CONTEXTUAL MODELING OF REMOTE SENSING IMAGES WITH CONDITIONAL RANDOM FIELDS

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

BY

GÜLCAN CAN

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

SEPTEMBER 2013

Approval of the thesis:

CONTEXTUAL MODELING OF REMOTE SENSING IMAGES WITH CONDITIONAL RANDOM FIELDS

submitted by **GÜLCAN CAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Prof. Dr. Fatoş Tünay Yarman Vural _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Göktürk Üçoluk
Computer Engineering Department, METU _____

Prof. Dr. Fatoş Tünay Yarman Vural
Computer Engineering Department, METU _____

Prof. Dr. Yasemin Yardımcı Çetin
Graduate School of Informatics, METU _____

Assist. Prof. Dr. Sinan Kalkan
Computer Engineering Department, METU _____

Dr. Onur Tolga Şehitoğlu
Computer Engineering Department, METU _____

Date: _____

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: GÜLCAN CAN

Signature :

ABSTRACT

CONTEXTUAL MODELING OF REMOTE SENSING IMAGES WITH CONDITIONAL RANDOM FIELDS

Can, Gülcan

M.S., Department of Computer Engineering

Supervisor : Prof. Dr. Fatoş Tünay Yarman Vural

September 2013, 102 pages

Large within-class variance is a challenging problem for classification tasks in remote sensing. Contextual models are promising to address this problem. In this thesis, a contextual conditional random field model is proposed for target detection in satellite imagery. The proposed algorithm has three stages. First, contextual cues of the target that come from domain knowledge are identified by sparse auto-encoders and shown to be statistically consistent. The region represented by the most repetitive feature learned by sparse autoencoders is used as central node in the proposed model and called candidate region. Other nodes of the model are chosen as land-use land-cover classes in the surroundings of the candidate regions, since the spatial context of the target class is defined over expected and unexpected classes in its neighborhood. Secondly, regions that represent these classes are obtained by merging segments with the same label according to support vector machines. These regions are called meta-segments. In the last stage, the same features are extracted from the meta-segments and candidate region to be used as unary features in the conditional random fields model. Pairwise features in conditional random fields are essential for representing contextual relations and they are designed as class co-occurrence frequencies in three different neighborhoods of the candidate region. For each candidate region, a dynamic conditional random fields model is generated. The proposed method is robust in terms of being threshold-free and selecting contextual cues via sparse auto-encoders. Performance of the method is competitive to rule-based methods and segmentation-based classification methods.

Keywords: Conditional Random Fields, Remote Sensing, Contextual Classification

ÖZ

UZAKTAN ALGILAMA GÖRÜNTÜLERİNİN KOŞULLU RASGELE ALANLARLA BAĞLAMSAL MODELLENMESİ

Can, Gülcan
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Fatoş Tünay Yarman Vural

Eylül 2013 , 102 sayfa

Uzaktan algılama alanında sınıf-içi çeşitliliğin çok fazla olması sınıflama çalışmalarında zorlu bir sorundur. Bağlamsal modeller, bu sorunu çözmekte umut vadetmektedir. Bu tezde, uydu görüntülerinde hedef tespiti için bağlamsal bir koşullu rasgele alan modeli önerilmektedir. Önerilen algoritma üç aşamalıdır. İlk olarak, ilgi alanı bilgisine dayalı, hedefe ait bağlamsal ipuçları seyrek oto-kodlayıcılarla tanınmış ve istatistiksel olarak tutarlılıkları gösterilmiştir. Seyrek oto-kodlayıcılarla öğrenilen en sık tekrarlanan özniteliğe karşılık gelen alan, önerilen modelde merkez düğümle gösterilmiş ve aday bölge olarak adlandırılmıştır. Modeldeki diğer düğümler, aday bölgenin civarındaki arazi örtüsü ve kullanımı sınıflarını temsil edecek şekilde seçilmiştir. İkinci olarak, bu sınıfları temsil eden alanlar, elle tasarlanmış özniteliklerin destek vektör makinalarına verilmesiyle elde edilen etiketlerden aynı olanlarının birleştirilmesiyle elde edilmiştir ve meta-bölüt olarak adlandırılmıştır. Son aşamada, aynı öznitelikler meta-bölütlerden ve aday bölgeden çıkartılıp koşullu rasgele alan modelinde tekli öznitelik olarak kullanılmıştır. Koşullu rasgele alanlardaki ikili öznitelikler bağlamsal ilişkileri temsil etmeleri açısından önemlidir ve aday bölgenin üç farklı komşuluğundaki sınıfların birlikte görülme sıklığı olarak tasarlanmıştır. Her aday bölge için ayrı bir dinamik koşullu rasgele alan modeli üretilmiştir. Önerilen yöntem, eşik değerlerinden bağımsız olmasıyla ve bağlamsal ipuçlarını seyrek oto-kodlayıcılarla seçmesiyle gürbüzdür. Yöntemin başarımı, kural tabanlı yöntemlerin ve bölüte dayalı sınıflama yöntemlerinin başarımıyla yarışmaktadır.

Anahtar Kelimeler: Koşullu Rasgele Alanlar, Uzaktan Algılama, Kavramsal Sınıflama

To my family

Aysel, Semih, Gizem

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Professor Fatoş Tünay Yarman Vural for giving me the opportunity to work with her. This dissertation would not be possible without her constructive criticism, positive attitude and willingness to help. She has always been a great resource for me not only in research issues, but also in many matters of life. I feel really lucky to be her advisee and once more would like to express my greatest thanks.

I would like express my thanks to Professor Selim Aksoy and Professor Sinan Kalkan for their enthusiasm during their courses which made me intrigued by computer vision. Their comments were of great value and their positive attitude have always motivated me.

I would like to thank Professor Göktürk Üçoluk, Professor Sinan Kalkan, Professor Yasemin Yardımcı Çetin and Onur Tolga Şehitoğlu for accepting to be members of my dissertation committee. Their valuable feedback was of great value to finalize this work.

I really owe much to Orhan Fırat. He supplied me with the material for the start up of this work. I would like to express my greatest thanks to him for his continuous support, feedback and contributions to this work.

For providing the dataset, I would like to acknowledge HAVELSAN Inc.

I would like to acknowledge my friends. I would like especially to thank to Okan, Dilek, Ruşen, Levent, Sinem. They made my life fun.

Finally, my deepest thanks go to my family. I would like to thank my parents Aysel and Semih, and my sister Gizem, for their support and patience. They have always made their best to provide me with a convenient environment and were the greatest supports of mine in all matters of life.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

LULC	Land Use/Land Cover
DEM	Digital Elevation Models
DTED	Digital Terrain Elevation Data
RoI	Region of Interest
NIR	Near-infrared
NDWI	Normalized Difference Water Index
NDVI	Normalized Difference Vegetation Index
SLIC	Simple Linear Iterative Clustering
PCA	Principled Component Analysis
LBP	Local Binary Pattern
LEP	Local Edge Pattern
HOG	Histogram of Oriented Gradients
VAR	Variance Difference
GLCM	Gray-Level Co-occurrence Matrix
KL	Kullback-Leibler
SVM	Support Vector Machine
k-NN	k-Nearest Neighbor
CNN	Convolutional Neural Network
RNN	Recursive Neural Network
BFGS	Broyden–Fletcher–Goldfarb–Shanno
LBP	Loopy Belief Propagation
MAP	Maximum a Posteriori
PGM	Probabilistic Graphical Models
MRF	Markov Random Fields
CRF	Conditional Random Fields
CS-CRF	Contextual Star Conditional Random Fields

CHAPTER 1

INTRODUCTION

From the ages of industrial revolution, automating the processes is the main aim of the mankind. For building the future, apart from automatizing the routines, producing smart processes is necessary. Artificial intelligence and machine learning community work hard for constructing such smart processes which can infer the statistical dependencies from the observations, combine them with domain knowledge and make a decision that maybe even better than the decision of a human being.

In this aspect, analyzing the visual scenes is one of the heavily studied areas. From automatic detection of objects in an image, tracking a person in a video for his suspicious behaviors, automatic annotation of images, to decomposition of a scene, there are various applications that attempt to build autonomous systems in the intersection of computer vision and machine learning.

This thesis presents a contextual model for detecting regions belonging to a target class in remote sensing images. In this model, spatial contextual relationships of the target class is embedded within a conditional random field which is constructed over meta-segments of land use/land cover (LULC) classes. Meta-segments are obtained by combining segments belonging to the same class after an initial classification. Algorithms used to construct the model are described, and then the experiments conducted are presented.

1.1 Motivation

A fundamental problem in computer vision tasks is that objects exhibit high within-class variance. In other words, samples from the same class may not share similar characteristics in terms of color, shape, texture, composition etc. Figure 1.1 illustrates this problem with some real world examples for the chair class.

Designing a system that recognizes samples which exhibit high within-class variance is a challenging problem. However, this recognition task becomes manageable if there exist a contextual cue of the class. For instance, if there are people sitting on the chairs in Figure 1.1, we get the intuition that the characteristic what makes a chair is its functionality. This

contextual cue can be evaluated in terms of co-occurrence of two classes as well as their spatial interaction. We would not consider an object as a chair if the person sits beneath the chair, rather than sitting on it. Therefore, integrating contextual cues for the classification of objects with high intraclass variance is expected to perform relatively better than that of the classifiers, which employs low-level visual features.

Figure 1.1: Examples of objects with high within-class variance.

It is well known that visual patterns and object occurrences in remote sensing images exhibit high intra-class variance. For example, two airfields may have entirely different color structures, composing roads, shapes, sizes and configurations of their sub-parts (e.g. one may have just one, crossing, parallel runway(s) having hammer shaped, circular, polygonal dispersal areas and located in sandy, snowy, coastal or urban terrain). Occasionally, these objects and object groups may even look more similar to instances within other classes than to instances within their own class, e.g. circular oil tanks of a refinery and circular dispersal area of a military airfield.

Contextual models are significantly useful in order to handle the huge variability within classes in the image because of their expressive representations. By forming a contextual framework, any object can be accurately classified not only by considering its low-level vision features but also its local context (spatial relations) over a probabilistic graphical model. Figure 1.2 illustrates some real world examples that exhibit high intra-class variance.

The ability to recognize such complex objects comes from both appearance cues of the object itself and contextual relations of the complex objects with their surroundings. These contextual relations can be defined as co-occurrence frequency of other classes in a predefined neighborhood of the complex object. For instance, if the complex object in consideration is an airfield, we would expect urban, vegetation, water existence nearby to be less than a certain ratio. This information comes either from domain knowledge or by explicit observations. However, deciding this ratio by a static threshold is not desirable, since less likely configurations are not allowed at all. As in the case of urban areas in the surrounding of an airfield, we may set a 20 percent urban co-occurrence threshold in 300 meters neighborhood by domain knowledge. Yet, Figure 1 demonstrates cases contradicting with such a threshold. Hence rather than determining crisp thresholds for the recognition task of a complex object, constructing a probabilistic model is much more flexible and suitable.

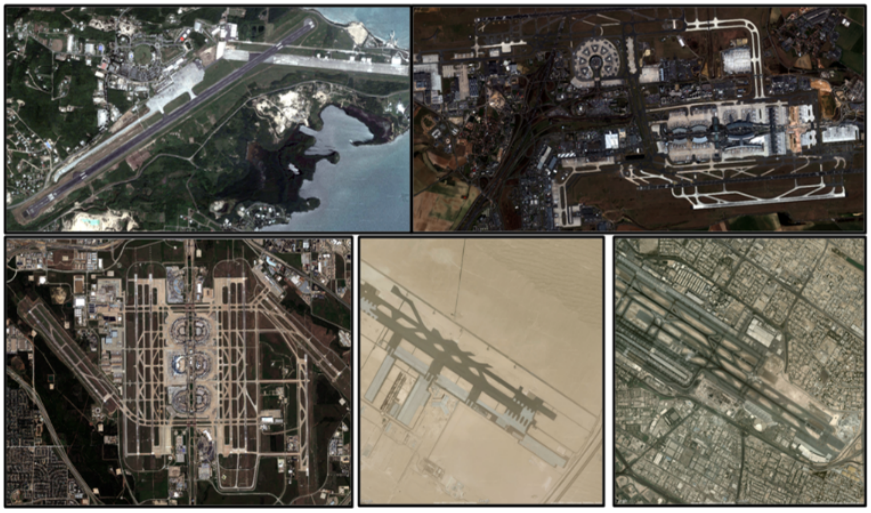

Figure 1.2: Examples of airfields from remote sensing images.

Probabilistic graphical models are the state-of-art approach for modeling contextual relations between semantic classes [1] and have many applications in remote sensing [2-3]. Since labels in spatial data are not independent as well as observations, assumptions on data being "independent and identically distributed" (i.i.d.) is violated by using traditional classifiers. Therefore such classifiers may produce undesirable results when applied to such data.

This problem motivates the use of Markov Random Fields (MRFs) and more recently Conditional Random Fields (CRFs) for spatial data. In the proposed approach, contextual relations between a complex object and its surroundings, which is characterized by LULC classes, are modeled within a CRF framework. The major contribution of the proposed model is that a random field is constructed over semantic classes rather than pixels or super-pixels as in the literature. Our model aims to correctly identify the complex object by recognizing the co-occurrence pattern of all other classes in its surroundings.

1.2 Contributions

This thesis makes four contributions to the existing methodologies in the literature.

1. *Seed node:* We introduce a new node type, called seed node, to represent the contextual information of the target class. Seed node is placed at the heart of the model, since its contextual influence builds the whole model and affects the interactions between seed node and other nodes, namely surrounding nodes. The random variable shown as the seed node in the model, represents a candidate region of the target class. This candidate region is obtained by the help of domain knowledge and demonstrated to be consistent by the help of sparse autoencoders.

2. *Meta-segments:* We introduce the concept of meta-segment to speed up the computation and improve the performance. Nodes around the center node (seed node), namely surrounding nodes, represent the meta-segments in the proposed model. Meta-segments are obtained by merging segments with the same label after an initial classification. Advantages of this approach can be listed as model complexity and feature representativeness. Using the meta-segments instead of segments assures that model complexity is bounded to the number of classes in the scene. For feature representativeness, meta-segments are expected to be more capable of capturing textural features rather than small segments.

3. *Spatial interactions:* As the nature of CRFs, interaction potentials can be designed as quite complex functions. Although, in earlier studies, interaction potentials are simply acquired by concatenating or taking difference of adjacent nodes, in this study we have followed a different approach and defined the spatial contextual interactions as co-occurrence statistics of classes in three different scaled neighborhoods.

4. *Star-structure:* We introduce a special topology called star structure to represent the context information of a target area/object. Star structure is especially chosen for the proposed model, since the goal is to correctly classify the candidate region (seed node) by the help of spatial contextual interactions of surrounding nodes. Interactions between nodes are restricted to seed node and surrounding nodes so that the final label of the seed node is not confused by the interactions between two surrounding nodes.

1.3 Thesis Outline

The rest of this thesis is organized as follows:

Chapter 2: An Overview of Conditional Random Fields Models in Remote Sensing. This chapter discusses the challenges in the classification of remote sensing data, first. Then, extracted features and segmentation algorithm utilized are described. Fundamentals of classification and conditional random fields are given next. Then, existing methodologies that uses graphical models on remote sensing data are discussed. After giving the formalization of sparse autoencoders, the chapter is concluded.

Chapter 3: A Contextual Model with Conditional Random Fields. In this chapter, first of all, overview of the proposed system is provided. Then, context notion is discussed and how the context is established in our model is explained. The candidate regions of the target class are determined by the contextual hints from the domain knowledge, however this choice is shown to be statistically stable by the help of sparse autoencoders. Meta-segments notion is given next. Then, the topology, energy function of the proposed model are provided. After providing the details about the parameter estimation and inference in the model, the chapter is concluded.

Chapter 4: Experiments. Experiments conducted during this study are presented in this chap-

ter. First of all, details of the dataset and how it is formed are explained. Then, several segmentation algorithms, namely simple linear iterative clustering , watershed transformation, and mean shift clustering, are examined. After giving details about the training of the sparse autoencoders, experimental results of two initial classifiers, namely support vector machines and k-nearest neighbor classifier, are evaluated. Then, different conditional random field approaches, segment-based CRF, fully-connected CRF and star CRF are compared.

Chapter 5: Conclusion and Discussion. The thesis is concluded with the summary of the proposed model and discussion of the future work.

CHAPTER 2

AN OVERVIEW OF CONDITIONAL RANDOM FIELDS MODELS IN REMOTE SENSING

In this chapter, background information about the problem domain, which is remote sensing and scene analysis, is given first. Then, features and segmentation algorithms used during classification of satellite imagery are overviewed. After providing fundamentals of classification and probabilistic graphical models (PGM), existing algorithms in conditional random fields, which are the state-of-the-art of PGM, are given. A brief introduction about sparse autoencoders is provided next for contextual invariance searching. Then, the chapter is summarized.

2.1 Challenges in Remote Sensing

Many remote sensing tasks, such as land use/land cover (LULC) classification, change monitoring and urban planning, are subject to computer vision and machine learning approaches for analysis and interpretation. They are analyzed with various supervised and unsupervised techniques, from rule-based classification to much complex machine learning algorithms, such as probabilistic graphical models. However, dealing with remote sensing data is quite challenging, in the sense that same type of objects/areas may not exhibit similar characteristics even in a close neighborhood and considering examples all around the world, great variety is observed. The data can be very complex either due to the nature of target object/area, i.e. composite objects, or scattered spectral characteristics of different samples from the same kind of object/area, i.e. buildings with various roof color or deciduous forests in different seasons. Furthermore, characteristics of each image sensor may be different. This means that classifiers which depend only on spectral values are prone to fail. On the other hand, other type of features like shape and texture features show great within-class variance for remote sensing objects. Thus, determining the pattern of objects/areas is a challenging problem. Most of the time, ancillary data is utilized for more general and dependable classification systems. Ancillary data may come from different sensors such as SAR imagery or Digital Elevation Models (DEM). Global information such as azimuth angle, resolution of the imagery, location of the area etc. can aid classification as well. Apart from these, contextual information

about objects/areas and semantic rules that come from expert knowledge are quite promising for classification tasks.

In the early days of analysis of remote sensing imagery, large areas such as forest, water areas, urban areas, etc. are studied for classification tasks. This is known as LULC classification in the remote sensing community and it is a fundamental step for further processing and analysis. Land cover classification refers to analysis of what occupies the earth surface such as forest, lake, agricultural land etc. whereas land use classification means analysis of what purpose the land is used by people such as urban, rural, industrial areas etc.

LULC classes are defined in a hierarchical manner. There are two widely adopted hierarchies, namely USGS [3] and CORINE 2000 class hierarchies [37]. Since remote sensing imagery is very complex and can contain high within-class variance, there are levels of classes in both hierarchies and in deeper levels more detailed sub-classes are defined. For instance, building is a sub-class of urban or built-up land class. In this thesis, a subset of USGS classes is adapted to be used during classification of complex objects in high resolution multispectral images. Used classes can be listed as urban land, water, agricultural land, forest, soil and concrete.

Recently, hyperspectral data have been drawing great attention. Compared to multispectral data, which has 4 to 8 bands (IKONOS, QUICKBIRD, GEOEYE, LANDSAT, and WORLD-VIEW-2); hyperspectral images from various satellites (AVIRIS, HYDICE, ARCHER, and Hyperion) provide 126 to 512 spectral channels [17]. Hence they provide more information to the researchers to process and extract. The extra information enables to work on detection/classification of subclasses e.g. classes of level 2 or 3 in USGS classes. It is clearly observed that as number of bands increases along with the spectral information the imagery carries, spectral mixing of classes decreases. Thus, classes with close reflectance values become separable. The most obvious example can be the comparison of LANDSAT data with 7 bands and IKONOS data with 4 bands. Even though resolution of LANDSAT data is low compared to IKONOS and resolution affects the performance due to mixing pixels problem, there are more variety of classes covered to be classified in the studies on LANDSAT data [67, 74]. Hyperspectral data can be considered as the upper bound for current satellite technology. With over 100 bands, even though based on solely reflectance values, classification can be performed on a broad range of classes. Yet, another significant issue to be considered emerges: extracting/selecting meaningful information from these bands that would match well with the classes. This problem is known as feature selection/reduction and there are stereotypic solutions applied in most studies. One of them can be stated as principled component analysis (PCA) [44]. Even though, hyperspectral images, which bring more information to process, are preferable, LULC classification over multispectral images is still common and needed as LULC classification may be essential for determining region of interest (RoI) for high resolution multispectral remote sensing applications like building and road extraction. With the enhancement of multispectral image resolutions, they have become more promising to detect sub-level classes as well.

As the resolution of imagery decreases, object recognition becomes feasible apart from land

cover classification. High resolution of images brings advantage of recognizing smaller objects such as single buildings, cars, airplanes, ships, etc. On the other hand, the detail in the imagery drastically increases the within-class variance and the classification tasks become harder. For high-resolution multispectral imagery, there are several approaches adopted in the literature to overcome insufficiency of spectral information, such as making use of textural features, shape information of target classes or embedding spatial information to get context involved. Since remote sensing classes have high within-class variance, capturing a unique texture or spectral reflectance for a target class is generally impossible. Even if very high performance is achieved after LULC classification of an image, there is no guarantee for obtaining great results on other images with the same classification strategy. This problem occurs due to change in illumination or reflectance characteristics of different scenes and different satellites. For example, when urban class and one of its subclasses, building, is considered, it is known that several building classes are defined which demonstrate different characteristics such as buildings with red roof and buildings with bright roof etc. However, determining the number of such classes is another issue to be addressed. In general, researchers choose this number by investigating the single imagery. Similar to this case, in forest identification studies number of classes are determined by the expert who evaluates the target imagery. Although this seems like a strong manipulation, expert/domain knowledge integration is a necessity in LULC analysis studies [40].

Formerly, due to coarse resolution, pixel and sub-pixel analysis were carried out in order to extract meaningful information from remotely-sensed data. Recently, as availability of high-resolution data increases, remote sensing objects have started to correspond to more than one pixel. This brings concept of "super-pixels" (segments) and popularity of object-based approaches as well.

2.2 Methods Employed Prior to Conditional Random Fields

In this section, we summarize the basic computer vision methods, such as feature extraction and segmentation, employed in remote sensing literature.

2.2.1 Feature Extraction

Feature extraction is the first step of every classification task. When classifying remote sensing imagery, selecting representative and discriminative features plays a crucial role. Considering the classes which are subject of interest, essential features are also computed as well as spectral features.

2.2.1.1 Normalized Difference Water Index

In multispectral images, water areas give reflectance near to zero in near-infrared (NIR) band and by taking difference of NIR value and green band value, water areas can be differentiated from green land or soil areas [55]. In the literature, a constant threshold value which is tuned according to dataset is widely used. However, to adapt to image-specific reflectance characteristics, dynamic threshold selection is preferred in this study. First, normalized difference water index (NDWI) map of images computed from the following equation

$$NDWI = \frac{Green - NIR}{Green + NIR}. \tag{2.1}$$

Then, a dynamic threshold value is determined by Otsu's method [59]. According to this dynamic threshold, water areas are determined by taking the region that have samller spectral values than the determined threshold. For feature extraction step, mean and variance of NDWI map in the region of interest is used along with ratio of water pixels in the region of interest.

2.2.1.2 Normalized Difference Vegetation Index

Similar to NDWI, there is another index frequently used in remote sensing literature to extract vegetation area. Due to sensitivity of red and near-infrared bands to chlorophyll pigments, their normalized ratio is used for identifying vegetation areas. Normalized difference vegetation index (NDVI) is computed as follows:

$$NDVI = \frac{NIR - Red}{NIR + Red}. \tag{2.2}$$

A similar feature extraction step as in NDWI case is applied to NDVI map, as mentioned in the previous section.

2.2.1.3 Textural Features

In texture classification, representativeness of features is crucial. Textural features have a common property to reflect the spatial configuration of a pattern beyond color-related features. Thus, they capture class-specific patterns and have the ability to discriminate similar or close-colored patterns. From this perspective, they can be used to represent objects as well.

In remote sensing area, textural features are used most widely for land use/land cover classification as well as object detection. With the availability of high resolution satellite data, characteristics of remote sensing objects or fields can be analyzed further. Differences between two forest types would be realistically tractable for instance. Thus, representation capability of textural features gains importance in that sense. Formerly, textural features

were examined in a statistical manner. Features adopted from those times can be counted as gray-level co-occurrence matrix method [34, 41] and filtering based approaches like Gabor filters [41, 79, 4, 31] and wavelet transform [14, 51]. Although these approaches can exhibit competitive performance with similar training and test data, they seem to suffer as within-class variance gets high or rotation problem gets involved [32]. Since remote sensing data is highly-variant, can contain complex structures and patterns learned can be in any directions, more representative features are desired. Recent approaches emerged from the necessity of rotation, scale and affine invariance, better demonstration and discrimination of similar classes etc. These approaches include local binary pattern (LBP) [75], local edge pattern (LEP)[75], histogram of oriented gradients (HOG) [19] and edge orientation extraction [75]. Ojala et.al. states that LBP feature can capture spatial configuration of pattern quite well, yet it would be preferable to combine it with variance difference (VAR) feature for a full representation [58]. Guo et al. asserts that LBP/VAR feature consider variance difference in a global sense, yet it should be local as well. Thus they propose LBPV for local representation of spatial arrangement as well as contrast difference [32]. HOG feature is proposed by Dalal and Triggs for human detection task [19], and recently used for car detection in remote sensing as well [26, 72]. It captures texture by filtering the image and features are obtained by combining histograms of gradient directions taken from image tiles. Edge orientation is favorable since it seems to be able to differentiate man-made and natural structures [75, 71]. Edge responses are obtained by applying steerable filter beforehand [23]. In order to make extracted edge orientation features rotation invariant Fast Fourier Transform is used as a post-processing step [75, 71].

In a comprehensive study, these state-of-the-art textural features and traditional textural features such as Gabor and gray-level co-occurrence (GLCM) features are compared for several remote sensing classes, namely water, forest, agricultural land and urban land [5]. Table 2.1 summarizes the suggested and not suggested textural features for the classes in that study.

It is concluded that Gabor textural features are representative for all the selected classes which are water, forest, agricultural land and urban area. Although, local binary patterns (LBP) and its variants seem to be competitive, different versions are recommended for each class. Thus, gabor textural features are chosen to be used in this thesis. Hence, original image is convolved with Gabor filters in seven directions. Response maps are superpositioned and a single Gabor map is obtained as in [65]. Then, mean, variance values and ratio of thresholded pixels in the region of interest are taken as textural features.

2.2.1.4 Elevation Variance

Digital terrain elevation data (DTED) of images are frequently employed as a supplementary feature. Level 2 DTED data, which is also used in this study, has the resolution of 30 meter per pixel. This is rather coarse, considering resolution of the multispectral imagery used is 2 meter per pixel. In order to use DTED data, an interpolation step is required. DTED data is adjusted such that pixels which do not overlap with DTED data are assigned to an interpolated value

Table 2.1: Textural feature review from [5]

Class Name	Recommended Features	Not Recommended Features
Water	Gabor, HOG, LBP-Uniform (DD), color histogram (DD)	LEP, edge orientation
Forest	Gabor, HOG, LBP-Uniform (DD), LBP-Rot. Inv. (k-NN), LBPV-Rot. Inv. (k-NN)	LEP, edge orientation
Urban	Gabor, LBP-Rot. Inv., LBPV-Rot. Inv. Uniform, LBPV-Rot. Inv. (DD) , LBPV-Rot. Inv. Uniform (DD), LEP, edge orientation	Color Histogram
Agricultural land	Gabor, LBP-Uniform, LBP Rot. Inv. Uniform	GLCM, HOG, LEP

according to elevation values of neighboring pixels. Thus, although slightly smoothed, an elevation map with the same resolution of the original image is obtained. Elevation variance in segments are used as representative features for airfield target class. From domain knowledge, elevation variance in airfield areas should be quite close to zero. This information is useful for differentiating airfields and long roads or agricultural land borders.

2.2.2 Segmentation

Segmentation partitions an image R into constituent subregions regions $R_1, R_2, ..., R_n$ such that

- Union of all R_i is equal to R, i = 1, ..., n. ($\cup_{i=1...n} R_i = R$)
- R_i is a connected set, i = 1, ..., n
- Intersection of two adjacent regions is empty. ($R_i \cap R_i = \varnothing$, for all i and j, $i \not\equiv j$)
- $Q(R_i) = TRUE$ for i = 1, ..., n
- $Q(R_i \cup R_j) = FALSE$ for any adjacent regions i and j

where $Q(R_k)$ is a logical predicate defined over the pixels in R_k. For $Q(R_k)$ to be true, all pixels in R_k have to be above a certain similarity threshold T[27]. Thus, homogeneous segments are obtained. As similarity between pixels in a region increase, homogeneity of the segment increases. Discontinuity or dissimilarity between pixels of adjacent segments is another important point during determining boundaries between segments. Segmentation is a fundamental process and frequently used as a pre-process of classification tasks.

In [20], segmentation techniques used in remote sensing literature are documented. Segmentation techniques are categorized into three parts: model driven vs. image driven approaches, approaches based on homogeneity measures, and approaches based on some image operations such as edge recognition, region growing/splitting.

Dey et. al. point out that image driven approaches start from pixels detected as edges and try to merge them in order to obtain the boundaries of the segments. On the other hand, model driven approaches assume that there are underlying patterns in the image and these patterns can become the distinguishing factor between segments. Various methods are presented as model-driven approaches. Some of them are object-background models that depend on thresholding, MRF-based models which are highly popular, yet quite complex, fuzzy models that address ambiguity problem, multi-resolution models which are favored for urban area segmentation in very high resolution imagery, and watershed approach [20].

As homogeneity measures, spectral, spatial, texture, shape, size, contextual, temporal and prior knowledge are remarked in [20]. Dey et.al. indicate that using only spectral values for segmentation does not suffice, thus a segmentation approach utilizing several the aforementioned measures is desirable [20]. However, engaging all these measures during segmentation could be hindering in terms of time concerns and model complexity. Furthermore, all of the classes in the classification task may not have prior knowledge or a discriminative texture either.

Hence, in this thesis, segmentation is considered just as a preprocessing step and a moderate approach in terms of time concern and performance is followed. Mean shift segmentation [18], which clusters pixels according to spectral values, is used during the experiments. Mean shift algorithm also considers spatial proximity and size of segments. In the end, segments consists of at least given number of pixels, and spectral and spatial difference of these pixels are not larger than given respective bandwidths.

Mean shift is a nonparametric feature space mode seeking algorithm. Modes are obtained by gradient ascent over density function. This is also called *kernel density estimation* over the distribution

$$\hat{f}(x) = \frac{1}{Nh^d} \sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right) \tag{2.3}$$

where x_i is the feature vector, $K(\xi)$ is a kernel function, d is the dimensionality of the feature space (originally d=5, however d=6 in this study, since an additional color value, near-infrared band value, is also utilized) and the parameter $h \in \mathbb{R}^+$ controls the bandwidth of the kernel. As s and r subscripts stand for the spatial and color ranges respectively, the kernel function is as follows:

$$K(\xi) = \frac{C}{h_s^2 h_r^3} \exp\left(-\frac{1}{2}\left\|\frac{\xi s}{h_s}\right\|^2\right) \exp\left(-\frac{1}{2}\left\|\frac{\xi r}{h_r}\right\|^2\right). \tag{2.4}$$

Basically, for each data point, a window is fixed around it and the mean is computed. Then the center of the window is shifted towards the mean till convergence. Points associated with the same mode belong to the same segment. In formal perspective, as $\{x_i\}_{i=1..n}$ are the original image points, $\{z_i\}_{i=1..n}$ are the convergence points and $\{L_i\}_{i=1..n}$ are a set of labels, mean shift algorithm can be summarized as follows [18]:

1. Mean shift procedure is run for each x_i, in other words $z_i \leftarrow \frac{\sum_{i \in I} K'(z_i - x_i) x_i}{\sum_{i \in I} K'(z_i - x_i)}$ until convergence and then convergence point is stored in z_i, i = 1..n.

2. The segments $\{C_p\}_{p=1..m}$ are obtained by grouping together all z_i which are closer than h_s in the spatial domain and h_r in the range domain.

3. Assign $L_i = \{p | z_i \in C_p\}$ for each i = 1..n

4. Optionally, spatial regions containing less than M pixels are eliminated. (M is another optional parameter.)

First strong point of mean shift algorithm is being nonparametric. Mean shift algorithm does not make assumptions about cluster number or shape of clusters as opposed to other segmentation algorithms such as k-means clustering algorithm [54], parzen-window approach [60] or other kernel estimation algorithms. Regardless, spatial and range bandwidth parameters h_s and h_r are expected to be tuned in order to determine the resolution of mode detection. Secondly, mean shift algorithm is not sensitive to initializations and robust towards outliers. On the other hand, it is computationally expensive, since it typically runs for each pixel or for uniformly selected pixels [18]. Mean shift algorithm has a time complexity of $O(In^2)$ as n is the pixel number, I is number of iterations. However, time complexity vs. segmentation accuracy trade-off here pays off for the mean shift algorithm, since its results are pleasing to eye in general, compared to other segmentation algorithms. Another weak point of mean-shift algorithm is that it does not scale well with the dimension of feature space [18].

2.3 Object Classification Problem in Remote Sensing

Classification is defined as a mapping between an input vector **x** and one of the pre-defined discrete set of class labels C_k as k = 1, 2, ..., K. Classification partitions the input space into non-overlapping regions by defining decision boundaries among the corresponding classes [8].

In this thesis, the data is represented by a feature matrix of segments (super-pixels) or conceptually-composed regions. Let $\mathbf{D_i}$ be the feature matrix, as i = 1, ..., n+1 and n is the total number of segments. The last row of the feature matrix, $n + 1_{th}$ row, belongs to the candidate region of target object or area. The rest of the rows, 1_{st} to n_{th} inclusive, belongs to the segments in the neighborhood of the candidate target region.

The target object or area has to have certain characteristics. First of all, it has to correspond to a contextually-consistent object or area. Contextually-consistent means that each sample of the corresponding class can be observed in the same context or exhibits the same type of contextual cue. For instance, for all the samples of airfields, there is always an airport building near an airfield. In the context of the airfield, airport building-airfield pair is contextually-consistent. Second, context of the target area can be defined in spatial perspective with respect to its neighbors, rather than its own location in the image or its rotation. For example, close neighborhood of an airfield does not contain forest land or urban land and generally consists of bare land, since it should be flat and empty for a safe take-off. On the other hand, rotation or location of the airfield in the image can be disregarded, since it does not provide any information for the classification task.

In this study, employed class labels are water, forest, agricultural land, urban land, soil, concrete and target class. In some of the experiments, regions are classified as either target class or not. In other experiments, regions or segments are classified to be one of the all aforementioned classes.

2.4 Probabilistic Graphical Models

A random variable is represented by a probability distribution of an entity in probability theory. For probabilistic graphical models, there are two types of random variables: observed input variables (X) and to-be-predicted output variables (Y). Each variable can be either discrete or continuous. In this study, the random variables are considered to be discrete in nature.

A probabilistic graphical model is composed of a set of random variables $V = X \cup Y$ which are factorized over a graph G. Factorization is defined as dividing the probability distributions in a graphical model according to independencies. A graphical model can be directed, as Bayesian networks, or undirected, as Markov networks. Both directed and undirected graphical models abide to factorization. In other words, the distribution is represented as a product of factors defined over subsets of variables, or *cliques*. Clique is defined as a subset of the nodes in a graph such that there exists an edge between all pairs of nodes in the graph [8]. A factor $F = \psi_c$ is a function with scope of a set of random variables in a clique c as $\psi_c : \nu_c \rightarrow \Re^+$, ν_c stands for any of the random variables in the clique, $\Re^+$ stands for set of positive real numbers, and $c \in C$ as C is the set of cliques in the graph.

Given the set of cliques C over random variables $\mathbf{Y} = (Y_1, ..., Y_n)$ and $\mathbf{X} = (X_1, ..., X_n)$, an undirected graphical model holds the following formalization;

$$P(\mathbf{Y}, \mathbf{X}) = \frac{1}{Z} \prod_{c \in C} \Psi_c(\mathbf{Y}_c, \mathbf{X}_c) = \frac{1}{Z} \exp\left\{ - \sum_{c \in C} \psi_c(\mathbf{Y}_c, \mathbf{X}_c) \right\}, \tag{2.5}$$

and a factor is defined as,

$$\psi_c(\mathbf{Y}_c, \mathbf{X}_c) = \sum_k \theta_{ck} f_{ck}(\mathbf{Y}_c, \mathbf{X}_c), \tag{2.6}$$

where θ_c is parameter vector and f_{ck} is feature functions or sufficient statistics. Factors $\Psi_c(\mathbf{Y}_c, \mathbf{X}_c)$ are also known as *potential functions*.

In formula 2.1, Z is a constant called as *partition function* and is computed as,

$$Z = \sum_{\mathbf{Y},\mathbf{X}} \prod_{c \in C} \psi_c(\mathbf{Y}_c, \mathbf{X}_c). \tag{2.7}$$

This constant assures that the distribution sums to 1. Although its computation is intractable in some cases, as it requires summing over an exponential number of assignments to the variables, there are various methods to approximate Z.

Independency relations can be deducted by the help of Hammersley-Clifford theorem [15] in undirected graphical models. According to Hammersley-Clifford theorem, given the Markovian property holds, which states that given the neighbors of a positive random variable or the clique it belongs, its probability is independent from the other variables in the graph, and as this applies to all variables, this probability distribution factorizes over that undirected graph [33].

Factorization is demonstrated by factor graphs. A factor graph [46] is a bipartite graph G = (V, F, E) as a node v_k (random variable) in the set of V is connected to a factor node if that factor takes v_k as an argument. In Figure 2.1, factor graph of an undirected graphical model is depicted. The square blocks represents the factors whereas shaded circles represent observed variables x.

Directed graphical models or Bayesian networks are factorized as,

$$p(Y, X) = \prod_{\upsilon \in V} p(\upsilon | \pi(\upsilon)), \tag{2.8}$$

where $\pi(\upsilon)$ are the parents of υ in graph G = (V,E). Figure 2.1 and 2.2 shows a simple pairwise Markov network and a simple Bayes network respectively.

The main difference between directed and undirected graphical models is that directed models are based on Naive Bayes method which models joint distribution P(Y, X) and thus they are generative models. On the other hand, undirected models are based on logistic regression method which models conditional probability $P(Y \mid X)$ and hence they are discriminative.

Modeling P(X) is rather a redundant step for classification. Avoiding to estimate the density function P(X) is one of the superiorities of discriminative models, since the data may contain interdependent features and may be difficult to estimate a statistical model. Generative

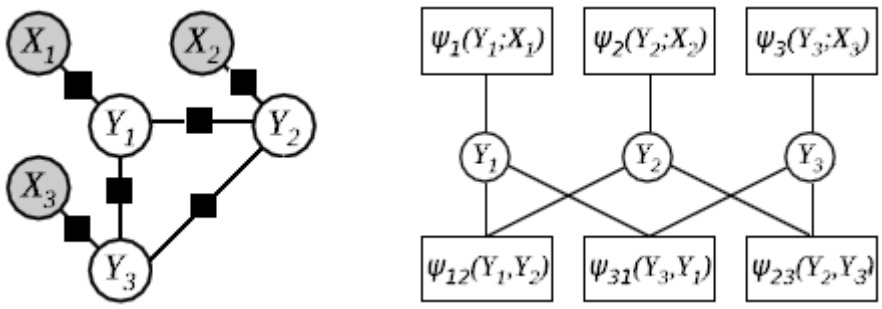

Figure 2.1: A pairwise undirected graphical model with factors (left) and its factor graph (right). In the factor graph, upper factors $\psi_i(Y_i, X_i)$ stand for unary potentials, and lower factors $\psi_{ij}(Y_i, Y_j)$ stand for pairwise potentials, (i, j) ∈ (1,2),(2,3),(3,1)

.

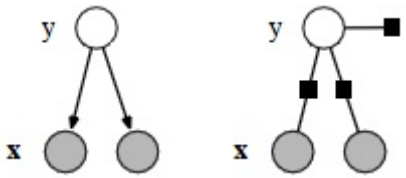

Figure 2.2: A simple directed graphical model (left) and its factor graph (right).

models make independency assumptions for data and these assumptions generally result in poor classification performance [69]. In general, remote sensing data which is likely to have interdependent features is not suitable to be modeled with directed graphical models.

On the other hand, a general undirected graphical model may not be sufficient to represent remote sensing data which exhibits strong spatial dependencies as well. For instance, Markov random fields only model dependencies between observation of a site X_i and its own label Y_i as well as labels of neighboring sites Y_i and Y_j. This means that sites are unaware of the observations of their neighbors. Thus, spatial updates of MRF only come from the labels of the neighboring sites and it only has label smoothing effects. However, conditional random fields take into account observations of neighboring sites as well and in the extreme case of whole image observations. Thus, CRF approach can be claimed to be unrestricted compared to MRF approach.

In conclusion, conditional random fields are more suitable for modeling and analysis of remote sensing data. In the following section, a brief overview of conditional random fields is provided.

2.4.1 Conditional Random Fields

Conditional Random Fields concept is proposed by Lafferty in 2001 for the sequence models in natural language processing domain [50]. Kumar and Hebert introduced 2-D version of this model into image domain under the name of *Discriminative Random Fields* [49]. In this thesis, gone with the original naming and 2-D random fields are referred as CRFs.

The general form of a CRF is given as follows:

$$P(\mathbf{Y} = \mathbf{y}, \mathbf{X} = \mathbf{x}) = \frac{1}{Z(x)} \exp\left\{-\sum_{c \in C} \psi_c(\mathbf{y}_c, \mathbf{x})\right\}, \tag{2.9}$$

where $\psi_c(\mathbf{y}_c, \mathbf{x})$ is a factor or *(log-space) clique potential* defined over random variables $\mathbf{Y_c} \subseteq \mathbf{Y}$, or clique c and $\mathbf{y}_c$ are the assignments to these variables.

The sum over the factors, or potential functions, is known as *energy function* and can be formalized as $E(\mathbf{Y}, \mathbf{X}) = \sum_c \psi_c(\mathbf{Y}_c, \mathbf{X})$. Then partition function can be written as $Z(\mathbf{X}) = \sum_{\mathbf{Y}} \exp\left\{-E(\mathbf{Y}, \mathbf{X})\right\}$.

As a fundamental step of conditional random fields, the field definition should be given. In computer vision applications, CRFs are defined over image sites such as pixels, segments or windows. After the field is defined, unary and pairwise potentials of the sites in the field are defined. Energy function in terms of unary and pairwise potentials can be formulated as

$$E(\mathbf{Y}, \mathbf{X}) = \sum_i \psi_i(Y_i, X_i) + \sum_{(i,j) \in \varepsilon} \psi_{ij}(Y_i, Y_j, \mathbf{X}) \tag{2.10}$$

where ε is the set of adjacent variable pairs.

In this study, potential functions are parameterized as log-linear models. Unary potential represents the likelihood probability of sites to take a certain label given the observed data as;

$$\psi_i(Y_i, X) = \log P(Y_i|X) \tag{2.11}$$

and

$$\psi_c(\mathbf{Y}_c, \mathbf{X}, \theta_c) = \theta_c^T \phi(\mathbf{Y}_c, \mathbf{X}). \tag{2.12}$$

where $\phi(\mathbf{Y}_c, \mathbf{X}) \in \mathbb{R}^n$ is fixed joint feature function and $\theta_c \in \mathbb{R}^n$ are parameters to be learned.

Pairwise potential represents how the neighboring labels Y_j and data X effect the label Y_i at site i,

$$\psi_{ij}(Y_i, Y_j, X, \theta_{ij}) = \mathbf{Y}_i \mathbf{Y}_j \theta_{ij}^T \phi_{ij}(\mathbf{Y}_i, \mathbf{Y}_j, \mathbf{X}). \tag{2.13}$$

Here $\phi_{ij}(\mathbf{Y}_i, \mathbf{Y}_j, \mathbf{X})$ can be defined as the concatenation or difference of unary potentials as suggested by Kumar and Hebert [49]. On the other hand, rather than using a function of unary potentials, totally different features can be extracted for adjacent nodes and they can be used as pairwise potential.

2.4.1.1 Parameter Estimation

The parameters ϕ_i and ϕ_{ij} are the weights introduced in the previous sections for unary and pairwise potentials respectively. Let $\theta = [\theta_i^T, \theta_{ij}^T]^T$ represent the whole parameter set. These parameters are estimated from the training set X = x_1, ..., x_M, where M is the number of training samples and Y = y_1, ..., y_M are the corresponding class labels. For estimating θ, negative log-likelihood, $-L(\theta) = -log(P(\theta|Y, X))$, is minimized in a gradient descent fashion by the help of limited-memory variant of Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [53] with either exact inference or loopy belief propagation which is an approximate inference method.

BFGS method is an optimization algorithm for solving unconstrained nonlinear problems. BFGS method is one of the quasi-Newton algorithms. Quasi-Newton algorithms are hill-climbing optimization techniques which search for optimal points in the search space by checking the first and second derivatives. BFGS method is quite popular among these algorithms and its limited-memory variant is suitable for problems with over 1000 variables [53].

Let us formalize the entire parameter estimation procedure by simplifying the problem to a binary case. Given a set of training examples M, $P = \left\{x^{(i)}, y^{(i)}\right\}_{i=1}^{M}$ we want to find the θ^* that maximizes the likelihood of the observed data. We will learn a model of the conditional probability P(Y|X) discriminatively, by making use of a logistic function

$$P(Y|X) = \frac{1}{1 + \exp^{-\theta^T x}} \tag{2.14}$$

$$\theta^* = \underset{\theta}{argmax}\, L(\theta; D) = \underset{\theta}{argmax} \prod_{m=1}^{M} P(y^{(m)} \mid x^{(m)}; \theta) \tag{2.15}$$

Equivalently, we can minimize the negative log-likelihood,

$$\theta^* = \underset{\theta}{argmin}\ -L(\theta; D) = \underset{\theta}{argmin} \sum_{m=1}^{M} -logP(y^{(m)} \mid x^{(m)}; \theta). \tag{2.16}$$

Estimating the parameters θ^* can be accomplished minimizing the negative log-likelihood. For this purpose, we can employ stepwise optimization techniques such as gradient descent, as also employed in this thesis. The key intuition is that by taking steps in the direction of the negative gradient of $-L(\theta; D)$ which is the direction of steepest descent and we will eventually converge to a minimizer of $-L(\theta; D)$ because the negative log-likelihood (cost function) of the CRF is convex [50].

Let us further simplify the model by collapsing all feature functions of both unary and pairwise in order to generalize also triplet and quadruple feature functions etc. Let $\phi(D_i) \in^n$ be any feature function where D_i is the set of variables in the scope of the i^{th} feature. Each

feature has an associated weight θ_i and given the features $\left\{\phi_i\right\}_{i=1}^{n}$ and the weights $\left\{\theta_i\right\}_{i=1}^{n}$, the distribution is defined as

$$P(Y \mid X; \theta) = \frac{1}{Z(\theta)} \exp \sum_{i=1}^{n} \theta_i \phi_i(D_i) \tag{2.17}$$

and partition function $Z(\theta)$ as follows,

$$Z(\theta) = \sum_{Y} \exp \left\{ \sum_{i=1}^{n} \theta_i \phi_i(D_i) \right\}. \tag{2.18}$$

As mentioned above our cost function is simply the negative log-likelihood which is

$$nll(Y, X; \theta) = log(Z(\theta)) - \sum_{i=1}^{n} \theta_i \phi_i(Y, X) + \frac{\lambda}{2} \sum_{i=1}^{n} \theta_i^2. \tag{2.19}$$

Note that the last term of equation 2.19 is the L_2 regularization term in order to control overfitting of the model by adjusting λ parameter. In order to employ gradient descent we have to calculate partial derivatives which is in the form:

$$\frac{\partial}{\partial \theta_i} nll(Y, X; \theta) = E_\theta[\phi_i] - E_D[\phi_i] + \lambda \theta_i. \tag{2.20}$$

Here we have two expectations in equation 2.20. $E_\theta[\phi_i]$ is the expectation of feature values with respect to model parameters and $E_D[\phi_i]$ is the expectation of the feature values with respect to the data instance D. By definition we have

$$E_\theta[\phi_i] = \sum_{Y'} P(Y^{'} \mid X) \phi_i(Y^{'}, X), \tag{2.21}$$

$$E_D[\phi_i] = \phi_i(Y^{'}, X). \tag{2.22}$$

In equation 2.21, we sum over all possible assignments to the Y variables in the scope of feature ϕ_i. However, the computational burden of CRFs are rooted from equation 2.21 as well. Computing the conditional probability $P(Y^{'} \mid X)$ for each assignment requires performing inference for the data instance X. Also a stepwise optimization such as gradient descent inference has to be performed in each iteration for each sample, which makes CRF training hard and prone to improvement. Lastly by following the parameter update rule $\theta := \theta - \alpha \bigtriangledown_\theta (-logP(Y \mid X)\theta)$, models parameters can be estimated correctly at the convergence.

2.4.1.2 Inference

Inference is generally defined as learning a model from the data for estimating the posterior probability $P(C_k \mid X)$ of each class, k = 1, ..., n. [8]. In CRF models, inference is finding the normalization constant Z and the marginal probabilities of each node for taking each state. After learning the model parameters θ by parameter estimation, maximum a posteriori (MAP) assignments of the random variables Y are inferred by energy minimization,

$$\underset{Y}{argmax}\, P(Y \mid X) = \underset{Y}{argmin}\, E(Y, X). \tag{2.23}$$

In this thesis, during star-CRF experiments where only upto seven nodes exist in the graph, exact inference is used. However, Loopy Belief Propagation (LBP) [24, 56] is used when the field is established over segments, since the excessive number of nodes hinders the usage of exact inference. Exact inference is a brute force method and as the number of nodes increases the computation becomes intractable.

LBP uses messages to update the probability of neighboring nodes even when there are cycles in the graphs. Although there is no guarantee for LBP to converge, in practice LBP is commonly used and gives satisfactory results. LBP is appreciated in terms of time complexity concern and generally preferred rather than other approximate inference methods for its simplicity [69].

As θ estimated, labels of test samples can be predicted by maximizing $P(Y \mid X)$ which is obtained at inference step.

2.4.1.3 Parameter Sharing

If the nodes in the graph have the same states and the lattice is homogeneous, the same parameters can be used across all the nodes. In this case, number of parameters to be estimated is much less than the full-parametrization case. Hence, the complexity of the model decreases. However, for some tasks, parameter sharing may not be appropriate, since different nodes which take the same state may be interpreted differently. Therefore, according to the task and design of the graphical model, parameter sharing issue should be addressed.

Let $\phi^{(i)}$ be the set of features that share parameter θ_i. Then our original CRF negative log-likelihood equation 2.19 can be expanded as below:

$$nll(Y, X; \theta) \equiv log(Z(\theta)) - \sum_{i=1}^{n} \theta_i \left(\sum_{j \in \phi^{(i)}} \phi_j(Y, X) \right) + \frac{\lambda}{2} \sum_{i=1}^{n} \theta_i^2 \tag{2.24}$$

and the partial derivatives can be calculated as

$$\frac{\partial}{\partial \theta_i} nll(Y, X; \theta) = \sum_{\theta_j \in \{\phi^{(i)}\}} E_\theta[\phi_j] - \sum_{\theta_j \in \{\phi^{(i)}\}} E_D[\phi_j] + \lambda \theta_i. \tag{2.25}$$

The main difference with the full parameterized model is that in the parameter sharing case θ_i is not necessarily related with ϕ_i any longer. Instead θ_i is associated with a set of features $\phi^{(i)}$. n is the total number of parameters and not necessarily the total number of features ϕ_j.

2.4.1.4 Decoding

Decoding is defined as estimating the class labels of a test sample. During testing phase, the same inference method is applied as in the training phase. After obtaining conditional probability estimations of each class, maximum a posteriori (MAP) method is used to assign the class label to a test sample.

2.4.2 Methodologies for Modeling Spatial Structures

Studies modeling spatial structures vary both in their representations used to encode the spatial information and their approaches for learning. Inference on the generated graphical models depends on the model selection and may be considered as a representation dependent step.

In [38], Hoberg and Rottensteiner applies a basic CRF to IKONOS 4m multispectral image over varying size of windows. They aim to classify settlement areas and show that even with basic features a CRF classifier which incorporates spatial relations outperforms a traditional classifier such as maximum likelihood classifier. For feature extraction, they obtain the gradient image first. Then, they construct a histogram by summing the responses in each selected direction. Mean and variance values of peak bins in this histogram and number of peak bins over the mean value of the histogram are used as features. As pairwise potential, difference of unary potentials is used. Hoberg and Rottensteiner state that loopy belief propagation (LBP) inference and Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm for parameter estimation is a frequent combination in CRF studies and they also employ these methods in their study [38].

A similar study is conducted in [6], for urban area detection in satellite images, where a CRF model is used for integration of cues such as color, texture and edges. After quantizing color and texton filter-bank responses, they apply joint boosting to obtain discriminative features. They also add edge cues and obtain unary features. Although, the unary features are more diverse, the fundamental lattice formation over segments stays the same. As for the inference step, they follow a different algorithm, namely Swendsen-Wang cut algorithm [77].

One of the pioneering studies which employs both contextual and hierarchical representations with a relationship learning process and Bayesian inference algorithm, is proposed by Porway

et. al [1]. Their approach presents a grammar-based hierarchical and contextual model for object recognition. This grammar-based model combines a stochastic context free grammar [52] with a Markov Random Field (MRF) to capture both local and global context and combines bottom-up information with top-down knowledge. They represent the frequency of occurrence and type of object parts with a stochastic context free grammar and model the spatial and appearance relationships between them using MRFs, thus create a constrained grammar that can represent a huge number of instances for a single category. Another contribution of this study is that, this contextual and hierarchical model learns statistical constraints on the appearances and relationships between different parts of the image classes in a minimax entropy framework [80]. This framework selects the set of contextual relationships necessary for modeling the object class; begins with a large set of relationships that could potentially exist between parts, then iteratively selects only those relationships that help the model best match true statistics for that image class. They separated hierarchy into two sets for objects and scene which enables to plug-in any object detection algorithm for bottom-up detection procedure. They employed compositional boosting [78] for some specific bottom-up proposals.

In [28], a region and object based model for object-detection is proposed through a hierarchy of CRFs. In the bottom level, a CRF is comprised of pixels as probabilistic graphical model nodes and features are extracted in pixel level accordingly, a unified energy function made it possible to incorporate bottom-middle and top level random fields. In the middle level, segments are formed as the model nodes and contextual relations between segments are revealed with region statistics. Finally, as the top-most level of the proposed hierarchical graphical model, segments and objects are connected to each other and contextual relations between objects are extracted from positional relations of the objects both considering segment level interactions at once. The model employed for this graphical model is a conditional MRF (CRF) that is trained by labeled images from both levels with logistic regression and inference is conducted by use of hill-climbing.

Jiang et.al. propose a context based concept fusion model for semantic concept detection [42]. In [42], posterior probabilities for several classifiers are fed to a CRF model for generating updated posterior probabilities through a fully-connected CRF where each node represents a concept. This corresponds to class labels in our case.

Lee et.al. propose a model, namely support vector random fields, which combines the ability of CRFs to model different types of spatial dependencies and the appealing generalization properties of support vector machines (SVM) [39]. Their approach employs an observation-matching potential by changing the unary potential in CRF model. Therefore they combined the discriminative classification power of SVMs with spatial context encoding power of CRFs.

Bovolo and Bruzzone's work [9] is another attempt to combine the discriminative power of SVMs and the strength of graphical models to address spatial-contextual cues. They utilize the class conditional densities of multiple SVMs and feed them to a MRF model in order to obtain the final class labels.

In the study of Roscher et.al. [63], they suggest to use import vector machine, which is a probabilistic version of SVM, and feed the probabilistic output as potentials to the CRF. They claim that it is a quite powerful classifier combination for land cover classification.

2.5 Sparse Autoencoders

Autoencoders come from the family of neural networks. An autoencoder neural network is an unsupervised algorithm which employs backpropagation by setting the target values to be equal to the inputs [57]. In other words, an approximation of the identity function, $h_{W,b}(x) = \hat{x} \approx x$, is aimed to be learned. The semantic in this approach is to observe the structure in the data by enforcing some constraints during the approximation process. For instance, when spectral values of pixels in a 10x10 patch, meaning that there are 100 input units, are fed to an autencoder where the number of hidden units (s_2) are fixed to 50, we force the autoencoder to learn compressed representation of the input data. As commented by the Ng, this autoencoder presents the correlations in the data and learns low-dimensional representation similar to a principal component analysis (PCA) output [57].

On the other hand, if the constraint on the number of hidden unit is to be quite large than the number of input units, most of the units become inactive, in other words their output values approach to zero. This is called sparsity and the constraint of large number of hidden units is called the sparsity constraint. This sparsity constraint is embedded to the formalization of a neural network by introducing an additional sparsity regularization term as follows,

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{s_2} KL(\rho \parallel \hat{\rho}_j), \tag{2.26}$$

where J(W,b) stands for the energy term for neural networks and KL stands for Kullback-Leibler divergence [47] and $\hat{\rho}_j$ is the average activation of the hidden unit j.

Before applying a sparse autoencoder, the input data is to be passed through several pre-processing steps, namely contrast normalization and whitening. Contrast normalization is accomplished by mean subtraction. It is then followed by whitening operation. Whitening transformation is a preprocessing step which decorrelates the covariance matrix M of a set of random variables to identity matrix of a new set of random variables. After applying whitening operation over each sample, adjacent pixels become less correlated and we manage to remove aliasing artifacts in the image which can improve the features learned [16]. Normalization and whitening steps carry significant importance while learning sparse autoencoders.

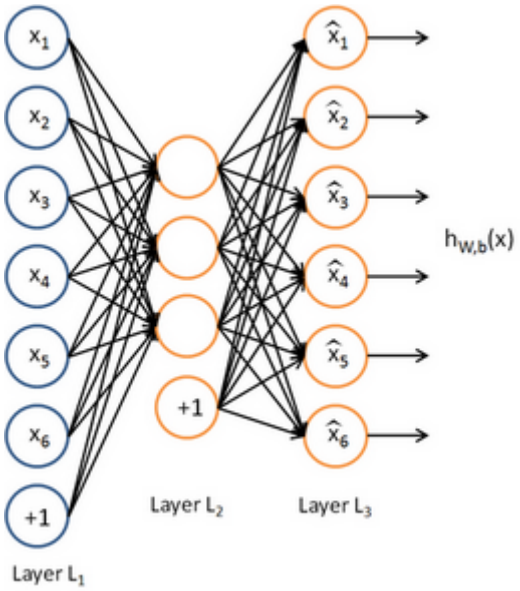

Figure 2.3: A sparse autoencoder example [57].

2.6 Chapter Summary

In this chapter, firstly, challenges encountered during classification of remote sensing area are discussed. Then, the computer vision methods, such as segmentation and feature extraction, which are fundamental steps employed to remote sensing data before classification are overviewed. Then, object classification task in remote sensing is formalized and principles of probabilistic graphical models, especially conditional random fields, are presented. After the brief introduction about conditional random fields, some of the methodologies in the literature which share some basics to the proposed method in this thesis are pointed out. Lastly, sparse autoencoders, which is utilized in this study for extracting representative character of a target area, is briefly explained.

CHAPTER 3

A CONTEXTUAL MODEL WITH CONDITIONAL RANDOM FIELDS

This chapter introduces the proposed contextual conditional random fields approach which is applied to remote sensing images. First of all, overview of the proposed system is provided. Then, a new concept to represent the context information about a specific target is defined in the framework of conditional random fields (CRF). Finally, a new model of CRF proposed which integrates the context information into a simple CRF topology, called star structure.

3.1 System Overview

In this thesis, a contextually consistent target classification scheme for satellite imagery, is proposed. For this purpose, a contextual conditional random field (CRF) model is developed. This chapter introduces our contextual target classification model which is a three-stage algorithm as shown in Figure 3.1.

At the first stage, the most discriminative feature of the target class is determined by sparse autoencoders presented in Chapter 2.5. Then, image is filtered to capture the areas with this discriminative feature and they are named as candidate regions which are also represented as the central node for our model. In our model, for each candidate region, a separate conditional random field is solved to determine whether the candidate region belongs to the target class or not.

The second stage is performed in order to obtain land use/land cover (LULC) class nodes in the conditional random fields which are formed around each central node with a predefined radius parameter. After segments are obtained by mean shift segmentation, extracted features from these segments are fed to support vector machine (SVM). Hence, class labels of segments, which can be water, forest, agricultural land, urban land, soil and concrete, are obtained. Segments with the same class label are combined into a single region, called meta-segments. In this manner, LULC classes can be used as concepts during the third stage. At the third stage, each of the combined regions represents a node in the conditional random field of that candidate in a neighborhood.

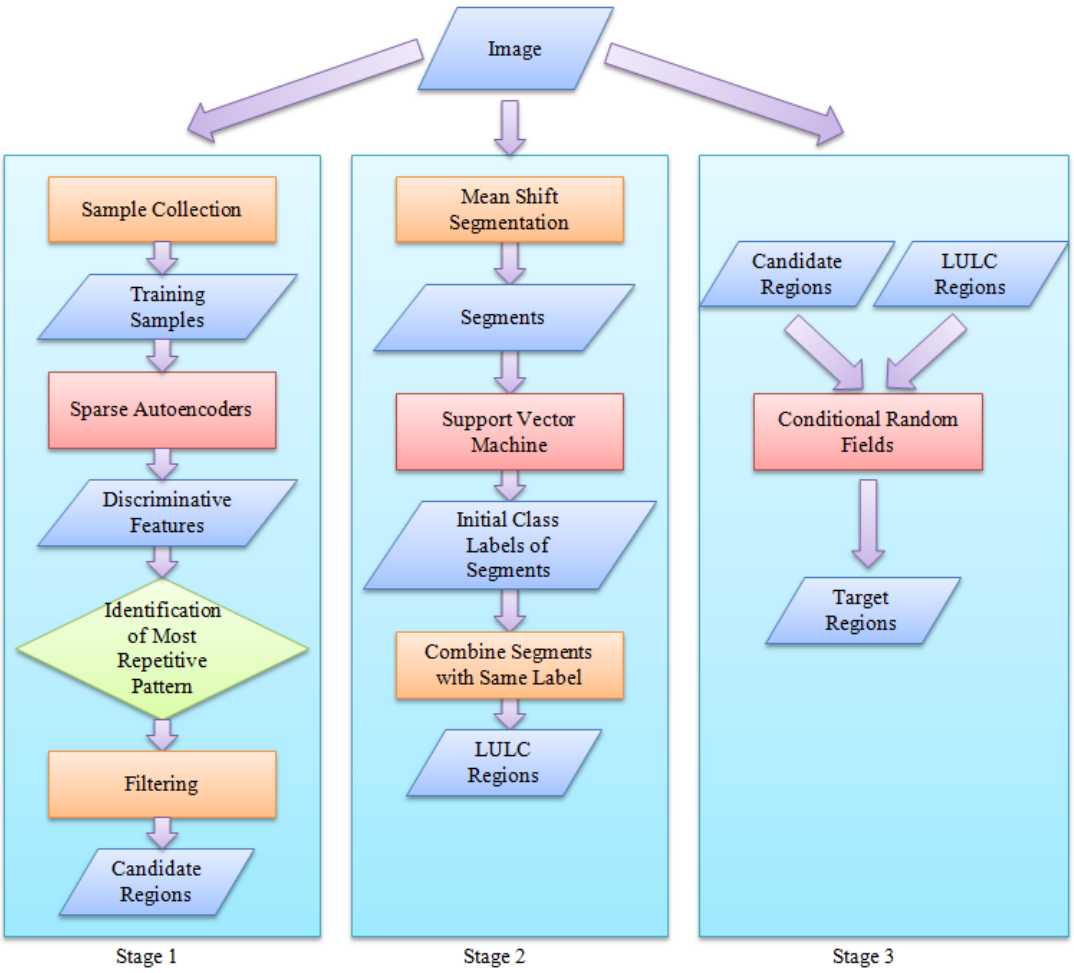

Figure 3.1: Flow chart of the proposed contextual conditional random fields model.

In the CRF framework, for each candidate region, a conditional random field is constructed such that the candidate region is the central node and combined regions of LULC classes in the neighborhood of that candidate region are the remaining nodes. Features are extracted from the candidate region and from combined class regions. These features are used for obtaining association potentials. As for the pairwise potentials, co-occurrence frequencies of LULC classes with the candidate region in three different proximity scales, namely adjacency, close neighborhood and context neighborhood, are used.

In this thesis, a target class is expected to have spatial contextual characteristics. For this reason, in this chapter, the notion of context is discussed first. Secondly, the methodology for defining the context of a target class is studied. Contextual cues about a target class can be gathered from domain knowledge and can be transformed into rules. On the other hand, in this thesis, we have demonstrated that rules that come from domain knowledge can be verified by machine learning algorithms, such as sparse autoencoders. Then, the neighborhood concept where the context is defined is explained. The main issue is to combine the contextual cues of the target and information that comes from the neighborhood. In the last section, the conditional random fields to achieve this task is introduced.

3.2 What is Context?

Context is a powerful cue in computer vision research and frequently associated in classification tasks. However, its formalization is hard. Turney and John et.al. make a formal definition of context in terms of *subsets of strongly or weakly relevant features* for supervised machine learning algorithms [73, 43]. They claim that a feature is primary if classification can be conducted just based on this feature. The feature is contextual if it helps the classification, yet not enough by itself for assigning a class label and the feature is irrelevant if it does not help classification at all.

Although the above suggestions are well formalized, definition of context alter for each problem domain. In the computer vision literature, a common point about context definitions is "any and all information that may influence the way a scene and the objects within it are perceived" [11]. Biederman's definition, "meaningful visual information comes to us in the form of scenes", has an impact on the formation of this common point [12]. Biederman defines context by examining relations between an object and its surroundings. According to Biederman's categorization of contextual relations, objects in a scene can be interpreted in terms of i) interposition, ii) support, iii) probability, iv) position and v) size. The first two relations can be coded by the physical space of the objects in the image. Last three relations, however, can be used to define context of objects according to each other.

Recent studies show that contextual features can be grouped into three categories; namely, semantic context (probability) which defines context as its co-occurrence with other objects in the scene, spatial context (position) which is the likelihood of existence of an object in some position with respect to other objects in the scene and scale context (size) which defines

context based on the scales of an object with respect to other objects in the scene [25].

However, there are many other types of context such as local pixel context, photogrammetric context (camera height, resolution etc.), illumination context (sun direction, sky color, shadow contrast, etc.), weather context, geographic context (GPS location, terrain type, etc.), temporal context(capture time, alongside frames, etc.) and cultural context (photographer bias, dataset selection bias, etc.) [10].

Context definitions can also be grouped into two levels, namely local and global, as discussed in [25]. Global context covers image as a whole (e.g. a highway in rush-hour indicates the presence of cars), local context considers context information from surrounding regions of the object(e.g. a car indicates the presence of the tires and windshield).

In this thesis, considering context in a local frame, semantic and spatial context are selected to interpret scenes in remote sensing images. Our definition of these concepts are as follows:

Definition 3.2.1 *Let A and B be random variables that represent two different objects (or areas). The semantic context of A is defined as either*

1. $p(B) \leqslant p(B \mid A)$ *and* $p(A) \leqslant p(A \mid B)$*, in other words, co-occurrence of A and B is quite likely and the existence of A gives a strong clue about the existence of B, or*
2. $p(B) \leqslant p(B \mid A)$ *and* $p(A) \leqslant p(A \mid B)$*, in other words, co-occurrence of A and B is quite unlikely and the existence of A gives a strong clue about the nonexistence of B in the scene.*

where p(.) indicates the probability of occurrence.

Definition 3.2.2 *Let A be a random variable that represents an area (or object), B be the set of random variables that support the likelihood of A, C be the set of random variables that support how unlikely of A to be in the scene. Let* ϵ_A *be a scalar for defining the neighborhood of the area A represents. Then spatial context of A is defined as;*

$$p(A) \leqslant p(A \mid B_i \in \epsilon_A,\ C_j \notin \epsilon_A)$$

, for any i and j. Meaning that, given the existence of a variable from set B and the nonexistence of a variable from C increase the occurrence probability of A to be in the scene. The following inequalities define the spatial context from the other way around,

$$p(B) \leqslant p(B \mid B \in \epsilon_A)\ ,\ p(C) \geqslant p(C \mid C \in \epsilon_A)$$

, likelihood of a variable from set B increases if given to be in the neighborhood of A, and similarly, likelihood of a variable from set C decreases if given to be in the neighborhood of A. In particular cases, there may be limits to these likelihoods as depicted below,

$$\alpha_{B_i} \leqslant p(B_i \mid B_i \in \epsilon_A)\ ,\ p(C_j \mid C_j \in \epsilon_A) \leqslant \alpha_{C_j}$$

, for any i and j. These limits, $alpha_{B_i}$ and $alpha_{C_j}$, can be defined specific to each problem and they can be acquired either from expert knowledge or be learned from data.

These definitions fit with the contextual feature definition of Turney [73]. Although, semantic and spatial context cues do not identify the class label, they are strongly relevant and support the likelihood of certain classes according to the definitions above.

The reason to avoid global contextual cues, such as geographic information and azimuth angle etc., is unavailability for some scenes. If metadata information of images are present, these global contextual cues can be incorporated to the classification process as well.

3.3 Whose Context?

Remote sensing objects or areas may seem to obey certain contextual rules according to objects in natural images. For instance, if a car in a natural image is assumed to be always on the road, a promotion car on the wall of a building, or cars in a multi-storey parking lot falsify this statement. Although more unlikely, a car in a satellite imagery can be observed on the top of a building rather than on the road as well. Thus, if the context of a remote sensing object is defined to comply with a specific rule, this context definition may fail for some unseen samples. However, this is generally more unlikely than the natural images and for some objects or areas in satellite images, certain rules are always followed by the nature of these objects or areas. In this thesis, such objects or areas are called as *contextually consistent*.

For recognizing an object, the main point is to capture the *invariance* of the object which is generally quite easy for human eye. Invariance is an important concept and generally hard to pinpoint. In an abstract point of view, invariance V of an object is the set of cues / features / definitions / properties that do not change regardless of the random changes in semantic and spatial context for the object of interest. For airfields, invariance is defined as long straight parallel lines in [22]. This intuition comes from the domain knowledge due to the function and nature of the area. Such invariants can be perceived as *compositional context* of an object as well. Hence, we consider this invariance as part of compositional context of airfield class, and obtained parallel straight lines longer than 1 kilometer by the help of an external line segment detector algorithm [76].

For the showing the statistical foundation of this choice, we have conducted experiments making use of sparse autoencoders. Representative features are obtained for the airfield class by feeding training samples to a single-layer sparse autoencoder. The visualization of the results can be observed in Figure 3.2. Each small square corresponds to the weights learned in one of the hidden units of the sparse autoencoder. Each pixel in each square represents the weight learned in the corresponding hidden unit for one of the inputs units.

As seen in Figure 3.2, the most repetitive feature is parallel lines, though there exist other features as shown, namely departure end and wheel track. For simplicity, squares that corre-

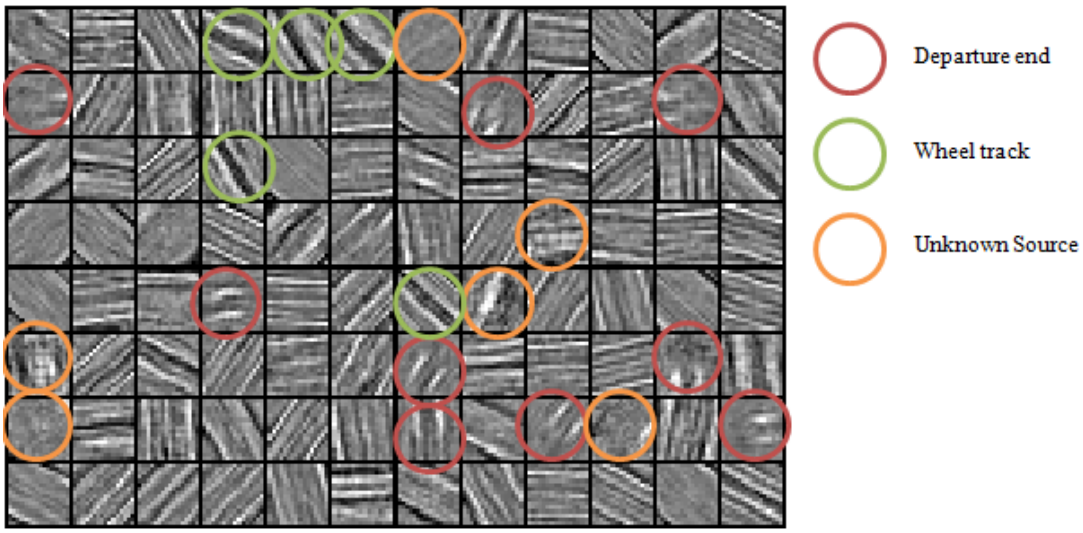

Figure 3.2: Visualization of the results obtained by sparse autoencoder for airfield class.

spond to parallel lines are not marked in Figure 3.2. Hence, the invariance or compositional context of the airfield class can be defined as a set of these visual cues, as statistically demonstrated by the sparse autoencoders. In this study, only the most repetitive feature, which are parallel lines, is utilized for simplicity.

As a further demonstration, learned representations are used to convolve. A test image and its response images are superposed in order to obtain the regions that have the highest response. As expected, only the runway regions and some main taxiroutes gave a response over an empirical threshold. As can be observed from Figure 4.5, the detected regions are all similar to the appearance of runways.

3.4 Where to Search for Context?

Given a scene which contains a characteristic object, its label can be inferred by analyzing its contextual relations within the scene. These relations can be formed over the surrounding related classes (local) or formalized over the entire scene (global). In this thesis, local contextual relations are investigated.

As the first step, pixels in an image are clustered into segments by mean shift algorithm. These segments are classified according to features which are non-informative about any contextual cues. Table 3.1 summarizes the features used in this classification step. Detailed information about the features can be found in section 2.2.1.

This step is to obtain the very first posterior labels. Most of the studies in literature stop after this step. Nevertheless, performance results may be poor if context is ignored. Context play

Table 3.1: Feature descriptions.

FEATURE	DESCRIPTION	ADDRESSED CLASS
NDVI (mean, variance and ratio of pixels over the dynamic threshold in segment)	Normalized difference vegetation index map obtained by the difference ratio of near-infrared and red bands	Forest Agricultural Land
NDWI (mean, variance and ratio of pixels over the dynamic threshold in segment)	Normalized difference water index map obtained by the difference ratio of near-infrared and green bands	Water
Elevation (variance and Kullback-Leibler divergence of elevation histogram from a uniform distribution)	Interpolated digital terrain elevation data	Airfield
Gabor response (mean, variance and ratio of pixels over the dynamic threshold in segment)	Single response map obtained by superposing Gabor filter responses in 8 directions	Urban land
Spectral (mean and variance in segment)	Red, green, blue, and near-infrared band values	All

an important role for the distinction of objects with similar spectral, textural or shape features. In this study, we suggest a second step which updates posterior labels of first step according to context.

Given a segmented image and a candidate region of characteristic object, segments with the same posterior label in the neighborhood of the candidate are combined and formed *meta-segments*. For a candidate region, number of meta-segments in its neighborhood can range from 1 to n, as n is the number of classes used in the initial classification (LULC classes).

The neighborhood concept here is also varying. It can vary from zero to half of the image size in terms of pixels. Let us examine three cases for the neighborhood selection. If it is selected as zero, only the meta-segments which lie under the candidate area would be considered during second step. If it is selected as the half of the image size, all meta-segments in whole image would be taken into account. This can be considered as one type of global contextual information (other global information could be image parameters such as acquisition time and azimuth angle for remote sensing images). If the neighborhood is selected as between these extreme cases, meta-segments which only exist in that neighborhood would be considered during formation of the contextual model around the candidate. This is the local contextual influence examined in this thesis. Since using distant segments would not bring any significant contextual information, an acceptable neighborhood, 600 meters around the airfield, which

comes from domain/expert knowledge is used.

Let the set of existing meta-segments in the selected neighborhood is $\{m_1, ..., m_e\}$, as it ranges from 1 to n. Conditional random field model is formed over this set of meta-segments and the candidate itself. The following section presents the details of the proposed model.

3.5 How to Search for Context?

In [62], studies integrating context for scene parsing problem are categorized into three groups, namely rule-based approaches, additional features approaches, and graphical model approaches. Scene parsing problem is similar to our problem, although we are only interested in the final label of the target area and not interested in the labels of other segments. Roncevic states that rule-based approaches, which integrate context as hard-coded rules during classification tasks, such as [68], are easy to interpret for humans, yet require extensive effort for hand-coding each rule and they are abandoned nowadays. Approaches in another category consider context as an additional feature alongside the appearance features during classification. Recent artificial neural networks approaches, such as Convolutional Neural Network (CNN) [30] and Recursive Neural Network (RNN) [66], belong to this category as well. The advantage of the approaches in this category is the ability to add contextual features easily. However, contextual interactions are needed to be simplified in most cases. Artificial Neural Networks can handle contextual interactions without simplifying, yet extending the architecture for new labels are quite hard in these networks and there is little control on how to learn contextual features in such systems [62].

As stated in [62], probabilistic graphical models are frequently used for including context in classification tasks. Due to their nature, graphical models are capable to represent contextual relations quite conveniently. Especially, conditional random fields allow quite complex interaction functions between nodes and hence becomes superior over other candidates such as Markov random fields.

After visiting the existing methodologies that integrate context into the classification, from here onwards, we introduce our contextual star conditional random fields model (CS-CRF).

3.6 Topology of CS-CRF

Designing the topology is a fundamental step for constructing a probabilistic graphical model. Recently, the topology of graphical models are designed by structure learning methods in computer vision. However, it requires processing massive data in order to comprehend the underlying model [45]. Therefore, rather than learning the structure, we propose a star topology in order to represent the contextual aspect.

Let us introduce the basic concepts for CS-CRF, defined over a graph $G = (N, E)$, where N

Table 3.2: Integration of context for scene parsing tasks in recent literature[62]

SYSTEM	CONTEXT LEVEL	CONTEXT TYPE	CONTEXT INTEGRATION
Socher et al., 2011 [66]	Local	Unknown	RNN
Tighe and Lazebnik, 2010 [70]	Global/Local	Spatial/ Semantic	MRF
Grangier et al., 2009 [30]	Local	Unknown	Deep CNN
Gould et al., 2008 [29]	Global/Local	Spatial/ Semantic	CRF
He et al. 2004, 2008 [36, 35]	Global/Local	Spatial/ Semantic	Multiscale CRF/ Mixture of CRF
Rabinovich et al., 2007 [61]	Global	Semantic/ External	CRF
Kumar, 2005 [48]	Global/Local	Spatial/ Semantic	Hierarchical Discriminative CRF
Carbonetto, 2003 [12]	Global/Local	Spatial/ Semantic/ External	MRF
Strat and Fischler, 1991 [68]	Global/Local	Spatial/ Semantic/ External	Rule-based

represents the set of nodes and E represents the set of edges. Set of nodes in G consists of namely the seed node n_0 and the surrounding nodes $\left\{n_i\right\}_{i=1}^{c}$. Definitions of seed node and surrounding nodes are given as following:

Definition 3.6.1 ***Seed Node:*** *The central node n_0 in the star topology is named as seed node, since the context of the seed node is the basis for the model. Seed node in our model represents the candidate region of the target class.*

Definition 3.6.2 ***Surrounding Node:*** *In the star topology, a surrounding node n_i is the node which is not the seed node and whose task is to support the seed node. A surrounding node in CS-CRF represents one of the LULC class regions, in other words a meta-segment, in the neighborhood of the candidate region.*

Definition 3.6.3 ***Edge:*** *An edge between two nodes of CS-CRF represents the contextual interaction of two corresponding regions in the image. In star topology, contextual interactions are limited to between a seed node and a surrounding node. Contextual interactions between two surrounding nodes, are omitted in CS-CRF, since we are only interested in the final label of seed node and the messages to be passed during inference is restricted to flow over seed node which has created the model with its contextual influence.*

Graphical comparison of star model with a fully-connected model can be observed in Figure 3.3 where $n_0 = C_t$ and $n_i = \{C_1, C_2, C_3, C_4, C_5, C_6\}$.

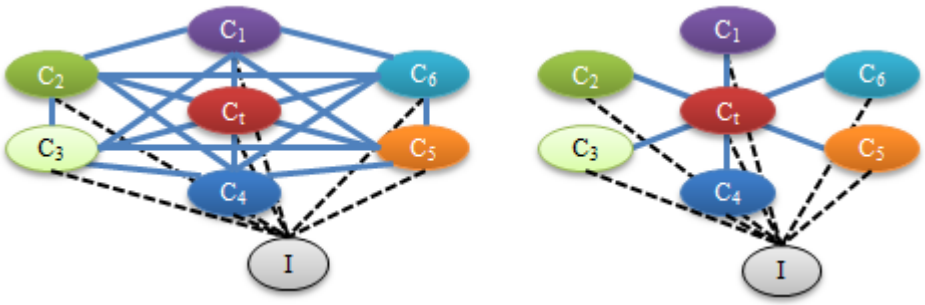

Figure 3.3: Fully connected CRF model (left), star CRF model (right).

The formulation difference arise from the adjacent pairs set which contribute to the energy function. In fully-connected model, adjacent pair set contains all pair combinations of nodes, $\varepsilon(i, j) = \{(1, 2), (1, 3), ..., (1, c + 1), (2, 3), ...(c, c + 1)\}$ and number of pairs is $\frac{c(c-1)}{2}$. In star model, set of adjacent pairs is $\varepsilon(i, j) = \{(1, c + 1), (2, c + 1), ..., (c, c + 1)\}$ and number of edges is n, as there are n meta-segments representing LULC classes and $(c + 1)^{th}$ node is the central node representing the candidate region of the target class.

3.7 Energy Function of CS-CRF

The proposed star model follows the definition in Section 2.4.1. The energy function in our model can be formalized as

$$E(\mathbf{Y}, \mathbf{X}) = \sum_i \psi_i(Y_i, X_i) + \sum_{(i,j)\in\varepsilon} \psi_{ij}(Y_i, Y_j, \mathbf{X}) \tag{3.1}$$

where $\psi_i(Y_i, X_i)$ stands for unary potential of Y_i in the clique i and $\psi_{ij}(Y_i, Y_j, \mathbf{X})$ stands for pairwise potential of Y_i and Y_j.

Unary potential in the proposed model is

$$\psi_i(Y_i, X_i, \theta_i) = \sigma(Y_i \mid \theta_i, \phi_i(Y_i, X_i)) \tag{3.2}$$

where σ stands for multi-class logistic classifier and ϕ_i is a feature function.

Pairwise potential in the proposed model is

$$\psi_{ij}(Y_i, Y_j, X, \theta_{ij}) = \sigma\left(y_{ij} \mid \theta_{ij}, \phi_{ij}(\mathbf{y}_i, \mathbf{y}_j, \mathbf{x})\right) \tag{3.3}$$

where σ stands for multi-class logistic classifier and ϕ_{ij} is another feature function.

3.8 Parameter Estimation of CS-CRF

In this section, two different approaches, adopted during parameter estimation, are specified. The first approach is full parameterization where all states of each node across different images are parameterized separately. With this approach, number of parameters to be estimated may be enormous, and parameter estimation may become computationally demanding. Let us consider a lattice generated from pixels in an image. As the number of pixels grow rapidly, estimating separate parameters for each pixel in its vicinity is neither computationally efficient nor there would be enough data for accurate estimation.

Second approach is parameter sharing. When there are random variables in the model such that they are replicated between and within models, they are called template variables and parameter sharing becomes possible. Figure 3.4 and Figure 3.5 demonstrate examples of sharing between and within models respectively. In these examples, a Bayesian network is constructed from a family tree where child nodes have directed edges coming from their parents' nodes. In Figure 3.4, corresponding Bayesian networks of a small family tree and a bigger family tree which contains the small family tree are depicted. As can be observed, the structure in the corresponding graph of the small tree is preserved in the graph of the bigger family tree. That is why it is called sharing between models.

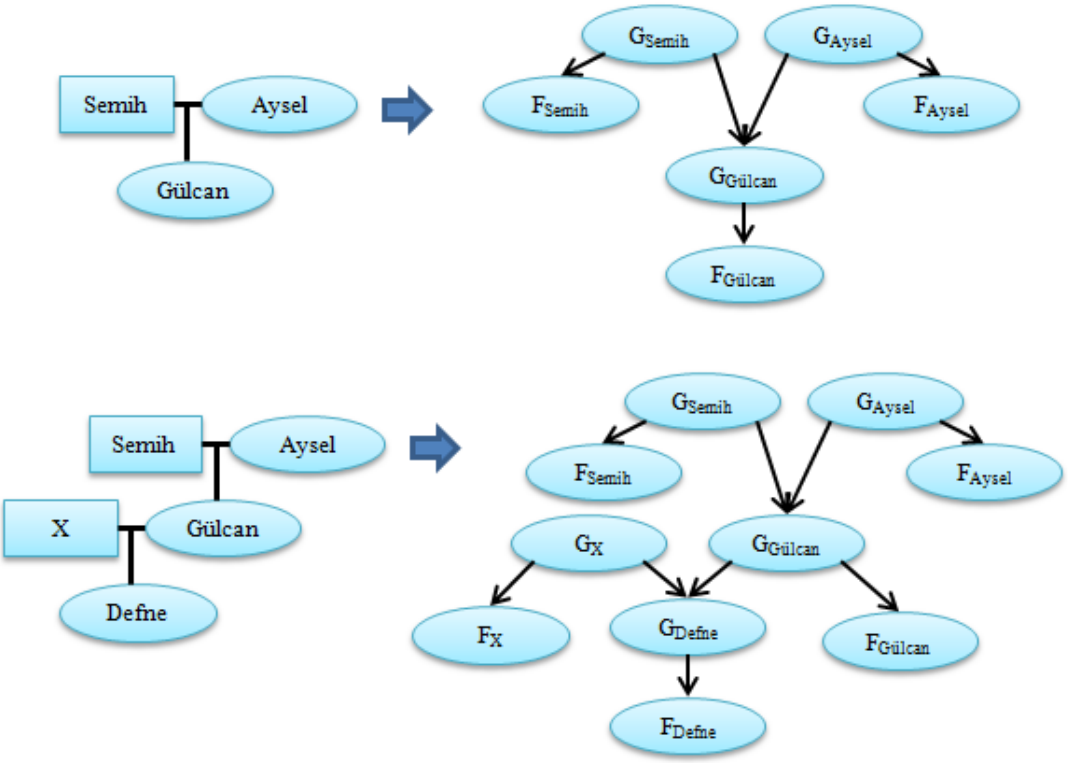

Figure 3.4: Example of sharing between models.

In Figure 3.5, a repetitive structure within the model, how two directed edges from parent nodes are connected to a child node, is depicted.

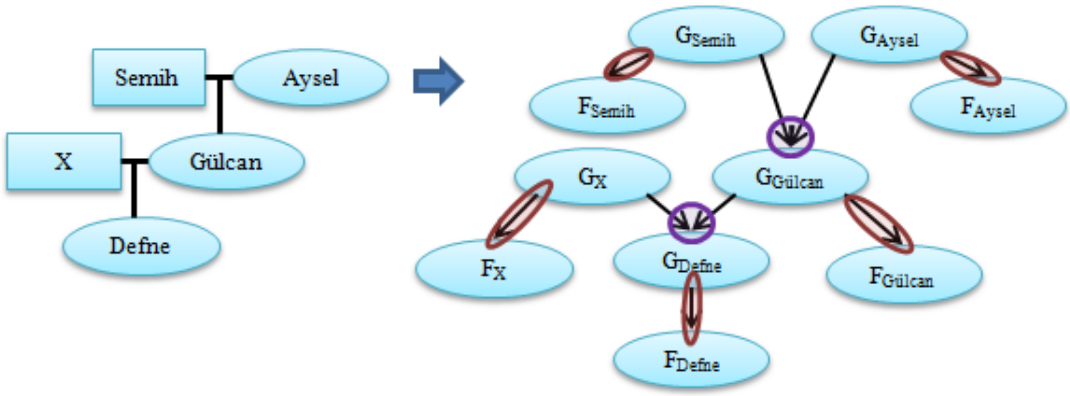

Figure 3.5: Example of sharing within a model.

Sharing is observed in our problem as well, in the forms of sharing across pairs of a node and its region, sharing across adjacent pairs of a candidate region and a meta-segment, and sharing across models obtained from different samples. Figure 3.6 illustrates within model sharing in our problem. As red node represents the seed node in our model, parameters of red edges between nodes are shared as well as parameters of edges between a node and its meta-segment.

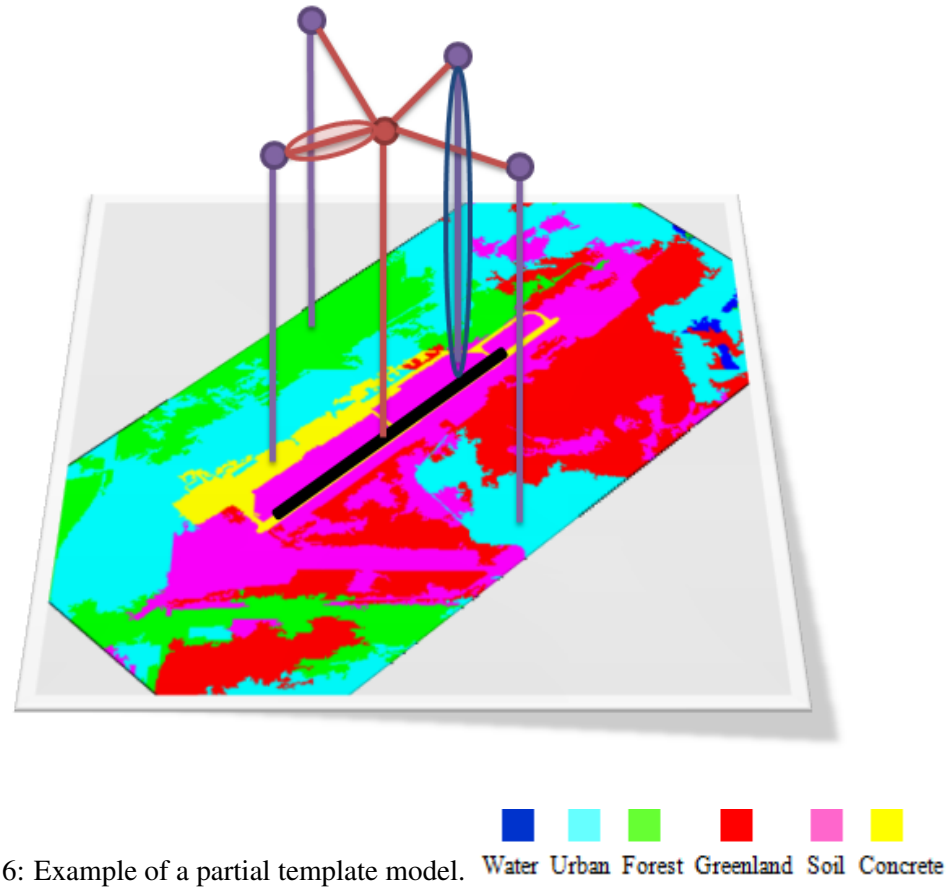

Figure 3.6: Example of a partial template model.

Figure 3.7 demonstrates differences of full parameterization and parameter sharing in the proposed star CRF model. In the parameter-shared star CRF model, there exists 7 states for each node, $y^{(t)} = \{1, 2, 3, 4, 5, 6, 7\}$. Meaning that, each node can be assigned to one of the seven classes. In other words, a soft-max classifier, which is the multi-class version of a logistic classifier, is used at the heart of the conditional random fields. For our problem, when parameters are shared across nodes and edges, number of parameters to optimize is reduced to 343 in total. In the fully parameterized version of the star model, each node has two states, meaning that its initial label is true or not, $y^{(g)} = \{0, 1\}$. This kind of representation is enough, since we are only interested in whether the candidate region belongs to the target class or not. When all the states of all the nodes have separate parameters, there exists 390 parameters in the proposed star model in total.

In our problem, number of node features, $|nf|$, is 21 with an additional bias feature and number of edge features, $|ef|$, is 4 with an additional bias feature. Number of nodes, c, is 7 and number of edges, $|\varepsilon(i, j)|$, is 6 for star model case and $\frac{c(c+1)}{2} = 21$ for fully-connected model. According to these values, Table 3.3 summarizes the number of parameters in discussed CRF models.

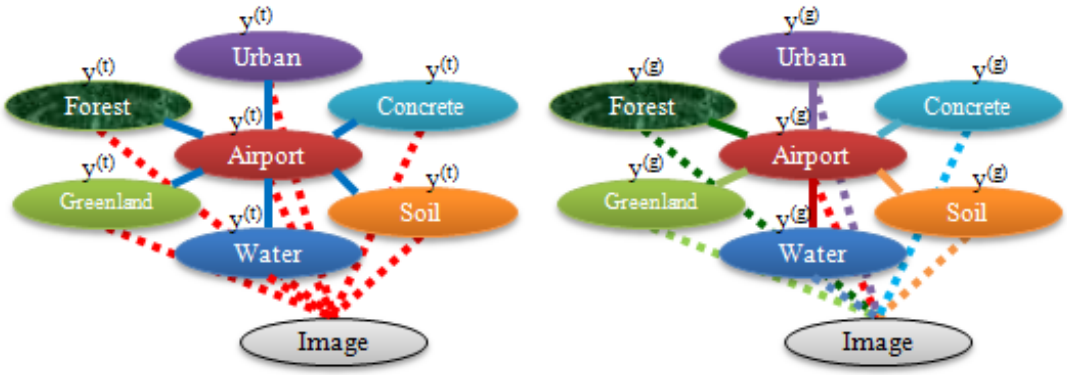

Figure 3.7: Template model and general model versions of star CRF model. Parameters associated with the edges of the same color correspond to shared parameters.

Table 3.3: Number of parameters and their calculations for fully connected and star models when parameters are shared or not.

Model	**Shared Parameters**	**How to Compute Number of Parameters**	**Number of Parameters**
Fully-connected parameter shared	node & edge parameters	$\|nf\| * \|y^{(t)}\| + \|ef\| * \|y^{(t)}\|^2$	343
Fully-connected full-parameterization	none	$c * \|nf\| * \|y^{(g)}\| + \|\varepsilon(i,j)\| * \|ef\| * \|y^{(g)}\|^2$	630
Star parameter shared	node & edge parameters	$\|nf\| * \|y^{(t)}\| + \|ef\| * \|y^{(t)}\|^2$	343
Star full-parameterization	none	$c * \|nf\| * \|y^{(g)}\| + \|\varepsilon(i,j)\| * \|ef\| * \|y^{(g)}\|^2$	390

3.9 Inference in CS-CRF

For computing a marginal probability, the joint probability distribution is to be summed or integrated over one or more variables. This computation can be performed as a sequence of operations by choosing a specific ordering of the variables, considering that the joint probability is a factored expression over subsets of the variables. Consequently, we can make use of the distributive law to move individual sums or integrals across factors that do not involve the variables being summed or integrated over. *Exact inference* is the process of summing probabilities while eliminating variables iteratively. Assuming that each individual sum or integral is performed exactly, then the overall algorithm yields an exact numerical result.

There are several other algorithms that produce exact marginals, but share intermediate terms in the individual computations, such as the sum-product and junction tree algorithms. These algorithms employ *message-passing* operations on graphs, where the messages are exactly these shared intermediate terms. When the convergence is reached, marginal probabilities for

all cliques of the original graph are obtained.

In the proposed star CRF model, since the graph is quite small, utilizing exact inference does not bring any hindrance. Therefore, for solving the proposed star CRF model, exact inference is used during both learning and testing steps.

3.10 Contributions of CS-CRF

In our study, a contextual conditional random field model is proposed for the classification of contextually consistent complex remote sensing objects. The novelties in our proposed model are listed as following:

- *Seed node:* The target object or area is represented as the center node in the conditional random field, since the model is motivated by the context of the target class.
- *Meta-segments:* In contrast to the *segment-based CRF* approach in many studies, the proposed model is based on meta-segments around the center node. Meta-segments are obtained by merging segments with the same label after the initial classification. This approach has two advantages over the classical CRF representation [38]:
  - *Model complexity:* Model complexity can be kept under control, since the number of nodes does not change by the number of segments, but by the number of classes. Graphical model node size can be maximum n+1, as n is the number of LULC classes and an additional node is for the target class. This is quite advantageous, since limited number nodes means that *exact inference* is possible.
  - *Feature representativeness:* Features extracted from meta-segments are expected to have superior representativity, especially for textural areas. One may claim that classification errors from the initial step may flatten the probability distribution of classes. However, we utilize a strong support vector machine for the initial classification and expect these cases to be minimal. Even with such wrong-labeled segments to be mixed with correctly-labeled segments, the deviation from the original distribution of the class that meta-segment represent is likely to have smaller effect on the posterior probability of the target class rather than having these segment to contribute individually in the graphical model. These segments are probably outliers, and direct effect of them on the posterior probability of the target class is not favorable.

  Although posterior probabilities of LULC segments cannot be individually updated in meta-segments approach, the aim of this study is to correctly identify contextual complex target area rather then its surroundings, thus this situation is disregarded.
- *Spatial interactions:* As the nature of CRFs, interaction potentials can be obtained after employing quite complex functions. Although, in earlier studies, interaction potentials

are simply acquired by concatenating or taking difference of adjacent nodes, in this study we have followed a different approach and defined the spatial contextual interactions as co-occurrence statistics of classes in three different scaled neighborhoods. Table 3.4 summarizes the utilized features.

Table 3.4: Class co-occurrence ratios as spatial context features applied in three different scales of neighborhood.

SCALE OF THE NEIGHBORHOOD	DESCRIPTION	FORMALIZATION
Overlapping	Percentage of each class (their corresponding meta-segments) overlapping with candidate region	$O_{ij} = \begin{cases} \frac{\lvert C_i \cap C_j \rvert}{\lvert C_i \cap C_{\forall j} \rvert} & C_i \cap C_j \neq \varnothing \\ 0 & C_i \cap C_j = \varnothing \end{cases}$
Adjacent	Percentage of each class (their corresponding meta-segments) adjacent with candidate region	$A_{ij} = \begin{cases} \frac{\lvert C_i \cap C_j \rvert}{\lvert C_i \cap C_{\forall j} \rvert} & C_j \in \varepsilon_{C_i}, \varepsilon = 1 pixel \\ 0 & C_j \notin \varepsilon_{C_i}, \varepsilon = 1 pixel \end{cases}$
Nearby	Percentage of each class (their corresponding meta-segments) in the contextually meaningful neighborhood of candidate region	$N_{ij} = \begin{cases} \frac{\lvert C_i \cap C_j \rvert}{\lvert C_i \cap C_{\forall j} \rvert} & C_j \in \varepsilon_{C_i}, \varepsilon = \gamma pixel \\ 0 & C_j \notin \varepsilon_{C_i}, \varepsilon = \gamma pixel \end{cases}$

- *Star-structure:* Star structure is one of the major strengths of the proposed model. As the aim is to correctly classify the candidate region which is positioned at the center of the model, other nodes are expected to interact over the central node. This way, the semantic or probabilistic context of the central node strengthens.

3.11 Chapter Summary

In this chapter, first of all context notion is discussed. Then the details of the proposed model, which employs context to the remote sensing object classification task, are given. The proposed model is a star-structured conditional random field, constructed over a seed node which represent the target class. The target class is expected to form its own context around its surroundings. The surroundings of the target area are initially classified to be merged and form meta-segments. Thus, the spatial contextual relations are formulated over the co-occurrence statistics of these meta-segments in the pairwise potentials of the CRF model.

CHAPTER 4

EXPERIMENTS ON REMOTE SENSING IMAGES

In this chapter, the importance of context during classification is analyzed by a set of experiments. First, classification with low-level features is conducted while disregarding any contextual information. In the second set of experiments, a second stage is proposed which includes conditional random fields over the labels obtained in the first experiment. This second stage is analyzed further in terms of model selection. Fully-connected model and star model are compared. Effects of parameter sharing is also examined in star models.

4.1 Classification of Target Regions in Remote Sensing Images

The aim of this study is to identify a target region in a remote sensing image by characterizing its context with co-occurrence statistics of LULC classes in its surroundings. In this study, we select airfield regions as our target regions and we use six auxiliary LULC classes, namely water, forest, agricultural land (greenland), urban, soil, concrete. These LULC classes are chosen empirically, after careful observations of most likely occurrences of classes in the neighborhood of the airfield candidates.

4.2 Dataset

In this thesis, four multispectral remote sensing images from GEOEYE satellite are used to test the validity of the proposed method. Resolution of each image is approximately 2 meter per pixel. In 4.1, the statistics of the dataset is summarized. Image size in terms of pixels are provided along with number of segments utilized in initial classification phase and number of LULC classes in that phase are given. Furthermore, number of candidate regions in each image and number of candidate regions which reside on target region are provided. In the dataset of four images, there exist 189 candidate regions in total and 77 of these candidate regions belong to the airfield areas. Figure A.1, A.2, A.3, and A.4 exhibits the red-green-blue band combination of each image respectively.

Table 4.1: Statistics about the dataset.

Image Name	Image Size	Number of Labeled Segments	Number of LULC Classes	Number of Candidates	Number of Candidates on Target
I_1	2778x2456	5587	6	23	9
I_2	3780x3808	3459	6	53	35
I_3	3967x3667	2448	6	24	14
I_4	3836x4008	12079	6	89	19

4.3 Finding the Context by Sparse Autoencoder

Identification of the context information is the most crucial part of the context dependent classification schemes. In this study, the contextual features of the target objects are identified by sparse autoencoders by the help of libORF library [21]. Sparse autoencoders are known as powerful tools for feature extraction from the image databases. After collecting square patch samples over airfield areas, patches are first normalized for contrast across all dataset. Contrast normalization is accomplished by mean subtraction from each band separately. Contrast normalization is followed by whitening where we employed zero-phase whitening with regularization in order to slightly smooth (or low-pass filter) the input patches. After applying whitening operation over each sample, adjacent pixels become less correlated and we manage to remove aliasing artifacts in the image which can improve the quality of learned features [16]. Whitening is a crucial preprocessing operation for removing redundancy in the input data.

During training a sparse autoencoder, determining patch size and number of hidden units that controls the sparsity, requires careful analysis of the dataset. While collecting sample patches, including the surrounding pixels that does not belong to the target class may confuse the autoencoder. However, to some degree including surrounding pixels may be useful and their correlation with the target class may be exposed. For analyzing the effect of patch window size, we have trained autoencoders with three empirical patch size values, 16, 20, and 28. Visualization of the hidden units shows what is learned in each unit. In each figure below, small square patches has the same size with the size of collected patches, and total number of patches in the visualization gives the number of hidden nodes in the model. Effect of patch windows size can be observed from figure 4.1, 4.2, and 4.3. As the patch window size increases, the learned features get smoother. This fact is expected in sparse autoencoders because, number of hidden units effects the sparsity of the learned model. All the hidden units share the same sparsity parameter and only the most discriminating hidden unit fires while suppressing other units to zero. As patch size increases, number of input units also increases and this results in more compressed hidden unit actions, in other words smoother features. In figure 4.4, learned characteristics has a wider range than in figure 4.3. For instance, crossing of taxiroutes, which is a part of airfield, can be observed in some hidden

units in figure 4.4, even though they cannot be observed as number of hidden nodes is 100. In addition to that hidden units representing departure end of an airfield is observed more frequently, almost at the same frequency with parallel lines, when the number of hidden units is 400.

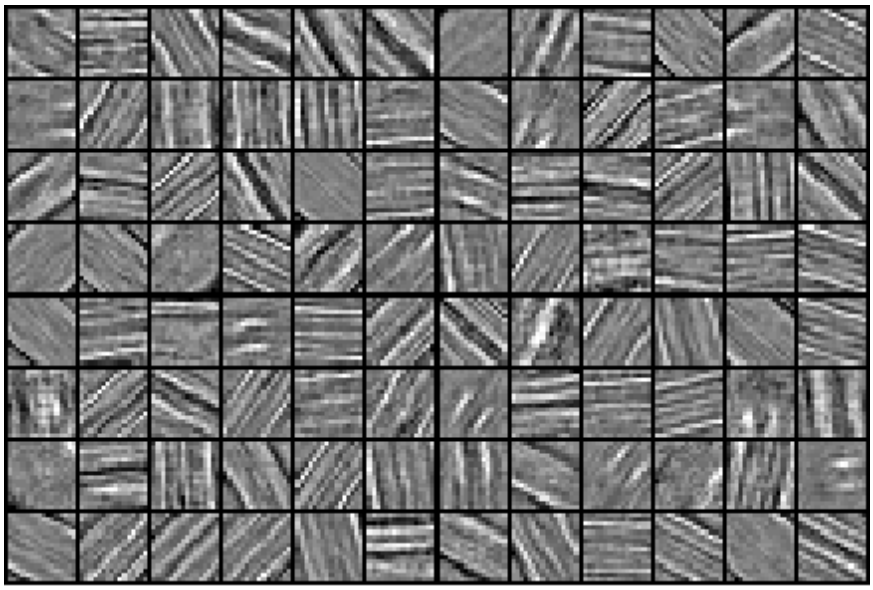

Figure 4.1: Patch dimension is 16 and hidden node size is 96.

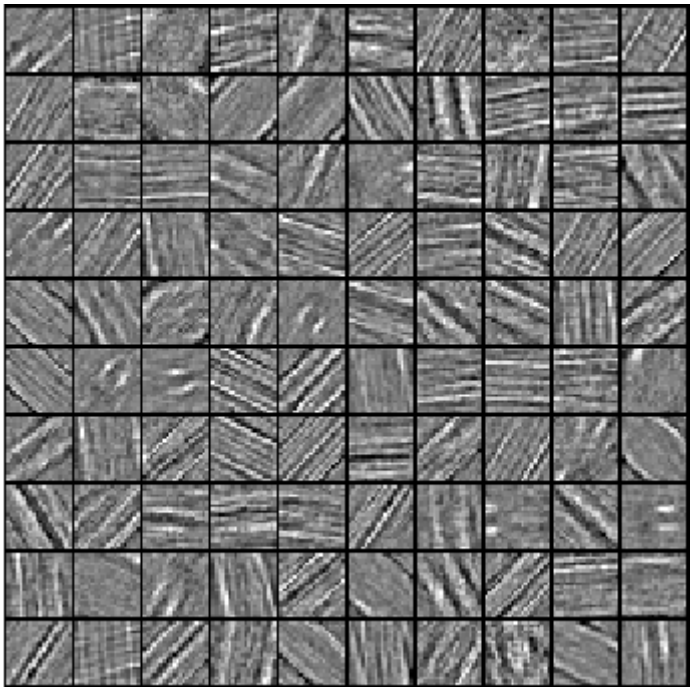

Figure 4.2: Patch dimension is 20 and hidden node size is 100.

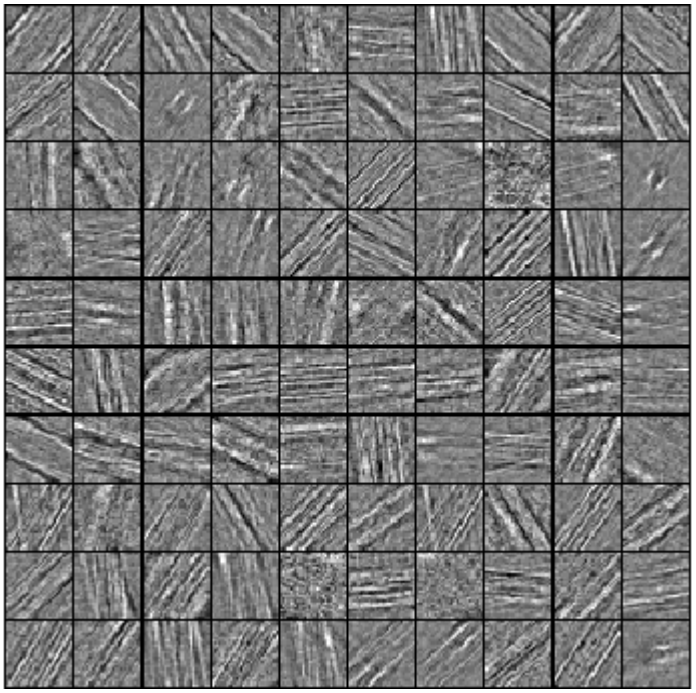

Figure 4.3: Patch dimension is 28 and hidden node size is 100.

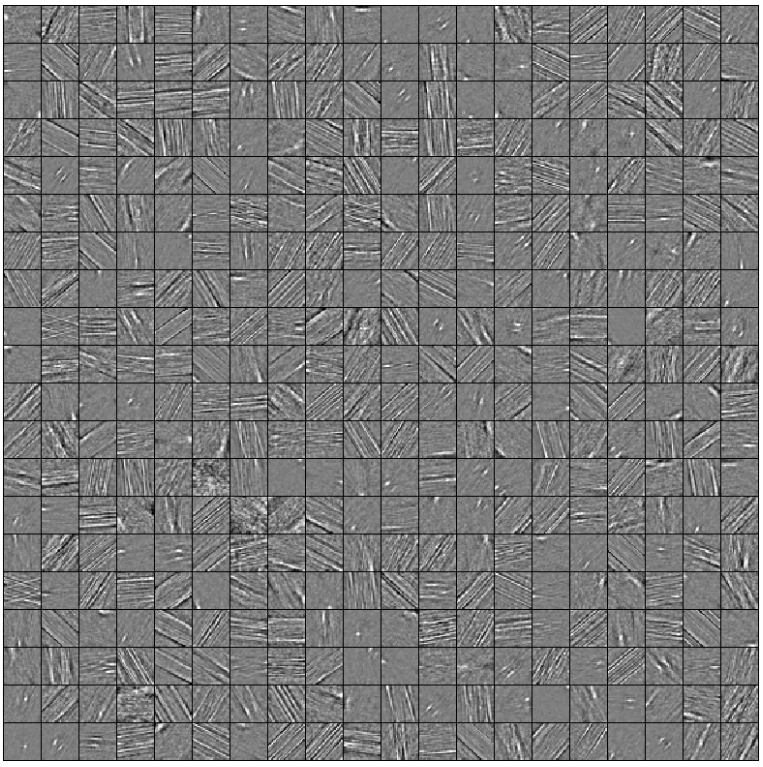

Figure 4.4: Patch dimension is 28 and hidden node size is 400.

In order to identify the contextual cue of airfields by using sparse autoencoders, we can either choose the most repetitive feature in the filter bank learned, or we can simply convolve the test image by using the learned features. While the former approach is taken in this thesis, the latter is also applicable and straightforward. Test image is convolved by using each of learned features separately and then all of the response images are added up together (superposed/superimposed). Superposing results in a heat map like combined-response image and it can easily be thresholded for further use. In the figure 4.5, original image on the left is convolved with the learned features illustrated in figure 4.1 and response image is empirically thresholded. As can be seen from the resulting mask in figure 4.5 right, highest response is observed continuously on the runway regions as parallel lines.

Figure 4.5: Original image (left) and binary mask of *long parallel lines* in the original image (right). Binary image is obtained by applying a threshold to the superposed responses of all convolution filters of sparse autoencoder. This demonstrates the choice of long parallel lines as the most repetitive and representative feature is statistically consistent.

Therefore, long parallel lines are used as context features of airfields in this study. In the following subsections the suggested contextual model is constructed and tested by a CRF architecture, built around parallel line feature.

4.4 Extraction of Candidate Regions

Region of interest in this study is the neighborhood of the target area which is the surroundings of an airfield in this case. The dataset is constructed by extracting long parallel lines with a line segment detector algorithm [76]. We have chosen lines which consist of more than 125

pixels, empirically. Thus the candidate regions for airfield class are the bounded areas of extracted long parallel lines. Figure 4.6 shows the superposed candidate parallel line bounded regions, which are called as PLBR after this point, as in [22]. In figure 4.6, the rightmost image shows the PLBRs overlaid on the image. In this image, each PLBR is marked by the color of its corresponding class. Red PLBRs indicate that they are on the airfield area. Class colors for the surrounding classes are as follows: Orange for soil, purple for urban, cyan for concrete or asphalt, dark green for forest, green for agricultural land and blue for water. As observed from figure A.5, A.6, A.7, and A.8, detected candidate PLBRs may be part of an urban area, such as part of a highway, or long straight soil roads, or part of soil lands, or border lines between agricultural lands, or part of water canals etc.

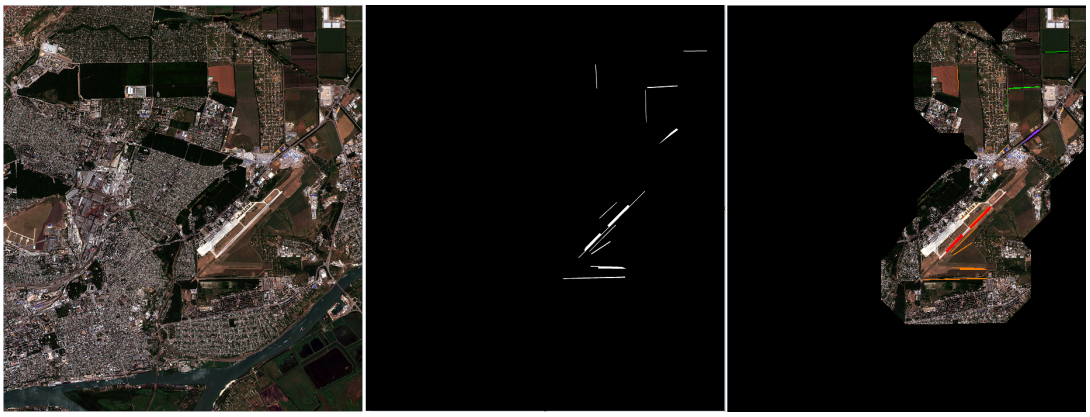

Figure 4.6: RGB band combination of original image (left), parallel line bounded candidate regions (middle), PLBRs overlaid on the image (right). In this image, each PLBR is marked by the color of its corresponding class. Class colors are red for airfield class, orange for soil, purple for urban, cyan for concrete or asphalt, dark green for forest, green for agricultural land and blue for water.

Recall that, in this study the spatial contextual relations are investigated. Therefore, the region of interest in the image is defined as the neighborhood of the candidate regions. Pixels up to 600 meter away from the candidate region are considered to be inside the region of interest. This neighborhood scalar is provided by the domain expert. By applying a dilation operation on the candidate PLBRs, region of interest is obtained as in figure A.5, A.6, A.7, and A.8. In the dataset, there exist 189 candidate PLBRs in total and 77 of candidate PLBRs belong to the airfield areas. Together with the segments in the region of interest, each candidate PLBR is labelled as one of the following classes: airfield, water, forest, agricultural land, soil, concrete, urban.

The following section presents the results of segmentation algorithms which are experimented with during this study.

4.5 Segmentation

Segmentation is the first crucial step in classification tasks. Selection of a segmentation algorithm needs careful examination, since the borders of a region may be essential cues for the classification and mixture of other classes in a segment may corrupt the statistics of the features of that segment. Errors sourced from segmentation step increase exponentially in the later steps, in the aspect of feature extraction, and later during assigning a class label to the segments, misclassification may become inevitable.

In this study, we examine the popular segmentation methods, named SLIC (simple linear iterative clustering), watershed and mean shift segmentation. An empirical analysis reveals that SLIC method, which is an efficient variant of k-means clustering algorithm [2], mostly yields an over-segmented partition of an image. Although the method is tested by several parameters to adjust and control the region size, the resulting mask consists of some undesirable small and large regions. This fact is depicted in figure 4.7, where the segmentation output is displayed for region size parameter of 10, 50, 100, 200, 250, 300, 400, 500.

Similar empirical analysis is conveyed for the watershed segmentation method [7]. It is observed that watershed is particularly an ill-conditioned method in heterogeneous regions, such as urban area. On the other hand, the homogeneous objects are clustered into rather regions with unsmooth borders, which is not desirable for our problem domain. The output of watershed segmentation is depicted in figure 4.8 for various threshold parameters. It is observed that the output masks do not preserve the output borders.

Finally, mean shift segmentation is empirically analyzed by changing the parameter of region size as depicted in Figure 4.9. Other parameters of mean shift segmentation, namely spatial (h_s) and color range (h_r) parameters, which determine the bandwidth of the kernel during mode seeking in the feature space, are determined as small values empirically.

When the region size parameter is 5000, under-segmentation occurs. Although, the result obtained with the region size value 500 is slightly over-segmented, it seems to keep the details such as thin roads better than other results. Furthermore, land use-land cover classes are kept isolated. For this reason, in this study, we proceed with region size value 500. It is concluded that the most appropriate segmentation method for the detection problem of this study is the mean shift segmentation method.

4.6 Initial Classifier Selection

The first fundamental step in the proposed model is assigning the candidate class labels to land use/land cover classes in an initial classification. Selection of the initial classifier requires significant care, since the meta-segments, which the proposed contextual conditional random fields is established over, are determined with the labeling of the initial classifier. In other words, a strong classifier is favorable for proceeding to the next stage with minimal error.

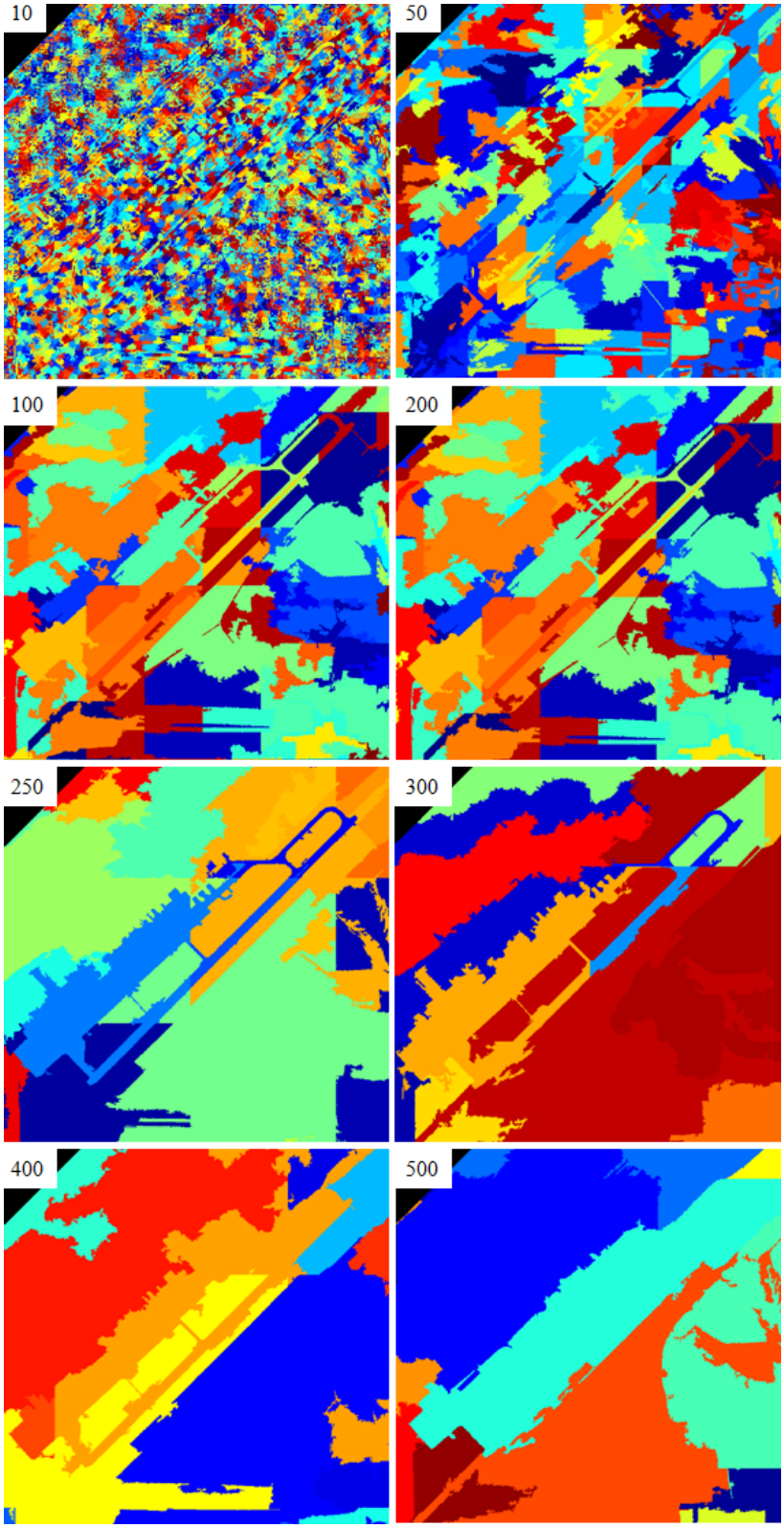

Figure 4.7: SLIC segmentations of I_1 while region size spanning the values 10, 50, 100, 200, 250, 300, 400, 500 and the regularizer value is 10.

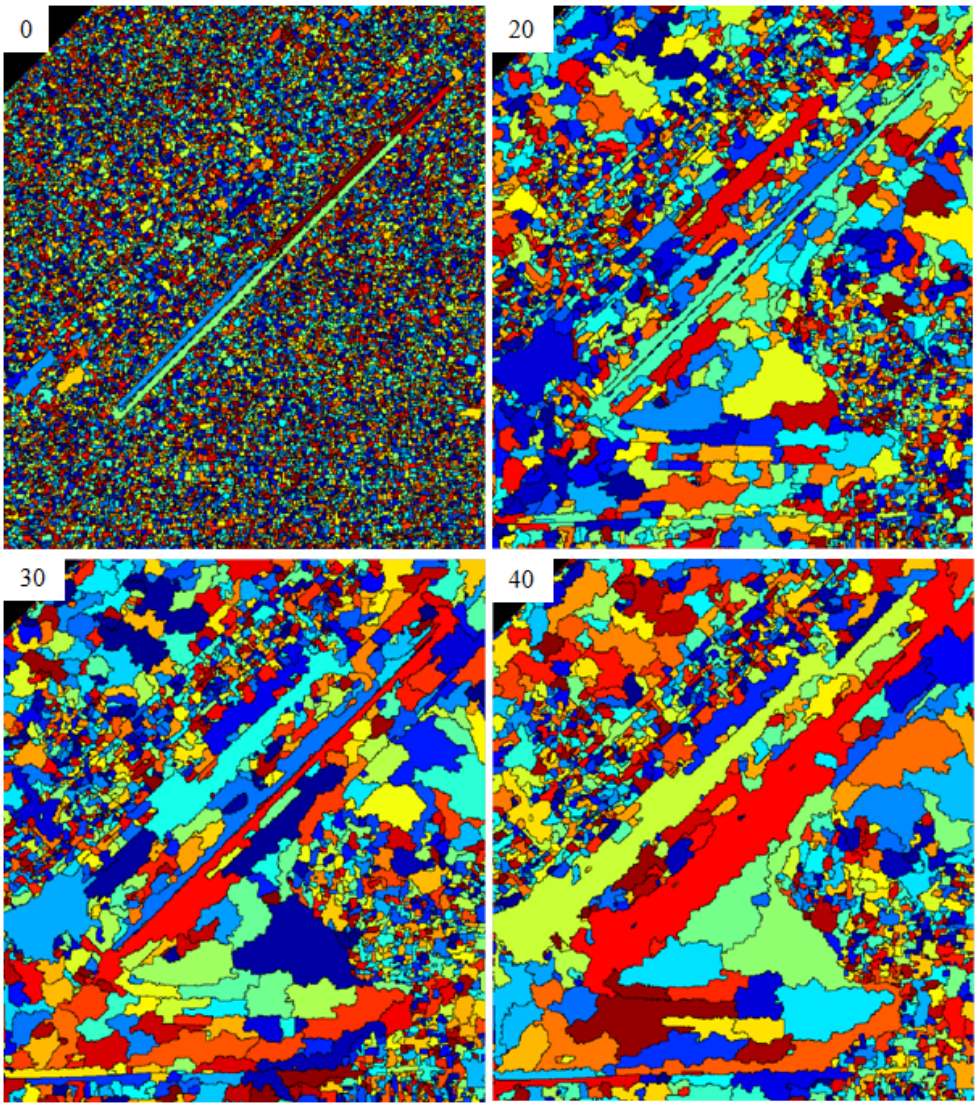

Figure 4.8: Watershed segmentations of image I_1 while shallow minimas are suppressed according to threshold values 0 (without suppression), 20, 30, 40.

For this purpose, we have decided to examine two popular classifiers, namely support vector machine (SVM) considering its ability to handle sparse data, and k-nearest neighbor method due to its ability to handle large sample size.

4.6.1 Feature Extraction

As described in section 2.2.1, features are extracted from the segments. In a more formal way, extracted features can be denoted as following,

$$[V_i, W_i, E_i, G_i, S_i]$$

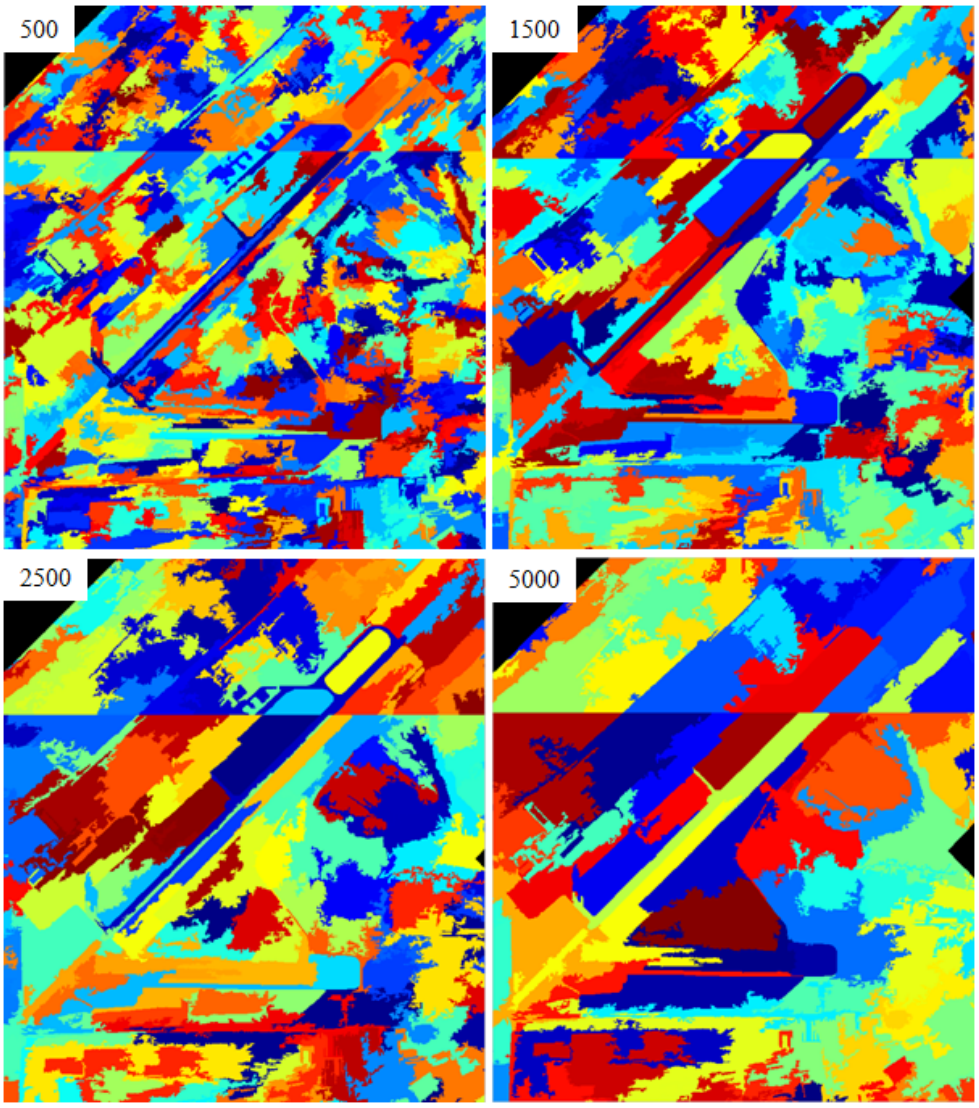

Figure 4.9: Mean shift segmentations of image I_1 while h_s and h_r are 2, and region size spanning the values 500, 1500, 2500, 5000.

This notation corresponds to concatenation of vegetation features (V_i), water features (W_i), elevation features (E_i), textural or Gabor filter features (G_i) and spectral features (S_i) extracted from segment i.

The detailed notation of these features are presented below.

$$V_i = [\mu_{NDVI^{c_i}}, \sigma_{NDVI^{c_i}}, \alpha_{NDVI^{c_i}}]$$

$$W_i = [\mu_{NDWI^{c_i}}, \sigma_{NDWI^{c_i}}, \alpha_{NDWI^{c_i}}]$$

$$G_i = [\mu_{Gabor^{c_i}}, \sigma_{Gabor^{c_i}}, \alpha_{Gabor^{c_i}}]$$

$$E_i = [\sigma_{DTED^{c_i}}, KL(hist_{DTED^{c_i}} \parallel U)]$$

$$S_i = [\mu_{Red^{C_i}}, \sigma_{Red^{C_i}}, \mu_{Green^{C_i}} \sigma_{Green^{C_i}}, \mu_{Blue^{C_i}}, \sigma_{Blue^{C_i}}, \mu_{NIR^{C_i}}, \sigma_{NIR^{C_i}}]$$

In these notations, μ stands for the mean value, σ stands for standard deviation, and α stands for the ratio of pixels over a dynamic threshold in segment i (C_i) in the corresponding map, which is given as subscript. For the elevation feature, we also consider Kullback-Leibler divergence of histogram of DTED values in segment i from the uniform distribution.

4.6.2 Support Vector Machine

For training a support vector machine for initial classification of land use-land cover classes, features extracted from segments in two images, I_1 and I_4 in the dataset, are fed to libSVM library [13] together with their class labels. The other two images, I_2 and I_3, which are partially labeled are used as test images. Number of labeled segments in all images are given in Table 4.1. LibSVM library uses one-vs-one strategy for handling multi class classification. Radial basis function (RBF) kernel is utilized during SVM classification. Parameters of RBF kernel is searched through parameter space via cross validation.

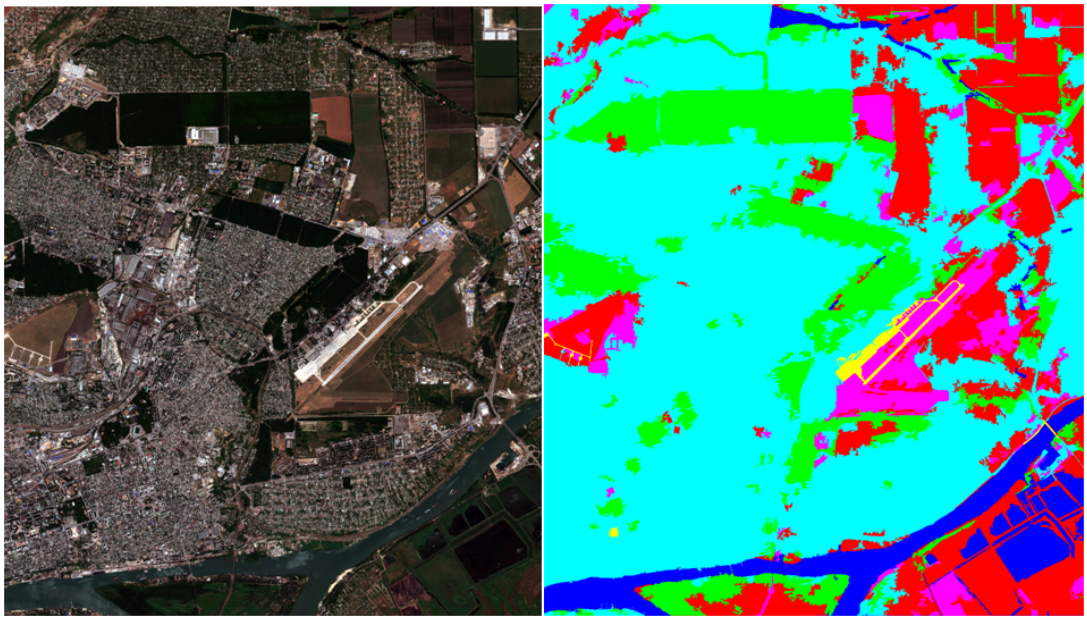

Figure 4.10: R-G-B combination of I_1 (left) and its SVM result (right).


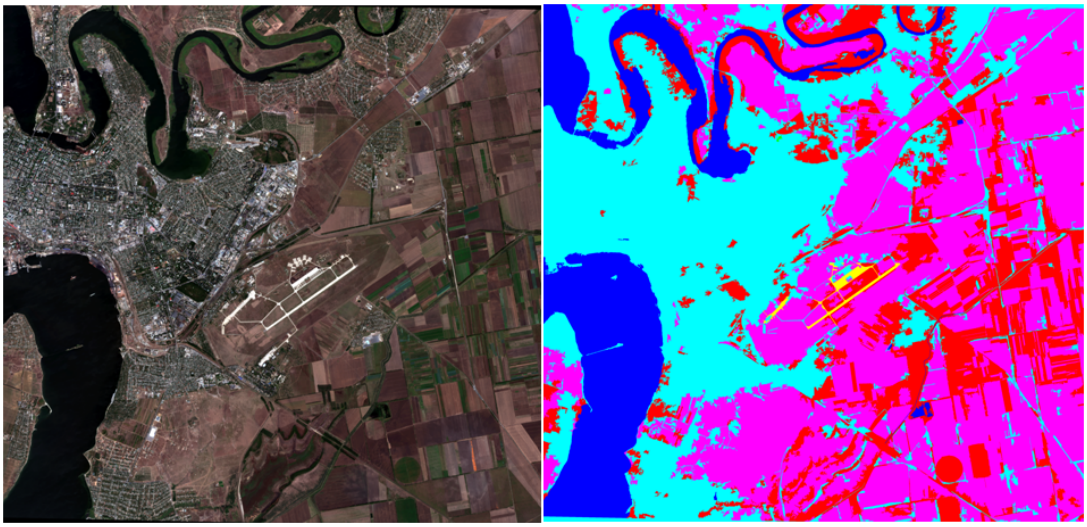

Figure 4.11: R-G-B combination of I_4 (left) and its SVM result (right).

Table 4.2: Confusion matrix for the initial SVM classification of I_1.

		Classifier Results							
		Water	**Forest**	**Greenland**	**Soil**	**Urban**	**Concrete**	**Truth Overall**	**User Accuracy (Recall)**
Truth Data	**Water**	5.08%	0.00%	0.09%	0.00%	0.00%	0.00%	5.17%	0.9827
	Forest	0.05%	13.44%	0.70%	0.05%	2.22%	0.00%	16.47%	0.8163
	Greenland	0.04%	0.61%	15.97%	0.55%	1.36%	0.00%	18.53%	0.8618
	Soil	0.00%	0.04%	1.29%	5.78%	0.43%	0.00%	7.54%	0.7672
	Urban	0.00%	0.54%	0.55%	0.11%	50.31%	0.00%	51.51%	0.9767
	Concrete	0.00%	0.00%	0.02%	0.00%	0.16%	0.61%	0.79%	0.7727
	Classification Overall	5.17%	14.62%	18.61%	6.50%	54.48%	0.61%	100.00%	
	Producer Accuracy (Precision)	0.9827	0.9192	0.8577	0.8898	0.9235	1.0000		

Confusion matrix of the SVM result of the I_1 is given in Table 4.2. This table is constructed according to the labels assigned to the segments. Each cell shows the percentage of total segments which fall into corresponding description, i.e. 5.08 % value in water-water cell means that 5.08 % of total segments are correctly classified as water. For this image, average precision of all classes is reported as 0.8629 and average recall is reported as 0.9288. From 5587 segments existing in this image, SVM classifier correctly classifies 5095 segments in total. Average accuracy is reported as 0.9119 by taking the ratio of these values.

Table 4.3: Confusion matrix for the initial SVM classification of I_4.

		Classifier Results							
		Water	**Forest**	**Greenland**	**Soil**	**Urban**	**Concrete**	**Truth Overall**	**User Accuracy (Recall)**
Truth Data	**Water**	6.47%	0.00%	0.12%	0.02%	0.21%	0.00%	6.82%	0.9478
	Forest	0.00%	0.00%	1.36%	0.11%	0.37%	0.00%	1.84%	0.0000
	Greenland	0.03%	0.00%	11.58%	4.30%	1.61%	0.00%	17.53%	0.6608
	Soil	0.01%	0.00%	3.27%	36.32%	5.40%	0.00%	45.00%	0.8072
	Urban	0.03%	0.01%	1.57%	0.60%	26.46%	0.00%	28.67%	0.9229
	Concrete	0.00%	0.00%	0.00%	0.00%	0.11%	0.04%	0.15%	0.2778
	Classification Overall	6.54%	0.01%	17.91%	41.35%	34.15%	0.04%	100.00%	
	Producer Accuracy (Precision)	0.9886	0.0000	0.6468	0.8783	0.7748	1.0000		

Confusion matrix of the SVM result of the I_4 is given in Table 4.3. This table is constructed according to the labels assigned to the segments. For this image, average precision of all classes is reported as 0.6028 and average recall is reported as 0.7147. From 12079 segments existing in this image, SVM classifier correctly classifies 9768 segments in total. Average accuracy is reported as 0.8086 by taking the ratio of these values.

Visual results of this step for the other two images can be observed in Figure 4.12 and 4.13 respectively.

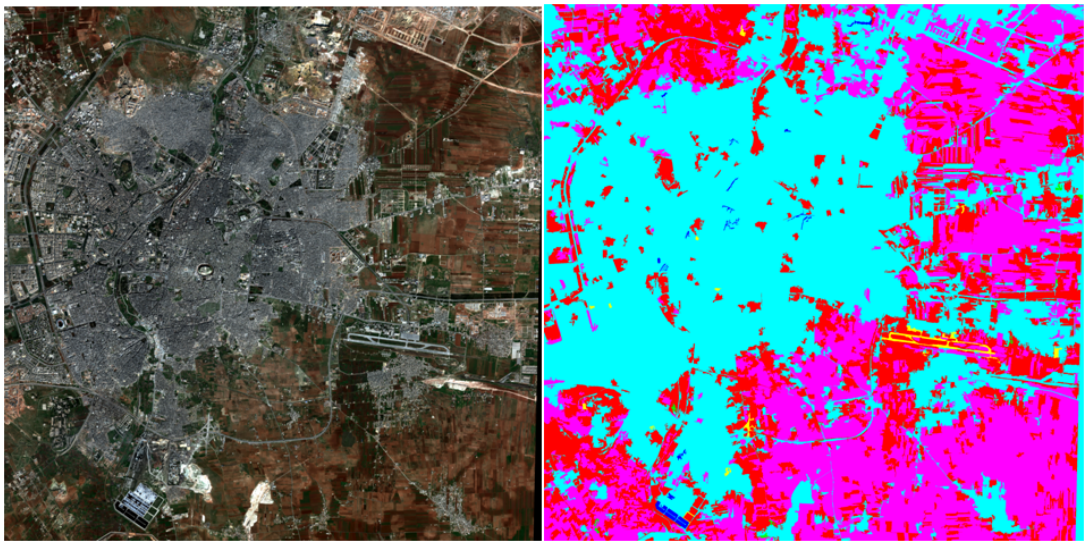

Figure 4.12: R-G-B combination of I_2 (left) and its SVM result (right).
Water Urban Forest Greenland Soil Concrete

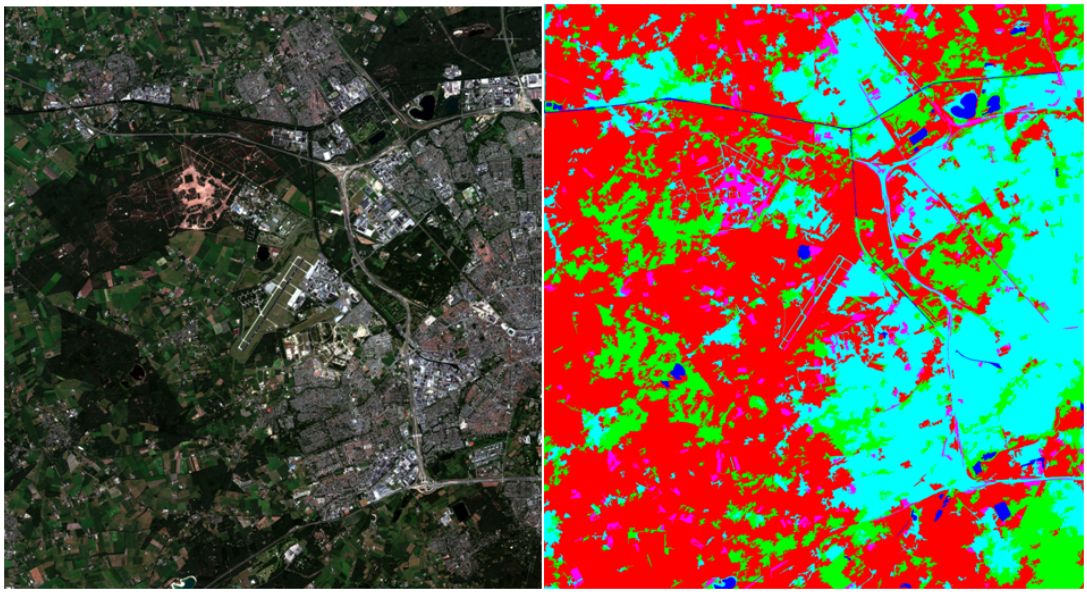

Figure 4.13: R-G-B combination of I_3 (left) and its SVM result (right).
Water Urban Forest Greenland Soil Concrete

Table 4.4: Confusion matrix for the initial SVM classification of I_2.

		Classifier Results							
		Water	**Forest**	**Greenland**	**Soil**	**Urban**	**Concrete**	**Truth Overall**	**User Accuracy (Recall)**
Truth Data	**Water**	0.14%	0.00%	0.00%	0.00%	0.61%	0.00%	0.75%	0.1923
	Forest	0.00%	0.00%	0.75%	0.09%	0.06%	0.00%	0.90%	0.0000
	Greenland	0.00%	0.09%	20.15%	23.19%	6.88%	0.00%	50.30%	0.4006
	Soil	0.00%	0.00%	1.13%	11.45%	1.33%	0.00%	13.91%	0.8233
	Urban	0.00%	0.03%	3.50%	3.79%	25.33%	0.09%	32.73%	0.7739
	Concrete	0.00%	0.00%	0.40%	0.03%	0.40%	0.58%	1.42%	0.4082
	Classi-fication Overall	0.14%	0.12%	25.93%	38.54%	34.61%	0.66%	100.00%	
	Producer Accuracy (Precision)	1.0000	0.0000	0.7770	0.2971	0.7318	0.8696		

Confusion matrix of the SVM result of the I_2 is given in Table 4.4. This table is constructed according to the labels assigned to the segments present in the region of interest of candidate regions. For this image, average precision of all classes is reported as 0.4330 and average recall is reported as 0.6126. From 3459 segments existing in the region of interest, SVM classifier correctly classifies 1994 segments in total. Average accuracy is reported as 0.5764 by taking the ratio of these values.

Table 4.5: Confusion matrix for the initial SVM classification of I_3.

		Classifier Results						
		Water	**Forest**	**Greenland**	**Soil**	**Urban**	**Truth Overall**	**User Accuracy (Recall)**
Truth Data	**Water**	1.51%	0.00%	0.00%	0.00%	0.16%	1.67%	0.9024
	Forest	0.04%	9.07%	16.58%	0.20%	0.45%	26.35%	0.3442
	Greenland	0.08%	2.57%	38.32%	0.65%	3.72%	45.34%	0.8450
	Soil	0.00%	0.12%	2.90%	1.35%	1.27%	5.64%	0.2391
	Urban	0.00%	2.17%	2.04%	1.14%	14.09%	19.44%	0.7248
	Concrete	0.00%	0.04%	0.25%	0.25%	1.02%	1.55%	0.0000
	Classification Overall	1.63%	13.97%	60.09%	3.59%	20.71%	100.00%	
	Producer Accuracy (Precision)	0.9250	0.6491	0.6377	0.3750	0.6805		

Confusion matrix of the SVM result of the I_3 is given in Table 4.5. This table is constructed according to the labels assigned to the segments present in the region of interest of candidate regions. For this image, average precision of all classes is reported as 0.5093 and average recall is reported as 0.6535. From 2448 segments existing in the region of interest, SVM classifier correctly classifies 1575 segments in total. Average accuracy is reported as 0.6433 by taking the ratio of these values. Unfortunately, SVM cannot classify any segment of concrete class correctly in this image. Therefore, concrete column in 4.5 is omitted.

4.6.3 k-Nearest Neighbor Classifier

Nearest neighbor approaches is commonly used in the literature for classification tasks. Especially k-nearest neighbor (k-NN) approaches, which considers labels of k nearest samples in the feature space in order to assign the label of a sample, is generally utilized in classification problems. The strong points of k-NN approaches are simplicity and applicability to datasets with large sample size.

In this study, k-NN approach is also examined for the initial classification. Number of neighbors to consider, parameter k, is selected to be 7 empirically, which is one more than the number of LULC classes. Visual results can be observed in Figure 4.14, 4.16 4.17 and 4.15.

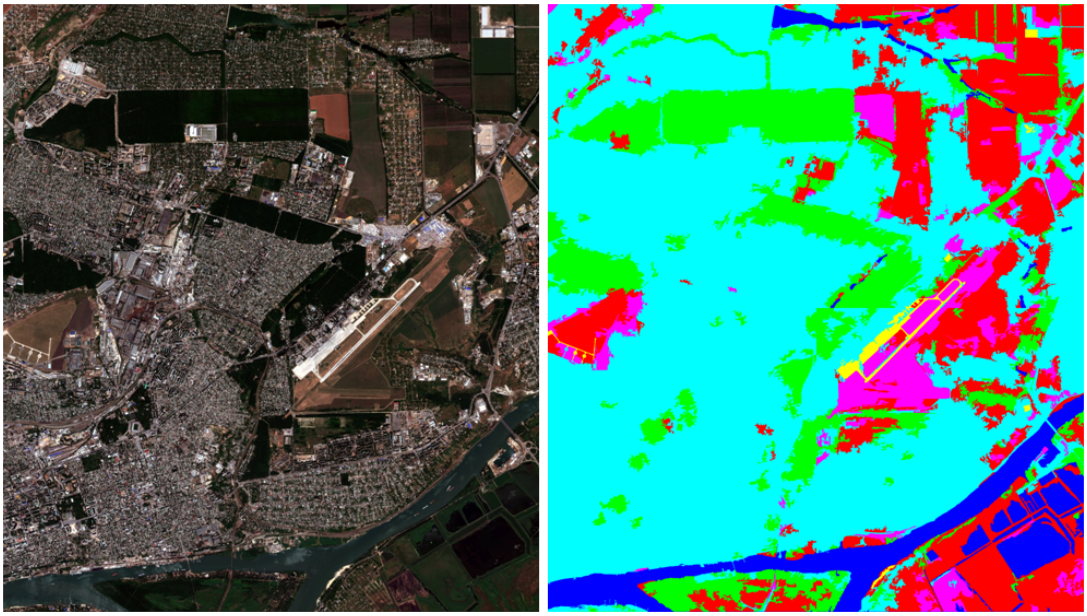

Figure 4.14: R-G-B combination of I_1 (left) and its k-NN result (right).

Table 4.6: Confusion matrix for the initial k-NN classification of I_1.

		Classifier Results							
		Water	**Forest**	**Greenland**	**Soil**	**Urban**	**Concrete**	**Truth Overall**	**User Accuracy (Recall)**
Truth Data	**Water**	4.96%	0.05%	0.07%	0.02%	0.07%	0.00%	5.17%	0.9585
	Forest	0.09%	13.23%	0.52%	0.07%	2.56%	0.00%	16.47%	0.8033
	Greenland	0.09%	1.72%	14.52%	0.57%	1.63%	0.00%	18.53%	0.7836
	Soil	0.00%	0.11%	1.36%	5.10%	0.97%	0.00%	7.54%	0.6770
	Urban	0.00%	0.43%	0.50%	0.13%	50.37%	0.09%	51.51%	0.9778
	Concrete	0.00%	0.00%	0.00%	0.02%	0.38%	0.39%	0.79%	0.5000
	Classification Overall	5.14%	15.54%	16.97%	5.91%	55.97%	0.48%	100.00%	
	Producer Accuracy (Precision)	0.9652	0.8514	0.8555	0.8636	0.8999	0.8148		

Confusion matrix of the k-NN result of the I_1 is given in Table 4.6. This table is constructed according to the labels assigned to the segments. For this image, average precision of all classes is reported as 0.7833 and average recall is reported as 0.8751. From 5587 segments existing in this image, k-NN classifier correctly classifies 4948 segments in total. Average accuracy is reported as 0.8857 by taking the ratio of these values.

Table 4.7: Confusion matrix for the initial k-NN classification of I_4.

		Classifier Results							
		Water	**Forest**	**Greenland**	**Soil**	**Urban**	**Concrete**	**Truth Overall**	**User Accuracy (Recall)**
Truth Data	**Water**	6.53%	0.02%	0.08%	0.07%	0.11%	0.00%	6.82%	0.9575
	Forest	0.00%	0.23%	0.84%	0.21%	0.56%	0.00%	1.84%	0.1261
	Greenland	0.02%	0.63%	8.38%	6.19%	2.30%	0.00%	17.53%	0.4780
	Soil	0.01%	0.12%	1.77%	38.70%	4.37%	0.02%	45.00%	0.8600
	Urban	0.03%	0.12%	0.29%	1.17%	27.05%	0.02%	28.67%	0.9434
	Concrete	0.00%	0.00%	0.00%	0.00%	0.08%	0.07%	0.15%	0.4444
	Classification Overall	6.60%	1.13%	11.36%	46.34%	34.47%	0.11%	100.00%	
	Producer Accuracy (Precision)	0.9900	0.2059	0.7376	0.8351	0.7846	0.6154		

Confusion matrix of the k-NN result of the I_4 is given in Table 4.7. This table is constructed according to the labels assigned to the segments. For this image, average precision of all classes is reported as 0.6349 and average recall is reported as 0.6948. From 12079 segments existing in this image, k-NN classifier correctly classifies 9778 segments in total. Average accuracy is reported as 0.8095 by taking the ratio of these values.

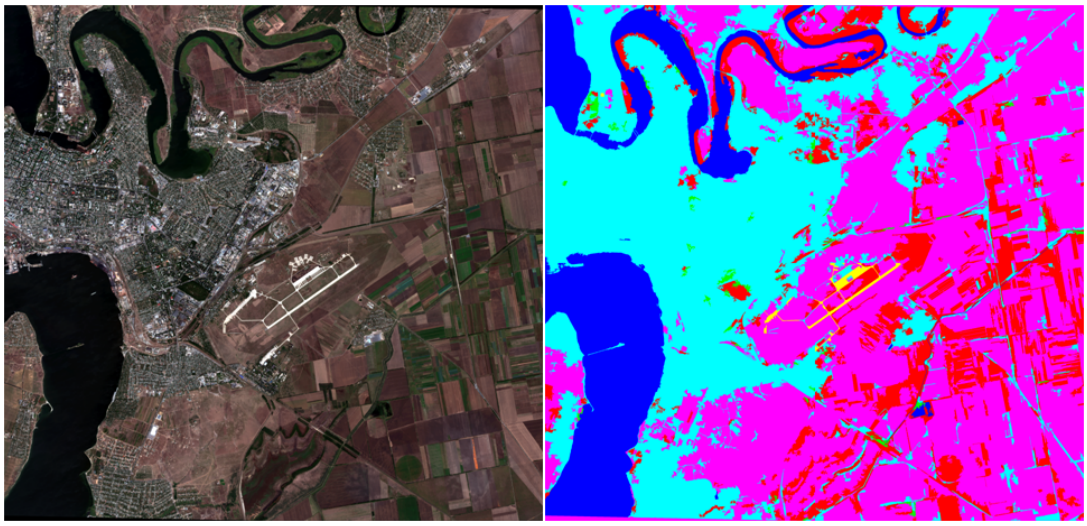

Figure 4.15: R-G-B combination of I_4 (left) and its k-NN result (right).
Water Urban Forest Greenland Soil Concrete

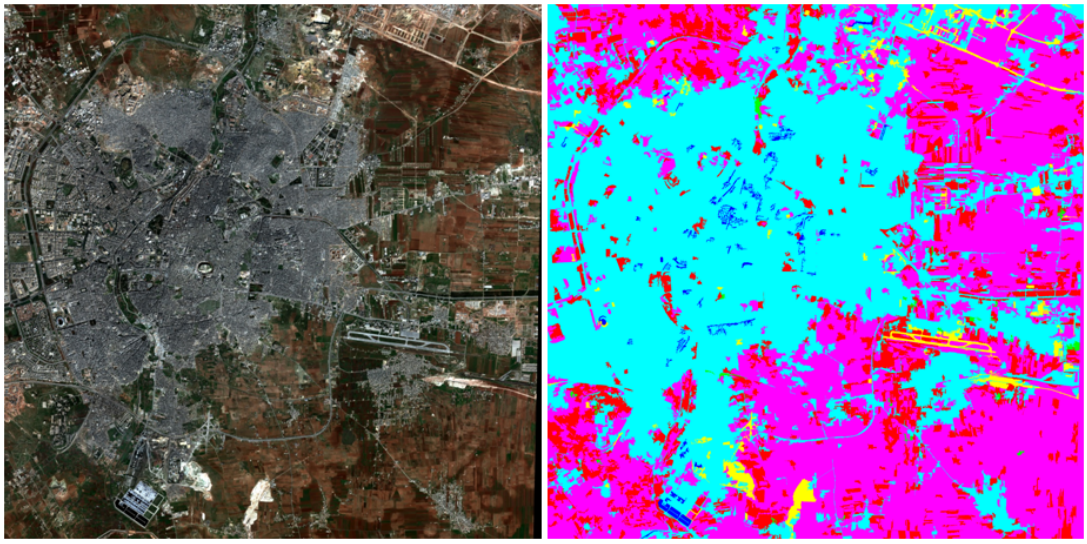

Figure 4.16: R-G-B combination of I_2 (left) and its k-NN result (right).
Water Urban Forest Greenland Soil Concrete

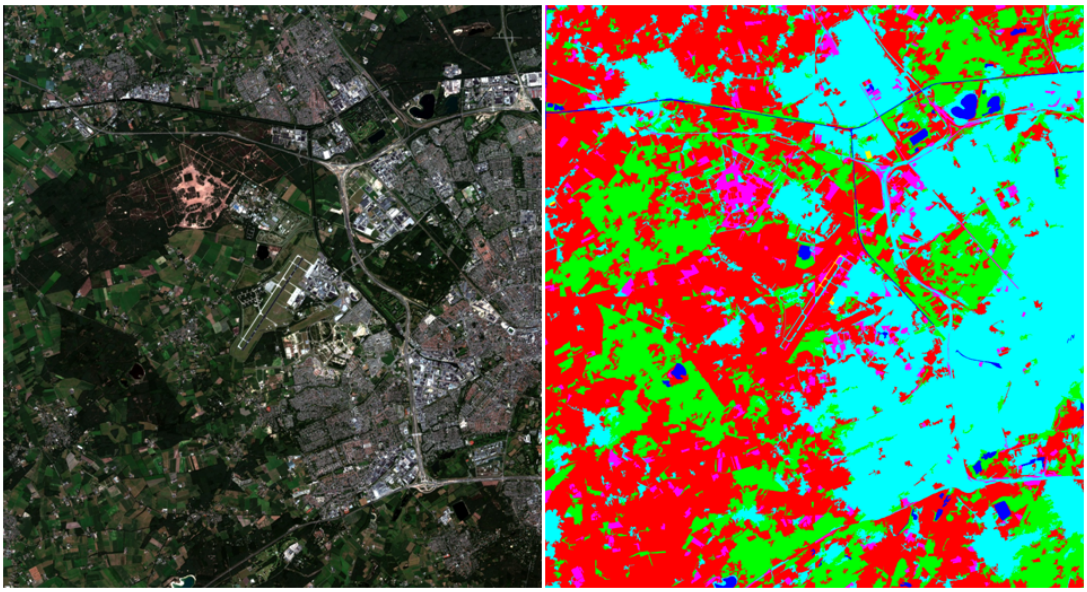

Figure 4.17: R-G-B combination of I_3 (left) and its k-NN result (right).

Table 4.8: Confusion matrix for the initial k-NN classification of I_2.

		Classifier Results							
		Water	**Forest**	**Greenland**	**Soil**	**Urban**	**Concrete**	**Truth Overall**	**User Accuracy (Recall)**
Truth Data	**Water**	0.17%	0.00%	0.00%	0.00%	0.58%	0.00%	0.75%	0.2308
	Forest	0.00%	0.00%	0.20%	0.61%	0.09%	0.00%	0.90%	0.0000
	Greenland	0.03%	0.23%	10.67%	30.10%	9.02%	0.26%	50.30%	0.2121
	Soil	0.00%	0.00%	0.17%	11.94%	1.33%	0.46%	13.91%	0.8586
	Urban	0.00%	0.06%	1.73%	4.51%	25.47%	0.95%	32.73%	0.7783
	Concrete	0.00%	0.00%	0.14%	0.40%	0.38%	0.49%	1.42%	0.3469
	Classification Overall	0.20%	0.29%	12.92%	47.56%	36.86%	2.17%	100.00%	
	Producer Accuracy (Precision)	0.8571	0.0000	0.8255	0.2511	0.6910	0.2267		

Confusion matrix of the k-NN result of the I_2 is given in Table 4.8. This table is constructed according to the labels assigned to the segments present in the region of interest of candidate regions. For this image, average precision of all classes is reported as 0.4044 and average recall is reported as 0.4752. From 3459 segments existing in the region of interest, k-NN classifier correctly classifies 1686 segments in total. Average accuracy is reported as 0.4874 by taking the ratio of these values.

Table 4.9: Confusion matrix for the initial k-NN classification of I_3.

		Classifier Results							
		Water	**Forest**	**Greenland**	**Soil**	**Urban**	**Concrete**	**Truth Overall**	**User Accuracy (Recall)**
Truth Data	**Water**	1.67%	0.00%	0.00%	0.00%	0.00%	0.00%	1.67%	1.0000
	Forest	0.04%	10.50%	13.64%	0.69%	1.47%	0.00%	26.35%	0.3984
	Greenland	0.16%	6.00%	32.48%	2.74%	3.92%	0.04%	45.34%	0.7162
	Soil	0.00%	0.08%	1.43%	2.57%	1.55%	0.00%	5.64%	0.4565
	Urban	0.00%	1.39%	1.18%	1.55%	15.32%	0.00%	19.44%	0.7878
	Concrete	0.00%	0.00%	0.33%	0.33%	0.61%	0.29%	1.55%	0.1842
	Classification Overall	1.88%	17.97%	49.06%	7.88%	22.88%	0.33%	100.00%	
	Producer Accuracy (Precision)	0.8913	0.5841	0.6619	0.3264	0.6696	0.8750		

Confusion matrix of the k-NN result of the I_3 is given in Table 4.9. This table is constructed according to the labels assigned to the segments present in the region of interest of candidate regions. For this image, average precision of all classes is reported as 0.5905 and average recall is reported as 0.6681. From 2448 segments existing in the region of interest, k-NN classifier correctly classifies 1538 segments in total. Average accuracy is reported as 0.6282 by taking the ratio of these values.

Table 4.10: Comparison of average accuracy values of SVM and k-NN for the dataset.

Average Accuracy	**SVM**	**k-NN**
I_1	0.9119	0.8857
I_2	0.5764	0.4874
I_3	0.6433	0.6282
I_4	0.8086	0.8095

Although, visual results seem to be competitive with the SVM results, SVM classifier performs better than k-NN in terms of average accuracy as observed in Table 4.10. Therefore, SVM is selected as the initial classifier for proceeding to the contextual model.

4.7 Classification with Segment-Based Basic Conditional Random Fields

Conditional random fields approaches in computer vision literature generally construct their fields over pixels, windows or segments [38, 6]. For emphasizing the difference and strengths of our proposed approach, we made a comparative study and adapted this conditional random fields approach, which is constructed over segments, to our problem. Let us call this approach basic-CRF. In basic-CRF, cites in the graph are land use-land cover class segments together with an additional target class segment. For obtaining unary potentials, the same features are extracted from segments, instead of meta-segments. Then pairwise potentials are selected to be concatenation of two adjacent pair of nodes in the graph, which is a common preference for basic-CRF approaches [38]. Another common choice for pairwise potentials is taking the difference of two adjacent pair of nodes in the graph, however this experiment is omitted in this study.

Since the aim is this study is to assign class labels to candidate regions of contextually consistent target regions, only the labels assigned to candidate regions catch our attention and performance of LULC classes is disregarded. To be fair to our proposed approach, the same policy is followed for evaluating this experimental setup as well.

As stated in the earlier sections, our dataset consists of 189 candidate regions of airfield class. As training ratio is selected to be 0.2, the experiments are completed in five folds. In each fold, 37 different candidate regions are used for training and the rest of the candidate regions are used during testing phase. The effect of regularization, which is applied during parameter estimation phase, is examined by conducting the experiments with ten different regularizer values. For analyzing the randomness, all the tests are repeated 10 times.

Lambda-fold analysis performances are presented in Table 4.11, 4.12, 4.13 and illustrated with Figure 4.18, 4.19 and 4.20.

Performance values averaged over folds and runs with varying regularizer (lambda) values

Table 4.11: Average precision values of each fold for corresponding lambda values for segment-based basic CRF model.

Lambda	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Average Precision
0	0.7990	0.7595	0.7506	0.6879	0.7454	0.7485
0.001	0.8030	0.7630	0.7639	0.6879	0.7454	0.7526
0.003	0.7998	0.7630	0.7639	0.6930	0.7355	0.7511
0.01	0.7941	0.7612	0.7464	0.6930	0.7481	0.7486
0.03	0.7957	0.7531	0.7659	0.7089	0.7429	0.7533
0.1	0.8344	0.7647	0.7952	0.7391	0.7558	0.7778
0.3	0.8611	0.8258	0.8192	0.7717	0.7924	0.8140
1	0.8589	0.8197	0.8204	0.8044	0.7962	**0.8199**
3	0.8535	0.8190	0.7926	0.7963	0.7802	0.8083
10	0.8542	0.8041	0.7861	0.7845	0.8000	0.8058

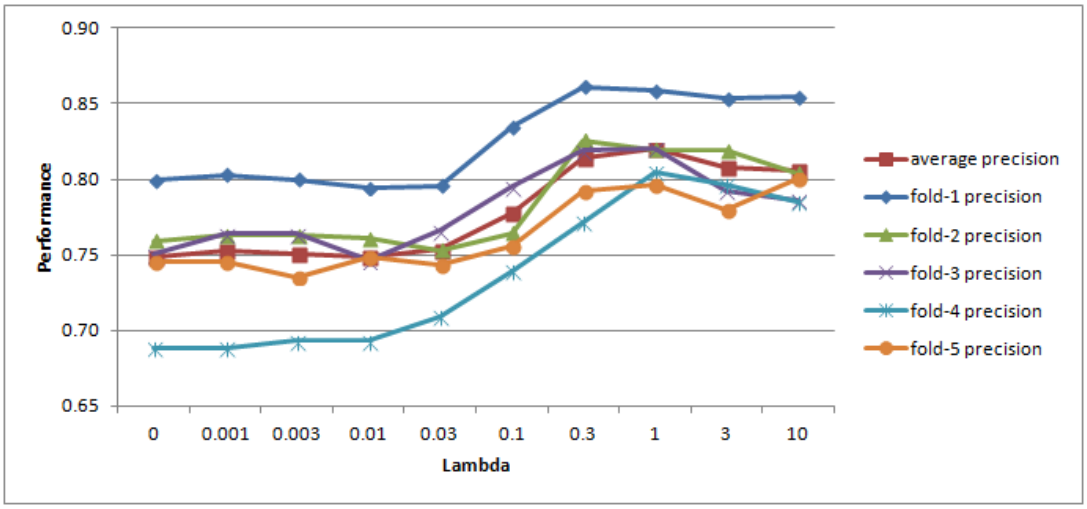

Figure 4.18: Average precision values of each fold for corresponding lambda values for segment-based basic CRF model.

are presented in Table 4.14 and illustrated with Figure 4.21. According to these performance values, best choices for the regularizer value seem to be 0.3 which maximizes the f-score value.

As observed from the tables above, performance results reach only to 0.81 for precision and 0.70 for recall measure with this basic CRF approach.

Table 4.12: Average recall values of each fold for corresponding lambda values for segment-based basic CRF model.

Lambda	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Average Recall
0	0.6748	0.7078	0.7260	0.6762	0.6974	0.6965
0.001	0.6870	0.7078	0.7260	0.6762	0.6974	0.6989
0.003	0.6804	0.7078	0.7260	0.6762	0.6974	0.6976
0.01	0.6754	0.7019	0.7260	0.6762	0.6974	0.6954
0.03	0.6887	0.7078	0.7482	0.6762	0.7027	**0.7047**
0.1	0.6748	0.7004	0.7374	0.6685	0.6849	0.6932
0.3	0.6695	0.6690	0.7266	0.6527	0.6813	0.6798
1	0.6471	0.6560	0.7023	0.6327	0.6700	0.6616
3	0.6609	0.6947	0.7075	0.6489	0.6594	0.6743
10	0.6759	0.7272	0.7159	0.6701	0.6734	0.6925

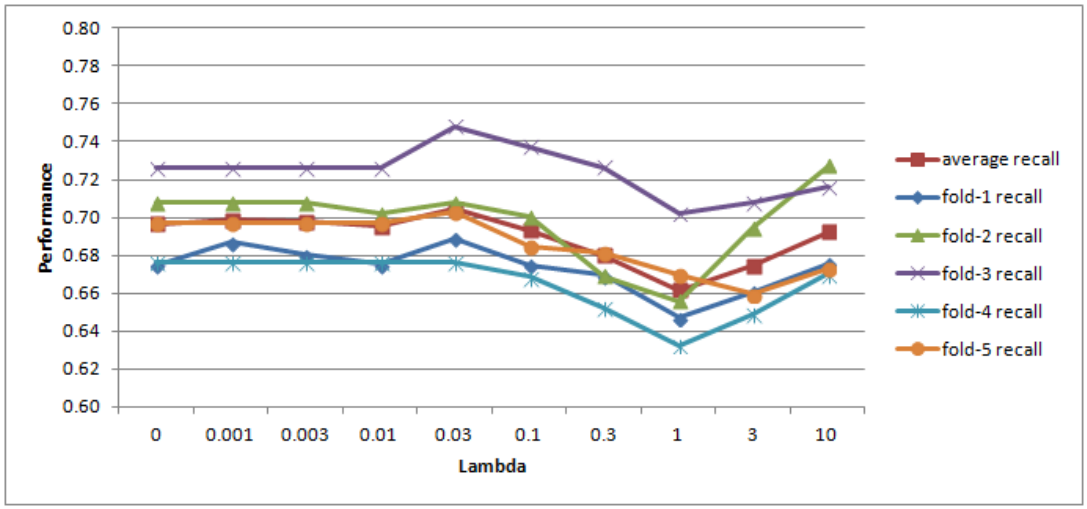

Figure 4.19: Average recall values of each fold for corresponding lambda values for segment-based basic CRF model.

Table 4.13: Average f-score values of each fold for corresponding lambda values for segment-based basic CRF model.

Lambda	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Average F-Score
0	0.7317	0.7327	0.7381	0.6820	0.7206	0.7215
0.001	0.7405	0.7344	0.7445	0.6820	0.7206	0.7248
0.003	0.7353	0.7344	0.7445	0.6845	0.7160	0.7233
0.01	0.7299	0.7304	0.7361	0.6845	0.7219	0.7210
0.03	0.7384	0.7298	0.7570	0.6922	0.7223	0.7282
0.1	0.7462	0.7311	0.7652	0.7021	0.7186	0.7331
0.3	0.7533	0.7392	0.7701	0.7072	0.7326	0.7409
1	0.7381	0.7288	0.7568	0.7083	0.7277	0.7323
3	0.7450	0.7517	0.7477	0.7151	0.7147	0.7352
10	0.7547	0.7637	0.7494	0.7228	0.7313	**0.7449**

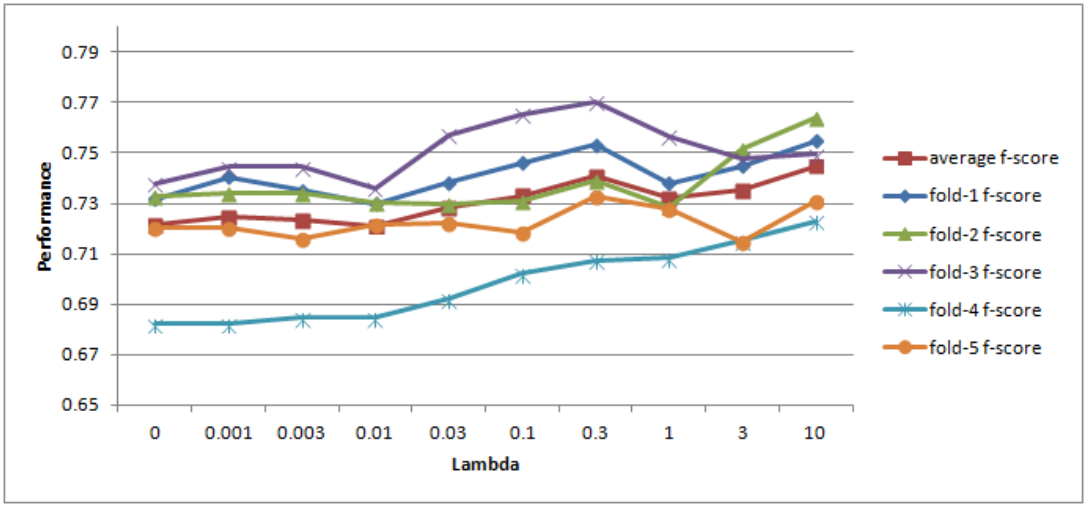

Figure 4.20: Average f-score values of each fold for corresponding lambda values for segment-based basic CRF model.

Table 4.14: Average precision, recall and f-score values and standard deviation in the recall and precision values for various lambda values for segment-based basic CRF model.

Lambda	Avg. Precision	Avg. Recall	Std. Precision	Std. Recall	F-Score
0	0.7485	0.6965	0.1164	0.1126	0.7211
0.001	0.7526	0.6989	0.1166	**0.1083**	0.7243
0.003	0.7511	0.6976	0.1175	0.1110	0.7229
0.01	0.7486	0.6954	0.1158	0.1125	0.7205
0.03	0.7533	**0.7047**	0.1100	0.1090	0.7278
0.1	0.7778	0.6932	0.1035	0.1209	0.7327
0.3	0.8140	0.6798	0.1135	0.1262	0.7404
1	**0.8199**	0.6616	0.1207	0.1251	0.7316
3	0.8083	0.6743	0.1164	0.1128	0.7350
10	0.8058	0.6925	**0.0964**	0.1098	**0.7448**

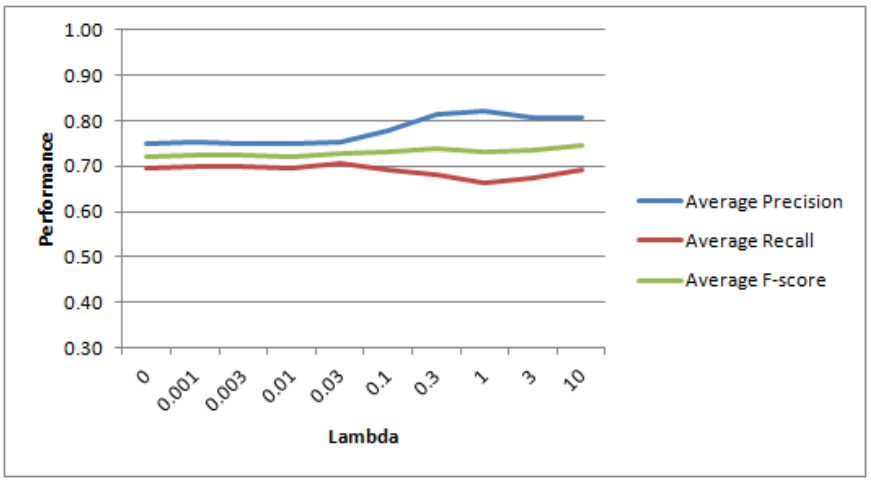

Figure 4.21: Average precision, recall and f-score values of corresponding lambda values for segment-based basic CRF model.

4.8 Classification with Contextual Conditional Random Fields

In this thesis, the proposed contextual conditional random fields model updates the initial label of a candidate region according to its spatial context. For this purpose, our first attempt is to construct a fully-connected model which contain the candidate region of the target class as its center node. After experimenting with fully connected model, we have decided that the interactions of cites in the graphical model should only occur by the help of the center node. In this way, its contextual influence can become more prominent. This approach is called the star CRF approach. The following subsections present the performance results obtained by each approach.

4.8.1 Fully-Connected Model

In [11], a fully-connected model is proposed for our problem. In the proposed model, there exist *fixed* seven nodes as central node represents the candidate region for the target class and other nodes represents LULC classes. Similar to our star model, fully connected model also constructed over meta-segments obtained by an initial SVM classification. However, absence of some LULC classes in the surroundings of a candidate region apart is disregarded, since the model is fixed, and behaves as if all classes are present in the neighborhood. Apart from being a fixed model, there are eight states of each node in this model, as the last state is added as a dummy state for representing the the mixed cases. The motivation behind this approach is meta-segments which may be constructed over the labeling of not-so-successful initial classification. In this case, some wrongly labeled segments would be combined with the correctly labeled segments and confuse the feature statistics of that meta-segments. We expect that the dummy state compensate the poor results from the initial step as well as the fixed structure of the model. Even though, mixed state is not handled during training phase, meaning that CRF is not fed with a sample that contain mixed state, the model is enabled to assign samples to eighth (mixed) state during testing phase. This issue may arises confusion during inference of the CRF and remains as a controversy of the model.

In this model, we use concatenation of the same features employed in the initial LULC regions classification step. However, at this step, we extract these features from meta-segments and candidate region. For the pairwise features in this model, we have experimented with three different approaches, namely concatenation of unary features, difference of unary features and class co-occurrence frequencies introduced in the previous chapter. In the more formal sense of the latter approach, pairwise feature function can be written as $\phi_{ij}(Y_i, Y_j, \mathbf{X}) = [O_{ij}, A_{ij}, N_{ij}]$.

Table 4.15 summarizes the performance results obtained by this fully connected model. As observed from Table 4.15, fully-connected model suffers from over parametrization. With 7 nodes, 8 states and 21 edges, there exists 6552 parameters in this model. Furthermore, adding a void state which is called as mixed state, forces CRF to discard most of the samples if model

is not certain about the predicted label. The high precision and relatively low recall results rooted from this fact.

Table 4.15: Performance results for the fully-connected model with different edge feature selection strategies and different loss functions.

		Edge feature selection					
		Difference of node features		*Concatenation of node features*		*Class co-occurrence frequency features*	
Loss function	Pseudo Negative Log-likelihood	84.61	44.9	100	20.41	92	46.94
	Loopy Belief Propagation	85.71	48.98	100	20.41	**93.33**	**57.14**
		precision	**recall**	**precision**	**recall**	**precision**	**recall**

4.8.2 Star Model

It is plausible to take into account that some classes may not exist in the neighborhood of a candidate region. In that case, instead of forcing the existence all the nodes as in the case of fully-connected model, which resulted in poor performance values, only the existing nodes should be considered. To make this possible, the model should not be fixed, but should be rather dynamic and change according to the case of each candidate region. The proposed star model changes dynamically according to the scene of each candidate region.

Another issue to be addressed is the few samples in the dataset, considering the dynamic graph samples. For instance, there may be only a couple of samples where there exist only water and urban classes in the neighborhood of the candidate region. Since the dataset is limited, emerging of overfitting problem is quite likely. In order to deal with this issue, we have paid great attention to regularization process. We have experimented with ten different regularizer values. Similar to the basic CRF experiments, the dataset is divided to 5 folds to investigate the training effects of different portions of the samples and each experiment is repeated 10 times to discard randomness effect of each run.

Star model is investigated in two different ways, namely parameter sharing and full-parametrization approaches. Following sections presents the performance results obtained by these methods respectively.

4.8.2.1 Parameter Sharing

In the parameter-shared star CRF model, there exists 7 states for each node. Meaning that, each node can be assigned to one of the seven classes. In other words, a soft-max classifier, which is the multi-class version of a logistic classifier, is used at the heart of the conditional random fields. For our problem, when parameters are shared across nodes and states, number of parameters to optimize is reduced to 130 in total. Even though number of parameters is small, training a soft-max classifier rather than a logistic classifier is more prone to over-fitting. For that reason, regularizer (lambda) effect is carefully examined.

Lambda-fold analysis performances are presented in Table 4.16, 4.17, 4.18 and illustrated with Figure 4.22, 4.23 and 4.24.

Table 4.16: Average precision values of each fold for corresponding lambda values for CS-CRF model with shared parameters.

Lambda	**Fold-1**	**Fold-2**	**Fold-3**	**Fold-4**	**Fold-5**	**Average Precision**
0	0.9172	0.9074	0.8277	0.9107	0.8983	**0.8923**
0.001	0.8963	0.8961	0.8159	0.8886	0.9125	0.8819
0.003	0.8940	0.8975	0.7931	0.8958	0.8986	0.8758
0.01	0.9040	0.8927	0.7885	0.9135	0.8981	0.8793
0.03	0.8938	0.8929	0.7856	0.9104	0.8937	0.8753
0.1	0.8990	0.8654	0.7781	0.8908	0.8950	0.8657
0.3	0.8784	0.8662	0.8009	0.8727	0.8804	0.8597
1	0.8424	0.8427	0.7336	0.8521	0.8459	0.8233
3	0.8302	0.8011	0.7051	0.7522	0.7919	0.7761
10	0.9096	0.7649	0.7113	0.7200	0.7712	0.7754

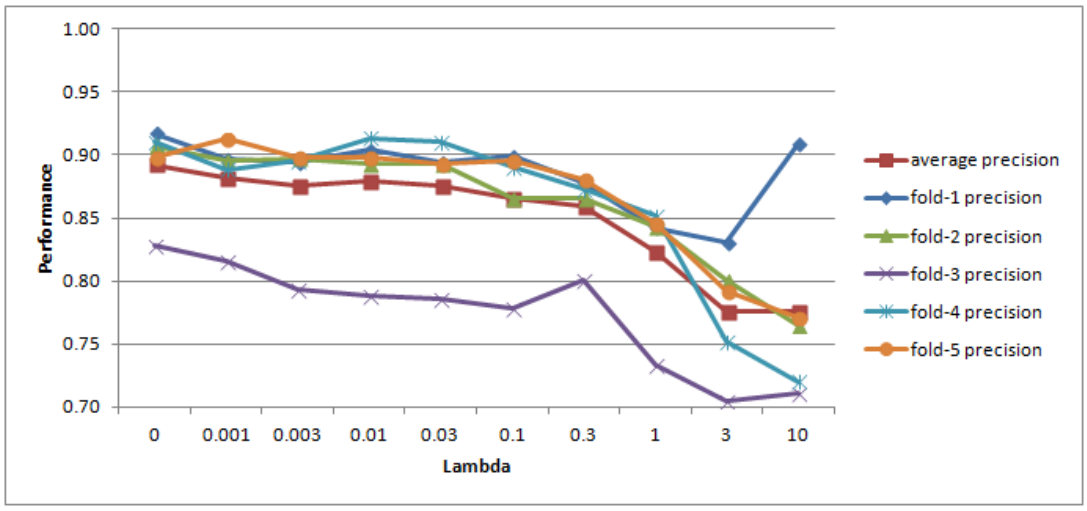

Figure 4.22: Average precision values of each fold for corresponding lambda values for CS-CRF model with shared parameters.

Table 4.17: Average recall values of each fold for corresponding lambda values for CS-CRF model with shared parameters.

Lambda	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Average Recall
0	0.7228	0.7881	0.7312	0.7864	0.7535	0.7564
0.001	0.7639	0.7871	0.7421	0.7877	0.7862	**0.7734**
0.003	0.7639	0.7812	0.7588	0.7824	0.7791	0.7731
0.01	0.7600	0.7840	0.7414	0.7848	0.7767	0.7694
0.03	0.7388	0.7840	0.7051	0.7705	0.7445	0.7486
0.1	0.6821	0.7007	0.6674	0.7223	0.6828	0.6911
0.3	0.6120	0.6550	0.6146	0.6182	0.5980	0.6196
1	0.4946	0.5296	0.5241	0.4848	0.4675	0.5001
3	0.4524	0.4630	0.4573	0.4072	0.3996	0.4359
10	0.3958	0.3611	0.3699	0.3480	0.3372	0.3624

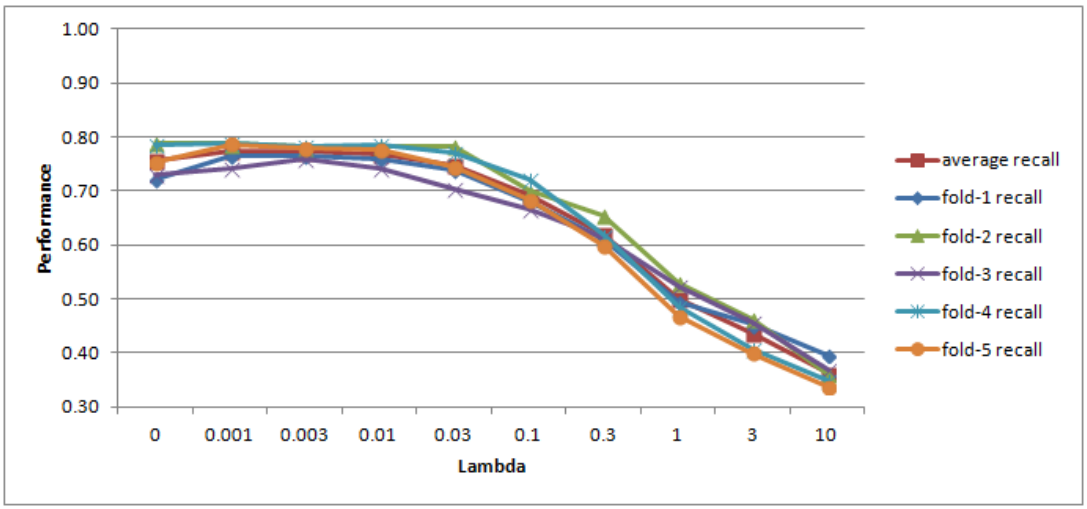

Figure 4.23: Average recall values of each fold for corresponding lambda values for CS-CRF model with shared parameters.

Table 4.18: Average f-score values of each fold for corresponding lambda values for CS-CRF model with shared parameters.

Lambda	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Average F-Score
0	0.8085	0.8435	0.7765	0.8440	0.8195	0.8187
0.001	0.8249	0.8380	0.7772	0.8351	0.8447	**0.8241**
0.003	0.8239	0.8353	0.7756	0.8353	0.8346	0.8212
0.01	0.8258	0.8348	0.7642	0.8443	0.8330	0.8207
0.03	0.8089	0.8349	0.7432	0.8346	0.8123	0.8070
0.1	0.7757	0.7744	0.7185	0.7978	0.7746	0.7686
0.3	0.7214	0.7459	0.6955	0.7237	0.7122	0.7201
1	0.6233	0.6504	0.6114	0.6180	0.6022	0.6222
3	0.5856	0.5868	0.5548	0.5284	0.5312	0.5583
10	0.5516	0.4906	0.4867	0.4693	0.4693	0.4940

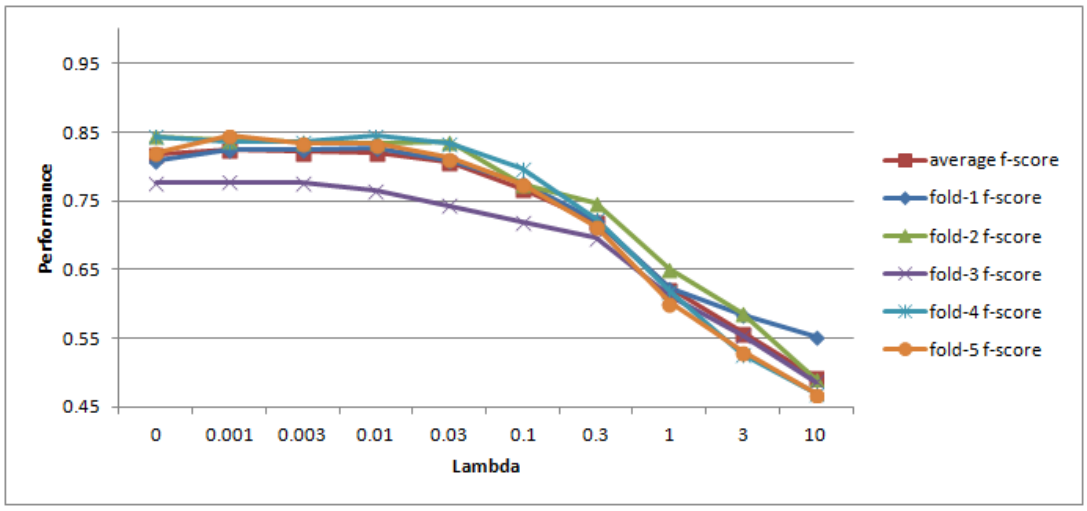

Figure 4.24: Average f-score values of each fold for corresponding lambda values for CS-CRF model with shared parameters.

Performance values averaged over folds and runs with varying regularizer (lambda) values are presented in Table 4.14 and illustrated with Figure 4.21. According to these performance values, best choices for the regularizer value seem to be 0.001, 0.003 or 0.01 which maximizes the f-score value.

Table 4.19: Average precision, recall and f-score values and standard deviation in the recall and precision values for various lambda values for CS-CRF model with shared parameters.

Lambda	Avg. Precision	Avg. Recall	Std. Precision	Std. Recall	F-Score
0	**0.8923**	0.7564	0.0883	0.1234	0.8184
0.001	0.8819	**0.7734**	**0.0844**	0.1112	**0.8240**
0.003	0.8758	0.7731	0.0873	0.1098	0.8212
0.01	0.8793	0.7694	0.0922	**0.1068**	0.8206
0.03	0.8753	0.7486	0.0951	0.1097	0.8067
0.1	0.8657	0.6911	0.1061	0.1307	0.7683
0.3	0.8597	0.6196	0.1117	0.1117	0.7198
1	0.8233	0.5001	0.1285	0.1168	0.6221
3	0.7761	0.4359	0.1414	0.1211	0.5580
10	0.7754	0.3624	0.1728	0.1111	0.4937

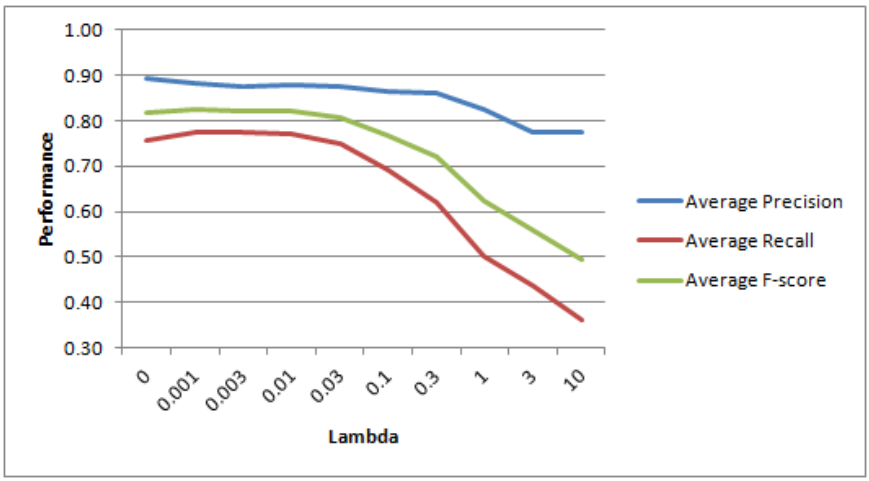

Figure 4.25: Average precision, recall and f-score values of corresponding lambda values for CS-CRF model with shared parameters.

From Figure 4.25, it is clearly seen that performance values drops after a point as the regularizer (lambda) value increases. This demonstrates that overfitting occurs in the parameter-shared star CRF model. This is expected considering the limited dataset size. Nevertheless, this model produces competitive results for some lambda values and it would be more powerful when there exists more data. With this model, final label of the target class can be predicted as well.

After pondering about how to overcome overfitting during the classification of our dataset, we come up with the idea of a fully-parameterized model where each node can take only two states, meaning that a soft-max classifier is switched to multiple logistic classifiers approach. The following section presents the results of that model.

4.8.2.2 Full Parameterization

In the fully parameterized version of the star model, each node has two states, meaning that its initial label is true or not. This kind of representation is enough, since we are only interested in whether the candidate region belongs to the target class or not. When all the states of all the nodes have separate parameters, there exists 374 parameters in the proposed star model in total.

Overfitting concern due to limited dataset size also applies to this case. In order to deal with limited sample size problem, we have proposed to populate samples by constructing complementary graphs of the sample in different combinations. In this way, we also handle absence of negative samples to feed logistic classifier during training phase. We generate *numberOfNodes-2* number of negative samples for each positive sample in the original region of interest by shifting one node features to the next one for each node except seed node. For

instance, for a graph with three nodes, namely soil class node, concrete class node and seed node, first negative graph sample is generated by considering the features of soil class belong to concrete class and features of concrete class belong to urban class. In this way, 5 negative samples for this graph sample is generated as can be observed from Figure 4.26.

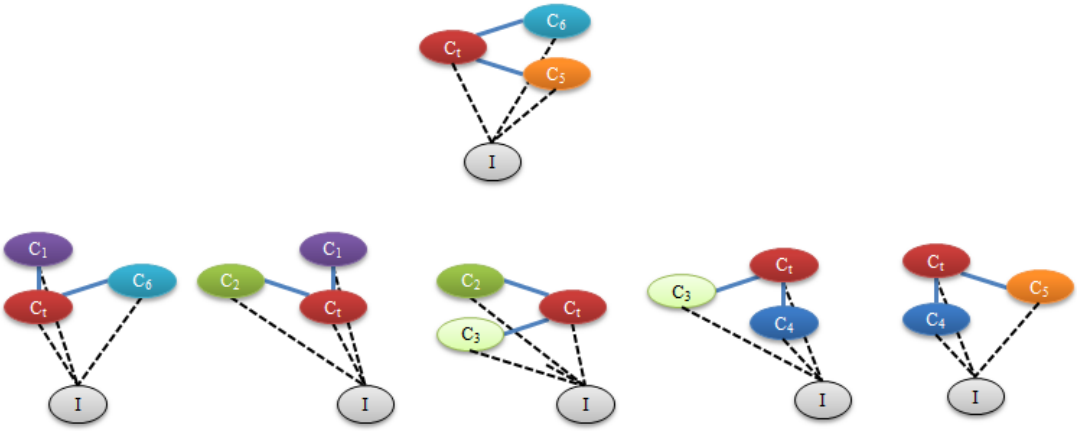

Figure 4.26: A sample graph with three nodes (upper), and its generated negative sample graphs (lower).

Like the parameter sharing case, dynamic model property still holds for the fully parameterized star CRF model.

Lambda-fold analysis performances are presented in Table 4.20, 4.21, 4.22 and illustrated with Figure 4.27, 4.28 and 4.29.

Table 4.20: Average precision values of each fold for corresponding lambda values for full-parameterized CS-CRF model.

Lambda	**Fold-1**	**Fold-2**	**Fold-3**	**Fold-4**	**Fold-5**	**Average Precision**
0	0.8300	0.8195	0.7385	0.8414	0.8205	0.8100
0.001	0.8301	0.8244	0.7403	0.8483	0.8162	0.8119
0.003	0.8308	0.8249	0.7435	0.8624	0.8149	0.8153
0.01	0.8307	0.8286	0.7553	0.8691	0.8203	0.8208
0.03	0.8375	0.8349	0.7693	0.8766	0.8335	0.8304
0.1	0.8550	0.8303	0.7840	0.8741	0.8367	0.8360
0.3	0.7613	0.6641	0.6527	0.6994	0.6646	0.6884
1	0.7301	0.6447	0.5997	0.6539	0.5999	0.6457
3	0.7070	0.6534	0.6070	0.6718	0.6319	0.6542
10	0.8676	0.8424	0.7976	0.8485	0.8638	**0.8440**

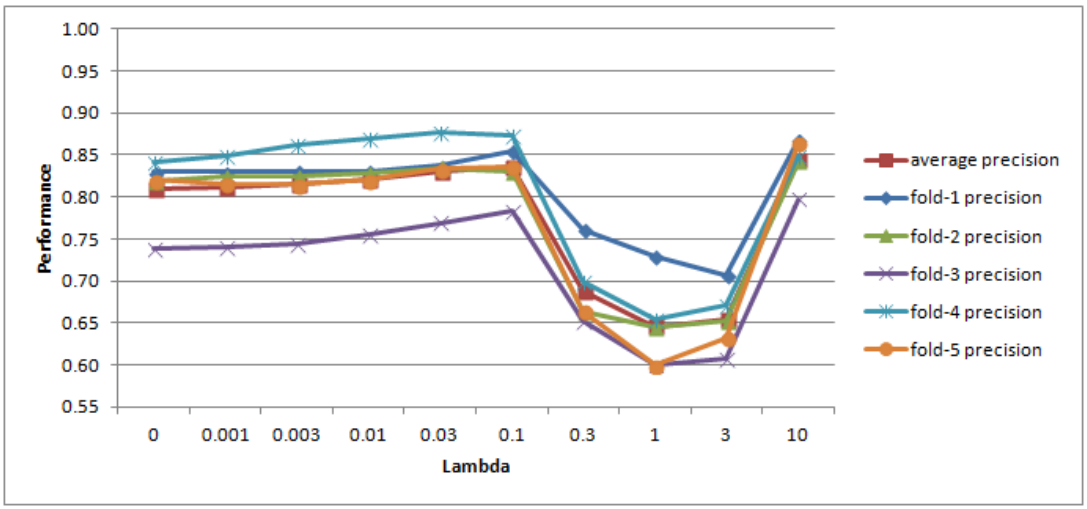

Figure 4.27: Average precision values of each fold for corresponding lambda values for full-parameterized CS-CRF model.

Table 4.21: Average recall values of each fold for corresponding lambda values for full-parameterized CS-CRF model.

Lambda	**Fold-1**	**Fold-2**	**Fold-3**	**Fold-4**	**Fold-5**	**Average Recall**
0	0.8273	0.8781	0.8050	0.8859	0.8969	0.8587
0.001	0.8287	0.8899	0.8063	0.8973	0.8956	0.8636
0.003	0.8318	0.8909	0.8080	0.8953	0.8965	0.8645
0.01	0.8264	0.8919	0.8200	0.8925	0.8983	0.8658
0.03	0.8257	0.8958	0.8180	0.8894	0.9004	0.8658
0.1	0.8210	0.8874	0.8117	0.8779	0.8827	0.8561
0.3	0.8658	0.9247	0.9208	0.9500	0.9347	**0.9192**
1	0.8756	0.9012	0.9445	0.9396	0.9802	**0.9282**
3	0.9101	0.9249	0.9246	0.9309	0.9592	**0.9299**
10	0.7444	0.7454	0.7521	0.7484	0.7418	0.7464

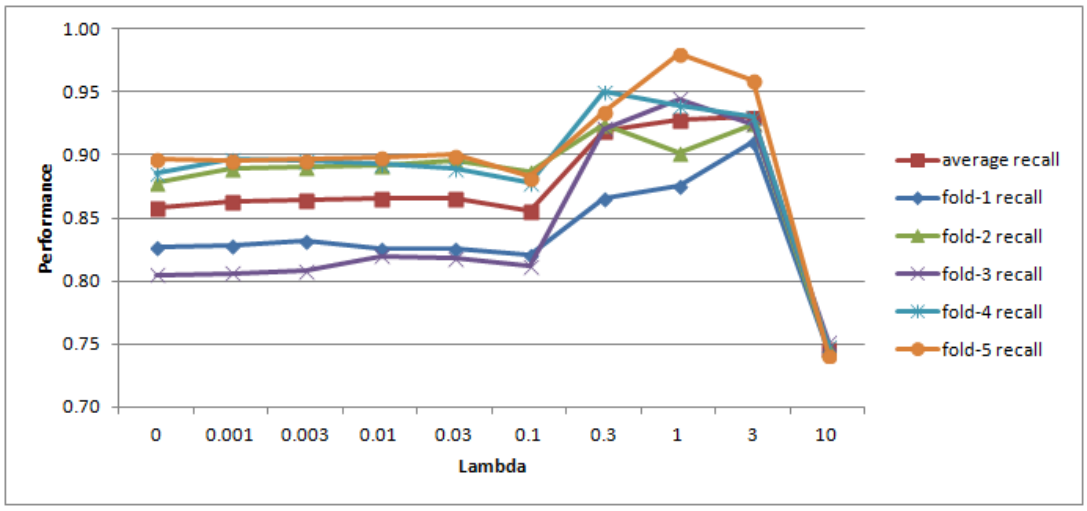

Figure 4.28: Average recall values of each fold for corresponding lambda values for full-parameterized CS-CRF model.

Table 4.22: Average f-score values of each fold for corresponding lambda values for full-parameterized CS-CRF model.

Lambda	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Average F-Score
0	0.8286	0.8478	0.7703	0.8631	0.8570	0.8336
0.001	0.8294	0.8559	0.7719	0.8721	0.8541	0.8369
0.003	0.8313	0.8566	0.7744	0.8785	0.8538	0.8392
0.01	0.8285	0.8591	0.7863	0.8807	0.8575	0.8427
0.03	0.8316	0.8643	0.7929	0.8829	0.8656	**0.8477**
0.1	0.8376	0.8579	0.7976	0.8760	0.8591	0.8460
0.3	0.8102	0.7730	0.7639	0.8057	0.7768	0.7872
1	0.7963	0.7517	0.7336	0.7711	0.7443	0.7616
3	0.7958	0.7658	0.7328	0.7804	0.7619	0.7681
10	0.8013	0.7910	0.7742	0.7954	0.7982	0.7922

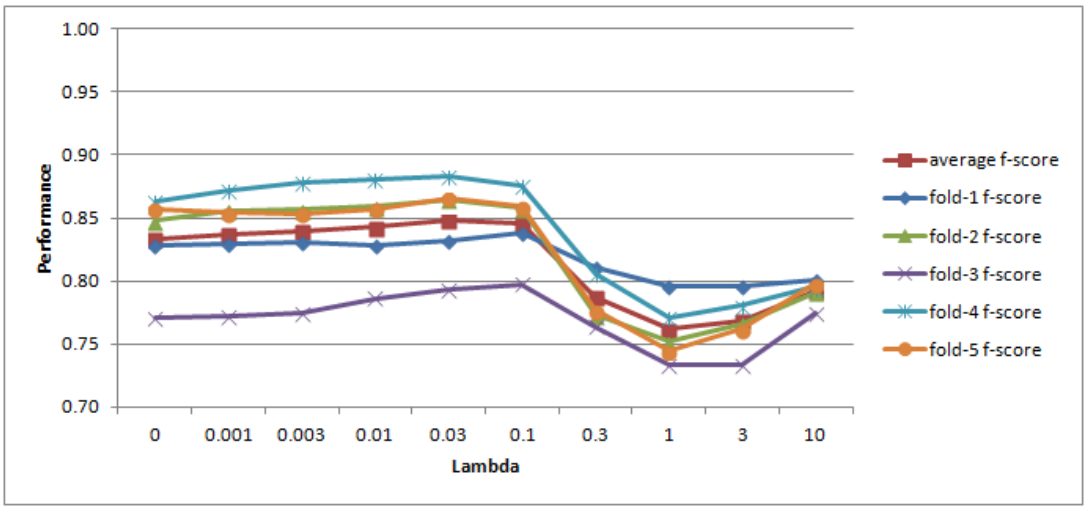

Figure 4.29: Average f-score values of each fold for corresponding lambda values for full-parameterized CS-CRF model.

Performance values averaged over folds and runs with varying regularizer (lambda) values are presented in Table 4.14 and illustrated with Figure 4.21. According to these performance values, best choices for the regularizer value seem to be 0.003 or 0.01 which maximizes the f-score value.

Table 4.23: Average precision, recall and f-score values and standard deviation in the recall and precision values for various lambda values for full-parameterized CS-CRF model.

Lambda	**Avg. Precision**	**Avg. Recall**	**Std. Precision**	**Std. Recall**	**F-Score**
0	0.8100	0.8587	0.1056	0.0925	0.8335
0.001	0.8119	0.8636	0.1039	0.0915	0.8369
0.003	0.8153	0.8645	0.1035	0.0940	0.8391
0.01	0.8208	0.8658	0.0996	0.0954	0.8426
0.03	0.8304	0.8658	**0.0969**	0.0959	**0.8476**
0.1	0.8360	0.8561	0.0987	0.1002	0.8459
0.3	0.6884	0.9192	0.1690	0.0962	0.7835
1	0.6457	0.9282	0.1669	**0.0871**	0.7575
3	0.6542	**0.9299**	0.1585	0.0947	0.7667
10	**0.8440**	0.7464	0.0980	0.0962	0.7920

Table 4.24: Comparison of performances of experimented CRF models.

Model	Avg. Precision	Std. Precision	Avg. Recall	Std. Recall	Avg. F-Score
Segment-based Basic CRF	0.8058	0.0964	0.6925	0.1098	0.7448
Fully-connected CRF	**0.9333**	0.0666	0.5714	0.1558	0.7088
Star CRF (parameter shared)	0.8819	0.0844	0.7734	0.1112	0.8240
Star CRF (full parameterization)	0.8304	0.0969	**0.8658**	0.0959	**0.8476**

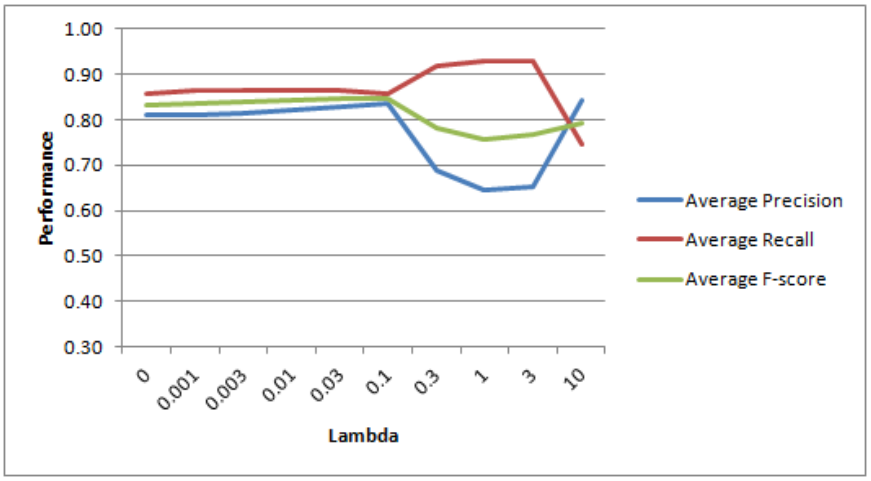

Figure 4.30: Average precision, recall and f-score values of corresponding lambda values for full-parameterized CS-CRF model.

From Figure 4.30, it is clearly seen that performance values are more stable compared to parameter sharing case as the regularizer (lambda) value increases. As lambda increases, the decrease in the performance values is less than the parameter sharing case. This demonstrates that multiple binary logistic classifier approach handles the overfitting problem better than the soft-max classifier when there is limited data. Since soft-max classifier is more complex, regardless of number of parameters learned in the model, if there are not enough samples in the dataset, it is hard to avoid memorization.

A comparison of all the CRF models experimented over our dataset is provided in Table 4.24. Even though, the highest precision is acquired by the fully-connected model, which is most probably due to the introduction of mixed state, average recall value of this approach is quite poor. It is observed from the table that star CRF model with full parameterization performs quite well in terms of both precision and recall values. Therefore star model with full parameterization has the highest f-score value.

On the other hand, in [22], Firat et.al. propose an unsupervised rule-based approach to detect airfields in multispectral satellite images and they report average precision value of as 0.75 and average recall value as 0.85. Even if our dataset differs from their dataset, we see that our star CRF model with full-parameterization outperform their rule-based approach which contains a lot of hand-tuned parameters. To our intuition, our model is more general and would outperform their rule-based approach when applied to a different dataset.

4.9 Chapter Summary

In this chapter, methodologies experimented with during the formation of the proposed model are discussed. First of all, the dataset constructed for the experiments is introduced. Then, the algorithms tried for the first stage of the proposed framework are stated. First stage of the framework consists of segmentation and an initial classification steps. For the segmentation step, watershed algorithm, SLIC algorithm and mean shift algorithm are examined and mean shift algorithm is found to be more suitable for our problem due to its detail preserving nature. Then, for the initial classification, performance of SVM and k-NN classifiers are compared and SVM is chosen to proceed to the contextual model construction phase. At the last part of the chapter, conditional random fields approaches which are experimented with are explained and the performance results are presented.

CHAPTER 5

CONCLUSION AND DISCUSSION

In this thesis, we have studied the effects of context integration for classifying the objects in remote sensing images. In this chapter, a summary of the proposed model is presented along with the possible future directions.

5.1 Summary

A contextual model, which emphasizes spatial interactions and co-occurrence statistics between a "contextually consistent" target and its surroundings, is proposed in this thesis. The proposed model had three stages, namely candidate region extraction step, meta-segment extraction step and conditional random fields step.

For extracting candidate regions, we have made use of domain knowledge. For the target class we have experimented with in our study, namely airfield, long parallel lines are known to be the indicator due to its nature. This is a compositional contextual cue which we have shown to be statistically meaningful by employing sparse autoencoders. It is observed that sparse autencoders captive the low level features which are shared across all the images of target object.

In the second stage, we introduce a new representation of image, called meta-segments, where the segments with the same label are merged according to an initial classification result. This novel representation enables us to gain a rough intuition about the surrounding regions of the target class which are then used for computing co-occurrence statistics and spatial interactions. Since our focus is on the final label of the target class, the information loss during merging of the LULC segments is does not have a significant impact on the final decision.

At the last stage, a conditional random field with star topology is constructed by placing the candidate region at the center and existing LULC classes in its neighborhood are added to the model as the surrounding nodes. We have solved separate conditional random field models for each candidate region. The proposed star model is dynamic, in the sense that only existing regions in the neighborhood of the candidate region are represented by a node in the graph of that candidate region. For example; if there are only soil and urban classes in the neighbor-

hood of a candidate region, its corresponding graph has only these three nodes. For increasing sample size, we have populated each graph samples with five negative graph examples. This is done by generating samples with non-existent nodes in the neighborhood of a candidate region. The pairwise potentials are designed to demonstrate the spatial interactions such that co-occurrence statistics in three different neighborhoods are used for the computation.

In chapter 4, we compare our proposed contextual star model to classic segment-based basic CRF model and to fully connected model. We also investigate the effects of parameter sharing across nodes.

Other comparative remarks in chapter 4 are made about the selection of segmentation algorithm and the initial classifier. Considering the detail preserving nature of the mean shift clustering, which is essentially needed in our problem since we deal with the airfields that are long thin regions, it has selected to proceed to the next steps. After comparing, SVM and k-NN classifiers, SVM is selected as the initial classifier since it performed slightly better than k-NN for our data.

5.2 Open Issues and Future Directions

This study has contributed for solving the problem of complex target detection in remote sensing data. However, our work is a preliminary study and there are points to be examined further some of which are discussed below in this section.

First of all, target class in our study is restricted to airfield class and we cannot guarantee the performance of other classes with the same feature set. Especially, for defining context and contextual relations, each class in remote sensing domain are considered separately in general, since each class exhibit different characteristics. Our sparse autoencoder approach brings a systematic to this process and leads the researchers about contextual cues of the target class. However, its effectiveness about contextual cue identification should be examined more with the experiments on other classes.

Nevertheless, our contextual model is not limited to remote sensing domain and can be applied to other spatial classification tasks as well. For instance, object detection in natural images can be accomplished by our contextual model where the contextual neighborhood defined. The only essential difference would arise during the feature extraction part. Apart from that additional contextual features can be utilized for obtaining pairwise potentials.

Another issue is about the error analysis of initial classification step. Effects of initial classification labeling can be examined more thoroughly, for deciding whether the CRF step is able to recover from a poor initial classification or not. Confusion analysis in CRF step due to misclassified and merged meta-segments deserves more attention and it would be part of a future analysis study.

In order to generalize our contextual model, to apply for different variants of classification

tasks, feature extraction steps should be automatized. One of the main challenges for the future work is to develop an algorithm for learning both unary and pairwise features from data by the help of sparse autoencoders. We believe this will be a thrilling study and most probably improve the results.

Another possible way to proceed is turning our model to a hierarchical architecture. This can be achieved by altering the energy function, in other words by adding extra terms for the interactions between and within additional layers. If our model is designed as a two-layer model, and at the upper layer the interactions between segments are studied, our model may become more powerful and our problem is converted into scene decomposition from single target classification, since the labeling of the LULC segments become possible as well.

Additionally, employing the probabilities from the initial classifier to our contextual model may bring advantage at the CRF stage and would be examined in later works.

REFERENCES

[1] A Hierarchical and Contextual Model for Aerial Image Parsing. *International Journal of Computer Vision*, 88(2):254–283, 2009.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.

[3] J. Anderson. *A land use and land cover classification system for use with remote sensor data*, volume 2001. US Govt. Print. Off., 1976.

[4] P. Bandzi, M. Oravec, and J. Pavlovicova. New statistics for texture classification based on gabor filters. *RADIOENGINEERING-PRAGUE-*, 16(3):133, 2007.

[5] U. Bayram, G. Can, S. Duzgun, and N. Yalab\ik. Evaluation of textural features for multispectral images. In *Proceedings of SPIE*, volume 8180, page 81800I, 2011.

[6] O. Besbes, N. Boujemaa, and Z. Belhadj. Cue Integration for Urban Area Extraction in Remote Sensing Images. In *Image Analysis and Recognition, 6th International Conference, ICIAR 2009*, pages 248–257, 2009.

[7] S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. *OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED-*, 34:433–433, 1992.

[8] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 1. Springer New York, 2006.

[9] F. Bovolo and L. Bruzzone. A context-sensitive technique based on Support Vector Machines for image classification. *Pattern Recognition and Machine Intelligence Proceedings*, 3776:260–265, 2005.

[10] G. Can, O. Firat, and F. Vural. Contextual object recognition with conditional random fields. In *Signal Processing and Communications Applications Conference (SIU), 2013 21st*, pages 1–4, 2013.

[11] G. Can, O. Firat, and F. Yarman Vural. Conditional random fields for land use/land cover classification and complex region detection. In G. Gimel'farb, E. Hancock, A. Imiya, A. Kuijper, M. Kudo, S. Omachi, T. Windeatt, and K. Yamada, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 7626 of *Lecture Notes in Computer Science*, pages 216–224. Springer Berlin Heidelberg, 2012.

[12] P. Carbonetto. *Unsupervised statistical models for general object recognition*. PhD thesis, The University of British Columbia, 2003.

[13] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[14] T. Chang and C.-C. Kuo. Texture analysis and classification with tree-structured wavelet transform. *Image Processing, IEEE Transactions on*, 2(4):429–441, 1993.

[15] P. Clifford. Markov random fields in statistics. *Disorder in physical systems*, pages 19–32, 1990.

[16] A. Coates and A. Y. Ng. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade*, pages 561–580. Springer, 2012.

[17] T. Cocks, R. Jenssen, A. Stewart, I. Wilson, and T. Shields. The HyMap airborne hyperspectral sensor: the system, calibration and performance. In M. Schaepman, D. Schläpfer, and K. I. Itten, editors, *Proceedings of the 1rd EARSeL Workshop on Imaging Spectrometry*, pages 37–42. Integrated Spectronics Pty Ltd, 1998.

[18] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[19] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[20] V. Dey, Y. Zhang, and Zhong. A REVIEW ON IMAGE SEGMENTATION TECHNIQUES WITH. *Symposium A Quarterly Journal In Modern Foreign Literatures*, XXXVIII:31–42, 2010.

[21] O. Firat. liborf - machine learning toolkit for deep learning, probabilistic graphical models and structured learning/prediction, Aug. 2013. Software available at `http://www.ceng.metu.edu.tr/~e1697481/libORF.html`.

[22] O. Firat, O. Tursun, and F. Vural. Application of context invariants in airport region of interest detection for multi-spectral satellite imagery. In *Signal Processing and Communications Applications Conference (SIU), 2012 20th*, pages 1–4, 2012.

[23] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[24] B. J. Frey and D. J. MacKay. A revolution: Belief propagation in graphs with cycles. *Advances in neural information processing systems*, pages 479–485, 1998.

[25] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.

[26] J. Gleason, A. V. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis. Vehicle detection from aerial imagery. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2065–2070. IEEE, 2011.

[27] R. C. Gonzalez and E. Richard. Woods, digital image processing. *ed: Prentice Hall Press, ISBN 0-201-18075-8*, 2002.

[28] S. Gould. *Probabilistic Models for Region-based Scene Understanding*. PhD thesis, Stanford University, June 2010.

[29] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.

[30] D. Grangier, L. Bottou, and R. Collobert. Deep convolutional networks for scene parsing. In *ICML 2009 Deep Learning Workshop*, volume 3. Citeseer, 2009.

[31] S. E. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on gabor filters. *Image Processing, IEEE Transactions on*, 11(10):1160–1167, 2002.

[32] Z. Guo, L. Zhang, and D. Zhang. Rotation invariant texture classification using lbp variance (lbpv) with global matching. *Pattern Recognition*, 43(3):706–719, 2010.

[33] J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. 1971.

[34] R. M. Haralick, K. Shanmugam, and I. H. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, 1973.

[35] X. He. *Learning structured prediction models for image labeling*. PhD thesis, Citeseer, 2008.

[36] X. He, R. S. Zemel, and M. A. Carreira-Perpinán. Multiscale conditional random fields for image labeling. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–695. IEEE, 2004.

[37] Y. Heymann, C. Steenmans, G. Croissille, and M. Bossard. CORINE Land Cover. Technical Guide. *Official Publications of the European Communities*, (November):1–94, 1994.

[38] T. Hoberg and F. Rottensteiner. Classification of Settlement Areas in Remote Sensing Imagery. *ISPRS Technical Commission VII Symposium 100 Years ISPRS Advancing Remote Sensing Science*, XXXVIII:53–58, 2010.

[39] C. hoon Lee, R. Greiner, and M. Schmidt. Support vector random fields for spatial classification. In *In Proc. of PKDD 2005*, pages 121–132, 2005.

[40] X. Huang and J. Jensen. A machine-learning approach to automated knowledge-base building for remote sensing image analysis with GIS data. *Photogrammetric engineering and remote sensing*, 63(10):1185–1193, 1997.

[41] Y. Huang, A. Yue, S. Wei, D. Li, M. Luo, Y. Jiang, and C. Zhang. Texture feature extraction for land-cover classification of remote sensing data in land consolidation district using semi-variogram analysis. *W. Trans. on Comp*, 7:857–866, 2008.

[42] W. Jiang, S.-F. Chang, and A. C. Loui. Context-Based Concept Fusion with Boosted Conditional Random Fields. In *2007 IEEE International Conference on Acoustics Speech and Signal Processing ICASSP 07*, volume 1, pages 949–952. Ieee, 2007.

[43] G. H. John, R. Kohavi, K. Pfleger, et al. Irrelevant features and the subset selection problem. In *ICML*, volume 94, pages 121–129, 1994.

[44] I. Jolliffe. *Principal component analysis*. Springer series in statistics. Springer-Verlag, 1986.

[45] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[46] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE TRANSACTIONS ON INFORMATION THEORY*, 47:498–519, 1998.

[47] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[48] S. Kumar. *Models for learning spatial interactions in natural images for context-based classification*. PhD thesis, Carnegie Mellon University, 2005.

[49] S. K. S. Kumar and M. Hebert. Discriminative random fields: a discriminative framework for contextual interaction in classification, 2003.

[50] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, ICML '01, pages 282–289, 2001.

[51] A. Laine and J. Fan. Texture classification by wavelet packet signatures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1186–1191, 1993.

[52] L. Lin, T. Wu, J. Porway, and Z. Xu. A stochastic graph grammar for compositional object representation and recognition. *Pattern Recognition*, 42(7):1297–1307, 2009.

[53] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[54] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[55] S. K. McFeeters. The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. *International Journal of Remote Sensing*, 17(7):1425–1432, 1996.

[56] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999.

[57] A. Ng. Sparse autoencoders.

[58] T. Ojala, M. Pietikäinen, and T. Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *Computer Vision-ECCV 2000*, pages 404–420. Springer, 2000.

[59] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[60] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[61] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[62] T. Roncevic, M. Braovic, and D. Stipanicev. Context-based natural image parsing: A critical survey. In *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pages 1–5. IEEE, 2011.

[63] R. Roscher, B. Waske, and W. Forstner. Kernel Discriminative Random Fields for land cover classification, 2010.

[64] M. Schmidt. Ugm: Matlab code for undirected graphical models, Aug. 2013. Software available at `http://www.di.ens.fr/~mschmidt/Software/UGM.html`.

[65] B. Sirmacek and C. Unsalan. Building detection using local gabor features in very high resolution satellite images. In *Recent Advances in Space Technologies, 2009. RAST '09. 4th International Conference on*, pages 283–286, 2009.

[66] R. Socher, C. C. Lin, A. Ng, and C. Manning. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 129–136, 2011.

[67] W. L. Stefanov, M. S. Ramsey, and P. R. Christensen. Monitoring urban land cover change : An expert system approach to land cover classification of semiarid to arid urban centers. *Remote Sensing of Environment*, 77:173–185, 2001.

[68] T. M. Strat and M. A. Fischler. Context-based vision: recognizing objects using information from both 2 d and 3 d imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1050–1065, 1991.

[69] C. Sutton and A. McCallum. An Introduction to Conditional Random Fields for Relational Learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*, chapter 4, page 93. MIT Press, 2006.

[70] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Computer Vision–ECCV 2010*, pages 352–365. Springer, 2010.

[71] K. W. Tobin, B. L. Bhaduri, E. A. Bright, A. Cheriyadat, T. P. Karnowski, P. J. Palathingal, T. E. Potok, and J. R. Price. Large-scale geospatial indexing for image-based retrieval and analysis. In *Advances in Visual Computing*, pages 543–552. Springer, 2005.

[72] S. Tuermer, J. Leitloff, P. Reinartz, and U. Stilla. Evaluation of selected features for car detection in aerial images. In *ISPRS Hannover Workshop, High-Resolution Earth Imaging for Geospatial Information*, 2011.

[73] P. Turney. The identification of context-sensitive features: A formal definition of context for concept learning. 1996.

[74] O. van Lier, J. Luther, D. Leckie, and W. Bowers. Development of large-area land cover and forest change indicators using multi-sensor Landsat imagery: Application to the Humber River Basin, Canada. *International Journal of Applied Earth Observation and Geoinformation*, 13(5):819–829, Oct. 2011.

[75] R. R. Vatsavai, A. Cheriyadat, and S. Gleason. Unsupervised semantic labeling framework for identification of complex facilities in high-resolution remote sensing images. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 273–280. IEEE, 2010.

[76] R. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(4):722–732, 2010.

[77] J. Wang and R. Swendsen. Nonuniversal critical dynamics in monte carlo simulations. *Physical review letters*, 1987.

[78] T. Wu, G.-S. Xia, and S. C. Zhu. Compositional boosting for computing hierarchical image structures. In *CVPR*. IEEE Computer Society, 2007.

[79] Y. Yang and S. Newsam. Comparing sift descriptors and gabor texture features for classification of remote sensed imagery. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1852–1855. IEEE, 2008.

[80] S. C. Zhu, Y. N. Wu, and D. Mumford. Minimax Entropy Principle and Its Application to Texture Modeling. *Neural Computation*, 9(8):1627–1660, 1997.

APPENDIX A

DATASET

Figure A.1: Red-green-blue combination of I_1.

Figure A.2: Red-green-blue combination of I_2.

Figure A.3: Red-green-blue combination of I_3.

Figure A.4: Red-green-blue combination of I_4.

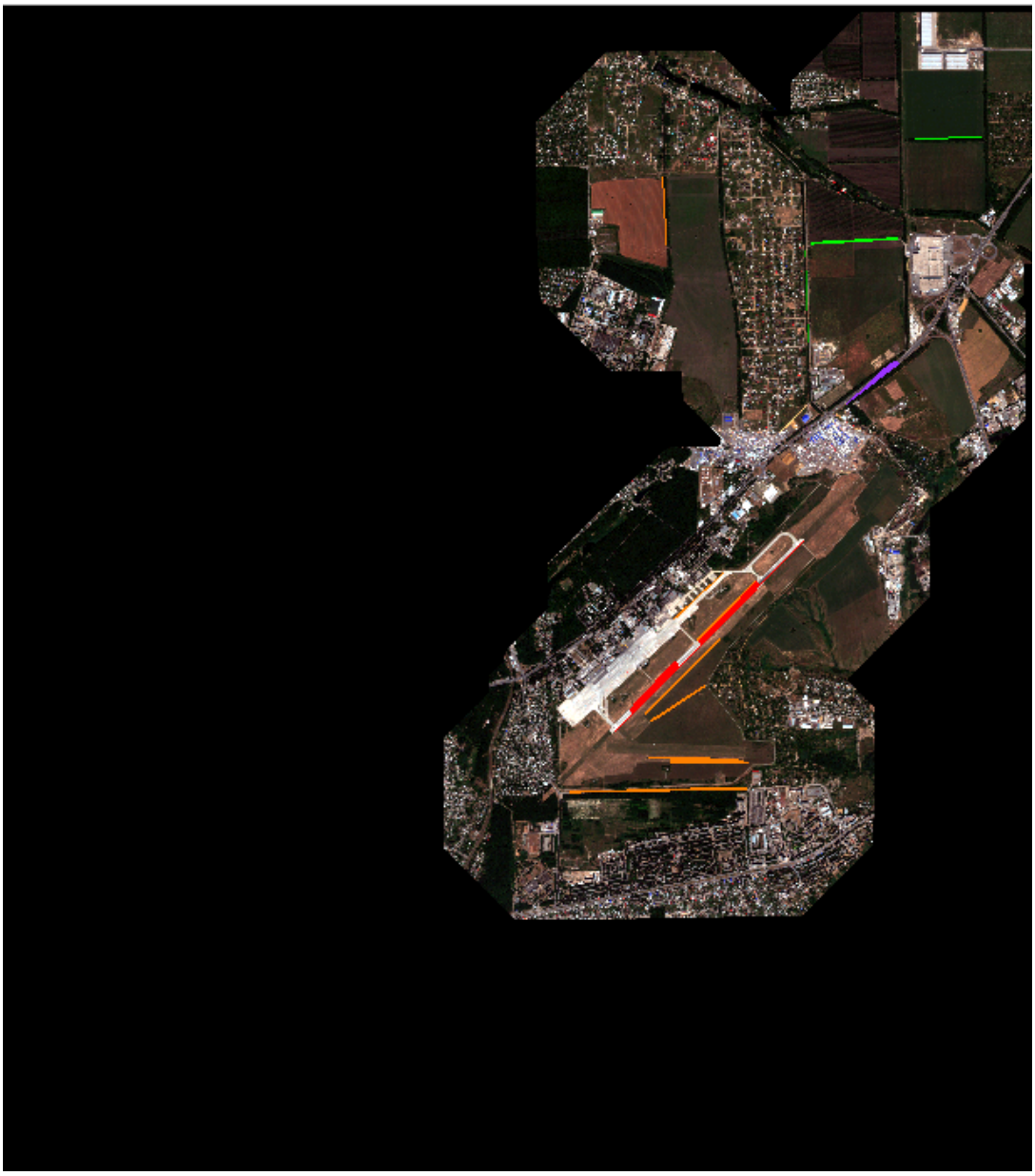

Figure A.5: Region of interest in I_1 obtained by dilation and superposition of all candidate PLBRs. In this image, each PLBR is marked by the color of its corresponding class. Class colors are red for airfield class, orange for soil, purple for urban, cyan for concrete or asphalt, dark green for forest, green for agricultural land and blue for water.

Figure A.6: Region of interest in I_2 obtained by dilation and superposition of all candidate PLBRs. In this image, each PLBR is marked by the color of its corresponding class. Class colors are red for airfield class, orange for soil, purple for urban, cyan for concrete or asphalt, dark green for forest, green for agricultural land and blue for water.

Figure A.7: Region of interest in I_3 obtained by dilation and superposition of all candidate PLBRs. In this image, each PLBR is marked by the color of its corresponding class. Class colors are red for airfield class, orange for soil, purple for urban, cyan for concrete or asphalt, dark green for forest, green for agricultural land and blue for water.

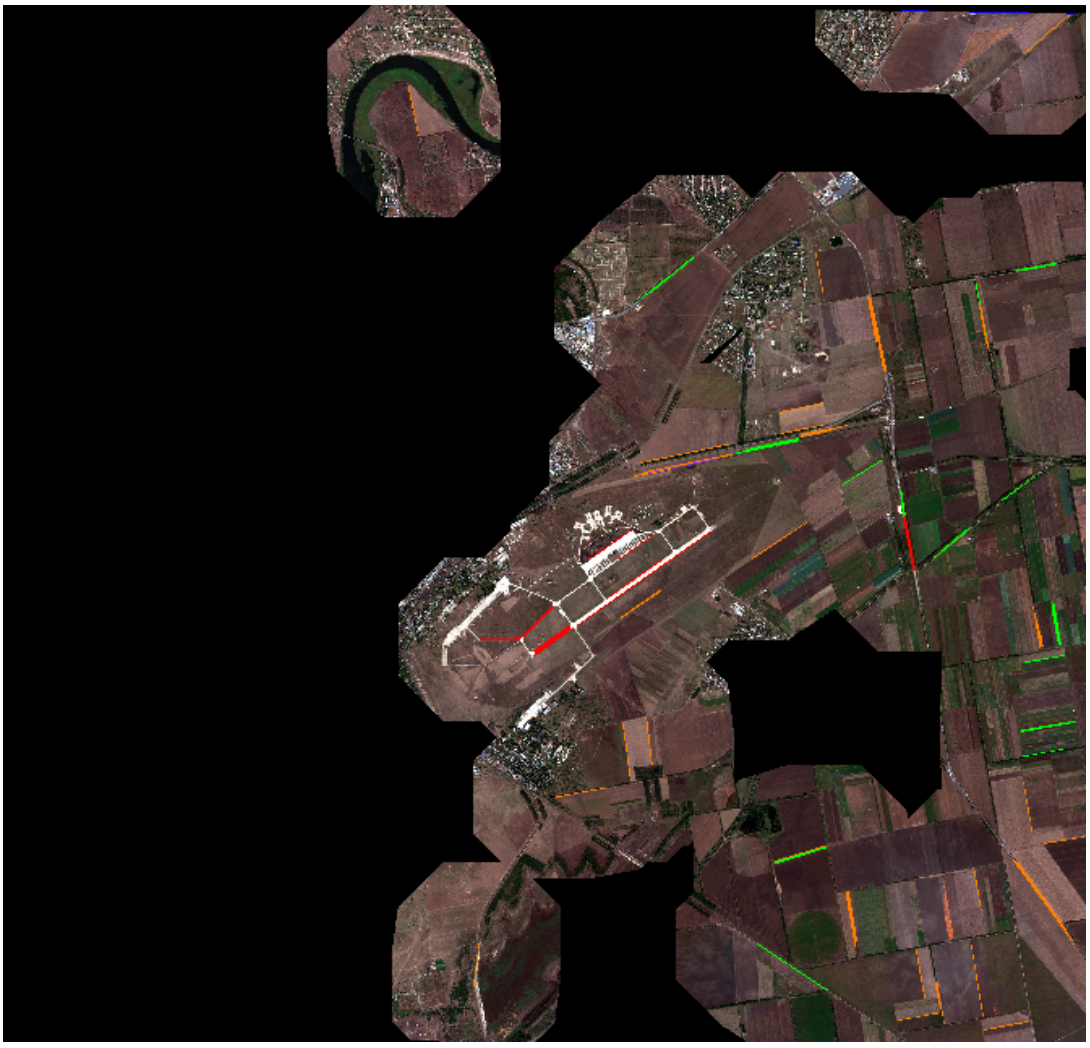

Figure A.8: Region of interest in I_4 obtained by dilation and superposition of all candidate PLBRs. In this image, each PLBR is marked by the color of its corresponding class. Class colors are red for airfield class, orange for soil, purple for urban, cyan for concrete or asphalt, dark green for forest, green for agricultural land and blue for water.