INTERACTIVE CLASSIFICATION OF SATELLITE IMAGE CONTENT BASED ON QUERY BY EXAMPLE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ORAL DALAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2006

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Volkan Atalay
Supervisor

Examining Committee Members

Prof. Dr. Volkan Atalay (METU, CENG) _____

Prof. Dr. Neşe Yalabık (METU, CENG) _____

Doç. Dr. Yaşar Becerikli (KOU, CENG) _____

Doç. Dr. Veysi Íşler (METU, CENG) _____

Yrd. Doç. Dr. Selim Aksoy (Bilkent, CENG) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Oral Dalay

Signature :

ABSTRACT

INTERACTIVE CLASSIFICATION OF SATELLITE IMAGE CONTENT BASED ON QUERY BY EXAMPLE

Dalay, Oral

M.S., Department of Computer Engineering

Supervisor : Prof. Dr. Volkan Atalay

January 2006, 60 pages

In our attempt to construct a semantic filter for satellite image content, we have built a software that allows user to indicate a few number of image regions that contains a specific geographical object, such as, a bridge, and to retrieve similar objects on the same satellite image. We are particularly interested in performing a data analysis approach based on user interaction. User can guide the classification procedure by interaction and visual observation of the results. We have applied a two step procedure for this and preliminary results show that we eliminate many true negatives while keeping most of the true positives.

Keywords: Interactive classification, query by example, remote sensing, linear classification, texture analysis

ÖZ

ÖRNEĞE DAYALI SORGULAMA ÍLE UYDU GÖRÜNTÜ ÍÇERÍĞÍNÍN
ETKÍLESÍMLÍ SINIFLANDIRILMASI

Dalay, Oral

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Volkan Atalay

Ocak 2006, 60 sayfa

Uydu görüntü içeriği için anlamsal süzgeç oluşturma çabamız çerçevesinde kullanıcının köprü gibi belli bir coğrafik nesneyi içeren az sayıda görüntü alanlarını belirttiği ve ardından benzer nesneleri içeren görüntü alanlarına eriştiği bir yazılım geliştirilmiştir. Özellikle kullanıcı etkileşime dayalı veri çözümleme ile ilgilenilmektedir. Kullanıcı sınıflandırma işlemini etkileşim ve sonuçların görsel incelenmesi ile yönlendirmektedir. Bu amaçla iki adımlı bir işlem uyguladık ve sonuçlar, çoğu gerçek olumluları korurken birçok olumsuzları elediğimizi göstermektedir.

Anahtar Kelimeler: Etkileşimli sınıflandırma, örneğe dayalı sorgulama, uzaktan algılama, doğrusal sınıflandırma, doku analizi

To my parents, my younger brother and
to her.

ACKNOWLEDGMENTS

I would like to thank Prof. Dr. Volkan ATALAY for his supervision, guidance and invaluable suggestions throughout the development of the thesis. I would also thank my family for their encouragement to finish my study. We are grateful to Prof. Dr. Mustafa Turker for providing satellite images. This work is partially supported by TUBITAK EEEAG-103E040 PIA Bosphorus and METU BAP-08-11-DPT2002-K120510. Finally, it is my pleasure to express my deepest gratitude to my friends for sharing hard times of my work.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Classification is one of the main tasks extensively studied in many disciplines such as machine learning, data mining and pattern recognition. Classification is an assignment of new objects to a class from a given set of classes based on the attributes of these objects.

d-features or measurements are used as a input representation of each pattern. It is viewed as a point in d-dimensional space. The goal of representation is to occupy compact and disjoint regions in a d-dimensional feature space. The effectiveness of feature set is determined by how well patterns from different classes can be separated. The objective is to construct decision boundaries in feature space which separate patterns belonging to different classes.

A recognition system is operated in two modes: training (learning) and classification (testing). Figure 1.1 oversimplifies the pattern classification procedure. Any operation contributing in defining a compact representation of pattern is performed in preprocessing module. In the training mode, appropriate representative features of input pattern are found and classifier is trained to partition the feature space. The trained classifier assigns the input pattern to one of the pattern classes based on measured features in the classification mode.

We can think that there exist two classification approaches : data driven and user driven approaches. Data driven classification approach reveals the internal structure of data based on defined algorithms while user driven architecture performs computations based on interaction between user and a machine. In literature, interactive user-driven approaches are less explored than data-driven approaches.

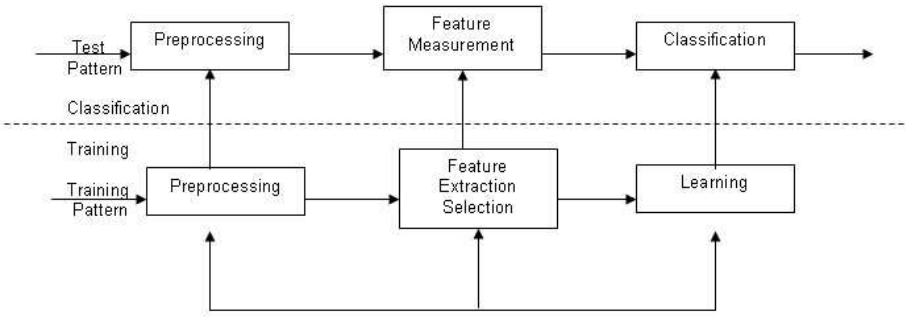


Figure 1.1: Pattern classifier. (adapted from [1])

Classification problem is posed by the users with various preferences, requirements and background knowledge. There is no universally applicable classification theory or method to serve the needs of users. However, user-driven approaches are becoming more popular than data-driven approaches.

Recently launched satellites are capable of capturing panchromatic images with higher resolutions. Ground sampling distance (GSD) of some satellites are shown in Table 1.1. Ground sampling distance is the distance of the centers of neighbored pixels projected to the ground. The pixel size on the ground is the physical size of the projected pixels. State-of-the-art remote sensing image analysis systems allow only queries by geographical location, time of acquisition or type of sensor. This information is often less important than the content of the scene, i.e. structures, objects or scattering properties in addition to the fact that these attributes are not appropriate for the access to the image content [2]. Meanwhile, many new applications of remote sensing data are closer to computer vision and require the knowledge of complicated spatial and structural relationships among image objects. Furthermore, with the increase in the resolution, the amount of data to be handled augments. Therefore, extracting data and then exploring are becoming challenging problems in remote sensing. One of the alternative approaches in the analysis of such large images is to profit from user interaction. For this purpose we designed and developed a basic software. The classification is based on a query by example (QBE) technique where user indicates a few number of example image regions that contains a specific geographical object, for example, road, building and would like to retrieve similar objects on the same or other satellite images.

Table 1.1: Ground sample distances (GSD) of some recent satellites.

Satellites	**GSD**
TK350	13m
KVR1000	2.2m
ASTER	15m
KOMPSAT-1	6.6m
IRS-1C	6.9m
SPOT5	5m
IKONOS	1m
QuickBird	0.6m

In the frame of our collaboration with Université René Descartes Paris V and Institut Geographique National, Paris, France, our general aim is to detect man made (structural) geographical objects from high resolution satellite images. User starts with a small set of sample images and refines incrementally the classification results by interaction. A simple diagram of our system is shown in Figure 1.2. The system includes mainly two phases: a coarse search and refined detection. Spectral, linear and textural features are used for coarse search while refined detection is performed based on structural features [3]. Structural feature extraction algorithms not only require existence of these geographical objects but demand a high computational time as well. Therefore, it is necessary to apply a coarse object search algorithm to find possible (candidate) locations on the image and then apply a refined algorithm (structural in this case) on each candidate region.

1.2 An Overview of Related Systems

This section provides a brief overview of related research on classification of satellite images emphasizing on interaction with the user. The problem of classification of high resolution satellite images are being studied recently, boosted by the need to access relevant information in an understandable and directly usable form and to provide user friendly interface for query and browsing [2]. The resolution of satellite images is very high due to recent technologies in the area of remote sensing as a result the automated classification is then very difficult and often leading to an undesired results.

To address this challenge, one possible solution is to profit from user interaction by creating visual queries in an interactive manner. The method in a search should

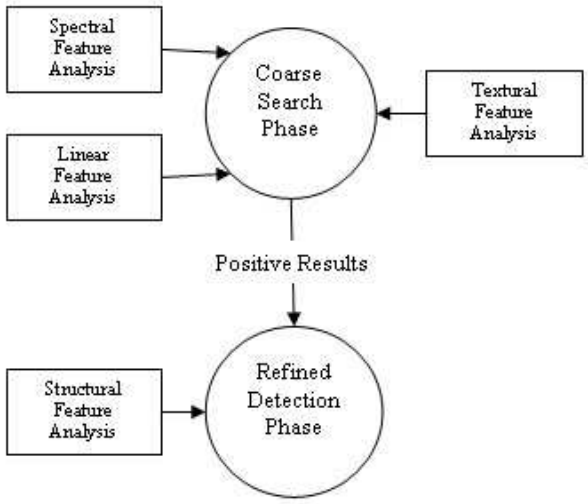


Figure 1.2: Simple diagram of our system.

operate in real time and should maximize the ratio between the quality of results and the amount of interaction between the user and the system. Incorporating image information and human expert knowledge with digital image processing improves remote sensing image analysis [4].

In recent years, the ”interactive systems” and ”human in the loop” became to be popular approaches. An important example system QBIC [5] allows queries on a large image and databases based on example images, user-constructed sketches and drawings and selected color and texture patterns. In the MIT system FourEyes [6] texture representations can be interactively managed another example WebSEEK system [7] includes the user's feedback for the dynamic feature vector re-computation;in the MARS system [8] a relevance feedback approach is used to improve the retrieval performance and in PicHunter [9, 10], a Bayesian relevance feedback architecture is applied.

The VisiMine [11] is a system for data mining and statistical analysis of large collections of remotely sensed images. It provides the methodology and the infrastructure required for the analysis of satellite images. The architecture supports three levels of features: pixel, region and scene. It distinguishes feature vectors of each level. The feature extraction at pixel level starts with the analysis of spectral and textural properties while the segmentation algorithm is the first step of extraction of region level features. Original images and the features extracted from these are stored in a

database system and indexed accordingly for fast retrieval. It also provides a user interface, gives user the option of browsing and manipulation of satellite images. Queries for data mining, visualization of the results of the data analysis and the images with associated features can be seen by interface. The image processing, statistics and machine learning libraries are written in C/C++, graphical user interface (GUI) is implemented in Java. For database applications, Oracle is used as relational database management system (RDBMS) which includes storing or accessing to satellite images, feature vectors and metadata.

Interactive Visual Image Classification (IVICS) [12] has been designed to provide the analyst with a set of interactive tools and displays which will aid in the proper selection and labeling of training samples for image classification. It is developed in 1993 funded by NASA.

A knowledge driven approach called knowledge driven information mining (KIM) is developed in [13]. There are 2 main modules in the system, the first includes indexing, image feature extraction and computationally intensive algorithms for offline data analysis in the archive. The second module consists of a graphical man-machine interface that manages the information fusion for interactive interpretation and the image information mining functions. It supports to adaptively incorporate application specific interests. Semantic image content defined by user interpretation is linked with Bayesian networks to a completely unsupervised content-index. Based on this stochastic link, the user can query the archive for relevant images and obtains a probabilistic classification of the entire image archive as an intuitive information representation.

Our system is not a final system, it is a preprocessing to structural analysis phase. The systems in the literature have more data and image analysis techniques employed but using these systems requires to be an expert however a naive user can employ our system to perform rough filtering.

1.3 Purpose and Improvements Achieved

We are particularly interested in performing a satellite image analysis based on user interaction. We have designed and implemented a basic software for this purpose. Our system allows user to select the region of interest and eliminate the non-representative

regions. In addition, it allows user to interact with the retrieved results of the system. This is called relevance feedback. The basic idea of relevance feedback is to perform an initial query, get feedback from user as to what regions are relevant and then add more regions from these relevant candidates to the query. The user interaction compensates the gap between user intention and visual appearance.

In our classification system, user initially points a few number of sample regions that contains the sought geographical objects and features are computed from these regions. Whole image is scanned to search for candidate regions that may contain similar objects. Feature values of both the sample and candidate regions can be projected into 2 dimensions for visualization purposes. User can subsequently eliminate some of the candidates and classification can be performed again in an iterative and interactive manner. Final candidates can be input to the second phase to refine the detection based on the structural properties.

There are several issues in our problem.

- It falls into the category of query-by-example.
- It is interactive.
- It is a binary classification problem: i.e. regions are labeled as bridge or non-bridge.

The feature extraction and data analysis techniques are available in the literature. In our approach, we combined these techniques to operate together in iterative and interactive manner. However, the most challenging part is to learn from a small set of samples which belong only to a single class. We alleviate this problem by incrementally refining the results with user interaction.

1.4 Organization of the Thesis

The organization of thesis is as follows: In Chapter 2, the techniques that we used in our approach are explained. The detailed system is described in Chapter 3. Chapter 4 presents experimental results. The thesis concludes with Chapter 5 in which the presented study is discussed.

CHAPTER 2

BASIC CONCEPTS

In this chapter, basic concepts and background information of the employed methods are given. Feature extraction and analysis methods are organized as image analysis and data analysis methods. Image analysis methods include spectral, linear and textural feature extraction. Support vector classifier (SVC) and multidimensional scaling (MDS) make up data analysis methods.

2.1 Image Analysis Methods

2.1.1 Spectral Feature Extraction

A graylevel image consists of a byte per pixel. Each pixel in a satellite image image has value ranging between 0 and 255. Satellite images are panchromatic; panchromatic images are graylevel images. The popular approach for spectral feature extraction is to use the mean and standard deviation of pixels in an image or image region. The mean is of a data set is simply the arithmetic average of the values in the set while the variance is the arithmetic average of the squared differences between the values and the mean. The values are computed as follows.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} I(x_i, y_i) \qquad \sigma = \frac{1}{n}\sum_{i=1}^{n}(I(x_i, y_i) - \mu)^2 \tag{2.1}$$

where $I(x_i, y_i)$ is the graylevel of the corresponding image pixel at (x_i, y_i) position at $i = 1 \cdots n$, n is the size of image region.

2.1.2 Linear Segment Extraction

The aim of linear segment extraction is to extract a set of linear segments. Each linear segment can be defined by Equation 2.2

$$L = \{P_1, P_2\} \tag{2.2}$$

where P_1 and P_2 are the *dominant points* of line L and,

$$P_1 =< x_1, y_1 > P_2 =< x_2, y_2 >$$

In edge-based methods, linear segment extraction starts with the detection of local information as the intensity edge points and continues with the boundary construction by thresholding and contour following, and polygonal approximation to the boundary curves.

Edge Detection by Canny's Method An edge is a local information attached to an individual image pixel and it is calculated from the image function behavior of neighborhood of pixel. It is a vector variable containing two values:

- magnitude, and
- direction.

Edge magnitude is the magnitude of the gradient $g_{mag}(x, y)$, and the edge direction $\phi(x, y)$ is rotated with respect to the gradient direction $\psi(x, y)$ by -90^o. The gradient direction ψ gives the direction of maximum growth of the gradient function.

Edge profile in the direction of gradient is usually given for an edge and Figure 2.1 illustrates in one dimension a few examples among several standard edge profiles. Edge detectors are generally tuned for a certain type of edge profile. Gradient magnitude, whose absolute value $|g_{mag}(x, y)|$ is often used, and gradient direction ψ continuous image functions calculated from Equation 2.3

$$|g_{mag}(x, y)| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \tag{2.3}$$

$$\psi(x, y) = \arctan(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}) \tag{2.4}$$

respectively, where arctan(*x,y*) is the angle (in radians) from the x-axis to the point (*x,y*). Since a digital image is discrete in nature, Equation 2.3 and 2.4 must be approximated by difference equations [14, 15]. Gradient operators can be examined in three categories:

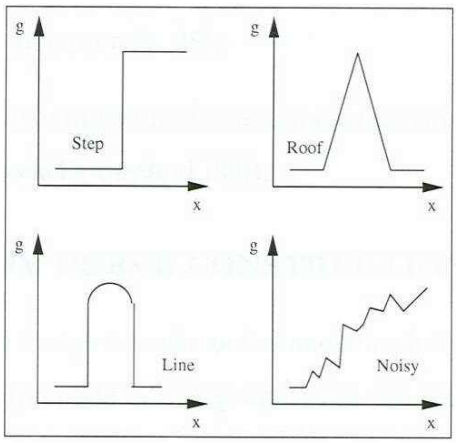


Figure 2.1: Typical edge profiles.

- Operators approximating derivatives of the image function using differences. Some of the are rotationally invariant (e.g. Laplacian [16, 15]), and thus are computed from one convolution mask [15]. Others that approximate first derivatives using several masks (e.g. Roberts, Prewitt and Sobel operators [14, 17, 15]).

- Operators based on zero crossings of the image function second derivative (e.g. Canny edge detector [18, 17]).

- Operators which attempt to match an image function to a parametric model of edges (e.g. Hueckel's method [19]).

Edge detection is the first phase of linear segment extraction, and Canny's edge operator is used for this purpose. It is an optimal edge detection approach for step edges corrupted by white noise [18] where a step edge is given by its position, orientation and magnitude.

Suppose G is a $2D$ Gaussian process defined in Equation 2.5, and assume that we wish to convolve the image with an operator G_n which is a first derivative of G in the direction $\mathbf{n}$.

$$G_n = \frac{\partial G}{\partial n} = \mathbf{n} \cdot \nabla G \tag{2.5}$$

The direction $\mathbf{n}$ should be oriented perpendicular to that of edge. Since this direction is not known in advance, a robust estimate of it can be obtained based on smoothed gradient direction. If I is the image, n can be estimated as in Equation 2.6

$$\mathbf{n} = \frac{\nabla(G * I)}{|\nabla(G * I)|} \tag{2.6}$$

Location of the edge is, then, at the local maximum in the direction **n** of the operator G_n convolved with image I.

$$\frac{\partial}{\partial n} G * I = 0 \tag{2.7}$$

Substituting G_n from Equation 2.5 into Equation 2.7, the Equation 2.8 is obtained.

$$\frac{\partial^2}{\partial n^2} G * I = 0 \tag{2.8}$$

Equation 2.8 gives the local maxima in the perpendicular direction to the edge *(non maximal suppression)*.

After an input image I is convolved with a symmetric Gaussian G using an estimate direction n, the directional second derivative can be computed. Then, the strength of the edge is measured as in Equation 2.9.

$$|G_n * I| = |\nabla(G * I)| \tag{2.9}$$

Algorithm 1 Canny Edge Detection Algorithm

1: *A gaussian of scale σ is used to convolve Image I*
2: *local edge normal directions n are estimated using Equation 2.6 for each pixel in the image*
3: *Edge locations are found using Equation 2.8 (non-maximal suppression)*
4: *The magnitude of the edges are calculated using Equation 2.9*

2.1.2.1 Boundary Curve Forming by Linking Nonzero Pixels

Once an edge map of an image is constructed and thresholded to produce a binary image, a connected components labeling algorithm can employed to group edge pixels into maximal connected regions [14].

If there is a sequence of pixels $(p_0, p_1, \ldots, p_n)$ where $p_0 = p, p_n = q$ and p_i is a neighbor of p_{i-1} for $i = 1, 2, \ldots, n$ then, two edge pixels p and q are belong to the

same connected component C. Thus, the definition of a connected component depend on the definition of the neighborhood. When only the North, South, East, West neighbors of a pixel are considered as a part of its neighborhood, then the resulting regions are called 4-connected. If Northeast, Northwest, Southeast, Southeast and Southwest neighbors are also considered, the the resulting regions are called 8-connected. Literature regarding connected components labeling contains various efficient and well known algorithms and detailed information about these algorithms can be found in [14, 15].

The second step is forming a boundary curve, the gradient or strength measure is used for marking each pixel as a result of edge detection method. The true edges are likely to be corresponded by the high valued pixels.

Canny's hystresis thresholding [18] uses both high threshold and low threshold values rather than using a single threshold on the gradient value. The main idea is; weak responses usually corresponds to noise, but if these are connected to a pixel with strong responses, then they are more likely to be actual edges in an image. Streaking can be eliminated by this method. This method prevents breaking an edge into many small disconnected pieces b starting an edge when a pixel is above the high threshold and keeping all pixels above the low threshold in intervals bounded by pixels above the high threshold. When tracking adjacent non-zero edge elements higher than low threshold, connected components are searched. Figure illustrates the results of linking with hystresis on a set of edge magnitude images.

2.1.2.2 Boundary Curve Segmenting by Polygonal Approximation

Edge detection and edge linking algorithms produce a sequence of linked $< x, y >$ coordinate pairs. Polygonal approximation is generally applied to obtain a more compact representation from the linked description of edge pixels, and to decrease memory and computational costs for feature extraction.

A polygonal approximation routine consists of three steps as indicated below.

- First step is the initialization where the dominant points are tentatively located in the curve data providing an initial approximation.
- In the second step, this initial polygon is iteratively adjusted, by adding additional dominant points. This adjustment continues until some specific good-

ness of criteria are met.

- Finally, in the third step, edges which have similar properties are merged or concatenated.

Let us have a boundary curve C composed of linked edge pixels defined in Equation 2.10

$$\begin{aligned} C &= P_1, \ldots, P_n \qquad where \\ P_1 &=< x_1, y_1 > \qquad and, \\ P_n &=< x_n, y_n > \end{aligned} \tag{2.10}$$

Main objective of these methods is to represent a given C with a list of connected straight lines L_i, defined in Equation 2.11

$$\begin{aligned} L &= L_1, \ldots, L_m \qquad where \\ L_1 &= < x_1, y_1 >, < x_i, y_i > \qquad and, \\ L_m &= < x_k, y_k >, < x_n, y_n > \end{aligned} \tag{2.11}$$

starting P_1 and ending at P_n which are presented in 2.10, such that the result is similar to the original digital boundary curve as much as possible.

A boundary curve can be approximated with varying precision; if a more precise description is necessary, a large number of line segments may be employed. One potential drawback of polygonal approximation is that the result is not usually unique; for example an approximation of same the curve that begin at a different initial point on the curve will likely yield different results.

There is a variety of techniques for segmenting digital arcs into simple line segments. The techniques range from iterative end-point fitting and splitting to using tangent angle detection, prominence, or high curvature as the basis of the segmentation, and related work can be found in [14].

The segmentation problem is formalized in [14] as follows. Let $L = (x, y)|\alpha x + \beta y + \gamma$ where $\alpha^2 + \beta^2 = 1$ be the line segment defined by the end points $< x_1, y_1 >$ and $< x_n, y_n >$.

For any point $< x_n, y_n >$, d_n is the distance between point $< x_n, y_n >$ and L. The distance between point and a line is calculated using an Equation 2.12.

$$d_n = \frac{|\alpha x_n + \beta y_n + \gamma|}{\sqrt{\alpha^2 + \beta^2}} \tag{2.12}$$

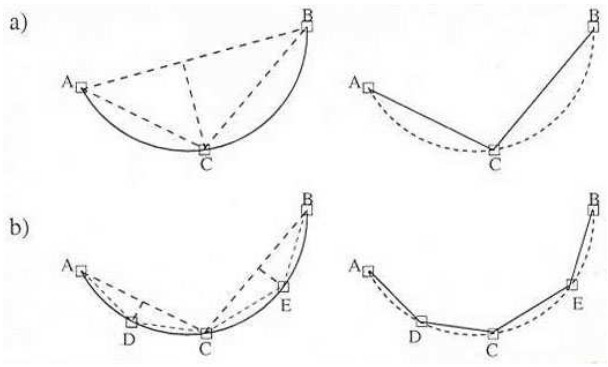


Figure 2.2: Geometry of the iterative end point fit and split. a) C is the pixel having the farthest distance to line AB. The algorithm segments the curve sequence at pixel C, creating two curve sequences. b) Splitting procedure is recursively applied to newly created segments.

If $d_m > d^*$, the sequence C is split into two subsequences, $S_1 = < x_1, y_1 >, \cdots, < x_m, y_m >$ and $S_2 = < x_{m+1}, y_{m+1} >, \cdots, < x_n, y_n >$, and these sequences are represented by their line segments L_1 and L_2.

Splitting is applied recursively until the edge is reduced into the short segments, if the threshold *granularity* is larger than the distance of point to the fragment the recursion is stopped.

2.1.3 Textural Feature Extraction

In computer vision and image processing applications, "texture" is one of the most important characteristics for identifying, discriminating and synthesizing an objects in an image. Textural features contain information about the spatial distribution of the tonal variations creating patterns on the surface. Analysis of images by using textural properties has a very wide of range of application areas such as medical imaging, remote sensing, industrial quality inspection and content based image retrieval.

There have been two alternative approaches to describe one dimensional (1-d) signals, the first one represents signals as a function of time and the second as a function of frequency. A representation can be transformed from one to another via the Fourier or inverse Fourier transforms and they thus carry the same information but in different forms. Both of the representations are somewhat idealizations since the first one operates on sharply defined time instants and the second with infinite

waveform trains on rigorously defined frequencies. It is motivated to construct an approach that represents signals as a function of both time and frequency. Dennis Gabor in 1946 proposed the use of special elementary functions, later named after him as Gabor functions, to represent signals simultaneously in time and frequency [20]. In this section, a brief introductory description of Gabor function is given. Interested readers may refer to the cited references for further investigation.

2.1.3.1 Gabor Functions

Gabor functions are one of the most widely used and biologically plausible texture feature extraction tools in the current state of the art of low-level image processing. Since Gabor filters can be considered as orientation and scale tunable edge and line (bar) detectors, and the statistics of these micro features in a given region are often used o characterize underlying texture information, they have been extensively used in several image analysis applications [21]. Gabor filtering in texture features and its advantages are confirmed by many researchers in the field of image processing and pattern recognition.

Another benefit of Gabor functions is their response which is highly localized in both spatial and frequency domain. Empirical studies suggest that distributed conjoint spatial/spatial-frequency representations are fundamental to visual information encoding in the cortex of all mammals, including the human species [22] and they are also ideally suited for low-level image representations. They are primarily useful because they improve pattern separability achieving at the same time geometrical invariance to scale, rotation changes and perspective distortions.

A two dimensional Gabor function *g(x, y)* and its Fourier transform *G(u, v)* is written in Equation 2.13 and Equation 2.14 respectively.

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) exp\left[-\frac{1}{2}\left(\frac{x^2}{{\sigma_x}^2} + \frac{y^2}{{\sigma_y}^2}\right) + 2\pi W j x\right] \tag{2.13}$$

and

$$G(u, v) = exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{{\sigma_u}^2} + \frac{v^2}{{\sigma_v}^2}\right]\right\} \tag{2.14}$$

where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$.

A set of filtered images obtained by convolving the given image with Gabor filters.

By appropriate rotations and dilations of $g(x, y)$ in Equation 2.13 through the functions generated on Equation 2.15 will form a self-similar filter dictionary.

$$\begin{aligned} &g_{mn}(x,y) = a^{-m}G(\tilde{x},\tilde{y}), \qquad a>1, \qquad m,n = integer \\ &\tilde{x} = a^{-m}(x\cos\theta + y\sin\theta), \qquad and \qquad \tilde{y} = a^{-m}(-x\sin\theta + y\cos\theta) \end{aligned} \tag{2.15}$$

where $\theta = n\pi/K$ and S is the number of scales and K is the total number of orientations. The scale factor a^{-m} in Equation 2.15 is meant to ensure that energy is independent of m.

2.1.3.2 Feature Representation

For an image $I(x, y)$, the discrete Gabor wavelet transform is given by a convolution:

$$W_{mn}(x,y) = \int I(x_1,y_1)g^*_{mn}(x-x_1, y-y_1)dx_1dy_1 \tag{2.16}$$

where * indicates complex conjugate.

$$\mu_{mn} = \int\int |W_{mn}(xy)|dxdy, \qquad and \qquad \sigma_{mn} = \sqrt{\int\int (|W_{mn}(x,y)| - \mu_{mn})^2 dxdy} \tag{2.17}$$

μ_{mn} and σ_{mn} are used to construct feature vector. The resultant feature vector becomes

$$\bar{f} = [\mu_{00}\ \sigma_{00}\ \mu_{01}\ \cdots\ \mu_{mn}\ \sigma_{mn}] \tag{2.18}$$

2.2 Data Analysis Methods

2.2.1 Data Classification

Classification is one of the main tasks extensively studied in many disciplines such as machine learning, data mining and pattern recognition. Classification is an assignment of new objects to a class from a given set of classes based on the attributes of these objects.

Two approaches to classification can be introduced. The first assumes the knowledge of the underlying class conditional probability density functions. In the second one, the data is used to estimate the decision boundaries directly without calculation of probability density functions.

The classification scheme is usually based on already labeled set of patterns. This set of pattern is called training set and learning strategy is then called as supervised learning. The strategy can also be unsupervised, in the sense that the system is not given an *a priori* labeling of patterns, instead it establishes the classes itself based on the statistical regularities of the patterns.

Figure 2.3 shows various dichotomies that appear in statistical pattern recognition which is adapted from [1]. Bayes decision theory 2.2.1.1 constitutes the basis of almost all classifiers.

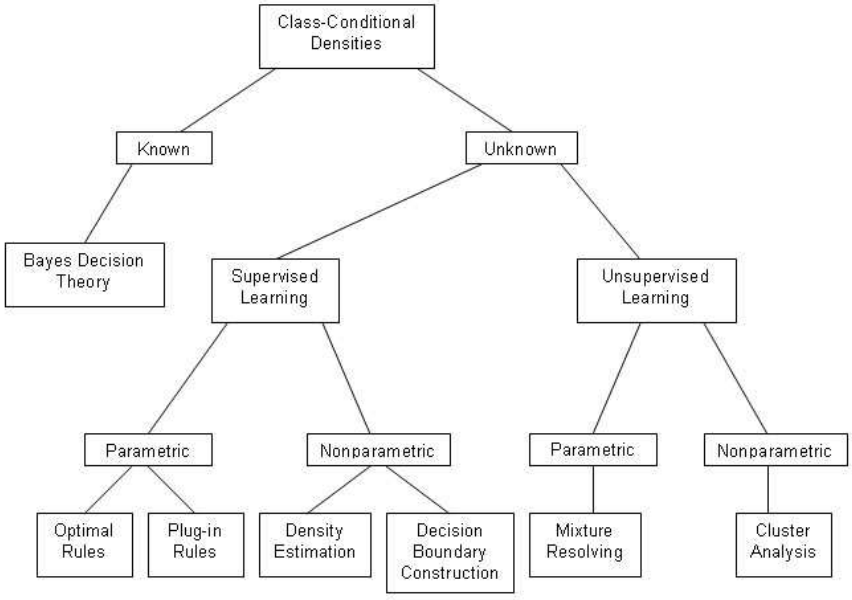


Figure 2.3: Various approaches in statistical pattern recognition. (adapted from [1])

2.2.1.1 Bayes Decision Theory

Consider C classes, $\omega_1, \cdots, \omega_C$ with *a priori* probabilities $p(\omega_1), \cdots, p(\omega_C)$, assumed known. If we have no other information regarding an object other than the class probability distribution then we would assign an object to class ω_j if

$$p(\omega_j) > p(\omega_k) \qquad k = 1, \cdots, C;\ k \neq j \tag{2.19}$$

If we have a *measurement vector* or *observation vector* **x**, we may wish to assign **x** to one of the C classes. A decision rule based on probabilities is if the probability of class ω_j given the observation **x** is greater than over all classes $\omega_1, \cdots, \omega_c$ then assign **x** to class ω_j. That is,

$$p(\omega_j|\mathbf{x}) > p(\omega_k|\mathbf{x}) \qquad k = 1, \cdots, C;\ k \neq j \tag{2.20}$$

The measurement space is partitioned into C regions $\Omega_1, \cdots, \Omega_C$ such that if $x \in \Omega_j$ then **x** belongs to class ω_j. It can also be expressed informally stating that

$$posterior = \frac{likelihood \times prior}{evidence} \tag{2.21}$$

The *a posteriori* probabilities $p(\omega_j|\mathbf{x})$ may be expressed in terms of the *a priori* probabilities and the class conditional density functions $p(\mathbf{x}|\omega_i)$ using Bayes' theorem

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)p(\omega_i)}{p(\mathbf{x})} \tag{2.22}$$

Therefore the decision rule may be written as : assign **x** to ω_j if

$$p(\mathbf{x}|\omega_j)p(\omega_j) > p(\mathbf{x}|\omega_k)p(\omega_k) \qquad k = 1, \cdots, C;\ k \neq j \tag{2.23}$$

Whenever we observe a particular **x**, the probability of error is

$$P(error|\mathbf{x}) = \begin{cases} P(\omega_1|\mathbf{x}) & if\ we\ decide\ \omega_2 \\ P(\omega_2|\mathbf{x}) & if\ we\ decide\ \omega_1 \end{cases} \tag{2.24}$$

2.2.1.2 Support Vector Machines

The decision boundaries can be obtained directly (geometric approach). In this study, we concentrate on decision boundary construction by optimizing certain cost functions.

The relationship between Bayes decision rule and Support vector machines (SVM) has already been explored and explained in the literature [23].

Consider we have a set of training patterns $\mathbf{x}_i, i = 1, \ldots, n$ assigned to one of two classes, ω_1 and ω_2, with corresponding labels $y_i = \pm 1$. A linear discriminant function is denoted as in Equation 2.25.

$$g(\mathbf{x}) = \omega^T \mathbf{x} + \omega_0 \tag{2.25}$$

with decision rule

$$\omega^T \mathbf{x} + \omega_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \mathbf{x} \in \begin{cases} \omega_1 \; with \; corresponding \; numeric \; value \; y_i = +1 \\ \omega_2 \; with \; corresponding \; numeric \; value \; y_i = -1 \end{cases} \tag{2.26}$$

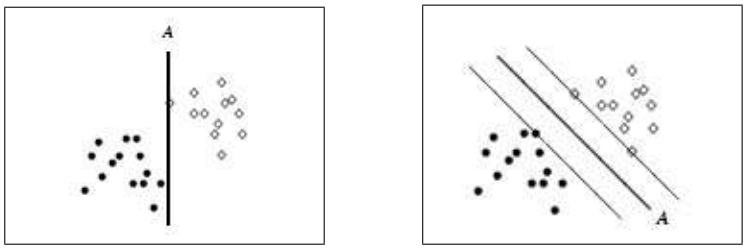
Thus, all training points correctly classified if

$$y_i(\omega^T \mathbf{x}_i + \omega_0) > 0 \qquad for \; all \; i \tag{2.27}$$

Support Vector Machines are learning machines that can perform binary classification and regression estimation tasks. They perform the structural risk minimization principle.

The idea behind SVM is as follows: it maps the input vectors $x_i \in \Re$ into a higher dimensional feature space $\phi(x_i) \in H$ and constructs Optimal Separating Hyperplane, which maximizes the margin the distance between hyperplane and the nearest data points of each class in the space H. The mapping ϕ is performed by a kernel function $K(x_i, x)$ which defines an inner product in the space H.

Figure 2.4b shows separable set of points with separating hyperplane, A. The *margin* is determined by the maximal margin classifier whose distance is largest to two parallel hyperplanes on each side of the hyperplane A. The larger the *margin*, the better the generalization error of the linear classifier defined by separating hyperplane.



(a) (b)

Figure 2.4: Two linearly separable sets of data with separating hyperplane. Thick line leaves the closest points at maximum distance. (adapted from [24])

Margin $b > 0$, is the variant of perceptron rule, and we seek a solution so that

$$y_i(\omega^T \mathbf{x}_i + \omega_0) \geq b \tag{2.28}$$

Without loss of generality, $margin \; b = 1$ may be taken, defining the term *canonical hyperplanes*, $H_1 : \omega^T \mathbf{x}_i + \omega_0 \geq +1$ and $H_2 : \omega^T \mathbf{x}_i + \omega_0 \geq -1$, and we have

$$\begin{aligned} \omega^T \mathbf{x}_i + \omega_0 \geq +1 \qquad & for\ y_i = +1 \\ \omega^T \mathbf{x}_i + \omega_0 \leq -1 \qquad & for\ y_i = -1 \end{aligned} \tag{2.29}$$

The distance of two hyperplanes to the separating hyperplane is *margin*, $1/|\omega|$. Figure 2.5shows the separating hyperplane and canonical hyperplanes for separable data sets. The points that lie on the canonical hyperplanes are called *support vectors*. Therefore maximizing the margin means that we are looking for a solution minimizing $|\omega|$. An approach to optimization problems with equality and inequality constraints is Lagrange formalism, that can be stated for SVM as follows.

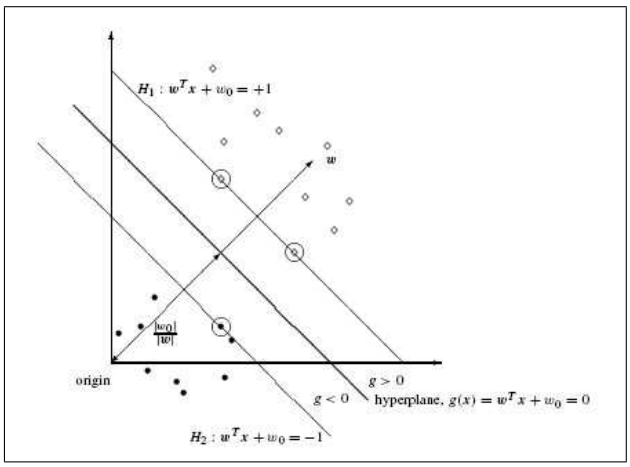


Figure 2.5: H_1 and H_2 are canonical hyperplanes, a hyperplane through closest points (marked as ring) are termed as *support vectors*. (adapted from [24])

$$L_p = \frac{1}{2}\omega^T\omega - \sum_{i=1}^{n} \alpha_i (y_i(\omega^T x_i + \omega_0) - 1) \tag{2.30}$$

where $\alpha_i, i = 1, \ldots, n; \alpha_i \geq 0$ are the Lagrange multipliers. ω and ω_0 are *primal parameters* and the number of parameters is $p + 1$, where p is the dimensionality of feature space.

Differentiating L_p with respect to ω and ω_0 and equating to zero yields

$$\begin{aligned} \textstyle\sum_{i=1}^{n} \alpha_i y_i &= 0 \\ \omega &= \textstyle\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \end{aligned} \tag{2.31}$$

Substitute Equation 2.31 into Equation 2.30 gives the *dual form* of the Lagrangian

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \tag{2.32}$$

which is maximized with respect to α_i subject to

$$\alpha_i \geq 0 \qquad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.33}$$

Once we have the Lagrange multipliers α_i, the value of ω_0 can be found through

$$\alpha_i(y_i(\mathbf{x}_i^T \omega + \omega_0) - 1) = 0 \tag{2.34}$$

using any of support vectors (patterns for which $\alpha_i \neq 0$) or an average of all support vectors.

$$n_{SV}\omega_0 + \omega^T \sum_{i \in SV} \mathbf{x}_i = \sum_{i \in SV} y_i \tag{2.35}$$

where n_{SV} is the number of support vectors. Equation 2.31 is used for solution of ω.

$$\omega = \sum_{i \in SV} \alpha_i y_i \mathbf{x}_i \tag{2.36}$$

since $\alpha_i = 0$ for other patterns. The new pattern $\mathbf{x}$ is classified according to the sign of,

$$\omega^T \mathbf{x} + \omega_0 \tag{2.37}$$

In many real world problems, there will be no linear boundary separating classes so searching for optimal hyperplane is meaningless. Support vector machines may be applied in a transformed feature space $\phi(x)$, for some nonlinear function ϕ.

$$g(\mathbf{x}) = \omega^T \phi(\mathbf{x}) + \omega_0 \tag{2.38}$$

with decision rule

$$\omega^T \phi \mathbf{x} + \omega_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \mathbf{x} \in \begin{cases} \omega_1 \; with \; corresponding \; numeric \; value \; y_i = +1 \\ \omega_2 \; with \; corresponding \; numeric \; value \; y_i = -1 \end{cases} \tag{2.39}$$

The transformed feature vectors can be replaced by kernel function.

$$\mathbf{K}(\mathbf{x}, \mathbf{y}) = \phi^T(\mathbf{x})\phi(\mathbf{y}) \tag{2.40}$$

The discriminant function becomes

$$g(\mathbf{x}) = \sum_{i \in SV} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + \omega_0 \tag{2.41}$$

The kernel representation has an advantage of that we need only **K** for training. There are different type of kernels can be employed. We are interested in only Radial basis function (RBF) kernel. The first reason of better performance is that the RBF kernel nonlinearly maps samples into a higher dimensional space, unlike the linear kernel. The second reason is the number of hyperparameters which influences the complexity of model selection. The polynomial kernel has more hyperparameters than the RBF kernel. Finally, the RBF kernel has less numerical difficulties.

$$K(x_i, x) = exp(-\gamma||x_i - x||^2), \qquad \gamma > 0 \tag{2.42}$$

2.2.2 Visualization by Multidimensional Scaling

Multidimensional scaling (MDS) is used as a visualization tool to present multivariate data in a human accessible form, and as a preliminary transformation applied to the data prior to the use of other analysis tools like clustering and classification.

Almost always, multivariate data are supplied in one of two basic forms: linear methods and nonlinear ones. In the second case, we use the term weights for the pairwise relationships between the data elements. Weight-based methods attempt to assign coordinates to the data elements in the low dimensional space such that their embedding reflects in some sense their pairwise relationships. Distances, similarities, and dissimilarities are the most commonly used types of weights. MDS is the customary notion for these methods that use distances or dissimilarities as weights.

MDS is a method for visualizing proximity data, that is, data where objects are characterized by dissimilarity values for all pairs of objects. MDS constructs maps of these objects in $\Re^k$ by interpreting the dissimilarities as distances [25].

Set of n points in p-dimensional space, $\mathbf{x}_1, \cdots, \mathbf{x}_n$, we can easily calculate the Euclidean distance between each pair of points. The coordinates of set of points in a dimension e can be achieved by a decomposition of $n \times n$ matrix **T**, the between-individual sums of squares and products matrix.

$$\mathbf{T} = \mathbf{X}\mathbf{X}^T \tag{2.43}$$

where $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]^T$ is the $n \times p$ matrix of coordinates. The distance between two individuals i and j is

$$d_{ij}^2 = T_{ii} + T_{jj} - 2T_{ij} \tag{2.44}$$

where

$$T_{ij} = \sum_{k=1}^{p} x_{ik} x_{jk} \tag{2.45}$$

If we think that the centroid of the points $\mathbf{x}_i$, $i = 1, \ldots, p$, is at the origin. Then the elements of the matrix $\mathbf{T}$ can be expressed in terms of dissimilarity matrix inverting Equation 2.44, giving

$$T_{ij} = -\frac{1}{2}[d_{ij}^2 - d_{i\cdot}^2 - d_{\cdot j}^2 + d_{\cdot\cdot}^2] \tag{2.46}$$

where

$$d_{i\cdot}^2 = \frac{1}{n}\sum_{j=1}^{n} d_{ij}^2; \qquad d_{\cdot j}^2 = \frac{1}{n}\sum_{i=1}^{n} d_{ij}^2; \qquad d_{\cdot\cdot}^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2; \tag{2.47}$$

Equation 2.47 allows us to construct $\mathbf{T}$ from $n \times n$ dissimilarity matrix $\mathbf{D}$. As it is a real symmetric, $\mathbf{T}$ can be written in the form

$$\mathbf{T} = \mathbf{U}\Lambda\mathbf{U}^T \tag{2.48}$$

where Λ is the diagonal matrix of eigenvalues $\lambda_1, \ldots, \lambda_n$, $\mathbf{U}$ is the eigenvectors of $\mathbf{T}$. Therefore

$$\mathbf{X} = \mathbf{U}\Lambda^{\frac{1}{2}} \tag{2.49}$$

as our matrix of coordinates.

CHAPTER 3

INTERACTIVE CLASSIFICATION OF SATELLITE IMAGES

The basic idea is to do an initial query with some example regions indicated by the user, get feedback from user at each step as to what regions are relevant and then add some more examples or remove some from known regions to the query. Positive results are example image regions containing sought geographical object, while negative classified results may contain any object.

3.1 Description of The System

We propose here a system of two serially working steps. The positive results of first step are passed into the second step for further analysis. The context diagram shown in Figure 3.1 shows the data flow of the system.

The first step starts initially by forming the positive set. This set is constructed by the user through our interface in an interactive manner. The negative set is automatically generated by our system. In the first step, the spectral or linear features of examples are analyzed to eliminate most of true negatives. For visualization purposes, multidimensional scaling (MDS) is used to show the 2D plot of feature set that we created for linear analysis. The simple data flow diagram for first pass is shown in Figure 3.2

The example set constructed in the first step is kept to be used during second step. The negative set again generated automatically from the positive labeled results of the first pass. Multidimensional scaling is used in order to visualize sets. The simple data flow diagram for second pass is shown in Figure 3.3

However the resultant regions are not the final results, they are input to the second phase; structural feature analysis phase.

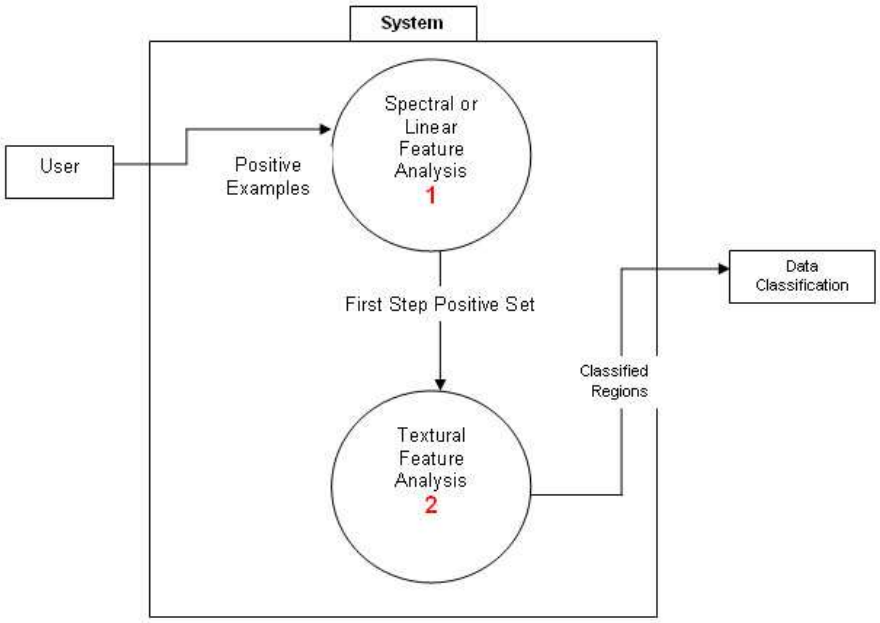


Figure 3.1: System flow diagram.

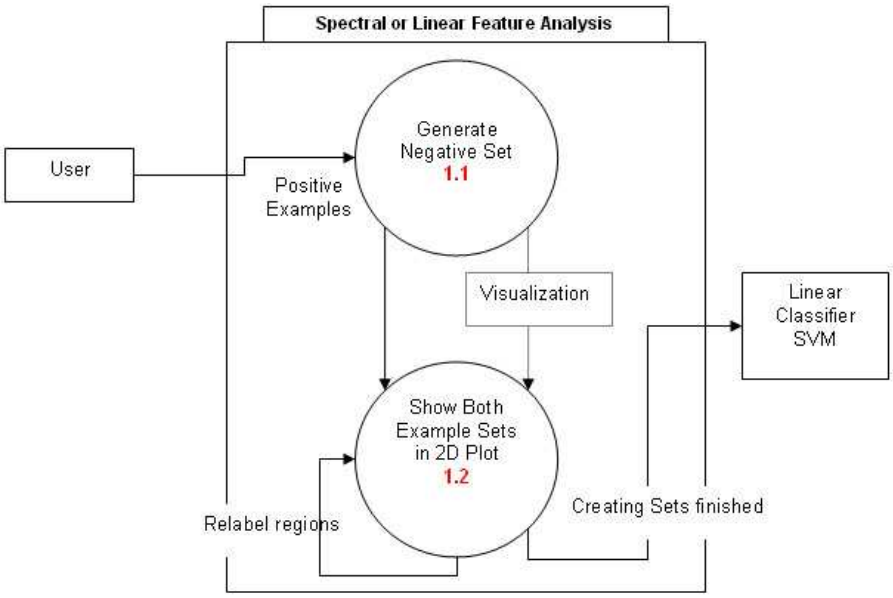


Figure 3.2: Flow diagram of the first step.

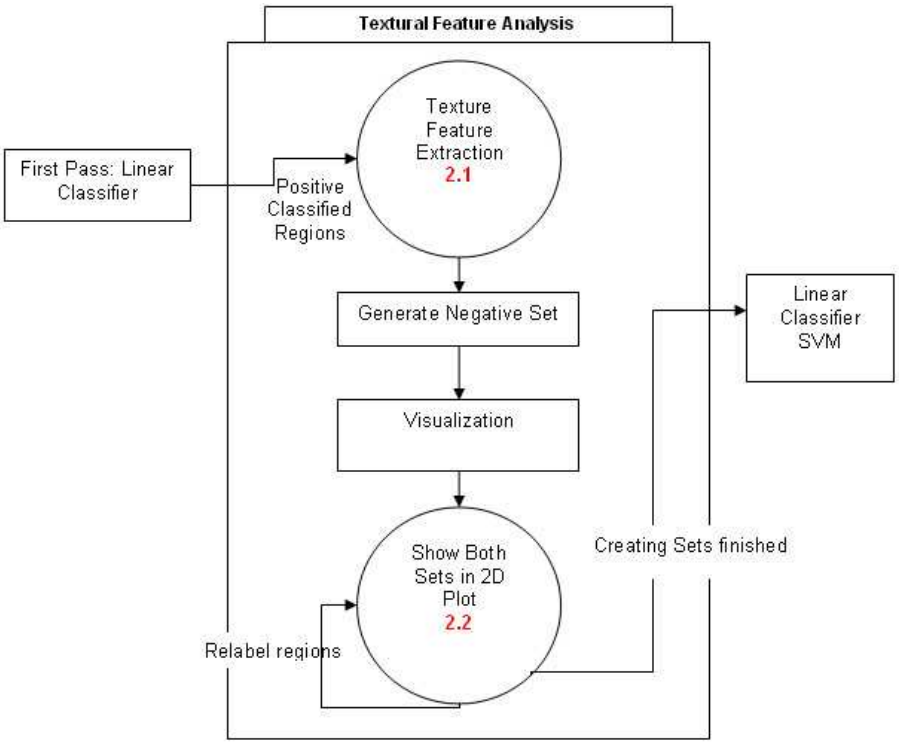


Figure 3.3: Flow diagram of the second step.

3.2 Overview of Software

We designed and implemented a basic software for the purposes of this study. A screen shot from the user interface of this software is shown in Figure 3.4. The software is implemented in C++ programming language while the user interface is developed using Qt. Qt is a comprehensive C++ application development framework.

User can perform the following operations on an image through our interface;

- Desired image can be opened.
- The whole image and its reference image are shown to the user. The reference image has a lower resolution copy of the original image of size 120×120 pixels. The reference image gives user the option of navigating through the original image, by clicking a corresponding area on reference image instead of searching in the original one. By clicking on a segment in reference image will show you the associated image region in the original image at the same time with the original resolution.
- Regions and labels can be saved as a project file for further operations. The

project will enable the user to work on previously created datasets in addition s/he can update these sets accordingly.

- The positive set can be formed by the user through our interface by visually looking regions both in 2D plot and their place in the whole image. The selection is made by dragging the selection window to the desired region on the original image, the positive set formed by this way is shown on the right hand side of our interface.
- positive set can be updated by adding new regions or removing existing ones.
- User is able to see a 2D plot the feature sets of labeled positive regions and of all regions. User is able to observe the corresponding image region by selecting a point on the plot. This option gives user the opportunity to observe visually and accordingly modify the members of positive and negative sets.

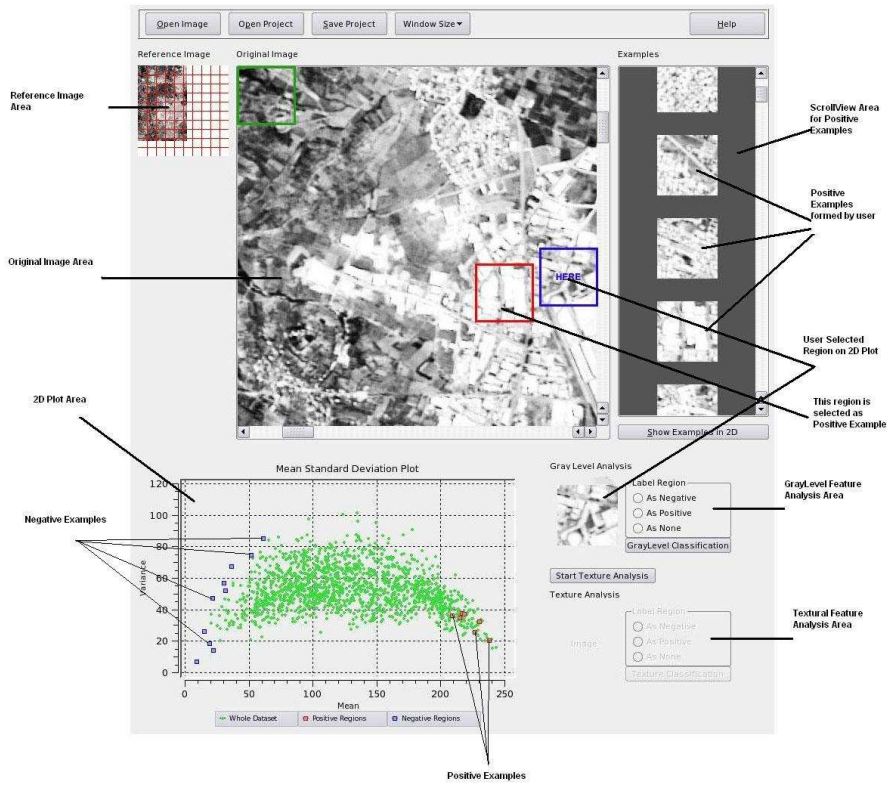


Figure 3.4: Interface of developed software.

The functions of each object on interface is briefly explained as follows:

- **Open Image :** This button is used to open an high-resolution image in any format e.g .jpeg, .bmp, etc. Reference image shown in Figure 3.4 is a down-sampled(120 x 120) copy of original image.
- **Save Project :** Used for saving the project for further operations.
- **Open Project :** User can continue working on previously saved projects by using this button.
- **Window Size :** This is required when the size of the cartographic object e.g bridge, road is not fit into the selection window. User can increase or decrease the size of window to fit the object into. As a result the size of each element in positive example set can vary accordingly.
- **Help :** This button provides a help document of software to make sure the user of efficient use.
- **Show Examples in 2D :** The features of examples in 2D plot located on the bottom of interface are shown to the user by clicking to this button.
- **Spectral or Linear Analysis :** This area is used to work on elements of both positive and negative example sets by changing their labels or adding new regions to these sets.
- **Textural Analysis :** This area is used to work on elements of both positive and negative example sets by changing their labels or adding new regions to these sets.

3.3 Employed Algorithms

This section is going to present the algorithm employed for both steps of system. The notations that we used in the algorithms are as follows;

- Λ : whole set,
- ρ : positive set,
- η : negative set

- $\bar{f}$: a feature set,
- L : boundary curve set,
- $d()$: distance function,
- h : histogram,
- μ : mean,
- σ : standard deviation,

3.3.1 Spectral Feature Analysis

In the first step, our aim is the application of linear classification methods such as support vector machines (SVM) in order to reduce the number of candidate regions.

User indicates some m reliable example regions that form the positive set ρ. Positive feature set is created by calculating gray level mean μ and standard deviation σ of each element in ρ using equations in Section 2.1. The feature set for ρ becomes

$$[\mu_1, \sigma_1, \ldots, \mu_m, \sigma_m] \tag{3.1}$$

For classification purposes, we need sample negative images. Negative set η is generated by labeling the farthest t regions of Λ to the mean of set ρ in terms of Euclidean distance shown in Equation 3.2.

$$d(i) = \sqrt{(\mu_i - \mu_\rho)^2 + (\sigma_i - \sigma_\rho)^2} \qquad i = 1, \ldots, ||\Lambda|| \tag{3.2}$$

where μ_ρ and σ_ρ is an average μ and σ of set ρ.

The complete steps of algorithm for graylevel feature extraction are shown in 2.

We have used SVM as a linear classifier [26].

3.3.2 Linear Feature Analysis

We implemented another technique which is linear segment extraction for the first step of our analysis. Linear segment extraction is a 3-stage approaches defined below.

User indicates some m reliable example regions that form the positive set ρ. The Canny edge detector is applied to each element of set ρ. Stage 2 is to form a boundary curve, the gradient or strength measure is a typical approach for marking each pixel

Algorithm 2 Spectral Analysis

1: **for** each $i \in \rho$ **do**
2: calculate μ_i and σ_i in equation 2.1
3: $\mu_\rho = \frac{1}{m}\sum_{i=1}^{m}\mu_i$ and $\sigma_\rho = \frac{1}{m}\sum_{i=1}^{m}\sigma_i$
4: **for** each $i \in \Lambda$ **do**
5: calculate μ_i and σ_i
6: calculate Euclidean distance $d(i)$ to $[\mu_\rho, \sigma_\rho]$
7: generate negative set η by t candidate regions that have largest $d()$
8: visualize ρ and η in 2D plot using μ and σ
9: modify ρ and η by user interaction
10: for final positive set ρ apply lines 1-2
11: **for** each $i \in \eta$ **do**
12: calculate μ_i and σ_i
13: apply a linear classifier on Λ using ρ, η and corresponding gray level features.

as a result of edge detection method. Canny's hysteresis thresholding is used for this purpose.

The last stage is to approximate the segments by polygonal approximation, it is used to obtain a more compact description from the linked representation of edge pixels. Given a curve sequence $C = < x_1, y_1 >, \cdots, < x_n, y_n >$, iterative end-point fit algorithm is applied to segment C into subsequences until the error condition is satisfied.

The feature set for positives is the histogram of lengths of segments. The histogram is nonuniformly quantized according to length of linear segments. The intervals of histogram are defined by taking into account various parameters such as the size of regions, the distance threshold and the Canny edge detector parameters. By defining the interval of histogram, the linear segments with small changes are mapped into the same interval instead of using different interval values.

The mean interval values $\mu_{\rho_{h_k}}$ of histogram is calculated as in Equation 3.3. It is used to generate a negative set where h and k correspond to histogram and the number of intervals, respectively. The Euclidean distance is calculated for each interval using an Equation 3.4.

$$\mu_{h_{\rho_k}} = \frac{1}{||\rho||} \sum_{i=1}^{||\rho||} h\rho_k i \tag{3.3}$$

where $k = 1, \cdots, K$.

$$d(\mu_{h_\rho}, h_\Lambda i) = \sqrt{(\mu_{h_{\rho_1}} - h_{\Lambda_1} i)^2 + \cdots + (\mu_{h_{\rho_K}} - h_{\Lambda_K} i)^2} \tag{3.4}$$

The steps of algorithm for linear feature analysis is shown in Algorithm 3.

3.3.3 Textural Feature Analysis

The objective of second step is to apply texture as an image feature for classification. Therefore, a texture feature representation based on Gabor filters 2.1.3 is applied. After having applied linear classification in first step, the resultant positive regions are showed through our interface and the boundary between the two sets is also drawn in 2D shown in Figure. The positive and negative results are marked in different colors for discrimination purpose on our plot.

For this step, image regions that are labeled as negative are ignored since these regions are irrelevant. We start studying on positive results of the first pass. The positive set ρ formed in the first pass is kept. Filter responses of images to the Gabor filter dictionary are computed by taking Fourier transform of the images and multiplying with the filters in this domain. A feature vector is formed using the means and standard deviations of the responses to the filters at the selected scale and orientations. In our experiments, different scales m and orientations n are used forming a feature vector of size $m \times n$.

We use the distance measure in Equation 3.5 to calculate the distance $d(i, j)$ between set ρ and the whole set Λ.

$$\begin{aligned} & d(i,j) = \sum_m \sum_n d_{mn}(i,j) \\ & where \\ & d_{mn}(i,j) = \left| \frac{\mu_{mn}^{(i)} - \mu_{mn}^{(j)}}{\alpha(\mu_{mn})} \right| + \left| \frac{\sigma_{mn}^{(i)} - \sigma_{mn}^{(j)}}{\alpha(\sigma_{mn})} \right| \end{aligned} \tag{3.5}$$

where $\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ are the standard deviations of the respective features over entire set, and are used to normalize the individual feature components. The negative set η is formed by taking the farthest t images to ρ.

Algorithm 3 Linear Feature Analysis

1: K is the number of interval
2: **for** each $i \in \rho$ **do**
3: apply Canny edge detection algorithm 1 to image i which results an image $C_\rho i$
4: apply hysteresis thresholding to $C_\rho i$ to form boundary curve set $L_\rho i$.
5: apply iterative end-point fit algorithm on set $L_\rho i$ using distance threshold
6: **for** each $j \in L_\rho i$ **do**
7: calculate the length of fragment j
8: update histogram $h_\rho i$ of image i using j
9: feature vector $\bar{f_\rho}$ becomes $[h_\rho 1 \cdots h_\rho ||\rho||]^t$
10: **for** each $j \in K$ **do**
11: $\mu_{h_{\rho_j}} = \frac{1}{||\rho||} \sum_{i=1}^{||\rho||} h_{\rho_j} i$
12: **for** each $i \in \Lambda$ **do**
13: apply Canny edge detection algorithm to image i shown in 1 to have a resultant image $C_\Lambda i$
14: apply hysteresis thresholding to $C_\Lambda i$ to form boundary curve set $L_\Lambda i$.
15: apply iterative end-point fit algorithm on set $L_\Lambda i$ using distance threshold
16: **for** each $j \in L_\Lambda i$ **do**
17: calculate the length of fragment j
18: update histogram $h_\Lambda i$ of image i using j
19: feature vector $\bar{f_\Lambda}$ becomes $[h_{\Lambda_1} \cdots h_{\Lambda_{||\Lambda||}}]^t$
20: **for** each $j \in K$ **do**
21: $\mu_{h_{\Lambda_j}} = \frac{1}{||\Lambda||} \sum_{i=1}^{||\Lambda||} h_{\Lambda_j} i$
22: **for** $i \in \Lambda$ **do**
23: Calculate distance $d(\mu_{h_\rho}, h_\Lambda i)$ using Equation 3.4
24: Construct negative set η by labeling farthest t candidate regions that have largest distance $d()$ in line 23
25: Apply Multi Dimensional Scaling to feature sets of both η and ρ and visualize the projections in 2D
26: Modify ρ and η by user interaction
27: Apply lines 1-9 for redefined positive set ρ
28: Apply lines 1-9 for redefined negative set η
29: Apply Support Vector Classifier onto Λ using $\bar{f_\rho}$ and $\bar{f_\eta}$

For visualization purposes, the feature sets of ρ and η are mapped into 2 dimensional space using Multidimensional scaling. User can modify both sets by clicking on a point in plot and examining visually the corresponding image region. When the user is satisfied, Support vector classifier finds the boundary between two sets and apply the classification on the whole image regions.

The complete steps of algorithm for second pass is shown in Algorithm 4.

3.3.4 Structural Feature Analysis

There are two modules in structural feature analysis system [3]. The preprocessing module performs the decomposition of an object in positive image regions into basic shapes. The topology and spatial information can be preserved while obtaining a decomposed image. Two different decomposition in parallel is proposed one for circles and rectangles and the other for circles and segments. In the second module, decomposed images created in the first module is transformed into ARGs (Attributed Relational Graphs). The ARGs are preferred to represent structural features of image regions. The representation rules are specified as the characteristics of geometric shapes are stored in the vertices, the characteristics of connections are stored in edges. The model is generated using ARGs, the generated model is mesh bounded by the MaxCSg (Maximal Common Subgraph) and MinCSg (Minimal Common Supergraph) that is the set of all graphs G such that $G \subseteq MinCSg$ and $G \supseteq MaxCSg$. The ARG of image regions in whole set Λ is generated. The ARG of an image region is in between $MinCSg$ and $MaxCSg$ then corresponding image region is classified as positive.

Algorithm 4 Texture Analysis using Gabor Filter

1: **for** each $i \in \rho$ **do**
2: **for** $a = 1$ to m **do**
3: **for** $b = 1$ to n **do**
4: Calculate $g_\rho(i, a, b)$ which is the Gabor filter output at scale a and orientation b
5: **for** $a = 1$ to m **do**
6: **for** $b = 1$ to n **do**
7: **for** each $i \in \rho$ **do**
8: Calculate $\mu_\rho(i, a, b)$ and $\sigma_\rho(i, a, b)$
9: A feature vector $\bar{f}_\rho$ of ρ is constructed using $\mu_\rho(i, a, b)$ and $\sigma_\rho(i, a, b)$.
10: μ_ρ is an average of mean features in $\bar{f}_\rho$ for each scale m and orientation n.
11: σ_ρ is an average of standard deviation features in $\bar{f}_\rho$ for each scale m and orientation n.
12: **for** each $i \in \Lambda$ **do**
13: **for** $a = 1$ to m **do**
14: **for** $b = 1$ to n **do**
15: Calculate $g_\Lambda(i, a, b)$ which is the Gabor filter output at scale a and orientation b
16: **for** $a = 1$ to m **do**
17: **for** $b = 1$ to n **do**
18: **for** $i \in \Lambda$ **do**
19: Calculate $\mu_\Lambda(i, a, b)$ and $\sigma_\Lambda(i, a, b)$
20: A feature vector $\bar{f}_\Lambda$ of Λ is constructed using $\mu_\Lambda(i, a, b)$ and $\sigma_\Lambda(i, a, b)$.
21: **for** $i \in \Lambda$ **do**
22: Calculate distance $d((\mu_\rho, \sigma_\rho), \bar{f}_\Lambda(i))$.
23: Construct negative set η by labeling the farthest t candidate regions as negative according to distance $d()$ in line 22
24: Apply Multi Dimensional Scaling to feature sets of both η and ρ and visualize the projections in 2D
25: Modify ρ and η by user interaction
26: Apply lines 1-4 for redefined positive set ρ
27: Apply lines 1-4 for redefined negative set η
28: Apply Support Vector Classifier onto Λ using $\bar{f}_\rho$ and $\bar{f}_\eta$

CHAPTER 4

EXPERIMENTS AND RESULTS

The proposed man-made cartographic object extraction process is composed of two steps;

- First step : spectral or linear feature analysis.
- Second step : textural feature analysis.

We tested different parameter values for each image feature analysis technique. The overall process parameters for image analysis steps which affects the results is explained below.

- The spectral feature extraction technique parameters:
  - the mean of pixel values μ ,
  - the standard deviation of pixels σ.
- The line segment extraction process parameters;
  1. Canny edge detector and edge linking parameters:
     - the size of Gaussian filter σ,
     - high hysteresis threshold **HTresh**,
     - low hysteresis threshold **LTresh**.
  2. polygonal approximation parameters:
     - the line length threshold of line segments **h**,
     - the distance threshold of point to line distance **d**.
- The textural feature extraction process parameters:
  - the frequency components of filter **m**,

- the directional components of filter **n**,
- filter mask dimension size **side**,
- high center frequency U_h
- low center frequency U_l.

For our experiments, we have used satellite images of size 4000×2400 pixels and 4800×3600 pixels shown in Figure 4.1 and Figure 4.2 which are a single band, grayscale SPOT Pan image. The images are divided into nonoverlapping subimages of size 80×80 pixels, thus creating a set of 1500 and 2700 image regions respectively. We have subjectively labeled each image region into one of the two classes, positive and negative. Therefore, we assume that we have the groundtruth information to assess the accuracy. The number of positive labeled is 16 for road set, 231 for urban area set and 581 for home set.

We created two test sets from the whole set Λ; image regions containing roads, image regions containing urban areas and image regions containing homes. There are two forms of experiments performed on these sets: test **A**; spectral followed by textural feature analysis, test **B**; linear followed by textural feature analysis.

For the classification, LIBSVM is used as a computational tool. The quality of SVM models depends on a proper setting of SVM meta-parameters. Therefore, the main issue in this study is how to set these parameter values for a given dataset. Experiments yield that RBF kernel gives better results than linear and polynomial kernels. There are two parameters while using RBF kernels: C and γ. It is not known beforehand which C and γ are the best for our problem. Consequently some kind of model selection must be done. Therefore we employed a code written in Python which is a model selection tool. It uses cross validation (CV) technique to estimate the accuracy of each parameter combination in the specified range and helps us to decide the best parameters for our problem.

The explanations of terms that we used mostly in our experiments are as follows;

True Positives : The positive instances that were reported as positive.

False Positives : The negative instances that were erroneously reported as positive.

True Negatives : The negative instances that were reported as negative.

False Negatives : The positive instances that were erroneously reported as negative.

Local Accuracy : The local accuracy is the proportion of number of positives to the

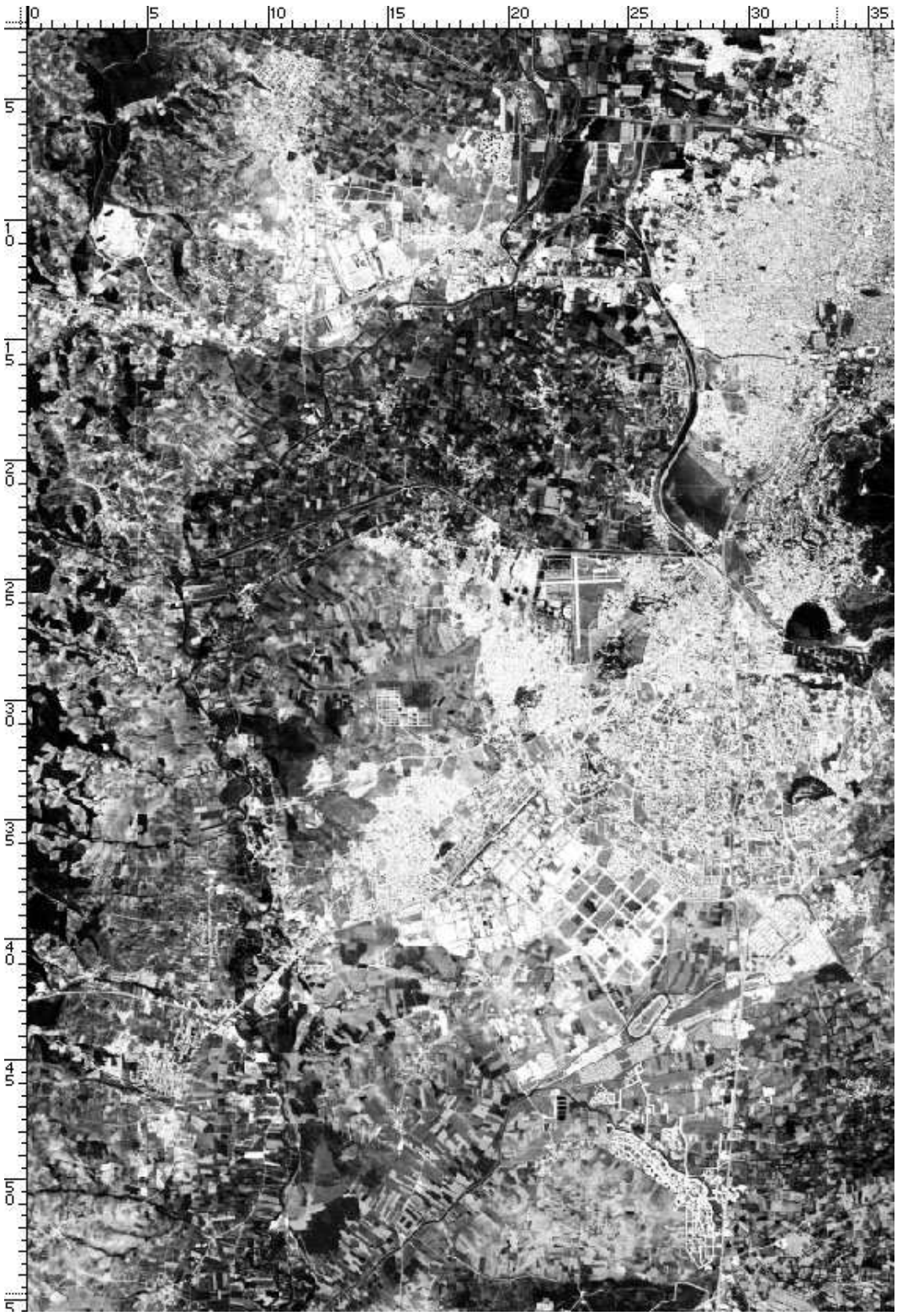


Figure 4.1: Single band, grayscale Image (SPOT Pan) of size (4000 x 2400) with spatial resolution 10m.

Figure 4.2: Single band, grayscale Image (SPOT Pan) of size (4800 x 3600) with spatial resolution 10m.

number of input set. It is calculated using an Equation 4.1.

$$local\ accuracy = \frac{\#\ of\ true\ positives\ +\ \#of\ true\ negatives}{\#\ of\ elements\ in\ input}\ \times\ 100 \tag{4.1}$$

Overall Accuracy : The total accuracy is the proportion of number of positives for each set to the number of whole set. It is calculated using an Equation 4.2.

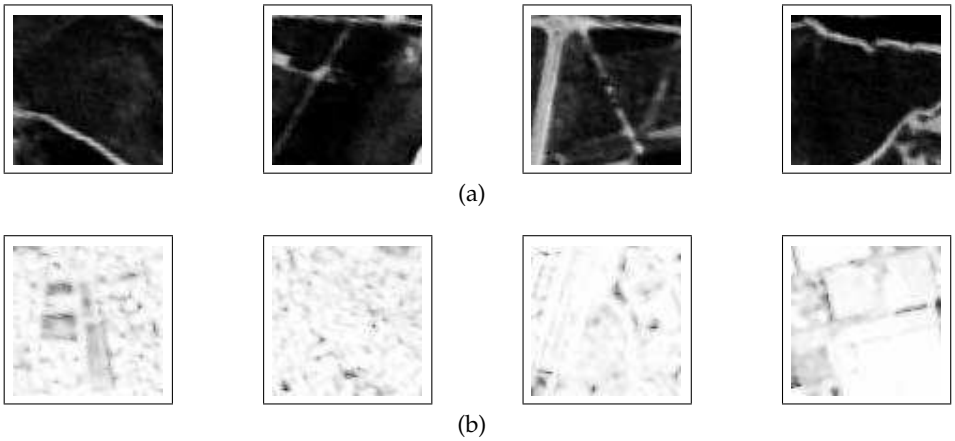
$$total\ accuracy = local\ accuracy \times \left(1\ +\ \frac{\#\ of\ true\ positives\ +\ \#of\ true\ negatives}{\#\ of\ elements\ in\ input}\right) \tag{4.2}$$

Local accuracy gives the accuracy at an individual step in terms of its input.

MDS is used to visualize our high dimensional data e.g. linear features of size $n \times 10$ and textural features of size $n \times 24$ where n is the number of regions.

4.1 Test A : Spectral and Textural Feature Analysis Results

Test **A** is performed on images in Figure 4.3, Figure 4.4 and Figure 4.5. Algorithm 2 is followed step by step. The positive set is formed by the user through our interface. The number of positives used in our test sets is 10. We generate a negative set by labeling the farthest 10 regions, these may be irrelevant to positive set but the user can update elements of negative set. The negative set is formed using spectral analysis distance metric written in Equation 3.2.

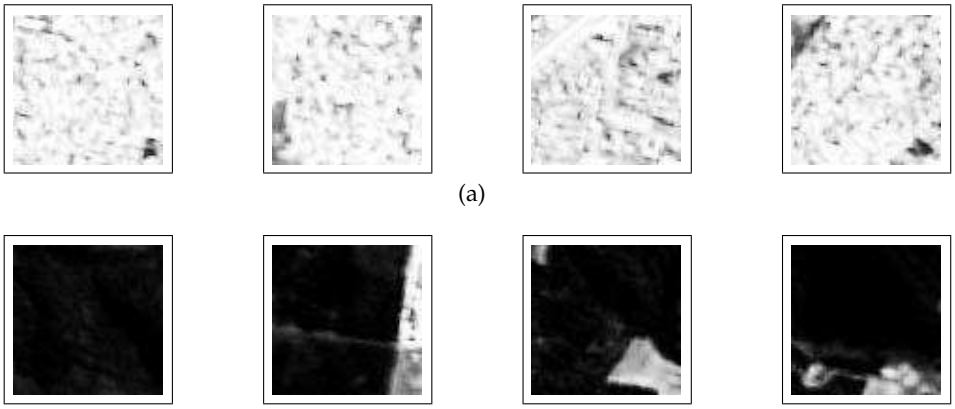


(a)

(b)

Figure 4.3: Spectral Feature Analysis : Some example regions regions in road set. (a) Positive regions, (b) generated negative regions.

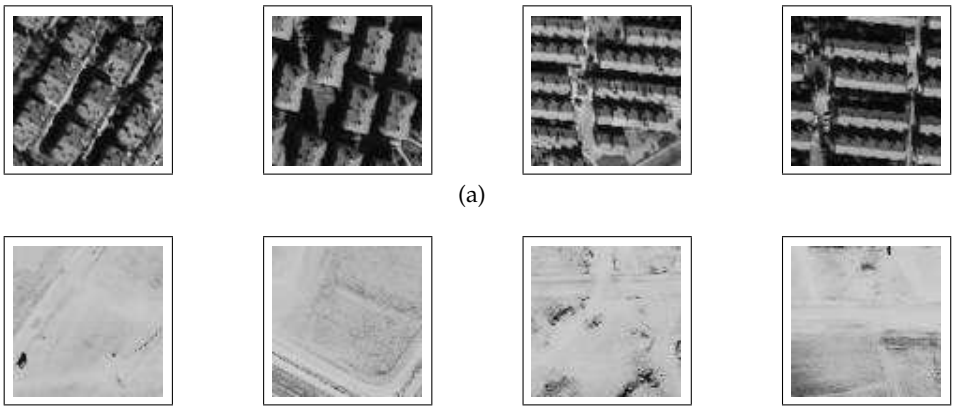
All test set's 2-dimensional mean standard deviation plot in Figure 4.6,Figure 4.7 and Figure 4.8 are shown to the user for relabeling the elements of positive and neg-



(a)

(b)

Figure 4.4: Spectral Feature Analysis : Some example regions regions in urban area set. (a) positive regions, (b) generated negative regions.



(a)

(b)

Figure 4.5: Spectral Feature Analysis : Some example regions regions in home set. (a) positive regions, (b) generated negative regions.

ative sets. It shows the distribution of whole set Λ including positive ρ and negative η sets.

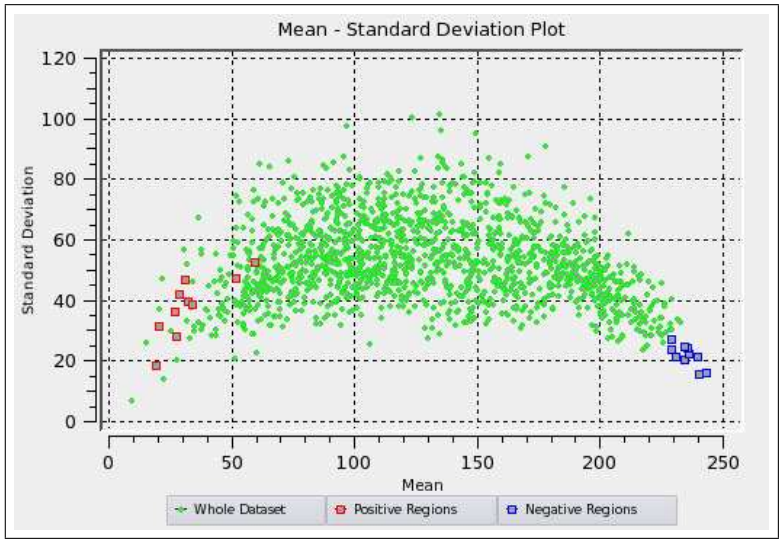


Figure 4.6: Spectral Feature Analysis : The 2-dimensional plot of whole road set including positives and negatives.

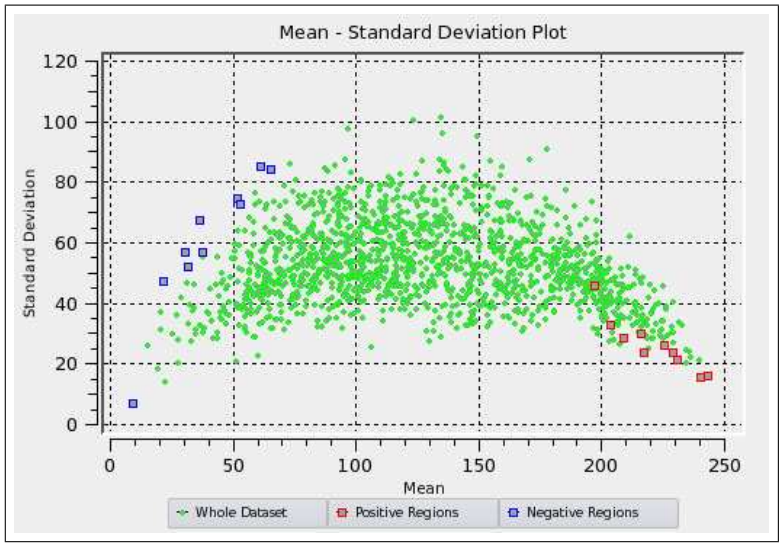


Figure 4.7: Spectral Feature Analysis : The 2-dimensional plot of whole urban area set including positives and negatives.

Test A is performed on our test sets which are road set, urban area set and home set. The positive set is formed by the user. Figure 4.3, Figure 4.4 and Figure 4.5 show negative and positive sets.

For our all three sets, Table 4.1, Table 4.2 and Table 4.3 show generated model parameters for a classifier respectively. It shows the parameters that maximizes true

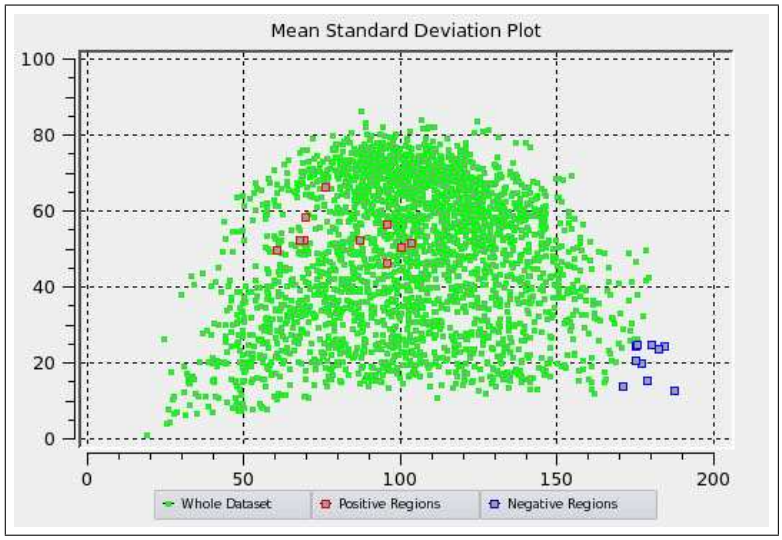


Figure 4.8: Spectral Feature Analysis : The 2-dimensional plot of whole home set including positives and negatives.

retrieval performance of classifier.

Table 4.1: Spectral Feature Analysis : The model parameters of road set.

	RBF Parameters
Type of SVM	C-SVC
Gamma	0.0078125
C	0.03125
# of Class	2
# of SVs	20
ρ	-0.00046177
label	1 -1

Table 4.2: Spectral Feature Analysis : The model parameters of urban area set.

	RBF Parameters
Type of SVM	C-SVC
Gamma	0.0078125
C	0.03125
# of Class	2
# of SVs	20
ρ	0.000998657
label	1 -1

The predicted labels with local classifier accuracy for all three test sets is shown in Table 4.4, Table 4.5 and Table 4.6.

The first step results show that positive labeled regions are classified correctly

Table 4.3: Spectral Feature Analysis : The model parameters of home set.

	RBF Parameters
Type of SVM	C-SVC
Gamma	0.0078125
C	0.03125
# of Class	2
# of SVs	20
ρ	0.000436745
label	1 -1

Table 4.4: Spectral Feature Analysis : The classification results of road set.

	RBF
True Positives	16
False Positives	879
True Negatives	605
False Negatives	0
Local Accuracy :	41.4%

Table 4.5: Spectral Feature Analysis : The classification results of urban area set.

	RBF
True Positives	231
False Positives	136
True Negatives	1133
False Negatives	0
Local Accuracy :	90.7%

Table 4.6: Spectral Feature Analysis : The classification results of home set.

	RBF
True Positives	581
False Positives	447
True Negatives	1672
False Negatives	0
Local Accuracy :	83.4%

while most of the negatives are eliminated. The image regions classified as positive in the first is fed to the second step that is textural analysis.

For Gabor filters ,since there is no best parameter set which gives best results, we tried different combination of parameters that can affect the performance of classifier considerably. Table 4.7 shows the parameters that we used for analysis. For low and high frequency parameters, $U_l = 0.1$ and $U_h = 0.6$ values are used. Same textural feature analysis process is followed for urban area set and home set. Table 4.8 and Table 4.9 shows the predicted labels for with classification accuracy.

Table 4.7: Textural Feature Analysis : The classification results for road set. RBF kernel is employed. Gabor filter parameters S : scale, K : orientation, Side : mask size.

Exp. #	True (+)'s	False (+)'s	True (-)'s	False (-)'s	Local Accuracy	Overall Accuracy
S = 3; K = 4; Side = 20	16	281	598	0	68.6%	81.2%
S = 3; K = 4; Side = 80	16	302	577	0	66.2%	79.8%
S = 3; K = 6; Side = 20	16	348	531	0	61.1%	76.8%
S = 3; K = 6; Side = 80	16	472	407	0	47.2%	68.5%
S = 4; K = 4; Side = 20	16	302	577	0	66.2%	79.8%
S = 4; K = 4; Side = 80	16	413	466	0	53.8%	72.4%
S = 4; K = 6; Side = 20	16	378	501	0	57.7%	74.8%
S = 4; K = 6; Side = 80	16	378	501	0	57.7%	74.8%

We have used multidimensional scaling to visualize higher dimensional results. Figure 4.9 and Figure 4.10 shows Gabor filtered outputs of the positives set; results of first step. Originally the dimension of Gabor output was $n \times 24$ where n is the

Table 4.8: Textural Feature Analysis : The classification results for urban area set. RBF kernel is employed. Gabor filter parameters S : scale, K : orientation, Side : mask size.

	True (+)'s	False (+)'s	True (-)'s	False (-)'s	Local Accuracy	Overall Accuracy
S = 3; K = 4; Side = 20	231	52	84	0	89.2%	96.5%
S = 3; K = 4; Side = 80	225	68	68	6	84.9%	95.0%
S = 3; K = 6; Side = 20	210	90	46	21	76.8%	92.6%
S = 3; K = 6; Side = 80	192	101	35	39	70.9%	90.6%
S = 4; K = 4; Side = 20	221	73	63	11	82.7%	94.4%
S = 4; K = 4; Side = 80	218	90	46	13	78.6%	93.1%
S = 4; K = 6; Side = 20	203	87	49	28	76.1%	92.3%
S = 4; K = 6; Side = 80	189	109	27	42	69.6%	89.9%

Table 4.9: Textural Feature Analysis : The classification results for home set. RBF kernel is employed. Gabor filter parameters S : scale, K : orientation, Side : mask size.

	True (+)'s	False (+)'s	True (-)'s	False (-)'s	Local Accuracy	Overall Accuracy
S = 3; K = 4; Side = 20	581	223	224	0	78.3%	91.8%
S = 3; K = 4; Side = 80	564	287	160	17	70.4%	88.7%
S = 3; K = 6; Side = 20	566	262	185	15	73.0%	89.7%
S = 3; K = 6; Side = 80	558	287	160	23	69.8%	88.5%
S = 4; K = 4; Side = 20	566	245	202	15	74.7%	89.7%
S = 4; K = 4; Side = 80	560	234	213	21	75.1%	90.5%
S = 4; K = 6; Side = 20	566	226	221	25	76.5%	90.7%
S = 4; K = 6; Side = 80	543	212	235	38	75.6%	90.7%

number of positives.

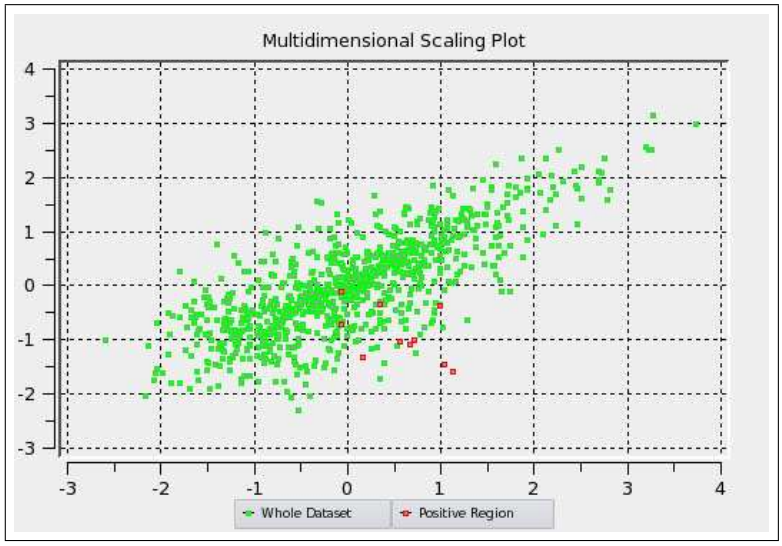


Figure 4.9: Multidimensional Scaling : The 2-dimensional plot of Gabor filtered outputs of road area set including positives.

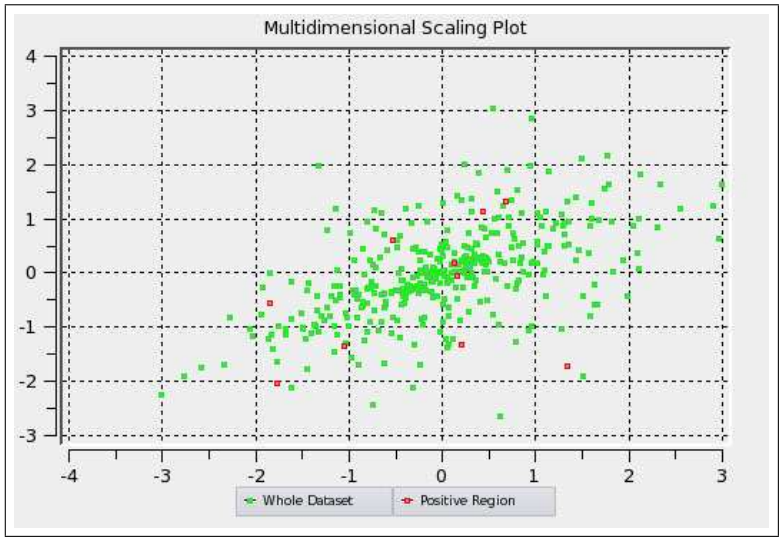


Figure 4.10: Multidimensional Scaling : The 2-dimensional plot of Gabor filtered outputs of urban area set including positives.

Table 4.10: Statistical measures for Test **A**.

	Sensitivity %	Recall %	F-score %	Specificity %
Test **A**				
Set 1	53	100	69	81
Set 2	81	100	89	89
Set 3	72	100	83	89

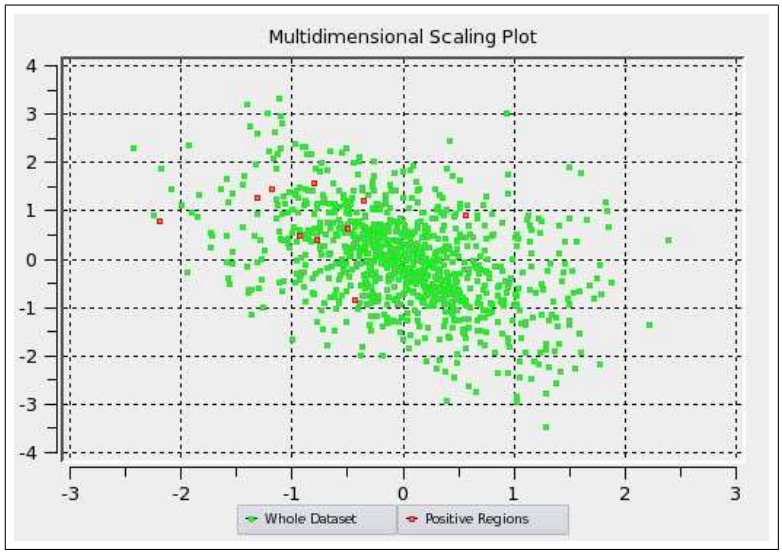


Figure 4.11: Multidimensional Scaling : The 2-dimensional plot of Gabor filtered outputs of home set including positives.

4.2 Test B : Linear and Textural Feature Analysis Results

Test B is performed on the same images that we worked on test A shown in Figure 4.3, Figure 4.4 and Figure 4.5. The approach is to use linear features for the first step of analysis instead of spectral analysis. It is influenced by the structure of the man-made cartographic objects. The aim is to eliminate most of the true negatives. The main idea and the applied algorithm are explained in Section 3.3.2.

We have employed different parameters for testing. The list of parameters that we used for road, urban area and home sets are shown in Table 4.11, Table 4.12 and Table 4.13.

Table 4.11: Linear Feature Analysis : The linear feature parameters for road set.

Experiment #	Sigma	Low Threshold	High Threshold	Distance
1	1.2	0.2	0.9	10
2	2.0	0.2	0.5	10
3	1.0	0.5	0.9	5
4	1.5	0.3	0.6	2

We have formed our training set by labeling 10 images as positive and accordingly generating 10 negative images. The images for all test sets are shown in Figure 4.12, Figure 4.13 and Figure 4.14.

Before classifying our test set, we look for best RBF kernel parameters to have

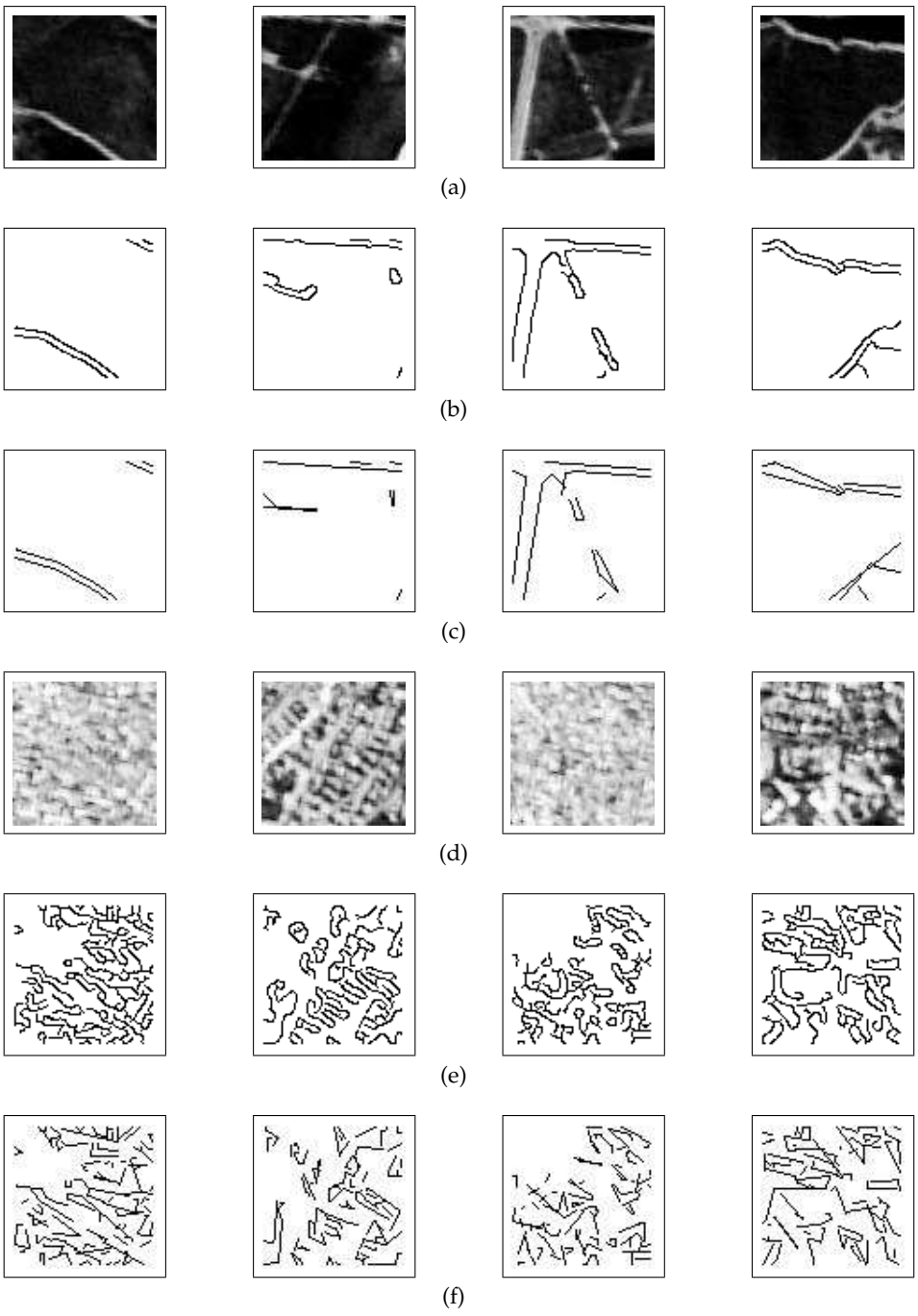


Figure 4.12: Linear Feature Analysis : Regions used for road set. (a) positive regions, (b) edge detection result regions hysteresis thresholded , (c) line fitted edge image, (d) generated negative regions, (e) edge detection result regions hysteresis thresholded , (f) line fitted edge image.

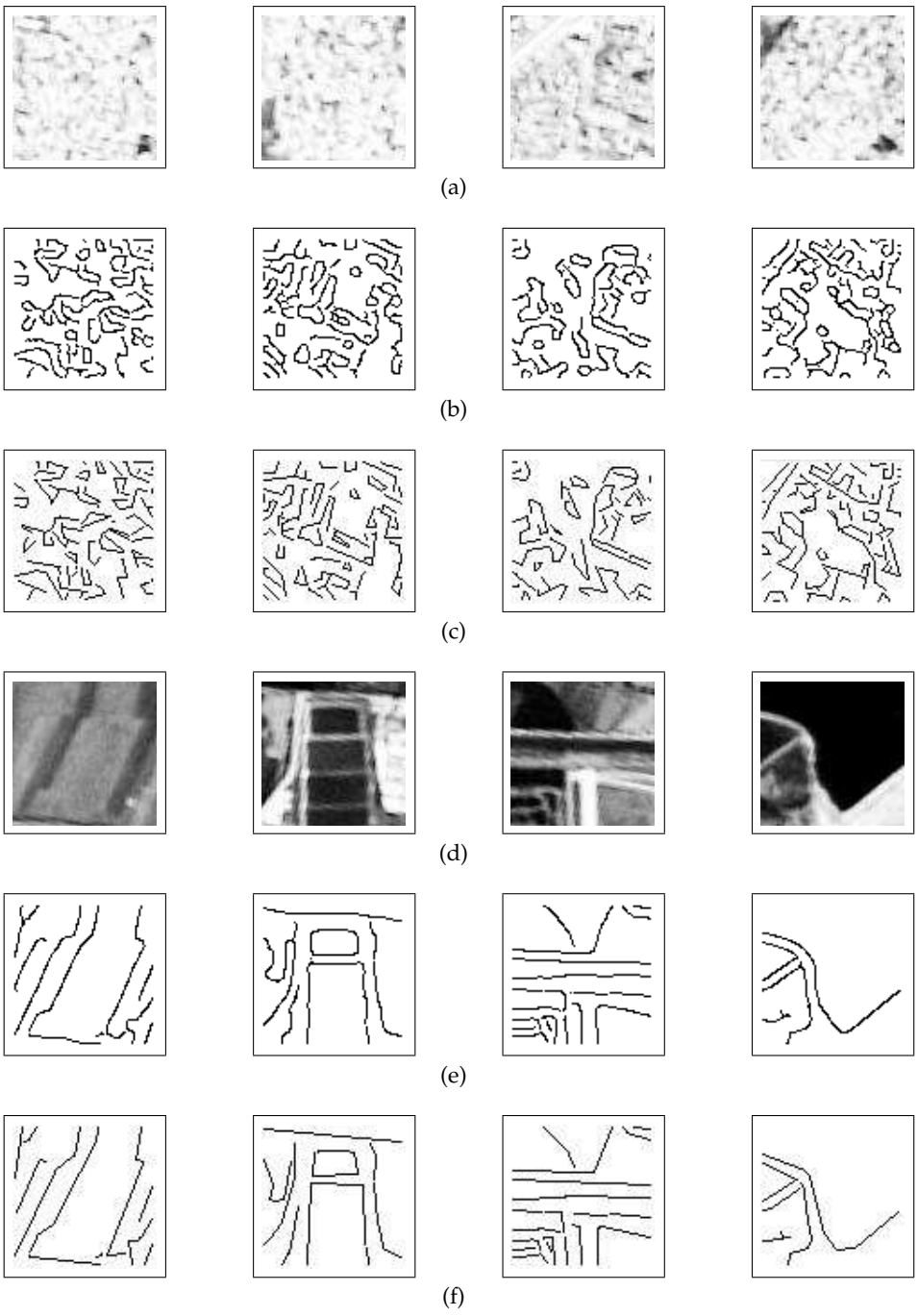


Figure 4.13: Linear Feature Analysis : Regions used for urban area set. (a) positive regions, (b) edge detection result regions hysteresis thresholded , (c) line fitted edge image, (d) generated negative regions, (e) edge detection result regions hysteresis thresholded , (f) line fitted edge image.

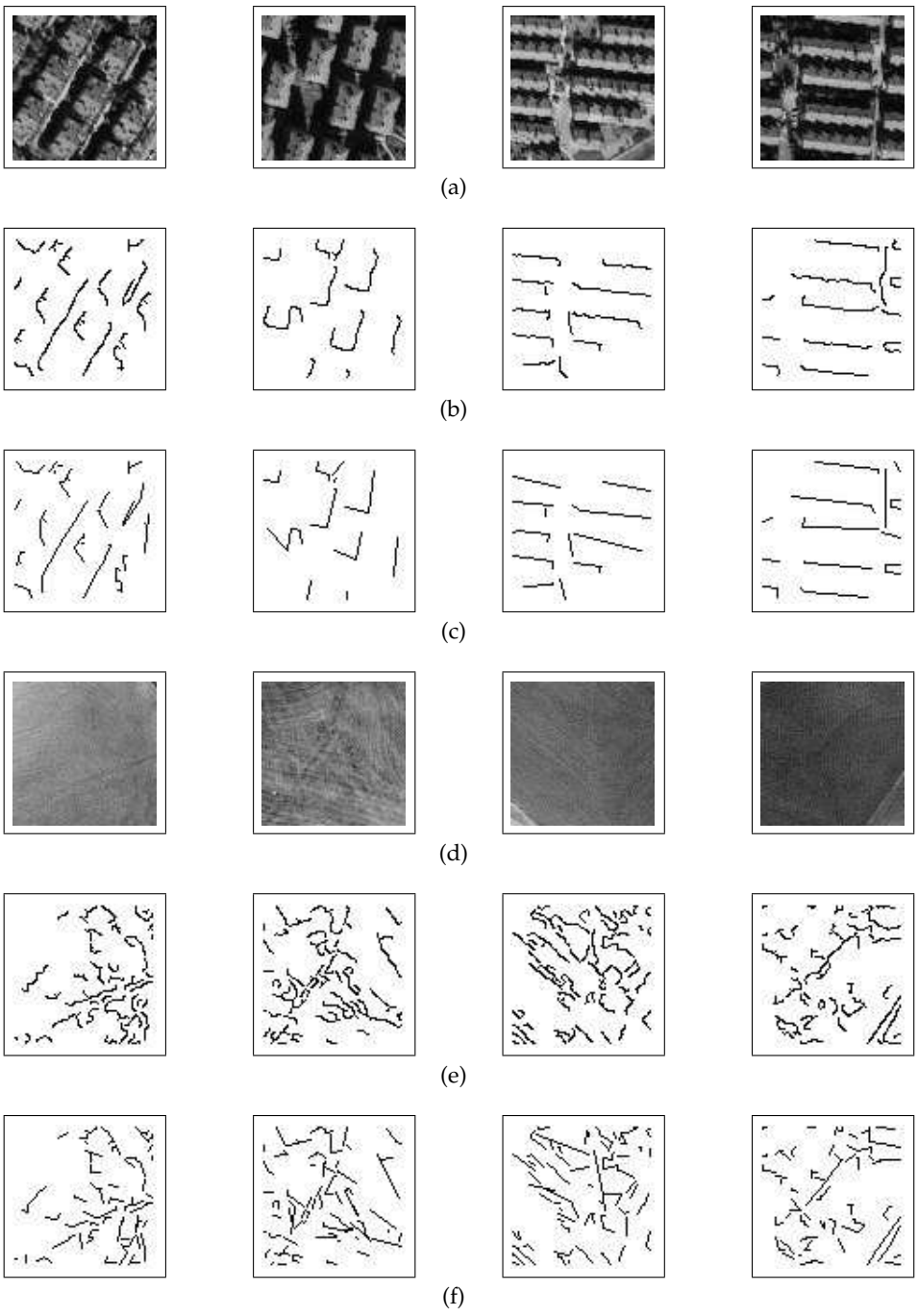


Figure 4.14: Linear Feature Analysis : Regions used for home set. (a) positive regions, (b) edge detection result regions hysteresis thresholded , (c) line fitted edge image, (d) generated negative regions, (e) edge detection result regions hysteresis thresholded , (f) line fitted edge image.

Table 4.12: Linear Feature Analysis : The linear feature parameters for urban area set.

Exp. #	Sigma	Low Threshold	High Threshold	Distance
1	1.2	0.2	0.6	5
2	2.0	0.5	0.9	3
3	2.0	0.2	0.5	3
4	1.0	0.2	0.9	10

Table 4.13: Linear Feature Analysis : The linear feature parameters for home set.

Exp. #	Sigma	Low Threshold	High Threshold	Distance
1	1.2	0.2	0.6	10
2	2.0	0.5	0.9	5
3	1.5	0.2	0.9	5
4	1.0	0.5	0.9	10

good classification accuracy. Table 4.14, Table 4.15 and Table 4.16 show the model files that we used during linear feature analysis step for both test sets.

Table 4.14: Linear Feature Analysis : The model file used for road set.

Exp. #	Kernel	Gamma	C	# of Classes	# of SVs	ρ
1	RBF	0.78125	0.03125	2	33	-1.00052
2	RBF	0.125	2.0	2	11	0.182485
3	RBF	0.0078125	0.03125	2	20	-0.00099
4	RBF	0.5	0.03125	2	20	0.000657

Table 4.15: Linear Feature Analysis : The model file used for urban area set.

Exp. #	Kernel	Gamma	C	# of Classes	# of SVs	ρ
1	RBF	0.0078125	0.03125	2	20	00205572
2	RBF	0.5	0.03125	2	20	0.03347
3	RBF	0.0078125	0.03125	2	20	0.00231229
4	RBF	0.5	2.0	2	20	0.411611

The linear classification results for all test sets are shown in Table 4.17, Table 4.18

Table 4.16: Linear Feature Analysis : The model file used for home set.

Exp. #	Kernel	Gamma	C	# of Classes	# of SVs	ρ
1	RBF	0.225	0.03125	2	27	-0.08352
2	RBF	0.125	1.2	2	10	-0.82485
3	RBF	0.0078125	0.03125	2	20	-0.00099
4	RBF	0.5	0.03125	2	15	0.011657

and Table 4.19.

Table 4.17: Linear Feature Analysis : Classification accuracy for road set.

Exp. #	Kernel	True (+)'s s	False (+)'s	True (-)'s	False (-)'s	Local Accuracy
1	RBF	16	69	1412	0	95.5%
2	RBF	16	283	1202	0	81.1%
3	RBF	16	283	1202	0	81.1%
4	RBF	16	316	1168	0	78.9%

Table 4.18: Linear Feature Analysis : Classification accuracy for urban area set.

Exp. #	Kernel	True (+)'s s	False (+)'s	True (-)'s	False (-)'s	Local Accuracy
1	RBF	173	251	1018	58	79.4%
2	RBF	141	522	747	90	59.2%
3	RBF	206	490	779	25	65.6%
4	RBF	195	593	676	36	58.0%

Positive labeled regions of linear analysis are kept for textural analysis. The classification results and the predicted number of labels using different Gabor filters shown in Table 4.20, Table 4.21 and Table 4.22.

Table 4.19: Linear Feature Analysis : Classification accuracy for home set.

Exp. #	Kernel	True (+)'s s	False (+)'s	True (-)'s	False (-)'s	Local Accuracy
1	RBF	581	247	1872	0	90.8%
2	RBF	581	421	1698	0	85.1%
3	RBF	581	543	1576	0	79.8%
4	RBF	581	316	1803	0	88.2%

Table 4.20: Textural Feature Analysis : The classification results for road set. RBF kernel is employed. Gabor filter parameters S : scale, K : orientation, Side : mask size.

	True (+)'s	False (+)'s	True (-)'s	False (-)'s	Local Accuracy	Overall Accuracy
S = 3; K = 4; Side = 20	16	28	44	0	69.7%	98.1%
S = 3; K = 4; Side = 80	16	32	40	0	65.1%	97.8%
S = 3; K = 6; Side = 20	13	42	30	3	61.7%	97%
S = 3; K = 6; Side = 80	12	44	28	4	46.5%	96.8%
S = 4; K = 4; Side = 20	12	44	28	4	46.5%	96.8%
S = 4; K = 4; Side = 80	12	50	22	4	39.5%	96.4%
S = 4; K = 6; Side = 20	9	50	22	7	36.0%	96.2%
S = 4; K = 6; Side = 80	9	50	22	7	36.0%	96.2%

Table 4.21: Textural Feature Analysis : The classification results for urban area set. RBF kernel is employed. Gabor filter parameters S : scale, K : orientation, Side : mask size.

	True (+)'s	False (+)'s	True (-)'s	False (-)'s	Local Accuracy	Overall Accuracy
S = 3; K = 4; Side = 20	194	162	328	12	75.0%	86.7%
S = 3; K = 4; Side = 80	194	198	292	12	69.8%	84.3%
S = 3; K = 6; Side = 20	176	246	244	20	60.3%	79.9%
S = 3; K = 6; Side = 80	164	246	244	42	58.6%	79.1%
S = 4; K = 4; Side = 20	188	230	260	18	64.3%	81.2%
S = 4; K = 4; Side = 80	178	230	260	28	62.9%	81.1%
S = 4; K = 6; Side = 20	178	281	209	28	55.6%	77.7%
S = 4; K = 6; Side = 80	168	302	188	38	51.1%	75.6%

Table 4.22: Textural Feature Analysis : The classification results for home set. RBF kernel is employed. Gabor filter parameters S : scale, K : orientation, Side : mask size.

	True (+)'s	False (+)'s	True (-)'s	False (-)'s	Local Accuracy	Overall Accuracy
S = 3; K = 4; Side = 20	581	95	152	0	88.5%	96.4%
S = 3; K = 4; Side = 80	581	183	64	0	77.8%	93.2%
S = 3; K = 6; Side = 20	573	165	82	8	79.1%	93.5%
S = 3; K = 6; Side = 80	547	197	50	34	72.1%	91.4%
S = 4; K = 4; Side = 20	564	182	65	17	68.7%	92.6%
S = 4; K = 4; Side = 80	524	182	65	57	71.1%	91.1%
S = 4; K = 6; Side = 20	539	178	69	42	73.4%	91.8%
S = 4; K = 6; Side = 80	539	178	69	42	73.4%	91.8%

Table 4.23: Statistical measures for Test **B**.

	Sensitivity %	Recall %	F-score %	Specificity %
Test **B**				
Set 1	36	100	52	98
Set 2	54	94	68	88
Set 3	88	100	93	95

4.3 Comments on Results

Table 4.10 and Table 4.23 shows some statistical measures of our approach. Specificity measures the ability to detect false positives, our specificity results indicate that most of false positives are rejected. Another statistic sensitivity is used to measure the ability to detect true positives. The sensitivity results show that we keep all true positives with less number of false positives.

If we compare the results of first step for road test set, the linear analysis works significantly better than spectral analysis because of shape and structure of roads. For urban areas, spectral features seems to be relatively more discriminative than linear features. All of true positives are predicted as positive while most of the true negatives are eliminated. Image regions predicted as positive are then fed to the second step in which textural analysis is performed. In this step, accuracy is improved that true positives are still kept but the number of false positives is reduced for both sets.

CHAPTER 5

CONCLUSION

In our attempt to construct a semantic filter for satellite image content, the presented system provides a basic framework that combines

- user interactivity,
- low level feature extraction techniques including spectral,linear and textural features,
- a linear classifier,
- and tools to visualize the results.

We have built a software that allows user interaction and visual observation of the results to guide the classification procedure. This software can be used to filter the number of candidate image regions when a user indicates on the same image a few number of examples of an object. User can decide which specific object s/he wants to retrieve by creating dynamic training sets containing a specific object using the interface.

During the classification, the major aim was to keep true positives while reducing the number of false positives. We employed different image features for a possible query object. The generic class of objects that we aimed in this study was man-made structures, such as roads, buildings and bridges where linear features are important attributes. On the other hand, previous studies have shown the discrimination power of textural features. Therefore, we have included Gabor filter features to extract textural properties. However, their extraction requires a high computational complexity. Nevertheless, we have applied a two step algorithm in which only the image regions labeled as positive by the spectral or linear feature analysis step are fed to the classifier that employs textural features.

Guiding classification by interaction improves considerably the retrieval performance. A 2-dimensional plot of feature vectors may give user an intuition of the distribution of sets. For spectral analysis, it is relatively straightforward to give a plot of image regions in terms of the values of their mean and standard deviation. If the dimension of feature vector is higher than 2, then visualization by linear or non-linear mapping can be applied. In this study, we used multidimensional scaling (MDS) for this purpose. For linear and textural feature analysis, MDS is performed and all data set is plotted in 2-dimensions. By this way, user can update training set by visually looking at each datum as an image region in the original image and 2D mapping of its feature vector as a point on the 2D plot as well. We think that accuracy may be improved by updating the elements of training set by seeing them as image regions and their features visually with respect to the other ones.

In order to assess the performance of the system, we have used a satellite image whose regions are labeled as positive and negative depending on the condition that they contain road or building. During the conducted experiments, only a few number of positive instances, for example 10 image regions containing the sought object are indicated by the user. The rest of the procedure is applied without the user intervention. Although, accuracy depends on the parameter values, the use of linear features for road object query was very promising. Overall accuracy for both of the objects were higher than 80%. We believe that this can be improved if there were user intervention at the intermediate steps.

As future work, we plan to include other types of image features which will also allow to perform feature selection for a specific query object. Furthermore, since user indicates only positive instances, we intend to use one class classifiers [27] in an iterative and interactive manner.

REFERENCES

[1] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition : A review," in *IEEE Transactions on Pattern Aanalysis and Machine Intelligence*, no. 22, pp. 4–37, 2000.

[2] A. R. Webb, *Statistical Pattern Recognition John Wiley Sons New York*. 2005.

[3] D. M. and S. K., "Image information mining: exploration of image content in large archives," in *IEEE Aerospace Conference Proceedings*, no. 3, pp. 253–264, 2000.

[4] G. Erus and N. Loménie, "Automatic learning of structural models of cartographic objects," in *IAPR - Workshop on Graph-based Representations in Pattern Recognition*, 2005.

[5] S. Aksoy, K. Koperski, G. Marchisio, and C. Tusk., "Interactive training of advanced classifiers for mining remote sensing image archives," in *The 10th ACM SIGKDD Symposium*, 2004.

[6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, and Q. Huang, "Query by image and video content: The qbic system," in *IEEE Comput.*, no. 28, 1995.

[7] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," in *International Journal on Computer Vision*, no. 18, pp. 233–254, 1996.

[8] J. R. Smith, "Integrated spatial and feature image systems: Retrieval, compression and analysis," in *Ph.D. Dissertation, Columbia University*, 1997.

[9] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in mars," in *International Conference on image Processing*, 1997.

[10] I. Cox, M. Miller, T. Minka, T. Papathornas, and P. Yianilos, "The bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments," in *IEEE Transactions on Image Processing*, no. 9, pp. 20–37, 2000.

[11] G. Giacinto and F. Roli, "Bayesian relevance feedback for content based image retrieval," in *Pattern Recognition*, no. 37, pp. 1499–1508, 2004.

[12] K. Koperski, G. Marchisio, S. Aksoy, and C. Tusk., "Visimine: Interactive mining in image databases," in *In Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, no. 3, pp. 1810–1812, 2002.

[13] T. A. Berendes and R. M. Welch, "Ivics : Interactive visual image classification system," 1993.

[14] M. Datcu, H. Daschiel, A. Pelizzari, M. Quartulli, and A. Galoppo, "Information mining in remote sensing image archives: System concepts," in *IEEE Transactions on Geoscience and Remote Sensing*, no. 41, pp. 2923–2935, 2003.

[15] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*. Addison-Wesley, 1992.

[16] M. Sonka, V. Hlavac, and R.Boyle, *Image Processing Analysis and Machine Vision*. Chapman and Hall, 1993.

[17] A. Huertas and G. Medioni, "Detection of intensity changes with subpixel accuracy using laplacian-gaussian masks," in *IEEE Transactions of Pattern Analysis and Machine Intelligence*, no. 5, pp. 651–664, 1986.

[18] D. Marr and E. Hildreth, "Theory of edge detection," in *Proceedings of the Royal Society of London, Series B*, no. 207, pp. 187–217, 1980.

[19] J. Canny, "A computational approach to edge detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 679–698, 1986.

[20] R. Nevatia, "Evolution of simpilied hueckel edge-line detector," in *Computer Graphics and Image Processing*, no. 6, pp. 582–588, 1977.

[21] G. D., "Theory of communications," in *Journal of Institution of Electrical Engineers*, no. 93, pp. 429–457, 1946.

[22] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval image data," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 18, pp. 837–842, 1996.

[23] D.Pollen and S. Ronner, "Visual cortical neurons as localized spatial frequency filters," in *IEEE Transactions on Systems, Man and Cybernetics*, no. 13, pp. 907–916, 1983.

[24] Y. Lin, "Support vector machines and the bayes rule in classification," in *Data Mining and Knowledge Discovery*, no. 6, pp. 259–275, 2002.

[25] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, and H. Hoffman, "Xgvis: Interactive data visualization with multidimensional scaling," in *Journal of Computational and Graphical Statistics*.

[26] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[27] D. M. J. Tax, "One-class classification; concept-learning in the absence of counter-examples," in *Delft University of Technology, Ph.D Thesis*, 2001.