions can be based on [2] [8]:

1. Simply using the difference in mean values of the regions. Merge regions if the difference in mean values is less than a certain threshold i.e. Merge region R1 and R2 if |R1| - |R2| <= threshold, where |R1| = mean values of region R1.

2. Define a set of attributes $\mathbf{x} = (a_1, a_2 \ldots a_n)$ using spectral and spatial parameters to describe a region like mean value, texture, shape etc. Merge the regions if they are located within a certain distance in the feature space i.e. Merge region R1 and R2 with attribute set (vector) **x1** and **x2** if,

$$|\mathbf{x}_1 - \mathbf{x}_2| = \sqrt{\sum_{i=0}^{n}(a1_i - a2_i)^2} < \text{threshold} \qquad 2.2$$

The merging order can be defined in the following ways [2] [8] [9] :

1. Randomly select a pair of adjacent region and test for merging.
2. Select the pair from the whole scene which satisfies that merging criteria best.
3. Use a pre-define order for testing the merging of regions.

2.5 Attribute Selection

The segmentation step leads to division of image to segments with a set of attributes attached to each segment. These attributes form the basis of classification in an OBIA paradigm while treating each segment as an object. Thus, in comparison to pixels, objects carry much more useful information in the form of attributes and not just the spectral or spectral derived information at pixel level. Table 2.1 lists some of the typical object attributes used for classification. These attributes to be used for classification may be categorized into four groups,

1. Spectral
2. Shape
3. Texture
4. Topological

Table 2.1 Typical attributes used for object based image classification [8]

Type	Object Attribute	Description	Calculated For Each Spectral Band
Spectral	Mean	Average value of all pixels	Yes
	Brightness	Average value of all spectral bands	No
	Minimum	Minimum pixel value	Yes
	Maximum	Maximum pixel value	Yes
	Ratio	Mean value/Brightness	Yes
Shape	Smoothness	Object /bounding box perimeter	No
	Compactness	Object area/bounding box area	No
	Area	Number of pixels	No
	Length/Width	Length/Width of bounding box	No
	Width	Width of bounding box	No
	Length	Length of bounding box	No
	Border Length	Border length of bounding box	No
Texture	Standard Deviation	Contrast	Yes
	Entropy	Randomness	Yes
	Uniformity	Statistical measure	Yes
	Third Moment	Statistical measure	Yes

The total number of attributes may typically be very large. No doubt, a large set of attributes enable proper distinction of different classes to be extracted, yet the use of unnecessary attributes in classification may decrease the efficiency and final accuracy [10], and the classification process may stuck up in what is known as Huges phenomenon. Another problem associated in using a large number of attributes is over-fitting where the classifier may not be able to estimate all the required parameters based on a small finite training data [11] [12].

Thus, the selection of relevant set of object attributes for classification is crucial step in OBIA. The selection process can performed manually by a human analyst based on the objective of study, familiarity with the geographical scene being analysed. However, manual attributes selection may be tedious and error prone. Therefore, many automatic attributes selection approaches have been developed. The attributes selection approaches in the field of remote sensing include Bhattacharyya distance [13], Jeffreys-Matusita distance [14], Classification Tree Analysis [15] and Feature Space Optimization [16]. In order to select a particular attributes selection approach, many factors like final classification accuracy, computational complexity, efficient workflow and reduction in attributes space etc., may have to be considered. Recently, a comparative study by [12] has shown the strengths and weaknesses off different attributes selection methods and has concluded that Jeffreys-Matusita distance and Classification Tree Analysis produce highest overall accuracies for mapping of vegetation from high resolution remote sensing images.

2.6 Object Classification

2.6.1 Overview

Classification of image objects is the final step in OBIA to retrieve information. The objective of classification is to allocate each object to one of the classes of interest. The classification steps uses the attribute data, as described in previous section, to classify each image object into different classes based on a decision rule. Using image objects as basic units instead of pixels as the units significantly reduces the computational load during the classification stage. However, the selection of a suitable decision rule or a classification method is indeed a difficult task. A number of factors such as the spatial resolution, source and objective of the study etc. need to be taken into account for selecting an appropriate classification method.

Varied results may be obtained by using different classifiers. Several different classification algorithms (classifiers) have been developed, each having their own merits and demerits in terms of efficiency and accuracy. Maximum Likelihood Classifier (MLC), Minimum Distance Classifier (MDC), K-means Classifier, Support Vector Machines (SVM), Artificial Neural Networks (ANN), Fuzzy Classifiers are some of the methods used for object based classification. Although MLC is the most commonly used supervised classifier for pixel based classification it has not been used extensively for object based classification. The performance of parametric classifiers depends on how well the data match the pre-defined model and on the accurate estimation of model parameters. Also parametric classifiers require sufficient amount of training data to estimate all the overcome the problem of Huges phenomenon.

For object based classification, non-parametric classifiers like minimum distance classifier (MDC) and artificial neural network have been used extensively [15]. Object-based classification does not suffer from the problem of the salt-and-pepper effects found in the classification results from traditional pixel based approaches, which thus results in improved accuracy of classification. MDC classifier is sensitive to the presence of irrelevant parameters in the data set [17] demonstrated that ANN produces better classification results than MDC using the same number of input attributes. ANN’s multilayer structure and its nonparametric properties enable it to given improved performance. Although neural networks perform well on remote sensing data they are very slow during the training phase.

Support Vector Machines (SVM) are a relatively new generation of classifiers, that have been proposed to overcome the limitation of existing classifiers. SVM was first introduced by [18] and discussed in more detail by [19] [20]. SVM is ideal for classification remote sensing data because it does not make an assumption about the statistical distribution of data. The aim while designing SVM is to maximize the margin between two classes of interest and placing a linear separating hyper plane between them. SVM have been extensively in fields like text classification, character recognition, face detection etc. Since SVM can easily classify high dimensional data they reduce the effect of Huges phenomenon.

2.6.2 Support Vector Machines

A Support Vector Machine (SVM) is designed on the principle of optimal separation of classes, it is a binary classifier. The goal is to find a linear hyper plane that separates the classes of interest. The hyper plane is also called decision boundary. The hyper plane is placed between the two classes in such a way that it satisfies the following conditions.

1. The data points belonging to a particular class are placed on the same side of the hyper plane.
2. The margin between the closest data points in both the classes is maximized [21].

In many cases a linear hyper plane is not able to separate data subjected to the above mentioned conditions. In such cases data is mapped to a higher dimensional attribute space using a non-linear transformation function. Mapping the data to a higher dimensional space spreads the data so that a linear separating hyperplane can be found. If the input data has high dimensionality, the transformation process may be computationally infeasible. This issue is resolved using *Kernel* functions. The use of kernels significantly reduces the computational load.

2.6.2.1 Linearly Separable Case

The linearly separable case is one the simplest of all the cases to design a Support Vector Machine. Consider a classification problem where the input data can be separated into two classes using linear hyperplane as shown in Figure 2.4.

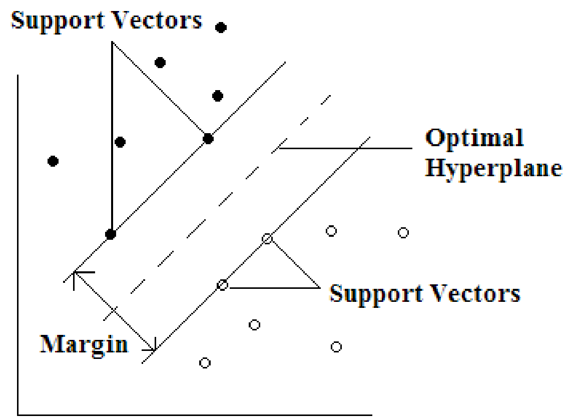


Figure 2.4 Linear separating hyperplane [22]

Consider k training samples obtained from the two classes, represented by ($\mathbf{x_1}$, y_1) ...... ($\mathbf{x_k}$, y_k), where $\mathbf{x_i}$ = (a_1, a_2 ...a_n) is a N dimensional attribute space with each training data sample belonging to one of the two classes represented y ϵ {+1, -1}. The training data is linearly separable if there exists a N dimensional vector **w** and a scalar **b** such that,

$$\mathbf{w.x_i} + b >= +1 \qquad \text{for all} \qquad y = +1 \qquad 2.3$$

$$\mathbf{w.x_i} + b <= -1 \qquad \text{for all} \qquad y = -1 \qquad 2.4$$

Points lying on the hyper plane satisfy $\mathbf{w.x_i} + b = 0$. The inequalities in 3.3 and 3.4 can be reduced to,

$$y_i\,(\mathbf{w.x_i} + b) - 1 >= 0 \qquad 2.5$$

The decision rule for can now be defined by a set of classifiers as,

$$f_{w,b} = \text{sign}\,(\mathbf{w.x} + b) \qquad 2.6$$

where sign() is the signum function. Signum returns +1 if the input value positive or zero and returns -1 otherwise.

The margin D(x; w, b), of the separation for a point from the hyperplane defined by both **w** and b is given by,

$$D(x; w, b) = \frac{|\mathbf{w} - x + b|}{||w||2} \qquad 2.7$$

where | . | is the absolute function, and || . $||_2$ is the 2-norm.

Let γ be the value of the margin between the two separating hyper planes. To maximize the margin, we express the value of γ as,

$$\gamma = \frac{|\mathbf{w} - x + b|}{||w||2} - \frac{|\mathbf{w} - x + b|}{||w||2} = \frac{2}{||w||2} \qquad 2.8$$

The maximization of equation 3.8 is equivalent to the minimization of 2-norm $\frac{||w||2}{2}$. Thus the objective function can be writen as,

$$\phi(\mathbf{w}) = \frac{1}{2}||w||2 \qquad 2.9$$

The equation 2.9 can be using quadratic programming and the solution can be expressed as,

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad 2.10$$

Only some α_i will be greater than zero and the corresponding $\mathbf{x_i}$ are the support vectors, which satisfy,

$$y_i (\mathbf{w.x_i} - b) = 1 \qquad 2.11$$

Now from equation 2.11 we can define the offset b,

$$\mathbf{w.x_i} - b = 1/y_i \rightarrow \mathbf{w.x_i} - b = y_i \rightarrow b = \mathbf{w.x_i} - y_i \qquad 2.12$$

2.6.2.2 Linearly Non-Separable Case

The SVM design discussed in the previous section is application only in the cases when the data is separable using a linear hyper plane. Most real world classification problems cannot be solved using a linear SVM. Figure 2.5 shows a case where the training data samples in two classes can't be separated by linear hyper plane. In this case, the data is mapped to a higher dimensional attribute space using a nonlinear transformation function. In the higher dimensional space, data is spread out and a linear separating hyper plane can be constructed as shown in Figure 2.6.

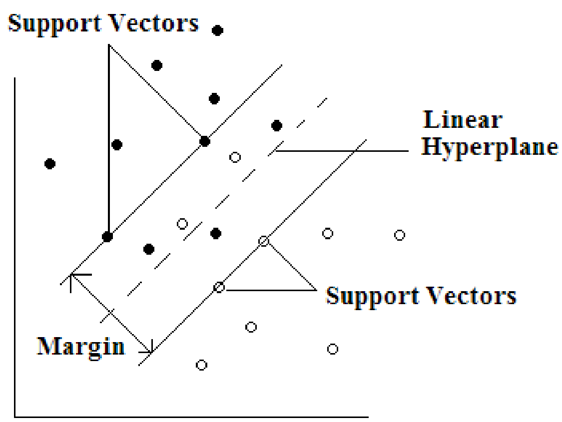


Figure 2.5 Linearly Non-Separable Case [22]

The problem with non-linear separable data is tackled using a non-linear SVM [23] [24]. It introduces slack variables $\xi_i > 0 \mid i = 1 \ldots. k$ to take into account the error in the dataset due to misclassification. Thus the constraint in eq. 2.5 is changed to,

$$y_i\,(\mathbf{w}.\mathbf{x}_i + b) - 1 + \xi_i >= 0 \qquad 2.13$$

Thus in the objective function a new term called penalty value C, $0 < C$, is added. The penalty value defines the trade-off between the number of noisy training samples and the classifier complexity. It is usually input by the user after by trial and error.

It can be shown [23] [24] that the optimization problem for the linearly non-separable becomes,

$$\text{Minimize } [\ \frac{1}{2} \|\, w \,\|2 + C \sum_{i=1}^{k} \xi_i\ ] \qquad 2.14$$

subject to the constraints,

$$y_i\,(\mathbf{w}.\mathbf{x}_i + b) - 1 + \xi_i >= 0 \qquad 2.15$$

and

$$\xi_i >= 0 \qquad \text{for } i = 1 \ldots k \qquad 2.16$$

In the non-linear case data are mapped to a higher dimensional feature space with a nonlinear transformation function. In the higher dimensional space, data are spread out and a linear separating hyper plane can be constructed as shown in Figure 2.6. For example the two classes in the Figure 2.6 are clearly not separable using a linear hyper plane.

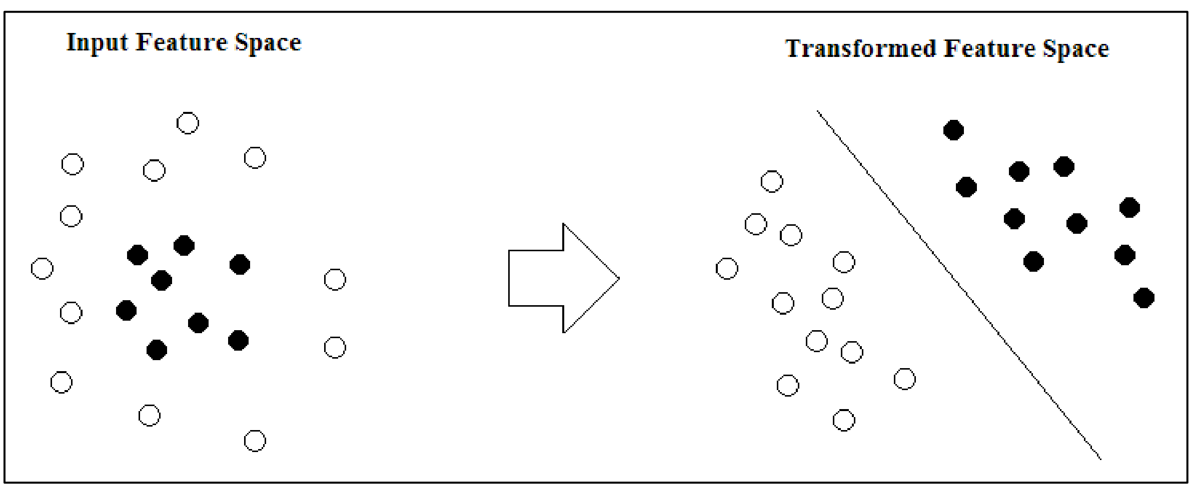


Figure 2.6 Mapping of non-linearly separable data to higher dimension [22]

The concept of creating a linear separating hyper plane in a higher dimensional feature space can be incorporated in the SVM by using a non-linear transformation function ϕ, which maps data to a higher dimensional space. In other words the data ϕ(x) represents **x** in the higher dimensional space. However while solving the optimization problem mentioned in eq 2.14 for the SVM, computing the dot product of two transformed data vectors is computationally very expensive. A kernel function is substituted for the dot product of the transformed vectors. The use of kernel function significantly reduces the computational load.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i).\,\phi(\mathbf{x}_j) \qquad 2.17$$

The selection a kernel function is essential for a particular problem. Examples of some well know kernel functions are provide in Table 2.2.

Table 2.2 Examples of well-known kernel functions [22] [23]

Kernel Name	Mathematical Definition	Description of Parameters
Linear	$\mathbf{x}.\mathbf{x}_j$	
Polynomial (with degree d)	$(\mathbf{x}.\mathbf{x}_j + 1)^d$	d is a positive integer
Radial Basis Function (RBF)	$\exp(\frac{-\|\|\mathbf{x}-\mathbf{xj}\|\|\mathbf{2}}{2\sigma})$	σ is a real value
Sigmoid	$\tanh(k(\mathbf{x}.\mathbf{x}_j) + \Theta)$	Θ and k are integer values

The development of a robust and comprehensive classification system based on the idea of OBIA is challenging task, however it promises capabilities which are not possible traditional pixel based classification systems the most important being the ability to separate classes containing objects with similar spectral responses by use of object attributes like shape and texture. Support Vector Machines are ideal for classification remote sensing data because it does not make an assumption about the statistical distribution of data. Furthermore, the use of kernel in SVM allows us to easily accommodate non linearity in the data. In the next chapter the details of the methodology used to develop an OBIA based classification system have been discussed.

CHAPTER 3
METHODOLOGY

3.1 Overview

The basic design of the proposed methodology closely follows that of a typical OBIA scheme, consisting of three distinct steps :

1. Image Segmentation
2. Attribute Selection
3. Classification

The segmentation algorithm developed belongs to the category of region growing and merging methods based on the work of [9]. To improve upon the qualitative and quantitative aspects of the algorithm, a pre-processing step has been added which dramatically improves the runtime of the image segmentation step. Attribute selection is accomplished using a scheme based on Classification and Regression Trees (CART) has been used. The classification of the image object is done using a method based on the concepts of hierarchical classification and Support Vector Machine. To assess the overall performance of the developed methodology, a classification accuracy assessment component has also been developed. The overall classification accuracy is an important measure of assessing the performance of the scheme but an equally important feature is the simplicity and efficiency of the overall workflow so that is requires least manual intervention possible.

3.2 Workflow

The workflow of the proposed methodology is shown in Figure 3.1, the various steps of the workflow are explained in detail in the next section.

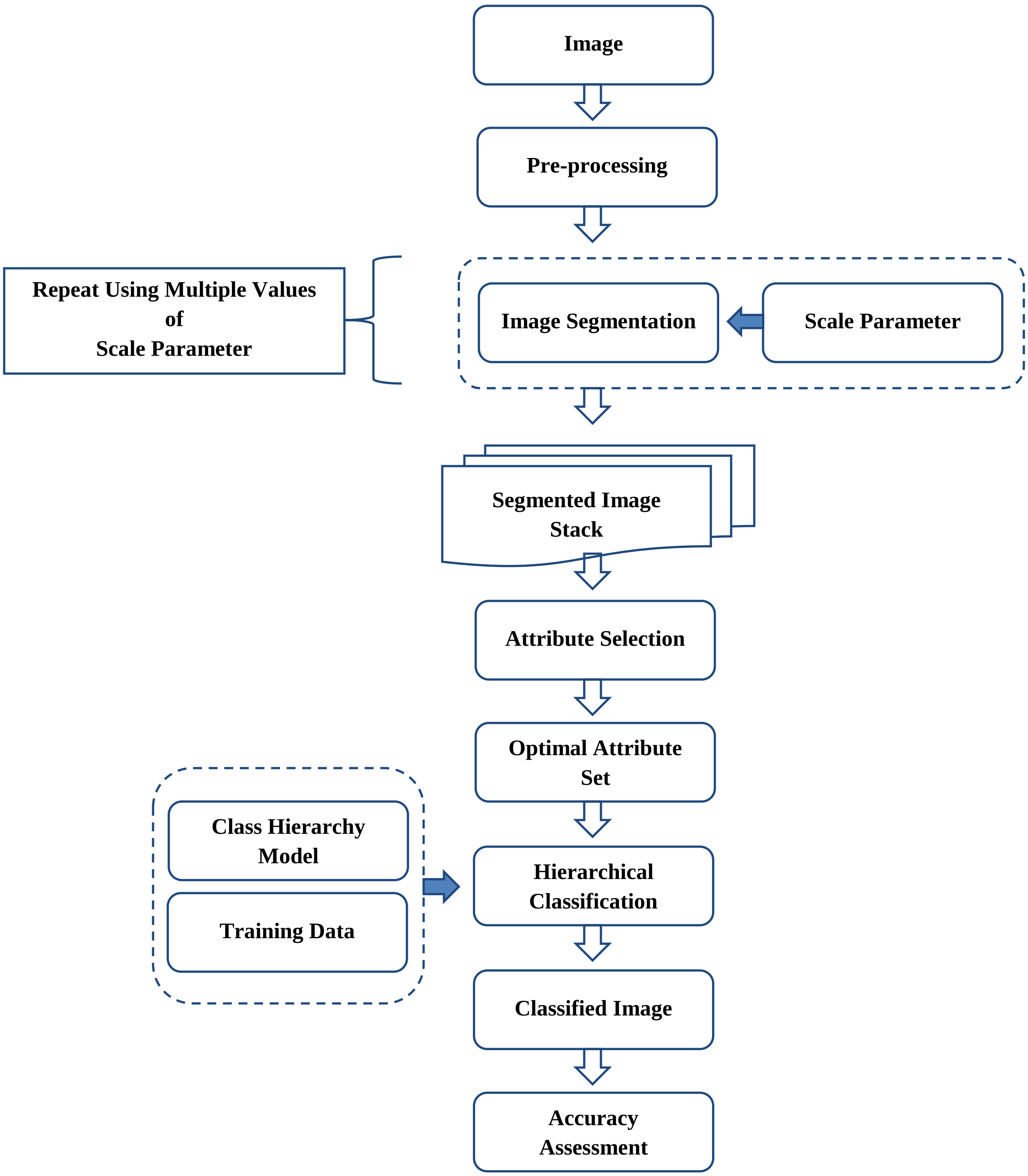


Figure 3.1 Workflow of the Proposed Methodology

3.2.1 Image Pre-processing

The first step in pre-processing of the image is to process the image using a median filter. The advantage of using a median filtering is that it not only removes salt pepper noise but also preserves the edges in an image. A median filter has a very simple working concept; each pixel value in the image is replaced by the median value of its neighbourhood. The neighbourhood of a pixels is defined using a 2-D matrix called the window which slides pixel by pixel over the entire image. A median filter with a [5 x 5] was used to filter the image.

The next step in pre-processing is segmenting the image using watershed transform [5]. Watershed transform uses the concept of ridges and valleys in real world for image segmentation. A watershed is basically a basin between highpoints in landmass where water collects when it rains. Applying the same idea to an image watershed transform works by modelling an image as a 3-D topographic surface with intensity values corresponding to surface height.

A major drawback of watershed segmentation is that it results in a hugely over segmented image. The property of the watershed segmentation to generate an over segmented image is used in the region growing and merging algorithm to enhance the quality of segments produced as well as to improve the runtime of the algorithm. The region growing and merging algorithm is modified to use initial segments generated by watershed transform are used as seed points for region growing instead of pixels. Watershed segmentation is used as a standalone segmentation approach in many applications; however in the proposed method it is essentially used as a pre-processing step.

Using the initial segments from watershed, an undirected graph is built where the segments form the nodes of the graph and edges are formed by the pair of segments which adjacent to each other in the segmented image. Results discussed in the next section show that by using such an approach up to 15 times increase in performance speed can be obtained.

3.2.2 Image Segmentation

The image segmentation algorithm belongs to the category of region growing and merging and is based on the work of [9] [25] The algorithm runs in linear time and can be easily adopted to work for multi-dimensional datasets such as multi-spectral remote sensing images. The segmentation algorithm is controlled using a user defined parameter which controls scale of the segmentation process. Greater the value of scale parameter, greater the homogeneity of segments produced (i.e. segments with smaller size). Detailed description of the algorithm is given below.

A. Selection of Seed Points

As discussed, there are different choices for selecting seed points in a region growing algorithm depending on the purpose. In this algorithm, we use the segments generated by watershed transform as the seed points for the region growing step. Using the initial segments from watershed, an undirected graph is built where the segments form the nodes of the graph and edges are formed by the pair of segments which adjacent to each other in the segmented image. After building the graph, the region growing and merging are applied on the image. Use of watershed segments as seed points dramatically improves the runtime of the region growing step.

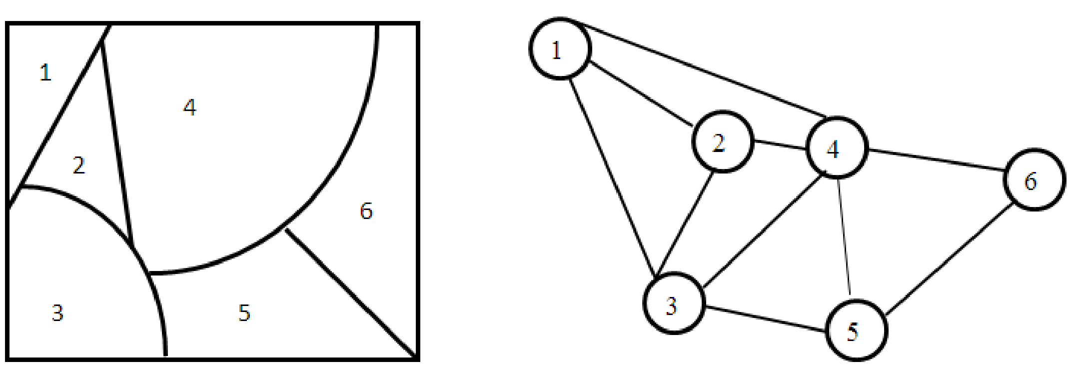


Figure 3.2 Building a graph from watershed segments

B. Merging predicate

The merging predicate used in the algorithm is derived from statistical theory by modelling the image pixels as random variables. The merging predicate which describes the condition for merging two regions R1 and R2 is as follows [1]:

$$P(R1, R2) = \begin{cases} \text{True} & \text{if } |R1 - R2| <= \sqrt{b^2(R1) + b^2(R2)} \\ \text{False} & \text{otherwise} \end{cases} \qquad 3.1$$

$$b(R1) = g\sqrt{\frac{1}{2Q|R1|}\ln(|R_{merge}|/\delta)} \qquad 3.2$$

$$b(R2) = g\sqrt{\frac{1}{2Q|R2|}\ln(|R_{merge}|/\delta)} \qquad 3.3$$

where,

	R1, R2	= neighbouring regions to be tested for merging
	R_{merge}	= region formed by merging of R1 and R2
	|R1|	= size of region R1
	|R2|	= size of region R2
	$\|R_{merge}\|$	= size of the merged region
	Q	= scale parameter (to be input by user)
	δ	= constant (= $\frac{1}{\|I\|}$, |I| = number of pixels in image I)
	g	= constant (= 256 for 8-bit images)

C. Merging order

During the execution of the algorithm, a method to decide the pair of adjacent regions to pick and test them for merging is required. There can be different approaches to decide this. In the proposed algorithm, a pre-defined order for testing the merging of adjacent regions has been used.

The merging order is defined on the undirected graph built from the intial watershed segments. The graph contains |*V*| nodes and |*E*| edges where,

V : All the vertices correspond to regions obtained from watershed segmentation

E : There is edge between vi, vj if the region represented by vi and vj are adjacent to each other in the segmented image. The weight of each edge is equal to the absolute difference in mean intensity values of the adjacent regions.

The first step in the segmentation algorithm is to sort all the edges in E in increasing order of their weights. The sorted set E gives us the merging order. During the run of the algorithm, the sorted set E is traversed only once, for any edge $(v1, v2) \in E$, we find the regions $R1$, $R2$ to which $v1$ and $v2$ belongs respectively. If $P(R1, R2)$ is True we merge $R1$ and $R2$. The pseudo code of the complete algorithm has been stated as following,

D. Image Segmentation Algorithm

Input : An image I

Output : Segmented Image

1. Compute the gradient of the image I
2. Apply watershed segmentation on the image gradient
3. Build a graph G = (V, E) from the segmented image.
4. Sort all the edges of the graph in the increasing order of their weights
5. for i = 1:size(E)
6. ($v1$, $v2$) = E(i)
7. $R1$ = Region represented by vertex $v1$
8. $R2$ = Region represented by vertex $v2$
9 if $R1$!= $R2$ and $P(R1, R2)$ = True
10. Merge region $R1$ and $R2$

E. Properties of the algorith\m

1. The input parameter Q allows the user to control the scale of segmentation i.e. homogeneity and size of region generated.
2. The number of region generated increases with increase in the value of input parameter Q.
3. The overall algorithm runs in O(nlogn) time, where n is the number of pixels in the image.
4. The only short coming of the algorithm is that it leads to over merging.
5. An O(n) time implementation is also possible by using radix sort and path compression for disjoint-set data structure.

Figure 3.3 shows the step by step procedure for segmentation of an image using the proposed algorithm. First the image is pre-processed using watershed segmentation on the gradient image. Then using the segments obtained from watershed, region growing algorithm is applied on the image to get the segmented image.

3.2.3 Attribute Selection

The selection of relevant set of object attributes for classification is a crucial step in OBIA. The selection process can performed manually by a human analyst based on the objective of study, his/her familiarity with the geographical scene being analyzed and past knowledge. However, manual attribute selection may be tedious and error prone. Therefore, many automatic attribute selection approaches have been developed. In order to select a particular attribute selection approach, many factors such as classification accuracy, computational complexity, efficient workflow and reduction in attribute space etc., may have to be considered. Recently, a comparative study by [12] has shown the strengths and weaknesses off different attribute selection methods and has concluded that Classification Tree based attribute selection approaches produce highest overall accuracies for mapping of vegetation from high resolution remote sensing images.

(a) (b)

(c) (d)

(e)

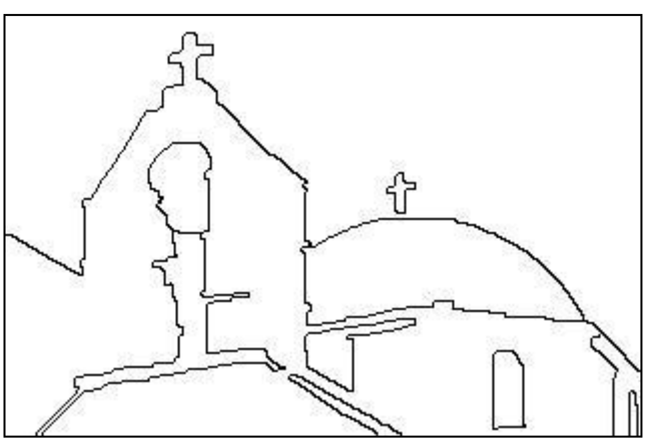


Figure 3.3 (a) Image (b) Gradient Image
(c) Segmentation using Watershed
(d) Segmentation using region growing
(e) Region boundaries

3.2.3.1 Classification & Regression Trees

CART is belongs to the category of classifiers which build a decision tree using training data and perform subsequent classification. A decision tree is basically a rule based classifier. A decision tree made up of a root node, internal nodes and leaf nodes called leaves. The internal nodes represent a decision and the leaf nodes represent final classification. The classification process is implemented by a set of rules that determine the path to be followed starting from the root node and ending at a leaf node which represents the final label for the object being classified. At each internal node a decision has to be taken, which is in the form of a simple mathematical inequality, about the path to be followed.

One of the important aspects of building a decision is the selection of appropriate criteria to define the splitting rule. The splitting rule is used at each node to divide the training into two parts with the aim of making the training data in the child nodes more pure, thus by splitting a node with remove the impurity within the training data. The homogeneity or purity of a node is formally defined using impurity function, Gini Index, Twoing Rule are some the most commonly used criteria for defining the impurity function. Thus CART recursively splits training data at each internal node into two child nodes at each step minimizing the objective function defined in eq. 3.4 until a child nodes is formed which contains data from only one class, such a child node is also called the left node of the decision tree.

Let Impurity() be the impurity function, t_{parent} be a parent node and t_{left} , t_{right} be the left and tight child nodes of the parent node t_{parent}. Consider a training dataset, represented by ($\mathbf{x_1}$, y_1) ...... ($\mathbf{x_k}$, y_k), where $\mathbf{x_i}$ = (a_1, a_2 ...a_n) is a N dimensional feature space with each sample belonging to either of the M classes labelled by $y \in \{C_1, C_2, \ldots C_M\}$. A shown in Figure 3.4, CART will try to separate the training data from the parent node to the child nodes in such a way that the sum of homogeneity of the child nodes is maximum. In order to maximize the homogeneity of child nodes the change in impurity should be maximum i.e. ΔImpurity(split) should be maximum. Now assuming P_{left}, P_{right} are the probabilities of right and left nodes respectively,

$$\Delta\text{Impurity (split)} = \text{Impurity } (t_{parent}) - P_l\text{Impurity } (t_{left}) - P_r\text{Impurity } (t_{right}) \qquad 3.4$$

Maximizing equation 3.4 implies searching for the best split decision. CART will search through all possible values of all attributes (a_1, a_2, a_3 …. a_n) for the best split decision, $a_i < K \mid K$ is a constant value, which will maximize the change of impurity measure $\Delta i(t)$. Each attribute is ranked according the change in impurity it can cause, the highest ranking attribute is then used as for splitting.

In the proposed object based classification method CART has not been performing classification but for the attribute selection step. At each step of the decision, CART picks only those attributes for performing the split which cause the maximum change in impurity. Thus the set of attributes which are picked by CART at each node of the decision tree to perform the split can be considered to be important for the final classification. Thus after the construction of the complete decision tree only those attributes are selected for final classification step which were used by CART.

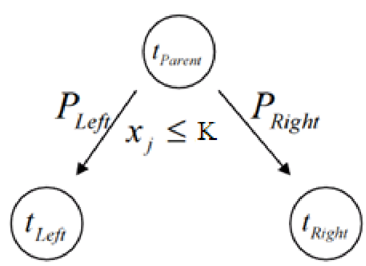


Figure 3.4 Splitting in CART [26]

3.2.4 Object Classification

This stage will involve the use of an object based classification method for classifying the image objects generated after the segmentation step into different categories representing corresponding classes. The selection of an appropriate classifier is the important factor to be kept in mind while developing a classification system. Different kinds of classifiers have varied requirements of training sample size, an inadequate training sample size often cause the problem of under fitting. The identification of classes of interest and the selection of appropriate training sample is performed manually a human analyst. Object based classifiers are considerably faster than pixels based classifiers because the use of image objects as basic units significantly reduces computational load.

A geographic scene is generally composed of multiple cascading structures i.e., a geographic scene can be broken down into multiple regions at a higher level which may contain substructure, which themselves contain substructure, etc. Digital images have a finite resolution, therefore the number of levels of structures is also finite [7]. A hierarchy therefore exists among different classes present in image. In order to effective exploit the hierarchical nature of geographic images a hierarchical classification method based on SVM has been developed. Most existing classification schemes for classification of remote sensing data use a straight forward approach, wherein an analyst first identifies different classes in the geographic scene being analyzed, following which training data is provided for different classes identified, after the training phase in over, the overall image is classified. Such straight forward approach is not capable of utilizing the hierarchical nature of geographic images. The developed hierarchical classification method is different from a straight forward classification scheme, the different phases of the method are described below,

1. **Identification of Primary Classes**

   Before performing the classification different classes of interest are identified in the image. These classes are of primary interest and appear in the final classification results. For e.g. consider the image shown in Figure in 3.5, it is a remote sensing image of Texas area acquire using *SPOT* sensor, containing roads, houses, trees, grasslands. Four classes of interest are identified in the given image namely *Street*, *House*, *Tree* and *Grassland*.

2. Creation of Class Hierarchy Model

After the identification primary classes in the geographic scene being analyzed, a class hierarchy model is built, this model reflects the relationship between different primary classes in an image. Figure 3.6 shows the hierarchical relationship between the four classes existing in the image. In this case the given classes in the given image are modelled using a two level model; the level zero consists of the image itself, at the first level image is broken into two separate classes, *Green Area* and *Manmade Structures*. At the second level the *Green Area* class breaks down into two separate classes *Tree* and *Grassland* and the *Manmade Structures* class breaks down into *House* and *Street*. The model described contains two types of classes,

i) Primary classes
ii) Auxiliary classes

The primary classes are *Tree*, *House*, *Street* and *Grassland*, these are the classes of primary interest and must appear in the final classification result, *Grassland* and *Manmade Structures* are the auxiliary classes, these classes have been created so that the relationship between Primary Classes can be described. The class hierarchy model is basically a multi-level tree which contains three types of nodes,

i) Root node represented by the given image.
ii) Auxiliary classes forming the internal node
iii) Primary classes forming the leaf nodes.

The reasons for choosing the described class model are based on intuition; the *Green Area* contains *Tree* and *Grassland* classes which have very similar spectral behaviour. Similarly the *Manmade Structures* class contains *House* and *Street* classes, both of which contain objects which are linear in shape. The modelling of relationship between different classes of interest has to perform by an analyst who is familiar with the geographic being analysed and based on his past experience.

3. Building a Hierarchical Network of Segmented Images

After the creation of class model, a hierarchical network of segmented images is created, this network is basically a stack of segmented image formed by segmentation of the remote sensing image at different values of scale parameter. The image segmentation using the algorithm has been described in Section 4.2.1 and 4.2.2. The number of images in this stack is equal to the number of levels in the class hierarchy model, thus each class in the class hierarchy model (both primary and auxiliary) is associated with a segmented image at the corresponding level in the segmented image stack. At the top most level the image is segmented at a coarse scale, as we move down the stack the segmentation scale varies from coarse to scale. The actual values of scale parameter used for segmenting the images at every level are chosen by the analyst based on prior knowledge and hit and trial. An image object at a particular level in the segmented image stack is connected to its neighbors in the same level as well as in the immediate lower and upper level also, thus forming hierarchical network of image objects. By building such a hierarchical network of image objects, the complexities in a geographic scene may be effectively modelled.

4. Training Phase

During the training phase, training data is provided by an analyst for the primary classes. For model described in Figure 3.5 training data is provided for the four primary classes *Tree*, *Street*, *House* and *Grassland*. The training data would basically be image objects for the corresponding primary classes from the lowermost level of the segmented image stack. The training data for the auxiliary classes is derived from training data given for the primary classes, for e.g. the training data for the auxiliary class *Green Area* would consist of image objects from the corresponding level in the segmented image stack which are connected to training image objects of *Tree* and *Grassland* in the lower level. Thus the training is performed in a *bottom-up* manner. The algorithm for training phase is described below:

5. **Hierarchical Classification**

Unlike the training phase the classification is done in a *top-down* manner. Starting from the root node and moving level by level classification is performed for each auxiliary class node. The order followed for classification at different nodes at a particular level is immaterial. At each node the classification process is performed on the image objects from the corresponding segmented image of the segmented image stack. Continuing with the example described above, first classification would be performed for the root node, and this would split all the image objects into two classes *Green Area* and *Manmade Structures*. The classification at the next level would be done for node corresponding to *Green Area* and *Manmade Structure*. The image objects marked as *Green Area* would be classified into *Tree* and *Grassland* similarly image objects marked as *Manmade Structure* would be split into *Tree* and *Grassland.* In such a way classification is performed recursively for all nodes until a leaf node is reached.

SVM has been chosen as the classifier for the proposed classification scheme. SVM is ideal for classification remote sensing data because it does not make an assumption about the statistical distribution of data. Furthermore, the use of kernel in SVM allows us to easily accommodate non linearity in the data. Since SVM is a non-parametric classifier, it eases the job of the analyst in the time intensive step of tuning the values of parameters thereby leading to an efficient workflow.

Figure 3.5 Remote sensing image of Texas area

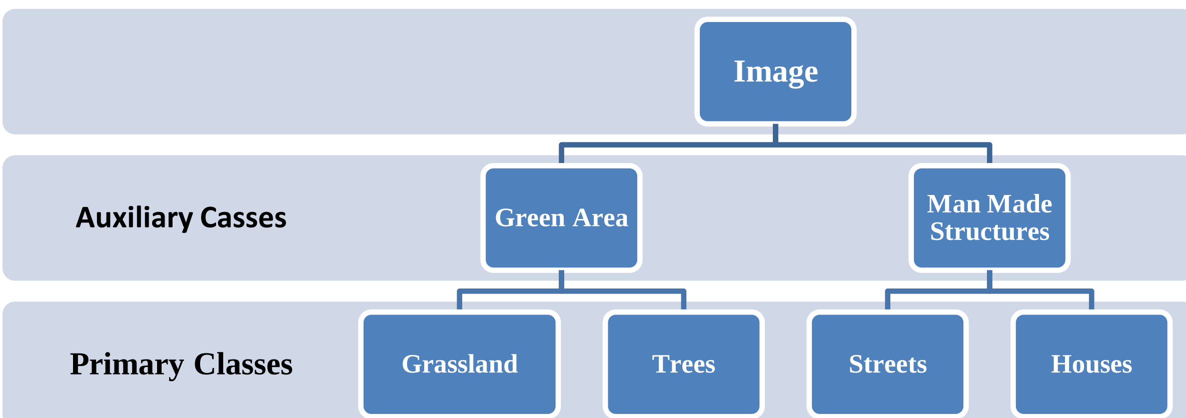
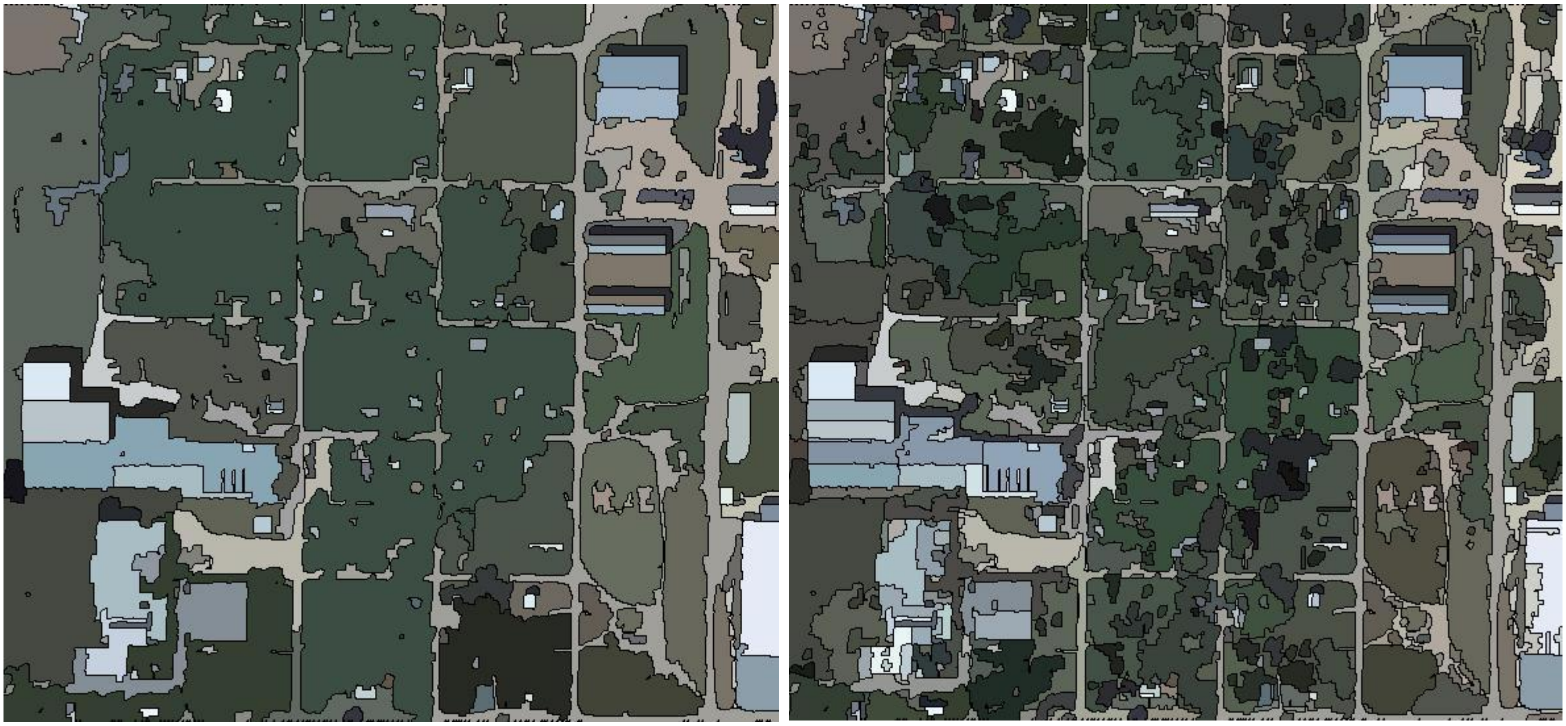
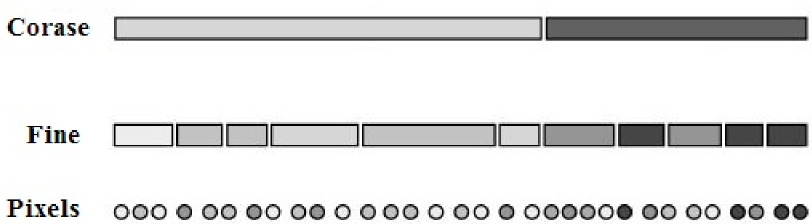


Figure 3.6 Hierarchical relationships between different classes



(a)



(b)

Figure 3.7(a) Segmented Image Stack (b) Schematic view of hierarchical network of Image objects [8]

3.2.5 Accuracy Assessment

Assessment of final classification accuracy is an integral part of classification system. In the developed methodology an error/confusion matrix based approach has been used for accuracy assessment. An error/confusion matrix is simply a square matrix of integers, the row of the matrix represent classified image pixels and the columns represent reference image pixels. A cell (i,j) of the error matrix contains an integer k means that, k pixels of class-j have been classified as pixels of class-i. Therefore the number of correctly classified pixels is given by the sum of the diagonal elements of the error matrix.

Table 4.1 shows the example of a simple error/confusion matrix. The *overall accuracy* for the classification is calculated by dividing the total number of correctly classified pixels by the size of the image i.e. the total number of pixels in the image. The classification accuracy for c individual classes is calculated in two ways :

1. <u>Producer's Accuracy</u> – It is computed by the dividing the number of correctly classified pixels in a particular category by the actual number of pixels belonging to this category. Producer's accuracy is actually a measure of omission error.

2. <u>User's Accuracy</u> - It is computed by the dividing the number of correctly classified pixels in a particular category by the total number of pixels classified in the corresponding category. Producer's accuracy is actually a measure of omission error.

Accuracy assessment provides very insight in the analysis of the results produced. The overall accuracy for error matrix given in Table 4.1 is 72% and the producer's accuracy for class *Road* is 83%, however the user's accuracy for class *Road* is only 55%. This means that of the pixels classified as class *Road* only 55% belong to class *Road* rest of the others have been misclassified.

		Reference Data				
		Road	**House**	**Grassland**	**Tree**	Row Total
Classified Data	**Road**	**66**	5	23	25	119
	House	7	**82**	6	9	104
	Grassland	1	12	**86**	20	119
	Tree	5	8	4	**91**	108
	Column Total	79	107	119	145	450

Table 3.1 Example of an Error Matrix [10]

Overall Accuracy = 325 / 450 = 72%

Producer's Accuracy

Road = 66 / 79 = 83%

House = 82 / 107 = 76%

Grassland = 86 / 119 = 72%

Tree = 91 / 145 = 62%

User's Accuracy

Road = 66 / 119 = 55%

House = 82 / 104 = 78%

Grassland = 86 / 119 = 72%

Tree = 91 / 108 = 84%

3.3 Implementation of Methodology

The implementation of the proposed methodology has been done in Matlab utilizing the Image Processing Toolbox, Statistics Toolbox and Compiler Toolbox. In order to make the implementation platform independent, the Matlab Compiler Toolbox has been used to develop a standalone application, which can be executed on any system. Details of implemented functions are discussed below:

1. Watershed Transform

The implementation of watershed transform has been done using the Image Processing Toolbox. Most practical implementations typically use the image gradient to interpret the image as a topographic surface. An image gradient is a directional change in the intensity of an image. The gradient image is obtained by performing convolution of the image with Sobel filter [5]. Convolution of the image with Sobel filter gives an approximate derivate of the image in the horizontal and vertical direction. The Sobel filter uses a [3 x 3] as shown in Figure 3.8,

$$Gx = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline +1 & +2 & +1 \\ \hline \end{array} \qquad Gy = \begin{array}{|c|c|c|}\hline -1 & 0 & +1 \\ \hline -2 & 0 & +2 \\ \hline -1 & 0 & +1 \\ \hline \end{array}$$

Figure 3.8 Sobel Convolution Filter

2. Disjoint Set Data Structure

A convenient and most commonly way used to store a disjoint set is by using a tree data structure. In such tree representation, a directed tree represents a disjoint set, and each node a tree represented an element of the set. To represent multiple sets a forest of such directed trees is used. In each tree the root element acts an identifier for whole set. All the nodes of the tree contain a pointer to its parent. By using this parent pointer, now given a random element the parent pointer can be used to find the set to which this element belongs. The procedure of merging two set can simply accomplished by merging the two trees representing these sets. Figure 3.9 describes two sets {A,B} and {U,V,W,X,Y,Z} using the proposed scheme.

The data structure described above is used to implement the image segmentation algorithm. At any step in the algorithm, the image is divided into many segments i.e. all the pixels are partitioned into disjoint set. Each disjoint set of pixels represents a segment. After the pre-processing stage, we have initial segments from watershed transform which are used as seed points for the region growing step. All the watershed segments are marked with a unique number as an identifier for that segment. At the start of the algorithm, disjoint-each set are constructed corresponding to each watershed segment. Each set consists of just one element i.e. the identifier for the corresponding segment. We can use the procedure MakeSet given below to create all the initial sets:

MakeSet (x)

Input : An integer x, which acts as identifier for the corresponding watershed segment

1. parent(x) = x
2. rank(x) = 0

Now, to find region to which a given watershed segment belongs, the following procedure may be used,

Find(x)

Input : An integer x, which acts as identifier for the corresponding watershed segment

```
1. while x != parent(x)
2.        begin
3.                x = parent(x)
4.        end
5. return x
```

The merging operation can also be easily accomplished. The merge operation basically merges the two trees that represent the corresponding regions. The procedure for the merge operation is described below:

Merge(R1, R2)

Input : Integers R1 and R2, which acts as identifiers for two regions to be merged

```
if rank(R1) > rank(R2)
        parent(R1) = R2
else
        parent(R2) = R1
        if rank(R1) == rank(R2)
                rank(R1) = rank(R1) + 1
        end
end
```

The procedure of creating and merging sets has been described in Figure 3.10, each tree represents a set and subscripts denote rank.

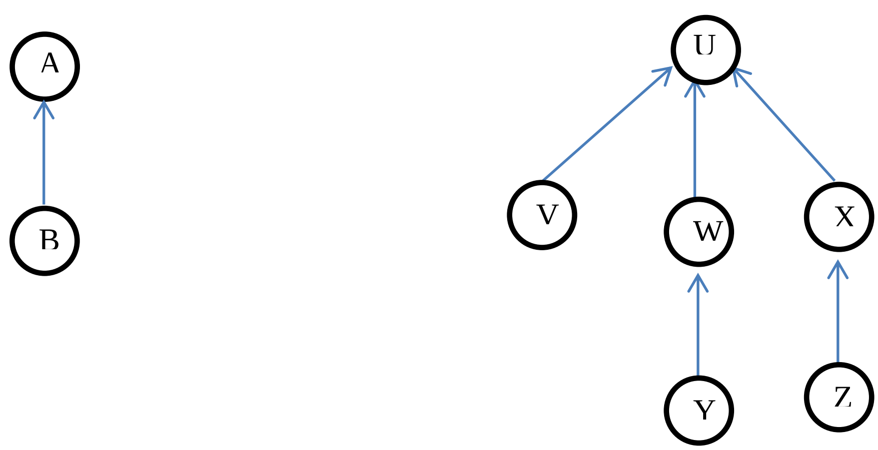
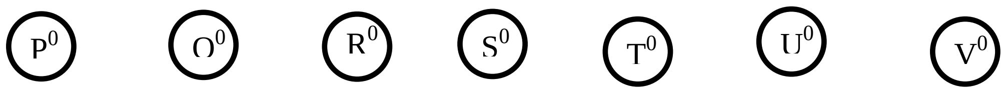
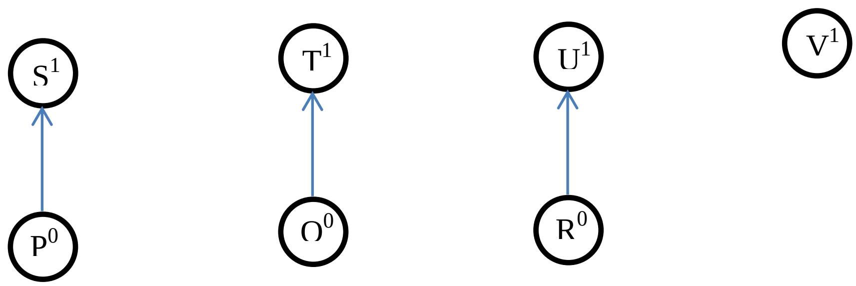


Figure 3.9 A directed tree representation of two sets

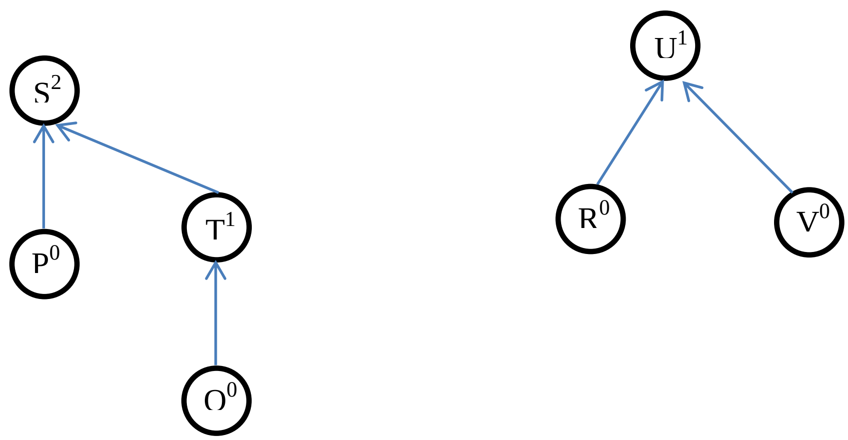
After **MakeSet**(P), **MakeSet**(Q), **MakeSet**(R), **MakeSet**(S), **MakeSet**(T), **MakeSet**(U) **MakeSet**(V) :



After **Merge**(P,S), **Merge**(Q,T), **Merge**(R,U):



After **Merge**(R,V), **Merge**(T,P):



After **Merge**(Q,V):

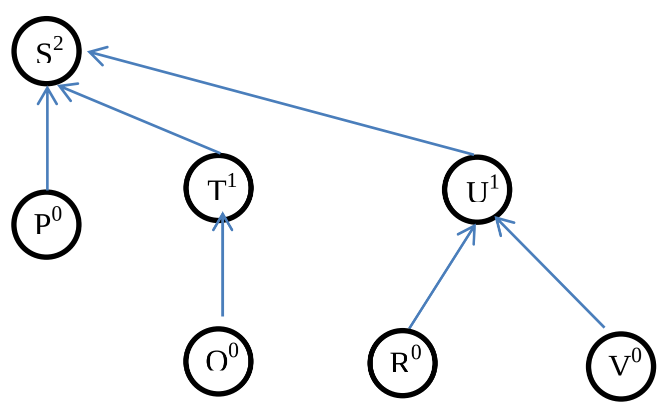


Figure 3.10 Sequences of disjoint-set operations [27]

3. CART and Support Vector Machines

The implementation of Classification & Regression Tree for attribute selection has been done using the Statistics Toolbox available in Matlab. The CART implementation in Matlab provides various functions for choosing the split criteria e.g. *Gini Index*, *Twoing Rule* and *Maximum Deviance Reduction*. In the proposed design Gini Index was used. Gini Index is the most commonly used rule for defining impurity function which in turn serves as the basis for defining the split criteria. Consider a training dataset T, represented by ($\mathbf{x_1}$, y_1) ...... ($\mathbf{x_k}$, y_k), where $\mathbf{x_i}$ = (a_1, a_2 ...a_n) is a N dimensional feature space with each sample belonging to either of the M classes labelled by $y \in \{C_1, C_2, \ldots C_M\}$. The impurity function defined by Gini Index is,

$$\text{Impurity}(T) = \text{Gini}(T) = 1 - \sum p^2_j \qquad 3.5$$

$$p_j = f(C_i, T) / |T| \qquad 3.6$$

Where, $f(C_i, T)$ = frequency of class C_i in training sample set T.

SVM classification has been implemented using an open source library *libsvm* [28] which contains implementation of Support Vector Machine as well as all the commonly used kernel functions. The *libsvm* library comes with a license which allows the user to freely use it given the condition that proper citations have been made. The library has been written in C++, in-order to use it within the Matlab environment, Matlab MEX [6] functionality was used. MEX-files, are dynamically-linked subroutines that allow a user to call C++ subroutines from within Matlab environment.

The basic design of SVM works only for classification problems which are binary in nature. Most real world problems especially those related to remote sensing data are inherently multi-class in nature. Therefore pair-wise classification approach, of the most common solutions to extend SVM's for multi-classification, was used. In the pair-wise approach classification is performed for all possible pairs of classes. If the number of classes is M, then the number of possible pairs of classes is equal to ${}^{M}C_2$ = M(M-1)/2. Now classification is performed for all of these $^{M}C_2$ pairs of classes with the corresponding training data. After classification for all possible pairs has been done the final class labels are decided based on a majority rule. Each image object will be assigned multiple classes but after the completion of pair-wise classification approach, the final classes label is the one which has been assigned maximum number of time. The advantage if this approach is that the number of training data required is small, for each class the corresponding training data need to be provided. However this method has a major disadvantage that the number of classifiers need is very large, for e.g. in a cases where 10 classes are to be used the number of classifier required will be $^{10}C_2 = 10(10-1)/2 = 45$.

3.3.1 Overview of Graphical User Interface

This section provides an overview of the GUI designed for the methodology developed. The GUI has been designed and developed in Matlab. The GUI consists of three basic modules,

1. Image Visualization Module
2. Segmentation Module
3. Classification & Attribute Selection Module

The Image Visualization module is used for loading remote sensing images and contains different components which can help a user to better understand and visualize the image. The segmentation module is used for performing Image Segmentation using the region growing and merging method described in section 3.2.2. Lastly Classification & Attribute Selection module is used for performing attribute selection using the CART based method described in section 3.2.3 and classification using Support Vector Machine. The Classification & Attribute Selection module has a limitation that it does not support the hierarchical classification method described in section 3.3.4, it can only perform a straight forward classification using SVM. In the following sections the details of each module have been described in details.

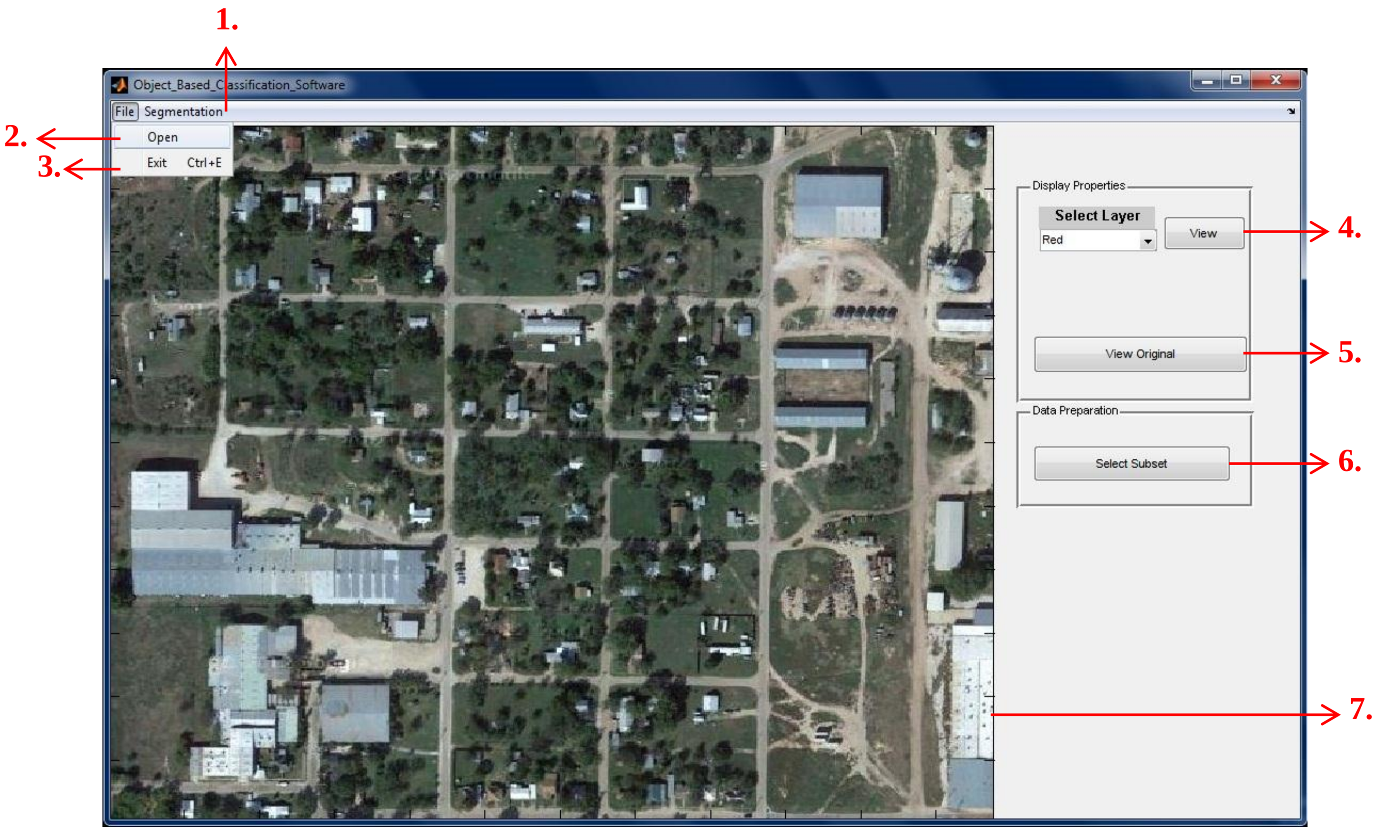
1. Image Visualization Module

The Image Visualization module forms the backbone of the complete software; it is the first window that the user encounters on opening the software. Figure 3.11 a screen shot the Image Visualization module. The module is split into three sections ,

i) Menu bar – Contains options for loading remote sensing image in the module, closing the module and launching the Image Segmentation module. At present only Jpeg, Tiff image are supported in the software.

ii) Option Pane – Contains two basic options, one for viewing different spectral layers of the image and second for selecting a subset of the image for performing subsequent operations.

iii) Image Area – This is the area on the window which displays the remote sensing image loaded in the module.



1. Menu item for opening the Image Segmentation module
2. Menu item for loading an image
3. Menu item for closing the module
4. Drop down menu for selecting the spectral layer to be viewed
5. Button for displaying the original image
6. Button for selecting subset of the displayed image
7. Area for displaying the image

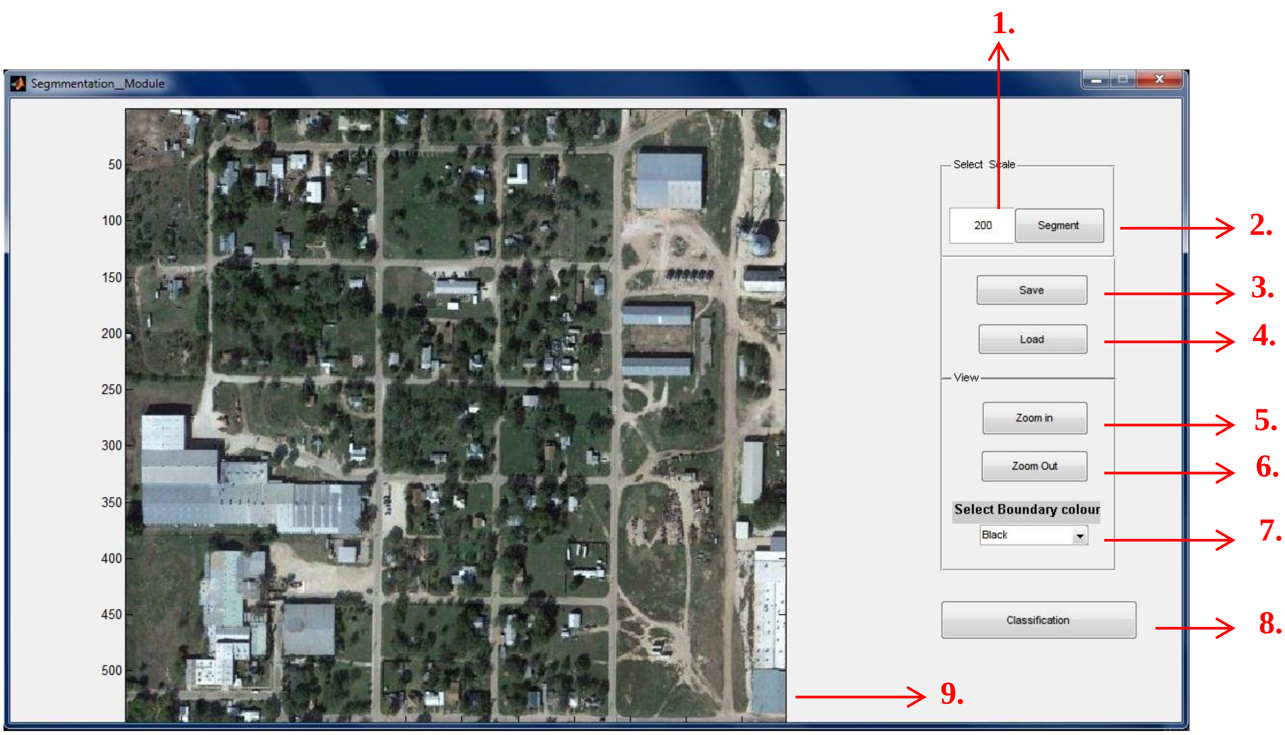
Figure 3.11 Image Visualization Module

2. Segmentation Module

The Segmentation module is launched from the Image Visualization module. It is used for performing image segmentation. Figure 3.12 a screen shot the Segmentation module. The module is split into two sections,

i) Option Pane – Contains options for entering scale parameter, save and loading results, zooming in, zooming out, drop down menu for selecting the colour of the segment boundaries and button for launching the Classification & Attribute Selection module.

ii) Image Area – This is the area on the window which displays the remote sensing image loaded in the module.

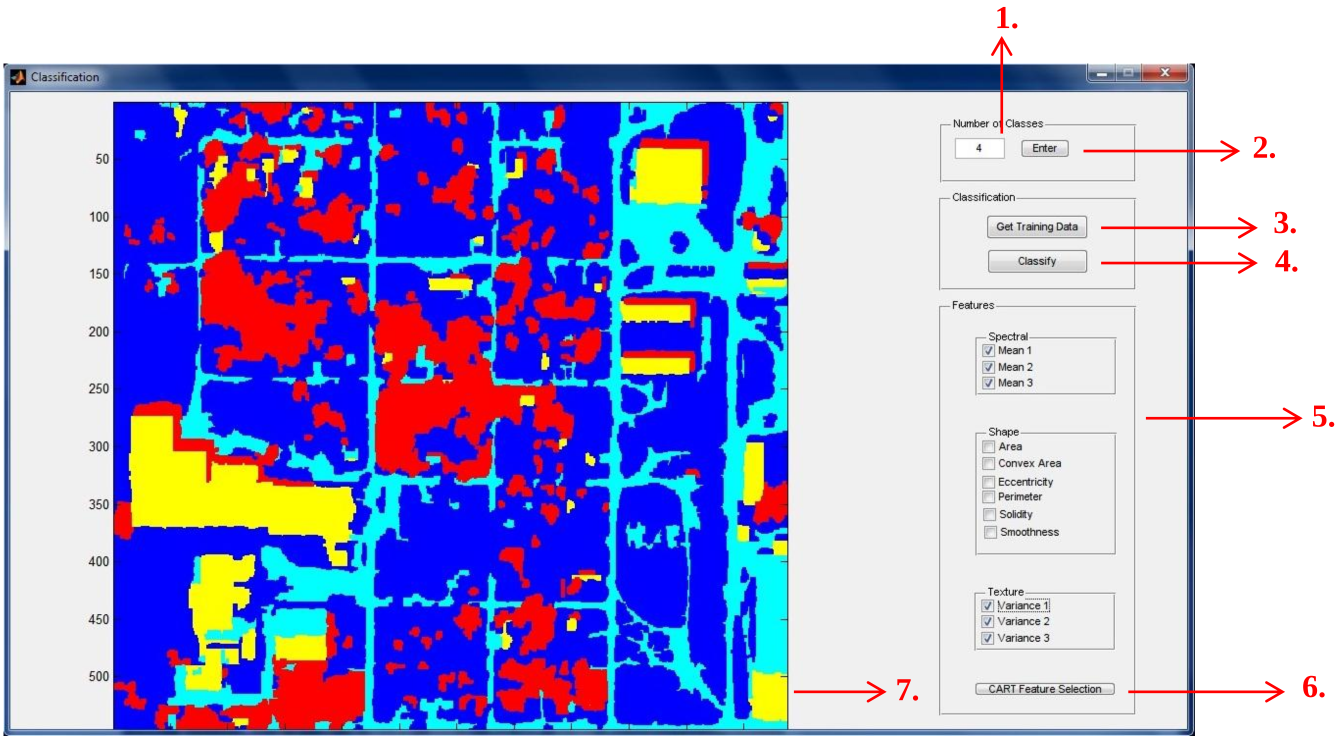


1. Text box for entering scale parameter
2. Button to start segmentation
3. Button for saving the segmented image
4. Button for loading previous results
5. Button for zooming in
6. Button for zooming out
7. Drop down menu for selecting the colour of segment boundaries
8. Button for launching Classification & Attribute Selection module

Figure 3.12 Segmentation Module

3. Classification & Attribute Selection Module

The Classification & Attribute Selection is for two purposes, first performing attribute selection using CART, second performing segmentation using SVM. Figure 3.13 shows a screenshot of the module with explanation of different options.



1. Text box for entering number of classes
2. Button to update the number of classes
3. Button to enter training data
4. Button to perform classification
5. Pane showing different attributes used for classification
6. Button to perform attribute selection
7. Area for displaying classified image

Figure 3.13 Classification & Attribute Selection Module

On the training data button, the mouse arrow changes to cross arrow using which the user can interactively select the segments which are to be given as input data for a particular class. The different attributes present the attribute pane can be selected or deselected manually by clicking on the checkbox in front of the corresponding attribute. After clicking on the *CART Feature Selection* button, a decision tree is built the selected attributes are automatically checked. Figure 3.14 shows a windows displaying decision tree produced by CART.

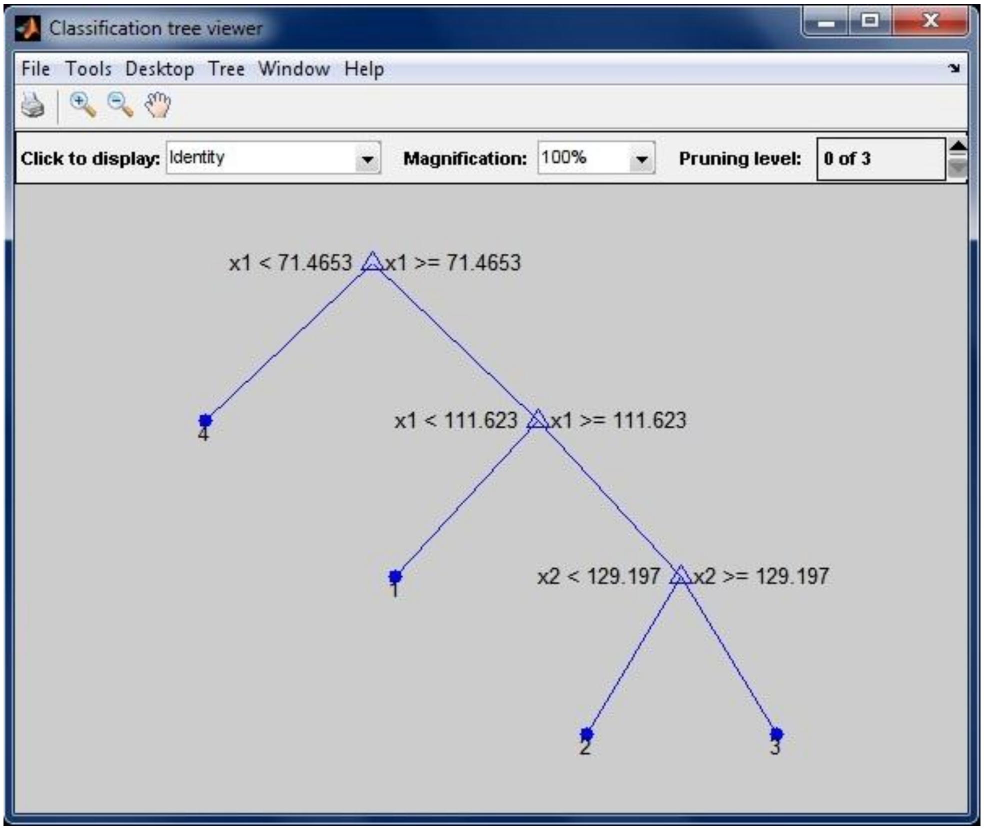


Figure 3.14 Window Showing Decision Tree Produced By CART

This chapter described the mathematical details of the developed methodology; in the next chapter the results of the developed methodology have been discussed. In order to demonstrate various aspects of developed methodology different kinds of datasets have been used.

CHAPTER 4

EXPERIMENTAL DATASETS AND RESULTS

4.1 Introduction

In this chapter, results of the developed proposed segmentation and classification methods have been discussed. Initially the results of the segmentation method on images from the Berkley Image Segmentation Dataset have been discussed. The experimental investigations of the classification system have been carried out in four phases. Firstly, the results of proposed classification system on synthetic have been discussed to demonstrate specific capabilities of the system in a simple and crisp manner, secondly classification results on remote sensing images have been discussed, and lastly comparisons with existing classification methods like Minimum Distance Classifier and Support Vector Machine have been discussed.

4.2 Berkley Image Segmentation Dataset

The Berkley Image Segmentation Data Set (BSDS) [29] contains 100 images in JPEG format together with segmentations produced by human subjects for each of the image as ground truth. BSDS has been used to qualitatively evaluate the results of the proposed image segmentation method described in Section 3.2.1 and 3.2.2. Figure 4.1 shows the results segmentation obtained on 5 images of Berkley Image Segmentation Dataset, the first column shows the image, the second column shows the ground truth (i.e. segmentation results obtained by human subjects) and the third column shows the segmentation obtained using the proposed algorithm. Next a comparison of the runtime of the proposed segmentation method over the original region merging algorithm Statistical Region Merging (SRM) by [9], has been discussed. Figure 4.2 shows the plot of runtime for the developed algorithm. The original algorithm which uses pixels as seed points has a very large runtime up to 25 seconds for images of size 1000 x 1000 pixels in Matlab. The proposed segmentation method gives a significant amount of improvement over the original algorithm. Image segmentation generally involves a lot of experimentation where the user manipulated the input parameters to obtain the desired level of segmentation. Hence we can ignore the time taken during the pre-processing stage to generate the watershed segments while analyzing the runtime of the improved region growing segmentation. Figure 4.3 shows the speed up obtained for the improved region growing segmentation relative to original region merging algorithm. For images of very large size i.e. greater than 1000 x 1000 the improved algorithm performs up to 15x faster.

4.3 Synthetic Dataset

In this section, the results of the developed classification system on synthetic images have been discussed. The aim of these experiments is to demonstrate the capabilities of the developed classification system to accurately classify image regions based on the similarity of spectral and geometric properties. Figure 4.4 shows a synthetic image, which contains varied shapes of different size, colour and texture. Figure 4.5 shows the result after segmenting the image in Figure 4.4 using the developed segmentation algorithm described in Section 3.2.1 and 3.2.2.

Classification was performed on the segmented image in two phases using the classification method based approach described in Section 3.2.4. A polynomial kernel function of degree 5 was used for classification. Firstly, four classes were created based on the shape of the segments, Figure 4.6 shows the training data for different classes and Figure 4.7 shows the result of final classification using geometric attributes. Secondly, four classes were created based on the colour i.e. spectral properties of the segments, Figure 4.8 shows the training data for different classes and Figure 4.9 shows the result of final classification using spectral attributes.

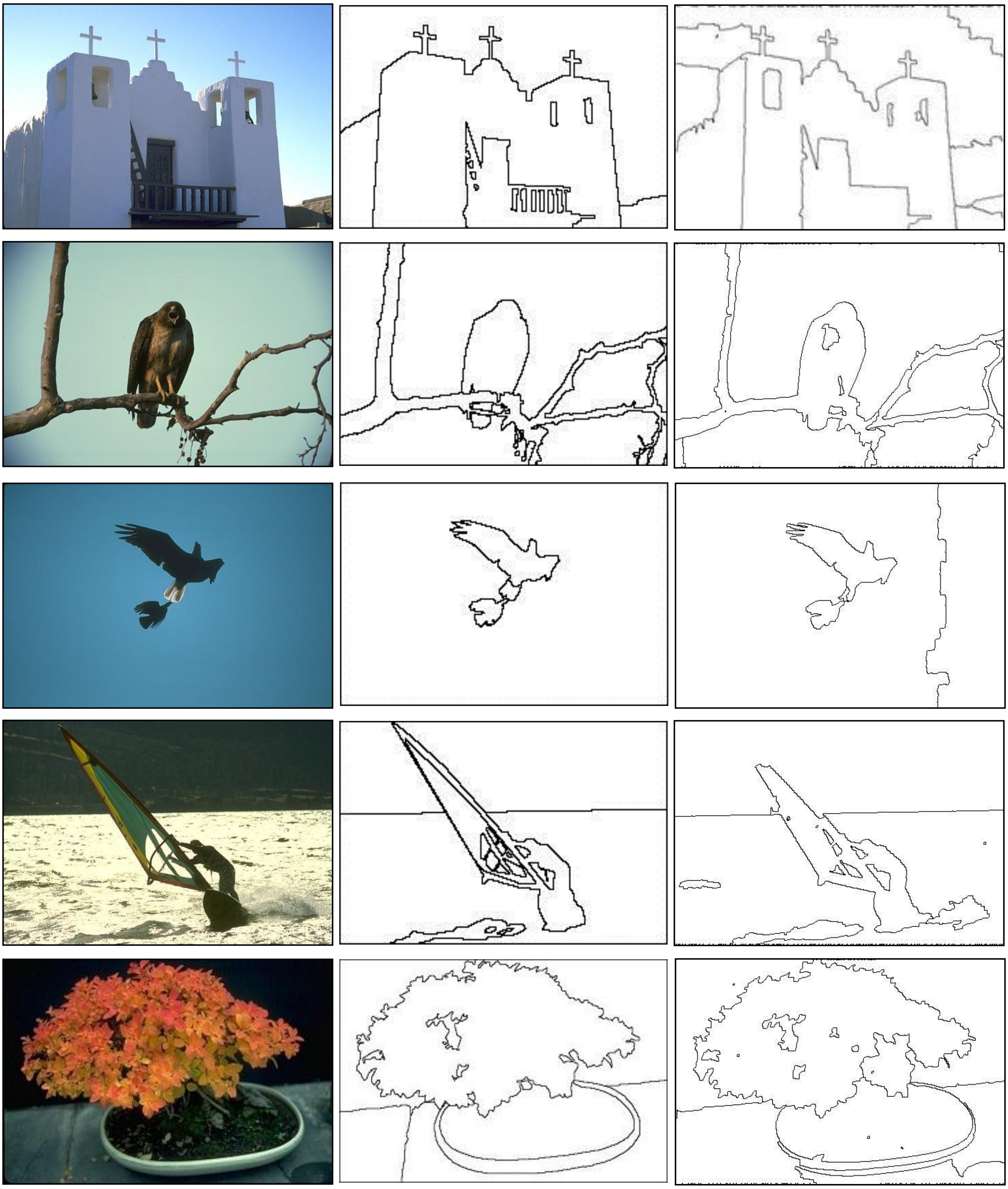


Image Ground Truth Result

Figure 4.1 Segmentation result on image from Berkley Image Segmentation Dataset

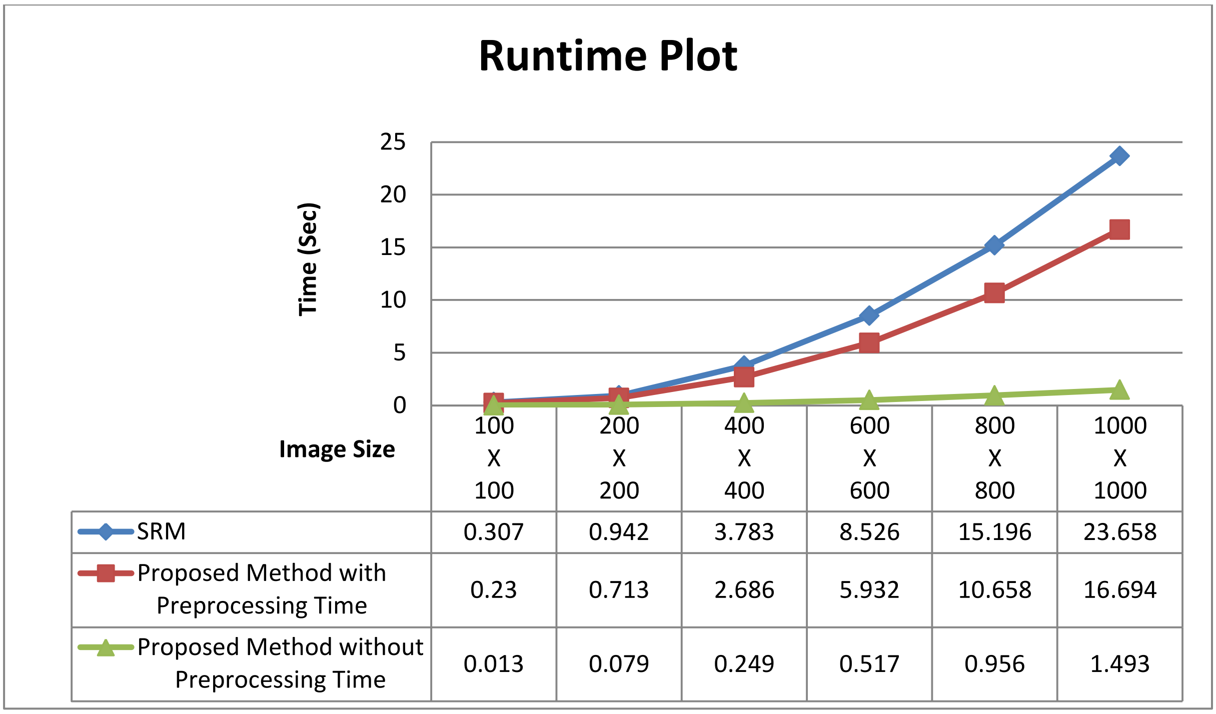
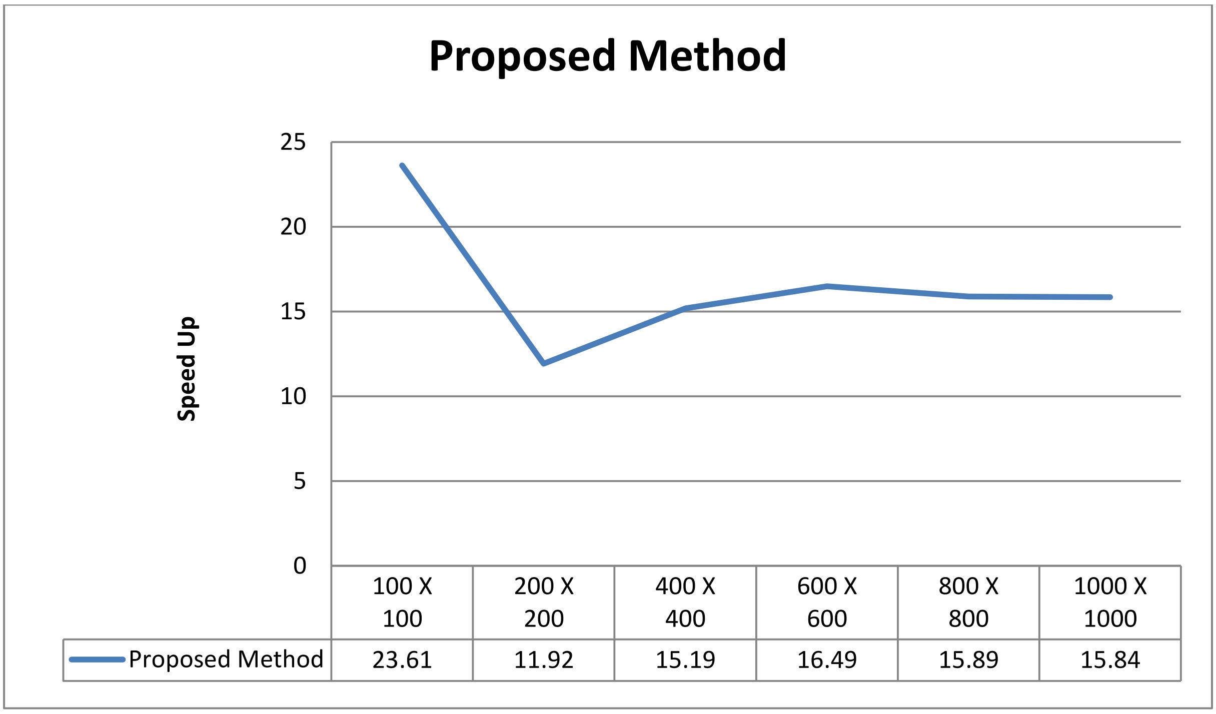


Image Size	100 X 100	200 X 200	400 X 400	600 X 600	800 X 800	1000 X 1000
SRM	0.307	0.942	3.783	8.526	15.196	23.658
Proposed Method with Preprocessing Time	0.23	0.713	2.686	5.932	10.658	16.694
Proposed Method without Preprocessing Time	0.013	0.079	0.249	0.517	0.956	1.493

Figure 4.2 Plot of runtime for the proposed image segmentation method



	100 X 100	200 X 200	400 X 400	600 X 600	800 X 800	1000 X 1000
Proposed Method	23.61	11.92	15.19	16.49	15.89	15.84

Figure 4.3 Speed up obtained for the proposed method relative to SRM [9]

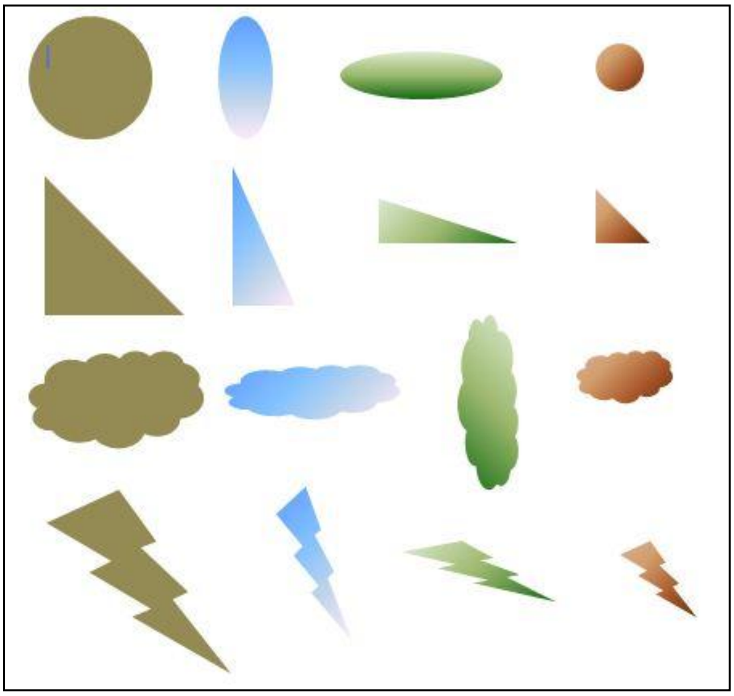


Figure 4.4 Synthetic Image

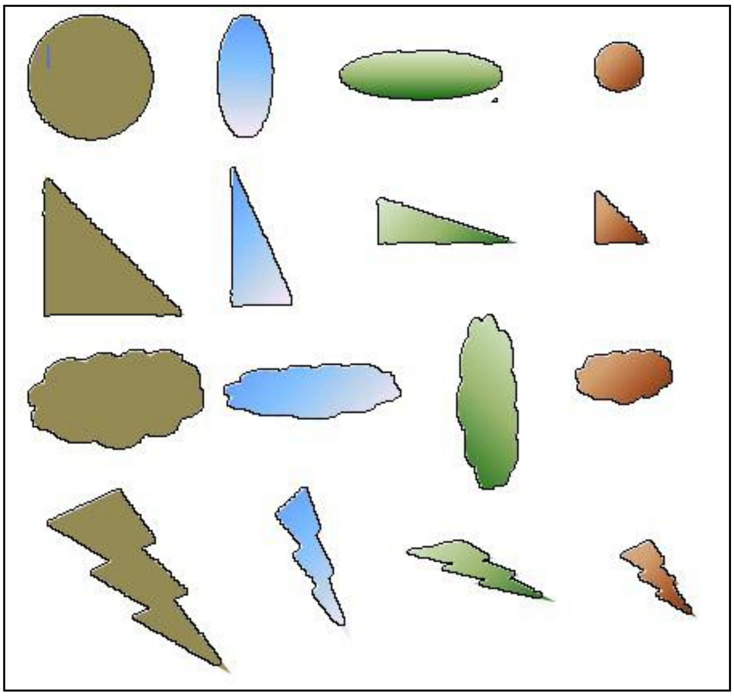


Figure 4.5 Segmented Image

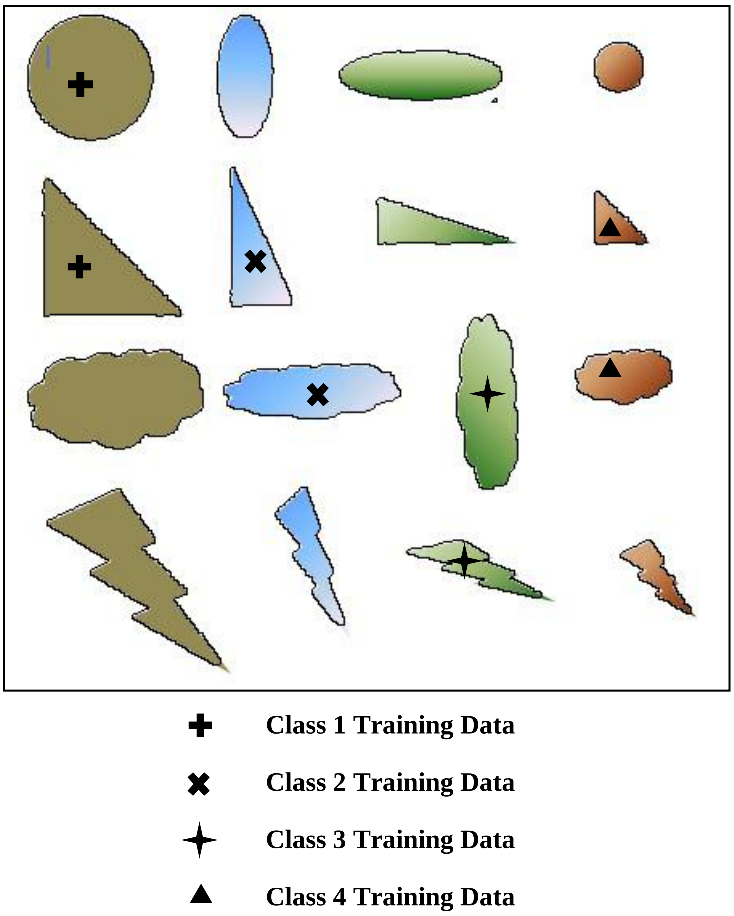


Figure 4.6 Training Data

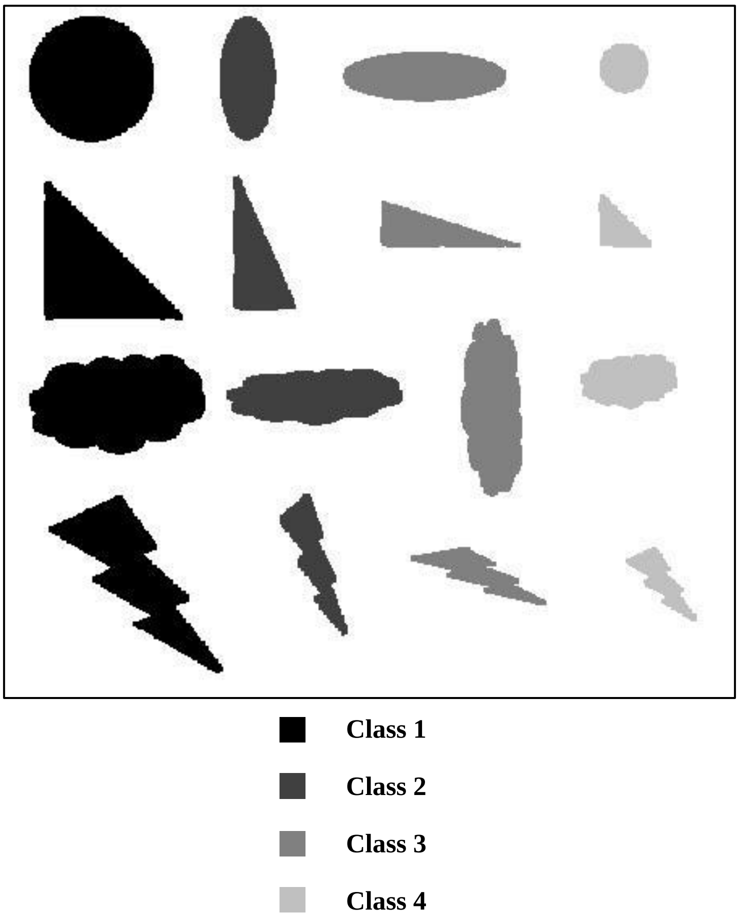


Figure 4.7 Classified Image

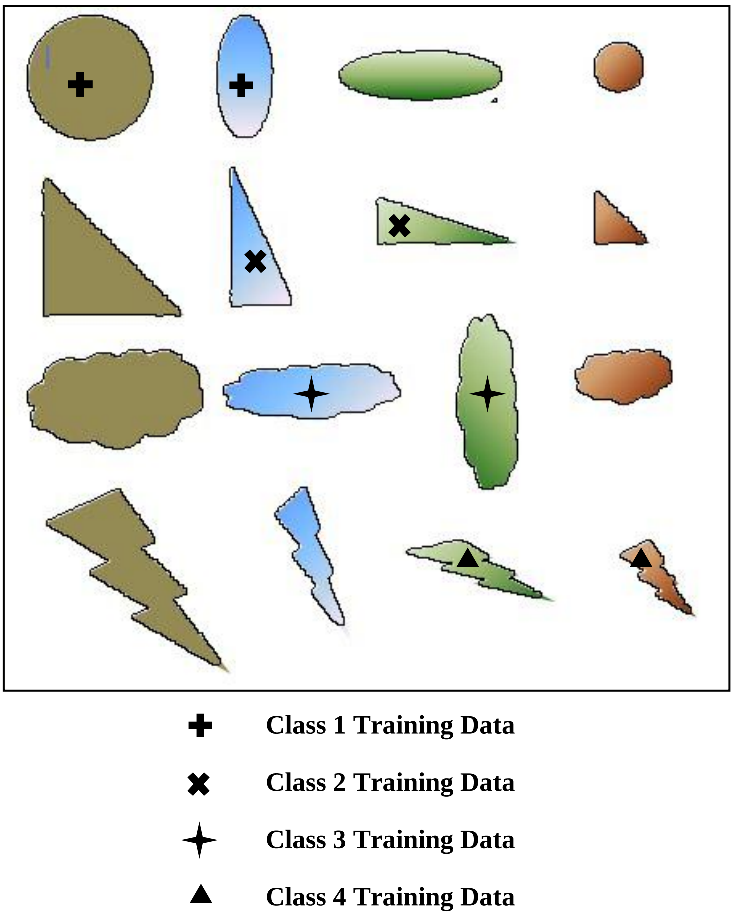


Figure 4.8 Training Data

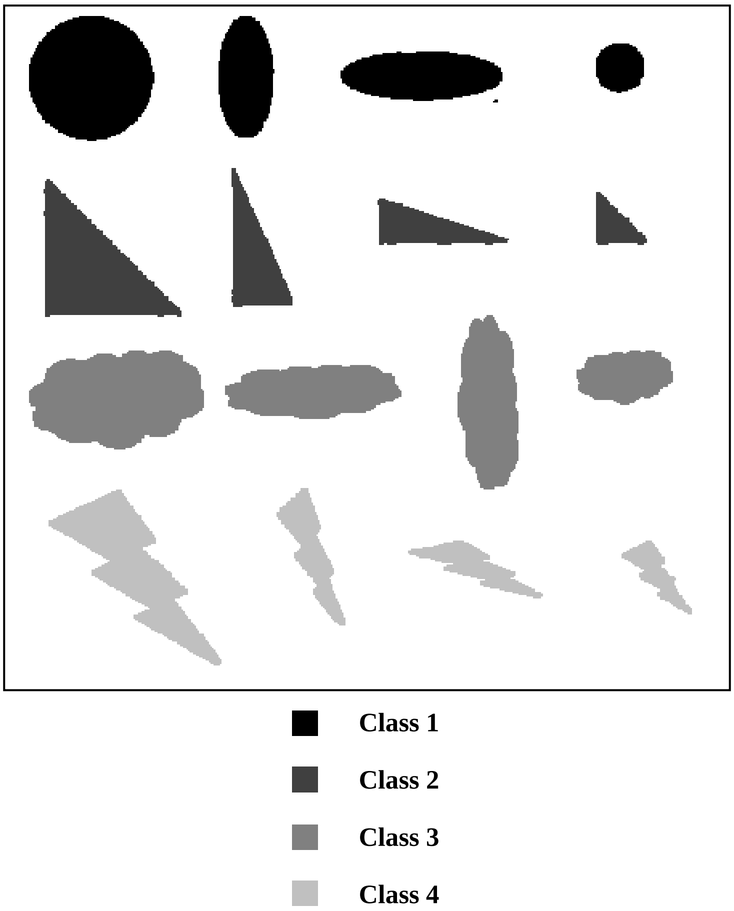


Figure 4.9 Classified Image

4.4 High Resolution Remote Sensing Dataset

In this section, the results of the developed classification system on high resolution remote sensing images have been discussed. The aim of these experiments is to demonstrate the capabilities of the proposed classification system to work with real world data. Figure 4.10 shows a 3-Band remote sensing image of Texas area acquired using *Spot* sensor. Four classes have been identified in the given image namely *House*, *Street*, *Tree*, *Grassland*. Figure 4.11 show the class hierarchy model using for classifying the image. *Tree*, *House*, *Street* and *Grassland* are the primary classes and *Grassland* and *Manmade Structures* are the auxiliary classes. In this case the given classes in the given image are modelled using a two level model; the level zero consists of the image itself, at the first level image is broken into two separate classes, *Green Area* and *Manmade Structures*. At the second level the *Green Area* class breaks down into two separate classes *Tree* and *Grassland* and the *Manmade Structures* class breaks down into *House* and *Street*. The reasons for choosing the described class model are based on intuition; the *Green Area* contains *Tree* and *Grassland* classes which have very similar spectral behaviour. Similarly the *Manmade Structures* class contains *House* and *Street* classes, both of which contain objects which are linear in shape. Figure 4.12 shows the results of segmentation on the image. Based on the class hierarchy model, two segmented images have been produced, the first segmented image was produced with a scale parameter of 50 and the second segmented image has been produced using scale parameter of 100. Figure 4.13 shows the training data used for classification, the training data was provided for the four primary classes *House, Street, Tree and Grassland.* Next the CART based attribute selection approach described in Section 3.2.3 has been used to select important attributes for classification. The decision generated by CART attribute selection is shown in Figure 4.14. CART selected only two attributes for classification, Band-1Mean and Band-2 Mean. CART as an attribute selection method has been found to sub-optimal as most of the important attributes were dropped. Therefore some of important attributes for classification have been introduced manually. Figure 4.15 shows the result of the final classification, *Street* class has been marked using blue colour, *House* using red, *Tree* using purple and *Grassland* using yellow. Table 4.1 shows the error matrix for the classification performed. The overall accuracy for classification comes out to be 83.4%. The producers accuracy for *House* and *Street* comes out to be 93.3% and 93.07% respectively, which implies that majority of the pixels which belonged to these classes have been correctly classified, however the user's accuracy for *Street* is only 80.63%, which means that although most of the Street pixels were correctly classified as *Street* there are some pixels which belong to other classes and have been incorrectly classified as *Street*. The producer's accuracy for *Tree* is 68.67%, which implies that the error rate for classification of actual *Tree* pixels is high but at the same time the user's accuracy is 90.64% which means most of pixels which have been classified as *Tree* belonged to this class. The producer's accuracy for *Grassland* is 89.5% and the user's accuracy is 77.64%. This suggests that there are many *Tree* pixels which have been incorrectly classified as *Grassland*.

Figure 4.10 Remote Sensing Image

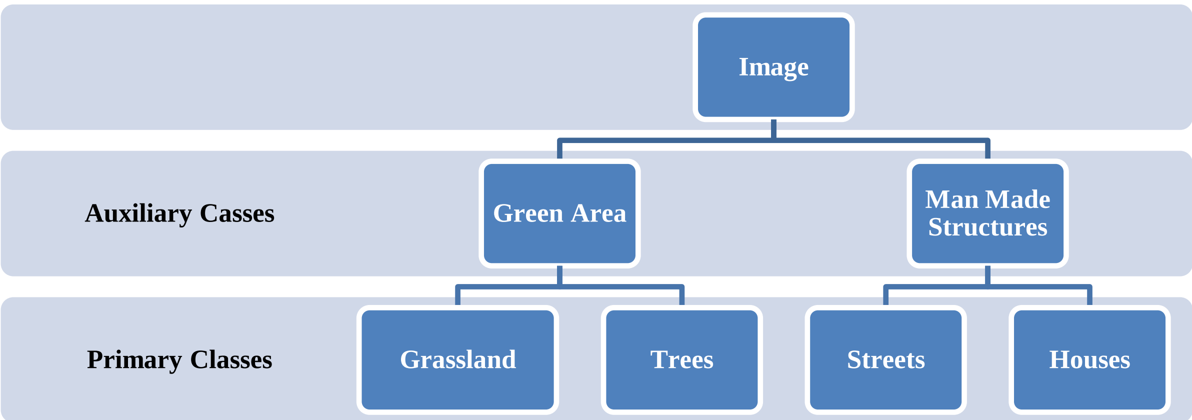
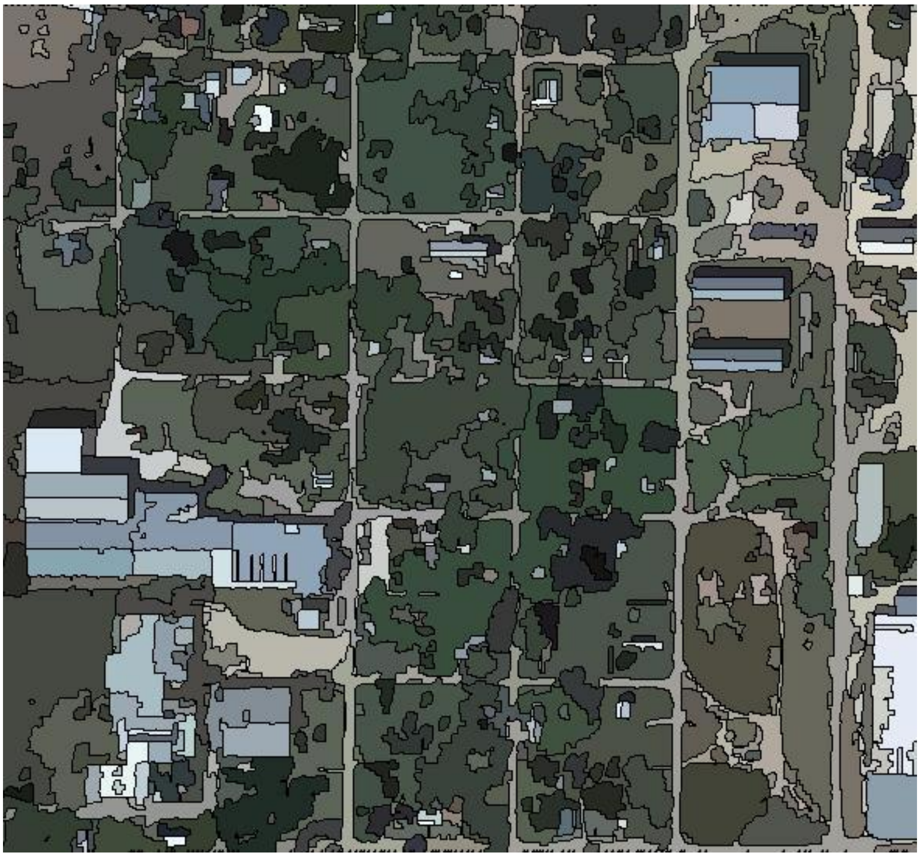


Figure 4.11 Class Hierarchy Model

(a)



(b)

Figure 4.12 Segmented Image **(a)** Coarse Scale **(b)** Fine Scale

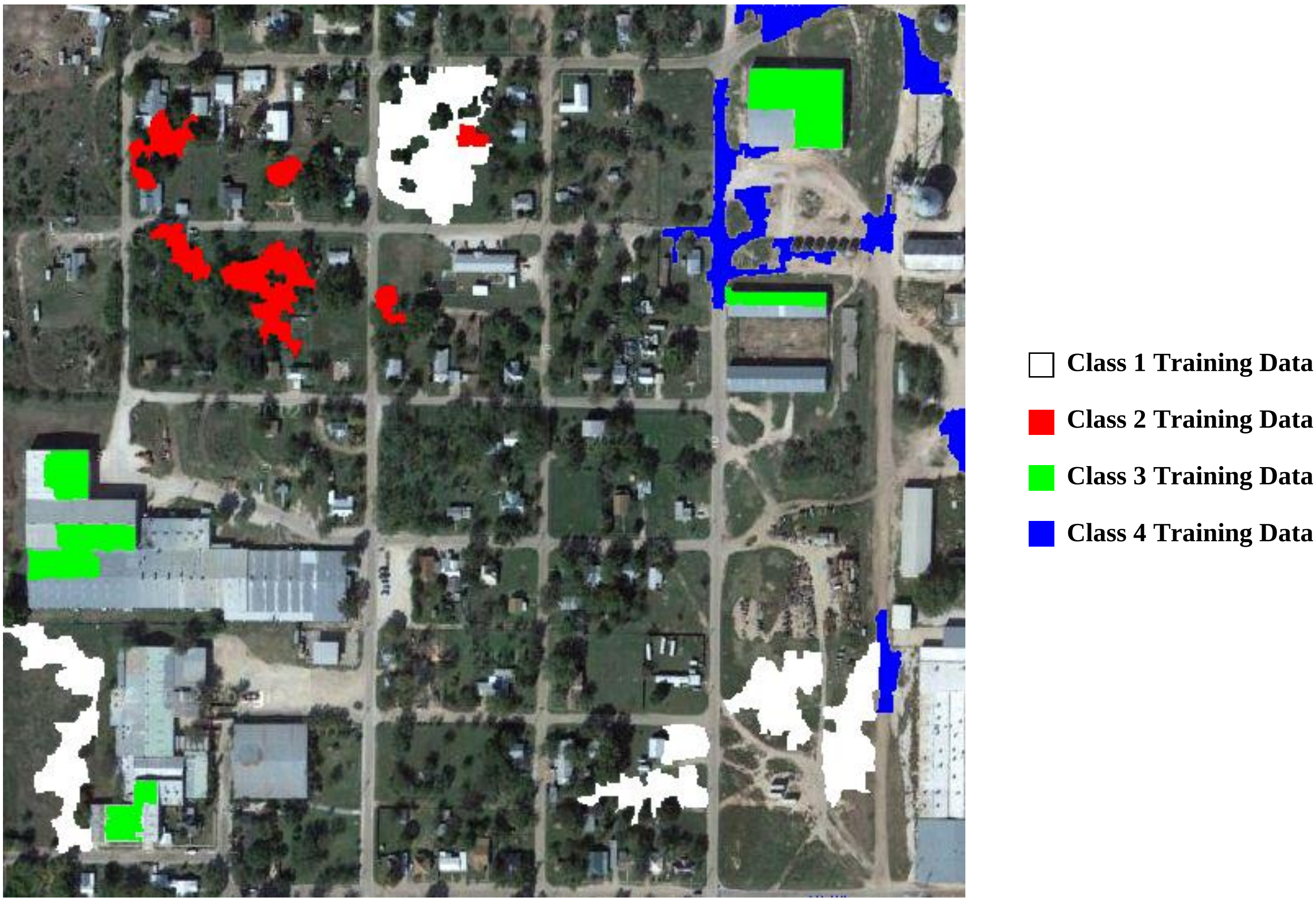


Figure 4.13 Training Data

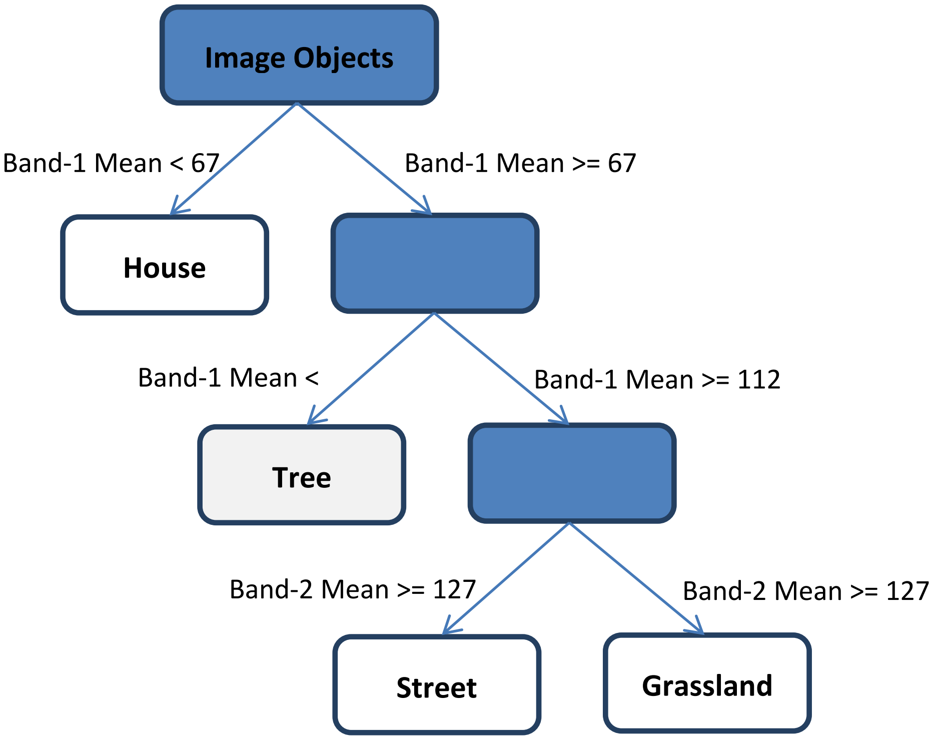


Figure 4.14 Decision tree produced by CART method

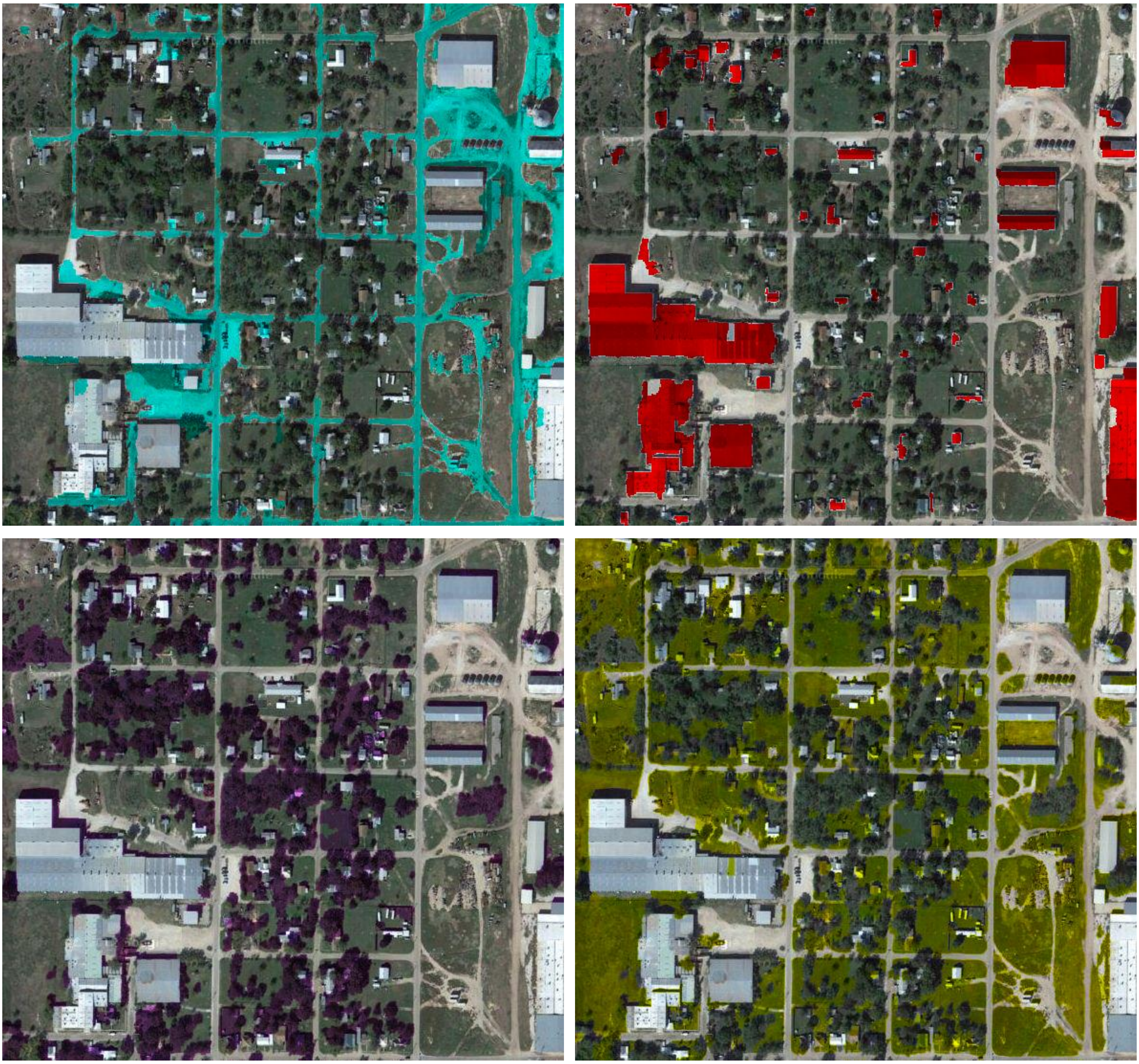


Figure 4.15 Classified Image **(a)** Streets (Blue) **(b)** Houses (Red)
(c) Trees (Purple) **(d)** Grassland (Yellow)

Reference Data

		House	Street	Tree	Grassland	Row Total
Classified Data	**House**	**31348**	477	1856	0	33681
	Street	1185	**46744**	5608	4439	57976
	Street	0	0	**80560**	8317	88877
	Grassland	1064	1065	29288	**109097**	140514
	Column Total	33597	48286	117312	121853	321048

Overall's Accuracy = (31348+ 46744+ 80560+ 109097) / 321048 = 83.4%

Producer's Accuracy

House = 31348/33597 = 93.3%

Street = 46744/48286 = 96.8%

Tree = 80560/117312 = 68.67%

Grassland = 109097/121853 = 89.53%

Users's Accuracy

House = 31348/33681 = 93.07%

Street = 46744/57976 = 80.63 %

Tree = 80560/88877 = 90.64%

Grassland = 109097/140514 = 77.64%

Table 4.1 Error Matrix

4.5 Comparison with Existing Classification Methods

In this section, comparison has been made between the classification accuracies of the developed classification system with some popular classification methods. Training data of Figure 4.15 was used for classification in all the three methods. Comparison was made with two other methods, Minimum Distance Classifier (MDC) [5] and Support Vector Machines [22] (without the use of hierarchical scheme). Table 4.2 and Table 4.3 show the error matrix for classification using MDC and SVM respectively. Figure 4.16 shows the overall accuracy the three classification methods. Figure 4.17 shows the user’s accuracy for different classes using the three classification methods. Figure 4.18 shows the producer’s accuracy for different classes. The overall accuracy using the proposed scheme is the highest.

Reference Data

		House	Street	Tree	Grassland	Row Total
Classified Data	**House**	**26253**	2523	3944	2917	35637
	Street	6412	**45375**	3272	844	55903
	Street	0	0	**65381**	2436	67817
	Grassland	932	388	44715	**115656**	161691
	Column Total	33597	48286	117312	121853	321048

Table 4.2 Error Matrix for MDC

Reference Data

		House	Street	Tree	Grassland	Row Total
Classified Data	**House**	**31391**	1376	3699	3025	39491
	Street	1274	**46522**	3828	941	52565
	Street	0	0	**62767**	2343	65110
	Grassland	932	388	47018	**115544**	163882
	Column Total	33597	48286	117312	121853	321048

Table 4.3 Error Matrix for SVM

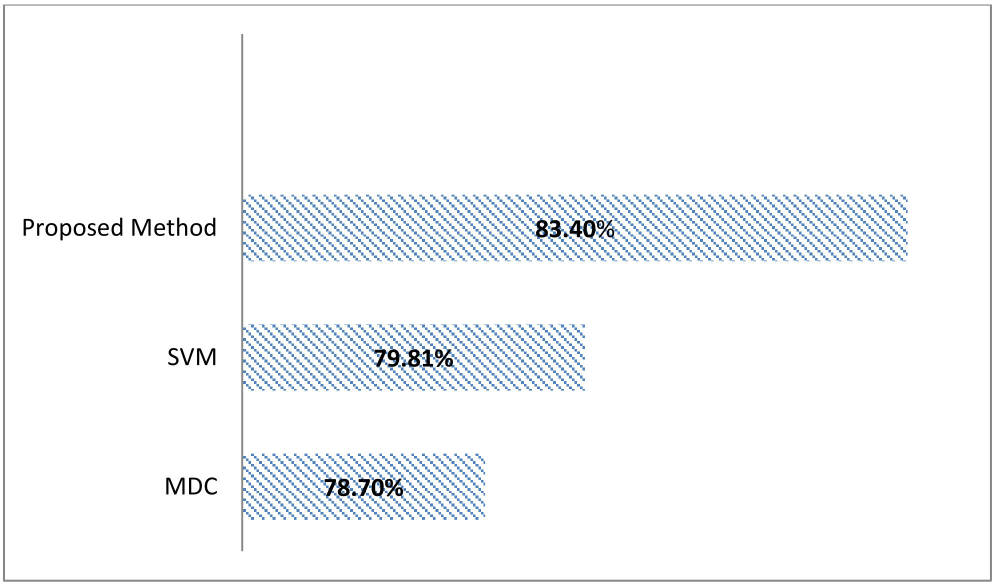


Figure 4.16 Comparison of Overall Accuracy

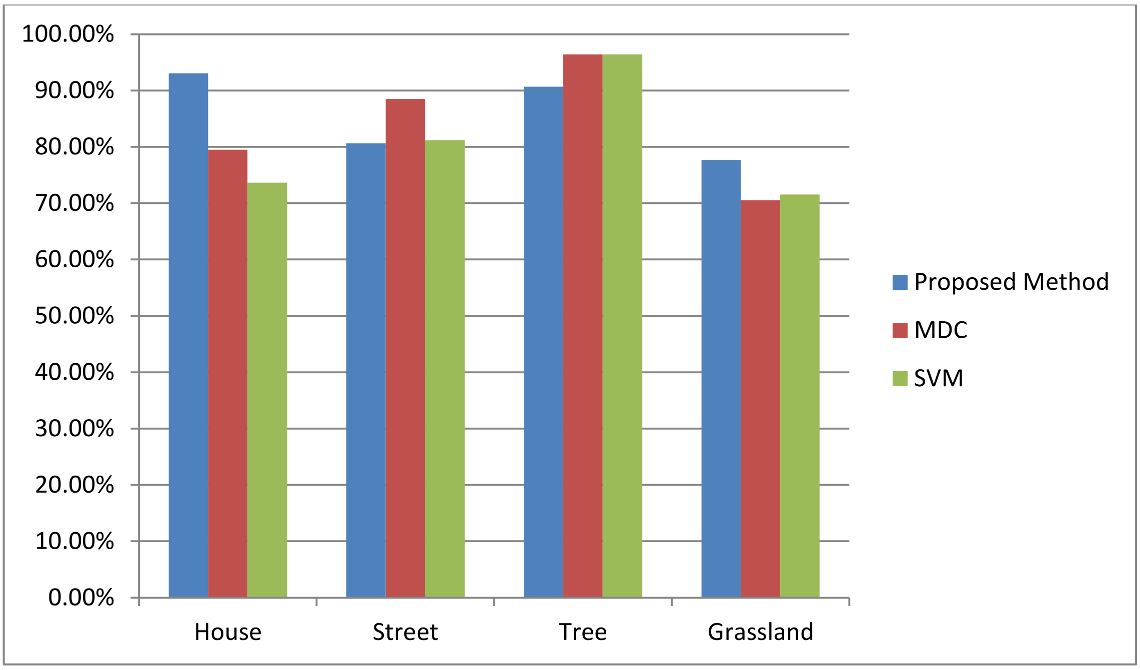


Figure 4.17 Comparison of user's accuracy

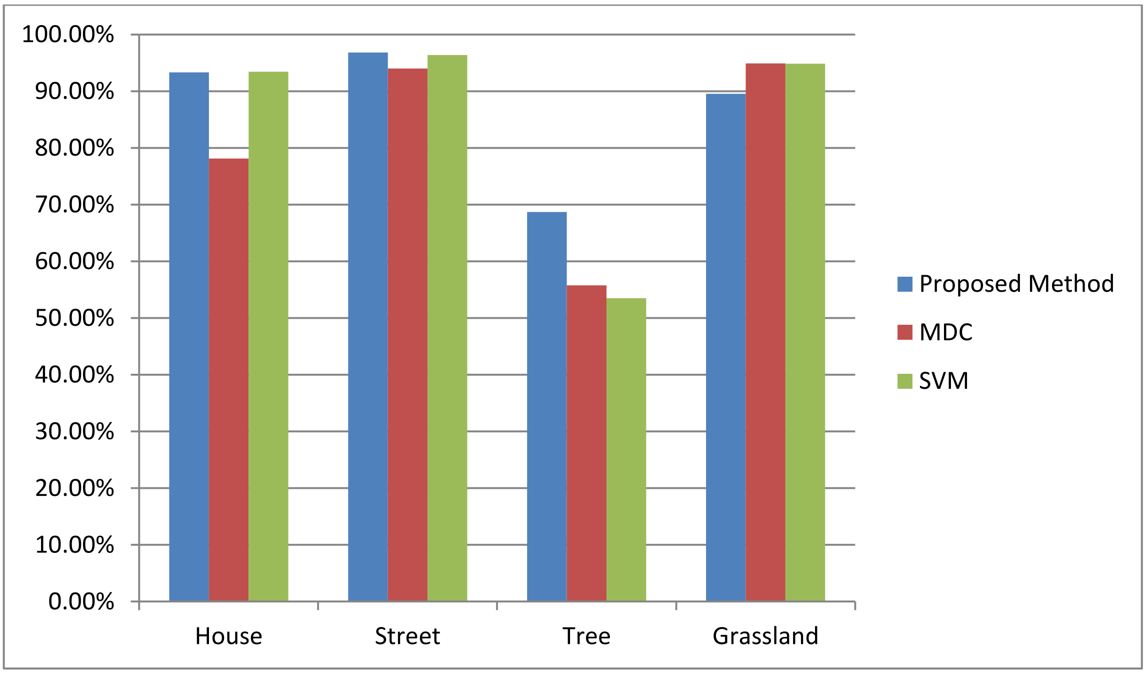


Figure 4.18 Comparison of producer's accuracy

Experimentations discussed in this chapter demonstrate the superiority of the proposed design over existing classification methods. It can be concluded Object Based Image Analysis is the future in deriving meaningful information form high resolution images. However a lot of research needs to be carried out in order to develop OBIA as an operationally established paradigm.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Remote sensing images on-board from a large variety of airborne borne sensors provide valuable data for about the earth's surface. The most common application of high resolution remote sensing images is extraction of land cover information map for the city governments to seek better planning and management practices with increasing urbanization.

In this dissertation, a new methodology based on the concept of Object Based Image Analysis (OBIA) was proposed for extraction of land cover information map from high resolution multispectral images. Results discussed in Chapter 4 demonstrate the superiority of the proposed design over existing classification methods. The following conclusions can be drawn from this dissertation,

1. A hierarchical classification scheme efficiently models the complexities in a geographic scene thereby leading to much accurate results.
2. Support Vector Machine is ideal for classification remote sensing data because it does not make an assumption about the statistical distribution of data. Furthermore, the use of kernel in SVM allows us to easily accommodate non linearity in the data.
3. The use of a Classification & Regression Tree (CART) for attribute selection was found to be sub-optimal because most of the time it dropped many important attributes and had to be reintroduced after manual analysis.
4. Object Based Image Analysis is the future in deriving meaningful information form high resolution images. However a lot of research needs to be carried out in order to develop OBIA as an operationally established paradigm.

5.2 Future Work

The work presented in this dissertation has led to some further directions, which may be worth considering in future,

1. Image segmentation is a time consuming task as it involves repeatedly performing segmentation at different scales until suitable results are obtained. Developing automatic scale selection methods is something worth looking into.
2. Developing a set of pre-defined class hierarchy models for frequently used geographic scenes could be done, so as to further ease the job an analyst.
3. Extending the proposed classification method to use topological attributes, this will further improve the accuracies of final result.
4. Standardizing different user parameters to be used in with different kinds of image sensors.

REFERENCES

1. Hay G.J., Castilla G., "Object-based image analysis: Strengths, weaknesses, opportunities and threats (SWOT). OBIA". *The international archives of the photogrammetry, remote sensing and spatial information sciences*, 2010.
2. Blaschke T., Burnett C., Pekkarinen A., "New contextual approaches using image segmentation for object-based classification", *Remote Sensing Image Analysis: Including the spatial domain, Kluver Academic Publishers, Dordrecht*, pp. 211-236, 2006.
3. Hay G.J., Blaschke, Marceau T., Bouchard A., "A comparison of three image-object methods for the multiscale analysis of landscape structure". *ISPRS Journal of Photogrammetry and Remote Sensing*, 57 (5 6), 327 345, 2010.
4. Benz C., Hofmann P., Willhauck G., Lingenfelder I., Heynen M., "Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information". *ISPRS Journal of Photogrammetry & Remote Sensing*, 58, 239– 258, 2009.
5. Gonzalez R.C., Woods R. E., Digital Image Processing 2nd Edition, Addison-Wesley Longman Publishing Co, 2001.
6. Matlab Documentation, http://www.mathworks.in/help/matlab/matlab_external/introducing-mex-files.html, Last Accessed : 24-05-2013.
7. Tabb M., Ahuja N., "Multiscale image segmentationby integrated edge and region detection". *IEEE Transactions on Image Processing*, pages 642–55, 1992.
8. Baatz M., Benz U., Dehghani S., Heynen M., Höltje A., "eCognition User Guide:Concepts and Methods", *Germany: Definiens Imaging*, 2007.
9. Nock R., Nielsen F., "Statistic region merging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1452–1458, 2011.
10. Lu D., Weng Q., "A survey of image classification methods and techniques for improving classification performance", International Journal of Remote Sensing, 28 (5), 823–870, 2007.
11. Coillie F., Verbeke L., Wulf R., "Feature selection by genetic algorithms in object-based classification of IKONOS imagery for forest mapping in Flanders", Belgium Remote Sensing of Environment, 110, pp. 476–487, 2006.

12. Lalibertea A.S., Browningb D.M., Rangob A., “A comparison of three feature selection methods for object-based classification of sub-decimeter resolution UltraCam-L imagery”, International Journal of Applied Earth Observation and Geoinformation 15, 70–78, 2012.
13. Carleer A. P., and E. Wolff, “Urban land cover multilevel region-based classification of VHR data by selecting relevant features”, International Journal of Remote Sensing, 27(6), pp. 1035-1051, 2006.
14. Marpu P. R., I. Niemeyer, S. Nussbaum, R. Gloaguen, “A procedure for automatic object-based classification”, Object-based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications. Springer, Berlin, Germany, pp. 169-184, 2008.
15. Yu Q., Gong P., Clinton N., Biging G., Kelly M., Schirokauer D., “Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery”, Photogrammetric Engineering and Remote Sensing, 72(7), pp.799-811, 2006.
16. Johansen K., Phinn S., Witte C., Philip S., Newton L., “Mapping banana plantations from object-oriented classification of SPOT-5 imagery”, Photogrammetric Engineering and Remote Sensing, 75(9), pp. 1069-1081, 2009.
17. Pu R., Landry S., Yu Q., ‘Object-based urban detailed land cover classification with high spatial resolution IKONOS imagery”. *Int. J. Remote Sens*, 32, 3285-3308, 2011.
18. Boser B.E., Guyon I.M., Vapnik V.N., “A training algorithm for optimal margin classifier”, *Fifth ACM Workshop on Computational Learning Theory*, pp. 144-52, 1992.
19. Vapnik V.N., The nature of statistical learning theory, New York : Springer, 1995.
20. Vapnik V.N., Statistical learning theory, New York : Wiley, 1998.
21. Vapnik V.N. and A.J. Chervonenkis, *Theory of pattern recognition,* Moscow : Nauka, 1974.
22. Burges C.J.C., “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Mining and Knowledge Discovery 2*, pp. 121-167, 1998.
23. Cortes C. and Vapnik V.N., “Support-vector networks”, *Machine Learning*, vol 20, pp. 273-97, 1995.
24. Bennett K.P. and Mangasarian O.L., “Robust linear programming discrimination of two linear inseparable sets”, *Optimization Methods and Software, vol. 1, pp. 23-34, 1992.*

25. Peng B., Zhang L., Zhang D., “Automatic image segmentation by dynamic region merging”, *IEEE Trans. On Image Processing*, No.99, pp. 1-14, 2011.
26. Timofeev R., “Classification and Regression Trees (CART) Theory and Applications”. Master Thesis CASE - Center of Applied Statistics and Economics Humboldt University, Berlin, 2004.
27. Dasgupta S., Papadimitriou C., Vazirini U., Algorithms, Tata McGraw-Hill Publishing Company Limited, 2008.
28. Chang C.C. and Lin C.J., http://www.csie.ntu.edu.tw/~cjlin/libsvm/, Last Accessed : 24-05-2013.
29. Arbelaez P., Fowlkes C.,Martin D., “Berkley Image Segmentation Dataset”, www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds,. June 2007.