TEXTURAL FEATURES BASED CLASSIFICATION OF SAR IMAGES

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree

of

INTEGRATED DUAL DEGREE

in

COMPUTER SCIENCE AND ENGINEERING
(With Specialization in Information Technology)

By

AJAY SHARMA

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2011

CANDIDATE'S DECLARATION

I hereby declare that the work is being presented in the dissertation ~~work~~ entitled "**Textural Features Based Classification of SAR Images**" towards the partial fulfilment of the requirement for the award of the degree of **Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology)** submitted to the **Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period from May, 2010 to June, 2011 under the guidance and provision of **Dr. Dharmenra Singh, and Dr. Rajdeep Niyogi, Department of Electronics and Computer Engineering, IIT Roorkee**.

I have not submitted the matter embodied in this dissertation work for the award of any other degree or diploma.

Date: June, 2011

Place: Roorkee

Ajay.

(AJAY SHARMA)

CERTIFICATE

This to certify that the work contained in the dissertation entitled "**Textural Features Based Classification of SAR Images**" by Ajay Sharma of Integrated Dual Degree in Computer Science and Engineering (with specialization in Information Technology), has not been submitted elsewhere for a degree or diploma to the best of my knowledge.

Date: June, 2011

Place: Roorkee

(Dr. Rajdeep Niyogi)
Assistant Professor,
E&CE Department,
IIT Roorkee, India

(Dr. Dharmendra Singh)
Associate Professor,
E&CE Department,
IIT Roorkee, India

ABSTRACT

Advances in remote sensing technologies have provided practical means for land use and land cover mapping which is critically important for landscape ecological studies. Satellite images contain grids of pixels from surfaces that may be evaluated, based upon their level of reflected radiation. But due the complexity of earth's terrain, and presence of various phenomenon (e.g. scatterings, absorption of radiation, noises etc), and low resolution of imagery, the quality of images is not always very good, and so there is a great need of good classifiers which can come up with good pattern recognition of the terrains.

In our work, we propose a classification method for satellite images which is based on texture features of the terrain. As the methods based on spectrum characteristics are very limited, texture based methods are becoming more and more popular because of their stable nature. In this proposed work, to compute various texture features, we have used gray level co-occurrence matrix (GLCM). GLCM gives a variety of, and multi-directional texture features. As most of the satellite imagery of low resolution, GLCM is a good option for computing texture features. We combine support vector machine with GLCM in the classification step. SVMs are becoming a very important tool in the field of classification because of their non-linear and non-probabilistic nature. As we don't have prior knowledge about pixel relationships in the terrain, SVM performs very well for it. In this method, we first compute feature vector with the help of GLCM in different directions, and then fed it along with a training vector to a SVMs decision tree for the final classification. We have applied this proposed method on synthetic aperture radar (SAR) images, which is a popular satellite imagery technique. Results shows that this proposed method can classify earth's terrain in number of different classes with a good accuracy. This method is implemented in MATLAB, with some prerequisites implemented with Envi (v 4.7) tool.

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. Dharmendra Singh**, and **Dr. Rajdeep Niyogi**, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for their trust in my work, their able guidance, regular source of encouragement and assistance throughout this dissertation work. I would state that the dissertation work would not have been in the present shape without their inspirational support and I consider myself fortunate to have done my dissertation under them.

I also extend my sincere thanks to **Dr. S.N. Sinha**, Professor and Head of the Department of Electronics and Computer Engineering for providing facilities for the work.

I also wish to thank **Mr. Rishi Prakash**, and **Ms. Pooja Mishra,** Ph.D. students at Indian Institutete of Technology Roorkee, India for their valuable suggestion and timely help.

I would like to thank all my friends, and colleagues at Remote Sensing Lab, IIT Roorkee, vho supported and encouraged me to finish this work.

inally, I would like to say that I am indebted to my parents for everything that they have iven to me. I thank them for sacrifices they made so that I could grow up in a learning nvironment. They have always stood by me in everything I have done, providing constant upport, encouragement, and love.

AJAY SHARMA

CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Introduction and Motivation

Advances in satellite imagery technologies have provided practical means for land use and land cover mapping which is critically important for landscape ecological studies. Satellite images contain grids of pixels from surfaces that may be evaluated, based upon their level of reflected radiation. Those values are dependent upon reflection, transmission and scattering of electromagnetic illumination and the reflected values are assigned numbers in bit ranges. However, remote sensing data is subject to low geometric resolution and different kinds of errors, and these errors can be carried over or propagated in subsequent landscape images. A pixel is subject to absorption of solar radiation, scattering to the sides as well as reflectance back to the sensor. Clearly, one pixel (area) can and does affect those around it, particularly through scattering. Boundaries may be very clear and easy to determine for some objects but for others the boundaries are indeterminate and are often fuzzy and some phenomenon might overlap. Soils, water quality, incidence of disease, fire intensity, pavement condition and ecological zones are good examples of entities that often do not have distinct sharp boundaries. Pixel values for these areas tend to vary due to the fact that they do not necessarily occur as homogenous representations. To deal with these problems 'Pattern Classification' is done on satellite data. Pattern classification is used to determine patterns and objects through the evaluation of individual pixels and their relationship to each other.

1.1.1 Satellite Images

Satellite sensors record the intensity of electromagnetic radiation reflected from the earth at different wavelengths. Energy that is not reflected by an object is absorbed. Each object has its own spectrum. Remote sensing relies on the fact that particular features of landscape such as crops, bush, water bodies etc reflect radiation differently at different wavelength [1]. Instruments mounted on the satellites detect and record the energy that has been reflected. The detectors are sensitive to particular range of radiation called bands. Another feature that characterises a satellite image is called 'resolution of the image'. This is the most important characteristic of satellite images and is described in next section.

1.1.2 Synthetic Aperture Radar (SAR) Images

A Synthetic Aperture Radar (SAR) system illuminates a scene with microwaves and records both the amplitude and the phase of the back-scattered radiation, making it a coherent imaging process. The received signal is sampled and converted into a digital image. The advent of SAR sensors lead to the concept of radar polarimetry. *Radar Polarimetry is the merging of the technological concept of radar (radio detection and ranging) and of the fundamental property of transverse nature of electromagnetic waves.* It is the science of acquiring, processing and analyzing the polarization state of an EM field [2].

Fully polarimetric SAR acquires four channels to obtain the complete scattering matrix, wherein the signal is transmitted in two orthogonal polarizations and received at two orthogonal polarizations. With Polarimetric radars it has become possible to extract more information available than conventional radars due to the preservation of phase term. The incorporation of coherent polarimetric phase and amplitude into radar signal and image processing promises to bring about further improvements in monitoring capabilities in SAR image analysis. The phase information, along with the conventional magnitude data, can be used to study the scattering mechanisms and resolve the ambiguities about the source of scattering [2]. This unique characteristic of polarimetric imaging radar makes it a powerful tool for land cover classification. The possible reasons which make polarimetric SAR a useful tool to characterize various targets of ecosystem for classification are mentioned below:

- SAR being an active sensor is a day light acquisition system (unlike optical sensors).
- Most of the radar sensors exhibit all weather capability. It can be seen that atmospheric characteristics such as cloud, light rain, haze, and smoke has little effect on the capability of RADAR data acquisition system as attenuation of atmosphere is negligible for wavelengths $\lambda > 3$ cm [3].
- SAR is not only sensitive to the dielectric, physical and geometric properties of various land cover types, but is also sensitive to the relative proportion and distribution of various scatterers within an area-extended target.
- SAR not only provides ground surface information but can also be used for obtaining information beneath the ground (for certain moisture value and ground density) due to its capability to penetrate into soil and vegetation canopy.

1.1.3 Resolution of Satellite Images

There are four types of resolution when discussing satellite imagery in remote sensing: spatial, spectral, temporal, and radiometric [4].

Spatial resolution : Spatial resolution is defined as the pixel size of an image representing the size of the surface area that is being measured on the ground,and is determined by the sensors' instantaneous field of view (IFOV) [Appendix 1].

Spectral resolution : Spectral resolution is defined by the wavelength interval size (discreet segment of the Electromagnetic Spectrum) and number of intervals that the sensor is measuring.

Temporal resolution : Temporal resolution is defined by the amount of time that passes between imagery collection periods.

Radiometric resolution : radiometric resolution is defined as the ability of an imaging system to record many levels of brightness (contrast for example). Radiometric resolution refers to the effective bit-depth of the sensor (number of greyscale levels) and is typically expressed as 8-bit (0-255), 11-bit (0-2047), 12-bit (0-4095) or 16-bit (0-65,535).

Often 'geometric resolution' is used to describe the resolution of a satellite image:

Geometric resolution : Geometric resolution refers to the satellite sensor's ability to effectively image a portion of the Earth's surface in a single pixel and is typically expressed in terms of Ground Sample Distance, or GSD [Appendix 2]. For example, if GSD is 30m, it means the smallest unit that maps to a single pixel within an image is 30m x 30m.

Spatial resolution is considered to be the most important, as size of earth is very large, so it is very important how much area of surface is contained in one pixel. On the basis of it, satellite images are classified as low resolution or high resolution images.

1.1.4 Pattern Recognition

Pattern recognition is the act of taking in raw data and taking an action based on the category of the pattern. Pattern recognition aims to classify data (patterns) based either on a priori knowledge or on statistical information extracted from the patterns [5]. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space. In other words Pattern Recognition are techniques for classifying a set of objects into a number of distinct classes by considering similarities of objects belonging to the same class and the dissimilarities of objects belonging to different classes.

Basic Concepts

A classifier model and its associated training algorithm are all that are usually associated with pattern recognition. However, a complete pattern recognition system consists of several components, shown in Fig. 1. These components can be described as following :

Data Acquisition : Apart from employing an appropriate classification algorithm, one of the most important requirements for designing a successful pattern recognition system is to have adequate and representative training and test datasets. Adequacy ensures that a sufficient amount of data exists to learn the decision boundary as a functional mapping between the feature vectors and the correct class labels.

Preprocessing : Its the goal is to condition the acquired data such that noise from various sources are removed to the extend that it is possible. Various filtering techniques can be employed if the user has prior knowledge regarding the spectrum of the noise.

Feature Extraction : Both feature extraction and feature selection steps (discussed next) are in effect dimensionality reduction procedures. In short, the goal of feature extraction is to find preferably small number of features that are particularly distinguishing or informative for the classification process, and that are invariant to irrelevant transformations of the data.

Feature Selection : Feature selection means selection of m features that provide the most discriminatory information, out of a possible d features, where $m<d$. In other words, feature selection refers to selecting a subset of features from a set of features that have already been identified by a preceding feature extraction algorithm.

Model Selection and Training : Only after acquiring and preprocessing adequate and representative data and extracting and selecting the most informative features is one finally ready to select a classifier and its corresponding training algorithm. One can think of the classification as a function approximation problem i.e. find a function that maps a d-dimensional input to appropriately encoded class information (both inputs and outputs must be encoded, unless they are already of numerical nature).

Performance Evaluation : Once the model selection and training is completed, its generalization performance needs to be evaluated on previously unseen data to estimate its true performance on field data. One of the simplest, and hence, the most popular methods for evaluating the generalization performance is to split the entire training data into two partitions, where the first partition is used for actual training and the second partition is used for testing the performance of the algorithm. The performance on this latter dataset is then used as an estimate of the algorithm's true (and unknown) field performance.

PATTERN RECOGNITION

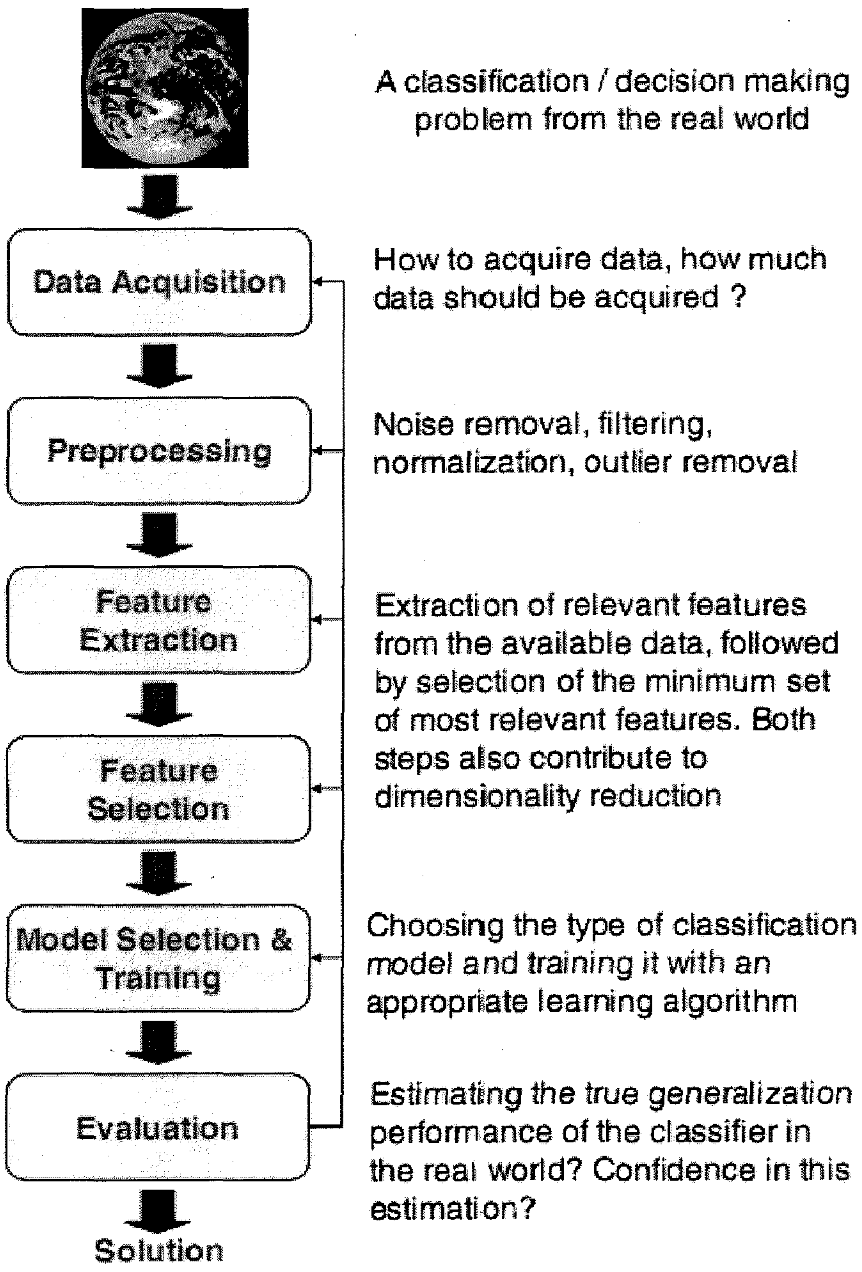


Figure 1.1 Components of a pattern recognition system

1.1.5 Supervised Classification

In supervised classification land cover classes are defined. Sufficient reference data are available and used as training samples. The signatures generated from the training samples are then used to train the classifier to classify the spectral data into a thematic map [5]. Examples : Naive Bayes classifier, K-Nearest Neighbour(KNN) , artificial neural network, decision tree classifier.

1.1.5 Unsupervised Classification

In Unsupervised classification approach Clustering-based algorithms are used to partition the spectral image into a number of spectral classes based on the statistical information inherent in the image. No prior definitions of the classes are used. The analyst is responsible for labelling and merging the spectral classes into meaningful classes [5].

Examples : ISODATA, K-means clustering algorithm etc.

1.2 Statement of the Problem

The problem statement for the proposed research work is as follows:

"*To design technique for Classification of Satellite Images using Texture Features*"

The following aspects have been considered while designing techniques:

- The proposed algorithms are designed for SAR data.
- Low resolution of satellite imagery has been considered.
- The various types of terrain e.g. water, land, urban has been considered.
- Accuracy and computational efficiency of algorithm has been considered.

1.3 Organization of the Dissertation

This report comprises of six chapters including this chapter that introduces the topic and states the problem. The rest of the dissertation report is organized as follows.

Chapter 2 discusses the related work and research done in the field of classification of satellite images. First, we introduce the concept of satellite imagery data and general techniques of handling such data and then we discuss previously proposed techniques of classifying satellite images. We also discuss in details the performance of these techniques in various scenarios.

Chapter 3 provides a detailed description of proposed algorithm for unsupervised classification, along with the description of the SAR data used. This chapter also includes the details of preprocessing done on the data.

Chapter 4 gives the brief description of the implementation of the proposed algorithm. It includes the description and implementation of various modules in algorithm.

Chapter 5 discusses the results and including discussion on them. It also provides an analysis on important performance parameters.

Chapter 6 concludes the work and gives the directions for future work.

CHAPTER 2
BACKGROUND AND LITERATURE REVIEW

Advances in satellite imagery technologies have provided practical means for land use and land cover mapping which is critically important for landscape ecological studies. There are lots of satellites which record different terrains of earth with various technologies and provide data in various formats. Today, we can get satellite covering of almost any place at all times. Patterns from this data have given us a huge opportunity to make most use out of it. We can set our parameters and, can assign and analyze various classes for our purpose. So, good classification is the most important part of satellite imagery, if we want to make this useful to us. There has always been lots of work in the field of pattern recognition from the very beginning of satellite era, which is still going on with a faster pace all around the world.

There are many Image Processing methods that have been used to classify satellite images using Texture Analysis till date. Many texture models have been proposed for classification of satellite images like Gray Level Co-occurrence Matrix method, MRF models, Gabor Wavelets, Contourlet etc. A brief literature review is presented below for classification of Satellite images using Texture parameters.

2.1 Texture Features

In satellite imagery, texture is considered a very important feature for various applications e.g. classification, segmentation, identification, and so on. By contrast with the spectrum characteristics, texture of satellite imagery is considered to be more stable and therefore is of great importance. Various algorithms for texture classification have been given over the time. We will review some of the important ones of these methods.

Haralick *et al* (1973) [6] proposed general procedure for extracting textural properties of blocks of image data. These features were calculated in the spatial domain and the statistical nature of texture was taken into account in the procedure, which was based on the assumption that the texture information in an image I was contained in the overall or "average" spatial relationship which the gray tones in the image have to one another. They computed a set of gray tone spatial-dependence probability-distribution matrices for a given image block and suggested a set of 14 textural features, which can be extracted from each of these matrices. These features contained information about such image textural characteristics as homogeneity, gray-tone linear dependencies (linear structure), contrast, number and nature of boundaries present, and the complexity of the image. It was important to note that the number of operations required computing any one of these features was proportional to the number of resolution cells in the image block. It was for this reason that these features were called quickly computable.

H.H.Loh *et al.* (1988) [7] presented a family of texture features that had the ability to discriminate different textures in a 3-D scene as well as the ability to recover the range and orientation of the surfaces of the scene. These texture features were derived from the gray level run length matrices (GLRLM's) of an image. The GLRLM's were first normalized so that they all had equal average gray level run length. Features extracted from the normalized GLRLM's were independent of the surface geometry. Experiments for the discrimination of natural textures had been conducted. The results demonstrated that these features had the ability to discriminate different textures in a nontrivial 3-D scene.

Chung-Ming Wu. e*t al.* (1992) [8] studied the classification of ultrasonic liver images by making use of some powerful texture features, including the spatial gray-level dependence matrices, the Fourier power spectrum, the gray-level difference statistics, and the Laws□ texture energy measures. In this paper, features of these types were used to classify three sets of ultrasonic liver images-normal liver, hepatoma, and cirrhosis.

D. Patel *et al.* (1994) [9] performed texture analysis for foreign object detection using a single layer neural network and PCA. The foreign objects (FOs) occur extremely rarely, but their presence causes alarm to both the consumer and the food manufacturer. FOs which can potentially occur in food products varied in form and size; they ranged from hard contaminants such as slivers of glass to soft contaminants such as small pieces of wood. This work presented a method to detect FOs in bags of frozen vegetables and in particular using bags of frozen corn kernels. The final stage of the production process was the inspection process where the bags had been packed and sealed. Then X-ray imaging was used to view the contents of the bag. Convolution masks sensitive to the particular local structural attributes were used to extract texture features and, based on these features, the FOs in the image were detected and segmented. These filters responded to the peculiarities of the structure of the texture and subsequently detected any anomalies, such as FOs within the food substrate. They also used principal component analysis (PCA) techniques to find orthogonal vectors in data space that accounted for as much as possible of the variance of the data. The vectors were then used as the coefficients of the convolution masks. The texture characterisation was based on a convolution operator which "adapts" and "learns" from representation of food samples containing no contaminants using the generalised Hebbian rule. The detected FOs was discrepancies within the food substrate which were highlighted by the convolution process.

Anne H. Schistad Solberg *et al.* (1995) [10] compared the performance of a number of different texture features for SAR image analysis. These features were derived from the gray-level co-occurrence matrix, local statistics, and lognormal random field models. The features were compared based on: (i) their invariance or robustness with respect to natural changes in SAR signatures; and (ii) their discrimination ability and classification accuracy. The invariance of the texture features was investigated on a set of 13 ERS-1 SAR images of the same scene, captured under different conditions. Two main conclusions were drawn from this study, firstly, the texture features were not invariant with respect to natural changes in the mean backscatter values; and secondly, texture fusion and selection by combining texture features obtained by different models significantly improved the classification accuracy.

A.H.Mir *et al.* (1995) [11] did some work in the use of texture for the extraction of diagnostic information from CT scan images. They obtained a number of features from abdominal CT scans of liver using the spatial domain texture analysis methods. This study investigated that whether texture could be used to discriminate between various tissue types that were inaccessible to human perception and if so, and then what were the most useful parameters. The result was that the texture features were helpful in diagnosing the onset of disease in liver tissue, which cannot be done by visible eye. Three texture features namely - entropy, homogeneity, and gray level distribution were found to be effective. The performance of these features was compared on the basis of significance tests. The results showed that mainly all the parameters could detect the early malignancy with a confidence level of above 99%.

Li Ma *et al.* (2003) [12] in their work described a new scheme for iris recognition from an image sequence. Iris recognition, as an emerging biometric recognition approach, has become a very active topic in both research and practical applications. The main concern of the research was the Iris recognition. First of all, the quality of each image in the input sequence was assessed and a clear iris image was selected. A quality descriptor based on the Fourier spectra of two local regions in an iris image was defined to discriminate clear iris images from low quality images due to motion blur, defocus, and eyelash occlusion. According to the distinct distribution of the iris characteristics, a bank of spatial filters was constructed for efficient feature extraction. A comparative study of existing methods for iris recognition was conducted on an iris image database including 2,255 sequences from 213 subjects. The experimental results demonstrated the effectiveness of the proposed method. A detailed performance comparison of existing methods for iris recognition had been conducted on the CASIA Iris Database. Conclusions based on such a comparison using a nonparametric statistical method (the bootstrap) provided useful information for further research.

A.Abhyankar *et al.* (2006) [13] proposed a novel approach to detect "liveness" associated with fingerprint scanners. The approach was based on underlying texture and density of the fingerprint images. The algorithm combined the features derived from multi resolution texture analysis as well as derived from local ridge frequency analysis. The features were further processed using FCM and error rates were calculated. Based on the association of all the points to particular type of a class, the classification rates were calculated. The training was performed through PCA and first two components were used to map the data non-linearly. This method was tested for three different types of scanner technologies, namely optical, opto-electrical, capacitive DC. Overall 95.36% classification rate was obtained. Advantages of this method were that it was purely software based and required only one image.

N. Agani *et al.* (2007) [14] proposed an algorithm for texture analysis. The proposed algorithm was a combination of wavelet extension transform with gray level cooccurrence matrix. Texture analysis provides a significant role in identification and classification of surface from medical images and many other types of images. It has a vital role in automatic visual inspection in a number of applications like industrial monitoring of product quality control, remote sensing and medical diagnosis. In this research, medical images of echocardiography (ECG) of heart were used to identify heart with suspected myocardial infarction problem. The wavelet extension transform was used to form image approximation with higher resolution. The gray level cooccurrence matrices were computed for each sub band. These matrices were then used to extract four feature vectors: entropy, contrast, energy (angular second moment) and homogeneity (inverse difference moment). Mahalanobis distance classifier was used in this work. The method was tested with clinical data from echocardiography images of 17 patients. For each patient, tissue samples were taken from suspected infected area as well as from non-infected (normal) area. For each patient, 8 frames separated by some time interval were used and for each frame, 5 normal regions and 5 suspected myocardial infarction regions of 16×16 pixel size were analyzed. The performance of the proposed method was compared to that energy base of wavelet image extension transform based on energy. It was found that the classification rate obtained using the wavelet extension based on energy technique was 81.62%, while the proposed method achieved 91.32% accuracy.

F. Zhang *et al.* (2008) [15] identified oil spills on the basis of textural information of marine SAR images. Oil spill pollution is a major environmental threat for many countries in the world, which can cause serious damage to marine environment. Synthetic aperture radar (SAR) has become a valuable tool for marine oil spill monitoring, because of its all-weather and all-day capabilities. In this work, cooccurrence matrix method was employed to extract textural features of marine SAR image first, then these features were analyzed and optimized, and then Support Machine Vector (SVM) method was employed to discriminate oil spill and other look-alikes in SAR images.

Vijaya V. Chamundeeswari *et al.* (2009) [16] analysed the role of various intensity and textural measures for their discriminative ability for unsupervised SAR image classification into various land cover types like water, urban, and vegetation areas. To make the algorithm adaptable, these textural features were fused using principal component analysis (PCA), and principal components were used for classification purposes. To highlight the effectiveness of PCA, the difference between PCA- and non-PCA-based classifications was also analyzed. The analysis of how every individual feature measure contributes for classification process has been presented, and then, textural measures for a feature set were chosen according to their role in improving classification accuracy. By analysis, it was observed that the feature set comprising mean, variance, wavelet components, semivariogram, lacunarity, and weighted rank fill ratio provided good classification accuracy of up to 90.4% than by using individual textural measures, and this increased accuracy justified the complexity involved in the process.

A.Kourgli *et al.* (2009) [17] developed a new nonparametric MRF model of textures which resulted in a new approach for fast texture analysis, synthesis and classification. The proposed method described a texture by identifying the spatial distribution of its neighbourhoods and deriving a likeness measure of their spatial organization. The identification conducted to the definition of a new nonparametric MRF model using the likeness measure that provided a consistent estimate of the joint distribution of pixels in a window of optimal size. For texture synthesis, they introduced a multi-scale approach incorporating discrete optimization. As a result, the method lead to fast and efficient texture synthesis and texture identification algorithms. The model proposed seems suitable for the practical application of terrain mapping of SAR images.

Anguela T. *et al.* (2010) [18] used textural features of the soil, and studied spatial variability of soil parameters. They established relationships between backscattering coefficients and soil's texture. Texture of soil and local roughness, along with backscattering coefficients were used for developing an empirical method to calculate moisture content of soil to the scale of individual field. Results show that using texture along with local parameters variation in soil moisture can be retrieved with a root-mean-square error 0.05 cm^3.

Ouma Y. O. *et al.* (2010) [19] used discrete wavelet transform (DWT) and grey-level-cooccurrence-matrices (GLCMs) to derive texture features to recognize and extract urban land cover information. They were able to recognize residential buildings, commercial buildings, parkings, roads, and green vegetation. The DWT filters capture the lower and mid-frequency texture information, whereas the GLCM captures the high-frequency textural components, for the same scene features. Edge information is arguably derived from the multi-scale and multi-directional components of the DWT. Results showed that DWT and GLCMs are essential for improving recognition and extraction of tested urban land cover.

Choudhary P.R. *et al.* (2011) [20] presented a study towards machine generation of landform maps from optical remote sensing data. Their approach uses an offline trained multilayer perceptron (MLP) as a classifier, which is subsequently used to identify the landform classes in a satellite image. The paper emphasizes building a reasonably extensive database using multispectral images from which relevant texture information is computed. Gray level co-occurrence texture statistics, which form the feature vector representing the pattern, are used for training the MLP. Results suggest that the textural method is promising for machine extraction of the landforms.

A lot of work has been done on use of texture features, frequency spectrum features etc for classification of SAR images. But still there is scope for a simple but effective classification technique based on the above reviewed parameters. The SAR images have been classified using statistical approaches, MRF models and Gabor filters in the review presented above.
Now, We will review another powerful method for non-linear classification, Support Vector Machines (SVMs), in the following section.

2.2 Support Vector Machines (SVM)

SVMs are semi-supervised classifiers. They becoming very popular in the field of classification for their non-probabilistic nature, and their adaptability to a vast number of kernel function. In the field of satellite imagery classification, SVM is not a very old phenomenon, but is getting popular day-by-day. We will review SVMs in the following works :

Vapnik V. (1995) [21] gave the idea of support-vector networks as a new learning machine for two-group classification problems. The machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high-dimension feature space. In this feature space a linear decision surface is constructed. Special properties of the decision surface ensures high generalization ability of the learning machine. High generalization ability of support-vector networks utilizing polynomial input transformations is demonstrated. They also compared the performance of the support-vector network to various classical learning algorithms that all took part in a benchmark study of Optical Character Recognition.

Brown M. *et al.* (2000) [22] applied linear spectral mixture models and support vector machines for *Remote Sensing*. Their work compared a well-established technique, linear spectral mixture models (LSMM), with a much newer idea based on data selection, support vector machines (SVM). It was shown that the constrained least squares LSMM is equivalent to the linear SVM, which relies on proving that the LSMM algorithm possesses the "maximum margin" property. This in turn showed that the LSMM algorithm can be derived from the same optimality conditions as the linear SVM, which provides important insights about the role of the bias term and rank deficiency in the pure pixel matrix within the LSMM algorithm. It also highlights one of the main advantages for using the linear SVM algorithm in that it performs automatic "pure pixel" selection from a much larger database. In addition, extensions to the basic SVM algorithm allow the technique to be applied to data sets that exhibit spectral confusion (overlapping sets of pure pixels) and to data sets that have nonlinear mixture regions.

Haigang Zhan *et al.* (2003) [23] applied support vector machines to retrieve the concentration of oceanic chlorophyll. They used non-linear transfer function between chlorophillic concentration and marine reflectance. Results showed that SVM perform as good as a optimal multi-layer perceptron (MLP).

Mantero P. *et al.* (2005) [24] implemented SVM based probability density functions for semi-supervised classification of satellite images. In this work, a classification strategy is described that allows the identification of samples drawn from unknown classes through the application of a suitable Bayesian decision rule. The proposed approach is based on support vector machines (SVMs) for the estimation of probability density functions and on a recursive procedure to generate prior probability estimates for known and unknown classes.

Fei B. *et al.* (2006) [25] presented a new architecture named Binary Tree of support vector machine (SVM), or BTS, in order to achieve high classification efficiency for multiclass problems. BTS decreased the number of binary classifiers to the greatest extent without increasing the complexity of the original problem. In the training phase, BTS has N-1 binary classifiers in the best situation (N is the number of classes), while it has $\log_{4/3}((N+3)/4)$ binary tests on average when making a decision. At the same time the upper bound of convergence complexity is determined. The experiments in this paper indicate that maintaining comparable accuracy, BTS is much faster to be trained than other methods. Especially in classification, due to its Log complexity.

Mathur A. *et al.* (2008) [26] extended binary SVM for a one-shot multiclass classification needing a single optimization operation. Here, an approach for one-shot multi- class classification of multispectral data was evaluated against approaches based on binary SVM for a set of five-class classifications. This approach reduces the number of SVMs needed, and lesser number of optimization steps.

Lardeux *et al.* (2009) [27] applied support vector machines to classify multi-frequency SAR polarimetric data. They evaluated the contribution of different polarimetric indicators that can be derived from a fully polarimetric data set. They deduced that because of SVM's ability to take numerous and heterogeneous parameters into account, such as the various polarimetric indicators under consideration, it can classify with great accuracy. The results are compared to those obtained with the standard Wishart approach, for single frequency and multifrequency bands. It is shown that, when radar data do not satisfy the Wishart distribution, the SVM algorithm performs much better than the Wishart approach, when applied to an optimized set of polarimetric indicators.

Zhe Liu *et al.* (2009) [28] used contourlet and support vector machines for SAR image segmentation. They computed various texture features with the help of contourlet transform, and build feature vector. This feature vector was then fed to SVMs decision tree for multi-class classification. SVM is used for its excellent nonlinear classification nature. This work classified three classes successfully. This work establishes that SVMs are a good choice for high dimension data, and work better than conventional methods.

Tarabalka Y. *Et al.* (2010) [29] used support vector based method for accurate classification of hyperspectral images. Due to high number of spectral bands acquired by hyperspectral sensors increases the capability to distinguish physical materials and objects, presenting new challenges to image analysis and classification. They used SVMs and Markov random fields for classification. Results shows that this method improved the accuracy of the classification as compared to spectral-spatial classification techniques.

We observe from above reviews that support vector machines (SVMs) are being used increasingly in the field of classification. The reasons for this are - non-linear nature of SVMs, can work with high-dimensional data, more efficient than multi-layer perceptron, easy transition from bi-class to multi-class problem, can handle various type of data etc.

2.3 Research Gaps

On the basis of reviews performed in sections 2.1 and 2.2, we observe following research gaps:

- There is a need to explore texture-based classification field, as classifiers based on texture are proving to be more stable than spectrum-based classifiers for SAR images.
- Multiclass SVMs can prove to be a good classification tool because of their non-linear and non-probabilistic nature.
- There is a need for a adaptive classifier for satellite imagery, as most of the times we don't have actual terrain info, so probabilistic models can fail in these cases.
- Earth terrain contains mixed and overlapping pixels, so there is a need for a classifier which can handle these scenarios. SVMs are a good option for this.

Analyzing these research gaps, the objective of our work is to design a classifier which is adaptive, stable, non-probabilistic, non-linear in nature, and can classify mixed or overlapping pixels with good accuracy.

CHAPTER 3

Proposed Texture Features Based Classifier Using Gray Level Co-occurrence Matrices (GLCMs) and SVMs Decision Tree, and Comparison with Some Standard Classifiers

Classification has been a very important area of research in the field of *Remote Sensing*, and due to various constraints (e.g. resolution limit) and issues (e.g. scatterings), this field has been equally challenging. There are several methods available for satellite images now a days, which uses various schemes and features of imagery to improve classification. Using texture features to classify satellite images has added an another dimension in the field of classification, and texture features based classifiers are proving to be much better than the traditional methods which don't uses texture features. In this chapter, we will discuss texture features of satellite images, GLCMs, SVMs decision trees, and after that we will discuss our proposed classifier. But first we will discuss the various parameters to evaluate a classifiers.

3.1 Parameters For Evaluating A Classifier

Most important parameter to evaluate a classifier is its accuracy. Accuracy is defined as the percentage of pixels which are classified correctly in their classes. In satellite imagery, accuracy is very critical, and of central importance, as earth's terrain is very complex and probability of mixed pixels is very high. So, a classifier is considered to be as good as its accuracy. Other parameters include time complexity and memory complexity of the method. We will consider only accuracy parameter in our work as it's the criterion to judge betterment of a classifier over others in most of the cases.

3.2 Texture Features of Satellite Images

Texture is used to describe the feel of non-tactile sensations. Textures are characteristic intensity variations that typically originate from roughness of object surfaces. For a well-defined texture, intensity variations will normally exhibit both regularity and randomness and for this reason texture analysis requires careful design of statistical measures.

Texture is a combination of repeated patterns with a regular frequency. In visual interpretation texture has several types, for example, smooth, fine, coarse etc. Texture analysis is defined as the classification or segmentation of textural features with respect to the shape of a small element, density and direction of regularity. Texture provides a way to segment and recognise the surfaces.

There are a number of definitions of texture that have been formulated by different people depending upon the particular application and there are no generally agreed upon definition.

3.2.1 Properties of Texture

- Texture is a property of areas and texture of a point is undefined. So, texture is a contextual property and its definition must involve gray values in a spatial neighbourhood.
- Texture involves the spatial distribution of gray levels. Thus, two-dimensional histograms or co-occurrence matrices are reasonable texture analysis tools.
- Texture in an image can be perceived at different scales or levels of resolution.

3.3 Texture Analysis Methods

Following are the different texture analysis methods :

1) Structural Methods
2) Statistical Methods
3) Model-based Methods, and
4) Transform-based Methods

3.3.1 Structural Method

Structural approaches represent texture by well defined primitives (micro texture) and a hierarchy of spatial arrangements (macro texture) of those primitives. To describe the texture, one must define the primitives and the placement rules. The choice of a primitive (from a set of primitives) and the probability of the chosen primitive to be placed at a particular location can be a function of location or the primitives near the location. [Haralick, 1979]

The advantage of the structural approach is that it provides a good symbolic description of the image; however, this feature is more useful for synthesis than analysis tasks. The abstract descriptions can be ill defined for natural textures because of the variability of both micro- and macrostructure and no clear distinction between them.

3.3.2 Statistical Method

In contrast to structural methods, statistical approaches do not attempt to understand explicitly the hierarchical structure of the texture. Instead, they represent the texture indirectly by the non-deterministic properties that govern the distributions and relationships between the grey levels of an image. Methods based on second-order statistics (i.e. statistics given by pairs of pixels) have been shown to achieve higher discrimination rates than the power spectrum (transform-based) and structural methods. Accordingly, the textures in grey-level images are discriminated spontaneously only if they differ in second order moments. Equal second order moments, but different third order moments require deliberate cognitive effort.

The statistical approach exploits the statistical properties of image or image regions in a bottom up fashion, starting from the pixel values in the neighbourhood. The cooccurrence matrix is in wide use in representing the dependence in the distributions of gray-level [4]. The co-occurrence matrix is a function of:

1) The image region,

2) A displacement vector, and

3) The number of gray-levels after quantization.

The matrix contains frequencies of co-occurrence of two gray-levels. After normalization, it becomes a probability matrix with all the elements summing up to 1.

3.3.3 Model-based Method

Model based texture analysis using fractal and stochastic models, attempt to interpret an image texture by use of, respectively, generative image model and stochastic model. The parameters of the model are estimated and then used for image analysis. In practice, the computational complexity arising in the estimation of stochastic model parameters is the primary problem. The fractal model has been shown to be useful for modelling some natural textures. It can be used also for texture analysis and discrimination; however, it lacks orientation selectivity and is not suitable for describing local image structures.

3.3.4 Transform-based Method

Transform methods of texture analysis, such as Fourier and wavelet transforms represent an image in a space whose co-ordinate system has an interpretation that is closely related to the characteristics of a texture (such as frequency or size). Methods based on the Fourier transform perform poorly in practice, due to its lack of spatial localization. Gabor filters provide means for better spatial localization; however, their usefulness is limited in practice because there is usually no single filter resolution at which one can localize a spatial structure in natural textures.

The statistical approach has been used in our work. It is the most widely used and more generally applied method because of its high accuracy and less computation time. Texture statistics is frequently classified into first-order, second-order and high order statistics. They are referring to the gray level distribution of pixel on an image. The gray scale is a black and white image at any given focus of pixel, typically there is a corresponding intensity on a range from 0 (black) to 255(white). That means an image is composed of an array of pixels of varying intensity across the image, the intensity corresponding to the level of grayness from black (0) to white (255) at any particular point in the image.

3.4 Gray Level Co-occurrence Matrices (GLCMs)

The gray-level co-occurrence matrix (GLCM), a frequency matrix, is a useful method for enhancing details and is frequently used as an aid for interpretation of an image. The GLCM is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image. The GLCM indicates the frequency of a pair of pixels that are at "exactly the same distance and direction of the displacement vector" [6].

From this principal, it uses to computes the relationships of pixel intensity to the intensity of its neighbouring pixels which are based on hypothesis that the same gray level configuration is repeated in a texture and pixels that are close together tend to be more related than pixels that are far away from each other. The "graycomatrix" command creates the GLCM by calculating how often a pixel with gray-level (gray scale intensity) value i occurs horizontally adjacent to a pixel with the value j. Each element (i,j) in GLCM specifies the number of times that the pixel with value i occurred horizontally adjacent to a pixel with value j.

The GLCM $P_{(d,\theta)}(l_1,l_2)$ represents the probability of occurrence of the pair of gray levels (l_1,l_2) separated by a given distance d at angle θ. GLCMs are constructed by mapping the gray-level co-occurrence counts or probabilities based on the spatial relations of pixels at different angular directions while scanning the image from left-to-right and top-to-bottom. Typically, four values of θ, namely 0°, 45°, 90° and 135° cover the orientations and most common choice of distance d=1 when θ is 0° or 90° and d= √2 when θ is 45° or 135°.

For an image with number of pixels P=36, gray levels K=4, and pixel values

1	0	2	3	1	2
1	2	3	2	1	1
2	3	2	0	1	2
3	2	1	0	2	2
2	1	1	2	3	2
0	2	2	3	2	1

Let us consider, for example, pairs of pixels positioned diagonally next to each other from left to upper right where d=√2 and θ=45°. A K by K matrix H(d, θ) is formed such that each element h_{ij} is the number of times a pixel with value i and another with value j are located in the selected relative position. For example, the count h_{01} is 3 and the count h_{33} is 4. The complex matrix H (√2, 45°) for this image is

	0	**1**	**2**	**3**
0	0	3	0	0
1	3	2	1	0
2	0	2	9	0
3	0	0	1	4

3.4.1 Order Of Texture Measures

- *First order* texture measures are statistics calculated from the original image values, like variance, and do not consider pixel neighbour relationships.
- *Second order* measures consider the relationship between groups of two (usually neighbouring) pixels in the original image e.g. second moment.
- *Third and higher order* textures (considering the relationships among three or more pixels) are theoretically possible but not commonly implemented due to calculation time and interpretation difficulty.

Haralick [6] has proposed a number of useful texture features that can be computed from the co-occurrence matrix. Here μ_x and μ_y are the means and σ_x and σ_y are the standard deviations of $P_d(x)$ and $P_d(y)$ respectively,

$$P_d(x) = \Sigma_j P_d(x,j) \qquad \text{(i)}$$

$$P_d(y) = \Sigma_i P_d(i,y) \qquad \text{(ii)}$$

Where $P_d(x)$ and $P_d(y)$ are partial probability density functions.

In order to define Haralick's Texture measures, GLCM is normalised as

$$p(l_1,l_2) = \frac{P(l_1,l_2)}{\sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} P(l_1,l_2)} \qquad \text{(iii)}$$

In the derivation of Haralick's texture measures the following entities are used:

$$p_x(l_1) = \sum_{l_2=0}^{L-1} p(l_1,l_2) \qquad \text{(iv)}$$

$$p_y(l_2) = \sum_{l_1=0}^{L-1} p(l_1,l_2) \qquad \text{(v)}$$

$$p_{x+y}(k) = \sum_{l_1=0}^{L-1}\sum_{l_2=0}^{L-1} p(l_1,l_2) \qquad \text{(vi)}$$

Where k=0, 1, 2, ... , 2(L-1).

The texture measures are defined as follows:

a) The **ASM (Angular Second Moment) or Energy** is defined as

$$f_1 = \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} p^2(l_1, l_2) \qquad \text{(vii)}$$

b) The **contrast feature** is defined as

$$f_2 = \sum_{k=0}^{L-1} k^2 \sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} p(l_1, l_2) \qquad \text{(viii)}$$

c) The **Correlation measure** represents linear dependencies of gray levels, is defined as

$$f_3 = \frac{1}{\sigma_x \sigma_y} \left[\sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} l_1 l_2 p(l_1, l_2) - \mu_x \mu_y\right] \qquad \text{(ix)}$$

d) Entropy, a measure of non uniformity in the image or the complexity of the texture, is defined as

$$f_4 = -\left[\sum_{l_1=0}^{L-1} \sum_{l_2=0}^{L-1} p(l_1, l_2) \log_2(p(l_1, l_2))\right] \qquad \text{(x)}$$

e) The **sum variance** feature is defined as

$$f_5 = \sum_{k=0}^{2(L-1)} (k - f)^2 p_{x+y}(k) \qquad \text{(xi)}$$

where,

f is sum-entropy.

f) **Mean** is defined as

$$f_6 = \sum_{l=0}^{2(L-1)} l p_{x+y}(l) \qquad \text{(xii)}$$

g) **Dissimilarity** is defined as

$$f_7 = \sum_{l_1, l_2}^{L-1} p(l_1, l_2) |l_1 - l_2| \qquad \text{(xiii)}$$

h) **Homogeneity** is defined as

$$f_8 = \sum_{l_1}^{L-1} \sum_{l_2}^{L-1} \frac{p(l_1, l_2)}{1 + |l_1 - l_2|} \qquad \text{(xiv)}$$

3.4.2 Physical Significance of Texture Features

i) ASM (Angular Second Moment) or Energy is a measure of homogeneity. A homogeneous image has a small number of entries along the diagonal of the GLCM with large values which leads to large value of ENERGY.

ii) Contrast feature is a measure of contrast or amount of variation present in the image

iii) Correlation feature is a measure of gray-tone linear-dependencies in the imge.

iv) Entropy is a measure of information or randomness. Entropy is high when all values in P[i,j] are of similar magnitude.

v) Variance is the sum of differences between intensity of the central pixel and its neighbourhood.

vi) Mean measure gives the average intensity in the window of interest.

vii) Dissimilarity is a measure of the evenness.

viii) Homogeneity is a measure of uniformity. When range of gray levels is small, p[i,j] will tend to be clustered around the main diagonal.

3.5 Support Vector Machines (SVMs)

An area that has seen much recent interest is support vector machines (SVM).SVMs (also called maximum margin classifiers) try to find the optimal decision boundary by maximizing the margin between the boundaries of different classes[21]. SVMs identify those instances of each class that define the boundary of that class in the feature space. These instances, considered to be the most informative ones, are called the support vectors.

In the case of support vector machines, a data point is viewed as a p-dimensional vector, and we want to know whether we can separate such points with a $p - 1$-dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum margin classifier.

In SVMs, finite dimensional data is mapped into a higher dimensional feature space so that cross products may be computed easily in terms of the variables in the original space, making the computational load reasonable. The cross products in the larger space are defined in terms of a kernel function $K(x,y)$ selected to suit the problem. The hyperplanes in the higher dimensional space are defined as the set of points whose inner product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters α_i of images of feature vectors that occur in the data base. With this choice of a hyperplane the points x in the feature space that are mapped into the hyperplane are defined by the relation:

$$\sum_i \alpha_i K(x_i,x) = constant$$

The original optimal hyperplane algorithm was a linear classifier (Vapnik,1963) . However, in 1992, Bernhard Boser, Isabelle Guyon and Vapnik suggested a way to create nonlinear classifiers by applying the kernel to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinear kernel-function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be nonlinear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space, it may be nonlinear in the original input space.

Some common kernels include:

- Polynomial (homogeneous):
- Polynomial (inhomogeneous):
- Gaussian radial basis function
- Hyperbolic tangent

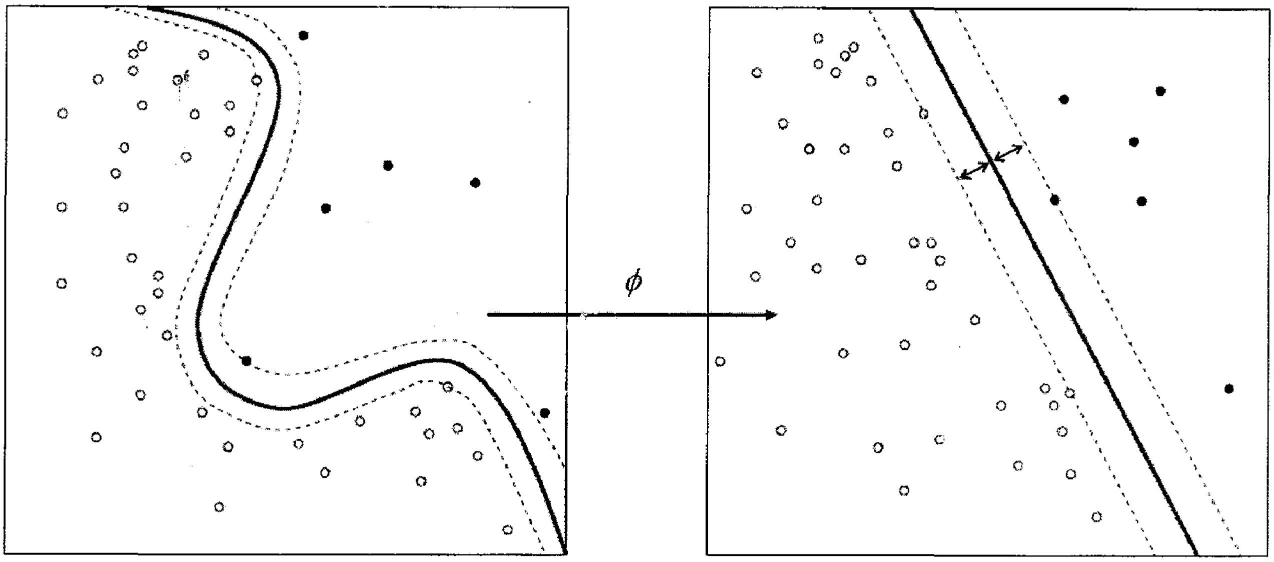


Figure 3.1 Kernels are used to compute a non-linearly separable functions into a higher dimension linearly separable function

A SVM can be applied directly only to two-class problem.

3.5.1 Multi-class SVM

There are many approaches to increase the number of classes from a primitive two-class SVM. The possible approaches are:

- One-against-all strategy classifies one class from other classes each time, and need n SVMs for a n-class problem.
- One-against-one strategy separates n classes into n(n-1)/2 different pairs, and classifies one class from a pair, thus needs n(n-1)/2 SVMs.
- Decision tree strategy uses a binary decision tree to construct the multi-class SVM, and needs (n-1) SVMs.

In our work, we have used decision tree strategy to implement the multi-class SVM.

3.6 Proposed Classification Strategy

The proposed classifier uses GLCMs and SVMs both. GLCMs are used to compute various texture features of the image, and feature vector is constructed. After this, a SVMs decision tree is trained, and applied to classify the input feature vector (input image) into prespecified number of classes. The whole schema of the clasifier can be understood by the following:

Stage 1 : Input preprocessed Image

In first stage, a raw satellite image is preprocessed for filtering noise, geocoding, and then is converted to ASCII format.

Stage 2 : Calculation of GLCMs

Now, Gray level co-occurrence matrices (GLCMs) are calculated. We have calculated GLCMs for different angles and different window-sizes, so that texture features can represent terrain more effectively.

Stage 3 : Calculation of averaged texture features

By calculating various texture feature for different angles of GLCMs, we calculate the average values for every feature, so that we have equal representation of every angle in the feature vector.

Stage 4 : Calculation of feature vector

After calculating texture features, a feature vector is constructed for the input image. This feature vectors have as many number of rows as there are pixels in image.

Stage 5 : Calculation of Training Vector

We construct a training vector from the pixels of which we know the respective class. This will be used to train the SVM decision tree. Constructing training vector is one of the most important stage, as accuracy depends on this only.

Stage 6 : Classification by SVM decision tree

Training vector and feature vector are fed to SVM decision tree (for n classes). SVM decision tree classifies each member (pixel) of feature vector in one of the prespecified classes. The accuracy of this step depends mainly on the training vector.

Step 7 : Classified image

After SVM decision tree classifies every pixel in its class, we reconstruct our image which shows pixels belonging only to the classes we specified.

The whole strategy can be shown by the following flow-chart (on next page):

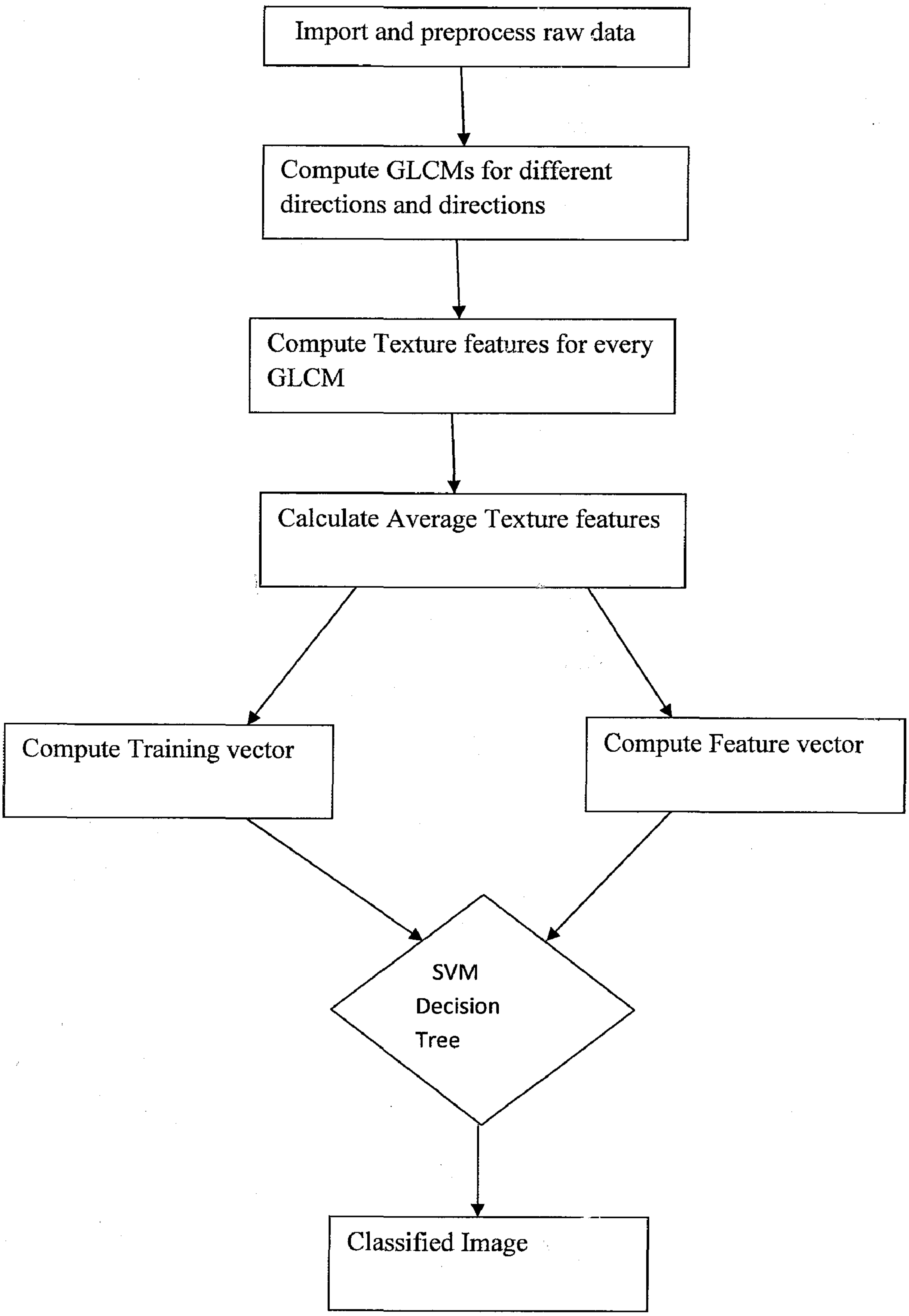


Figure 3.2 Flow-chart of proposed classifier

3.7 Some Standard Classification Methods

We will discuss some of the standard classification methods, which are frequently used in pattern recognition. Then, we will compare our proposed classifier with these standard classifiers.

We will discuss k-means classifier which is a unsupervised classification method, and Artificial Neural Networks (ANNs) which are supervised classification method.

3.7.1 k-means Classification Method :

k-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k sets ($k<n$) $S=\{S_1,S_2,\ldots,S_k\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|^2,$$

where x_n is a vector representing the n^{th} data point and μ_j is the geometric centroid of the data points in j^{th} cluster.

The k-means clustering algorithm is commonly used in image segmentation. The results of the segmentation are used to aid border detection and object recognition.

3.7.2 Artificial Neural Networks (ANNs) :

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. Among countless number of neural network structures, there are two that are used more often than all others: the multilayer perceptron (MLP) and the radial basis function (RBF).

Multilayer Perceptrorn (MLP) :

The neuron, modelled as the unit shown in Figure 3.3, is typically called a node. It features a set of inputs and a weight for each input representing the information arriving, a processing unit, and an output that connects to other nodes (or provides the output).The weights, positive or negative, represent the excitatory or inhibitory nature of neuronal activation, respectively.

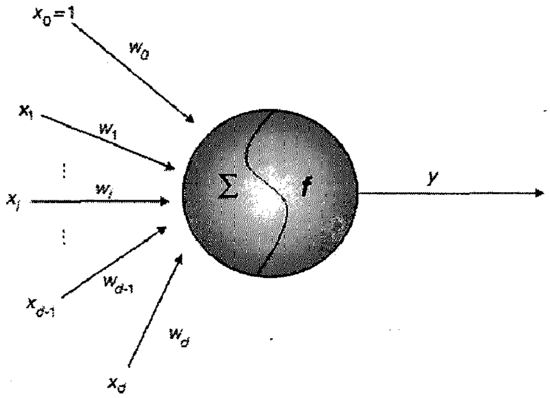


Figure 3.3 node model of a neuron

The output of the node is calculated as the weighted sum of its inputs, called the net sum, or simply net, passed through the activation function f:

$$net = \sum_{i=0}^{d} w_{ji} x_i = \mathbf{x} \cdot \mathbf{w}^T + w_0$$

$$y = f(net).$$

$$net = \sum_{i=0}^{d} w_{ji} x_i = \mathbf{x} \cdot \mathbf{w}^T + w_0$$

$$y = f(net).$$

Although a single perceptron may not be of much practical use, appropriate combinations of them can be quite powerful and approximate an arbitrarily complex nonlinear decision boundary. Arguably the most popular of classifiers constructed in such a fashion is the ubiquitous multilayer perceptron (MLP). An MLP is a feed-forward type neural network, indicating that the information flows is fully connected to every other node in the following layer through the set of weights Wji. The hidden layer nodes use a nonlinear activation function. The outputs of the hidden layer nodes are then fully connected either to the next hidden layer's nodes, or more commonly to the output layer nodes through another set of weights, Wkj (Figure 3.4).

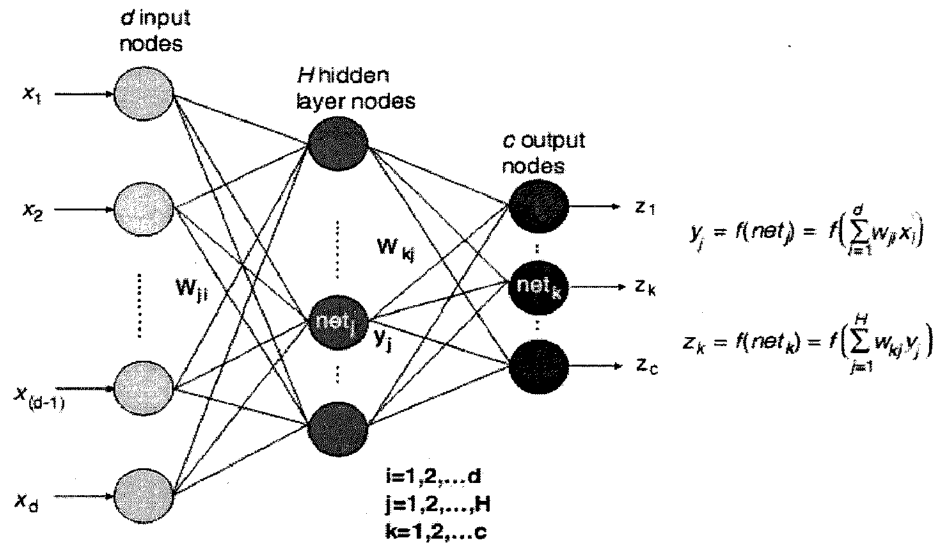


Figure 3.4 The MLP architecture

Radial Basis Function (RBF) Networks :

The classification problem can also be cast as a function approximation problem, where the goal of the classifier is to approximate the function that provides an accurate mapping of the input patterns to their correct classes. It is for such function approximation applications that the RBF networks are most commonly used.

Given a set of input/output pairs as training data drawn from an unknown (to be approximated) function, the main idea in RBF network is to place radial basis functions, also called kernel functions, on top of each training data point, and then compute the superposition of these functions to obtain an estimate of the unknown function.RBF is also a feed-forward network

3.8 Comparison of Proposed Classifier with k-means and ANNs :

We can compare our proposed classifier with k-means and ANNs by considering various parameters shown in table 3.1.

	Supervised/ Unsupervised	**Linear/ Non-linear**	**Convergence**	**Multiclass/ clusters**	**Statistical/ Non-statistical**
Proposed Classifier	semi-supervised/ Adaptive	Non-linear	converges	Multiclass	statistical
k-means Classifier	Unsupervised	Cluster based	Not guaranteed, Depends on initialization	Multi-cluster	Non-statistical
Artificial Neural Networks	Supervised/ Adaptive	Non-linear	Not guaranteed, Depends on cost function and model	multiclass	statistical

Table 3.1 Comparison of Classifiers

CHAPTER 4
IMPLEMENTATION DETAILS

4.1 Implementation

The implementation of the proposed method is done using Matlab. The expanse of the language and its useful features like image processing tools, availability of a number of predefined functions, data structures, and easy to understand nature etc are some of the main reasons for choosing Matlab as the development tool. Its modularity and object oriented nature gives the freedom to individually build and test the modules. Some of the preprocessing has been done using Envi-v4.7, which is a very popular tool in the field of remote sensing, as it can handle a variety of data, and provides a number of features like interpreting raw satellite data, filtering, and geocoding etc. which are very essential for preprocessing.

4.2 Design and Development of the Whole Classification Strategy

Whole classification process can be divided into the following stages:

4.2.1 Raw Satellite Data Acquisition

In our work, we have used Synthetic Aperture Radar (SAR) data. Information obtainable from SAR includes backscattering coefficients, which exemplifies the strength of microwave irradiation emitted from the antenna and returned after scattering on the surface of targeted substances. Analysis of the backscattering coefficients enables to estimate, volume of water contained in soil, volume of biomass in forests, conditions of waves in the ocean and so on.

PALSAR has different observation modes. Here, in this work, data is from polarimetric mode, which has following characteristics

i)Polarization : 4 polarization (HH,HV,VH,VH)

ii) Format : CEOS/VEXCEL Standard SLC (Single Look Complex)

iii) Data Type: IEEE 32 bit floating point

iv) Data size of one pixel (sample) : Real and imaginary parts of complex number 8 bytes in total for single polarization 32 bytes in total for 4 polarization.

Data Characteristics :

Advanced Land Observatory Satellite (ALOS) PALSAR polarimetric data taken on date 6^{th} April 2009 was used in the study. The data has four different modes: HH, HV, VH and VV polarization. The ALOS PALSAR product is level 1.1 data in VEXCEL format, which is single look complex data on slant range. The product has single number of looks on range and azimuth. The default off nadir angle for polarimetric acquisition mode is 21.5^{o}. The product has resolution on ground of 30m (range) x 20 m (azimuth) [25].

ii) Ambala Region

This study area has centre latitude 30.4629^{o} and longitude 77.0000^{o}. It is shown on map in figure 4.2. It covers Ambala, jagadhari, Panchkula regions.

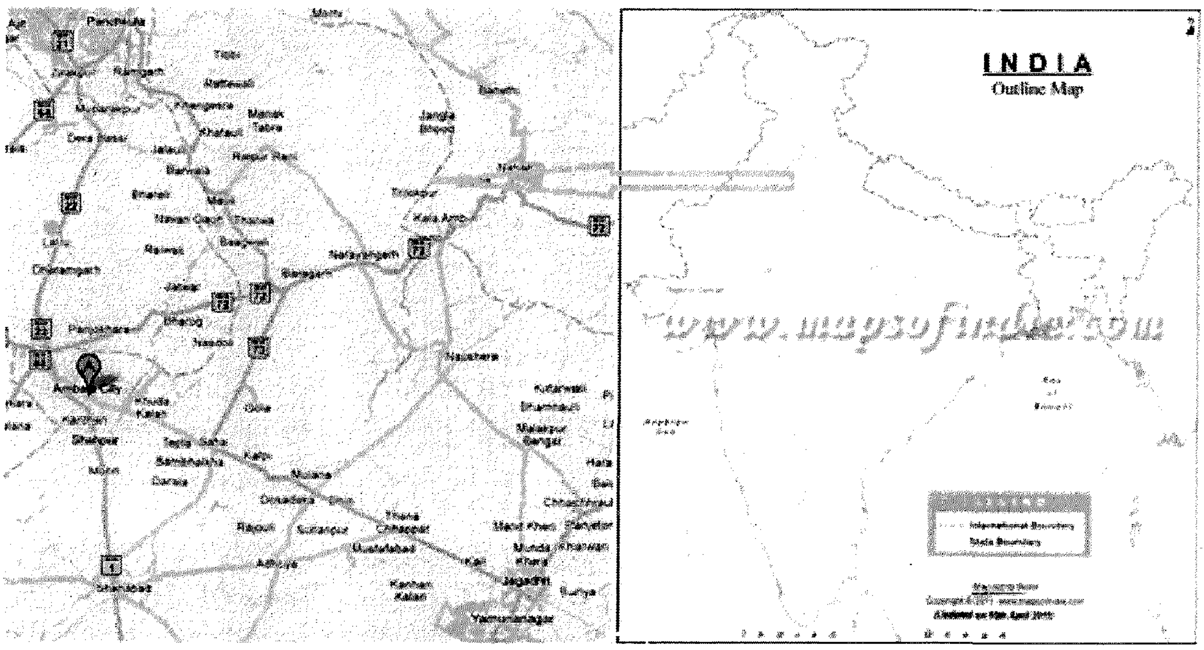


Figure 4.2 Study Area 2 - Ambala region

Data Charecteristics

CEOS Palsar data was taken on Nov. 13^{th},2010. This dats is polarimetric i.e. has all the four modes HH, HV, VH, VV. The CEOS PALSAR product is level 1.1 data in CEOS format, which is single look complex data on slant range. The product has single number of looks on range and azimuth. The default off nadir angle for polarimetric acquisition mode is 23.1^{o}. The product has resolution on ground of 30m (range) x 20 m (azimuth)

4.2.2 Data Preprocessing

Data preprocessing has been done using ENVI (v. 4.7). Various steps involved in preprocessing are shown in the following flow-chart :

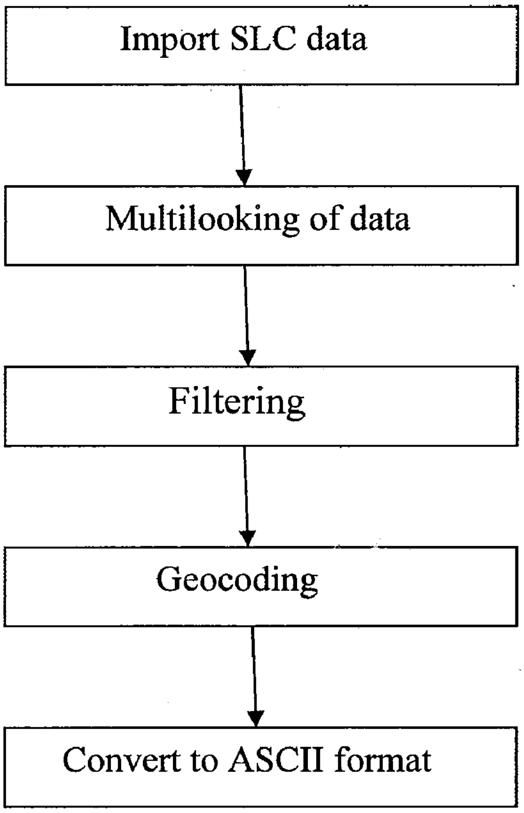


Figure 4.3 : Flow-chart of data preprocessing

This ASCII formatted data is used for next steps.

4.2.3 Calculating Gray Level Co-occurrence Matrices (GLCMs)

The gray level co occurrence matrix of the given images is calculated using MATLAB image processing toolbox. The matrix is calculated for every image and in all the 4 directions that is, 0, 45, 90 and 135 degree.

The following command creates a gray-level co-occurrence matrix (GLCM) from image I.

glcm = graycomatrix(I).

glcm = graycomatrix(I, param1, val1, param2, val2,...) returns one or more graylevel co-occurrence matrices, depending on the values of the optional parameter/value pairs.

Parameter names can be abbreviated, and case does not matter. The various parameters are described in the Table 4.1 .

GLCM of images is calculated using the following syntax:

GLCM = graycomatrix(I,'Offset',[0 1]);

The offset value is used to determine the distance and direction of pixel neighbour. [0 1] is the default value, that is, 0° direction in which the relation between the gray level value of pixel of interest is found out relative to its next right neighbour. We have used four different directions to calculate the values of GLCM, that is, 0°, 45°, 90° and 135°.

Parameter	Description
'GrayLimits'	Two-element vector, [low high], that specifies how the gray scale values in I are linearly scaled into gray levels. Gray scale values less than or equal to low are scaled to 1. Gray scale values greater than or equal to high be scaled to Num Levels. If gray limits is set to [], gray co-matrix uses the minimum and maximum gray scale values in the image as limits, [min (I (:)) max (I (:))].
'NumLevels'	Integer specifying the number of gray-levels to use when scaling the gray scale values in I. For example, if Num Levels is 8, gray comatrix scales the values in I so they are integers between 1 and 8. The number of gray-levels determines the size of the gray-level cooccurrence matrix (GLCM).
'Offset'	P-by-2 array of integers specifying the distance between the pixel of interest and its neighbour. Each row in the array is a two-element vector, [row_offset, col_offset], that specifies the relationship, or offset, of a pair of pixels. row_offset is the number of rows between the pixel-of-interest and its neighbor. col_offset is the number of columns between the pixel-of-interest and its neighbor. Because the offset is often expressed as an angle, the following table lists the offset values that specify common angles, given the pixel distance D. Angle Offset 0 [0 D] 45 [-D D] 90 [-D 0] 135 [-D -D] D is 1 in our study.

Table 4.1. GLCM Parameters

4.2.4 Computation of Texture Features

To compute texture features, first GLCM are normalized so that the sum of its elements is equal to 1. Each element (r,c) in the normalized GLCM is the joint probability occurrence of pixel pairs with a defined spatial relationship having gray level values r and c in the image. Normalized GLCM is used to calculate various texture feature as defined in section 3.4 of chapter 3 of this report. Following features are computed:

- Angular Second Moment
- Contrast
- Correlation
- Variance
- Entropy
- Mean
- Dissimilarity
- Homogeneity

We implemented these feature in MATLAB. We calculated these features in all four directions i.e. 0, 45, 90, and 135 degrees.

After calculating these features in four directions, mean values are calculated as follows

$$f = (f1 + f2 + f3 + f4)/4$$

where 'f' is mean feature value, and f1,f2,f3,f4 are feature values from 4 directions.

4.2.5 Computation of feature Vector

From features calculated, we constructed a feature vector. A feature is a 'NxF' vector, where 'N' is the number of pixels in the image, and 'F' is the number of features. For example, if a image is of '400x400' size, and number of features is 8, feature vector will be of dimensions '160000 x 8'.

This feature vector along with training vector will be fed to SVM decision tree (as described in next sections) for the classification.

4.2.6 Training Vector :

A training vector is constructed with the help of the points we know the classes of. This training vector consists of the same number of columns (features) as are in the feature vector of the target image. This vector is used to train the SVM.

We have used Urban, Water, Vegetation, Bareland etc. as classes, and constructed training vector accordingly.

In many cases where terrains are similar, classification can be done by training the proposed classifier once with some known data, this feature makes our proposed method a adaptive one.

4.2.7 Classification by SVM Decision Tree

Final classification is done using SVMs decision tree. SVM is trained with training vector, and after that feature vector is fed to trained SVM, which classifies it. We have used Matlab's SVM tool for this purpose, and improved it to meet our requirement of a decision tree. We can put the whole process in the following stages:

- **Training of SVM**
  The syntex for training a SVM is

SVMStruct = svmtrain(Training,Group, 'Kernel_Function', Kernel_FunctionValue)

Return values and various parameters (as described in Matlab documentation) are given in the following table-

SVMStruct	Structure containing information about the trained SVM classifier, including the following fields: ▪ KernelFunction ▪ KernelFunctionArgs ▪ GroupNames etc.
Training	Matrix of training data, where each row corresponds to an observation or replicate, and each column corresponds to a feature or variable.
Group	Column vector, character array, or cell array of strings for classifying data in Training into two groups. It has the same number of elements as there are rows in Training. Each element specifies the group to which the corresponding row in Training belongs.
Kernel_FunctionValue	String or function handle specifying the kernel function that maps the training data into kernel space. e.g. Linear, Quadratic, Gaussian radial basis function, polynomial etc.

Table 4.2 svmsstruct parameters and return values

- **Classification by SVM**

For two class problems, we use the following SVM method provided by Matlab:

Group = svmclassify(SVMStruct, Sample)

Description (as given in Matlab documentation):

Group = svmclassify(SVMStruct, Sample) classifies each row of the data in Sample using the information in a support vector machine classifier structure SVMStruct, created using the svmtrain function. Sample must have the same number of columns as the data used to train the classifier in svmtrain. Group indicates the group to which each row of Sample has been assigned.

This svmclassify works for two class problem only.

- **Multi-class SVM**

For more than two classes problem, we have several strategies, which uses primitive binary-class SVMs for multi-class classification purpose. Some of them are listed below

a) One-against-all strategy classifies one class from other classes each time, and need n SVMs for a n-class problem.

b) One-against-one strategy separates n classes into n(n-1)/2 different pairs, and classifies one class from a pair, thus needs n(n-1)/2 SVMs.

c) Decision tree strategy uses a binary decision tree to construct the multi-class SVM, and needs (n-1) SVMs.

We have used 'Decision tree strategy' as it needs least number of SVMs of all strategies.

SVMs Binary Decision Tree :

In this strategy, at each node of binary decision tree a SVM is implemented. That node is evaluated against a given a class, if the input feature vector corresponds to this class, tree is terminated at that node, and that feature vector is assigned that class, else decision moves to the next node, and same process is repeated until all classes have been checked.

We can show this whole process by the following figure :

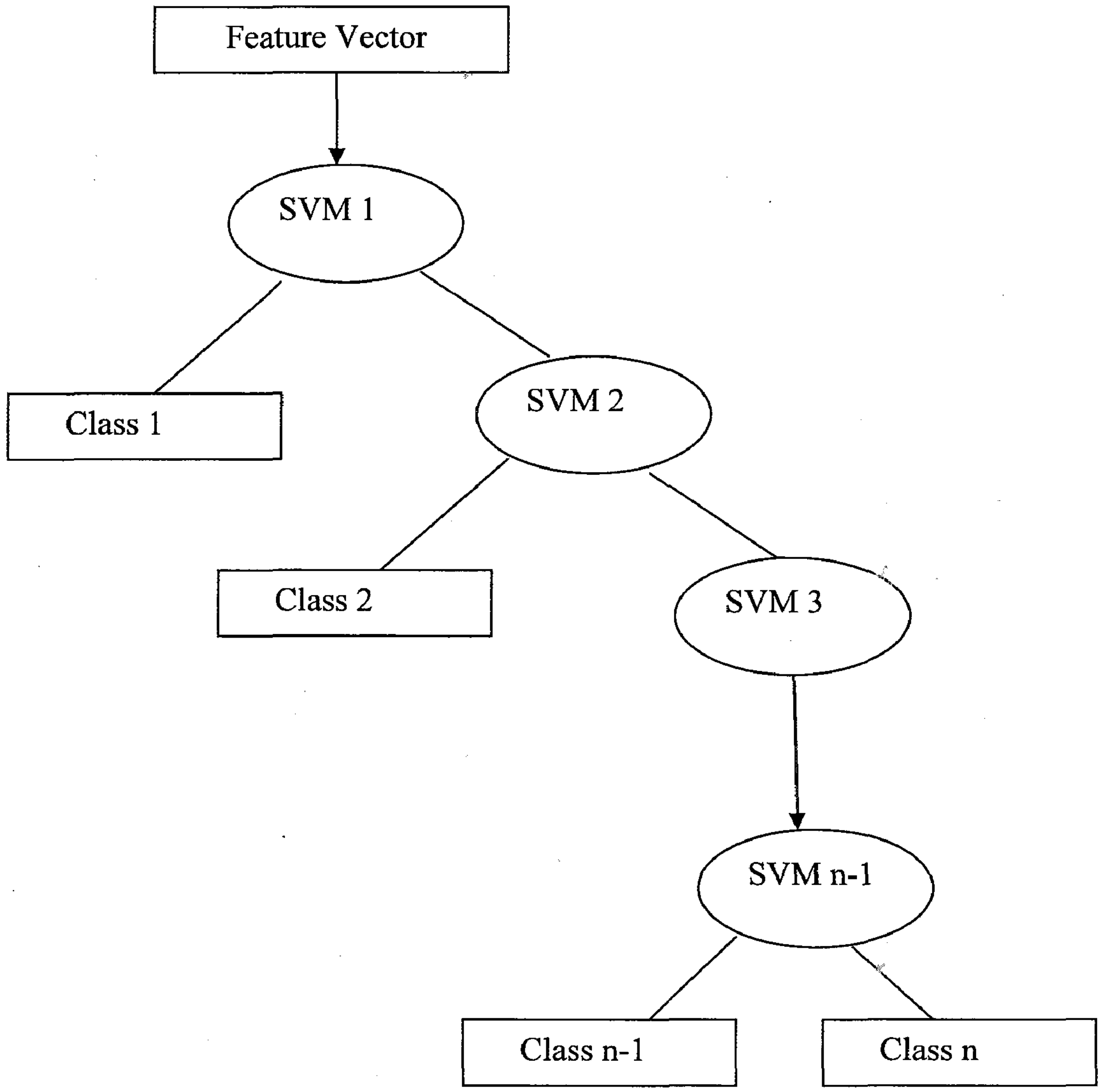


Figure 4.4 : SVMs Decision Tree for 'n' Classes.

In our work we have classified earth's terrain in three to four classes *viz.* urban, vegetation, water, big trees, and bareland, depending upon the availability of classes in terrain.

Implementation of this classifier on real satellite imagery and results will be shown and discussed in the next chapter.

4.3 Pseudo Code

The whole classification strategy can be shown with the following pseudo-code :

```
// Input raw satellite data and preprocessing

Import (raw_data);

Multilook (Imported_data);

Filter(Multilooked_data);

Geocode (Filtered_data);

Export_Ascii (Geocoded_data);

// Computation of texture features using GLCMs in four directions

I  = Ascii_data;

Glcm_0 = graycomatrix(I,'offset',[0,1]);

Glcm_45 = graycomatrix(I,'offset',[-1,1]);

Glcm_90 = graycomatrix(I,'offset',[-1,0]);

Glcm_135 = graycomatrix(I,'offset',[-1,-1]);

Normalize(GLCMs);

Calculate(GLCM, Texture_feature);

//Compute average features

Avg_feature  = (feature_0 +  feature_45 + feature_90 + feature_135) / 4;

// calculate Feature Vector

Feature_vector (Average_features);

//Compute Training vector

Training_vector(features,known_class_points);

// Train SVM

SVMStruct = svmtrain(Training_vector,Known_classes, 'Kernel_Function', rbs);
```

```
//Classification into n classes using SVM decision tree
For  j = 1:no_pixels
For i = 1:no_classes
Group = svmclassify(SVMStruct[i], Feature_vector[j]);
If (Group == Class[i])
Assign(Feature_vector[j], Class[i]) ;
End
End
End
Output_classified_image;
```

4.4 Implementation of k-means and ANN

For comparison purpose, we implemented one non-supervised classifier (k-means), and one supervised classifier (ANN). Implementation details of the two is given in following sections:

4.4.1 k-means

We implemented k-means classifier using ENVI (v 4.7) with following parameters :

a) Roorkee region

Number of Classes : 3

Change Threshold : 10%

Maximum Iterations : 100

b) Ambala region

Number of classes : 4

Change Threshold : 5%

Maximum Iterations : 200

4.4.2 Artificial Neural Network

We implemented ANN using MATLAB's neuron-pattern-recognition tool (nprtool). We used following parameters :

a) Roorkee Region

Training set : training set consists of three classes; 50 pixels from water, 50 pixels from urban, and 50 pixels from vegetation.

Number of hidden neurons is set to 20.

Training is done using Levenberg-Marquardt backpropagation.

b) Ambala region

Training set : training set consists of four classes; 50 pixels from water, 50 pixels from urban, 50 pixels from short vegetation, and 50 pixels from tall-vegetations.

Number of hidden neurons is set to 20.

Training is done using Levenberg-Marquardt backpropagation.

We will compare results from these two classifiers with our proposed classifier in next chapter.

CHAPTER 5
RESULTS AND DISCUSSION

In the following sections, we will discuss performance of proposed classifier across various parameters. The proposed method has been tested on real satellite data.

5.1 Introduction

We applied proposed classifier on Synthetic Aperture Radar (SAR) data. Texture features chosen have been described in section 4.2.4 of chapter 4. Training sets were chosen from sub-parts of the images. Number of classes were chosen depending upon their availability on earth's terrain. Whole process can be understood step by step from the following :

- SAR images were chosen to test proposed classifier
- Texture features were selected (as described in sec 4.2.4)
- GLCMs were calculated in all 4 directions
- Average texture features were calculated
- Sub parts of images were used to terrain classifier
- Accuracy is calculated of the proposed classifier

Main performance metric is, in the case of classifiers, is their accuracy, so we have used the same parameter to evaluate the performance of our proposed classifier with respect to various parameters.

5.2 Results

Results obtained after applying the proposed classifier have been discussed in the following sections.

5.2.1 Actual Images of Test Areas :

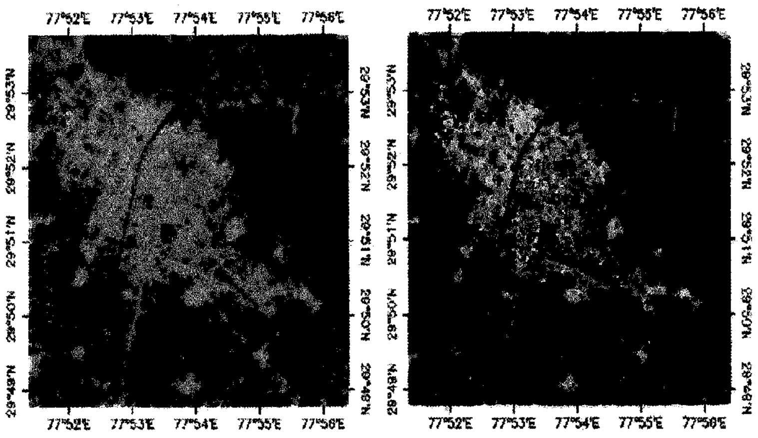
Actual images of the two test areas (described in chapter 4) as obtained by PALSAR (Radar), and Google Earth (Optical) are shown in figures (5.1) – (5.5).

a) **Roorkee Area**

- Toposheet of Roorkee area

Figure 5.1 Toposheet of Roorkee Area

- Images observed by PALSAR :



(a) (b)

Figure 5.2 PALSAR Image of Roorkee region (a) HV- Polarisation (b) Color composite image (hh = red, hv = green, vv= blue)

- Image from Google Earth

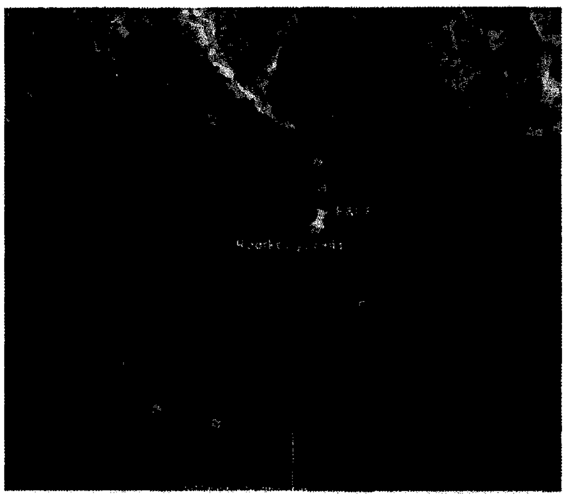
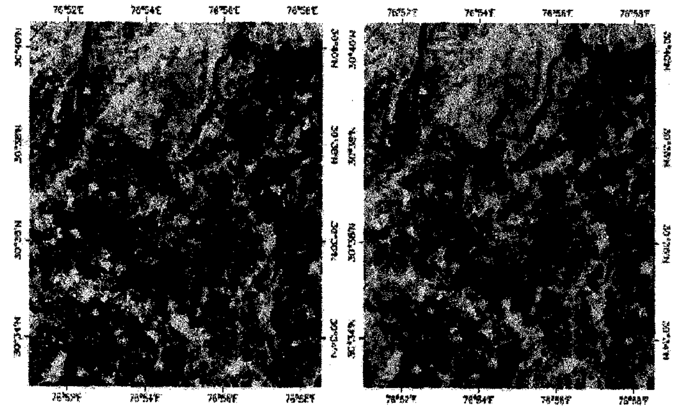


Figure 5.3 Google Earth image of Roorkee (date : March 28th, 2011)

b) Ambala Region

PALSAR images of Ambala region :



(a) (b)

Figure 5.4 PALSAR Image of Ambala region (a) HV- Polarisation (b) Color composite image (hh = red, hv = green, vv= blue)

- Google Earth image of Ambala image

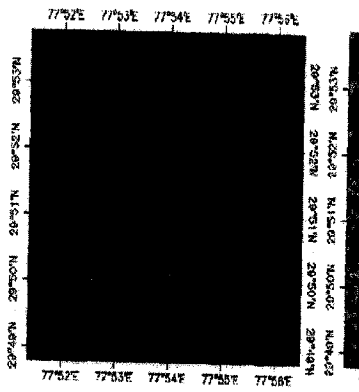
Figure 5.5 Google Earth image of Ambala region (date : March 3[th], 2010)

5.2.2 Texture features from GLCMs :

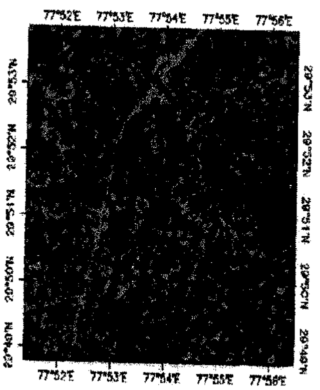
We calculated following texture features from gray level co-occurance matrices (described in chapter 3) :

- Angular Second Moment
- Contrast
- Variance
- Entropy
- Mean
- Dissimilarity
- Homogeneity

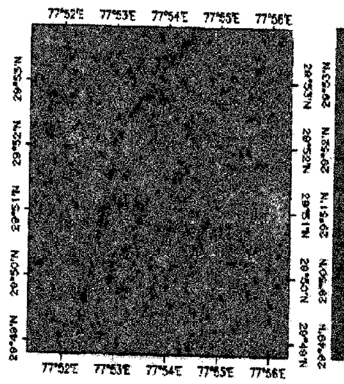
a) Roorkee Region



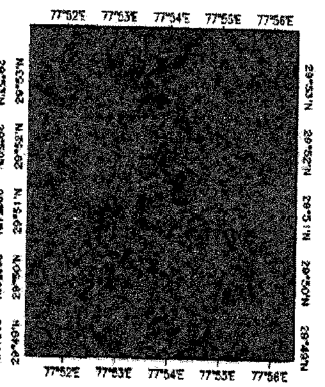
(a)



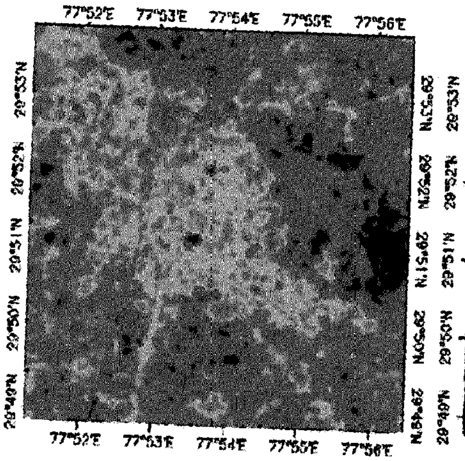
(b)



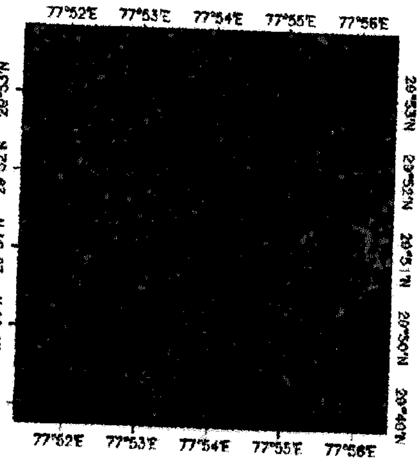
(c)



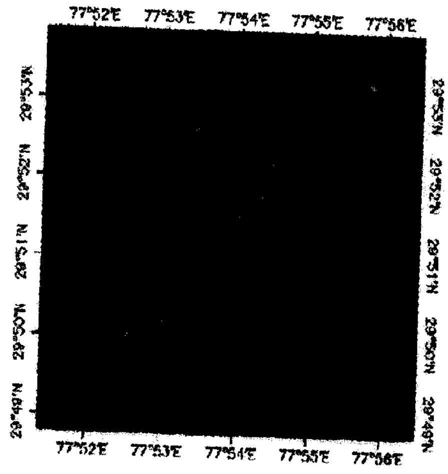
(d)



(e)



(f)



(g)

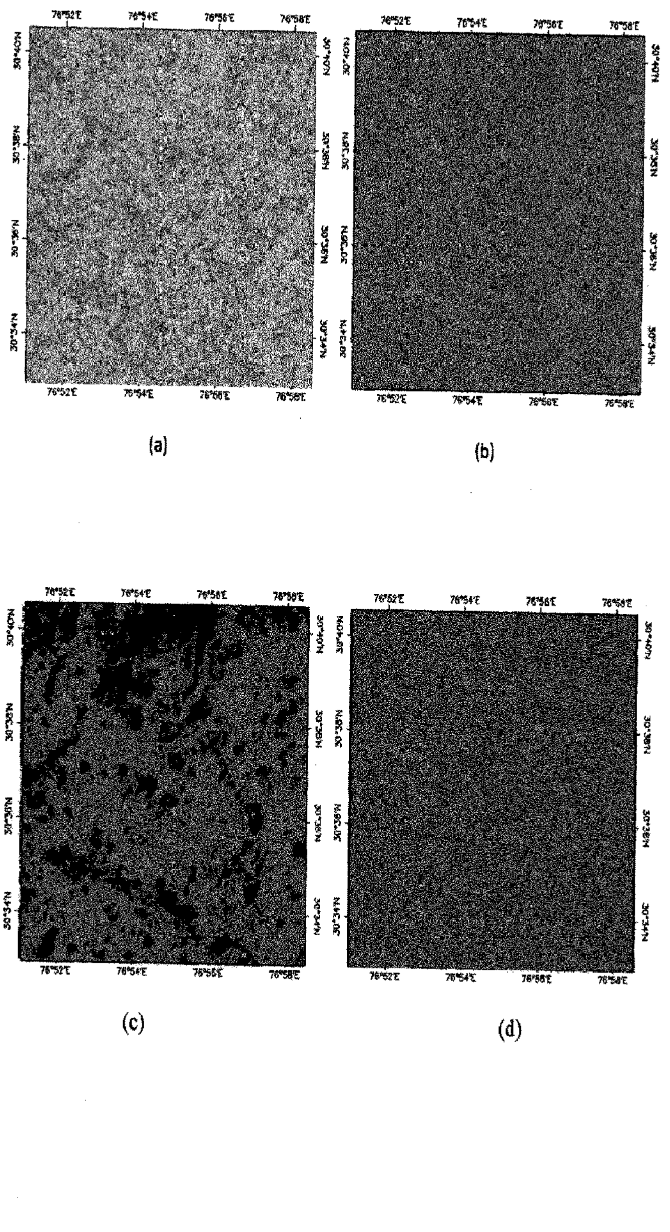
Figure 5.6 Texture features of Roorkee region (a) contrast (b) dissimilarity (c) homogeneity (d) entropy

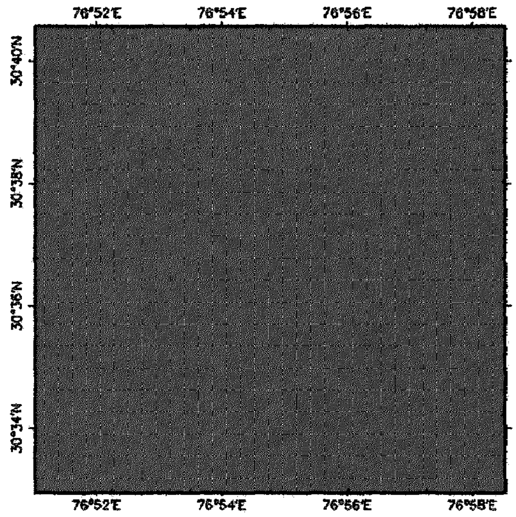
(e) mean (f) angular second moment (g) variance

Texture features were calculated from gray level co-occurrence matrices in four directions i.e. 0, 45, 90, 135 degrees with window size 3. Final features were calculated by taking mean of features from the four GLCMs. Average features calculated for Roorkee region are shown in figure 5.6.

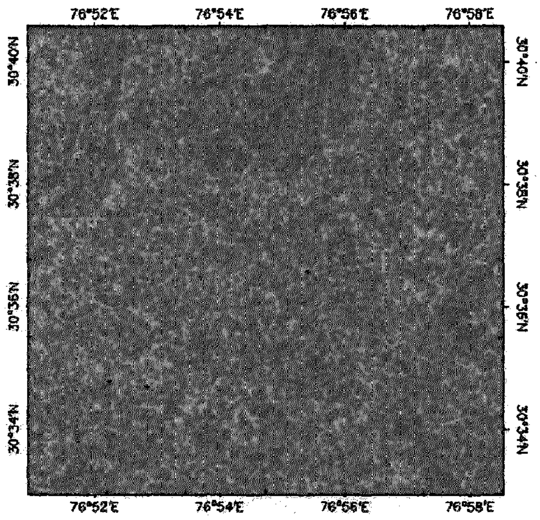
(b) Ambala region

Mean texture features were calculated from gray level co-occurance matrices in four directions i.e. 0, 45, 90, 135 degrees with window size 3, and are shown in figure 5.7.

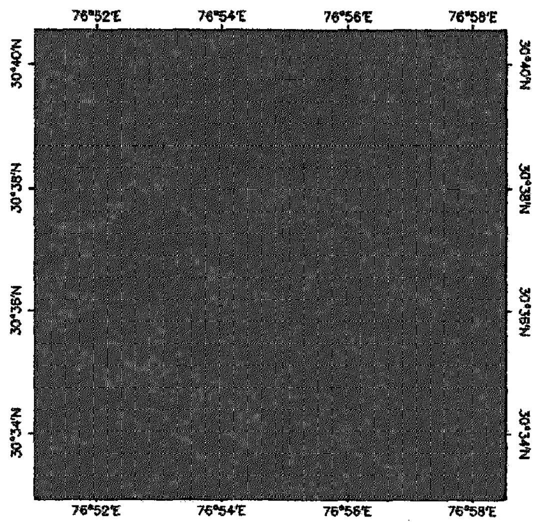




(e)



(f)



(g)

Figure 5.7 Texture features of Ambala region (a) dissimilarity (b) homogeneity (c) mean (d) entropy

(e) angular second moment (f) variance (g) contrast

We will form feature-vectors from these features, and with training-vector, will feed to support vector machine decision tree for final classification.

5.2.3 Classification by SVM Decision Tree

SVM decision tree (described in chapter 4) is used for final classification. Results from this decision tree have been discussed in next sections.

(a) Roorkee Region :

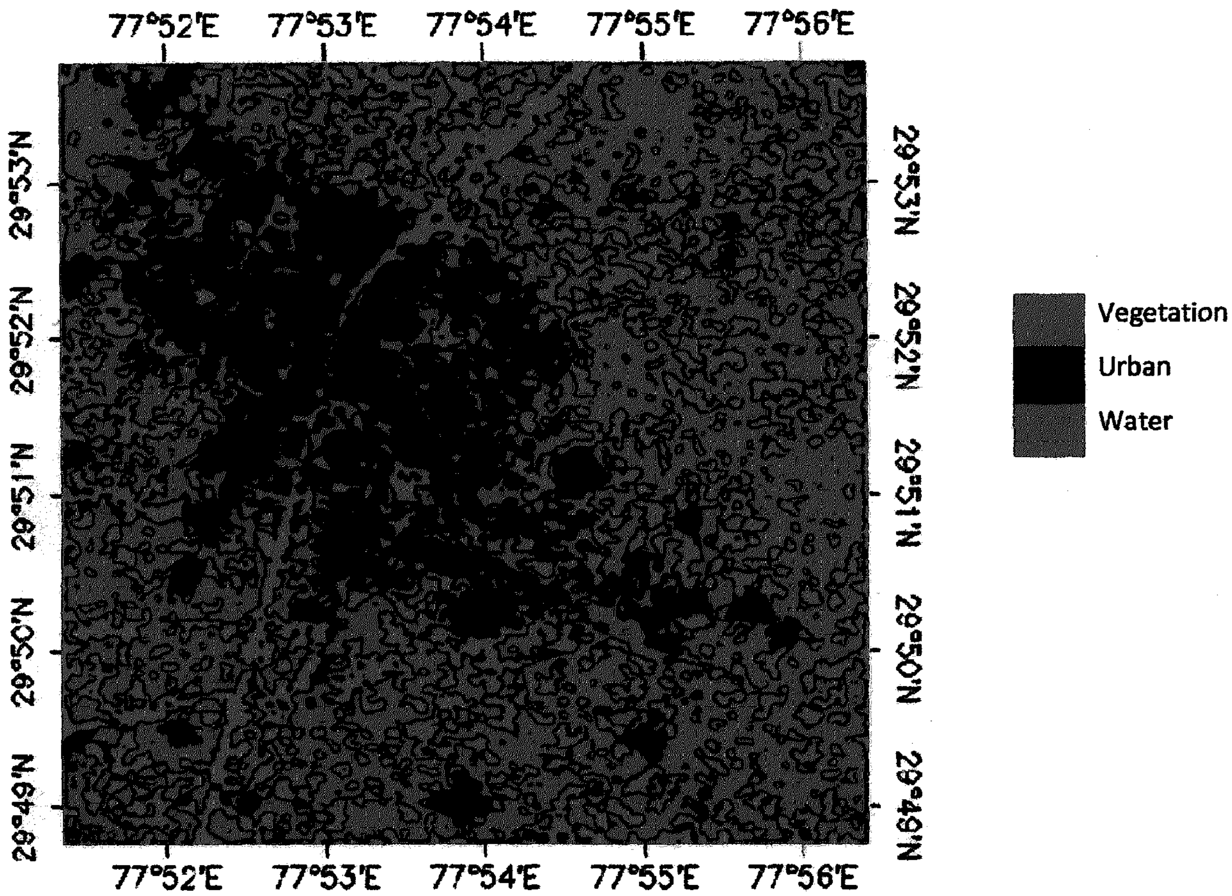


Figure 5.8 Classified Image of Rookee region with three classes.

We have classified region of Roorkee for three classes *viz.* urban, vegetation, and water, with the help of the proposed classifier. Classified image is shown in figure 5.8. We trained SVM decision tree with the points we knew class of. Decision tree worked pixel-by-pixel by checking feature-vector of every pixel and assigning it a class. As we can observe visually, image is classified satisfactorily, with most of the pixels belonging to the specific field are classified together.

Accuracy of the classifier is calculated using confusion matrix (error matrix), which compares the classification result with ground truth information. Regions of interest for the calculation of accuracy are shown by circles in figure 5.9, and confusion matrix is given in table 5.1.

Accuracy measure (Confusion matrix) :

We selected ensquared pixels in the figure 5.9, for the calculation of accuracy of the proposed classifier.

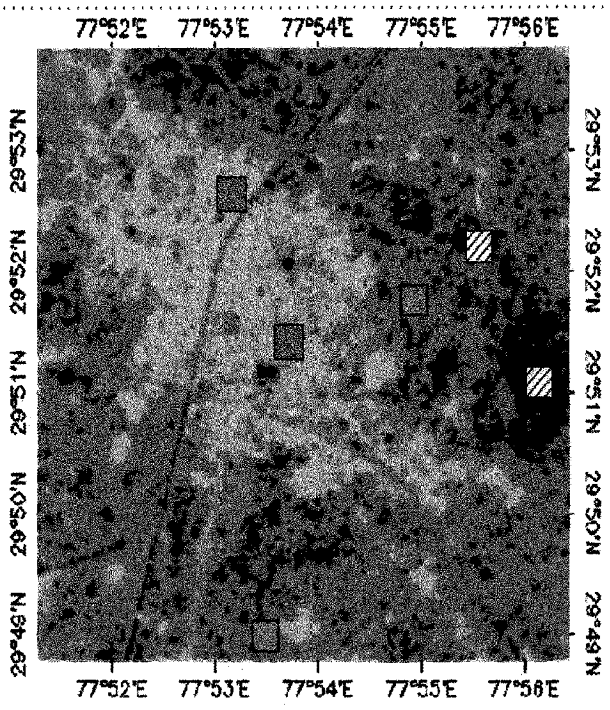
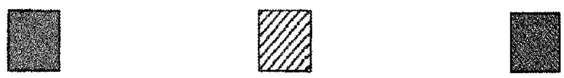


Figure 5.9 Test pixels to calculate accuracy for Roorkee region



(Vegetation) (Water) (Urban)

Confusion matrix is shown in table 5.1 on next page.

Table 5.1 Confusion Matrix for Roorkee region

Class	Urban	Vegetation	Water	Total
Urban	470	22	8	500
Vegetation	13	455	32	500
Water	2	24	474	500

From table 5.1,

False-negative (urban) = (30/500)x100 % = 6%

False-positive (Urban) = (15/500)x100 % = 3%

False-negative (vegetation) = (45/500)x100 % = 9%

False-positive (vegetation) = (46/500)x100 % = 9.2%

False-negative (water) = (26/500)x100 % = 5.2%

False-positive (water) = (30/500)x100 % = 6%

Overall accuracy = (pixels classified correctly/ total number of pixels) x 100 %

= (470+455+474)/1500 x 100 %

= (1399/1500)x100 %

= 93.26%

(b) Ambala Region

We have classified region of Ambala for four classes *viz.* urban, short vegetation, bare land, and tall vegetation with the help of the proposed classifier. Classified image is shown in figure 5.10. We trained SVM decision tree with the points we knew class of. Decision tree worked pixel-by-pixel by checking feature-vector of every pixel and assigning it a class. As we can observe visually, image is classified satisfactorily, with most of the pixels belonging to the specific field are classified together.

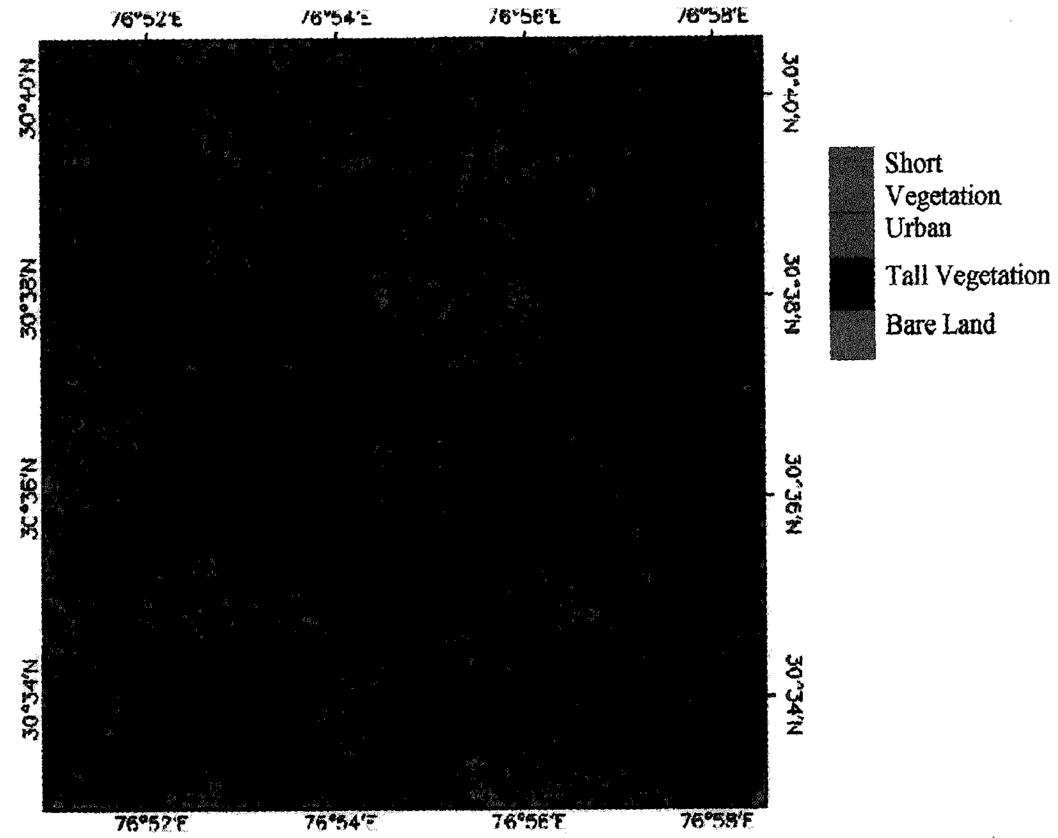


Figure 5.10 Classified Image of Ambala region with four classes.

Accuracy measure (Confusion matrix) for Ambala region is shown in table 5.2.

Table 5.2 Confusion Matrix for Ambala region

Class	Urban	Short vegetation	Long vegetation	Bare land	Total
Urban	256	6	15	23	300
Short vegetation	9	240	30	21	300
Long vegetation	5	17	122	6	150
Bare land	11	5	2	82	100

Overall accuracy = (256+240+122+82)/850 x100 %

= 700/850 x 100 %

= 82.35%

5.3 Other Classifiers

We will show results for one unsupervised classifier (k-means), and one supervised classifier (ANN), and will compare results from these two classifier with our proposed classifier.

5.3.1 k-means

Results from k-means classifier for Roorkee and Ambala regions are as follows:

a) Roorkee region

We classified Roorkee region in three classes. Classified image is shown in figure 5.11.

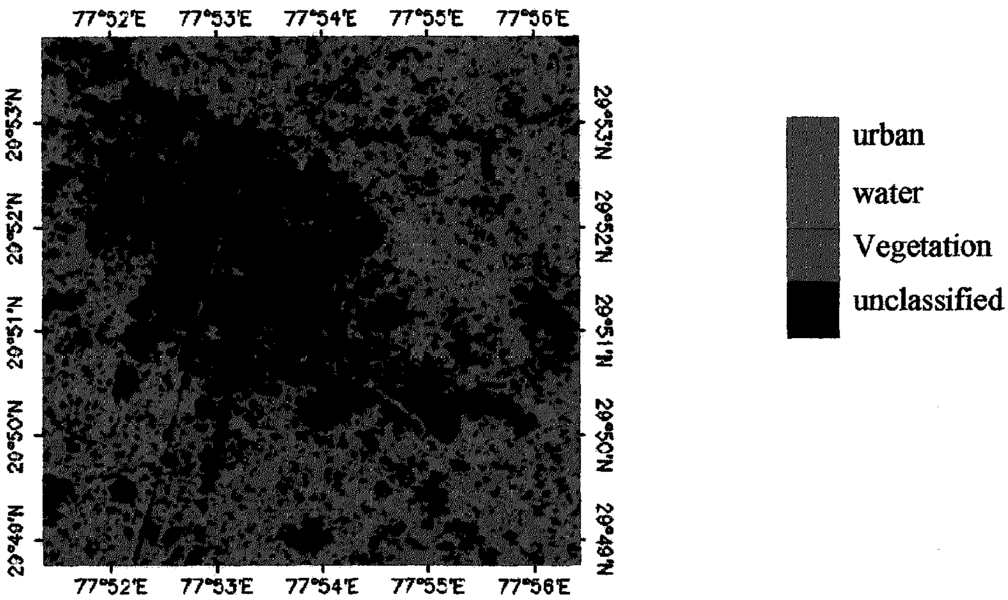


Figure 5.11 k-means classification of Roorkee region

Table 5.3 Confusion Matrix for Roorkee region from k-means

Class	urban	vegetation	water	unclassified	Total
Urban	392	28	16	64	500
vegetation	23	405	30	42	500
water	36	56	357	48	500

Overall accuracy = (1154/1500)x100% = 77%

a) Ambala region

We classified Ambala region in four classes. Classified image is shown in figure 5.12

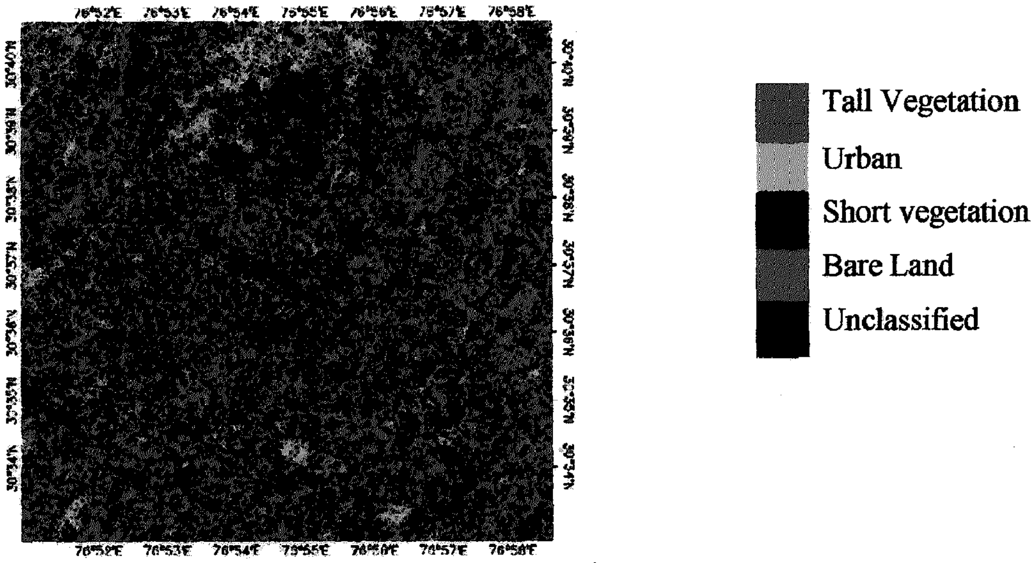


Figure 5.12 k-means classification of Ambala region

Table 5.4 Confusion Matrix for Ambala region from k-means

Class	Urban	Short vegetation	Long vegetation	Bare land	Unclassified	Total
Urban	201	16	25	53	5	300
Short vegetation	27	183	68	21	13	300
Long vegetation	11	29	91	16	3	150
Bare land	22	11	10	57	0	100

Overall accuracy = (532/850)x100% = 63%

5.3.2 Artificial Neural Network

a) Roorkee region

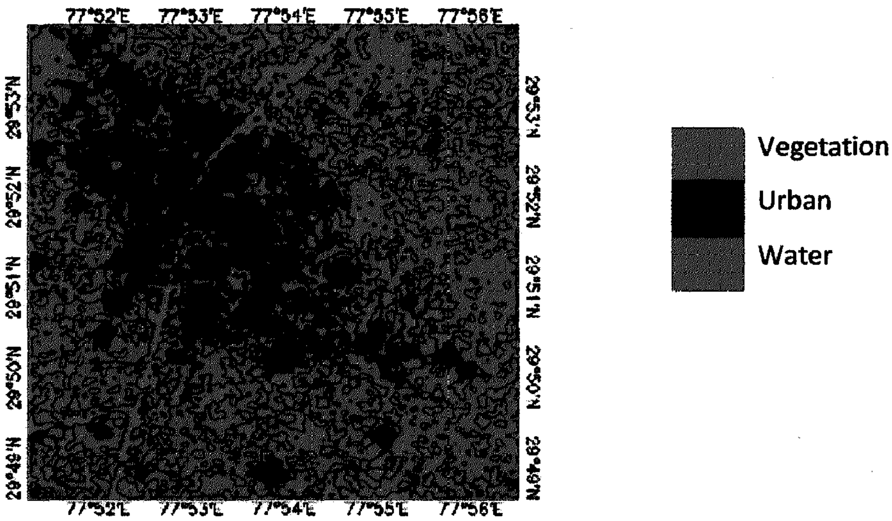


Figure 5.13 ANN classified image of Roorkee region

Table 5.5 Confusion Matrix for Roorkee region from ANN

Class	Urban	Vegetation	Water	Total
Urban	453	28	19	500
Vegetation	17	448	35	500
Water	8	31	461	500

Overall accuracy = (1362/1500)x100% = 90.80%

b) Ambala region

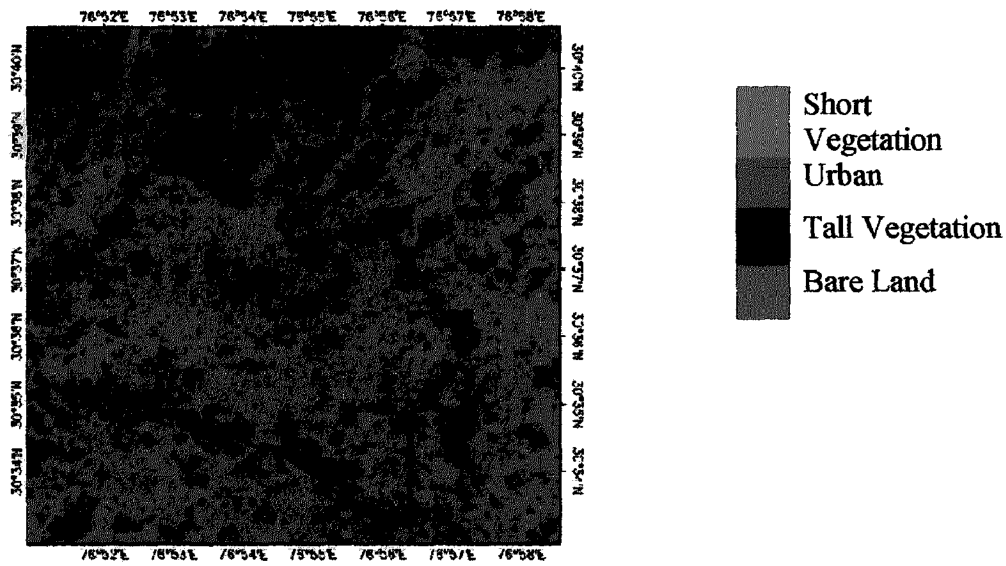


Figure 5.14 ANN classified image of Ambala region

Table 5.6 Confusion Matrix for Ambala region from ANN

Class	Urban	Short vegetation	Long vegetation	Bare land	Total
Urban	251	9	18	24	300
Short vegetation	12	232	34	22	300
Long vegetation	7	15	120	8	150
Bare land	12	4	3	81	100

Overall accuracy = (684/850)x100% = 80.4%

5.4 Accuracy Comparison

We can compare performance of our proposed classifier with k-means and ANN from overall accuracies calculated in previous sections. Results are given table 5.6.

Table 5.7 Performance Comparison

	Roorkee	Ambala	Total
Proposed Classifier	93.26%	82.35%	89.31%
k-means	77%	63%	71.94%
ANN	90.80%	80.4%	87%

Results show that proposed method classifies satellite images with good accuracy, nad performs better than k-means and ANN. Accuracy in second case (Ambala region) is lower than first case (Roorkee) region. Reasons for this are

i) increased number of classes (more mixed and overlapping pixels)

ii) some classes shows somewhat similar behavior e.g. bare land and urban, short and tall vegetation. We can verify this from Confusion Matrix of Ambala region.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this work, we proposed a method for the classification of satellite images to assign pixels in their respective classes. We used texture features of the pixels calculated from gray level co-occurrence matrices (GLCMs). We calculated texture features in different direction, and then took mean values of them. We used support vector machines (SVMs) for the classification purpose due to their non-linear and non-probabilistic nature. As pixels on earth's terrain are not well separated, so they form large number of clusters, and in most of the scenarios mixed pixels are there i.e. there is ambiguity about the class of a pixel. In these complex scenarios, traditional classifiers based on spectral features do not work properly. Results show that using texture features over spectral ones is better and more stable. As a support vector machine can classify data into two classes only, we used a support vector machine decision tree for multi-class classification. Results show that support vector machine is a excellent non-linear classifier, when trained well.

We applied proposed method on synthetic aperture radar (SAR) data, which is one of the most widely used satellite imagery system. We calculated energy, entropy, anisotropy, contrast, correlation, mean, dissimilarity, and homogeneity from gray level co-occurrence matrix in four directions, and used average values for better accommodation of directional relationships of pixels. Training vector was calculated using known pixels. Results were evaluated for two test sites with urban, vegetation, water, bare lands, and big trees. Evaluation of results shows that this method classifies earth's terrain satisfactorily with high accuracy. Results also indicate an improvement from traditional methods, which classify earth's terrain poorly. This work opens the area of research in using gray level co-occurrence matrix and support vector machines for multi-class classification of earth terrain.

Suggestions for Future Work

The proposed work classifies earth's terrain using GLCMs and SVMs. We suggest following things for future work :

We applied this classifier on synthetic aperture radar (SAR). Performance of this classifier can be tested on other types of satellite data.

There are many texture features which can be defined from GLCMs. Results can be evaluated for different sets of texture feature, and optimum number of features can be set. Dimension reduction can also be applied using *Principal Component Analysis* (PCA).

Performance of support vector machines can be evaluated for different kernel functions. Reducing number of SVMs for a fixed number of classes problems is also an exciting field.

REFERENCES

[1] NASA (1986), "Report of the EOS data panel, Earth Observing System, Data and Information System, Data Panel Report", Vol. IIa., NASA Technical Memorandum 87777, June 1986.

[2] Pottier et al., 'Bistatic radar polarimetry theory,' chapter 14 in James D. Taylor (ed.), *Ultra wideband radar technology*, CRC press, e-book ISBN-978-1-4200-3729-6, 2001.

[3] J. Fransis Reintjes et al., *Principles of Radar,* third edition, McGraw-Hill book company, Inc., New York, 1952.

[4] Campbell, J. B. "Introduction to Remote Sensing" , New York London: The Guilford Press, 2002.

[5] R. Duda, P. Hart, and D. Stork, "Pattern Classification", 2nd ed. New York: Wiley, 2000.

[6] R.M.Haralick, K.Shanmugam and H. Dinstein, "Textural Features for Image Classification", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. No. 3, pp. 610 – 621, 1973.

[7] H.H.Loh, J.G.Leu and R.C.Luo, "The Analysis of Natural Textures Using Run Length Features", *IEEE Transactions on Industrial Electronics*, Vol. No. 35, pp. 323-328, 1988.

[8] Chung-Ming Wu, Yung-Chang Chen and Kai-Sheng Hsieh, "Texture Features for Classification of Ultrasonic Liver Images", *IEEE Transactions on Medical Imaging*, Vol. No. 11, pp. 141-152, 1992.

[9] D.Patel, I.Hannah and E.R.Davies, "Texture Analysis for Foreign Object Detection Using a Single Layer Neural Network", *IEEE International Conference on Neural Networks,* Vol. No. 7, pp. 4265-4268, 1994.

[10] A.H.S.Solberg and A.K.Jain, "A Study of the Invariance Properties of Textural Features in SAR Images", *International Geoscience and Remote Sensing Symposium, IGARSS '95,* Vol. No. 1, pp. 670-672, 1995.

[11] A.H.Mir, M.Hanmandlu and S.N.Tandon, "Texture Analysis of CT Images", *IEEE Engineering in Medicine and Biology Magazine,* Vol. No.14, pp 781-786, 1995.

[12] L.Ma, T.Tan, Y.Wang and D.Zhang, "Personal Identification Based on Iris Texture Analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. No. 25, pp. 1519-1533, 2003.

[13] A.Abhyankar and S.Schuckers, "Fingerprint Liveness Detection using Local Ridge requencies and Multiresolution Texture Analysis Techniques", *IEEE International Conference on Image Processing,* pp. 321-324, 2006.

[14] N.Agani, S.A.R.Abu-Bakar & S.H.Sheikh Salleh, "Application of Texture Analysis in Echocardiography Images for Myocardial Infarction Tissue", *Journal Technology,* 46(D), pp. 61-76, 2007.

[15] F.Zhang, Y.Shao, W.Tian and S.Wang, "Oil Spill Identification Based on Textural Information of SAR Image", *IEEE International Geoscience and Remote Sensing Symposium IGARSS 2008,* Vol. No. 4, pp. 1308-1311, 2008.

[16] V.V.Chamundeeswari, D.Singh, and K.Singh, "An Analysis of Texture Measures in PCA-Based Unsupervised Classification of SAR Images", *IEEE Geoscience and Remote Sensing Letters,* Vol. No. 6, pp. 214-218, 2009.

[17] A.Kourgli, A. Belhadj-Aissa and Y. Oukil, "SAR Image Classification Using Textural Modeling", *Radar Conference-Surveillance for a Safer World,* pp. 1-6, 2009.

[18] Anguela T., Zribi, and Baghdadi N., "Analysis of Local Variation of Soil Surface Parameters with TerraSAR-X Radar Data Over Bare Agricultural Fields ", *IEEE transaction on Geoscience and Remote Sensing*, vol. 48, no. 2, pp. 874-881, 2010.

[19] Ouma Y.O., Tateishi R., and Sri-Sumantayo, "Urban Features Recognition and extraction From Very-high Resolution Multi-spectral Satellite Imagery: A Micro-Macro Texture Determination and Integration Framework", *IET Image Processing*, vol. 4, no. 4, pp. 235-254, 2010.

[20] Choudhary P.R., Deshmukh B., Goswami A.K., "Neural Network Based Dunal Landform Mapping From Multispectral Images Using Texture Features", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 1, pp. 171-184, 2011.

[21] V. Vapnik, " The nature of Statistical Learning Theory", Springer, 1995.

[22] M. Brown, H.G. Lewis, S.R. Gunn, "Linear spectral mixture models and support vector machines for remote sensing", *IEEE trans. on Geoscience and Remote Sensing*, vol. 38, no. 5, pp. 2346-2350, 2000.

[23] Haigang Zhan, Ping Si, Chuqun Chen, "Retrieval of Oceanic Chlorophyll Concentration using Support Vector Machine", *IEEE trans. on Geoscience and Remote Sensing*, vol. 41, no. 12, pp. 2947-2951, 2003.

[24] P. Mantero, G. Moser, S.B. Serpico, "Partially Supervised Classification of Remote Sensing Images through SVM-based probability density estimation", *IEEE trans. on Geoscience and Remote Sensing*, vol. 43, no. 3, pp.559-570, 2005.

[25] B. Fei, Jinbai Liu, "Binary tree of SVM: a new fast multiclass training and classification algorithm", *IEEE trans. on Geoscience and Remote Sensing*, vol. 17, no. 3, pp. 696-704, 2006.

[26] A. Mathur, G.M. Floody, "Multiclass and Binary SVM Classification: Implications for Training and Classification Users", *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 2, pp. 241-245, 2008.

[27] C. Lardeux, P.L. Frison, C. Tison, "Support Vector Machine for Multifrequency SAR polarimetric Data Classification", *IEEE trans. on Geoscience and Remote Sensing,* vol. 47, no. 12, part 2, pp. 4143-4152, 2009.

[28] Zhe Liu, Xiaowei Fan, Fangyuan Lv, "SAR Image Segmentation Using Contourlet and Support Vector Machine", *International Conference on Natural Computation, ICNC '09,* vol. 2, pp. 250-254, 2009.

[29] Y. Tarabalka, M. Fauvel, J. Chanussot, "SVM- and MRF-based Method for Accurate Classification of Hyperspectral Images", *IEEE Geoscience and Remote Sensing Letters,* vol. 7, no. 4, pp. 736-740, 2010.

[30] Yoshio Yamaguchi, W. M. Boerner, Jian Yang, Ryoichi Sato, and Hiroyoshi Yama, "ALOS-PALSAR quad. pol. images and their applications," *Proceedings IEEE,* 2009.

G.21025