Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Deutsches Zentrum
für Luft- und Raumfahrt

Investigating the Separability of Slums in Mumbai based on RADARSAT Images using Object-based Image Analysis Methods

Bachelor Thesis

Hochschule Karlsruhe - University of Applied Sciences

Faculty of Information Management and Media

Bachelor Degree Program Geo-Information Management

Author:	Tatjana Bürgmann
Matriculation number:	36427
E-mail address:	tatjana.buergmann@gmail.com
First advisor:	Prof. Dr.-Ing. Berthold Pfeiffer
Second advisor:	Prof. Dr.-Ing. Gertrud Schaab
External advisor:	Dr. Michael Wurm, DLR
Date of submission:	06/30/2015
Duration:	3 months

Karlsruhe, June 2015

HOCHSCHULE KARLSRUHE – UNIVERSITY OF APPLIED SCIENCES
Faculty of Information Management and Media – Bachelor Degree Program Geo-Information Management

B a c h e l o r t h e s i s

for Ms. Tatjana Bürgmann

Topic: Investigating the Separability of Slums in Mumbai based on RADARSAT Images using Object-based Image Analysis Methods

Research Issue:

Through global urbanization, slums are expanding on a vast scale. Especially in developing countries and mega cities like Mumbai, India, migration rates are remarkably high and are supposed to rise even further in the future. With the development of new technologies and techniques, remote sensing is being increasingly utilized in the mapping of urban land cover classes. The unique spatial structure shared by slums in the same geographic area allows for the accurate outlining of these areas. Many studies have been conducted using optical remote sensing data but only a few have investigated the use of Radar data for slum classification.

Aim of the Thesis:

The goal of this research is to classify slum areas and other urban land cover classes throughout the city of Mumbai using multi-polarized *RADARSAT* images. For this purpose different approaches and methods are investigated. Using the software eCognition various texture features are calculated for the different polarizations and Kennaugh elements of the Radar images. For the classification an object-based approach, applying the classification algorithms Linear Discriminant Analysis, Support Vector Machine and Random Forest, is performed, using the programming language R. The effect of the used combinations of texture features, polarizations and Kennaugh elements on the classification is statistically analyzed. Specifically, the following elements of the work are to be carried out:

1. Literature research on basic methods of slum classification using Radar data.
2. Calculation and subsequent statistical analysis of textural characteristics of blocks of buildings for all polarizations and Kennaugh elements.
3. Object-based classification using Linear Discriminant Analysis, Support Vector Machine and Random Forest.
4. Analyzing the influence of individual features on the classification result.
5. Accuracy analysis of the classification results.

After submission of the thesis the subject matter will be presented within a 30-minute colloquium. The topic is also to be prepared graphically on a poster and also for an internet presentation.

Duration: 3 months

Date of Issue: _____

Date of Submission: _____ Prof. Dr.-Ing. B. Pfeiffer

Acknowledgements

At this point I would like to thank everyone, who has supported and helped me in accomplishing this work.

First of all I would like to greatly thank my advisor Dr. Michael Wurm of DLR Earth Observation Center in Oberpfaffenhofen for enabling me to write this thesis and providing constant support, motivation and insight during this work.

Furthermore I want to express my sincere gratitude towards my professors and supervisors from Hochschule Karlsruhe, Prof. Dr.-Ing. Berthold Pfeiffer and Prof. Dr.-Ing. Gertrud Schaab, for their kind assistance and advice.

I would also like to show my greatest appreciation to the European Space Agency (ESA) for providing the satellite imagery through the ESA TPM (Third Party Mission) proposal no. 21071 titled “Investigation on the use of dual-polarized C-Band data for the detection and characterization of informal settlements”.

Moreover I would like to thank Dr. Andreas Schmitt (DLR Oberpfaffenhofen) for preprocessing the data and patiently answering all my questions about the multi-polarized Radar data and Kennaugh elements. Likewise a big thank you is also due to Julia Kuner for providing the accurately digitized and attributed blocks of buildings of Mumbai.

Special thanks also goes to my family, for their continuous love, support and encouragement during my lifetime and also throughout the duration of this work.

Finally, I would like to thank my dad, Devon Libby and Matt O’Neill who have taken the time and made the effort to carefully and critically proofread this work.

Statutory Declaration

I, Tatjana Bürgmann, born on January 7, 1992 in Friedrichshafen, declare that I have authored this thesis independently, that I have not used other than the declared sources and resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources. This thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

Bürgmann, Tatjana

Karlsruhe, June 30, 2015

Abstract

Global urbanization and the subsequent development of mega cities, like Mumbai in India, drive the emergence of slums. For urban planners, disaster and aid organizations, among others, it is crucial to know the location of slum areas in order to enhance living conditions of the occupants and ensure prompt and organized aid in case of natural disasters. The importance of proper identification of these areas is exacerbated by the high population density and poorly developed infrastructure.

In this study, informal (slum) and formal settlements are classified throughout Mumbai using Radarsat-2 imagery. Optical remote sensing imagery has been widely used for urban classification purposes and therefore highly-developed techniques have evolved. On the contrary, the use of Radar remote sensing has not been as prevalent in the past, due to insufficiencies that were better met with the high resolution capabilities of optical remote sensing. However, with the advent of VHR Radar imagery this data source gained renewed interest for urban settlement classification, especially since it is capable of weather- and daytime-independent acquisition, limitations which are present in optical remote sensing imagery.

Since techniques for delineating informal settlements in urban environments using Radar imagery have not yet had the maturation time optical data has, they are still in need of further investigation. With the launch of Sentinel-1A in April 2014 and the launch of Sentinal-1B following in 2016, the utility of Radarsat-2 data for informal settlement classification may draw conclusions about the usefulness of the free and widely-available Sentinel-1 data, since specifications are very similar.

Various characteristics can be used in order to classify informal settlements. In this thesis several textural features were tested, specifically texture measures derived from Gray Level Co-occurrence Matrices, i.e. Homogeneity, Contrast, Dissimilarity, Entropy, Angular Second Moment, Mean, Standard Deviation and Correlation. Using the texture measures derived for all blocks of buildings of Mumbai, the following three different classification algorithms were investigated: Linear discriminant analysis, Support Vector Machines and Random Forest. The best result was achieved using Random Forest with an Overall Accuracy of 91% and a User's Accuracy for informal settlements of likewise 91%. Linear Discriminant Analysis showed an Overall Accuracy of 89% but a User's Accuracy for informal settlements of only 55%, while Support Vector Machines got an Overall Accuracy of 89% and a User's Accuracy of 64%. All three classifiers underestimated informal settlements to a certain degree.

Table of Contents

List of Abbreviations

ASM	Angular Second Moment
AUC	Area Under Curve
BSA	Back Scattering Alignment
CRAN	Comprehensive R Archive Network
CSA	Canadian Space Agency
EMR	Electromagnetic Radiation
ESA	European Space Agency
FSA	Forward Scattering Alignment
GLCM	Gray Level Co-occurrence Matrix
LDA	Linear Discriminant Analysis
MCGM	Municipal Corporation of Greater Mumbai
MMR	Mumbai Metropolitan Region
OBIA	Object-based Image Analysis
RAR	Real Aperture Radar
RF	Random Forest
ROC	Receiver Operating Characteristic
RS	Remote Sensing
SAR	Synthetic Aperture Radar
SLAR	Side-Looking Airborne Radar
SLR	Side-Looking Radar
SRTM	Shuttle Radar Topography Mission
SVM	Support Vector Machine
TSS	True Skill Statistic
UA	Urban Agglomeration
UN	United Nations
VHR	Very High Resolution

1 Introduction

In this chapter an introduction to the background of the thesis, the state of research in informal settlement classification, research objectives and the organization of this thesis will be given.

1.1 Background

Urbanization, the worldwide process of people migrating from rural to urban areas, is steadily increasing. In 2008, the number of people living in cities surpassed the 50 percent mark at about 3.2 billion people for the first time (see Figure 1-1). Today, over half of the world's population is residing in urban areas and the percentage is expected to grow even further in the future from 54% in 2014 to 66% by 2050. This trend is expected due to both migration and population growth (United Nations, 2014).

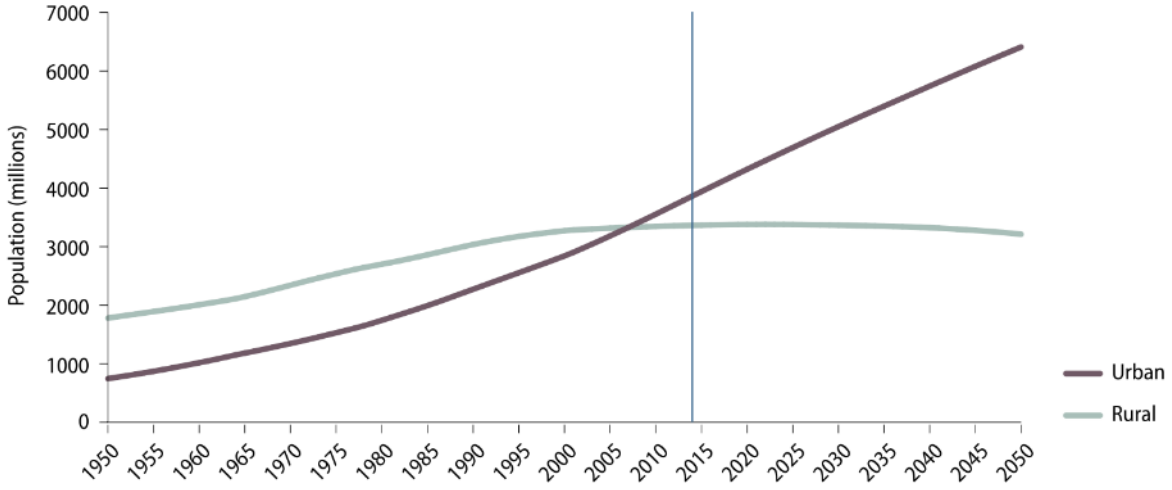


Figure 1-1: Urban and rural population of the world in millions, 1950 – 2050

Source: United Nations (2014)

Developing countries are expected to contribute most to this process, since cities in these countries are urbanizing faster than in other countries. Ninety percent of the urbanization process is supposed to take place in Asian and African cities. Asia' urban population is expected to rise from 48% to 64% by 2050. India, China and Nigeria are projected to comprise 37% of the expected growth of urban population between 2014 and 2050 with India adding most to the urban population (United Nations, 2014). India's population has more than doubled in the last fifty years from about 500 million to today's 1.3 billion. Its urban population, however, has risen nearly five times from 93 million in 1965 to 420 million in 2015 (Taubenböck & Kraff, 2014; Worldometers, 2015). Urbanization also drives the development of megacities, defined by the United Nations (UN) as cities having more than 10 million inhabitants. As of 2014, globally 28 megacities exist, above all Tokyo, Delhi, Shanghai, Mexico City, Mumbai and São Paulo, all having more than 20 million inhabitants. By 2030, the world is projected to be home to 41 megacities (United Nations, 2014).

Since administrative organizations cannot handle the great amount of people flocking into the city, the high rate of global urbanization results not only in the physical horizontal and vertical growth of urban areas, but also in congestion and housing shortages, thereby leading to a rapid increase and proliferation of slums (Adhikari, 2004; Graesser et al., 2012). The development of slums is a phenomenon especially seen in developing countries and is one of the primary indicators of urban poverty (Arimah, 2010; Taubenböck & Kraff, 2014). According to studies by the UN-Habitat, approximately a third of the urban population in developing countries were slum dwellers in 2012 (UN Habitat, 2012). This aspect is remarkably severe in India (Taubenböck & Kraff, 2014), where according to Census data from 2011 roughly 17.4% of urban households and about 65.4 million people live in slums (Census of India, 2011a; Shrinivasan, 2013). Besides urbanization and social and demographical change, weakness of institutions is mainly held responsible for the fact that the slum population in India is still the highest in the world (Kit & Lüdeke, 2013).

Considering that the Government of India projected India's slum population to exceed 100 million by 2015, efforts of authorities and non-governmental institutions to approach the problem are noticeable (Government of India, 2010). These efforts include programs for upgrading slums (Arimah, 2010) by implementing strategies to improve current infrastructure and install new water pipe systems, as well as supplying electricity and sewerage (Brunotte et al., 2002). More malicious solutions involve slum demolition and eviction (Adhikari, 2004). A good example for the political consciousness is the "Target Goal 7.D" of the Millennium Development Goals, that aimed to "achieve, by 2020, a significant improvement in the lives of at least 100 million slum dwellers" (United Nations, 2015), which was met before the deadline and even exceeded the goal by 100 million slum dwellers (United Nations, 2012b). This resulted in a decline of the world's slum population from 39% in 2000 to 33% in 2012. Nevertheless, slums still form significant part of many urban agglomerations and are currently home to about 828 million people worldwide (Kit & Lüdeke, 2013; United Nations, 2012a).

With respect to the present amount of slums, especially in developing countries, research in this field is still essential (Taubenböck & Kraff, 2014). According to various authors, existing spatial slum data is either obsolete, generalized, not available, not accurate enough or does not provide the information needed or is at best fragmented (Arimah, 2010; Hofmann, 2001a; Taubenböck & Kraff, 2014). Especially in the case of India, the reliability of national surveys and official statistics on slums has been criticized in the literature (Agarwal, 2011; Risbud, 2010; Satterthwaite, 2010). This might be caused by the fact that detecting slums is considered to be one of the most challenging tasks within urban remote sensing (Hofmann, 2001a; Kit & Lüdeke, 2013). The dynamic nature of slums makes it difficult to keep spatial data up-to-date, since the evolvement of slums is a steady process that might take decades while the removal of slums can be conducted in a few days (Kit & Lüdeke, 2013; Kohli et al., 2013).

Delineating and mapping slum boundaries is valuable for many important decision making bodies and institutions like urban planners and managers, disaster and aid organizations, policy makers, among many others on the local, national and international level (Graesser et al., 2012; Kohli et al., 2013; Kuffer et al., 2013). These organizations and authorities often lack accurate and up-to-date data which is essential to make plans for the development of the city, address typical issues in slums like poverty, high population density or a lack of basic

infrastructure and also to be able to provide improved post-disaster assistance after an event (Graesser et al., 2012; Kuffer et al., 2013). In order to evolve government plans and programs, almost continuous data is needed (Lillesand et al., 2004). The same holds true for Mumbai, where city planners were currently confronted with a lack of slum data concerning location, extent and type (Kuffer et al., 2013). In order to address the issues that come along with slums and to improve the lives of slum dwellers, it is essential to know where slums are located and how many people are affected.

Traditionally, census surveys have been used to collect information about the urban population, which were not able to accurately outline slum areas and also were not frequently updated (Purkis & Klemas, 2011). However, modern Remote Sensing (RS) methods are capable of providing physical, spatially disaggregated, area-wide, up-to-date information at variable scales at comparably low cost in order to spatially analyze the location and pattern of slums (Kohli et al., 2013; Taubenböck & Kraff, 2014). By refining remote sensing technologies and analysis techniques, remote sensing proved to be an accurate tool for land cover and land use mapping in urban areas (Graesser et al., 2012; Hofmann, 2001b; Hofmann et al., 2008; Jain, 2007; Kit & Lüdeke, 2013; Kohli et al., 2013; Niebergall et al., 2008; Owen & Wong, 2013; Shekhar, 2012; Taubenböck & Kraff, 2014). Particularly the development of very high resolution (VHR) remote sensing imagery has initiated numerous attempts to map urban structures, since it enables extraction of individual objects like buildings or roads. Spatial, structural, morphological and contextual characteristics can be used to accurately distinguish slum settlements from formal, residential, industrial or commercial districts (Graesser et al., 2012).

Another promising trend in urban remote sensing is the usage of Radar data, since Radar sensors are capable of operating independently from weather conditions and daylight. With the advent of VHR Radar sensors like Radarsat-2, TerraSAR-X, TanDEM-X, COSMO-Skymed or ALOS-PALSAR, various urban land-use studies have been conducted using Radar remote sensing data (Ban et al., 2010; D'Elia et al., 2014; Esch et al., 2005, 2006, 2010; Niu & Ban, 2013; Qi et al., 2012; Taubenböck et al., 2012). The planned launch of Radar satellite Sentinel-1B in early 2016 will be particularly promising for further research in this field, since it is supposed to provide quick data delivery (with a delay of only one hour) at a global scale (Ban et al., 2015). However, other studies have shown that urban slum identification is still challenging, given the large variability in definition and appearance (Kohli et al., 2013; Taubenböck & Kraff, 2014). Image classification is not a straightforward process either and the heterogeneous landscape of urban areas proves to be a difficult case (Graesser et al., 2012). Since delineating urban areas is still a difficult task, detecting different urban land use classes like informal settlements is even more challenging (Purkis & Klemas, 2011).

1.2 State of Research in Informal Settlement Classification using Radar Data

The application of remote sensing techniques for settlement classification was first introduced in the mid-1950's (Henderson & Xia, 1997). Since then, remote sensing systems operating in the optical spectrum of the electromagnetic radiation (EMR) have received far more attention and application in the urban environment than Radar systems, and therefore offer more advanced, in-depth explored, widely-used and manifold techniques e.g. object-based (Jacquin et al., 2008) vs. pixel-based approaches, fuzzy classification, (Tiwari et al., 2004), semantic classification (Taubenböck et al., 2009), Support Vector Machine (Tuia et al., 2009) or Random Forest (Zhang et al., 2014).

Techniques for informal settlement classification using optical RS data have been explored at a high level (Graesser et al., 2012; Hofmann, 2001a; Hofmann et al., 2008; Jain, 2007; Kit & Lüdeke, 2013; Kohli et al., 2013; Niebergall et al., 2008; Owen & Wong, 2013; Shekhar, 2012). Even single houses of informal settlements have already been successfully delineated using VHR data (Asmat & Zamzami, 2012).

However, Radar images offer considerable benefits over optical RS systems, mainly due to the all-day and all-weather acquisition capability. With the recent development of VHR SAR imagery along with an improvement of computational efficiency, the classification of settlement structures using SAR imagery has captured more and more attention in the last few years. The size of the objects that can be detected by the sensor is limited by the spatial resolution of the sensor e.g. with a resolution of 30 m it is not possible to detect single houses. Many studies have been conducted and thus numerous methods have been developed to map urban structures using HR SAR imagery like Radarsat-2. Similar systems include the German TerraSAR-X, the Italian CosmoSkyMed and the Japanese ALOS satellite. Up to date, TerraSAR-X and CosmoSkyMed are the SAR satellites achieving the highest spatial resolution of only 1 meter in special imaging modes.

Henderson & Xia (1997) were the first to investigate SAR remote sensing techniques for urban settlement classification in a systematic way. Many studies with the aim of classifying urban structures using SAR imagery chose an object-based over a pixel-based approach (Ban et al., 2010; D'Elia et al., 2014; Esch et al., 2005, 2006, 2010; Niu & Ban, 2013; Qi et al., 2012; Taubenböck et al., 2012), due to the highly textured nature and speckle of Radar imagery that might cause difficulties for the classification of urban areas when a pixel by pixel approach is employed (Zhang et al., 2014). Jensen (2005) also states that the spectral complexity of urban areas, due to the variety of materials used, contributes to the fact that per-pixel approaches are limited in urban classification.

Ban et al. (2010), Niu & Ban (2013) and Qi et al. (2012) also used Radarsat data in order to classify built-up areas. Ban et al. (2010) compared the results of object-based and rule-based classification in eCognition using multi-temporal Radarsat data compared to optical Quickbird data and the fusion of both. Using Quickbird data provided an overall accuracy of 87.9% (Kappa 0.868) while using Radarsat provided an overall accuracy of 86.6% (Kappa 0.852). The fusion of both produced a slight increase of overall accuracy at 89.5% (Kappa 0.885). The

accuracy values of single land-cover classes were significantly improved by fusion of Radar and multispectral RS data. Ban et al. (2010) used the multi-scale segmentation in eCognition. For the classification process NDVI was used to extract built-up area and further shape and texture measures i.e. Entropy, derived from a Gray-Level Co-occurrence Matrix, were used to extract commercial-industrial areas.

Zhang et al. (2014) also used a fusion of optical and Radar remote sensing data and the four GLCM texture measures Homogeneity, Dissimilarity, Entropy, and Angular Second Moment in order to reveal urban footprints of Guangzhou, Shenzhen and Hong Kong.

Niu & Ban (2013) also used multi-temporal Radarsat-2 data, both ascending and descending, and an SVM object-based classification approach in combination with rules to improve the result and yielded a very high Kappa value of 0.91. For SVM classification the library LIBSVM and a radial kernel was used, which is written in C++, but can also be used in R using the package e1071 (interface to LIBSVM).

Qi et al. (2012) used polarimetric decomposition, Pol-SAR interferometry, object-oriented image analysis and decision tree algorithms for classifying Radarsat-2 data in Guangzhou City, China and reached an overall accuracy of 86.64% and a Kappa value of 0.84. Multi-resolution segmentation was conducted in eCognition and a QUEST (Quick, Unbiased, Efficient, Statistical Tree) algorithm was used for decision tree classification. Texture measures were also used in many studies, especially measures derived from Gray Level Co-occurrence Matrices, like Correlation and Variance (Ban et al., 2015; Gamba & Aldrighi, 2012; Gamba et al., 2011).

While the studies mentioned above concentrated on urban land-cover classes, there are also studies which aimed at classifying slums in urban environments. Dell'Acqua et al. (2006) for example used an unsupervised (ISODATA) and a supervised approach for detecting formal and informal settlements in Sudan and Senegal from ENVISAT ASAR and 5m-resolution SPOT imagery. The aim of this study was to develop a semi-automatic algorithm for informal settlement detection. The supervised approach was based on a fuzzy neural network classification using the Gray Level Co-occurrence Matrix texture features Mean, Dissimilarity and Correlation. In a further study, Stasolla & Gamba (2008) used local autocorrelation and morphological processing for detecting formal and informal settlements from higher resolution Radar imagery recorded by Radarsat-1 and ALOS PALSAR.

Jiang et al. (2011) used an advanced SAR Interferometry (InSAR) technique, termed Interferometric Point Target Analysis (IPTA), to detect reclamation settlements in Macao, China using 22 ENVISAT ASAR images.

Taubenböck & Kraff (2014) used a combination of Landsat, TerraSAR-X, Quickbird and spatial data e.g. OpenStreetMap street data or building footprints to classify informal settlements based on building density, size and height in order to generate a classified 3D-Model.

Malcolm, Piwowar, Hall, Cotlier, & Ravenna (2014) have conducted a study to identify informal settlements in Rosario, Argentina using a combination of fine beam Radarsat and Landsat imagery. Although the Landsat image has a lower resolution (30 m) compared to Radarsat fine beam (8 m), it could support the rather coarse Radarsat data for classification. Malcolm et al.

used a supervised and an unsupervised classification approach whereby the supervised turned out to lead to better results using a maximum likelihood classifier. Using only multi-temporal Radarsat data no useful information could be obtained. However, by combining Landsat Band 4 and two Radarsat images from April and January (both filtered using the Enhanced Lee Filter) significant improvements could be achieved with an overall accuracy of 63.63% and a Kappa coefficient of 0.56. This way, 53.3% of informal settlements were classified correctly.

A similar study to this one was conducted by Weigand (2014), who attempted to classify urban land cover classes for a subset of Mumbai using multi-polarized TerraSAR-X imagery (HH/HV, VV/VH and HH/VV), texture measures derived from Gray Level Co-occurrence Matrices and a Linear Discriminant Analysis algorithm. The best classification accuracies achieved were 69.33% using an object-based approach and 69.95% using a kernel-based approach. This study has shown that multi-polarized SAR products like Kennaugh-elements add beneficial information for urban land cover classification.

1.3 Research Objectives

The objective of this thesis is to explore the potential of multi-polarized Radarsat-2 imagery to analyze spatial characteristics of slums in an urban context using object-oriented image analysis methods. For classification, the algorithms Linear Discriminant Analysis (LDA), Support Vector Machines (SVM) and Random Forest (RF) will be used to analyze the slum expansion throughout the whole megacity area of Mumbai, India. Textural characteristics derived from Gray Level Co-occurrence Matrices, used for the classification, will be calculated for all blocks of buildings, all polarizations (HH/HV and VV/VH) and Kennaugh elements (K_0, K_1, K_5, K_8, Absorption, Diattenuation, Polarizance, Retardance and Scale) within the software eCognition. The object-based image classification will be performed using the programming language R. A statistical analysis will examine the effect of various texture features, polarizations and Kennaugh elements on the classification result. Finally, an accuracy assessment will be conducted to identify the precision of the classification results and a comparison of the classification algorithms will be given. An overview of the tasks to be performed can be found in Figure 1-2.

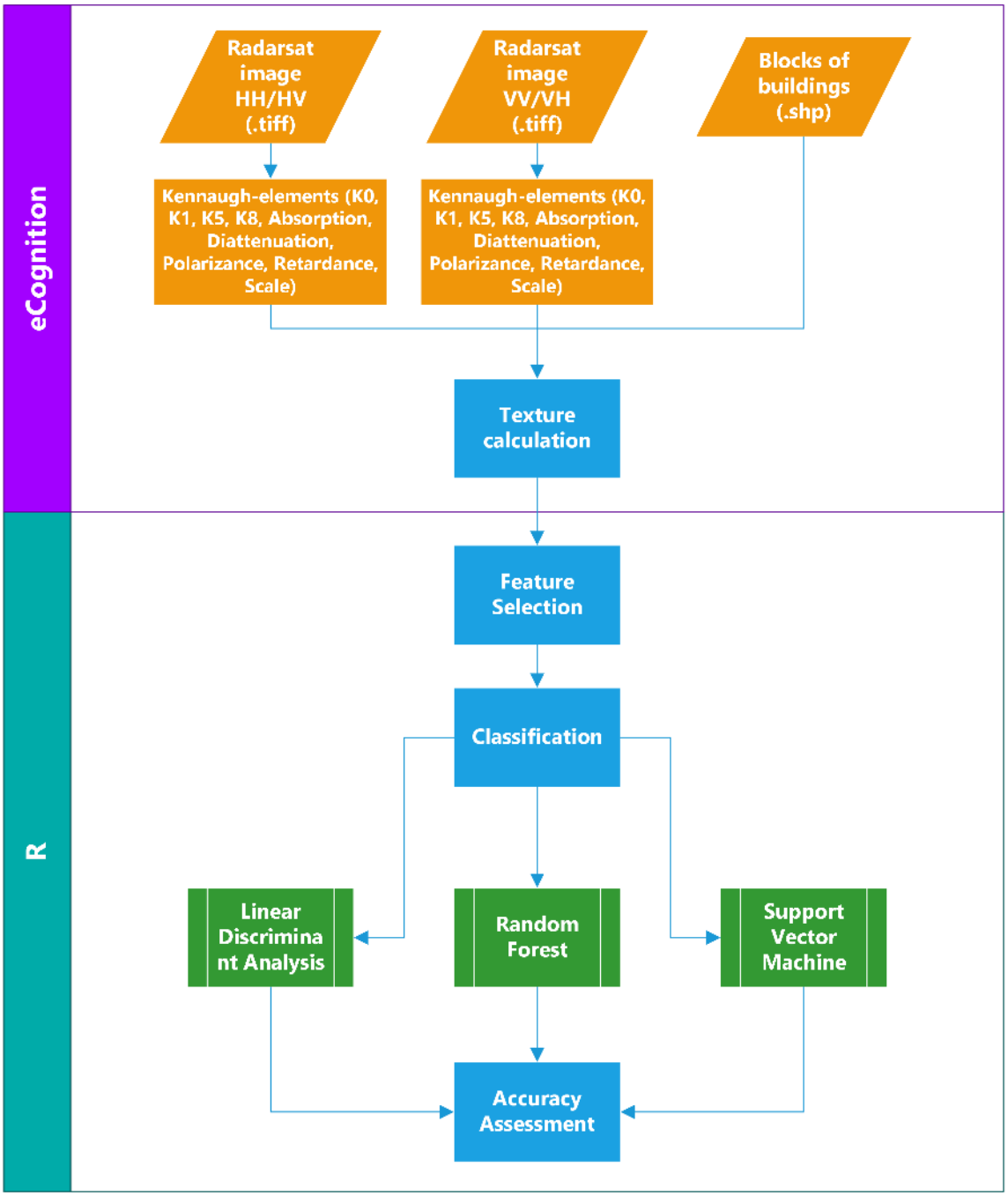


Figure 1-2: Flow chart showing an overview of this thesis

1.4 Thesis Organization

This thesis has been organized as follows. Chapter 1 gives an overview over research background and objectives. Further, the results of the literature research on basic methods of slum classification using Radar data are presented to reveal the state of research in this field. In Chapter 2 fundamental principles and methods in terms of definition of slums, principles of Radar data, polarization modes and current available SAR satellites, as well as the basics of texture analysis, Gray Level Co-occurrence Matrices, Object-based Image Analysis, the classification algorithms used and accuracy assessment will be illustrated.

Chapter 3 will shed light on the study area of the megacity Mumbai, India, concerning its geographical location, history, climate, administrative division and population with an emphasis on the slum population. Additionally, the data used in this study, remote sensing imagery as well as reference data, will be introduced.

Chapter 4 will present the software and the strategy of the data evaluation. The calculation methods of the texture measures, the feature selection, classification and accuracy assessment will be explained. Finally, in Chapter 5 the results of the classification and accuracy assessment will be shown and discussed afterwards. Furthermore the classification algorithms LDA, SVM and RF will be opposed in a comparison. Chapter 6 comprises the conclusion and will give an outlook on future research needs.

2 Principles of Radar Data and Slum Classification

In this chapter, a definition of slums will be given and the fundamentals of Radar data, texture analysis, Object-based Image Analysis, the classification algorithms used and accuracy assessment will be explained.

2.1 Slum Delineation

Since in present times, the term “slum” is part of the ordinary linguistic usage, its meaning seems obvious, although there is no definite and generally agreed on definition (Taubenböck & Kraff, 2014). However, for image-based classification, it is important to have a clear conceptualization of slums as the object of interest (Kohli et al., 2013). Therefore a major challenge is that existing definitions of slums are diverse, objective and vague. Although their appearance varies across the globe, nevertheless many issues remain the same e.g. poverty, high population density, and a lack of basic infrastructure (Graesser et al., 2012).

There are various synonyms used for the term slum such as “informal settlement”, “squatter settlement”, “shanty town” or “ghetto”. Across the globe different languages also developed various designations e.g. “Favela” in Portugese or “Biidonvilles” in French. In India there are even several terms for slums in different geographical locations: In Maharashtra the term “Zopadpattis” is used whereas in Mumbai “Chawls” is used for rental one-room tenements that were provided by factory owners for low-income workers. “Patra chawls” is used for semi-permanent housing of authorized as well as unauthorized type. Workers living in hutments built on pavements close to their working place are called “Pavement Dwellers” (UN-Habitat, 2003). However, in this work, the terms “slum” and “informal settlement” will be used consistently and interchangeably for all kinds of deprived areas with poor slum like living conditions.

The Slum Areas (Improvement and Clearance) Act, established in India 1956, defines slums as

“[…] any area that the buildings in that area -

a) Are in any respect unfit for human habitation; or
b) Are by reason of dilapidation, overcrowding, faulty arrangement and design of such buildings, narrowness or faulty arrangement of streets, lack of ventilation, light or sanitation facilities, or any combination of these factors, are detrimental to safety, health or morals” (Government of India, 1956).

The UN-Habitat Expert Group Meeting (EGM) on slum indicators states that a slum is a contiguous settlement whose inhabitants are characterized as having inadequate housing and basic services. The UN-Habitat further specifies basic services as durable housing, sufficient living space, easy access to drinking water and adequate sanitation, as well as security of tenure (UN Habitat, 2007). Schools, health centers, post offices, roads or electricity are usually obsolete in slums (Adhikari, 2004).

Slums are often settlements that were not implicitly planned by the city authorities but were built by residents themselves without any approval (Kohli et al., 2013). However, the variety of slums ranges from unauthorized settlements to regularized areas that are just lacking basic

services and amenities (Kuffer et al., 2013). Additionally, they are often built in detrimental environmental conditions along urban infrastructure like railway tracks, highways or the airport area (Bhagat & Jones, 2013) or even in hazardous areas such as flood plains, landfills, steep slopes, swamps or chemical dumps (Kuffer et al., 2013).

While most slums are generally positioned at the periphery of the city, Dharavi, one of the largest and most well-known slums, located in Mumbai, is placed in the city center, since its origins go back to the British colonial era when the Koli fishermen were relocated and the city kept on spreading around the slum (see Chapter 3.1 for details). The spatial delineation and population estimates of the Dharavi slum vary greatly from different organizations, due to the fuzzy qualitative measures mentioned above. Inhabiting 300,000 to 1 million people, depending on the source, Dharavi is one of the most densely populated places in the world (Taubenböck & Kraff, 2014).

From a remote sensing perspective, it is important to know the physical or morphological characteristics that distinguish slums from other urban area, to be able to clearly delineate them. These unique spatial characteristics that slum areas usually share in contrast to residential, industrial or commercial structures include:

- High to very high building density
- Small building sizes and narrow streets
- Continuous and irregular settlement and street pattern
- High variance in building materials
- Lack of open space or green areas
- Close spatial proximity to railroads, highways or hazardous zones

However, slums differ in size, location, history, population, culture and appearance, even throughout the same city. According to Kuffer et al. (2013) this phenomenon can also be found in Mumbai, where they classified five different types of slums based on field work and local knowledge ranging from claimed land along transport infrastructure to 2-3 story buildings with open space in between. Formal areas on the other hand commonly have larger building sizes, a relatively regular building layout, more homogeneous building materials and more space in between buildings (Graesser et al., 2012; Kuffer et al., 2013). As physical indicators may vary depending on the geographical location, it is important to take the city's background into account. Since Mumbai is located on a peninsula it offers only limited space, so housing densities are extremely high and therefore do not necessarily indicate slum areas (Kuffer et al., 2013). Another challenge of remote sensing based image analysis is that we can only derive physical features from the bird's eye perspective, but we are not able to perceive information about the social status of the underlying population. Consequently there might be people living in a community with a low social status but having many physical characteristics visually indicating a formal settlement (Graesser et al., 2012).

2.2 Principles of Radar Data and Polarization Modes

Unlike the human eye, remote sensing sensors are capable of detecting radiation outside of the visible portion (blue, green and red) of the electromagnetic spectrum. Radar data is information derived from a sensor that operates in the microwave portion of the electromagnetic spectrum with wavelengths ranging from less than 1 mm to 1 m (corresponding to frequencies of 300 GHz – 30 MHz). One advantage of this is that microwaves are able to penetrate interfering atmospheric particles like clouds, smoke, haze, fog or even light rain and snow (Lillesand et al., 2004). Therefore, Radar sensors are able to operate in almost all weather conditions with just few constraints, e.g. for wavelengths less than 4 cm (Lillesand et al., 2004). Atmospheric influences are present, but can be removed relatively simply. In combination with an inch-perfect flight path, a positional accuracy of the observed objects far below the spatial resolution is reached. Optical sensors typically provide easier interpretable data with high spatial resolution, but are bounded by daylight and cloudless vision, so acquisition cannot be precisely planned.

Radar is the acronym for "**RA**dio **D**etection **A**nd **R**anging" (Lillesand et al., 2004) and was developed by a collaboration of several countries during the Second World War by means of using radio waves (microwaves) to detect remote objects and their distance. Along with the improvement of nonmilitary sensing capabilities, Radar sensing became a widely-used tool for earth surface observations and is especially promising in regions having persistent cloud cover (Lillesand et al., 2004). Radar sensing belongs to the active remote sensing systems providing their own source of energy by transmitting long-wavelength microwaves (Lillesand et al., 2004). Therefore Radar sensors are independent from the sunlight and are able to operate during day and night. Radar systems send a pulse and measure the travel time of the reflected pulse, as well as the intensity of the backscattering signal (Taubenböck & Roth, 2010).

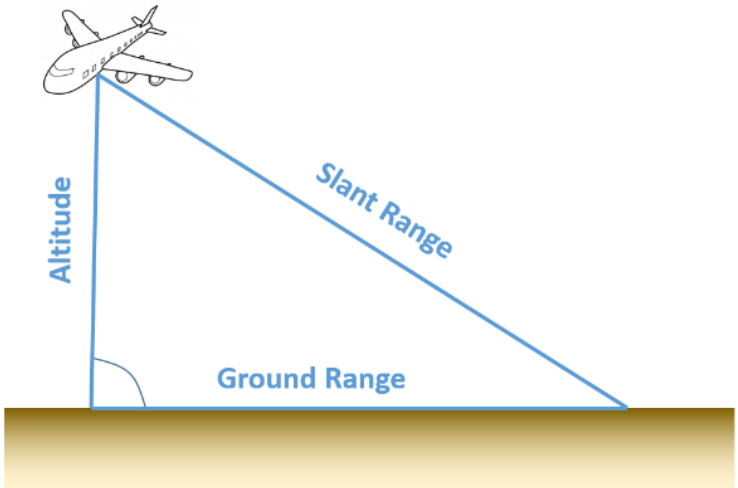


Figure 2-1: Slant range, ground range and altitude of an aircraft in relation to an object on the ground

Source: Own illustration after TeachEngineering (2015). The source of this material is the TeachEngineering digital library collection at www.TeachEngineering.org. All rights reserved.

By electronically measuring the travel time t that it takes for the signals to reach the ground and return to the antenna, the distance (slant range, not to be confused with the ground range, see Figure 2-1) of objects can be determined:

$$\overline{SR} = \frac{c \times t}{2} \quad (1)$$

with speed of light c ($3 \times 10^8\, {}^{m}/_{sec}$) (Lillesand et al., 2004).

2.2.1 Side-looking Radar and Synthetic Aperture Radar Systems

Imaging Radar systems carry an antenna that transmits short pulses of microwave energy in the direction of interest and records the strength and origin of the reflections received from the objects within the field of view, by consistently switching the antenna from transmitter to receiver mode using a synchronizer switch (Lillesand et al., 2004). In order to obtain a delay between the parts of the returned pulse that are the farthest away from the antenna and those parts that are closest, the antenna naturally needs to be pointed to the side. These systems, consisting of a fixed side-looking antenna below an aircraft, were the first operating Radar systems and are termed Side-Looking Radar (SLR) or Side-Looking Airborne Radar (SLAR) in case of airborne systems (see Figure 2-2) (Lillesand et al., 2004). The aircraft travels in a straight line that is called the azimuth flight direction. Conical pulses of microwave electromagnetic radiation illuminate strips of the terrain at right angles to the aircraft's direction of travel (sideways), which is called the range or look direction. Through side-looking, only the swath of the terrain beside the sensor is recorded (Albertz, 2009). SLAR systems impose a few restrictions: relatively short range and wavelengths, which leads to atmospheric attenuation and dispersion, limited resolution as well as low altitude restriction, which is linked to small coverage (Lillesand et al., 2004).

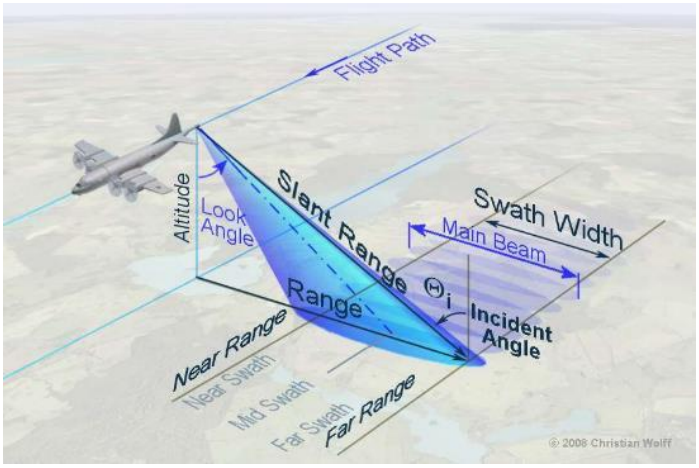


Figure 2-2: Principle of Side-looking airborne Radar imaging

Source: Wolff (2008)

As can be seen in Figure 2-2, the look angle of the sensing system describes the angle between nadir and look direction. Its complement is called depression angle. The incidence angle, or incident angle, is the angle between a normal to the earth's surface at the point of

incidence and the incident Radar beam (see Figure 2-2). Due to curvature of the earth, the incidence angle is slightly greater than the look angle for satellite based systems, whereas they are about equal for airborne systems (Lillesand et al., 2004).

Through side-looking, some complications for the interpretation of Radar images result. As long as the landscape is rather flat, these effects are almost imperceptible. However, in areas with greater relief variations, like mountainous or urban areas, mainly three relief displacement effects occur: Foreshortening, layover and shadow. Foreshortening is the shortened display of the sensor facing slopes. Layover describes the process of flipping the image, when the inclination of a surface relative to the sensor is too steep. Shadows are usually caused by steep faces or buildings. Some of these geometric effects can be overcome by satellite systems, since they are usually operating with lower incidence angles (Yang, 2011). Another geometric distortion is the slant-range scale distortion. Using the slant-range (see Figure 2-1) for spacing between the pixels, the image scale is compressed at near range and expanded at far range. This problem can be overcome as long as the flight height is known by applying a hyperbolic correction, in order to calculate a ground-range image. To measure the flight height, all systems usually contain a GPS and inertial navigation and control system.

Another special Radar effect is the so called speckle effect that is caused by interference of the backscattering waves through particles at the ground. The result is a strengthening or weakening of the intensity that is not predictable, but reproducible. Thus homogenous surfaces are not represented as homogenous but look rather grainy. Through multiple-look processing, several images are averaged in order to smoothen the image and reduce the speckle effect (the more images the less amount of speckle) (Lillesand et al., 2004). The number of images is also called number of looks. This is only one possible method of speckle reduction among many others, for example filters. For more information on these methods see (Soergel, 2010) or (Tso & Mather, 2009).

The ground resolution is mainly dependent on two parameters: The length of the pulse in a timely manner (range direction) and the antenna beam width (azimuth direction):

$$Range\ Resolution = \frac{c\tau}{2cos\theta} \tag{2}$$

$$Azimuth\ Resolution = \beta \times d \tag{3}$$

with speed of light c ($3 \times 10^8\ ^m/_{sec}$), pulse duration τ, depression angle θ, antenna beam width β and ground range d (Tso & Mather, 2009).

In order to distinguish two objects on the ground that are located close to each other in range direction, it is necessary that the two reflected signals are received separately (Lillesand et al., 2004). The beam width β again is contingent on the length of the antenna L (Tso & Mather, 2009),

$$\beta = \frac{\lambda}{L} \tag{4}$$

with λ being the wavelength of the transmitted pulse (Lillesand et al., 2004). So in order to achieve a fine resolution at long range with a so called Real Aperture Radar (RAR) system, a very narrow and strong pulse would be needed in order for remote areas to backscatter

measurably. Therefore the antenna would require a length of several kilometers which is simply not feasible with only one antenna (Richards & Jia, 2006).

Hence Synthetic Aperture Radar (SAR) was invented, which carries only a short physical antenna, but synthesizes a very long antenna by combining multiple recordings of the real antenna to a computationally generated longer antenna (Schmitt, 2011). This causes a stronger overlap of the Radar beams than with RAR. The SAR principle makes the spatial resolution independent of the distance to the observed object, so satellites are theoretically able to provide images in the same resolution as aircrafts, despite their much higher flying altitude. Points on the ground at near range are viewed by fewer antenna elements than those at far range, since the effective antenna length increases in range direction. Thus the azimuth resolution is constant in spite of range. The ground-range resolution, in contrast to the slant-range resolution, decreases with distance from the spacecraft (Lillesand et al., 2004). Large coverage, with partially reduced resolution, is also possible. Future antenna concepts promise great coverage with high spatial resolution (Schmitt, 2011).

Since every received pulse contains a mixture of backscattering of many objects, the Doppler Effect is used to separate the signals of the respective signals. The Doppler Effect is a frequency shift of a wave for an observer moving relative to its source. In case of Radar sensing, the sensor is moving relative to the reflector or the ground. As the sensor moves towards the signal source, the Doppler Effect is positive, the frequency is upshifted (higher); as the sensor moves away from the source, the effect is negative and the frequency is downshifted (lower). When the sensor is at the point of smallest distance to the signal source, there will be little to no frequency shift. This phenomenon is called Zero-Doppler and is taken as reference for all other observations. For remoter objects the approach is slower and therefore the Doppler Effect is less significant whereas for closer objects it is stronger. By processing the return signals according to their Doppler frequency shifts, a very small effective beamwidth can be generated.

2.2.2 SAR Satellite Systems

The first satellite having Radar sensing capability, called Seasat, was launched in 1978 (Lillesand et al., 2004) and was operated by NASA, USA. Since 1981 the Shuttle Imaging Radar (SIR) joined Seasat for experimental missions (Albertz, 2009). These two were quickly followed by ERS-1 (European Space Agency (ESA)), Almaz-1 (former Soviet Union) and JERS-1 (Japan) in 1991 and 1992, respectively. Radarsat's first satellite Radarsat-1 (Canadian Space Agency (CSA), Canada) was launched in 1995, see Figure 2-3. ERS and Radarsat were the first continuously observing Radar systems (Albertz, 2009). The batch of Radar satellites continued to grow with the global Shuttle Radar Topography Mission (SRTM) aiming to create a consistent, high-resolution digital terrain model. The SRTM is a joint project of NASA (SIR-C) and DLR (X-SAR). In 2002, the ESA launched another Radar satellite, called Envisat. The PALSAR system on ALOS-1 (Japan) followed in 2006. In 2007, three of the most current and advanced systems were launched, namely COSMO-SkyMed (Italy), TerraSAR-X (Germany) and Radarsat-2 (Canada), all of which are active today (Lillesand et al., 2004). Sentinel-1A (launched in 2014) and soon Sentinel-1B (planned launch in 2016) are and will be the newest, currently available Radar data sources operated by the ESA. Sentinel-1A has a

much higher revisiting time than Radarsat-2 with 12 days instead of 24. ALOS-2 (Japan) is the follower of ALOS-1 and was also launched in 2014.

The electromagnetic spectrum can be divided into several different spectral bands e.g. the microwave portion, which can be further divided into X-, C- and L-band. The letters, with which the bands were named, were chosen arbitrarily reaching back to the origins of Radar sensing in order to ensure military security (Lillesand et al., 2004). The currently available Radar satellites operate in the microwave bands X (COSMO-SkyMed, TerraSAR-X), C (Sentinel-1A, Radarsat-2) and L (ALOS-2). X-band operates in the wavelengths of approximately 2.4-3.75 cm (8-12.5 GHz), C-Band in 3.75–7.5 cm (4-8 GHz) and L-band in 15-30 cm (1-2 GHz) (Lillesand et al., 2004). The shorter wavelengths of the X band allow for a higher resolution of the imagery, but also represent the limit of all-weather mapping capability of Radar data, because the Rayleigh scattering is inversely proportional to the wavelength (Soergel, 2010).

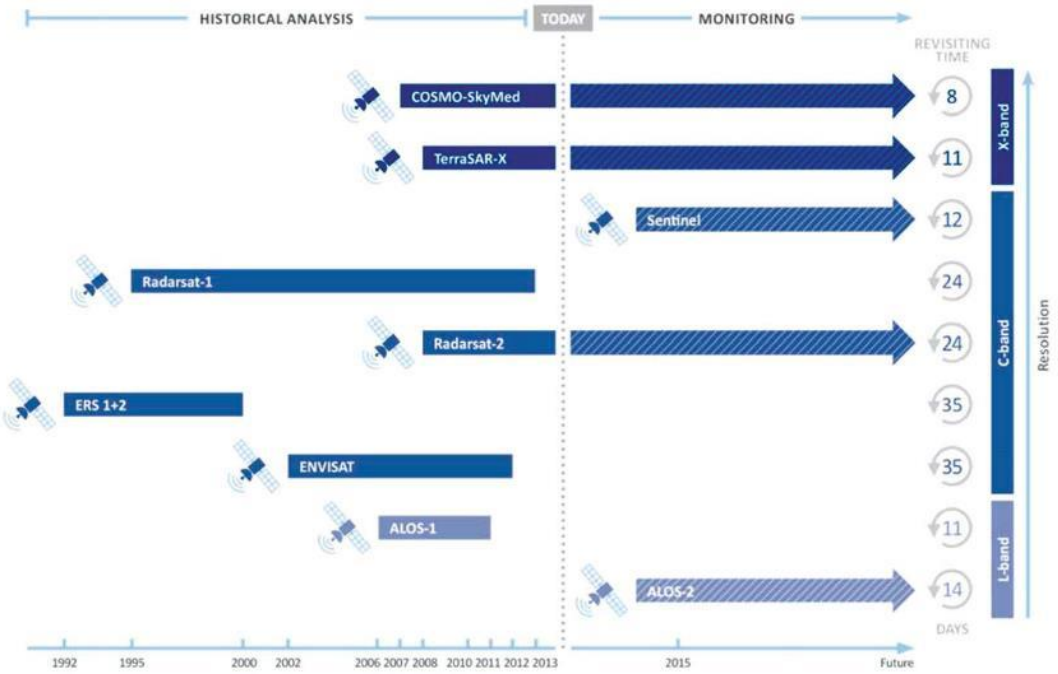


Figure 2-3: Current and historical SAR satellite constellations available

Since spaceborne Radar data was available, the applications of Radar sensing have increased considerably. Current spaceborne SAR systems reach resolutions of a few meters, in special imaging modes even less: TerraSAR-X and COSMO-SkyMed (VHR SAR sensors) allow for the best spatial ground resolution achieved, down to 1 meter in specific imaging modes (Brunner et al., 2010), and are particularly useful for imaging of small-scaled building structures. In this study, the Radarsat-2 satellite will be used, and its sensor specifications will be explained in more detail.

2.2.3 Radarsat-2

Radarsat-2 is an active earth observation satellite, operated by the Canadian Space Agency (CSA) in cooperation with the Canadian global communications and information company MacDonald, Dettwiler and Associates Ltd. (MDA). MDA owns and operates the satellite and ground station while the CSA is in charge of construction and launch of the satellite and regains its investments by supplying the Government of Canada with data throughout the mission. Radarsat-2 is the most advanced commercial C-Band SAR satellite currently available (Qi et al., 2012).

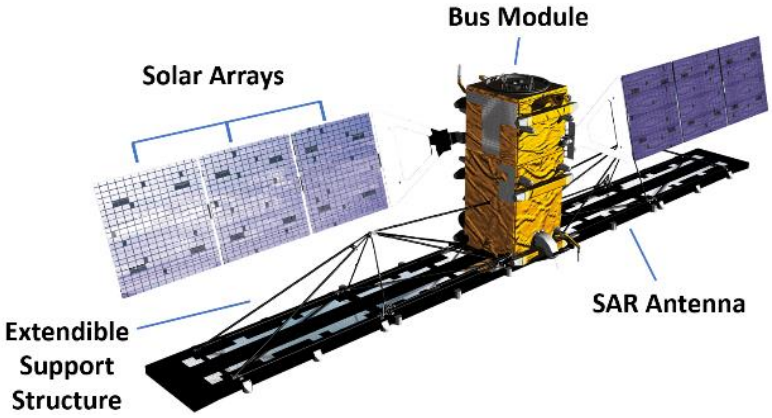


Figure 2-4: Components of the Radarsat-2 satellite

Source: Canadian Space Agency (2007)

Since its launch in 2007, the satellite circles the earth in a sun synchronous orbit with a flight height of 798 kilometers and an inclination of 98.6 degrees. Radarsat-2 is the follower of Radarsat-1 which ended its mission in 2013. The satellite carries a SAR sensor working in fully polarimetric mode acquiring data in HH, HV, VV and VH polarization at the same time. Each polarization mode is sensitive to different surface characteristics (Qi et al., 2012). The SAR antenna carried by the satellite has a size of 15 m x 1.5 m (see Figure 2-4), and is capable of right- and left looking operation (see Figure 2-5).

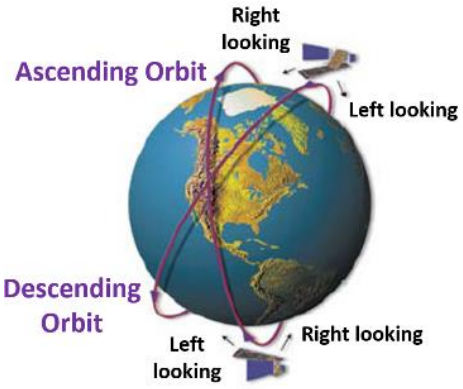


Figure 2-5: Orbit of the Radarsat-2 satellite

Source: Canadian Space Agency (2007)

Radarsat-2 offers 18 different beam modes in total, ranging from Spotlight, over Ultra-Fine, Wide Ultra-Fine, Extra-Fine, Multi-Look Fine, Wide Multi-Look Fine, Fine, Wide Fine, Fine Quad-Pol, Wide Fine Quad-Pol, Standard Quad-Pol, Wide Standard Quad-Pol, Standard, Extended Low, Wide, Extended High and ScanSAR Narrow to ScanSAR Wide (see Figure 2-6) (MacDonald Dettwiler and Associates Ltd., 2014b).

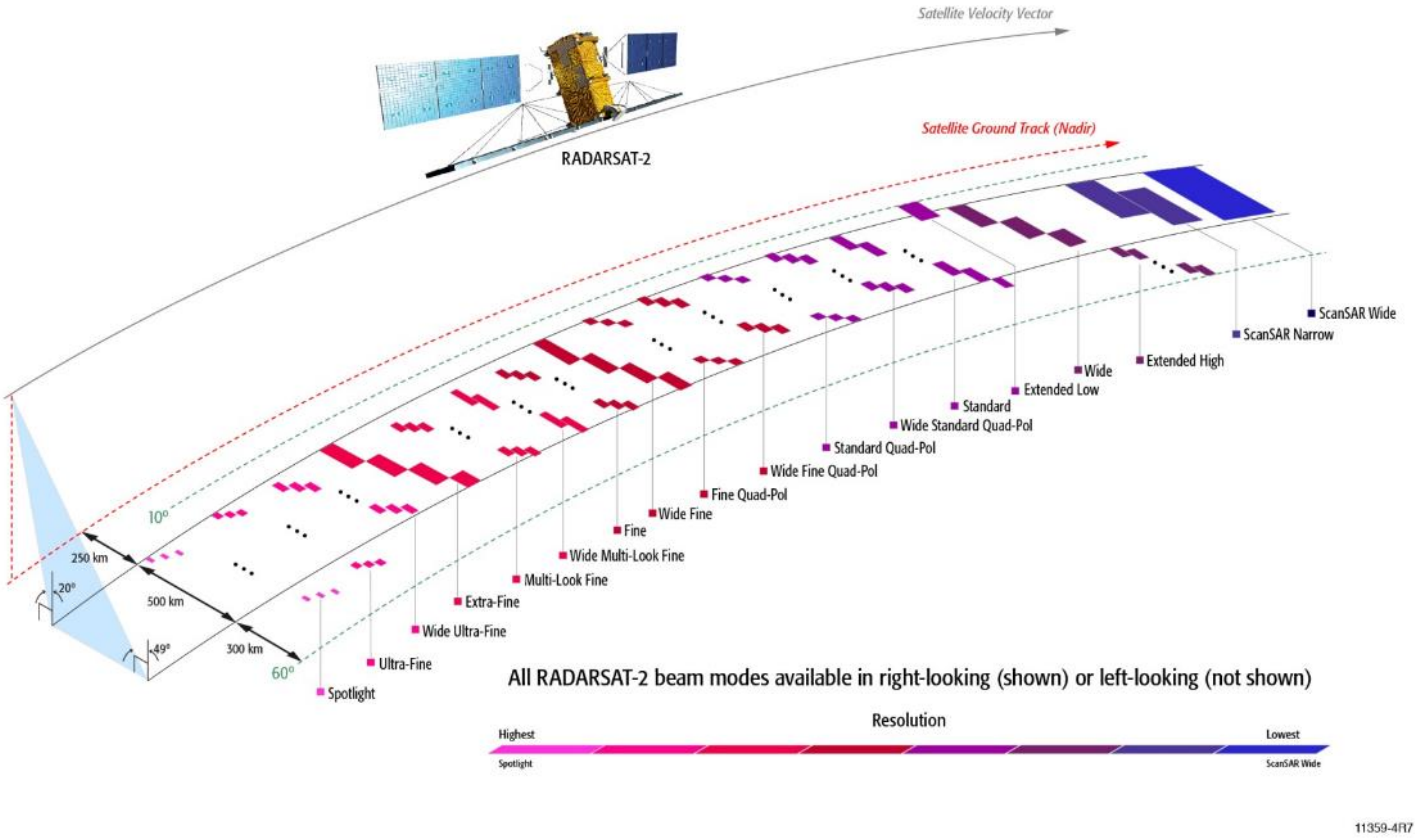


Figure 2-6: Radarsat-2 beam modes

Source: MacDonald Dettwiler and Associates Ltd. (2014b)

Depending on the beam mode, a variety of different resolutions and incidence angles can be acquired (see Figure 2-7). To date, an incidence angle greater than 50° is commercially not available. The highest accuracy achieved by the sensor is 3x1 meter in Spotlight mode and 3x3 meters in Ultra Fine Mode with a positional accuracy of 100 meters. The frequency of the radiated signal from the antenna is 5.405 GHz (C-Band), corresponding to a wavelength of 5.6 cm. The swath width, also depending on the beam mode, accounts up to 500 kilometers and the repeat cycle is 24 days. Acquiring more than 30,000 images a year, Radarsat-2 offers high-qualitative commercially available Radar images for users worldwide. Applications of Radarsat-2 data are manifold and comprise land use mapping, crop monitoring, sea ice mapping, flood mapping, vessel monitoring as well as disaster management and research (Canadian Space Agency, 2011). Radarsat-2 is capable of sending and receiving Radar waves in both, horizontal and vertical polarization. All combinations of horizontal (H) and vertical (V) polarization are available: HH, VV, HV and VH.

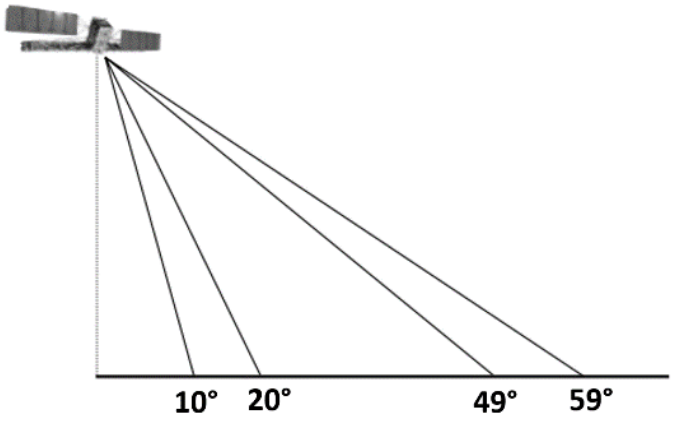


Figure 2-7: Range of incidence angles of the Radarsat-2 satellite

Source: MacDonald Dettwiler and Associates Ltd. (1999)

2.2.4 Interpretation of Radar Data

Radar systems, in comparison to optical remote sensing systems, do not measure the reflectance of the surface directly. Instead, they measure the intensity of the radiation that is backscattered by the surface, meaning the portion of the incident energy that is reflected directly back towards the sensor. However, microwave reflections do not have a direct relationship to those from the visible portion of the spectrum and provide a very different view on the earth. In general, the appearance of pixels in Radar images is affected by two groups of factors (Albertz, 2009; Purkis & Klemas, 2011):

- **Parameters of the sensor system:**
  - Wavelength
  - Polarization
  - Incidence angle
- **Parameters of the terrain surface:**

  a) Surface characteristics

  - Surface roughness
  - Geometric structure
  - Orientation

  b) Electrical characteristics

  - Moisture content
  - Conductivity
  - Dielectric constant

For example, for an incidence angle of 0°-30° topographic slope dominates, for 30°-70° surface roughness and for more than 70° Radar shadows dominate (Lillesand et al., 2004). Slopes facing the sensor usually appear brighter than slopes facing away from the sensor. Surfaces that might appear “rough” in the visible portion might be seen “smooth” in the microwave

portion (Lillesand et al., 2004). Very smooth surfaces act as a specular reflector (see Figure 2-8), like water, whereas rough surfaces act as diffuse reflector (see Figure 2-9), like forest. Specular reflectors appear very dark in Radar images, no matter if its reflectance is very low or very high (Lillesand et al., 2004). Often, hybrid forms of specular and diffuse reflectors appear (Albertz, 2009). However, the Radar reflectivity of various surfaces under various conditions is very complex and not in all cases known precisely and therefore still under investigation through empirical observations (Albertz, 2009; Lillesand et al., 2004; Schmitt, 2011).

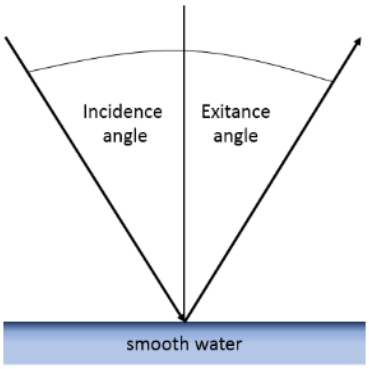


Figure 2-8: Specular reflection

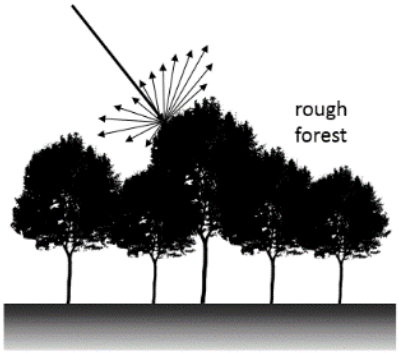


Figure 2-9: Diffuse reflection

Source: Own illustration after Lillesand et al. (2004)

In the urban environment, SAR data helps to detect manmade objects, since those usually have very high backscatter values and appear very bright in the image. This is caused by so called corner reflections, also called multi-bounce or double-bounce scattering (see Figure 2-10) (Purkis & Klemas, 2011). Buildings having complex metal structures or corners usually return a strong Radar echo. This effect can also be used for marking pass points in the landscape by mounting corner reflectors out of metal (Albertz, 2009). Flat rooftops of large industry buildings, however, can easily be detected since they usually have lower backscatter values (Weydahl et al., 2005).

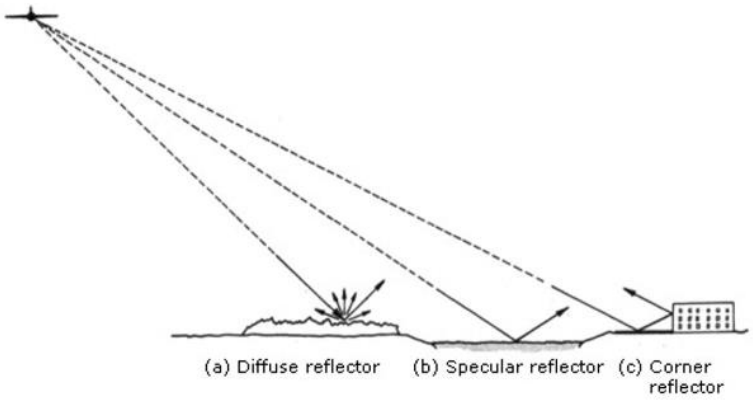


Figure 2-10: Radar reflection from diffuse (a), specular (b) and corner (c) reflectors

Source: Lillesand et al. (2004)

Urban areas experience a higher heterogeneity due to the strong reflections of buildings (Soergel, 2010).There are some effects that pose difficulties when classifying urban areas. One of them is the so called cardinal direction effect. If the city or street pattern is laid out to the cardinal directions of the compass and perpendicular to the Radar beam, very strong backscatter returns. Another effect occurs, when the backscatter of a very bright object is high enough to exceed the range of the Radar system. This effect can be evidenced by bright cross-shaped formations radiating from the center of the bright object e.g. building (Lillesand et al., 2004). In general, urban environments pose various challenges, since many different materials e.g. cement, metal, glass or wood are used and various land cover types like roads, swimming pools, parks etc. need to be included in one class, so the standard deviation of this class is typically very high and the accuracy of the classification is likely to be rather small (Purkis & Klemas, 2011).

Despite the needs to learn more about how Radar data can be interpreted, various successful studies have been conducted, showing the usefulness of this kind of data. Especially in regions, where cloud cover limits the effectiveness of optical remote sensing data, Radar data offers an alternative to optical RS systems (Lillesand et al., 2004). Another great advantage of SAR remote sensing is the flexibility of temporal coverage. Radar satellites are situated on near pole orbits, so that they are able to cover the same area from the same angle within only a few days. This is imperative for change detection applications. From time to time the aiming beam or even the whole satellite can pivot and record the same area from a different angle. SAR data also offer a big advantage in time critical events like natural hazards or earthquakes, when current data is needed promptly (Soergel, 2010). For example, SAR data was used to map out the damages caused by the 2011 Tohoku earthquake in Japan (Park et al., 2013).

Radar images were in the focus of research for years but only a few methods have managed to be used in the practical sense. One reason for this is likely the need for automation of the whole process, from data collection and information extraction to the delivery of a clearly defined product (Schmitt, 2011).

2.2.5 Mathematical Background of Radar Waves and Polarization

A Radar pulse of electromagnetic radiation can be displayed as wave pattern (see Figure 2-11).

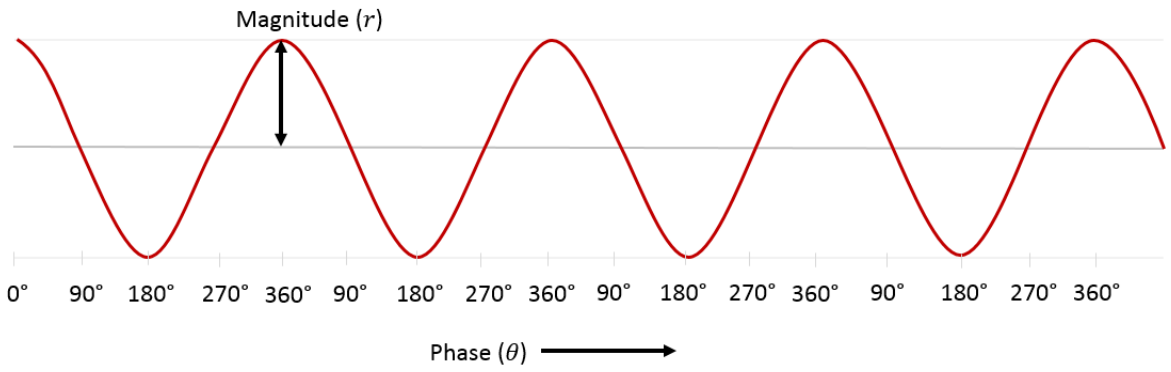


Figure 2-11: Radar wave with magnitude r and phase θ

Source: Lillesand et al. (2004)

The height of the wave is referred to as magnitude, whereas the position of the wave is given by the phase, ranging from 0° to 360°. Therefore Radar waves can be defined by the equation

$$I = r \cos \theta \tag{5}$$

with transmitted signal I, magnitude r and phase θ.

The Radar signal received by the antenna is multiplied with two reference signals, one in phase with the transmitted signal (real component I, see equation above) and one shifted by 90° to the transmitted signal (imaginary component Q) that can be calculated by:

$$Q = r \sin \theta \tag{6}$$

The amplitude A of the Radar signal is then calculated by

$$A = I + jQ \tag{7}$$

with $j = \sqrt{-1}$. The intensity of the Radar signal can be calculated by

$$i = I^2 + Q^2 \tag{8}$$

(Soergel, 2010).

SAR images are comprised of complex numbers (see Equation (7)), and therefore out of two components: real (I) and imaginary part (Q). By looking at both components on their own, little structure can be seen and objects are not detectable. By looking at the absolute value or amplitude of the complex number, the strength of backscattering can be revealed. The phase information can generally not be interpreted for dual-polarized imagery. This information comes into play, if the difference of the phase of several images can be calculated e.g. by SAR interferometry. A consistent mathematical framework used to represent SAR data in real number format is the Kennaugh matrix, which will be introduced in the next chapter.

Polarization is a special characteristic of Radar waves. It describes the geometric plane, in which the Radar wave is oscillating (see Figure 2-12). Radar sensors send and receive only fully coherent (amplitude and phase) and polarized radiation. The detection of incoherent (meaning the phase differences are random or unknown, like for example the sun's illumination (Purkis & Klemas, 2011)) and unpolarized parts is technically impossible. Although every polarization would be possible, nowadays operational systems usually use linearly polarized radiation in either horizontal (H) or vertical (V) direction.

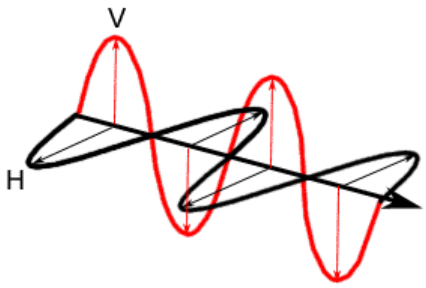


Figure 2-12: Horizontally and vertically polarized wave

Source: Natural Resources Canada (2013)

The goal of polarimetric Radar remote sensing is not to describe the polarized wave itself, but to compare the transmitted wave with the received and draw conclusions about the shape of the earth's surface from the change of the wave on its way to the surface and back. The antenna can be set to receive only a certain type of polarization e.g. horizontally or vertically. Thus the combinations of co-polarized signals HH and VV, meaning that the polarization state of sent and received waves are the same, and cross-polarized signals HV and VH, meaning that the state of the Radar wave during transmission is orthogonal to the state during reception can be used, whereby the first letter stands for the transmitted polarization and the second for the received polarization.

To further increase the information content, often multi-polarized (measuring more than one polarization) SAR images are used. Dual-polarized images for example consist of two images: one co-polarized (HH or VV) and one cross polarized (HV and VH) (Lillesand et al., 2004). The typical cross-pol configuration for Radarsat-imagery is HH/HV, meaning alternating transmitting horizontally and receiving both, horizontally and vertically, and VV/VH, transmitting vertically and receiving both, vertically and horizontally (Schmitt et al., 2015).

Fully polarimetric or quad-polarized images consist of four images acquired at the same time: Two co-polarized images (HH and VV) and two cross-polarized images (HV and VH). Quad-polarization offers the advantage of being capable of polarimetric analysis. Polarimetric analysis is a method that allows extracting detailed information about the structure and shape of the surface (MacDonald Dettwiler and Associates Ltd., 2014a).

2.2.6 Kennaugh Matrix

In this work, the Kennaugh element framework based on Schmitt (2011) was used for representation of the Radar data. The Kennaugh element framework is a mathematical framework that can be used to consistently represent SAR imagery acquired in different acquisition modes and therefore having different geometry, radiometry and also polarimetry (Schmitt et al., 2015).

Since nearly any decomposition also means a loss of polarimetric information, a pure scattering description was used in form of a "Kennaugh matrix" for data preparation and delivery. The decomposition into Kennaugh matrix elements ensures a consistent mathematical description independent of the polarization. Various mathematical properties of this matrix facilitate processing and conversion into integer image formats. The Kennaugh Matrix is a symmetrical 4x4 matrix containing real elements in the Back Scattering Alignment (BSA) coordinate system (see Equation (9)).

$$\begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix}_{received} = \begin{bmatrix} K_0 & K_4 & K_5 & K_6 \\ K_4 & K_1 & K_9 & K_8 \\ K_5 & K_9 & K_2 & K_7 \\ K_6 & K_8 & K_7 & K_3 \end{bmatrix} \times \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix}_{transmitted} \tag{9}$$

The BSA coordinate system is defined from the viewpoint of the antenna in direction to the target and is generally used for Radar, while the Forward Scattering Alignment (FSA) coordinate system is tied to the viewpoint of the wave and is usually used in optics. During the transition to the FSA coordinate system only a few signs of the Kennaugh matrix would change, affecting the symmetry (Schmitt, 2011). The resulting matrix of the transition is called Müller matrix:

$$M = \begin{bmatrix} K_0 & K_4 & K_5 & K_6 \\ K_4 & K_1 & K_9 & K_8 \\ -K_5 & -K_9 & K_2 & K_7 \\ K_6 & K_8 & -K_7 & K_3 \end{bmatrix} \tag{10}$$

The Müller matrix, as well as the Kennaugh matrix and Jones matrix are all transformation matrices, transforming the Stokes Vector and describing the changes of the polarization state of the electromagnetic wave from backscattering. In comparison to the Müller matrix, the Kennaugh matrix can also be used for multi-polarized SAR imagery. The numeration of the Kennaugh elements was chosen freely to facilitate their interpretation.

The Kennaugh Matrix converts the Stokes Vector of the transmitted wave in the Stokes Vector of the received wave (see Equation (9)), why it is sometimes also referred to as Stokes Matrix. The Stokes Vector is exclusively based on real intensities, not on phase differences. The first element g_0 describes the total intensity of the wave independent of its polarization, g_1 contains the difference between horizontal and vertical polarized radiation, g_2 is the difference between diagonal and counter diagonal polarized radiation and g_3 is the difference between right and left-circular polarized radiation (Schmitt, 2011).

The polarization optics distinguishes between three groups of Kennaugh elements: Absorption, Diattenuation and Retardance (see Table 2-1). The coordinate systems are oriented parallel

to the linear axes, either axially parallel, diagonal or circular. Absorption (K_1, K_2, K_3) describes the loss of polarization during the scattering process, Diattenuation (K_4, K_5, K_6) reveals the amplitude modifications during scattering and Retardance (K_7, K_8, K_9) means the phase delay in the scattering process (Schmitt et al., 2015). K_0 contains the overall intensity and is orientation independent, K_1 for example indicates how much of the axially parallel intensity difference is again reflected back parallel to the axis and K_8 shows the relation of the circular and axially parallel intensity differences (Schmitt, 2011).

Table 2-1: Groups of individual elements of the Kennaugh Matrix

	Totally	Axially parallel	Diagonal	Circular
Absorption	K_0	K_1	K_2	K_3
Diattenuation		K_4	K_5	K_6
Retardance		K_7	K_8	K_9

For separation of backscatter intensity and polarimetric information, the Kennaugh matrix needs to be normalized on the total intensity, which is given in K_0 (see Equation (11)).

$$K = \begin{bmatrix} K_0 & K_4 & K_5 & K_6 \\ K_4 & K_1 & K_9 & K_8 \\ K_5 & K_9 & K_2 & K_7 \\ K_6 & K_8 & K_7 & K_3 \end{bmatrix} = I_{total} \times \begin{bmatrix} 1 & k_4 & k_5 & k_6 \\ k_4 & k_1 & k_9 & k_8 \\ k_5 & k_9 & k_2 & k_7 \\ k_6 & k_8 & k_7 & k_3 \end{bmatrix} \quad (11)$$

Thus all elements are projected to the range between -1 and 1. This is a big advantage over the previous presentations, because the entire polarimetric information can be represented on the same set of real numbers. In addition, the individual elements can be visualized easily and without loss of information.

The Kennaugh matrix offers several advantages over other representations: It uses exclusively real elements (every other matrix uses partially or completely complex numbers), it is interpretable for coherent as well as incoherent radiation, it keeps its full dimension in co-polarized (HH and VV) as well as in cross-polarized (HH and HV or VV and VH) case (some elements are set to null so that these combinations remain: For the co-polarized case the number of independent Kennaugh elements reduces to K_0, K_3, K_4 and K_7 and for the cross-polarized case the independent Kennaugh elements reduces to K_0, K_1, K_5 and K_8. Additionally it is twin-pole capable and flexible concerning the further processing.

As mentioned before in Chapter 2.2, in case of dual-polarized imagery, the phase does not hold any useful information, so the intensities of both input channels are the only source of information, e.g. $|S_{HH}|^2$ and $|S_{HV}|^2$ for HH/HV polarization. Thus following Kennaugh elements can be derived for the data used in this work:

$$K_0 = |S_{HH}|^2 + |S_{HV}|^2 \quad (12)$$

$$K_1 = |S_{HH}|^2 - |S_{HV}|^2 \quad (13)$$

$$K_5 = Re\{S_{HH}S_{HV}^*\} \quad (14)$$

$$K_8 = Im\{S_{HH}S_{HV}^*\} \quad (15)$$

Whereby K_0 and K_1 are only intensity-based and K_5 and K_8 contain the real and imaginary part of the inter-channel correlation. K_0 represents the total intensity and corresponds to the brightness of the SAR imagery. K_1 reveals the difference between co- and cross-polarized intensity and differentiates between clearly defined targets such as faces or edges on one side and diffuse dipoles, e.g. forest on the other side. K_5 and K_8 are only useful for detecting deterministic (manmade) features.

For VV/VH polarization basically the same equations apply:

$$K_0 = |S_{VV}|^2 + |S_{VH}|^2 \quad (16)$$

$$K_1 = |S_{VV}|^2 - |S_{VH}|^2 \quad (17)$$

$$K_5 = Re\{S_{VV}S_{VH}^*\} \quad (18)$$

$$K_8 = Im\{S_{VV}S_{VH}^*\} \quad (19)$$

The Kennaugh-elements are able to transfer the complex products of SAR sensors with a real and an imaginary part to real number data sets. These describe different aspects of the differences between the sent and received polarization levels. By combination of diverse Kennaugh-elements, additional products are available (Schmitt, 2011; Schmitt et al., 2015):

- **Absorption**: Analogous to the intensities over all orientations

$$Absorption = |K_1| \quad (20)$$

- **Diattenuation**: Description of the amplitude modifications during scattering

$$Diattenuation = |K_5| \quad (21)$$

- **Polarizance:** Polarimetric information contribution of K_1, K_5 and K_8

$$Polarizance = \sqrt{\frac{K_1^2 + 2K_5^2 + 2K_8^2}{3}} \quad (22)$$

- **Retardance**: Indicator of the phase delay resulting from scattering

$$Retardance = |K_8| \quad (23)$$

Additional products can also be derived from metadata, geocoding and multi-scale multi-looking:

- **Scale:** Non coherent scattering from multi-looking calculations
- **Gim**: Contains the incidence angle, derived from geocoding
- **Looks:** Look numbers (scales of the original image) from multi-scale multi-looking
- **Noise:** Derived from the metadata of acquisition

For classification, Absorption, Diattenuation, Polarizance, Retardance and Scale were used. In the following, these products will be referred to as deduced Kennaugh-elements.

2.3 Fundamentals of Texture Analysis and Gray Level Co-Occurrence Matrix

Various characteristics or features can be used in order to classify an image and to identify specific objects. In this context, features mean inputs to the classifier that separate objects into groups. Humans naturally use spectral, textural and/or contextual features for object detection. The same characteristics can be used in image analysis. Spectral features represent the tonal variations in different bands, textural features describe the spatial (statistical) distribution of tonal variations within a band, whereas contextual features comprise information about the surrounding area. Texture is along with spectral tone one of the most important characteristics in identifying areas of interest in images by computer algorithms (Haralick et al., 1973). Texture can vary from being fine, coarse, smooth, rippled, molled, irregular or lineated (Haralick et al., 1973). Due to the high spectral variability of urban features, texture measures are often favored and have been increasingly used in the urban field over the past few years (Baud et al., 2010; Herold et al., 2005; Kuffer et al., 2013). Since slums share a rather homogeneous texture in comparison to other urban areas, texture is also a very useful tool in detecting informal settlements.

In order to find texture patterns in RS imagery, different measures can be used. The most widely used measures in settlement classification are the Gray Level Co-Occurrence Matrix (GLCM), Semivariogramm (Carr & De Miranda, 1998; Chamundeeswari et al., 2009; Dekker, 2003; Gamba et al., 2006) and specially Moran's I in the urban environment (Stasolla & Gamba, 2008). The Semivariogramm is a measure of the spatial properties of an image in a specified direction (Richards & Jia, 2006). Moran's I measures how dispersed, uniformly distributed or clustered the texture pattern is (Purkis & Klemas, 2011).

The GLCM, also called gray-tone spatial-dependency matrix, is one of the most well-known and long-standing texture features (Graesser et al., 2012; Richards & Jia, 2006). It was first introduced by Haralick et al. (1973) and therefore texture features derived from the matrix are also referred to as Haralick features, e.g. Contrast or Homogeneity.

Each resolution cell of an image, excluding those on the periphery, is assumed to have eight nearest-neighbor resolution cells (see Figure 2-13).

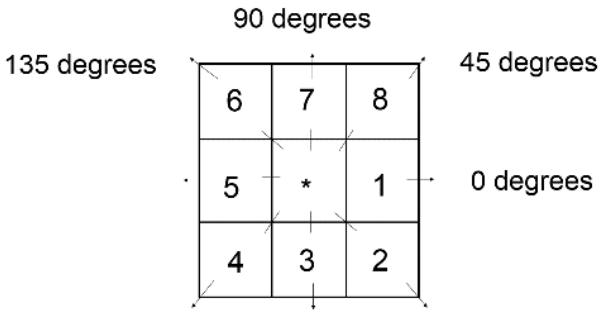


Figure 2-13: Resolution cell * and its nearest neighbors

Source: Haralick et al. (1973)

Resolution cells numbered 1 and 5 in Figure 2-13 are the horizontal nearest neighbors to resolution cell * in the middle at 0°. Cells 2 and 6 are 135° (left diagonal), 3 and 7 90° (vertical) and 4 and 8 45° (right diagonal) nearest neighbors. GLCMs are a function of the angular relationship and the distance between the neighboring cells (Haralick et al., 1973). In eCognition, GLCMs can be calculated in either 0°, 45°, 90°, 135° direction or in “all directions”, comprising the mean values of the directions 0°, 45°, 90° and 135° (Trimble Germany GmbH, 2014). In this study “all directions” was chosen, assuming that the texture of informal and formal settlements does not vary significantly with orientation.

A GLCM is a statistical feature and contains frequency values of occurrence of a pair of gray tone cells within one band of an image, separated by a certain distance and in a certain direction (Kohli et al., 2013). The dimensions of the GLCM depend on the number of possible gray levels in the image, e.g. 1024 for 10 bit data (Richards & Jia, 2006). For example to detect a texture pattern in a horizontal direction, the number of pixels of the particular gray levels in this direction in a certain distance is counted. By looking for the same pattern in the other directions e.g. vertically or diagonally, four matrices would be received for the chosen angle and distance. An example is given in Figure 2-14 (Richards & Jia, 2006; Tso & Mather, 2009).

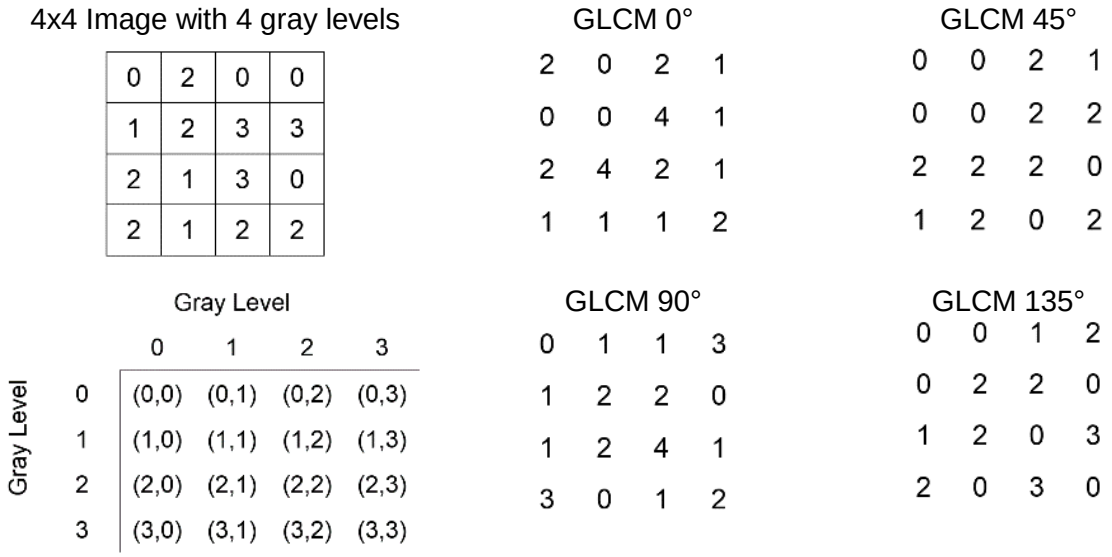


Figure 2-14: 4x4 example image (upper left), general form of the GLCM for an image with four gray levels (lower left) and calculated GLCMs (on the right) for the example image with distance d = 1 and directions 0°, 45°, 90° and 135°

Source: Tso & Mather (2009)

GLCM measures are usually calculated for certain regions, e.g. a user-specified moving window (kernel based approach) or predefined spatial objects (object based approach) (Lillesand et al., 2004). In this study a prescribed shapefile containing objects of all blocks of buildings in Mumbai was used. For more detailed information about the auxiliary data, see Chapter 3.2.2. In order to avoid scaling effects, the GLCM is usually normalized by dividing each cell by the total number of pairs or sum of all elements of the matrix, respectively (Tso & Mather, 2009).

Haralick et al. (1973) suggest a number of textural features that can be derived from GLCMs, e.g. Inverse Difference Moment, Entropy or Maximal Correlation Coefficient. In this study, the features that can be easily calculated in eCognition, Angular Second Moment (ASM), Contrast, Correlation, Homogeneity, Dissimilarity, Entropy, Mean and Standard Deviation, are used (Trimble Germany GmbH, 2014). These textural features are calculated from the GLCMs in order to find out if the texture is rather coarse, meaning it has large homogeneous patches, or if it is fine and small-scaled. If there are large patches of similar or equal gray values in the image, the neighboring pairs also tend to have similar values. Therefore the GLCM reveals higher values in its principle diagonal. On the other hand, if the image has a rather noisy and small-scaled texture pattern and the distance equals about the scale of the texture structure, neighboring pairs tend to have very different values. As a conclusion, the GLCM comprises higher values elsewhere than in its principal diagonal (Tso & Mather, 2009). In order to quantify this information derived from the GLCM, textural features are used. The textural features used in this study are the following:

Angular Second Moment (ASM) is calculated by the sum of squares of the relative frequencies $p(i,j)$ with which two neighboring resolution cells separated by distance d occur in the image, one with gray tone i and the other with gray tone j (see Equation (24)) (Haralick et al., 1973). The relative frequency $p(i,j)$ is calculated by the number of times each gray level pair is detected divided by the total number of possible gray level pairs (Richards & Jia, 2006). ASM is a measure of homogeneity of the image. Following, the more homogeneous the image, the lower are the ASM values. Homogeneous means the image tends to be smoother and reveals a rather small-scaled texture (Tso & Mather, 2009).

$$f_{ASM} = \sum_i \sum_j \{p(i,j)\}^2 \tag{24}$$

Contrast, however, is a measure of the amount of local variations within the image. The higher its value, the more contrast the image has (see Equation (25) (Haralick et al., 1973)). In terms of GLCM, this measure assigns a greater weight to the gray value pairs that are more distant from the principal diagonal, since those pairs reveal a greater difference and therefore show a higher contrast of the image,

$$f_{Contrast} = \sum_{n=0}^{N_{g-1}} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ |i-j|=n}}^{N_g} p(i,j) \right\} \tag{25}$$

with N_g being the gray value range of the original image or dimension of the GLCM.

Correlation is a measure of gray tone linear dependencies in the image (see Equation (26) (Haralick et al., 1973)),

$$f_{Correlation} = \frac{\sum_i \sum_j (ij) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \tag{26}$$

where σ and μ are the mean and standard deviation respectively.

Homogeneity takes higher values, if the gray level pairs reveal smaller differences. Regarding the GLCM, Homogeneity is more sensitive to the principal diagonal (Pacifici et al., 2009). So as weight decreases away from the diagonal, the output will be greater for areas having little contrast. Homogeneity weights values by the inverse of Contrast or Dissimilarity (see Equation (27) (Zheng et al., 2004)).

$$f_{Homogeneity} = \sum_i \sum_j \frac{p(i,j)}{|i-j|} \tag{27}$$

As opposed to Contrast, where the weights increase exponentially moving away from the diagonal, for **Dissimilarity** they increase linearly. This measure reveals high values when the local region has a high contrast (Pacifici et al., 2009). Since Homogeneity and Dissimilarity are inversely correlated with each other, Dissimilarity was later excluded from this study.

$$f_{Dissimilarity} = \sum_i \sum_j p(i,j)\,|i-j| \tag{28}$$

Entropy is the measure of orderliness within a band of an image and relates to textural homogeneity (see Equation (29) (Haralick et al., 1973)). Entropy returns high values if all entries of the GLCM are about equal and the image is not highly textured and low values if the entries vary greatly and the image is highly textured (Richards & Jia, 2006).

$$f_{Entropy} = -\sum_i \sum_j p(i,j)\,\log\big(p(i,j)\big) \tag{29}$$

The **Mean** can be calculated using either i or j, since the GLCM is symmetrical:

$$f_{Mean} = \sum_i \sum_j i\big(p(i,j)\big) = \sum_i \sum_j j\big(p(i,j)\big) \tag{30}$$

Standard Deviation (see Equation (31)) is very similar to Variance (see Equation (32)). The only difference is that they have a different value range. Variance and Standard Deviation are measures of dispersion of the values around the mean, similar to Entropy.

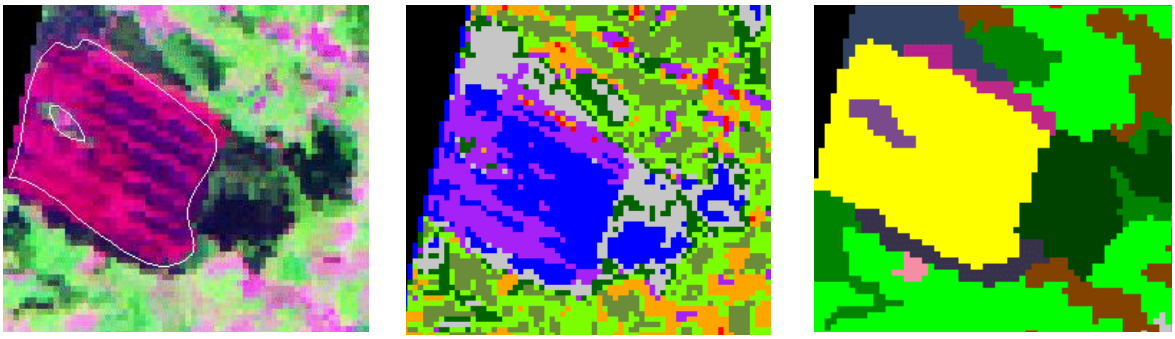
$$f_{Standard\ Deviation} = \sqrt{\sigma_i^2} = \sqrt{\sigma_j^2} = \sigma_i = \sigma_j \tag{31}$$

$$f_{Variance} = \sum_i \sum_j (i-\mu)^2 p(i,j) = \sigma_i^2 = \sigma_j^2 \tag{32}$$

Some of these features are strongly correlated with each other. Therefore it is recommended to apply a feature selection procedure to select a subset or linear combinations of the 28 suggested features in total by Haralick et al. (1973) (see Chapter 4.2.2). Some of these features have been employed in several studies separating slums from formal built-up areas. Kuffer et al. (2013) for example used the variance of the GLCM as texture measure and found that slums tend to have lower variance values compared to formal built-up areas.

2.4 Principles of Object-based Image Analysis

VHR imagery providing data with up to 1 meter resolution and the spectral diversity of urban areas both drove the development of an object-based classification approach (Jensen, 2005). In contrast to pixel-based methods based on single pixels, object-based image analysis (OBIA), also referred to as geographic object-based image analysis (GEOBIA), is based on a set of similar pixels, called objects. Pixels are grouped to image objects based on measures like spectral properties (e.g. color), size, shape, context from the neighboring pixels or texture (Lang et al., 2006), according to Chapter 2.3. One advantage of this is that the so called “salt and pepper effect” of pixel-based image analysis can be overcome. Figure 2-15 shows an example of an image- and pixel-based classification in comparison with an object-based classification of a bog in Southern Germany (Lang et al., 2006).



Original image | Pixel-based image analysis | Object-based image analysis

Figure 2-15: Pixel-based vs. object-based image analysis

Source: Lang et al. (2006)

The idea of OBIA is basically, that the original image consists of larger patches of pixels belonging to the same class (Lillesand et al., 2004). Especially in VHR and Radar remote sensing, where images are highly textured, OBIA plays a great role since it helps to aggregate pixels. OBIA is a relatively new technique used in digital image analysis (Blaschke, 2010). Through increasing spatial resolution and the advent of VHR imagery, OBIA methods are capturing more and more attention since 2000. The most popular software used for OBIA include *Definiens eCognition*, *ENVI Feature Extraction Module* and the *Feature Analyst extension of ArcGIS and ERDAS Imagine*. *eCognition* was the first software capable of OBIA and is still the most popular one (Soergel, 2010). Successful studies using *eCognition* have been many and varied (Blaschke, 2010; Esch et al., 2006; Hofmann, 2001b; Shekhar, 2012).

OBIA is comprised of two main steps: Image segmentation and image classification. Image segmentation is the process of grouping homogeneous pixels into meaningful image objects based on spectral or textural properties (Lillesand et al., 2004; Yang, 2011). In Figure 2-16 an example of a result of image segmentation can be seen. The scale of the image segmentation process depends on the sensor used and the size of features that need to be detected (Lillesand et al., 2004). Segmentation at a coarse scale for example could reveal forests, while at a fine scale, single trees could be identified. The roots of image segmentation lead back to

industrial image processing in the 1970s. Since the 1980s, this method has also been increasingly used in remote sensing applications. The aim of image segmentation is to divide the whole image into non-overlapping, homogeneous, delimitative regions (Blaschke & Lang, 2006). Image segmentation can additionally take into account spatial data layers like roads next to the image data (Jensen, 2005).

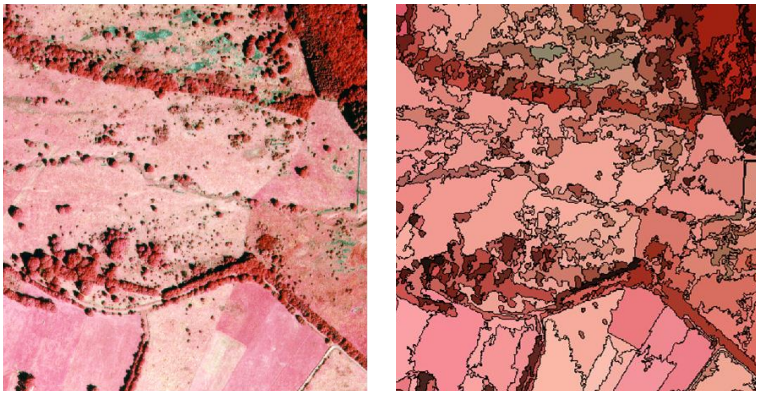


Figure 2-16: Original image (left) and segmentation (right)

Source: Lang et al. (2006)

The second step of an OBIA is the classification of the image objects. Classification means the labeling of objects or the assignment of objects to a certain number of classes, for example formal and informal settlement, respectively. There are two different types of classification in OBIA: Sample-based classification and rule-based classification. Sample-based classification uses a training sample to decide which class the new object belongs to whereas rule-based classification uses prior external knowledge in the form of rules to decide the class membership. In this work, sample-based classification algorithms have been used that are described in more detail in the following Chapter 2.5.

2.5 Classification Algorithms

The classification algorithms used in this elaboration are Linear Discriminant Analysis (LDA), Random Forest (RF) and Support Vector Machines (SVM), which are all machine learning concepts. Machine learning involves all algorithms that are trained by and make predictions on data. These algorithms build a model by learning from the data in order to make decisions or predictions based on the input data.

All three classification algorithms also belong to the subclass of supervised learning algorithms, meaning they are all trained by labeled training examples. A supervised learning algorithm uses the training data to infer a function that is used to predict the class labels of new examples. In unsupervised learning, on the other hand, unlabeled data is used as input, so the aim of an unsupervised algorithm is to recognize patterns in the input data.

2.5.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) in its original form was first introduced by Ronald A. Fisher in 1936, who attempted to identify species of Iris plants by using linear functions (Fisher, 1936). Thereafter, many variations were further developed, like logistic or squared discriminant analysis. But still, the original linear approach is widely-used, due to its simplicity and good performance (Decker et al., 2010). Squared discriminant analysis is supposed to be universally applicable, but on the other hand, needs double computing effort and does not always reveal better results than LDA. Discriminant analysis is a multivariate statistical method, like SVM and RF, meaning two or more characteristic variables are utilized to investigate the membership of objects to groups. It is widely-used in a variety of research fields like social sciences, biology, marketing and remote sensing, among others (Decker et al., 2010).

LDA can be used to predict the group or class membership, so objects with unknown class membership are predicted using characteristic variables and derived discriminant functions (separating functions), to find the group that has highest probability of membership. An advantage of the usage of LDA is, that the variables that contribute most to the prediction of group memberships can be revealed and thereafter the dimension of input variables can be reduced. The number of discriminant functions used to resolve the problem is dependent on the number of groups, the objects need to be split into. For solving a two-class-problem for example, only one discriminant function is needed. The discriminant function sums up the weighted characteristic variables. The weight of the characteristic variables, also called discriminant coefficients, are chosen in such a way, that the discriminant function reaches the best delineation capacity of groups. In a two-dimensional feature space, the discriminant function can be visualized as straight line (see Figure 2-17), whereas in a higher dimensional feature space it would be displayed as a discriminant plane.

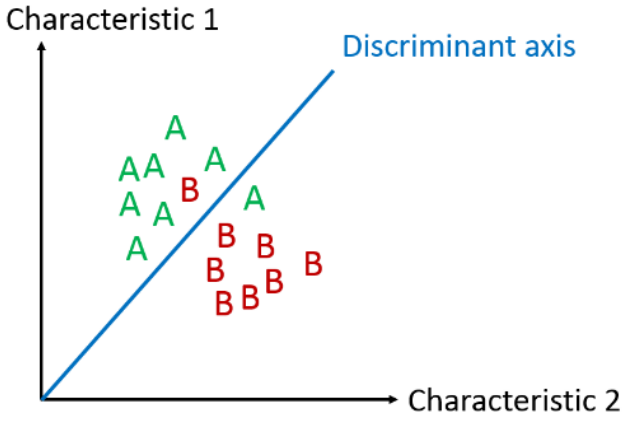


Figure 2-17: Linear discriminant function separating two groups (A, B)

Source: Own illustration after Decker et al. (2010)

The discriminant analysis transforms the initial data from an n-dimensional feature space into a one-dimensional feature space (Decker et al., 2010) (see Figure 2-18).

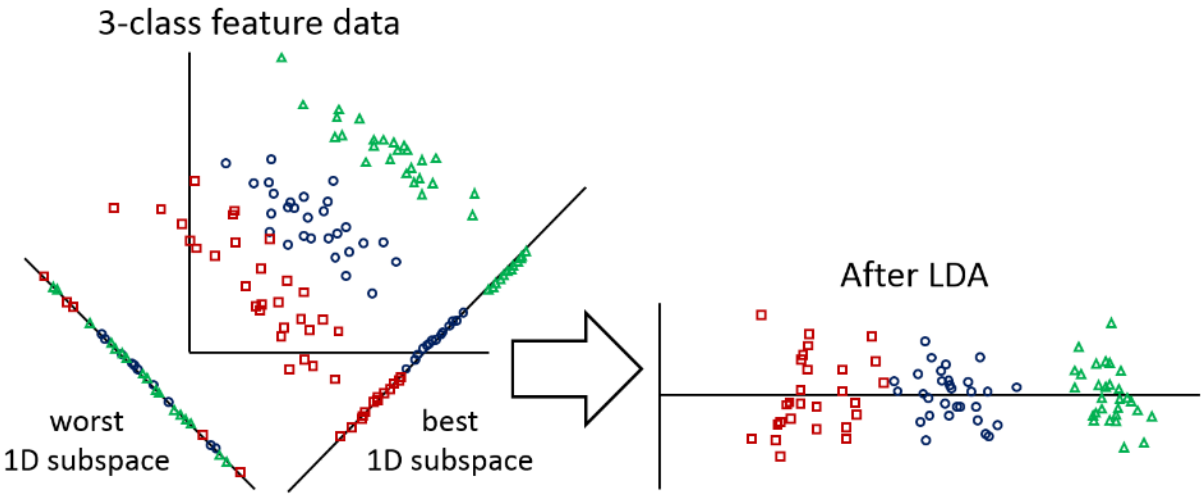


Figure 2-18: Transformation into one-dimensional space by LDA

Source: Own illustration after PBworks (2007)

The general form of the LDA is given by the equation:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_j x_j \tag{33}$$

whereby $x_1, \ldots, x_j$ are the characteristic variables, $b_1, \ldots, b_j$ the appropriate discriminant coefficients and b_0 a constant value (Backhaus et al., 2011). For a two-class-problem the linear discriminant function is:

$$y = b_0 + b_1 x_1 + b_2 x_2 \tag{34}$$

For every object i with $i = 1, \ldots, I_g$ belonging to group g, the discriminant value y_{gi} is determined using Equation (33). For more than two groups ($g > 2$), only one linear discriminant

function would be insufficient, therefore a number of $g - 1$ discriminant functions, dependent on the number of groups, would be used (Decker et al., 2010).

Using the discriminant values y_{gi}, derived from the equation above, the group centroids $\bar{y}_g$ can be calculated using the equation:

$$\bar{y}_g = \frac{1}{I_g} \sum_{i=1}^{I_g} y_{gi} \tag{35}$$

with object i with $i = 1, \dots, I_g$ belonging to group g.

For prediction of the unknown discriminant coefficients the discriminant criterion Γ needs to be maximized:

$$\Gamma = \frac{SS_B(y)}{SS_W(y)} \rightarrow max \tag{36}$$

with the “Sum of Squares Between” revealing the scattering between the groups,

$$SS_B(y) = \sum_{g=1}^{G} I_g (\bar{y}_g - \bar{y})^2 \tag{37}$$

using the total average of discriminant values $\bar{y}$,

and the “Sum of Squares Within” revealing the scattering within the groups.

$$SS_W(y) = \sum_{g=1}^{G} \sum_{i=1}^{I_g} (y_{gi} - \bar{y}_g)^2 \tag{38}$$

The separation of the groups is maximized when the scattering between groups is high and the scattering within groups is low. Therefore the difference between group centroids $\bar{y}_g$of discriminant values should be maximized. In order to find the extreme value of Equation (36), the so called eigenvalue γ, the first derivative of function (36) is calculated and set to null (Backhaus et al., 2011).

$$\gamma = \max(\Gamma) = \Gamma' \rightarrow 0 \tag{39}$$

2.5.2 Support Vector Machines

Support Vector Machines (SVM) is a training algorithm that can be used either for classification (pattern recognition) or regression (function approximation) (Huang et al., 2006). In this work SVM will be used and explained as classification algorithm only. The algorithm analyzes the example objects and builds a model from the training data, in order to predict to which class the new objects belong. The idea of separating data using a hyperplane leads back to Fisher (1936). Vapnik and Chervonenkis were the ones to publish the idea of Support Vector Machines in 1974. However, the theoretical idea did not find its way into common practical usage until 1992 (Huang et al., 2006). Today SVM has been used in a variety of remote sensing studies and showed improved classification results in comparison to traditional classification algorithms like maximum likelihood (Tso & Mather, 2009), since it is able to resolve the trade-off between under- and overestimating very well (Huang et al., 2006). The algorithm is also very computationally efficient, meaning it is able to deal with 100,000 objects sample size (Christianini & Shawe-Taylor, 2000).

SVM includes several different models. The simplest one is the maximal margin classifier (see Figure 2-19). Starting point for every SVM model is a set of training objects with known class membership. Each object is represented by a vector. Assuming the data should be split into two different categories, the maximal margin classifier tries to fit a hyperplane (decision boundary) in between the two classes separating the training objects in the feature space. The gap (margin) between the two classes is maximized so that other example objects will more likely be classified correctly into the right group. In the example of Figure 2-19, hyperplane A fits a larger empty area in between the two classes than hyperplane B, so hyperplane A is also called maximum-margin hyperplane. The hyperplane is only dependent on the training objects that are located closest to it. These vectors are also called support vectors. Hyperplane A, for example, has four support vectors, highlighted in Figure 2-19.

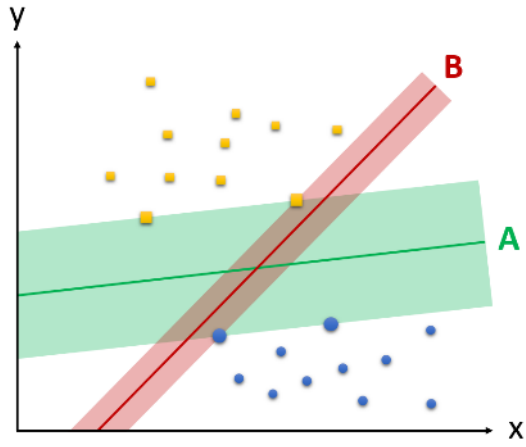


Figure 2-19: Maximal Margin Classifier

Source: Own illustration after Wikimedia (2010)

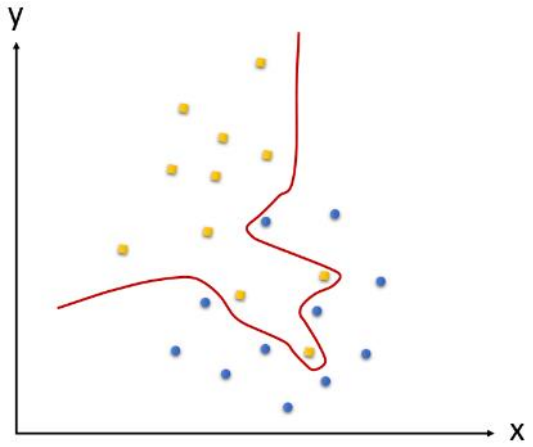


Figure 2-20: Not linearly separable set of objects

Source: Own illustration after Wikimedia (2006)

Often the training data is not linearly separable like in Figure 2-19, but more likely is distributed like in Figure 2-20. This might be caused by measurement errors or just because the two classes naturally overlap. In this case, the hyperplane cannot be fit between the two classes and the maximal margin classifier cannot be used for this type of data. In order to perform a non-linear classification anyway, SVM uses a so called kernel trick. The kernel trick converts the vector space and all training vectors in a higher-dimensional feature space, so that the training vectors become linearly separable and the hyperplane can be fit in-between them. When transforming the hyperplane back into the lower-dimensional feature space, the hyperplane is not linear anymore, maybe not even contiguous. This is when the kernel trick comes into play. Using kernel functions to describe the hyperplane in higher-dimensional feature space, the resulting hyperplane in lower-dimensional feature space is still mathematically describable. That way it is possible to perform the transformation into higher-dimensional feature space and back without having to actually implement the transformations computationally. The support vectors are enough to sufficiently describe the class limits. In contrast to LDA, SVM does not transform the initial data into a lower-dimensional feature space.

For linear as well as for non-linear SVMs it is possible, that no hyperplane exists that is able to split the training data into two separable classes. In this case, a modification of the maximal margin classifier, the so called soft margin method is used. By adding slack variables, that measure the degree of misclassification, the SVM models become more flexible. Slack variables allow the algorithm to classify single objects falsely, but also penalize every incorrect classification in order to keep misclassification to a minimum. Using the soft margin method fewer support vectors are needed and the risk of overfitting is minimized (Richards & Jia, 2006). For the interested reader, a mathematical explanation can be found in (Richards & Jia, 2006) or (Tso & Mather, 2009).

2.5.3 Random Forest

Over the past years, RF has become a widely established and often used algorithm for regression and classification problems throughout different scientific fields (Schmidt et al., 2011). In comparison to SVM, less parameters are needed with equal performance (Tso & Mather, 2009). RF belongs to the category of decision tree classification algorithms. A common binary decision tree always consists of a root node and several other nodes, each divided into two further nodes, except the end or leaf nodes (see Figure 2-21). The class label of a new object is predicted by putting the input vector of a new object down the tree based on a decision rule. The final classification or "vote" of the tree is given by the appropriate end node (Breiman & Cutler, 2001).

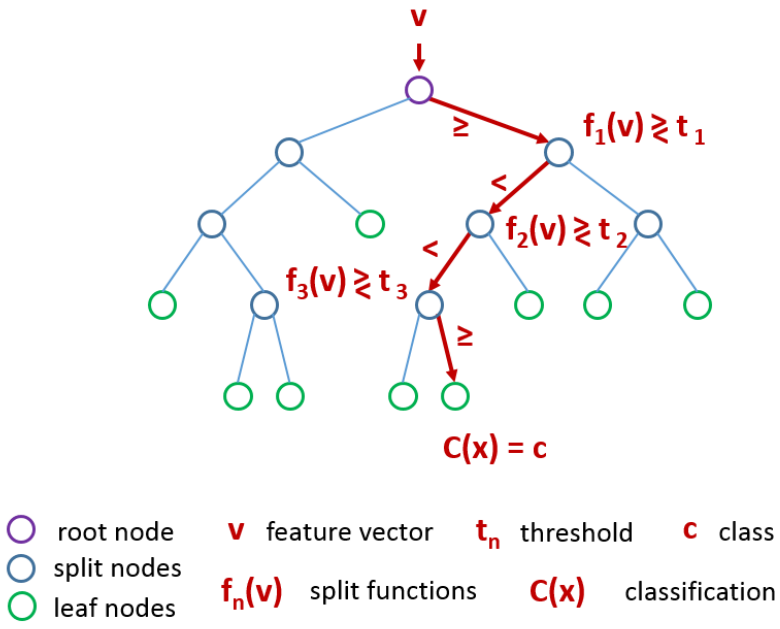


Figure 2-21: Decision tree

Source: Own illustration after Landscape Toolbox Wiki (2013)

In contrast to other decision tree classifiers a RF consists of multiple randomly generated, de-correlated decision trees (see Figure 2-22). The decision trees are de-correlated in order to reduce variance (Hastie et al., 2009). Each tree of the RF gives a class vote. The class membership is finally decided using majority vote (Hastie et al., 2009). First, the algorithm draws random bootstrap samples of size N from the training data for each decision tree. For each of the bootstrap samples a random forest tree T_b is grown. In contrast to common decision trees, a RF decision tree only searches for the best split point for a subset of features. For each end node, the algorithm recursively repeats the following steps until the minimum node size n_{min} is reached:

1. Select m variables at random from the p variables with $m \leq p$
2. Pick the best variable/ split-point among the m
3. Split the node into two daughter nodes

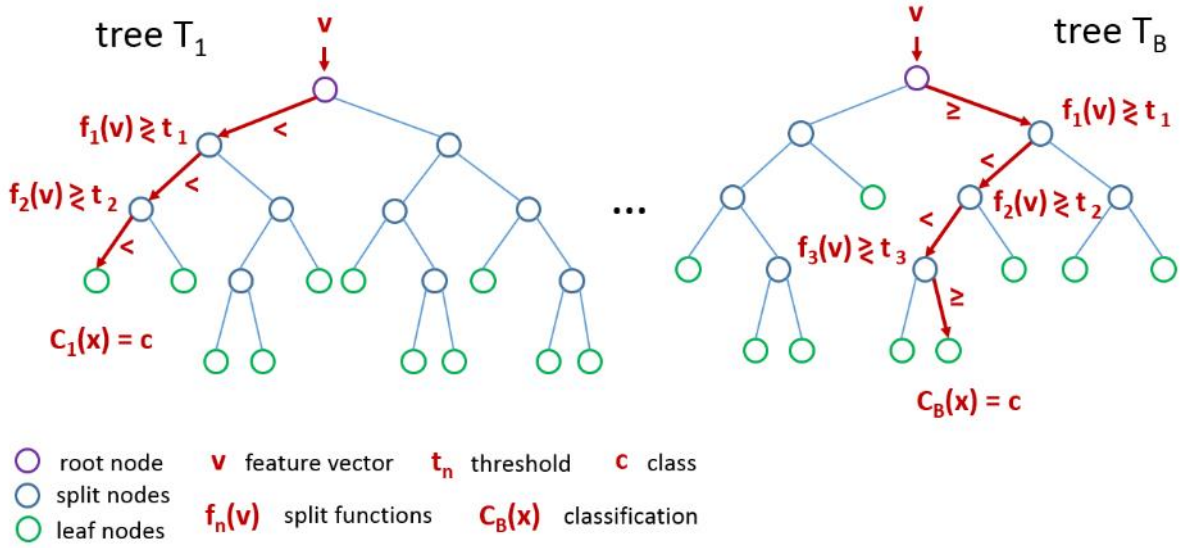


Figure 2-22: Random Forest model

Source: Own illustration after Shotton et al. (2013)

Reducing m also reduces the correlation between any pair of trees in the ensemble and therefore also reduces the variance. The inventors of RF recommend to select a number of

$$m = \sqrt{p} \tag{40}$$

variables. Hastie et al. (2009), however, recommend to treat m as a tuning parameter, since its best value depends on the problem.

Output of the algorithm is an ensemble of trees T_b with $b = 1, \dots, B$. In order to make a prediction at a new point x every random forest tree predicts the class value of x: $C_b(x)$. The final class membership of x, $C_{rf}^B(x)$ is estimated using a majority vote (Hastie et al., 2009):

$$C_{rf}^B(x) = majority\ vote\{C_b(x)\}_1^B \tag{41}$$

In contrast to standard decision trees, RF is able to take more than one feature into account at one node. By using random sampling and the majority vote of several decision trees, RF also has the advantage of being insensitive to noise in the data or overtraining. Additionally, in consequence of random sampling, RF is very efficient on large amounts of data, concerning number of classes, samples and features. However, the training time for a random forest increases linearly with the number of trees.

2.6 Fundamentals of Accuracy Assessment

Accuracy assessment reveals the quality of a thematic map, created from remotely sensed data, also called thematic accuracy, in contrast to positional accuracy (Congalton & Green, 2009). So in order to find out how accurate and useful the resulting classification is, it is helpful to conduct an accuracy assessment, especially, when resulting maps are going to be used for decision making purposes, like developing city plans, addressing slum issues or disaster prevention. It is not only useful in these situations, but very much needed, as Congalton & Green (2009) stated: “A classification is not complete until its accuracy is assessed”. It helps us to estimate, how good the classification is and also to improve its quality by finding sources of errors. In order to compare certain classification techniques or algorithms, like in case of this work, an accuracy assessment can be used to identify which one works best for the certain kinds of data (Congalton & Green, 2009). It also helps to investigate the suitability of the classification model and to find parameters needing adjustment (Allouche et al., 2006).

The accuracy assessment is usually conducted using reference data that is believed to reveal the correct land cover or land use. Sources generally include ground truth data or field measurements, respectively, satellite or aerial images with a higher resolution as well as maps or GIS layers derived from these images with a very high accuracy. At this point, we need to note that all reference data, no matter what the source is, may introduce inaccuracy to some degree. Other sources of inaccuracy may result from temporal inconsistencies between reference data and satellite imagery or incorrect training data used for classification. Since it is often impossible to assess the accuracy of every single pixel of the classification, usually samples are taken. In general it is better to take only few but highly accurate samples opposed to many samples with diminished accuracy. Every sample value of the classification is then compared to the reference data in order to finally analyze the reasonableness and statistical significance of the classification result (Congalton & Green, 2009). In this case, ground-truth information was given for every single block of buildings in Mumbai.

A contemporary and widely-used tool for computing measures of quality is the error matrix, also called confusion matrix or contingency table (Lillesand et al., 2004), see example in Table 2-2. The error matrix is a square matrix with class labels of the reference data as columns and class labels of the classified data as rows. The diagonal contains the number of pixels or objects classified correctly, according to the reference data, whereas the numbers on the right and left of the diagonal represent pixels or objects that were misclassified.

Table 2-2: Example error matrix

		Reference Data					
		Class 1	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Row total**
Classified Data	**Class 1**	**70**	8	5	0	0	83
	Class 2	3	**55**	0	0	0	58
	Class 3	0	0	**37**	0	0	37
	Class 4	0	0	0	**99**	2	101
	Class 5	0	0	0	6	**121**	127
	Column total	73	63	42	105	123	**406**

Quality measures derived from the error matrix include Overall accuracy, User's accuracy, Producer's accuracy, Kappa coefficient and True Skill Statistic. These are non-site specific quality measures, since they do not take into account the location of misclassifications.

The Overall accuracy is calculated by dividing the sum of the major diagonal (correctly classified objects or pixels) by the total number of objects or pixels. For the example of Table 2-2 the Overall accuracy would be:

$$Overall\ accuracy = \frac{sum\ of\ major\ diagonal}{total\ number\ of\ samples} = \frac{382}{406} = 0.94 = 94\% \qquad (42)$$

The problem with the Overall accuracy is that it is only an average value of all classes, so it does not reveal, if the error is evenly distributed between all classes or if the classification might have performed really badly for some classes while the results of other classes are very good.

Producer's and User's accuracy, on the other hand, describe the accuracies of individual classes. Producer's accuracy is a quality measure of omission error. Omission errors are errors of exclusion that occur when a pixel or object is not classified as belonging to the class it actually belongs to. The producer's accuracy shows how accurate the classification is from the perspective of the producer of the map, so the measure represents the percentage of pixels or objects on the map that are classified correctly according to the reference data. Producer's accuracy is calculated by dividing the diagonal value (correctly identified pixels or objects of a given class) by the column total (number of pixels or objects actually in that class according to reference data):

$$Producer's\ accuracy = \frac{diagonal\ value}{column\ total} \qquad (43)$$

For Class 1 for example, the Producer's accuracy would be 96%:

$$Producer's\ accuracy = \frac{70}{73} = 0.96 = 96\% \tag{44}$$

with an omission error of 4%:

$$Omission\ error = \frac{3}{73} = 0.04 = 4\% \tag{45}$$

User's accuracy by contrast represents a quality measure of commission error or errors of inclusion that occur, when a pixel or object is classified as belonging to a class it actually does not belong to. Every commission error is always an omission error of another class. The User's accuracy reveals the accuracy of a classification map from the perspective of the user, so it represents the percentage of pixels or objects on the map that are actually what they are labeled as. It is calculated by dividing the number of correctly identified pixels or objects by the number of pixels or objects claimed to be in that class from the classification:

$$User's\ accuracy = \frac{diagonal\ value}{row\ total} \tag{46}$$

For Class 1 for example the User's accuracy would be 84%:

$$User's\ accuracy = \frac{70}{83} = 0.84 = 84\% \tag{47}$$

with a commission error of 16%:

$$Commission\ error = \frac{13}{83} = 0.16 = 16\% \tag{48}$$

Another basic quality measure used in this work is the Kappa coefficient or Kappa statistic. Kappa is a statistical measure and was first introduced by Cohen in the psychology field, why it is also called Cohen's Kappa (Cohen, 1960). Congalton was the first to introduce the measure to the remote sensing community in 1981. Today, the Kappa coefficient belongs to the standard quality measures given in every accuracy assessment. The idea of Kappa is that a certain amount of agreement is also to be expected when the classification algorithm would judge purely coincidental. So the attempt is made to eliminate the influence of chance agreement from the quality measure.

Kappa measures the difference between the observed agreement (major diagonal of the error matrix) and the agreement expected by chance (row and column totals) of classification and reference data. Given an error matrix with i rows and j columns the observed agreement p_0 and the chance agreement p_c are calculated as follows:

$$p_0 = \sum_{i=1}^{k} p_{ii} \tag{49}$$

$$p_c = \sum_{i=1}^{k} p_{i+} p_{+j} \tag{50}$$

With p_{i+} being the row totals and p_{+j} being the column totals.

The Kappa coefficient κ is then calculated using the equation:

$$\kappa = \frac{p_0 - p_c}{1 - p_c} \tag{51}$$

The resulting values of Kappa range between -1 and +1. However, in remote sensing a positive correlation between classification and reference data is expected, so the Kappa values should always be positive with 1 indicating a complete agreement between classification and reference data and 0 indicating no agreement. Landis & Koch (1977) categorized the kappa values into six groups:

- 0 poor,
- 0.01–0.2 slight,
- 0.21-0.4 fair,
- 0.41-0.6 moderate,
- 0.61-0.8 substantial, and
- 0.81-1 almost perfect

agreement. Therefore, Kappa can also be used to compare two error matrices.

Although Kappa is one of the most widely used quality measures, it has been criticized in many studies for being dependent on prevalence. In order to address this issue a relatively new quality measure, the True Skill Statistic (TSS), that is said to be independent of prevalence while still keeping all advantages of kappa, was tested in this study (Allouche et al., 2006). Additionally, this measure of model performance is not affected by the size of the validation data set. TSS, also called Hanssen-Kuiper skill score or Hanssen-Kuipers discriminant is calculated from a 2x2 error matrix, see example in Table 2-3.

Table 2-3: 2x2 example error matrix

		Reference Data	
		Class 1	**Class 2**
Classified Data	**Class 1**	**a**	b
	Class 2	c	**d**

The TSS is calculated from the sum of the probability that the model will correctly classify class 1 (sensitivity) and the probability that the model will correctly classify class 2 (specificity) minus one, see equation below:

$$TSS = \frac{ad - bc}{(a + c)(b + d)} = Sensitivity + Specifity - 1 = \frac{a}{a + c} + \frac{b}{b + d} - 1 \tag{52}$$

Like Kappa, TSS ranges from -1 to +1 with 1 indicating perfect agreement and 0 indicating a performance no better than random (Allouche et al., 2006).

3 Study Area and Data Description

In this Chapter, the geographical location, history, climate, administrative division, population and slum distribution of the study area and megacity Mumbai will be illustrated. Further, the Radar imagery used for classification and the reference data used for training and accuracy assessment will be introduced.

3.1 Profile of the Study Area Mumbai

The study area of this thesis comprises the megacity of Mumbai at the Western coast of India. Mumbai, until 1996 officially termed Bombay, is the capital of Maharashtra state. The city is located on a long and narrow peninsula bounded by the Arabian Sea, called Salsette Island (see Figure 3-1), partially shared with the Thane district. The peninsula was originally formed out of seven small islands separated by swamps that were merged into one as a consequence of silting and land reclamation (Risbud, 2003).

The climate in Mumbai is tropical hot and humid (Bhagat & Jones, 2013). The region experiences four distinct seasons: Winter (December - February), Summer (March - May), Monsoon (June - September) and Post-monsoon (October - December) (India Meteorological Department, 2015). During monsoon season heavy rainfalls occur that average about 2000mm per year (Bhagat & Jones, 2013).

Administratively, the mega-urban region is divided into the three following entities, namely from the smallest to the largest (Bhagat & Jones, 2013):

1. Municipal Corporation of Greater Mumbai (MCGM)
2. Mumbai Urban Agglomeration (UA)
3. Mumbai Metropolitan Region (MMR)

This study is focused on the area under administration of the MCGM, hereafter simply referred to as Greater Mumbai, which covers about 603.4 square kilometers and comprises the Island City or Mumbai City district and the surrounding Suburban districts. The suburbs of Greater Mumbai are very much part of the city, although they are administratively designated as suburbs (Bhagat & Jones, 2013).

Mumbai UA covers 1,135 sq. km. and consists of Greater Mumbai and five other Municipal Corporations, namely Mira-Bhayandar, Thane, Navi Mumbai, Kalyan-Dombivli and Ulhasnagar, as well as two Municipal Councils, namely Ambarnath and Badlapur. Covering 4,355 square kilometers, the MMR is the largest urban agglomeration of India and consists of the Mumbai UA plus two additional Municipal Corporations, namely Vasai-Virar and Bhiwandi-Nizampur, and seven Municipal Councils, namely Alibag, Karjat, Khopoli, Matheran, Panvel, Pen and Uran (Risbud, 2003).

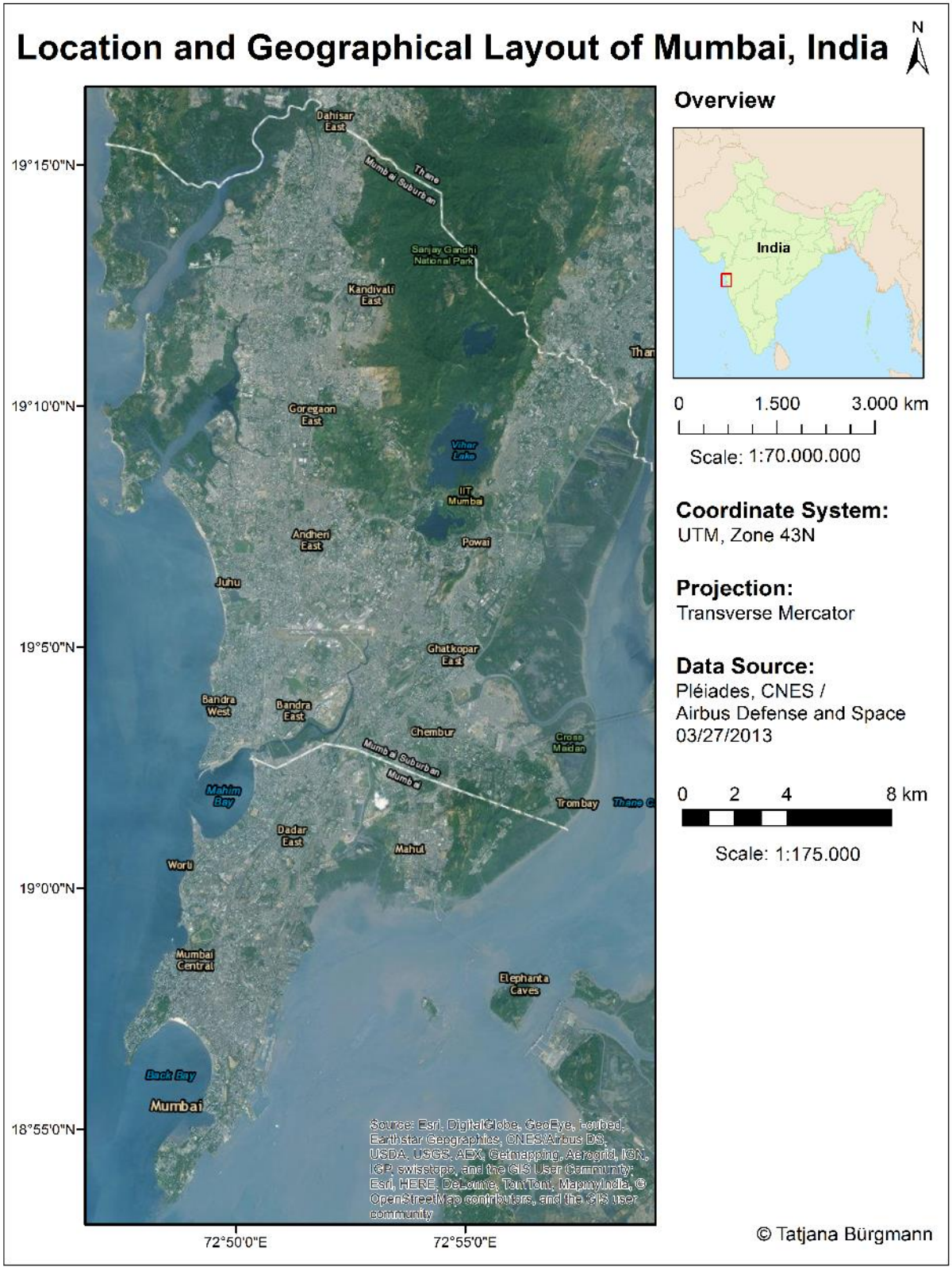


Figure 3-1: Location and geographical layout of the study area Mumbai, India

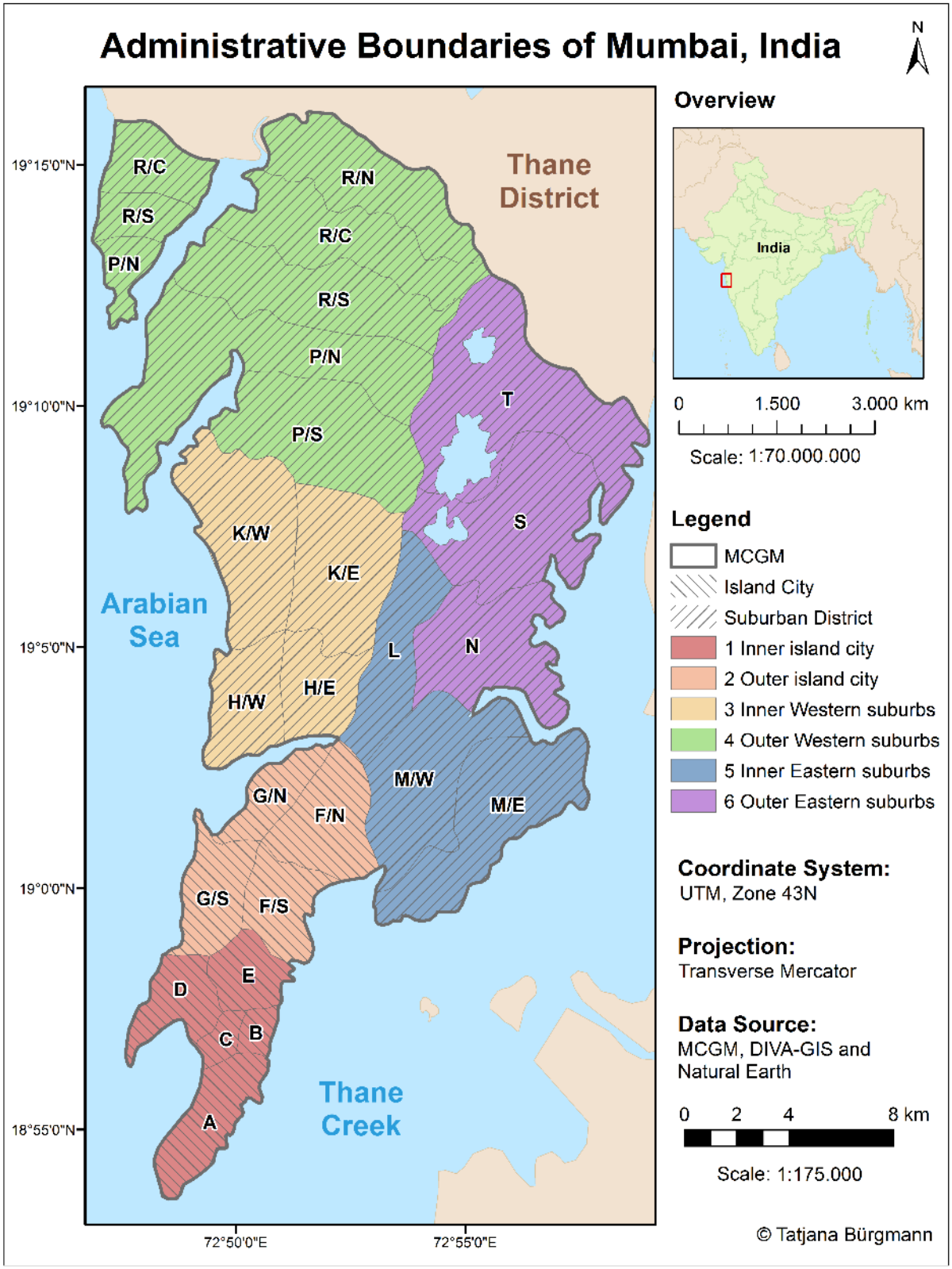


Figure 3-2: The study area of Greater Mumbai with its wards, zones and districts

Source: Own illustration based on data derived from the Municipal Corporation of Greater Mumbai (2015) (ward boundaries), DIVA-GIS (2009) (country outlines) and Natural Earth (2015) (administrative boundaries)

To simplify the administration of the Municipal Corporation, Greater Mumbai is again divided into six zones, each consisting of three to five so called “wards” that are named alphabetically, sometimes with the addition of North (N), South (S), East (E), West (W) or Center (C) (see Table 3-1 and Figure 3-2).

For certain Indian cities like Mumbai the term *ward* is used as administrative union or city district, respectively. The Greater Mumbai area covers 24 wards in total (Risbud, 2003).

Table 3-1: Districts, zones and wards of Greater Mumbai

District	Zone	Wards
Mumbai City	1 Inner island city	A, B, C, D, E
	2 Outer island city	FS, FN, GS, GN
Mumbai Suburban	3 Inner Western suburbs	HE, HW, KE, KW
	4 Outer Western suburbs	PN, PS, RS, RC, RN
	5 Inner Eastern suburbs	L, ME, MW
	6 Outer Eastern suburbs	N, S, T

In the 16th century, today’s megacity Mumbai began its existence as a fishing village, locally known as *Kolis*. In 1534 the Portugese captured the seven islands and established a factory on the new-gained land, called “Bom Bahia” by them, in English meaning “the good bay”. Englishmen, who attacked Bombay while they were at war with Portugal, then later pronounced its name “Bombay” (Risbud, 2003). The city was established during the British rule of India, more specifically in 1661 when the British East India Company occupied the islands (Bhagat & Jones, 2013). Bombay developed as an important port, since it is a natural deep-water port, that was used by the British for more than two centuries mainly for trading (Risbud, 2003). The Municipal Corporation of Bombay was established in 1872 (Bhagat & Jones, 2013).

With the growth of manufacturing facilities, especially in the cotton textile sector, the city became more and more industrialized and finally grew from a fishing village and trading post to the largest urban agglomeration and the financial capital of India, generating 33% of the country’s income tax, 60% of customs duty, 20% of central excise duty and 40% of foreign trade (Bhagat & Jones, 2013). The industrialization resulted in the displacement of the Koli fishermen into run-down housing with poor living conditions, today called Dharavi slum. As the city of Mumbai progressively expanded, both industrially and commercially, the number of slums increased dramatically (Adhikari, 2004). After the late 1970s, the manufacturing sector began to decline, leading to a growth of the service sector including transport, communications, social and personal services besides real estate, construction, banking, financial and IT sector, especially within the city core (Bhagat & Jones, 2013).

According to official figures, today Greater Mumbai is home to 12.47 million people and has a very high population density of 20,692 persons per square kilometer (Census of India, 2011a).

Thus Mumbai is one of the currently 28 megacities in the world with a population over ten million (United Nations, 2014). Notably the population living in areas outside the official metropolitan area is steadily increasing (Bhagat & Jones, 2013). Mumbai UA has a population of about 18 million people (Census of India, 2011b). MMR grew from a population of 3 million in 1951 over 12 million in 2001 to a population of approximately 21 million as of Census of India, 2011. The agglomeration of almost 21 million inhabitants accounts Mumbai to be the fifth largest city, concerning the population, in the world (United Nations, 2014) and the most populous city of India, with highest population densities (Taubenböck & Kraff, 2014). Greater Mumbai accounts for more than half of the population of MMR although it only comprises about 14% of the geographic area. The population in Island City was more or less stagnating over the last several decades, whereas in the Suburbs population rose from about 11 thousand in 1981 to about 21 thousand in 2011 (Bhagat & Jones, 2013).

41.3% (Census of India, 2011a) of the population in Greater Mumbai, about 5.2 billion people, live in one of the about 1,959 slums (Census of India, 2001), but occupy only 12.85% of the city's total area (Asha, 2006). With slums distributed all over the city, Mumbai is one of the worst living places across the globe, especially for the poor (Adhikari, 2004). Only a small percentage of the slum population is located on the Island City, whereas most are located in the Eastern and Western suburbs (Bhagat & Jones, 2013). The greatest slum settlement areas are found in the Inner Western Suburbs (Risbud, 2003). The slum growth in the Eastern and Western suburbs was mainly driven by migration but also because cotton textile and other factory owners used to provide housing for their workers in close proximity to their factories. The decline of the formal sector and rapid increase of the informal sector since the 1980s, resulted in housing shortage and thus workers who migrated to Mumbai were forced to look for their housing on their own. Since the migrants were not paid very well, their choices of housing were limited. This is the reason, why many people started to and still continue to live in one of the many slums of Mumbai (Bhagat & Jones, 2013).

3.2 Data Description

In the following the data basis in form of two Radarsat images and a shapefile containing all blocks of buildings of Mumbai with respective information about formal or informal settlement.

3.2.1 Radarsat-2 SAR Data

In this study, two dual-polarized (HH/HV and VV/VH), multi-date Radarsat-2 images of Mumbai were used (see Figure 3-3 and Figure 3-4). For both polarizations four Kennaugh-elements were calculated during data-preparation, namely K_0, K_1, K_5 and K_8, since these elements are also available for Sentinel-1. Therefore the results of this study can be seen as preparation for the Sentinel-1 mission (European Space Agency, 2015).

Both images were acquired in September 2014 at the end of monsoon season, when vegetation is in full growth and therefore maximizes the difference between urban and rural areas (see Table 3-2). The images cover approximately 4,160 square kilometers including the whole MCGM area of Mumbai. The image size in pixels is 13,000 x 12,800 with a resampled resolution of 5 meters. The original pixel size or nominal pixel spacing (range x azimuth) was 4.7 x 5.1 meters (MacDonald Dettwiler and Associates Ltd., 2014b)

Table 3-2: Radarsat-2 imagery details

Acquisition date	Polarization	Incidence angle	Beam Mode	Swath
09/21/2014	HH/HV	48.2°	Fine	50 km
09/28/2014	VV/VH	48.2°	Fine	50 km

According to Henderson & Xia (1997), shorter wavelengths (i.e. X-, C- and L-Band), cross polarizations and an incident angle greater than 35 degrees is preferably for urban settlement detection using SAR data, which is all given by the two images used.

Both images are Single Look Complex (SLC) products and have been georeferenced and aligned with the satellite track (MacDonald Dettwiler and Associates Ltd., 2014b). Figures Figure 3-5 to Figure 3-16 each illustrate a subset of the single bands of the HH/HV polarized Radarsat image (Kennaugh-elements K_0, K_1, K_5 and K_8) showing the slum of Dharavi, in comparison to an optical imagery. Figure 3-4 shows the location of the subset in the overall scene. The deduced Kennaugh elements Absorption, Diattenuation, Polarizance, Retardance and Scale can be seen in Figures Figure 3-11 to Figure 3-16. For the respective illustration of the single bands of the VV/VH polarized image see Appendix I: VV/VH polarized Radarsat imagery.

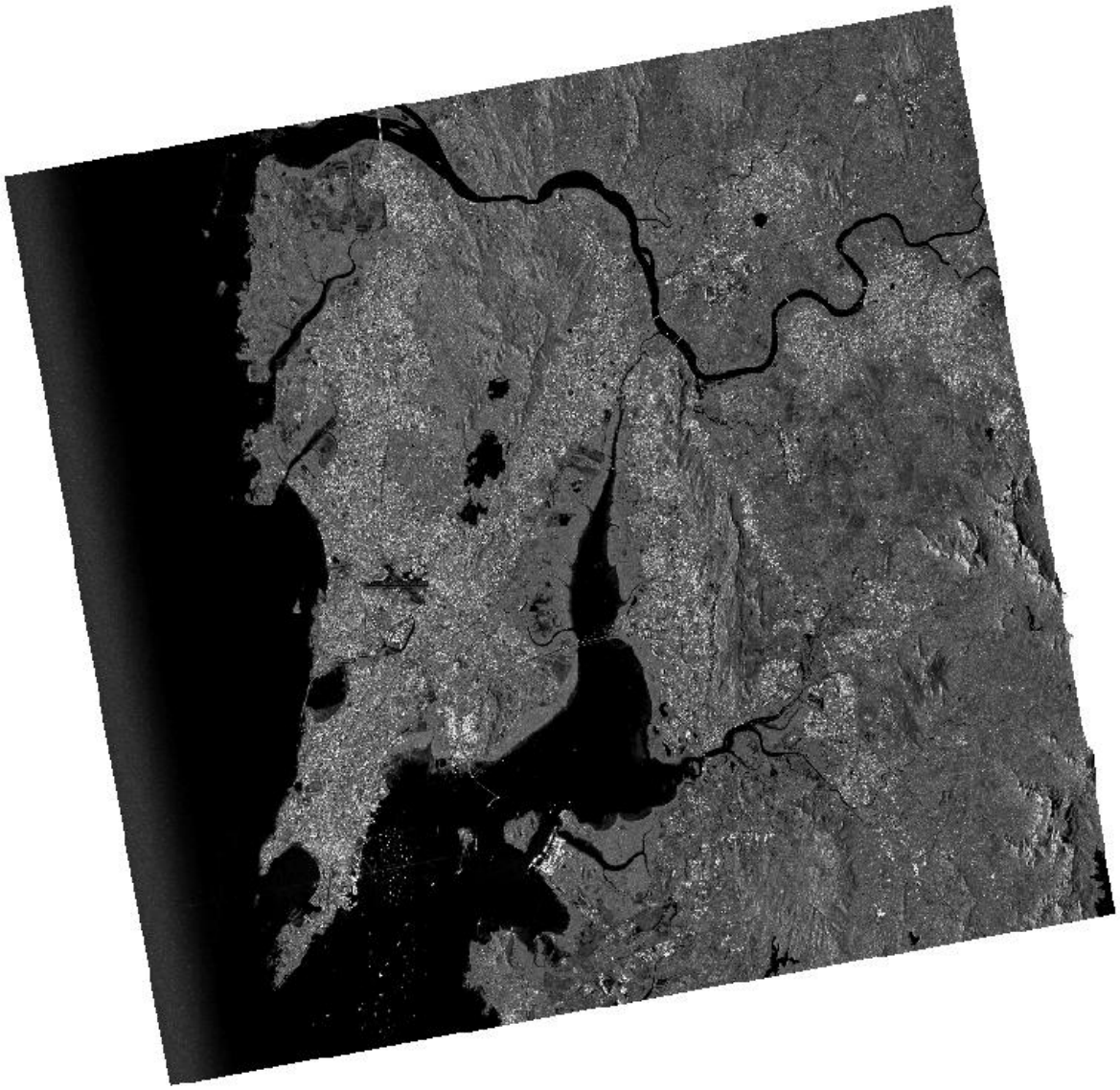


Figure 3-3: Intensity image of Mumbai of 09/28/2014 having VV/VH polarization)

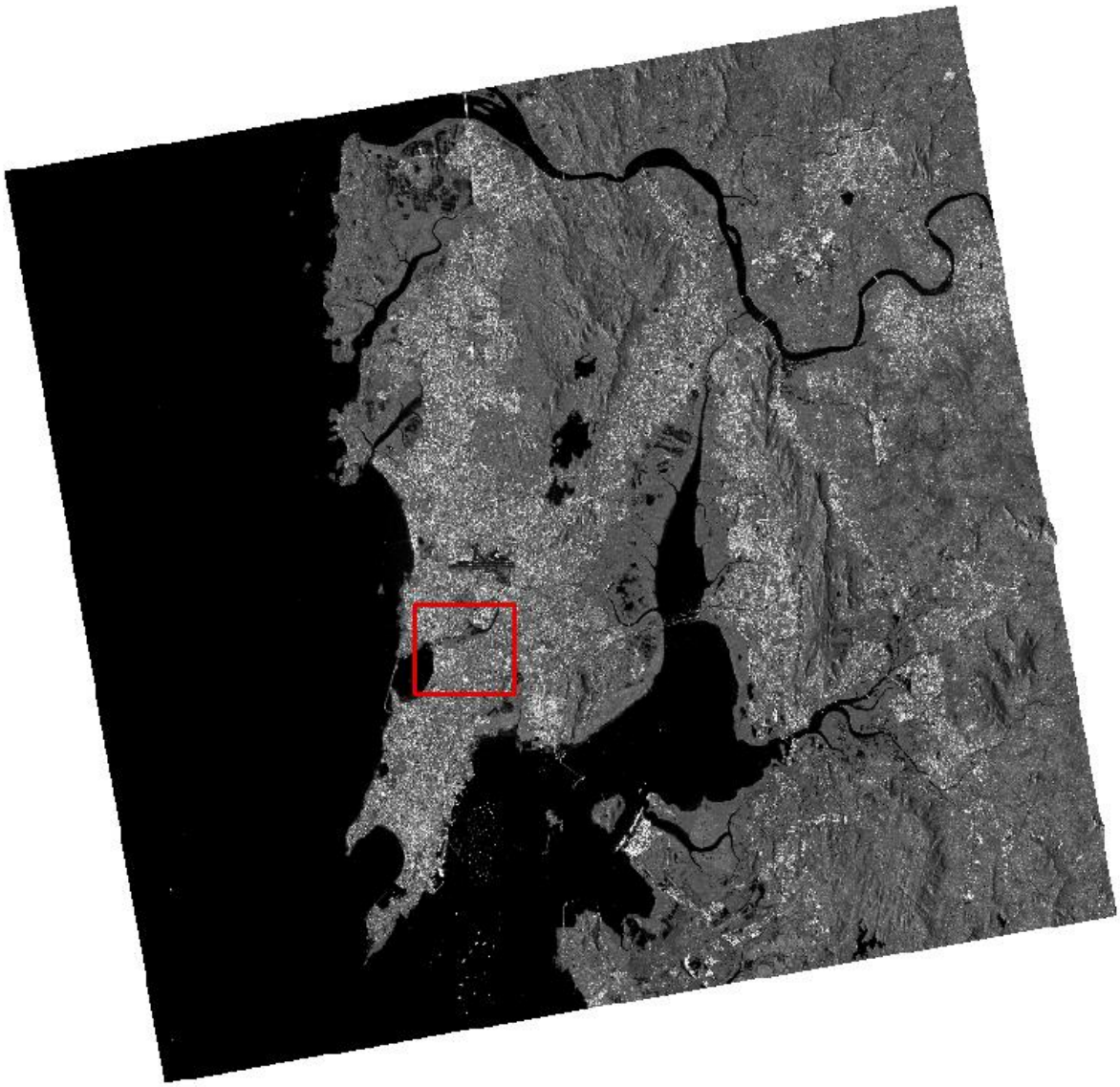


Figure 3-4: Location of the subset in the overall scene (in the background: intensity image of Mumbai of 09/21/2014 having HH/HV polarization)

Figure 3-5: Kennaugh-element K_0 (HH/HV)

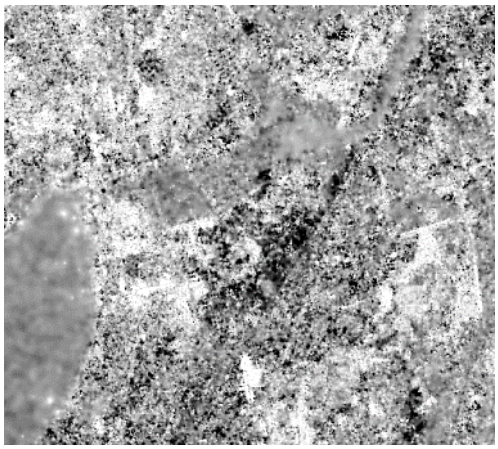


Figure 3-6: Kennaugh-element K_1 (HH/HV)

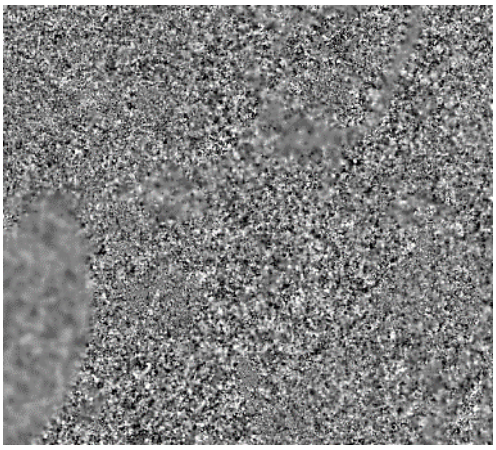


Figure 3-7: Kennaugh-element K_5 (HH/HV)

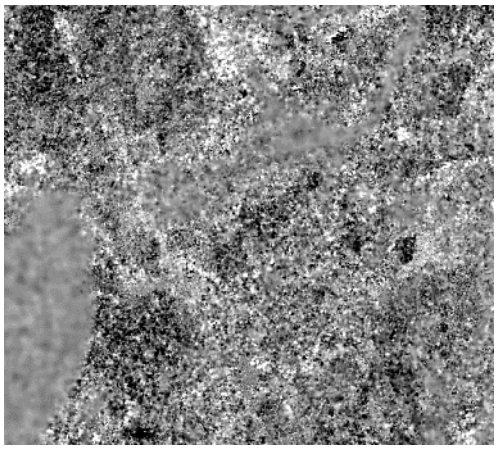


Figure 3-8: Kennaugh-element K_8 (HH/HV)

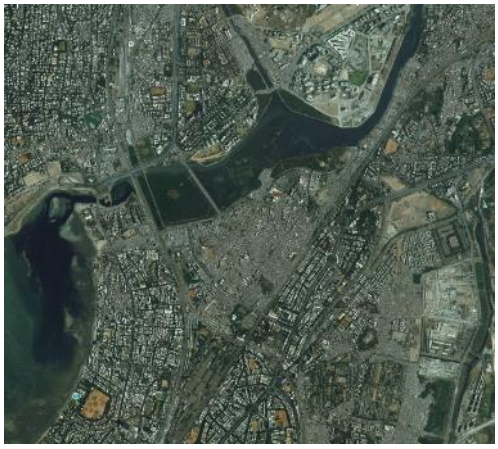


Figure 3-9: Optical Pléiades image with 0.5m resolution

Figure 3-10: Auxiliary data (formal areas in green and slum areas in red)

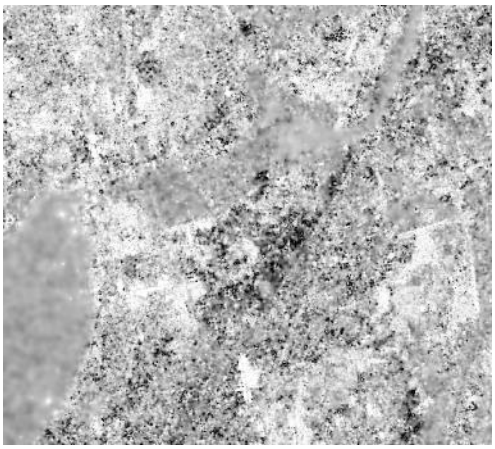


Figure 3-11: Absorption (HH/HV)

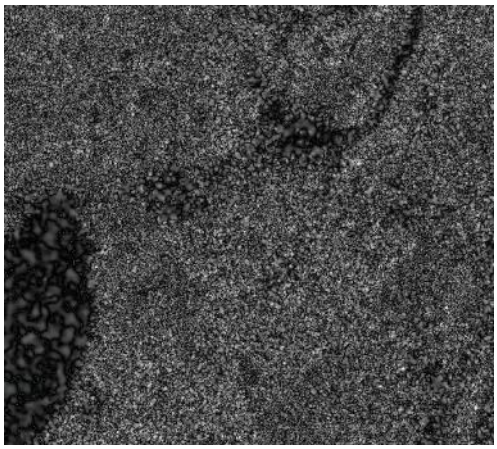


Figure 3-12: Diattenuation (HH/HV)

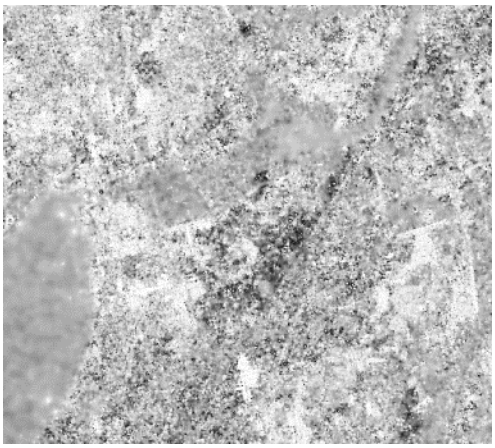


Figure 3-13: Polarizance (HH/HV)

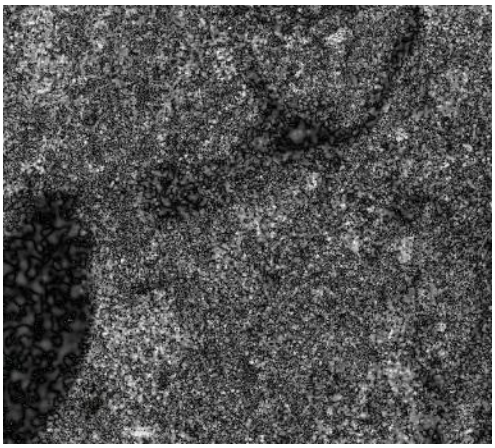


Figure 3-14: Retardance (HH/HV)

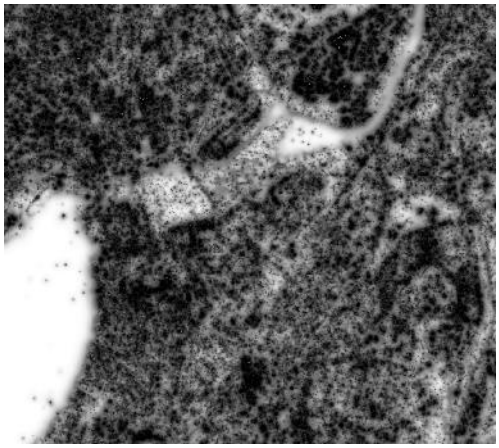


Figure 3-15: Scale (HH/HV)

Figure 3-16: Optical Pléiades image with 0.5m resolution

3.2.2 Auxiliary Data

The auxiliary data, in form of a digitized shapefile of all blocks of buildings of Mumbai, is reused from a former study conducted by Kuner (2014). In that study, a DigitalGlobe World View-2 scene from May 3, 2012 with a resolution of 0.5 meters and an accuracy of 4.25 meters was used as reference. All adjacent buildings used for human habitation and exhibiting a similar structure were summarized as blocks of buildings and manually classified according to their height and density. All blocks of buildings having a low height, minimal distance between buildings, with streets visible between houses and being unstructured but displaying a compact structure, were classified as slums. All other blocks of buildings were classified according to their height (high, medium, low) and density (high, medium, low). Using these criteria, 9,446 blocks of buildings in total, covering an area of 128.7 square kilometers, were digitized as polygons and provided with structural features.

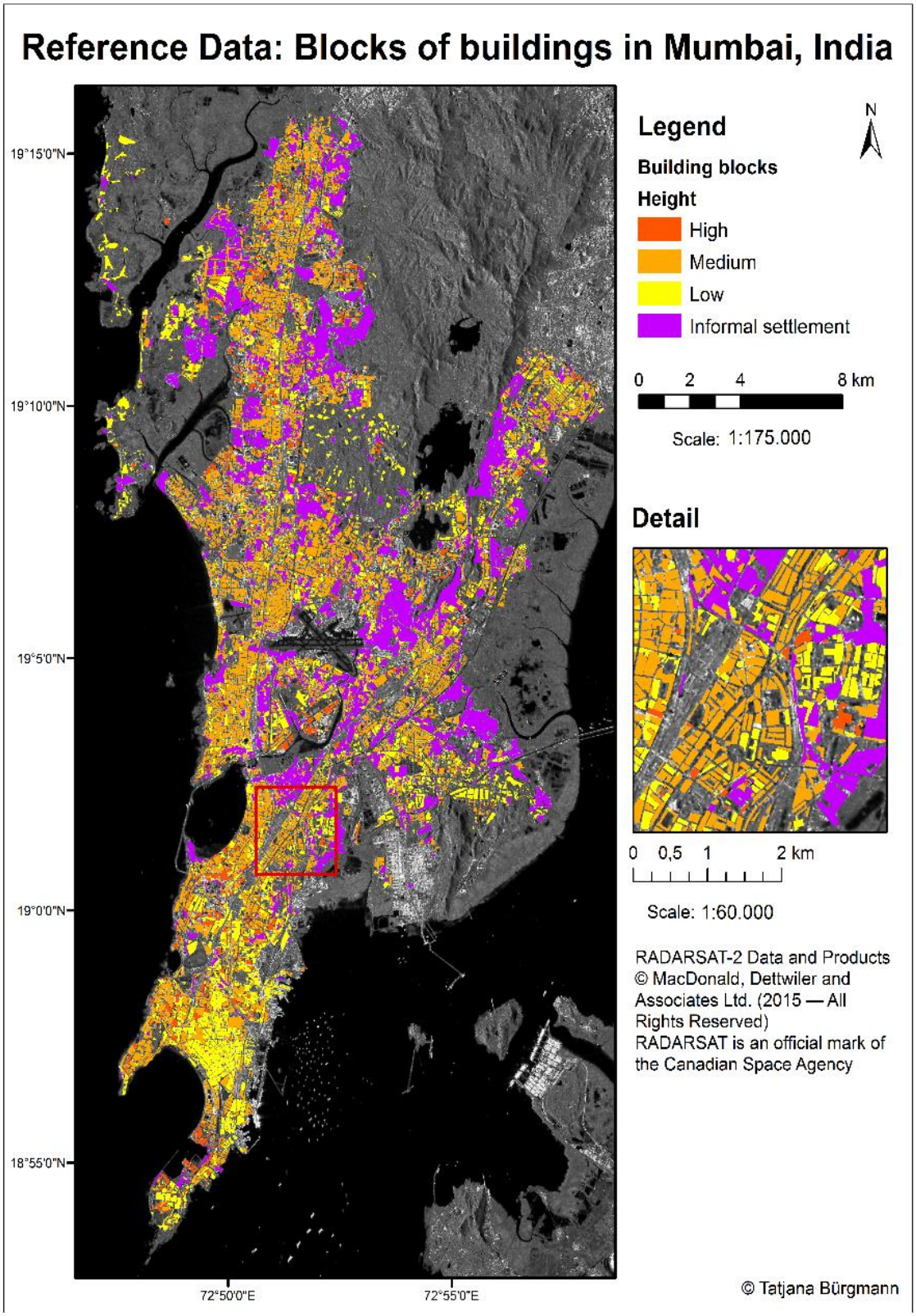


Figure 3-17: Reference data in form of digitized blocks of buildings of Mumbai classified into formal and informal with the Radar data (HH/HV K0) in the background

4 Methodology

In this Chapter, the software used and the strategy of data evaluation will be presented. The process of deriving texture measures, feature selection, classification and accuracy assessment will be explained in detail.

4.1 Software

This Subchapter gives an overview over the software used: eCognition, used for calculating GLCM texture measures and R for subsequent feature selection, classification and accuracy assessment.

4.1.1 Trimble's eCognition®

The software eCognition® was developed by Definiens AG® in Munich, Germany, and is available on the market since 2000. Definiens® business assets, including the eCognition® software, were absorbed by Trimble Navigation Ltd® in 2010. The software can be used for geospatial applications, such as automatic data interpretation, feature extraction and change detection. It was the first computer program to introduce object-based image analysis, in contrast to traditional pixel-based classification methods. It is also capable of handling a variety of geospatial data sources, like satellite, aerial, laser scanner, Radar, and hyperspectral data.

The eCognition Suite® includes three software components: eCognition Developer®, eCognition Architect® and eCognition Server®. They are all stand-alone products, but can also be used in combination. eCognition Developer® is used to develop rule sets to solve image analysis tasks using remote sensing data. eCognition Architect® can be used for non-technical users to calibrate and execute rule sets prepared in eCognition Developer®. eCognition Server® provides an environment for batch processing of image analysis applications (Trimble Germany GmbH, 2015). In this work eCognition Developer® 9.0 was utilized to develop rule sets for the calculation of GLCM texture measures.

4.1.2 Statistical Programming Language R

R is a free, open source programming language and software environment for conducting statistical analysis and graphics. The advantage of open source software is that everyone is allowed to access the code and many people contribute to it and develop new functions or packages (Field et al., 2012). These packages are all stored in the Comprehensive R Archive Network (CRAN). R was developed by Robert Gentleman and Ross Ihaka at the Statistics Department of the University of Auckland, New Zealand. The name of the programming language origins from the initial letters of its developers, but is also evidence of the programming language S that was inspiration for the development of R. Due to this fact, R has more object-oriented features than other statistical programming languages. It can be seen as a different implementation of S, since most code written for S also runs in R. S was developed by John Chambers, who today is member of the development core team of R.

R provides many statistical functions, such as modelling, statistical tests, time-series analysis, classification or clustering and can be further extended by packages. Another highlight of R is

the plotting function that can be used to easily produce well-designed plots for publication purposes. The software environment is platform independent, so it can be used under UNIX, Windows as well as MacOS platforms. Data formats in R include vectors, matrices, arrays, data frames and lists. A data frame is structured similarly like a table in a relational database.

Although R for Microsoft Windows comes with a graphical user interface (GUI) called RGUI, there are several other GUIs available, such as RStudio, Java GUI for R or R Commander. In this work RStudio was used, since it is a platform independent and open source development environment. For further information on R and its usage, the book of Field et al. (2012) is recommended.

4.2 Strategy for Data Evaluation

Figure 4-1 shows the steps of the data evaluation in this thesis. Each step is explained in more detail in the following.

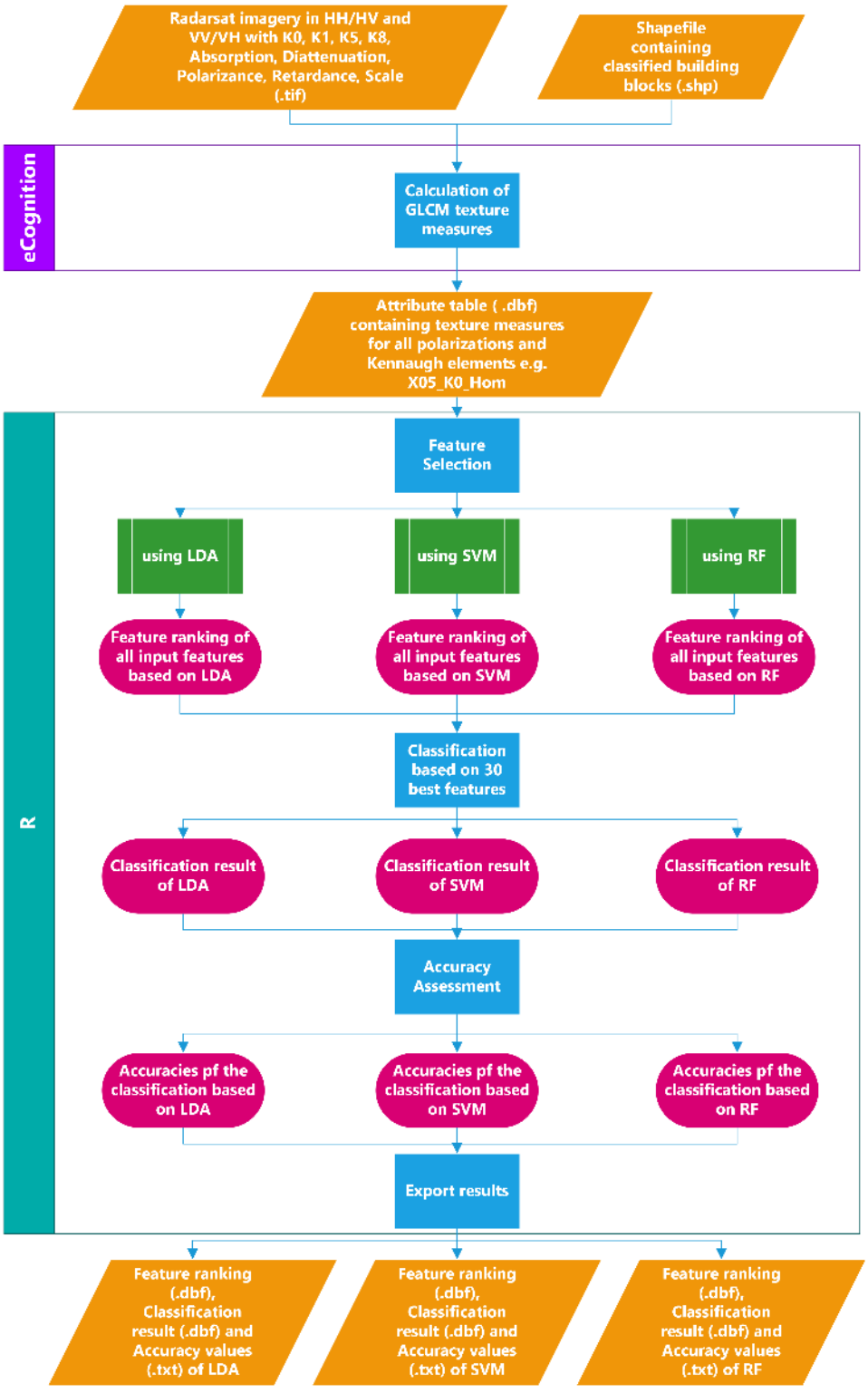


Figure 4-1: Flow chart showing the strategy for data evaluation

4.2.1 Calculation of Texture Measures

Since this study procured an already existing shapefile delineating all blocks of buildings of Mumbai (see Chapter 3.2.2), the standard process of image segmentation could be skipped. Particularly for SAR data the segmentation process poses significant difficulties due to the partially very high backscatter values.

At first, GLCM texture measures were calculated, according to Chapter 2.3, using the software eCognition. For that reason, the shapefile of the blocks of buildings had to be imported in eCognition and converted to image objects in order to be able to use them for later texture calculation. For this purpose, the chessboard segmentation function was used. In this case, chessboard segmentation was used to convert the objects of the shapefile (lines) in vector format to raster format (areas) on pixel level using the pixel size of the radar imagery. The result of the chessboard segmentation totals to 16,122 image objects. One example can be seen in Figure 4-2. The turquoise line represents the original shapefile borders, whereas the dark blue line shows the contour of the image objects, created by chessboard segmentation. In the background of the images the Radar data (Kennaugh-element K_1) is displayed.

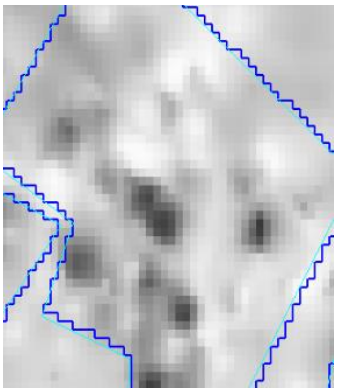


Figure 4-2: Result of the chessboard segmentation (dark blue line)

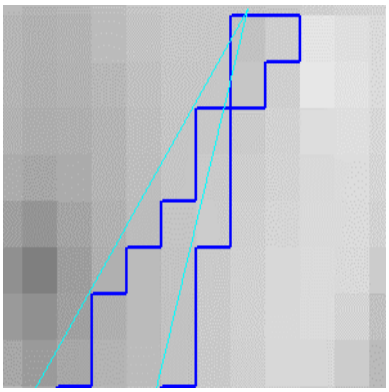


Figure 4-3: Artefacts occurring as a result of chessboard segmentation

One problem that occurred during the process of rasterizing the data were artifacts of just a few single pixels like in Figure 4-3, especially for pointed blocks. Since it does not make much sense to calculate texture measures for very small objects containing just a few single pixels, this problem was alleviated using a minimum area criterion during the classification process.

In a next step of the ruleset developed in eCognition, all image objects were attributed according to the thematic layer shapefile in formal and informal settlements, since that information got lost during segmentation but is needed for later training and accuracy assessment of the classifier. Additionally, only areas with a minimum size of 6 pixels were used in this step (see Figure 4-4).

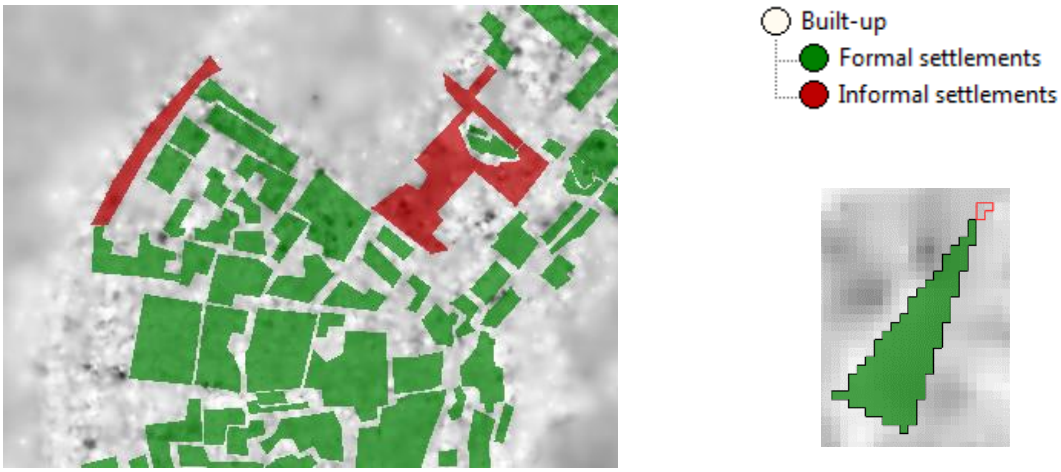


Figure 4-4: Result of the attribution of image objects using eCognition

Finally, for the Kennaugh elements K_0, K_1, K_5 and K_8 and all five deduced Kennaugh-elements of both polarizations (HH/HV and VV/VH), the GLCM measures according to Table 4-1 were automatically calculated by eCognition, as described in Chapter 2.3.

Table 4-1: Abbreviations for combinations of polarization, Kennaugh-element and GLCM measure

	Abbreviation	Meaning
Polarization:	05	HH/HV
	10	VV/VH
Kennaugh-elements:	K0	K0
	K1	K1
	K5	K5
	K8	K8
Deduced elements:	Abs	Absorption
	Dia	Diattenuation
	Pol	Polarizance
	Ret	Retardance
	Sca	Scale
GLCM measures:	Ang	Angular Second Moment
	Con	Contrast
	Cor	Correlation
	Dis	Dissimilarity
	Ent	Entropy
	Hom	Homogeneity
	Mea	Mean
	Sta	Standard Deviation

The resulting features were named as follows: “Polarization_Kennaugh-element_GLCM-measure”, for example “05_K0_Hom”. The abbreviations are explained in Table 4-1.

A shapefile of the blocks of buildings with an attribute table, containing all 144 combinations (2 polarizations x 9 Kennaugh-elements x 8 GLCM measures) an ID column as well as a Class column, containing the classes informal settlement and formal settlement, was exported for further processing (see Figure 4-5).

GLCM

FID	Shape	ID	Class	05_Abs_Ang	05_Abs_Con	05_Abs_Cor	…	10_Sca_Mea	10_Sca_Sta
0	Polygon	375	Formal settlements	0,069073	742,820942	0,884404		125,337284	42,695158
1	Polygon	978	Informal settlements	0,055782	402,692887	0,943075		123,346825	43,024341
2	Polygon	975	Informal settlements	0,126543	1598,468167	0,771372		127,512195	43,608024
3	Polygon	375	Formal settlements	0,064272	710,187582	0,888797		125,796626	42,744578
4	Polygon	376	Formal settlements	0,041041	787,564116	0,886646		125,30764	41,393001
5	Polygon	376	Formal settlements	0,051969	728,414273	0,894339		125,231993	42,10367
6	Polygon	356	Formal settlements	0,023096	871,430646	0,865698		124,979458	44,142208
7	Polygon	356	Formal settlements	0,08607	4124,376238	0,5433		127,683168	40,750613
8	Polygon	355	Formal settlements	0,050026	1210,039776	0,826044		127,456977	44,190357
9	Polygon	979	Informal settlements	0,033035	631,400825	0,908932		124,655858	41,597492
10	Polygon	974	Informal settlements	0,061537	991,318824	0,851714		126,37352	42,392221
…									
9398	Polygon	108	Formal settlements	0,04074	570,786026	0,917183		123,832457	45,570383
9399	Polygon	107	Formal settlements	0,316316	2700,061195	0,576865		126,047315	44,88159
9400	Polygon	109	Formal settlements	0,115038	1171,70053	0,832764		123,874298	44,10174

Figure 4-5: Subset of the attribute table containing all 144 texture measures calculated from the GLCM for all 9401 blocks of buildings

4.2.2 Feature Selection

Since, as previously pointed out in Chapter 2.3, some GLCM texture measures are correlated, a feature selection was conducted before the final classification. First of all, the .dbf-file of the shapefile, containing the attribute table with the GLCM texture measures (see Figure 4-5) was imported in an R script and a subset was selected, containing all texture measures except for Dissimilarity (2 polarizations x 9 Kennaugh-elements x 7 GLCM measures = 126 features), since it was found to be inversely correlated with Homogeneity (see Chapter 2.3).

For feature selection, a backwards selection approach was performed in a separate function in R that is called up in the script for classification. Feature selection was conducted based on the respective classification algorithms. The classification process is explained in more detail in Chapter 4.2.3. For feature selection, the classification is conducted for every feature, so 126 times in total, in order to calculate the accuracy values of the classification and the ranking criterion. After every of the 126 loop passes, the feature having the smallest ranking criterion is removed from the initial data set.

The criterion used for ranking is the Area Under Curve (AUC), which is, like TSS, independent of prevalence (Allouche et al., 2006). Curve in this case means a Receiver Operating Characteristic (ROC) curve. The AUC reveals the performance of the classifier using a particular feature. The ROC curve is a function of sensitivity (true-positive-rate) and false-positive-rate of the classification, both ranging from 0 to 1. ROC is commonly used for visualizing the performance of binary classifiers, meaning classifiers differentiating between two classes. The ROC curve usually runs in a bent and ascending direction. The curve is approaching the edges of the diagram, if the classifier is separating the two classes well and is approaching the diagonal, if the classifier separates the classes poorly (see Figure 4-6).

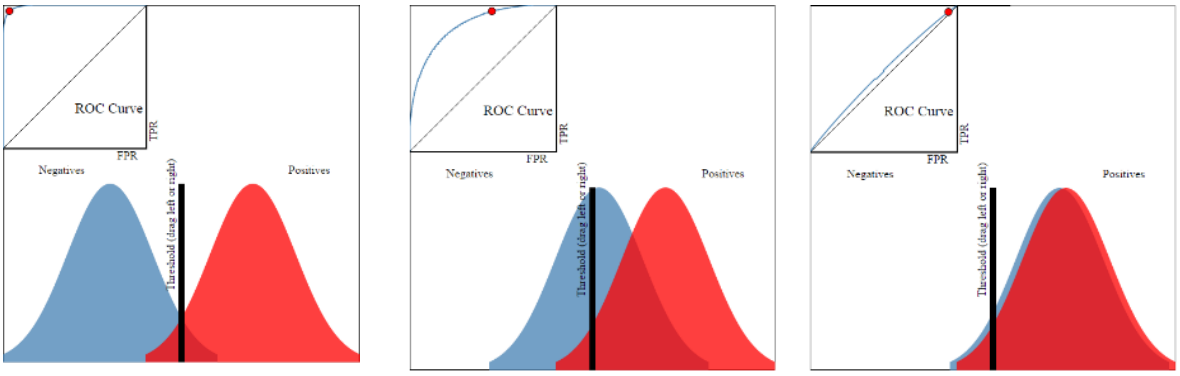


Figure 4-6: ROC curves for a very good (left), moderate (middle) and a very bad (right) classifier

Source: Navan (2014)

In order to get a quantification of the visualization using ROC, the AUC is calculated:

```
auc = colAUC(sample[,3:ncol(sample)], as.factor(sample[,1]), plotROC=FALSE,
             alg="ROC")
```

The AUC is the portion of the area of the unit square under the curve. Therefore its value ranges between 0 and 1. Since the diagonal is basically equal to the ROC curve produced by random guessing, an AUC value of 0.5 reveals very bad performance of a classifier, whereas a value of 1 reveals a perfect classifier.

Based on this ranking criterion a ranking of all features for every classifier was produced (see Chapter 5.1 for the results). For final classification, a subset including the thirty best features of the ranking was tested in comparison to using all features. Empirical tests for 10, 20, 30, …, 100 features were conducted using LDA (Figure 4-7), to show the relation of increasing number of features and accuracy enhancement. The accuracy assessment values in tabular format can be found in Appendix V: Accuracy Assessment Results, Table E-1.

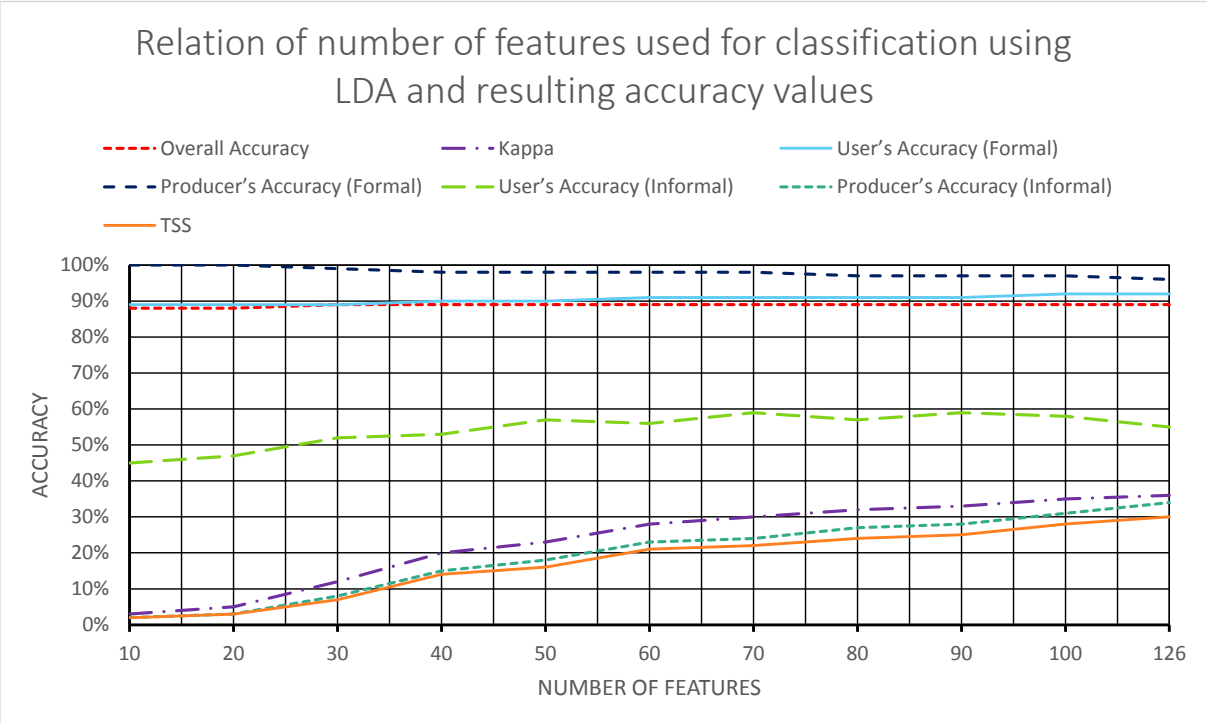


Figure 4-7: Line diagram showing increasing number of features versus accuracy values of the classification using Linear Discriminant Analysis

As Figure 4-7 shows, the accuracy increases with the number of features. The best results using LDA were achieved using all texture features. Therefore, the results shown in Chapter 5 were based on all features, but can be compared to the results using only 30 features in Appendix IV: Classification results using the 30 best Features. The scripts for classification and feature selection exemplary for RF can be found in Appendix II: R Script for Classification using Random Forest and Appendix III: R Script for Feature Selection using Random Forest. The complete scripts for all three classifiers can be found digitally on the enclosed CD.

4.2.3 Classification

For classification, the two classes informal settlements and formal settlements were distinguished, using the characteristics shown in Table 4-2, since the focus of this study is on informal settlements.

Table 4-2: Characteristics of classes used for classification

Class	Characteristics	Example
Informal Settlement	- blocks of buildings having a low height and high density (hardly any distance between buildings) - streets visible between houses - unstructured but having a compact structure	
Formal Settlement	- blocks of buildings having low to high building heights and low to high densities - structured	

The classifications using the algorithms LDA, SVM and RF, which are explained in Chapter 2.5, were all implemented using R. First, a subset of the attribute table was generated in R containing all GLCM features except Dissimilarity.

For classification, a random sample of 20% of objects for each class was used as training input for the models. For drawing of the samples, the function "sample.by.var"[1] was used:

```
sample = sample.by.var(dataset, var="Class", probs=0.2)
```

Based on the sample, the models for LDA, RF and SVM, respectively, were trained given the texture measures and according class membership of the sample objects, e.g. for RF:

```
rf.model = randomForest(sample[,3:ncol(sample)], as.factor(sample[,1]),
                        importance=TRUE, ntree=1000)
```

[1] Provided by Dr. Michael Wurm

Then, the trained model was used to predict the class membership of all objects or blocks of buildings that are not contained in the sample dataset:

```
prediction.class = predict(rf.model, other.features, type = "class")
```

The probability, with which the object belongs to the certain class was calculated using the function:

```
pred.prob = predict(rf.model, other.features, type = "prob")
```

Finally, a .dbf-file containing the ID, predicted class membership and probabability of all objects was exported and joined with the original shapefile that contains the true class values in ArcGIS, in order to visualize the classification results (see Chapter 5.2).

4.2.3.1 Classification using Linear Discriminant Analysis

For LDA the function LinDA (Faes, 2014) was used:

```
lda.model = linda.lda(sample, erweitert=TRUE, gruppenP=TRUE)
```

The function LinDA calculates the discriminant functions with the appropriate discriminant coefficients as pointed out in Chapter 2.5.1. Using the created model, the class membership and probability values were calculated similar to RF:

```
prediction = predict(lda$lda.funktion, other.features)
other$Group = prediction$class
other$Prob = round(apply(prediction$posterior, 1, max),2)
```

Using LDA, no parameters had to be tuned.

4.2.3.2 Classification using Support Vector Machine

For SVM the package “e1071”, an interface of the robust, fast and state-of-the-art LIBSVM implementation designed in C++ (Karatzoglou et al., 2006) , was used. Other possibilities for SVM implementations are the packages “kernlab”, “klaR” and “svmpath”. Karatzoglou et al. (2006) tested all four packages and came to the conclusion that the package “e1071” is a robust interface with the shortest training time of all four packages. Their findings led to the decision of using this package in this study. For the classification, a linear Kernel was used. The parameter cost was tuned using the grid search of the tune function of package “e1071” over a supplied range (0.25 – 4):

```
svm.tune = tune(svm, train.x = sample[,3:ncol(sample)],
                train.y = as.factor(sample[,1]),
                validation.x=sample[,3:ncol(sample)],
                validation.y=as.factor(sample[,1]),
                ranges=list(cost = 2^(-2:2)),
                kernel = "linear")

svm.model = svm(sample[,3:ncol(sample)], as.factor(sample[,1]),
                type="C-classification",
                kernel="linear",
                cost = svm.tune$best.parameters$cost,
                cachesize=500,
                probability=TRUE)
```

As suggested by Tso & Mather (2009), exponentially growing sequences were used. Default value of the cost of constraints violation is 1. Using a grid search better results could be achieved than by using the default values.

4.2.3.3 Classification using Random Forest

For RF there is only one package, called *randomForest*, available for R, that was originally implemented by Leo Breiman and Adele Cutler in Fortran and is now maintained in R by Andy Liaw and Matthew Wiener.

The number of variables randomly sampled as candidates at each split were used as tuning parameter, since Hastie et al. (2009) recommend to treat them as a tuning parameter, because their best values depends on the problem:

```
rf.model_tuning = tuneRF(sample[,3:ncol(sample)], as.factor(sample[,1]),
                         ntreeTry=1000, stepFactor=2, trace=TRUE,
                         plot=TRUE, doBest=FALSE)
```

The default value is the square root of the number of variables (126), in this case approximately 11. Using a grid search implemented in the tuneRF function of the *randomForest* package, no improvements of the out-of-bag (OOB) error estimate could be reached, so the number of variables randomly sampled at each split were left at their default value of 11 for classification (see Figure 4-8). The OOB error estimate is calculated by testing each tree of the RF on the samples that were not used in building the trees.

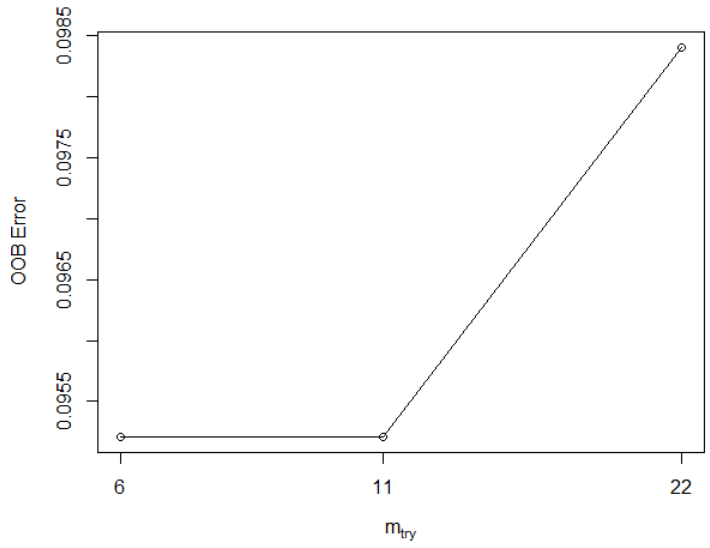


Figure 4-8: Result of the tuning process of the RF model: No better out-of-bag (OOB) error rates could be reached

For RF, each forest was set to consist of 125 individual trees for classification, since this number was found to be a good compromise between runtime elongation and precision, (see Figure 4-9):

```
rf.model = randomForest(sample[,3:ncol(sample)], as.factor(sample[,1]),
                        importance=TRUE, ntree=1000)
```

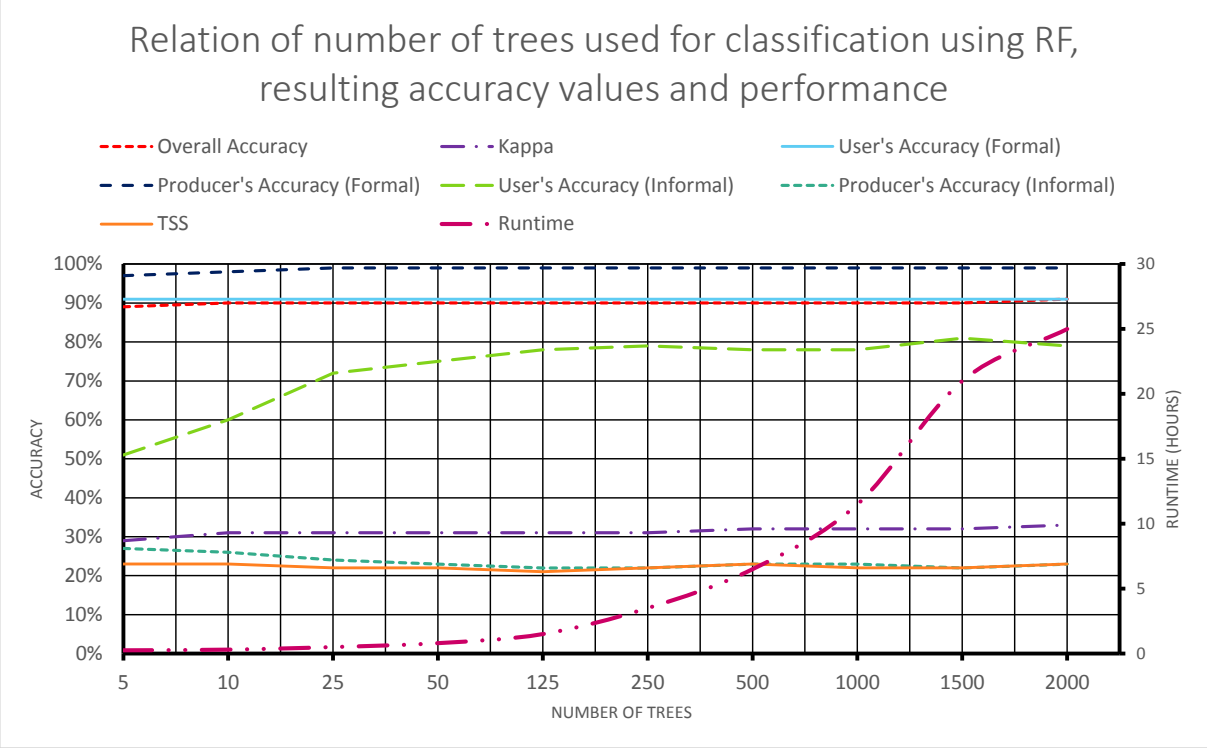


Figure 4-9: Line diagram showing the number of trees versus accuracy and runtime values of the classification using Random Forest

The accuracy and runtime values can also be found in tabular format in Appendix V: Accuracy Assessment Results, Table E-2.

4.2.4 Accuracy Assessment

In this work, since the outlining of slum areas has been done visually in advance, the accuracy assessment greatly helps with comparison of the three different classification algorithms LDA, SVM and RF, in order to find out which one works best for this type of data and environment. The accuracy measures used were Overall accuracy, User's accuracy, Producer's accuracy, Kappa and TSS. A detailed explanation is given in Chapter 2.6. For the implementation in R, except for the TSS an already developed function called kappa was used, derived from Rossiter (2004).

For calculation of the TSS, the elements of the confusion matrix were derived in order to calculate sensitivity and specifity:

```
confusionMatrix = as.matrix(error.matrix)
a = confusionMatrix[1,1]
b = confusionMatrix[1,2]
c = confusionMatrix[2,1]
d = confusionMatrix[2,2]
sensitivity = a/(a+c)
specificity = d/(b+d)
tss = sensitivity + specificity - 1
```

5 Results and Discussion

In this Chapter, the results of feature analysis, classification and accuracy assessment will be presented and further discussed.

5.1 Results of the Feature Analysis

Table 5-1: Feature ranking of the 30 best features and respective AUC values for LDA, RF and SVM using AUC

	LDA		RF		SVM	
	Feature	**AUC**	**Feature**	**AUC**	**Feature**	**AUC**
1	X10_Ret_Ang	-	X10_Ret_Ang	-	X10_Ret_Hom	-
2	X10_Ret_Hom	0.72	X10_Ret_Hom	0.71	X05_Ret_Hom	0.71
3	X10_Ret_Ent	0.71	X05_Ret_Hom	0.72	X10_Ret_Ang	0.72
4	X05_Ret_Hom	0.71	X10_Ret_Ent	0.72	X10_Ret_Ent	0.72
5	X05_K0_Hom	0.70	X05_K0_Hom	0.70	X05_K0_Hom	0.70
6	X10_Pol_Cor	0.70	X10_Pol_Cor	0.69	X05_Ret_Ang	0.70
7	X05_Pol_Cor	0.69	X10_Pol_Con	0.69	X05_Pol_Cor	0.70
8	X05_Ret_Ang	0.70	X05_Ret_Ang	0.70	X10_Pol_Cor	0.70
9	X05_K0_Ang	0.67	X05_Ret_Ent	0.71	X10_Pol_Con	0.70
10	X10_Pol_Con	0.68	X05_Pol_Cor	0.70	X05_Ret_Ent	0.69
11	X05_Ret_Ent	0.68	X05_K0_Ang	0.67	X05_K5_Hom	0.71
12	X05_Pol_Con	0.67	X05_Dia_Hom	0.67	X05_K0_Ang	0.67
13	X05_Dia_Hom	0.67	X05_K5_Hom	0.67	X05_Dia_Hom	0.66
14	X05_K8_Hom	0.66	X05_K8_Hom	0.67	X05_K8_Hom	0.66
15	X05_K5_Hom	0.67	X05_Pol_Con	0.68	X05_Pol_Con	0.67
16	X05_Abs_Cor	0.66	X05_Sca_Mea	0.67	X05_Abs_Cor	0.66
17	X05_K0_Ent	0.66	X05_Abs_Cor	0.65	X05_Sca_Mea	0.67
18	X05_Abs_Con	0.65	X10_Abs_Cor	0.65	X10_Dia_Hom	0.67
19	X10_Abs_Cor	0.66	X05_K0_Ent	0.66	X05_K0_Ent	0.66
20	X05_Sca_Mea	0.66	X10_Dia_Hom	0.67	X10_Abs_Cor	0.66
21	X05_Sca_Cor	0.65	X05_K0_Mea	0.68	X05_Abs_Con	0.66
22	X10_Dia_Hom	0.66	X10_Abs_Con	0.67	X10_Abs_Con	0.65
23	X10_K1_Hom	0.66	X05_Abs_Con	0.65	X05_Sca_Cor	0.66
24	X05_K0_Mea	0.66	X05_Sca_Cor	0.65	X05_K0_Mea	0.66
25	X10_Abs_Con	0.65	X10_K1_Hom	0.67	X10_Sca_Ent	0.65
26	X10_Sca_Ent	0.66	X10_Abs_Hom	0.66	X05_Sca_Con	0.66
27	X05_Sca_Ent	0.66	X05_Sca_Con	0.66	X05_Sca_Ent	0.67
28	X05_K8_Ang	0.65	X05_Sca_Ent	0.66	X10_K1_Hom	0.67
29	X05_K8_Ent	0.65	X10_Dia_Ang	0.66	X05_K8_Ang	0.67
30	X10_Abs_Hom	0.65	X05_K8_Ang	0.66	X10_Abs_Hom	0.65

The results of the feature ranking using the AUC criterion are very similar for all three classifiers. The feature analysis revealed a slighty higher occurrence of HH/HV than VV/VH polarization in the top 30 feature ranking (see Table 5-1). The feature ranked highest is of HH/HV polarization for all three classifiers. The Kennaugh-elements Retardance, Absorption and Scale were the most frequently appearing in the top 30 features, in descending order. Retardance, K0 and Polarizance were the most influential Kennaugh-elements for informal settlement classification, since they revealed the highest AUC values.

The GLCM measure Homogeneity turned out to appear the most often in all three rankings, followed by Entropy, Correlation, Contrast and ASM. The texture measure Mean contributed the least to the classification, while Standard Deviation does not appear in the top 30 feature ranking and therefore seems to be inappropriate for informal settlement classification using Radarsat-2 data.

The combinations of VV/VH polarization, Retardance and ASM (X10_Ret_Ang) and VV/VH polarization, Retardance and Homogeneity (X10_Ret_Hom) reached one of the first three scores in all rankings and therefore contributed most to the results for informal settlement classification, using LDA, SVM, as well as RF.

5.2 Results of the Classification

In this Chapter, the results of the classification using Linear Discriminant Analysis, Random Forest and Support Vector Machines will be shown and further discussed. Since using all features instead of the 30 best features revealed better results, only these results will be discussed in the following in more detail. For the classification results using only the 30 best features see Appendix IV: Classification results using the 30 best Features.

Blocks of buildings that were used for training of the algorithm were not further used in the classification process, in order to not unintentionally falsify the accuracy assessment results. These areas are displayed in gray on the following classification maps (Figure 5-1 to Figure 5-3). The blocks of buildings that were classified as formal settlement are displayed in yellow and those classified as informal settlement are shown in violet. As can be seen on the resulting classification maps, SVM classified less blocks of buildings as informal settlements than LDA and RF.

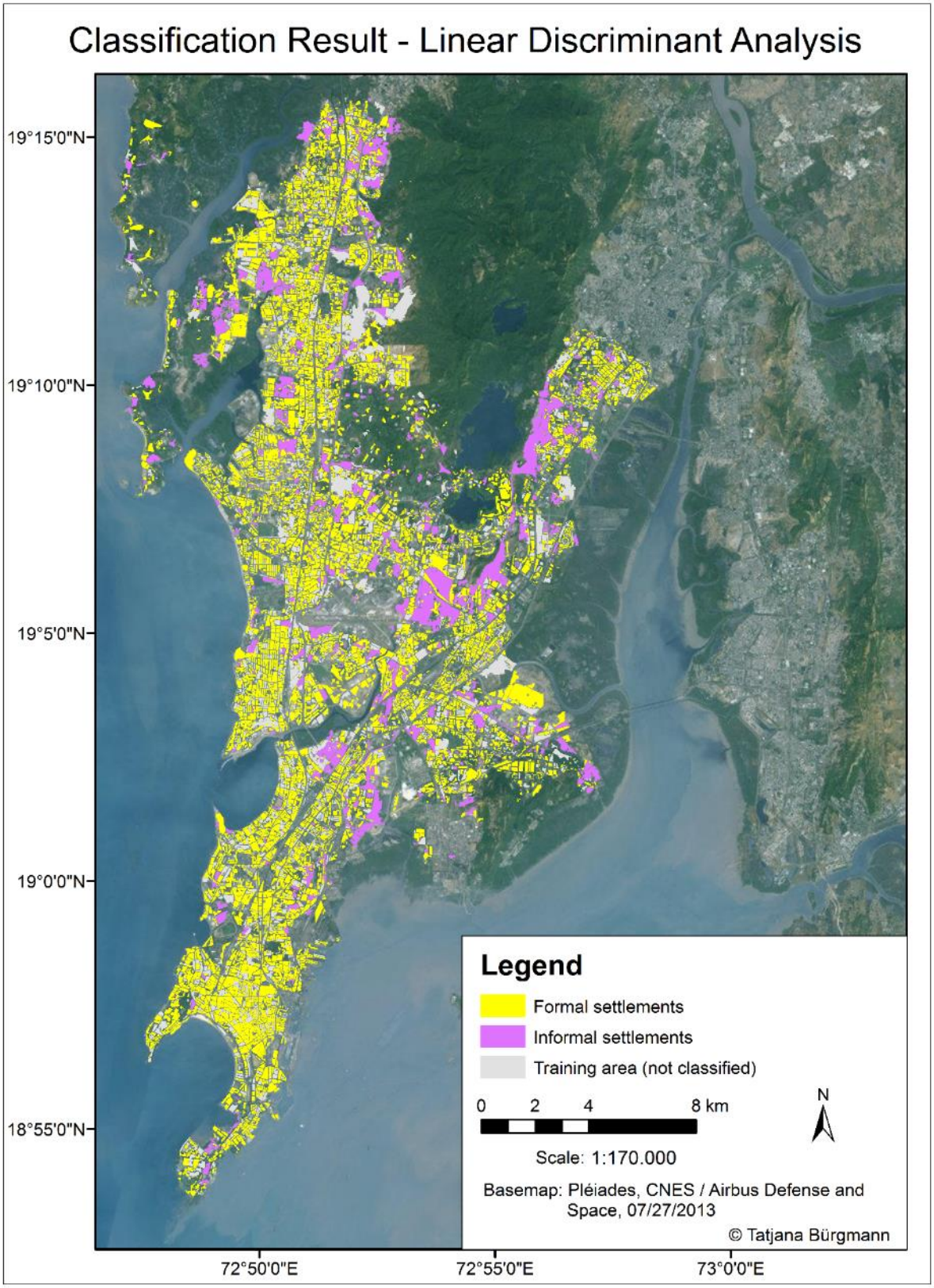


Figure 5-1: Result of the classification using Linear Discriminant Analysis

The statistical analysis reveals

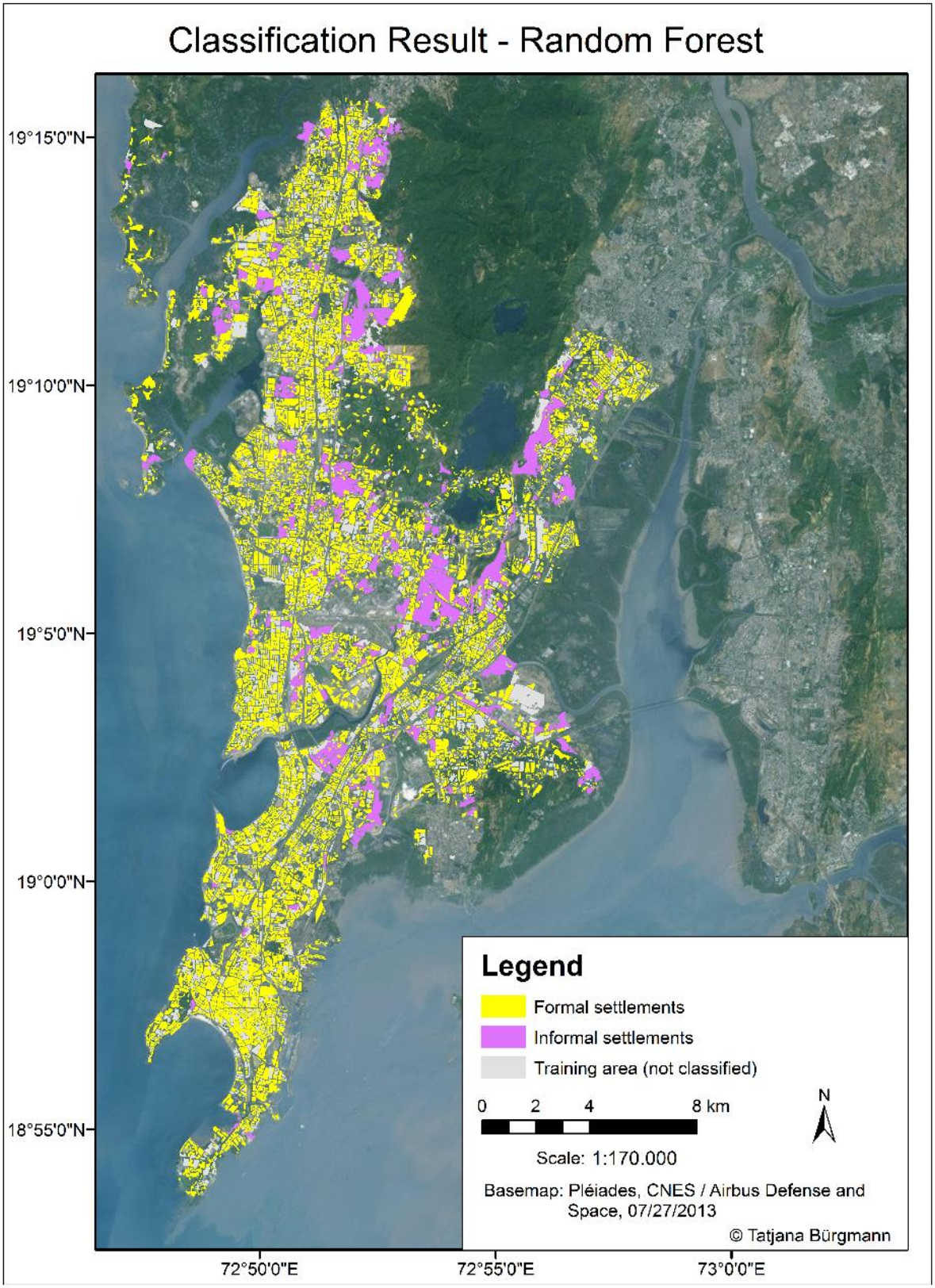


Figure 5-2: Result of the classification using Random Forest

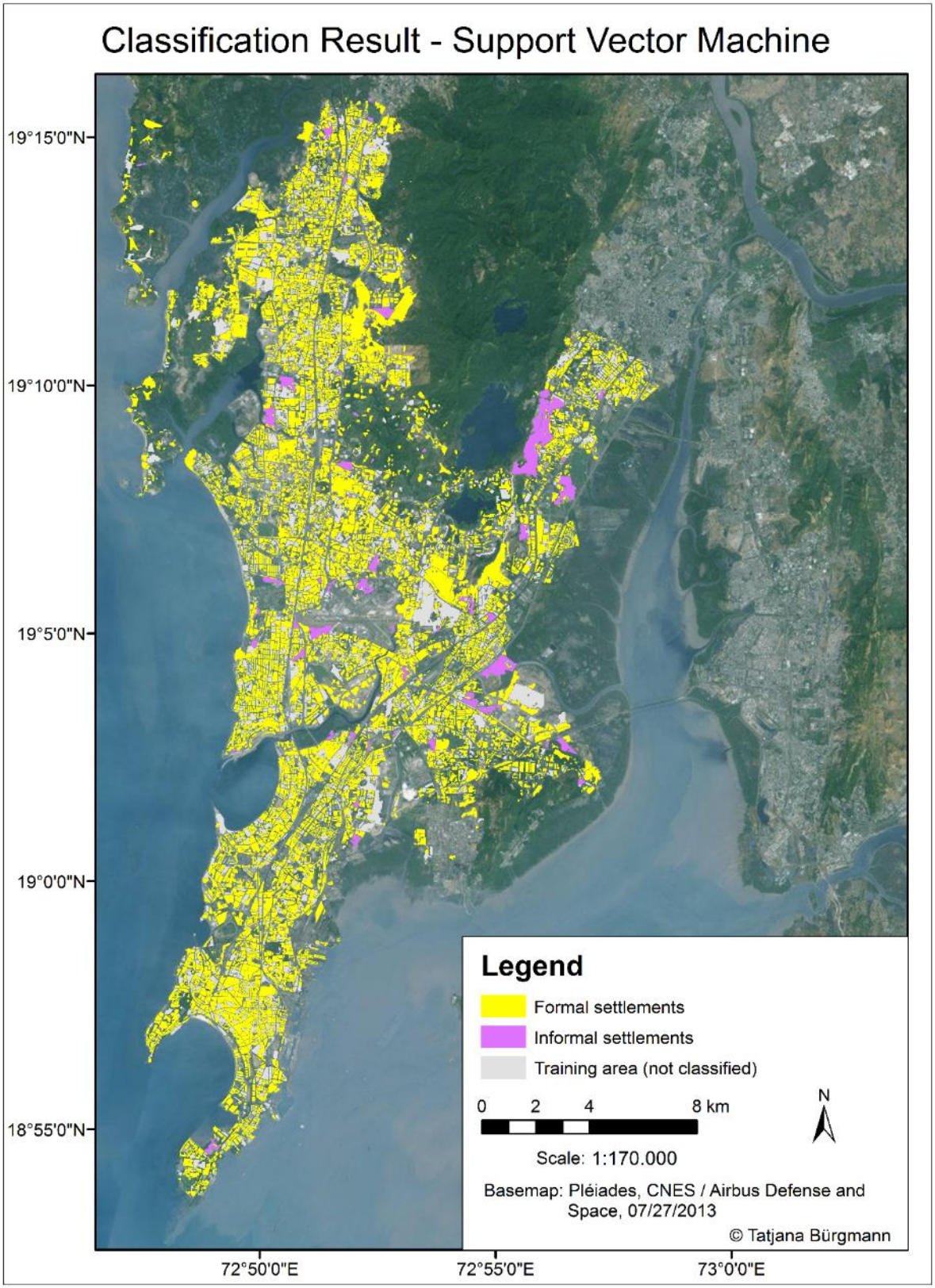


Figure 5-3: Result of the classification using Support Vector Machines

Since the classification maps on their own do not provide much information when comparing the performance of different classifiers, validation maps were created, displaying not only the classified class, but also giving additional information on the correctness of the classification. As for the classification maps, training areas are displayed in gray. Blocks of buildings classified correctly as informal settlements are represented in green and those blocks that were classified incorrectly are marked in red. For formal settlements the color for correctly classified blocks is turquoise and orange for incorrectly classified blocks, respectively.

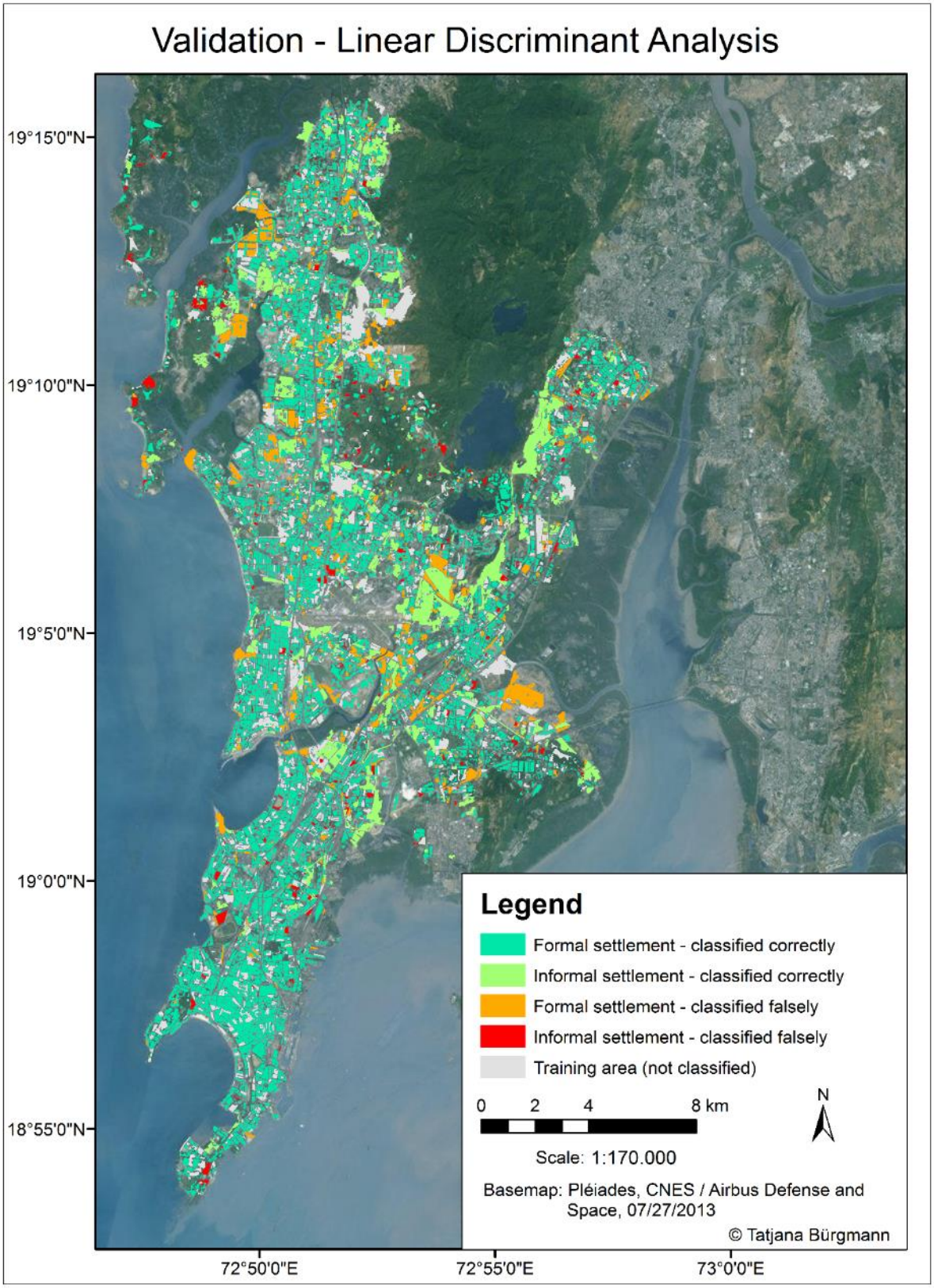


Figure 5-4: Validation of the classification result using Linear Discriminant Analysis

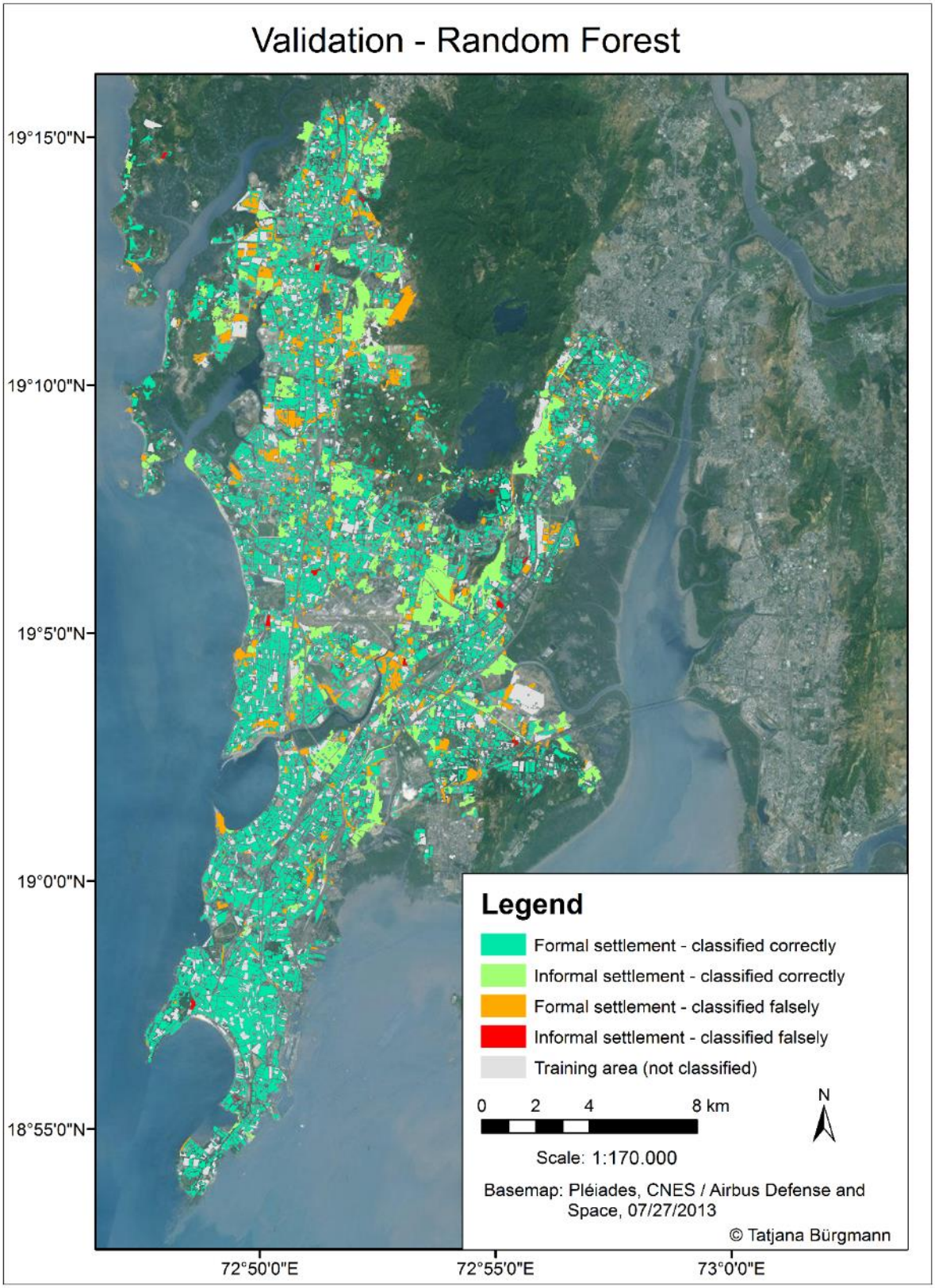


Figure 5-5: Validation of the classification result using Random Forest

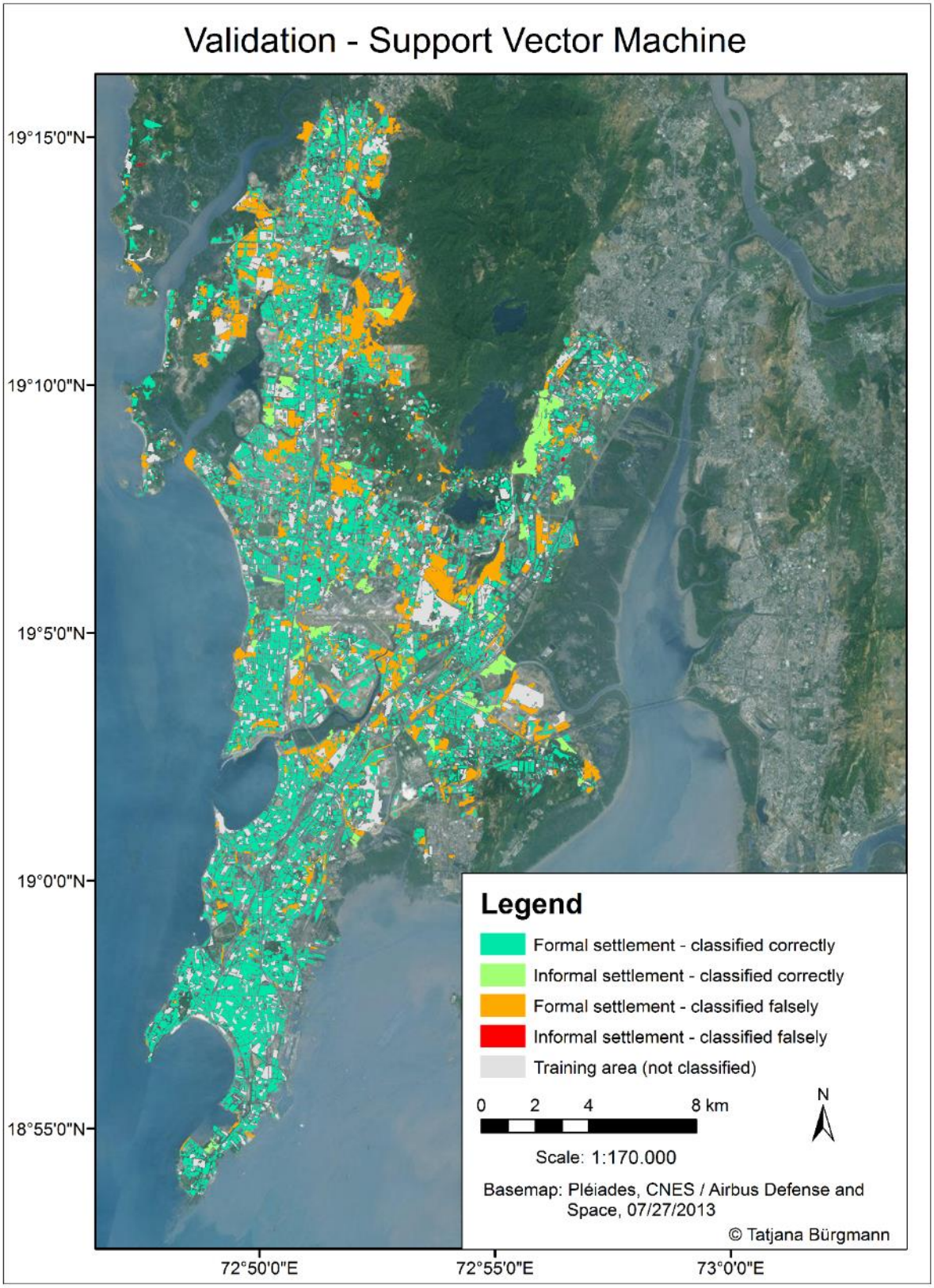


Figure 5-6: Validation of the classification result using Support Vector Machine

Comparing all three maps visually, the classification using Random Forest shows the best results for informal settlement classification across the whole Mumbai area, using Radarsat imagery. Only very small and few formal settlements were misclassified as informal settlement (colored in red). A few more informal settlement blocks were misclassified as formal settlement (colored in orange).

Linear Discriminant Analysis shows the second best results. Using this classifier, a few more informal settlements were not classified as such.

Support Vector Machines strongly underestimated slum areas in this study. Even larger blocks of buildings of slums were classified incorrectly as formal settlement. On the other hand, very few informal settlements were classified incorrectly as formal.

In order to compare the results of the three classification algorithms, the following maps (Figure 5-7 to Figure 5-8) show all blocks of buildings despite of informal or formal settlement that were classified correctly in green and those that were classified incorrectly in red for a subset of the Suburban District of Mumbai, since the Island City comprises only few informal settlements.

Figure 5-10 to Figure 5-12 show only informal settlements that were classified correctly (green) and incorrectly (red). Figure 5-13 to Figure 5-15 basically present the same for formal settlements.

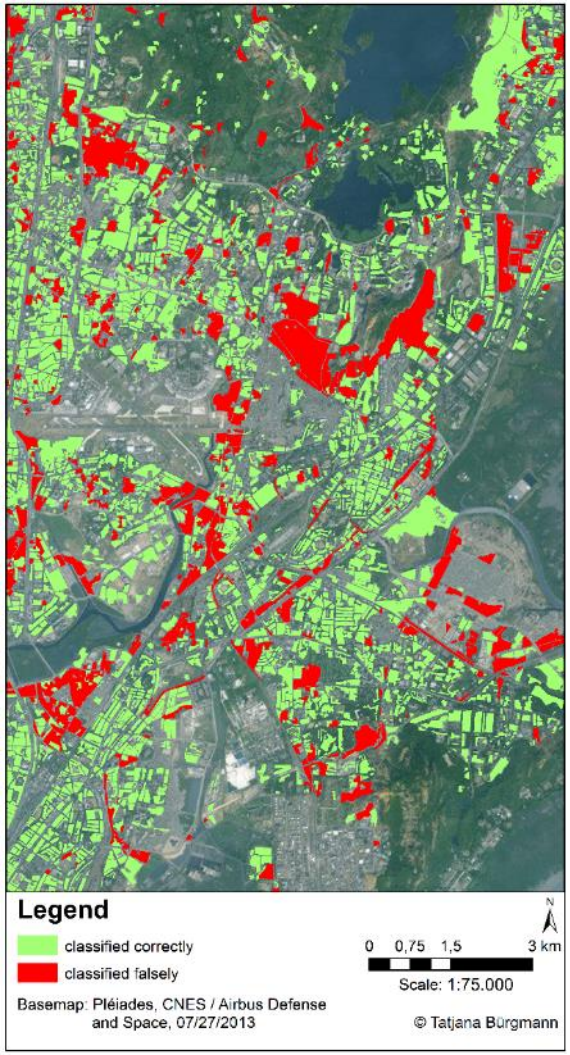


Figure 5-9: Validation of correctly and incorrectly classified blocks using SVM:

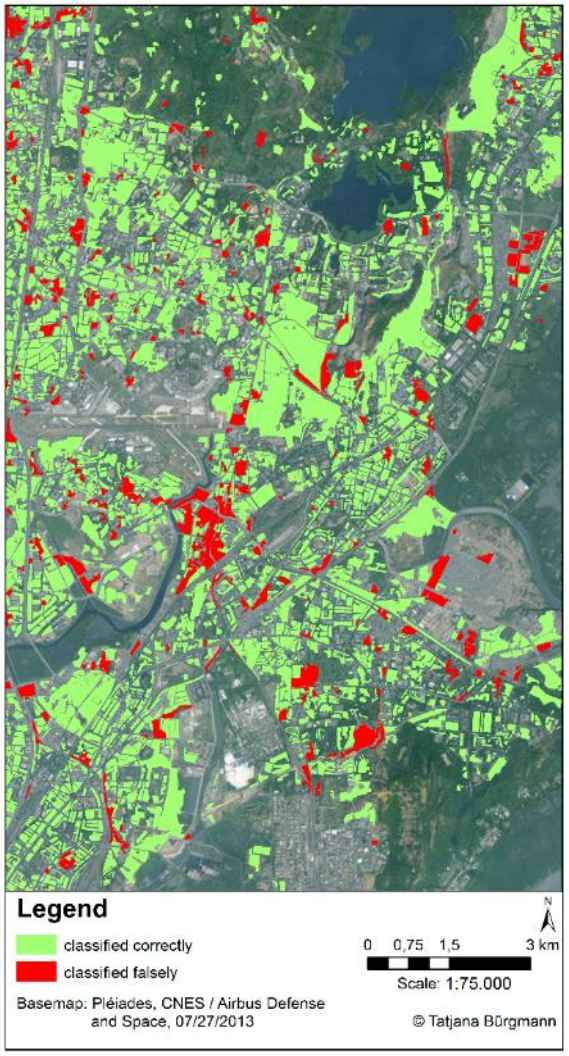


Figure 5-8: Validation of correctly and incorrectly classified blocks using RF

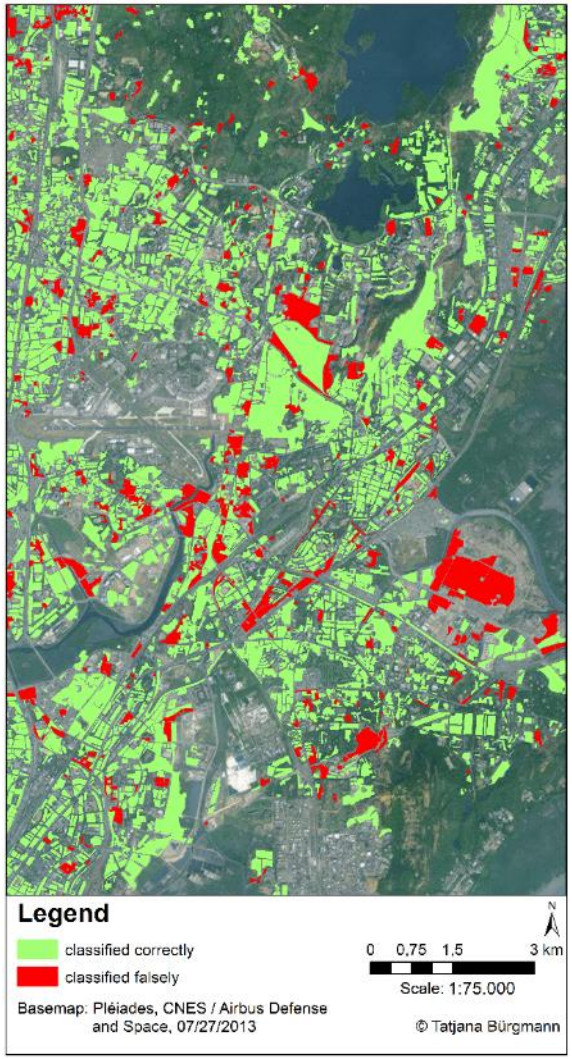


Figure 5-7: Validation of correctly and incorrectly classified blocks using LDA

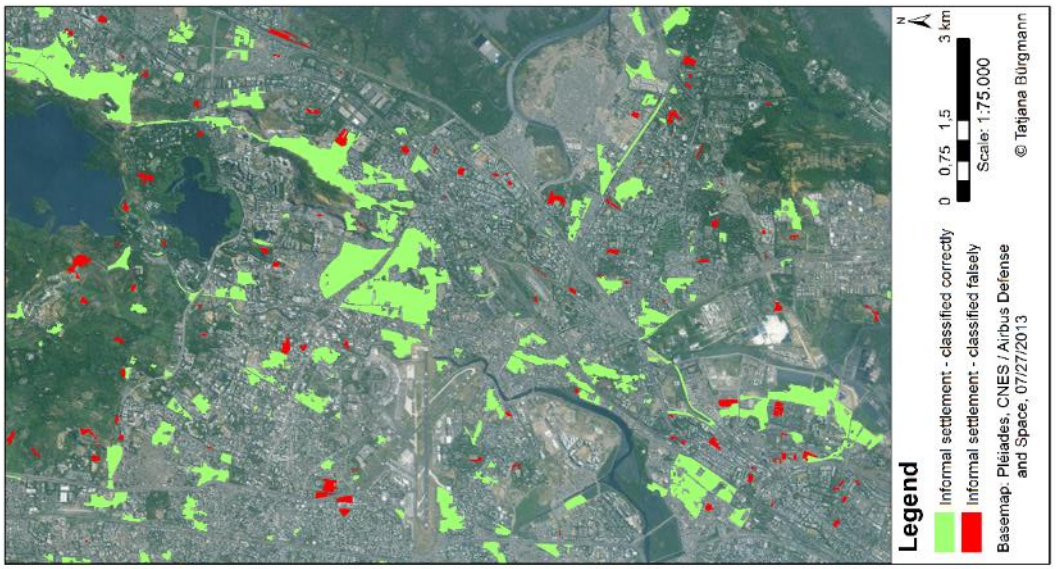


Figure 5-10: Validation of Informal Settlements using LDA

Figure 5-11: Validation of Informal Settlements using RF

Figure 5-12: Validation of Informal Settlements using SVM:

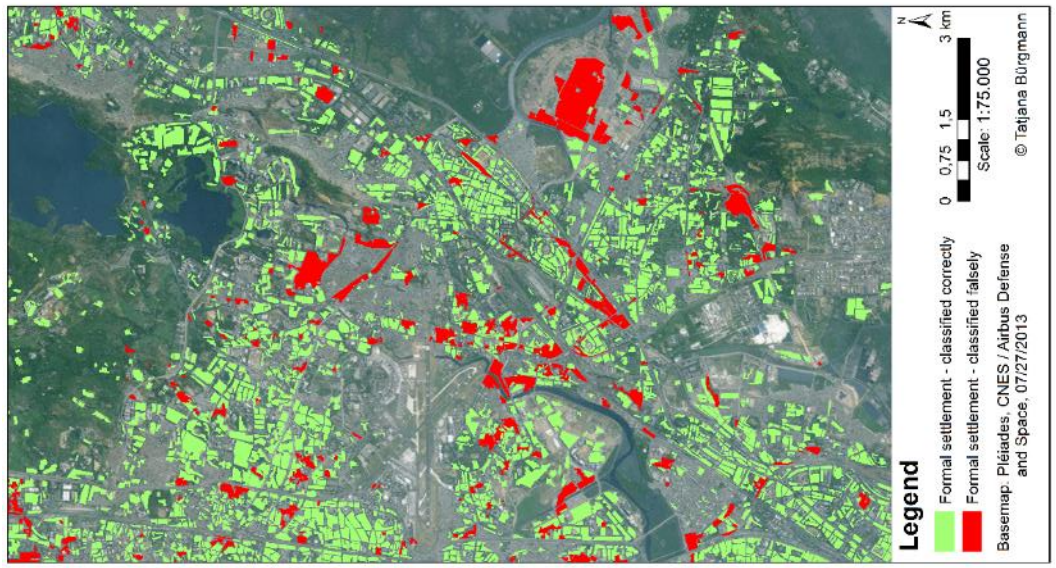


Figure 5-13: Validation of Formal Settlements using LDA

Figure 5-14: Validation of Formal Settlements using RF

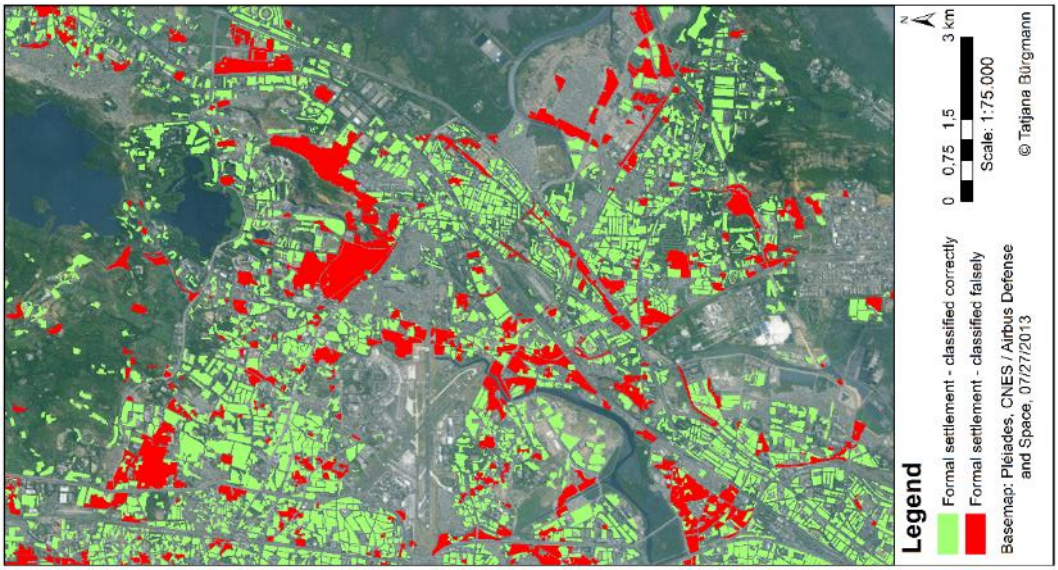


Figure 5-15: Validation of Formal Settlements using SVM:

As can be seen on the maps of Figure 5-7 to Figure 5-8, SVM strongly underestimated informal settlements and therefore revealed the worst results of all three classifiers, while RF could detect more informal settlements than LDA

The maps in Figure 5-10 to Figure 5-12 show that the informal settlements classified as such are very likely to be informal settlements in ground truth. Only very small informal settlements have been misclassified as formal settlements.

Figure 5-13 to Figure 5-15 show that informal settlements were underestimated, since many formal settlements were misclassified as formal but are actually informal settlements. These maps also show that using SVM, informal settlements were underestimated the most, while using RF the less formal settlements were misclassified of all three classifiers.

The results using only the 30 best features strengthen these assumptions. Summarizing, RF showed the best results, before LDA and SVM, but all three classifiers underestimated informal settlements to a certain degree.

In the following Chapter, the accuracy of the classifications is represented quantitatively.

5.3 Results of the Accuracy Assessment

Using Random Forest as classification algorithm, an overall accuracy of 91% was achieved. Support Vector Machines and Linear Discriminant Analysis revealed a slightly lower overall accuracy value of 89%. All three classifiers achieved very high Producer's as well as User's accuracies for formal settlements. For informal settlements the highest Producer's accuracy (91%) was reached using RF, followed by SVM with 61% and LDA with 55%. The highest User's Accuracy, however, was reached by LDA with 34%, followed by RF with 21% and SVM with only 9%. Kappa values are 0.36 for LDA, 0.31 for RF and 0.13 (due to the low user accuracy for informal settlements) for SVM. TSS accuracy values are accordingly 0.21 for RF, 0.3 for LDA and 0.09 for SVM. To sum up, RF achieved the best accuracy values of all three classifiers for informal settlement classification according to the Overall Accuracy. However, the User's Accuracy of LDA is higher than that of RF, what is reflected in the Kappa and TSS values. Deviations may originate from sensor limitations, methods used in the analysis or landscape complexity (Chuvieco & Huete, 2010). In this case, sensor limitations in sense of spatial resolution and the complex urban environment might be hold responsible for introducing some major deviations in the classification results.

Table 5-2: Results of the Accuracy Assessment and performance of LDA, RF and SVM

	LDA	RF	SVM
Overall	89%	91%	89%
Kappa	0.36	0.31	0.13
User's (Informal)	55%	91%	64%
Producer's (Informal)	34%	21%	9%
User's (Formal)	92%	91%	89%
Producer's (Formal)	96%	100%	99%
TSS	0.3	0.21	0.09
Runtime	36 s	10 min	2 min

All three classifiers showed considerable variations regarding the runtime: The complete feature selection and classification using LDA took only 36 seconds, while SVM needed about 2 minutes and RF about 10 minutes. As a final conclusion, it can be stated that RF achieved the best accuracies, while also consuming the longest runtime. LDA achieved the second best results while needing only a very small portion of the runtime needed for RF. SVM achieved the lowest accuracy results while taking up only about a fifth of the runtime needed for RF. Compared to other studies, e.g. (Weigand, 2014) high Overall Accuracy values could be achieved but Producer's Accuracies for informal settlements were rather low, since these areas were underestimated by the classifiers. For the results of the accuracy assessment using the 30 best features, see Appendix IV: Classification results using the 30 best Features.

6 Conclusion and Future Research

In this work, formal and informal settlements were classified using Radarsat-2 imagery. Using the example of the mega city of Mumbai, all blocks of buildings were classified and later verified in an accuracy assessment. Two images were used, one with HH/HV and the other with VV/VH polarization. Each image is described by the four Kennaugh-elements K_0, K_1, K_5, K_8 and the deduced Kennaugh-elements that are: Absorption, Diattenuation, Polarizance, Retardance and Scale. The Gray Level Co-occurrence Matrix texture measures Homogeneity, Contrast, Entropy, Angular Second Moment, Mean, Standard Deviation and Correlation have been calculated and likewise used for classification. Three different object-based classification algorithms have been tested: Linear Discriminant Analysis, Support Vector Machine and Random Forest. Using a selective backward feature selection algorithm and the Area Under Curve measure, the best combinations of polarization, Kennaugh-elements and Gray Level Co-occurrence Matrix texture measures were revealed. However, all 126 features were used for final classification since better results could be obtained than with using only the 30 best features. For validation of the classification, an accuracy assessment was conducted, in order to reveal the usefulness of the classification results and subsequently assess the performance of the respective classifiers. For this purpose, an available reference GIS data layer was used, containing all blocks of buildings of Mumbai with appropriate attribution of formal and informal settlements, derived from optical Very High Resolution satellite data. This data layer was also used in order to create image objects for the calculation of texture measures.

The feature selection revealed that a combination of both polarization modes, HH/HV and VV/VH, works best for informal settlement classification. The Kennaugh-elements Polarizance, K_8, K_1 and Absorption were the most useful in informal settlement detection. The GLCM texture measures Contrast, Correlation, Entropy and Angular Second Moment achieved the best results. The combination of HH/HV polarization, Polarizance and Contrast turned out to be one of the best features for slum classification using Radarsat-2 imagery for all three classifiers. Further, the combinations of VV/VH polarization, Polarizance and Contrast, as well as the combinations of HH/HV polarization, Polarizance and Correlation and VV/VH polarization, Kennaugh-element K_8 and Entropy contributed most to the classification result. However, the Area Under Curve values used as ranking criterion for feature selection showed only small differences for all features.

This work has shown, that a few Kennaugh-elements and Gray Level Co-occurrence Matrix measures work slightly better for informal settlement classification using Radarsat-2 imagery, than others. This study also revealed, that the Random Forest classifier worked best for this kind of data in this specific environment of the study area of whole Mumbai. The high accuracies achieved have a negative impact on the runtime that is rather long, compared to the other two classifiers. Linear Discriminant Analysis seems to be a good trade-off, if the runtime is an important criterion for algorithm selection. However, all three classifiers showed some limitations using rather coarse Radarsat imagery in order to detect very small-scale building structures like informal settlements. Therefore in future research, a combination of Very High Resolution optical satellite data, combined with Radarsat-2 data should be tested, in order to achieve better classification results with higher accuracies.

References

Adhikari, S. (2004). Urban Planning and Politics of Slum Demolition in Metropolitan Mumbai. In *Proceedings of the 9th Asian Real Estate Society (AsRES) International Conference*. New Delhi. Retrieved from www.asres.org/2004Conference/papers/Adhikari.doc

Agarwal, S. (2011). The state of urban health in India; comparing the poorest quartile to the rest of the urban population in selected states and cities. *Environment and Urbanization*, *23*(1), 13–28. http://doi.org/10.1177/0956247811398589

Albertz, J. (2009). *Einführung in die Fernerkundung: Grundlagen der Interpretation von Luft- und Satellitenbildern* (4th ed.). Darmstadt: Wissenschaftliche Buchgesellschaft.

Allouche, O., Tsoar, A., & Kadmon, R. (2006). Assessing the accuracy of species distribution models: Prevalence, kappa and the true skill statistic (TSS). *Journal of Applied Ecology*, *43*(6), 1223–1232. http://doi.org/10.1111/j.1365-2664.2006.01214.x

Arimah, B. C. (2010). *The Face of Urban Poverty: Explaining the Prevalence of Slums in Developing Countries*.

Asha, K. (2006). Urban Slums in India-the myths and the reality. *Asha-Seattle's Quarterly Newsletter*, *12*(2), 1–3. Retrieved from http://data.ashanet.org/files/Chapters/Seattle/Newsletters/Newsletter-2006Q1.pdf

Asmat, A., & Zamzami, S. Z. (2012). Automated House Detection and Delineation using Optical Remote Sensing Technology for Informal Human Settlement. *Procedia - Social and Behavioral Sciences*, *36*, 650–658. http://doi.org/10.1016/j.sbspro.2012.03.071

Backhaus, K., Erichson, B., Plinke, W., & Weiber, R. (2011). *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung* (13th ed.). Heidelberg: Springer-Verlag.

Ban, Y., Hu, H., & Rangel, I. M. (2010). Fusion of Quickbird MS and RADARSAT SAR data for urban land-cover mapping: Object-based and knowledge-based approach. *International Journal of Remote Sensing*, *31*(6), 1391–1410. http://doi.org/10.1080/01431160903475415

Ban, Y., Jacob, A., & Gamba, P. (2015). Spaceborne SAR Data for global urban mapping at 30 m resolution using a robust urban extractor. *ISPRS Journal of Photogrammetry and Remote Sensing*, *103*, 28–37. http://doi.org/10.1016/j.isprsjprs.2014.08.004

Baud, I., Kuffer, M., Pfeffer, K., Sliuzas, R., & Karuppannan, S. (2010). Understanding heterogeneity in metropolitan India: The added value of remote sensing data for analyzing sub-standard residential areas. *International Journal of Applied Earth Observation and Geoinformation*, *12*(5), 359–374. http://doi.org/10.1016/j.jag.2010.04.008

Bhagat, R. B., & Jones, G. W. (2013). *Population Change and Migration in Mumbai Metropolitan Region: Implications for Planning and Governance* (No. 201, ARI Working Paper). Singapore: Asia Research Institute.

Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *65*(1), 2–16.

Blaschke, T., & Lang, S. (2006). Object based image analysis for automated information extraction – A synthesis. In *Measuring the Earth II ASPRS Fall Conference* (pp. 6–10). San Antonio, Texas.

Breiman, L., & Cutler, A. (2001). Random Forests. Retrieved January 1, 2015, from https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

Brunner, D., Lemoine, G., Greidanus, L., & Harm, B. (2010). Building height retrieval from VHR SAR imagery based on an iterative simulation and matching technique. *IEEE Transactions on Geoscience and Remote Sensing*, *48*(3), 1487–1504.

Brunotte, E., Gebhardt, H., Meurer, M., Meusburger, P., & Nipper, J. (Eds.). (2002). *Lexikon der Geographie Band 3*. Berlin: Spektrum Akademischer Verlag Heidelberg.

Canadian Space Agency. (2007). Radarsat-2: Built for performance and versatility. Retrieved May 31, 2015, from http://www.asc-csa.gc.ca/eng/satellites/radarsat2/description.asp

Canadian Space Agency. (2011). Radarsat-2 Satellite Characteristics. Retrieved January 1, 2015, from http://www.asc-csa.gc.ca/eng/satellites/radarsat2/default.asp

Carr, J. R., & De Miranda, F. P. (1998). The semivariogram in comparison to the co-occurrence matrix for classification of image texture. *IEEE Transactions on Geoscience and Remote Sensing*, *36*(6), 1945–1952. http://doi.org/10.1109/36.729366

Census of India. (2011a). Provisional Population Totals: Cities / Cities having population 1 lakh and above. Retrieved June 21, 2015, from http://censusindia.gov.in/2011-prov-results/paper2/data_files/India2/Table_2_PR_Cities_1Lakh_and_Above.pdf

Census of India. (2011b). Provisional Population Totals: Urban Agglomerations / Cities having population 1 lakh and above. Retrieved June 21, 2015, from http://censusindia.gov.in/2011-prov-results/paper2/data_files/India2/Table_3_PR_UA_Citiees_1Lakh_and_Above.pdf

Chamundeeswari, V. V., Singh, D., & Singh, K. (2009). An Analysis of Texture Measures in PCA-Based Unsupervised Classification of SAR Images. *IEEE Geoscience and Remote Sensing Letters*, *6*(2), 214–218.

Christianini, N., & Shawe-Taylor, J. (2000). *An introduction to Support Vector Machines and other kernel-based learning algorithms*. Cambridge: Cambridge University Press.

Chuvieco, E., & Huete, A. (2010). *Fundamentals of Satellite Remote Sensing*. CRC Press Inc.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*, 37–46.

Congalton, R. G., & Green, K. (2009). *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices* (2nd ed.). CRC Press Inc.

D'Elia, C., Ruscino, S., Abbate, M., Aiazzi, B., Baronti, S., & Alparone, L. (2014). SAR Image Classification Through Information-Theoretic Textural Features , MRF Segmentation , and Object-Oriented Learning Vector Quantization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(4), 1116–1126.

Decker, R., Rašković, S., & Brunsiek, K. (2010). Diskriminanzanalyse. In C. Wolf & H. Best (Eds.), *Handbuch der sozialwissenschaftlichen Datenanalyse* (pp. 495–523). Wiesbaden: VS Verlag für Sozialwissenschaften.

Dekker, R. J. (2003). Texture analysis and classification of ERS SAR images for map updating of urban areas in The Netherlands. *IEEE Transaction on Geoscience and Remote Sensing*, *41*(9), 1950–1958.

Dell'Acqua, F., Stasolla, M., & Gamba, P. (2006). Unstructured Human Settlement Mapping with SAR sensors. *2006 IEEE International Symposium on Geoscience and Remote Sensing*, 3602–3605. http://doi.org/10.1109/IGARSS.2006.927

DIVA-GIS. (2009). Administrative areas (GADM) of India. Retrieved February 27, 2015, from http://www.diva-gis.org/gdata

EE Publishers. (2014). SAR Satellites. Retrieved May 31, 2015, from http://www.ee.co.za/wp-content/uploads/2014/09/optron-239-09-2014-fig1.jpg

Esch, T., Roth, A., & Dech, S. (2005). Robust approach towards an automated detection of built-up areas from high resolution radar imagery. In *Proceedings of the ISPRS WG VII/1"Human Settlements and Impact Analysis" 3rd International Symposium Remote Sensing and Data Fusion Over Urban Areas (URBAN 2005) and 5th International Symposium Remote Sensing of Urban Areas (URS 2005)*. Tempe, AZ, USA.

Esch, T., Roth, A., & Dech, S. (2006). Analysis of Urban Land Use Pattern Based on High Resolution Radar Imagery. *2006 IEEE International Symposium on Geoscience and Remote Sensing*, 3598–3601. http://doi.org/10.1109/IGARSS.2006.926

Esch, T., Thiel, M., Schenk, A., Roth, A., Müller, A., & Dech, S. (2010). Delineation of Urban Footprints From TerraSAR-X Data by Analyzing Speckle Characteristics and Intensity Information. *IEEE Transactions on Geoscience and Remote Sensing*, *48*(2), 905–916.

European Space Agency. (2015). Introducing Sentinel-1. Retrieved June 21, 2015, from http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Sentinel-1/Introducing_Sentinel-1

Faes, G. (2014). Lineare Diskriminanzanalyse. Retrieved March 12, 2015, from www.faes.de

Field, A., Miles, J., & Field, Z. (2012). *Discovering Statistics Using R*. Los Angeles: SAGE Publications.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*.

Gamba, P., & Aldrighi, M. (2012). SAR Data Classification of Urban Areas by Means of Segmentation Techniques and Ancillary Optical Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *5*(4), 1140–1148.

Gamba, P., Aldrighi, M., & Stasolla, M. (2011). Robust Extraction of Urban Area Extents in HR and VHR SAR Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *4*(1), 27–34.

Gamba, P., Dell'Acqua, F., & Trianni, G. (2006). Semi-automatic choice of scale-dependent features for satellite SAR image classification. *Pattern Recognition Letters*, *27*(4), 244–251.

Government of India. (1956). The Slum Areas (Improvement and Clearance) Act,. Retrieved March 3, 2015, from http://indiankanoon.org/doc/839084/

Government of India. (2010). *Report of the Committee on Slum Statistics/Census*. New Delhi. Retrieved from http://mhupa.gov.in/w_new/slum_report_nbo.pdf

Graesser, J., Cheriyadat, A., Vatsavai, R. R., Chandola, V., Long, J., & Bright, E. (2012). Image Based Characterization of Formal and Informal Neighborhoods in an Urban Landscape. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *5*(4), 1164–1176.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer Science & Business Media.

Henderson, F. M., & Xia, Z.-G. (1997). SAR Applications in Human Settlement Detection, Population Estimation and Urban Land Use Pattern Analysis : A Status Report. *IEEE Transactions on Geoscience and Remote Sensing*, *35*(1), 79–85.

Herold, M., Couclelis, H., & Clarke, K. C. (2005). The role of spatial metrics in the analysis and modeling of urban land use change. *Computers, Environment and Urban Systems*, *29*, 369–399. http://doi.org/10.1016/j.compenvurbsys.2003.12.001

Hofmann, P. (2001a). Detecting informal settlements from IKONOS image data using methods of object oriented image analysis - an example from Cape Town (South Africa). *Regensburger Geographische Schriften*, *35*, 107–118.

Hofmann, P. (2001b). Detecting urban features from IKONOS data using an object-oriented approach. *Proceedings of the First Annual Conference of the Remote Sensing & Photogrammetry Society*, 28–33.

Hofmann, P., Strobl, J., Blaschke, T., & Kux, H. (2008). Detecting informal settlements from Quickbird data in Rio de Janeiro using an object based approach. *Object-Based Image Analysis - Lecture Notes in Geoinformation and Cartography*, 531–553. http://doi.org/10.1007/978-3-540-77058-9_29

Huang, T.-M., Kecman, V., & Kopriva, I. (2006). *Kernel Based Algorithms for Mining Huge Data Sets*. Heidelberg: Springer-Verlag.

India Meteorological Department. (2015). Frequently Asked Questions (FAQs) on Monsoon. Retrieved March 1, 2015, from http://www.imd.gov.in/section/nhac/dynamic/FAQ_monsoon.htm

Jacquin, A., Misakova, L., & Gay, M. (2008). A hybrid object-based classification approach for mapping urban sprawl in periurban environment. *Landscape and Urban Planning*, *84*, 152–165. http://doi.org/10.1016/j.landurbplan.2007.07.006

Jain, S. (2007). Use of IKONOS satellite data to identify informal settlements in Dehradun , India. *International Journal of Remote Sensing*, *28*(15), 3227–3233. http://doi.org/10.1080/01431160600705122

Jensen, J. R. (2005). *Introductory Digital Image Processing: A Remote Sensing Perspective* (3rd ed.). Pearson Prentice Hall.

Jiang, L., Lin, H., & Cheng, S. (2011). Monitoring and assessing reclamation settlement in coastal areas with advanced InSAR techniques: Macao city (China) case study. *International Journal of Remote Sensing*, *32*(13), 3565–3588. http://doi.org/10.1080/01431161003752448

Karatzoglou, A., Meyer, D., & Hornik, K. (2006). Support Vector Machines in R. *Journal of Statistical Software*, *15*(9), 28. Retrieved from http://www.jstatsoft.org/v15/i09/paper

Kit, O., & Lüdeke, M. (2013). Automated detection of slum area change in Hyderabad, India using multitemporal satellite imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *83*, 130–137. http://doi.org/10.1016/j.isprsjprs.2013.06.009

Kohli, D., Warwadekar, P., Kerle, N., Sliuzas, R., & Stein, A. (2013). Transferability of Object-Oriented Image Analysis Methods for Slum Identification. *Remote Sensing*, *5*(9), 4209–4228. http://doi.org/10.3390/rs5094209

Kuffer, M., Pfeffer, K., Baud, I., & Sliuzas, R. (2013). Analysing sub-standard areas using high resolution remote (VHR) sensing imagery: Case study of Mumbai, India. In *Proceedings of the 14th N-AERUS / GISDECO conference, 12-14 September 2013*. Enschede, Netherlands.

Kuner, J. (2014). *Abschätzung der kleinräumigen Bevölkerungsverteilung in der Megastadt Mumbai mit Methoden der Fernerkundung*. Unpublished bachelor thesis, Universität Augsburg.

Landis, J., & Koch, G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*, 159–174.

Landscape Toolbox Wiki. (2013). Random Forests. Retrieved May 31, 2015, from http://wiki.landscapetoolbox.org/lib/exe/detail.php/remote_sensing_methods:random_forests_2.png?id=remote_sensing_methods%3Arandom_forests

Lang, S., Albrecht, F., & Blaschke, T. (2006). OBIA-Tutorial - Introduction to Object-based Image Analysis, V 1.0. *Centre for Geoinformatics, Paris-Lodron University Salzburg*. Salzburg. Retrieved from http://extras.springer.com/2008/978-3-540-77057-2/OBIA_Tutorial/OBIA Tutorial V1 - including text notes.PDF

Lillesand, T. M., Kiefer, R. W., & Chipman, J. W. (2004). *Remote sensing and image interpretation* (5th ed.). John Wiley & Sons Ltd.

MacDonald Dettwiler and Associates Ltd. (1999). How Radarsat sees the world. Retrieved May 31, 2015, from http://gs.mdacorporation.com/products/sensor/radarsat/rsiug98_499.pdf

MacDonald Dettwiler and Associates Ltd. (2014a). Radarsat-2 Beam Modes. Retrieved January 1, 2015, from http://gs.mdacorporation.com/SatelliteData/Radarsat2/BeamModes.aspx

MacDonald Dettwiler and Associates Ltd. (2014b). Radarsat-2 Product Description. Retrieved from http://gs.mdacorporation.com/products/sensor/radarsat2/RS2_Product_Description.pdf

Malcolm, N. W., Piwowar, J. M., Hall, G. B., Cotlier, C., & Ravenna, A. (2014). An Integration of RADARSAT and Landsat Imagery to Identify Areas of Urban Poverty in Rosario, Argentina. *Canadian Journal of Remote Sensing*, *27*(6), 663–668. http://doi.org/10.1080/07038992.2001.10854908

Municipal Corporation of Greater Mumbai. (2015). Ward Information. Retrieved February 27, 2015, from http://www.mcgm.gov.in/irj/portal/anonymous?NavigationTarget=navurl://745be63a8c86ec6b74ccad4690ad4100

Natural Earth. (2015). 1:10m Cultural Vectors. Retrieved February 27, 2015, from http://www.naturalearthdata.com/downloads/10m-cultural-vectors/

Natural Resources Canada. (2013). Radar Polarimetry. Retrieved May 31, 2015, from http://www.nrcan.gc.ca/earth-sciences/geomatics/satellite-imagery-air-photos/satellite-imagery-products/educational-resources/9275

Navan. (2014). Navan's blog: Understanding ROC curve. Retrieved May 31, 2015, from http://www.navan.name/roc/

Niebergall, S., Loew, A., & Mauser, W. (2008). Vulnerability in Delhi, India: identification and integrative assessment of informal settlements using remote sensing and GIS. In *Joint EARSeL and GISDECO 8 workshop: Integrating GIS and Remote Sessing in a Dynamic World*. Istanbul.

Niu, X., & Ban, Y. (2013). Multi-temporal RADARSAT-2 polarimetric SAR data for urban land-cover classification using an object- based support vector machine and a rule-based approach. *International Journal of Remote Sensing*, *34*(1), 1–26. http://doi.org/10.1080/01431161.2012.700133

Owen, K. K., & Wong, D. W. (2013). An approach to differentiate informal settlements using spectral, texture, geomorphology and road accessibility metrics. *Applied Geography*, *38*, 107–118. http://doi.org/10.1016/j.apgeog.2012.11.016

Pacifici, F., Chini, M., & Emery, W. J. (2009). A neural network approach using multi-scale textural metrics from very high-resolution panchromatic imagery for urban land-use classification. *Remote Sensing of Environment*, *113*(6), 1276–1292. http://doi.org/10.1016/j.rse.2009.02.014

Park, S., Yamaguchi, Y., & Kim, D. (2013). Polarimetric SAR remote sensing of the 2011 Tohoku earthquake using ALOS/PALSAR. *Remote Sensing of Environment*, *132*, 212–220. http://doi.org/10.1016/j.rse.2013.01.018

PBworks. (2007). Microarray Dimension Reduction Techniques: Fisher's Linear Discriminant. Retrieved June 21, 2015, from http://compbio.pbworks.com/w/page/16252904/Microarray Dimension Reduction Techniques

Purkis, S., & Klemas, V. (2011). *Remote Sensing and Global Environmental Change* (1st ed.). John Wiley & Sons Ltd.

Qi, Z., Yeh, A. G.-O., Li, X., & Lin, Z. (2012). A novel algorithm for land use and land cover classification using RADARSAT-2 polarimetric SAR data. *Remote Sensing of Environment*, *118*, 21–39. http://doi.org/10.1016/j.rse.2011.11.001

Richards, J. A., & Jia, X. (2006). *Remote Sensing Digital Image Analysis* (4th ed.). Springer Science & Business Media.

Risbud, N. (2003). Urban Slums Reports: The case of Mumbai, India. *Understanding Slums: Case Studies for the Global Report on Human Settlements 2003*, 1–20.

Risbud, N. (2010). Typology of Slums and Land Tenure in Indian Cities. In *Presentation at the National Workshop on Land Tenure Issues in Slum Free Planning*. Ahmadabad, India.

Rossiter, D. G. (2004). Technical Note: Statistical methods for accuracy assesment of classified thematic maps. *International Institute for Geo-Information Science & Earth Observation (ITC), Enschede*. Retrieved from http://www.itc.nl/personal/rossiter/teach/R/R_ac.pdf

Satterthwaite, D. (2010). *Urban Myths and the Mis-use of Data that Underpin them* (No. 2010, 28, Working paper). Helsinki: World Institute for Development Economics Research.

Schmidt, M., Esch, T., Thiel, M., & Dech, S. (2011). Detection of building structures from single-polarized TerraSAR-X data. In *Proc. of SPIE* (Vol. 8181). http://doi.org/10.1117/12.898348

Schmitt, A. (2011). *Änderungserkennung in multitemporalen und multipolarisierten Radaraufnahmen*. Dissertation, Universität Fridericiana zu Karlsruhe (TH).

Schmitt, A., Wendleder, A., & Hinz, S. (2015). The Kennaugh element framework for multi-scale, multi-polarized, multi-temporal and multi-frequency SAR image preparation. *ISPRS Journal of Photogrammetry and Remote Sensing*, *102*, 122–139. http://doi.org/10.1016/j.isprsjprs.2015.01.007

Shekhar, S. (2012). Detecting Slums From Quick Bird Data in Pune Using an Object Oriented Approach. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *39*(8), 519–524. http://doi.org/10.5194/isprsarchives-XXXIX-B8-519-2012

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Blake, R., … Andrew, K. (2013). Real-Time Human Pose Recognition in Parts from Single Depth Images. *Communications of the ACM*, *56*(1), 116–124. Retrieved from http://research.microsoft.com/pubs/145347/BodyPartRecognition.pdf

Shrinivasan, R. (2013). 17% of urban India lives in slums: Census. *The Times of India*. New Delhi. Retrieved from http://timesofindia.indiatimes.com/india/17-of-urban-India-lives-in-slums-Census/articleshow/19118219.cms

Soergel, U. (2010). Review of Radar Remote Sensing on Urban Areas. In U. Soergel (Ed.), *Radar Remote Sensing of Urban Areas* (pp. 1–47). Springer Science & Business Media.

Stasolla, M., & Gamba, P. (2008). Spatial Indexes for the Extraction of Formal and Informal Human Settlements From High-Resolution SAR Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *1*(2), 98–106. http://doi.org/10.1109/JSTARS.2008.921099

Taubenböck, H., Esch, T., Felbier, A., Wiesner, M., Roth, A., & Dech, S. (2012). Monitoring urbanization in mega cities from space. *Remote Sensing of Environment*, *117*, 162–176. http://doi.org/10.1016/j.rse.2011.09.015

Taubenböck, H., & Kraff, N. J. (2014). The physical face of slums: a structural comparison of slums in Mumbai, India, based on remotely sensed data. *Journal of Housing and the Built Environment*, *29*(1), 15–38.

Taubenböck, H., & Roth, A. (2010). Fernerkundung im urbanen Kontext. In H. Taubenböck & S. Dech (Eds.), *Fernerkundung in urbanem Raum: Erdbeobachtung auf dem Weg zur Planungspraxis*. Darmstadt: WBG (Wissenschaftliche Buchgesellschaft).

Taubenböck, H., Wurm, M., Setiadi, N., Gebert, N., Roth, A., Strunz, G., … Dech, S. (2009). Integrating Remote Sensing and Social Science - The correlation of urban morphology with socioeconomic parameters. In *Urban Remote Sensing Event, 2009 Joint*.

TeachEngineering. (2015). Slant Range. Retrieved May 31, 2015, from https://www.teachengineering.org/collection/utpa_/activities/utpa_perfectwaves/utpa_perfectwaves_activity1_figure3.jpg

Tiwari, P. S., Pande, H., & Nanda, B. N. (2004). Building footprint extraction from IKONOS imagery based on multi-scale object oriented fuzzy classification for urban disaster management. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *34*.

Trimble Germany GmbH. (2014). *Trimble eCognition ® Developer Reference Book*.

Trimble Germany GmbH. (2015). eCognition. Retrieved from http://www.ecognition.com/

Tso, B., & Mather, P. M. (2009). *Classification Methods for Remotely Sensed Data*. CRC Press Inc.

Tuia, D., Pacifici, F., Kanevski, M., & Emery, W. J. (2009). Classification of Very High Spatial Resolution Imagery Using Mathematical Morphology and Support Vector Machines. *IEEE Transactions on Geoscience and Remote Sensing*, *47*(11), 3866–3879. http://doi.org/10.1109/TGRS.2009.2027895

UN Habitat. (2007). State of the World's Cities 2006/7. *United Nations Publications*. Retrieved from https://sustainabledevelopment.un.org/content/documents/11292101_alt.pdf

UN Habitat. (2012). State of the World's Cities 2012/2013 Prosperity of Cities. *United Nations Publications*. Retrieved from mirror.unhabitat.org/pmss/getElectronicVersion.aspx?nr=3387&alt=1

UN-Habitat. (2003). *Slums of the World: the face of urban poverty in the new millennium? Monitoring the Millenium Development Goal, Target 11- World-wide Slum Dweller Estimation*. Nairobi, Kenya: Global Observatory. Retrieved from http://www.unhabitat.org/pmss/listItemDetails.aspx?publicationID=1124

United Nations. (2012a). The future we want: Cities. Retrieved from http://www.un.org/en/sustainablefuture/cities.asp

United Nations. (2012b). The Millennium Development Goals Report 2012. New York: United Nations Publications.

United Nations. (2014). World Urbanization Prospects: The 2014 Revision, Highlights. New York: United Nations Publications.

United Nations. (2015). Millenium Development Goals. Retrieved June 29, 2015, from http://www.un.org/millenniumgoals/environ.shtml

Weigand, M. (2014). *Discrimination of formal and informal settlements based on multipolarized TSX imagery in the Megacity Mumbai, India*. Unpublished bachelor thesis, Julius-Maximilians-Universität Würzburg.

Weydahl, D. J., Bretar, F., & Bjerke, P. (2005). Comparison of RADARSAT-1 and IKONOS satellite images for urban features detection. *Information Fusion*, *6*, 243–249. http://doi.org/10.1016/j.inffus.2004.07.001

Wikimedia. (2006). Lineare Trennbarkeit. Retrieved May 31, 2015, from http://upload.wikimedia.org/wikipedia/de/a/a0/Diskriminanzfunktion.png

Wikimedia. (2010). Zwei mögliche Trenngeraden mit verschiedenen Randgrößen. Retrieved May 31, 2015, from http://upload.wikimedia.org/wikipedia/commons/thumb/f/f2/Svm_intro.svg/232px-Svm_intro.svg.png

Wolff, C. (2008). Side Looking Airborne Radar (SLAR). Retrieved May 31, 2015, from http://www.radartutorial.eu/20.airborne/ab06.en.html

Worldometers. (2015). Population of India. Retrieved May 9, 2015, from http://www.worldometers.info/world-population/india-population/

Yang, X. (2011). *Urban Remote Sensing: Monitoring, Synthesis and Modeling in the Urban Environment*. John Wiley & Sons Ltd.

Zhang, Y., Zhang, H., & Lin, H. (2014). Improving the impervious surface estimation with combined use of optical and SAR remote sensing images. *Remote Sensing of Environment*, *141*, 155–167. http://doi.org/10.1016/j.rse.2013.10.028

Zheng, Z., JiXian, Z., GuoMan, H., & Rong-bin, W. (2004). The textural analysis and interpretation of high resolution AIRSAR Images. *Journal of Information Processing*, *2*(1).

List of Figures

List of Tables

A Appendix I: VV/VH polarized Radarsat imagery

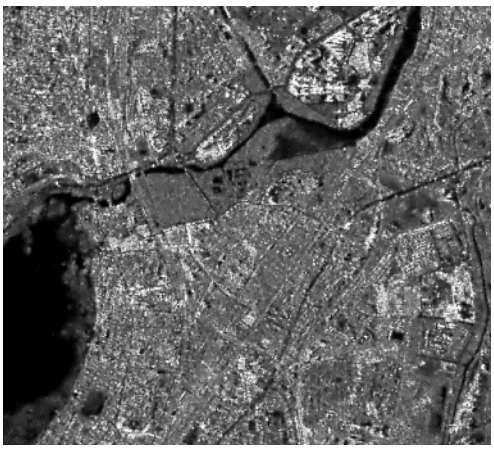


Figure A-1: Kennaugh-element K_0 (VV/VH)

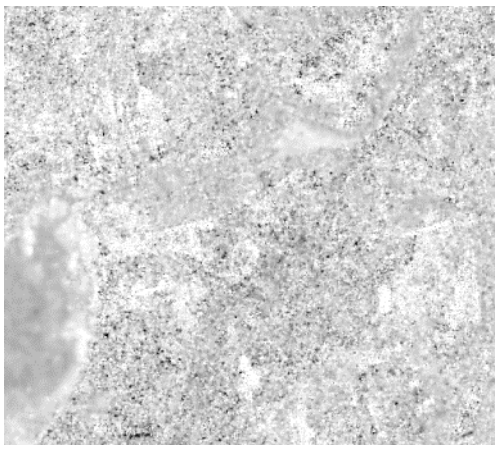


Figure A-2: Kennaugh-element K_1 (VV/VH)

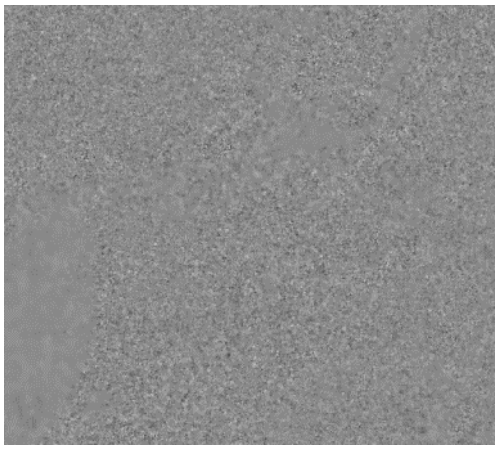


Figure A-3: Kennaugh-element K_5 (VV/VH)

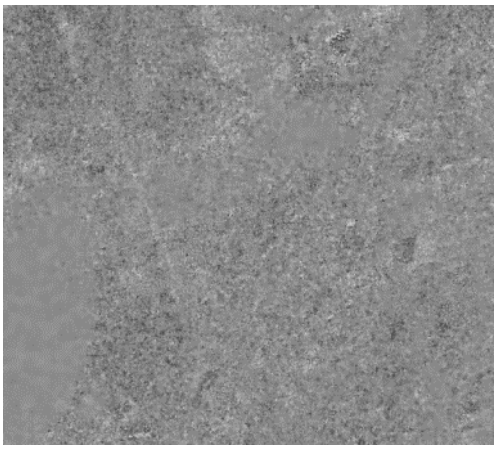


Figure A-4: Kennaugh-element K_8 (VV/VH)

Figure A-5: Optical Pléiades image with 0.5m resolution

Figure A-6: Auxiliary data (formal areas in green and slum areas in red)

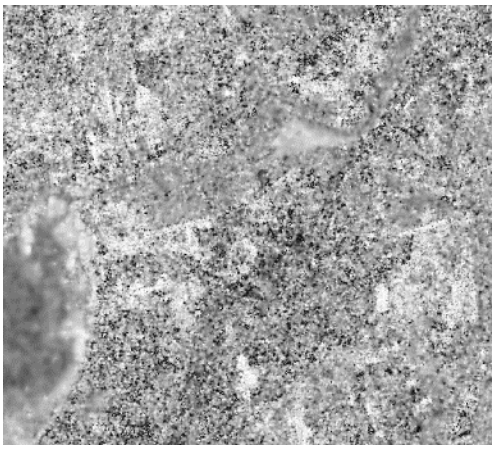


Figure A-7: Absorption

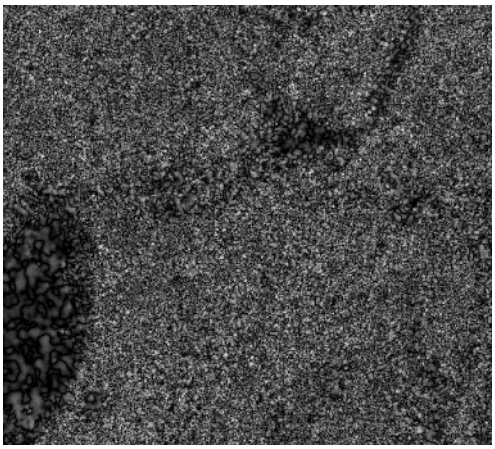


Figure A-8: Diattenuation

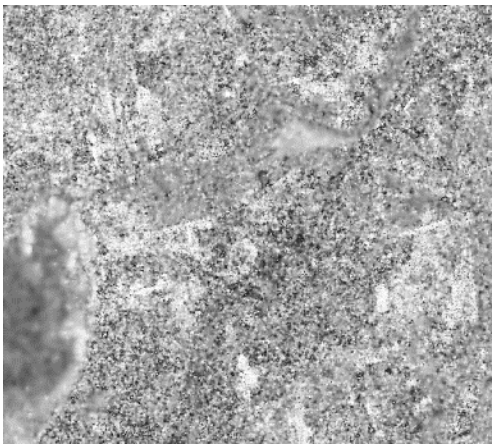


Figure A-9: Polarizance

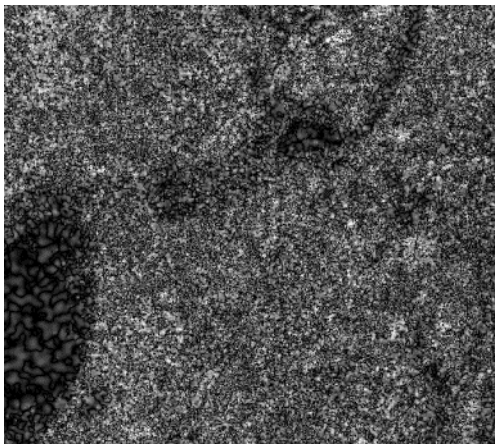


Figure A-10: Retardance

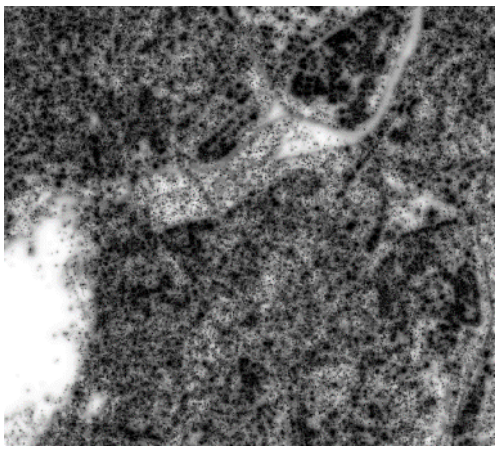


Figure A-11: Scale

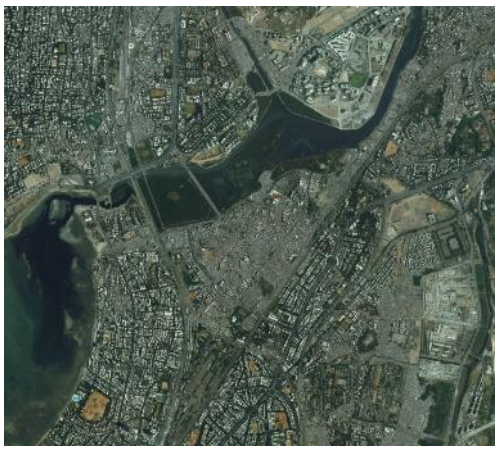


Figure A-12: Optical Pléiades image with 0.5m resolution

B Appendix II: R Script for Classification using Random Forest

```
################################################################################
#         Feature Selection and Classification based on Random Forest
################################################################################
# Author: Tatjana Bürgmann
# Date: March 12, 2015
# Program description: Script for automatic backward selection of features based
#                      on Random Forest (RF) and subsequent classification using
#                      the 30 best features.

# Feature selection is implemented by training a RF with 125 trees and
# eliminating the feature with smallest ranking criterion. The criterion used is
# the area under curve (AUC). The RF model is calculated based on a small sample
# of objects and run on the residual data set. Finally, the class membership is
# estimated by RF to the output data set.

# The input dataset should have the following structure:
#  - one column should contain the class as factor
#  - one column should contain the object-ID
#  - other columns should contain the features
# ------------------------------------------------------------------------------

# save start time for calculation of runtime
startTime = Sys.time()

library(foreign)      # package to support reading .dbf (data base files)
library(randomForest) # package for RF functions

################################################################################
#                               Data preparation
################################################################################

# set working directory for input file
setwd("E:/Thesis/Praxis/R/Input")

# read .dbf file
data.orig = read.dbf ("GLCM.dbf")

# change the order of variables and select a subset of features
data.orig = subset(data.orig, select=c(Class, ID, X05_Abs_Ang, X05_Abs_Con,
X05_Abs_Cor, X05_Abs_Ent, X05_Abs_Hom, X05_Abs_Mea, X05_Abs_Sta, X05_Dia_Ang,
X05_Dia_Con, X05_Dia_Cor, X05_Dia_Ent, X05_Dia_Hom, X05_Dia_Mea, X05_Dia_Sta,
X05_K0_Ang, X05_K0_Con, X05_K0_Cor, X05_K0_Ent, X05_K0_Hom, X05_K0_Mea,
X05_K0_Sta, X05_K1_Ang, X05_K1_Con, X05_K1_Cor, X05_K1_Ent, X05_K1_Hom,
X05_K1_Mea, X05_K1_Sta, X05_K5_Ang, X05_K5_Con, X05_K5_Cor, X05_K5_Ent,
X05_K5_Hom, X05_K5_Mea, X05_K5_Sta, X05_K8_Ang, X05_K8_Con, X05_K8_Cor,
X05_K8_Ent, X05_K8_Hom, X05_K8_Mea, X05_K8_Sta, X05_Pol_Ang, X05_Pol_Con,
X05_Pol_Cor, X05_Pol_Ent, X05_Pol_Hom, X05_Pol_Mea, X05_Pol_Sta, X05_Ret_Ang,
X05_Ret_Con, X05_Ret_Cor, X05_Ret_Ent, X05_Ret_Hom, X05_Ret_Mea, X05_Ret_Sta,
X05_Sca_Ang, X05_Sca_Con, X05_Sca_Cor, X05_Sca_Ent, X05_Sca_Hom, X05_Sca_Mea,
X05_Sca_Sta, X10_Abs_Ang, X10_Abs_Con, X10_Abs_Cor, X10_Abs_Ent, X10_Abs_Hom,
X10_Abs_Mea, X10_Abs_Sta, X10_Dia_Ang, X10_Dia_Con, X10_Dia_Cor, X10_Dia_Ent,
X10_Dia_Hom, X10_Dia_Mea, X10_Dia_Sta, X10_K0_Ang, X10_K0_Con, X10_K0_Cor,
X10_K0_Ent, X10_K0_Hom, X10_K0_Mea, X10_K0_Sta, X10_K1_Ang, X10_K1_Con,
X10_K1_Cor, X10_K1_Ent, X10_K1_Hom, X10_K1_Mea, X10_K1_Sta, X10_K5_Ang,
X10_K5_Con, X10_K5_Cor, X10_K5_Ent, X10_K5_Hom, X10_K5_Mea, X10_K5_Sta,
X10_K8_Ang, X10_K8_Con, X10_K8_Cor, X10_K8_Ent, X10_K8_Hom, X10_K8_Mea,
X10_K8_Sta, X10_Pol_Ang, X10_Pol_Con, X10_Pol_Cor, X10_Pol_Ent, X10_Pol_Hom,
X10_Pol_Mea, X10_Pol_Sta, X10_Ret_Ang, X10_Ret_Con, X10_Ret_Cor, X10_Ret_Ent,
X10_Ret_Hom, X10_Ret_Mea, X10_Ret_Sta, X10_Sca_Ang, X10_Sca_Con, X10_Sca_Cor,
```

```
X10_Sca_Ent, X10_Sca_Hom, X10_Sca_Mea, X10_Sca_Sta))

################################################################################
#                    Selective Backwards Feature Selection
################################################################################

# Function for backward feature selection that returns a ranked list of features
source("E:/Thesis/Praxis/R/Skripte/RF_Feature_Selection.R")
featureRankedList = rfFeatureRanking(data.orig, "E:/Thesis/Praxis/R/Output/RF")

# features selected by feature selection
featuresSelected = featureRankedList[1:30,1]

# subset of the original data set containing the 30 best ranked features
data.set = data.orig[,c("Class","ID", featuresSelected)]

################################################################################
#                                Classification
################################################################################

# create random samples for 20% of the objects of each class as calibration
# objects for RF
source("E:/Thesis/Praxis/R/Skripte/sample.r") # source: Michael Wurm, DLR
sample = sample.by.var(data.set, var="Class", probs=0.2)

# list containing the IDs of the sample objects
list = as.numeric(sample$ID)

# validation objects for accuracy assessment
# (all objects not included in the sample)
other = data.set[!(data.set$ID %in% list),]

# data set used for prediction
other.features = other[,c(3:ncol(other))]

################################################################################
#                                 Random Forest
################################################################################

# grow model
rf.model = randomForest(sample[,3:ncol(sample)], as.factor(sample[,1]),
                        importance=TRUE, ntree=125, mtry=11)

################################################################################
#             Prediction of the function on the residual data set
################################################################################

# prediction
prediction.class = predict(rf.model, other.features, type = "class")
prediction.class = as.data.frame(prediction.class)

# transfer the class membership into the data set with ID and Class column
other$Group = prediction.class$prediction.class

# calculate and transfer the probability into the data set with ID and Class
# column (the maximal value (probability of group that was chosen) is taken
# and rounded to two decimal places)
pred.prob = predict(rf.model, other.features, type = "prob")
pred.prob.data = as.data.frame(pred.prob)
colnames(pred.prob.data) = c("prob.1","prob.2")
# 1 = row by row, max = maximal value, 2 = rounded to two decimal places
other$Prob = round(apply(pred.prob.data[,1:ncol(pred.prob.data)], 1, max),2)

################################################################################
#           Accuracy assessment on basis of reference classification
```

```
################################################################################

# validation of the classification in a contingency matrix
error.matrix = table(other$Group, other$Class)

source("E:/Thesis/Praxis/R/Skripte/kappa.txt") # kappa function
# Source: Rossiter, D. G. (2004)

# calculation of accuracy values
kappa.function = kappa(error.matrix)

# kappa, overall, user's and producer's accuracy
kappa = kappa.function$sum.kappa
overall.accuracy = kappa.function$sum.naive
users.accuracy = kappa.function$user.naive
producers.accuracy = kappa.function$prod.naive

# true skill statistic (TSS)
confusionMatrix = as.matrix(error.matrix)
a = confusionMatrix[1,1]
b = confusionMatrix[1,2]
c = confusionMatrix[2,1]
d = confusionMatrix[2,2]
sensitivity = a/(a+c)
specificity = d/(b+d)
tss = sensitivity + specificity - 1

################################################################################
#                       Export of the classification result
################################################################################

# set working directory for output
setwd ("E:/Thesis/Praxis/R/Output/RF")

# export data frame containing ID, predicted group membership and probability
# values of the features
result = data.frame(other[,2], other$Group, other$Prob)
colnames(result) = c("ID", "Group", "Probability")
write.dbf(result, file="RF_classification.dbf")

# export feature ranking
ranking.result = as.data.frame(featureRankedList)
write.dbf(ranking.result, file="RF_feature_ranking.dbf")

# export text file containing the accuracy values of the classification
# (overall accuracy, kappa, user's and producer's accuracy)
cat(paste("Overall accuracy: ", print(round(overall.accuracy,2))),
    file="RF_accuracy.txt", sep="\n")
cat(paste("Kappa", print(round(kappa,2))),file="RF_accuracy.txt",sep="\n",
    append=TRUE)
cat(paste("User's accuracy (Formal): ", print(round(users.accuracy[1],2))),
    file="RF_accuracy.txt", sep="\n", append=TRUE)
cat(paste("Producer's accuracy (Formal): ", print(round(producers.accuracy[1],2))),
    file="RF_accuracy.txt",sep="\n", append=TRUE)
cat(paste("User's accuracy (Informal): ", print(round(users.accuracy[2],2))),
    file="RF_accuracy.txt", sep="\n", append=TRUE)
cat(paste("Producer's accuracy (Informal): ",
print(round(producers.accuracy[2],2))),
    file="RF_accuracy.txt", sep="\n", append=TRUE)
cat(paste("True Skill Statistic (TSS): ", print(round(tss,2))),
    file="RF_accuracy.txt", sep="\n", append=TRUE)

# export text file containing the start, end and runtime
endTime = Sys.time()
runtime = difftime(startTime, endTime)
```

```
cat(paste("Start time: ", print(startTime)),file="runtime.txt",sep="\n")
cat(paste("End time", print(endTime)),file="runtime.txt",sep="\n",
    append=TRUE)
cat(paste("Runtime", print(runtime)),file="runtime.txt",sep="\n",
    append=TRUE)
```

C Appendix III: R Script for Feature Selection using Random Forest

```
#############################################################################
#     Selective backward feature selection based on Random Forest
#############################################################################
# Author: Tatjana Bürgmann
# Date: March 12, 2015
# Program description: Script for automatic backward selection of features based
#                      on Random Forest (RF)

# Feature selection is implemented by training a RF model and removing the
# feature with smallest ranking criterion. The criterion used is the area under
# curve (AUC). The RF model is calculated based on a small sample of objects and
# run on the residual data set. The class membership is estimated by RF to the
# output data set. Output of this script is a data frame containing the ranking
# of the features and according AUC values.

# The input dataset should have the following structure:
#  - one column should contain the class as factor
#  - one column should contain the object-ID
#  - other columns should contain the features
# ---------------------------------------------------------------------------

rfFeatureRanking = function(dataset, workingDirectory) {

  library(randomForest) # package for RF functions
  library(caTools)      # package for AUC function

  # number of features to be ranked (without class and ID column)
  ToBeRankedFeatures = ncol(dataset)-2

  # data frame for ranked features and AUC values
  featureRankedList = data.frame(matrix(vector(), nrow=ToBeRankedFeatures,
                      ncol=8, dimnames=list(c(), c("Feature", "AUC", "Kappa",
                      "Overall", "Users1", "Users2", "Producers1",
                      "Producers2"))), stringsAsFactors=F)

  # perform experiment until only 1 feature remains
  while(ToBeRankedFeatures > 1) {

    # create random samples for 20% of the objects of each class as
    # calibration objects for RF
    source("E:/Thesis/Praxis/R/Skripte/sample.r") # source: Michael Wurm, DLR
    sample = sample.by.var(dataset, var="Class", probs=0.2)

    # list containing the IDs of the sample objects
    list = as.numeric(sample$ID)

    # validation objects for accuracy assessment (all objects not included in
    # the sample)
    other = dataset[!(dataset$ID %in% list),]

    # data set used for prediction
    other.features = other[,c(3:ncol(other))]

    ###########################################################################
    #                               Random Forest
    ###########################################################################

    # grow model
    rf.model = randomForest(sample[,3:ncol(sample)], as.factor(sample[,1]),
```

```
                        importance=TRUE, ntree=125, mtry=11)

#############################################################################
#            Prediction of the function on the residual data set
#############################################################################

# prediction
prediction.class = predict(rf.model, other.features, type = "class")
prediction.class = as.data.frame(prediction.class)

# transfer the class membership into the data set with ID and Class column
other$Group = prediction.class$prediction.class

# calculate and transfer the probability into the data set with ID and
# Class column (the maximal value (probability of group that was chosen)
# is taken and rounded to two decimal places)
pred.prob = predict(rf.model, other.features, type = "prob")
pred.prob.data = as.data.frame(pred.prob)
colnames(pred.prob.data) = c("prob.1","prob.2")
# 1 = row by row, max = maximal value, 2 = rounded to two decimal places
other$Prob = round(apply(pred.prob.data[,1:ncol(pred.prob.data)], 1,
                         max),2)

#############################################################################
#       Accuracy assessment on basis of reference classification
#############################################################################

# validation of the classification in a contingency matrix
error.matrix = table(other$Group, other$Class)

source("E:/Thesis/Praxis/R/Skripte/kappa.txt") # kappa function
# Source: Rossiter, D. G. (2004)

# calculation of accuracy values
kappa.function = kappa(error.matrix)

# kappa, overall, user's and producer's accuracy
kappa = kappa.function$sum.kappa
overall.accuracy = kappa.function$sum.naive
users.accuracy = kappa.function$user.naive
producers.accuracy = kappa.function$prod.naive

#############################################################################
#                  Ranking Criteria for feature selection
#############################################################################

# calculate AUC (Area Under Curve) for every feature (column)
auc = colAUC(sample[,3:ncol(sample)], as.factor(sample[,1]), plotROC=FALSE,
             alg="ROC")

#############################################################################
#                Find and drop variable of minimal importance
#############################################################################

auc = as.data.frame(auc)
rownames(auc) = c(1:nrow(auc)) # number rows consecutively

ranking = t(auc)
ranking = as.data.frame(ranking)
ranking$feature = rownames(ranking)
rownames(ranking) = nrow(c(ranking))
colnames(ranking) <- c("AUC", "feature")

# rank features in ascending order
final.ranking = ranking[with(ranking, order(-AUC, feature)),]
```

```
    # selects the element in the last row and second column (variable with
    # minimal partial clarification)
    featureMin = final.ranking[nrow(final.ranking),2]

    # combine variable name and accuracy of the variable with minimal partial
clarification
    rankedFeature = cbind(rankedFeature, as.character(featureMin))
    rankedFeature = cbind(rankedFeature, as.numeric(round(auc,2)))
    rankedFeature = cbind(rankedFeature, as.numeric(round(kappa,2)))
    rankedFeature = cbind(rankedFeature, as.numeric(round(overall.accuracy,2)))
    rankedFeature = cbind(rankedFeature, as.numeric(round(users.accuracy[1],2)))
    rankedFeature = cbind(rankedFeature, as.numeric(round(users.accuracy[2],2)))
    rankedFeature = cbind(rankedFeature,
as.numeric(round(producers.accuracy[1],2)))
    rankedFeature = cbind(rankedFeature,
as.numeric(round(producers.accuracy[2],2)))

    # update feature ranking
    featureRankedList[ToBeRankedFeatures,] = rankedFeature[1,]

    # update number of features to be ranked
    ToBeRankedFeatures = ToBeRankedFeatures - 1

    # eliminate the feature with minimal importance from the data set used for
sampling
    dataset = dataset[, !(colnames(dataset) %in% c(featureMin))]
  }

  # write name of last feature in the ranked feature list
  featureRankedList[1,1] = colnames(dataset[3])

  return (featureRankedList)
}
```

D Appendix IV: Classification results using the 30 best Features

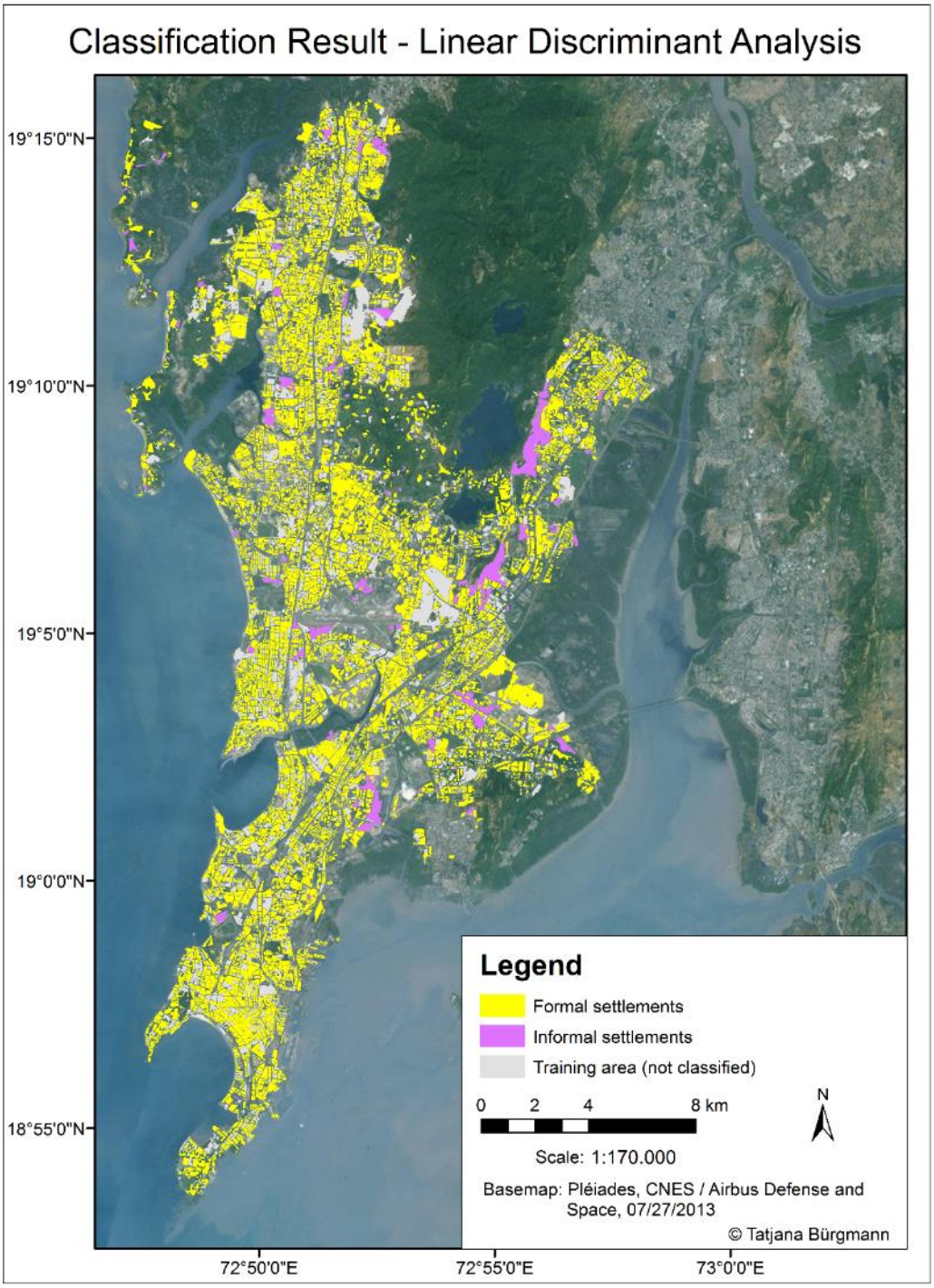


Figure D-13: Result of the classification using Linear Discriminant Analysis

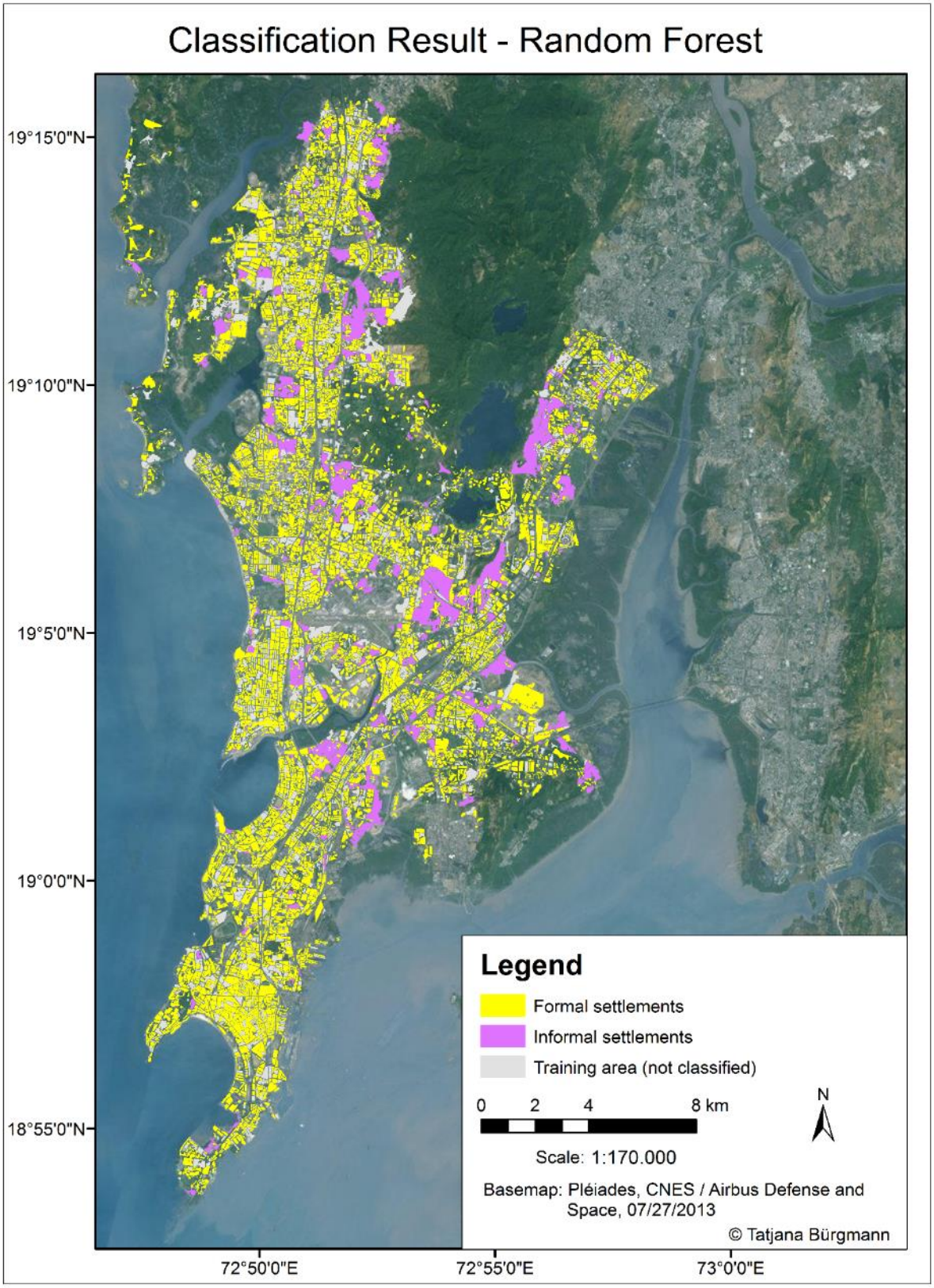


Figure D-14: Result of the classification using Random Forest

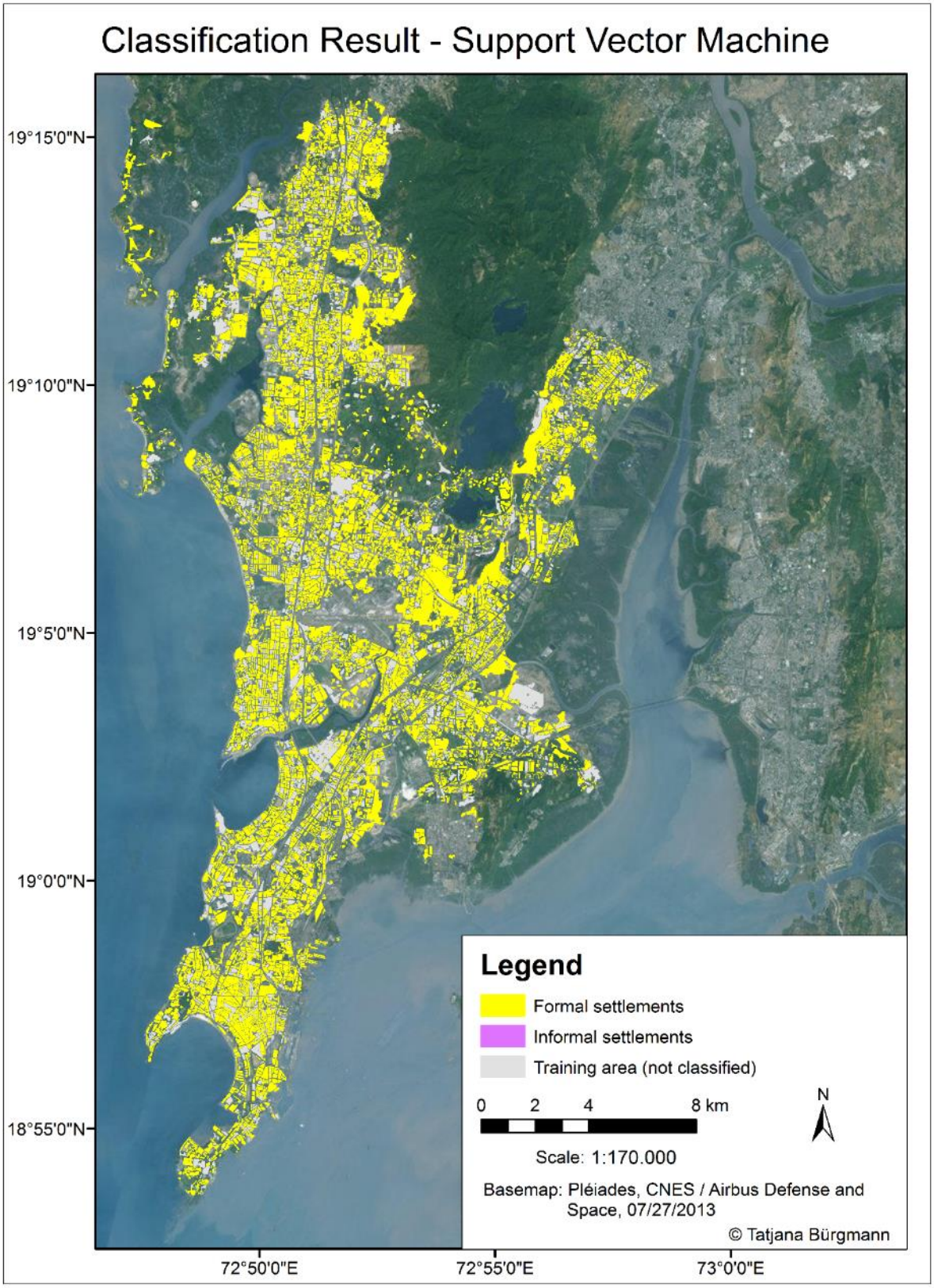


Figure D-15: Result of the classification using Support Vector Machines

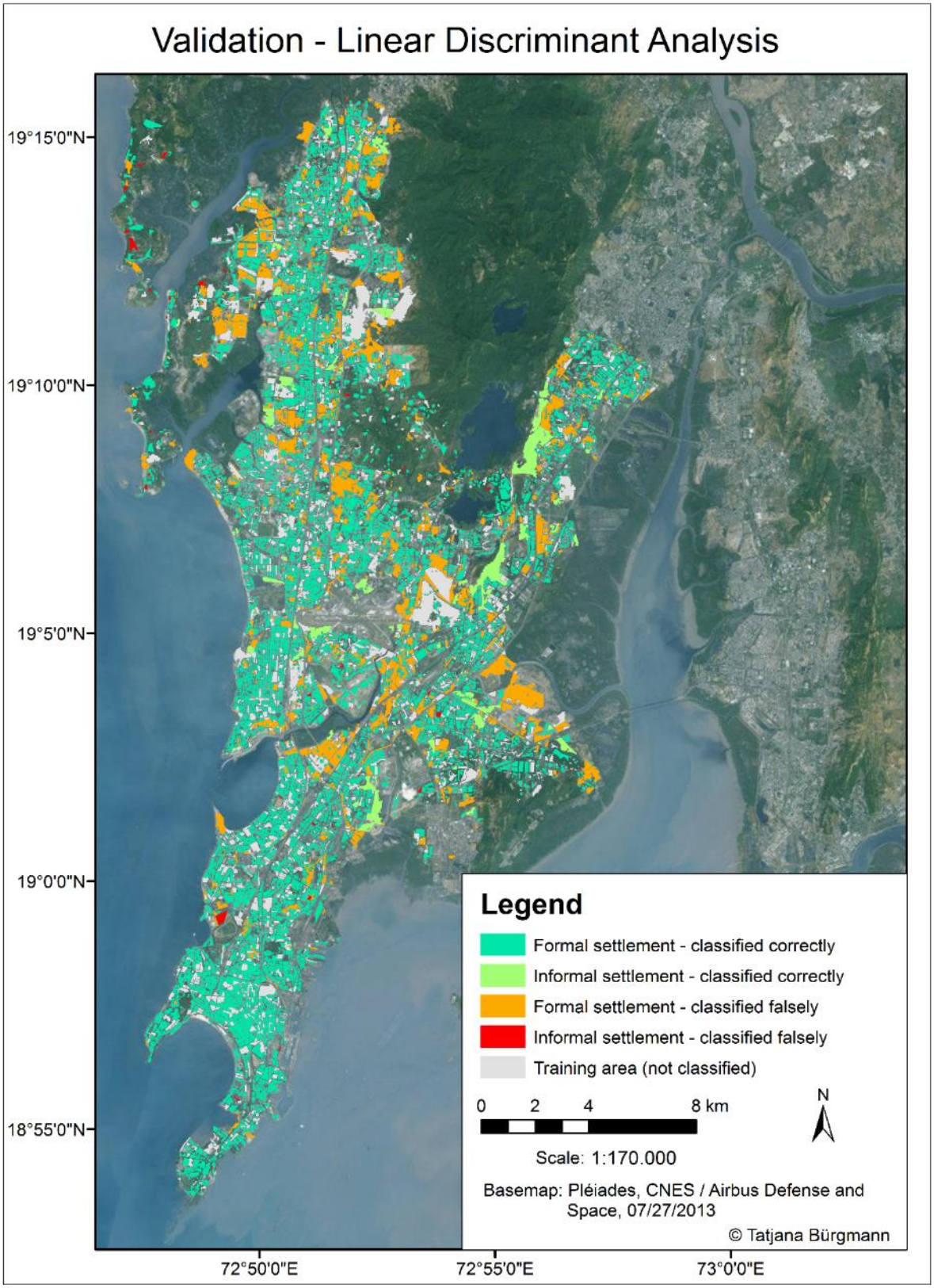


Figure D-16: Validation of the classification result using Linear Discriminant Analysis

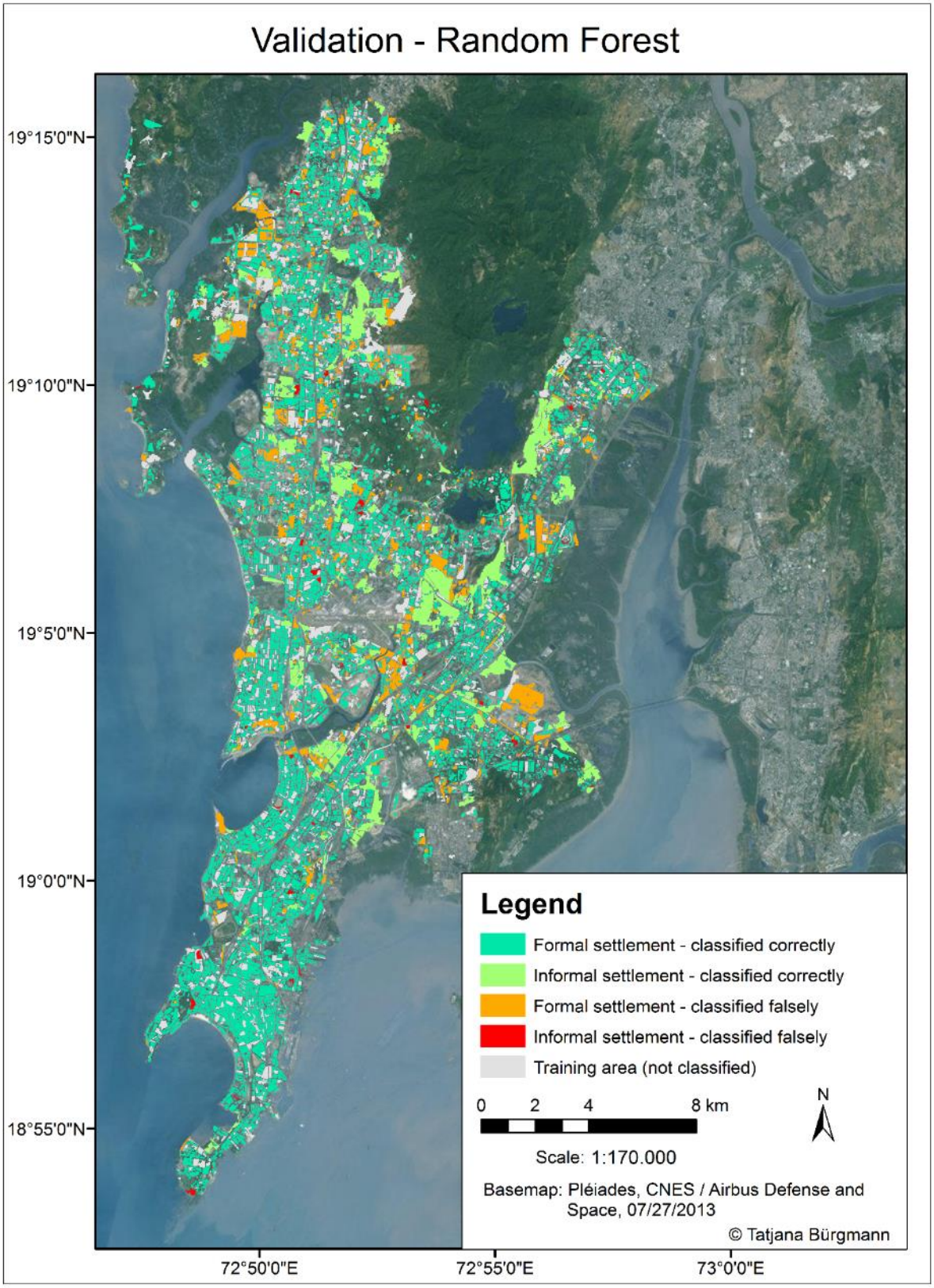


Figure D-17: Validation of the classification result using Random Forest

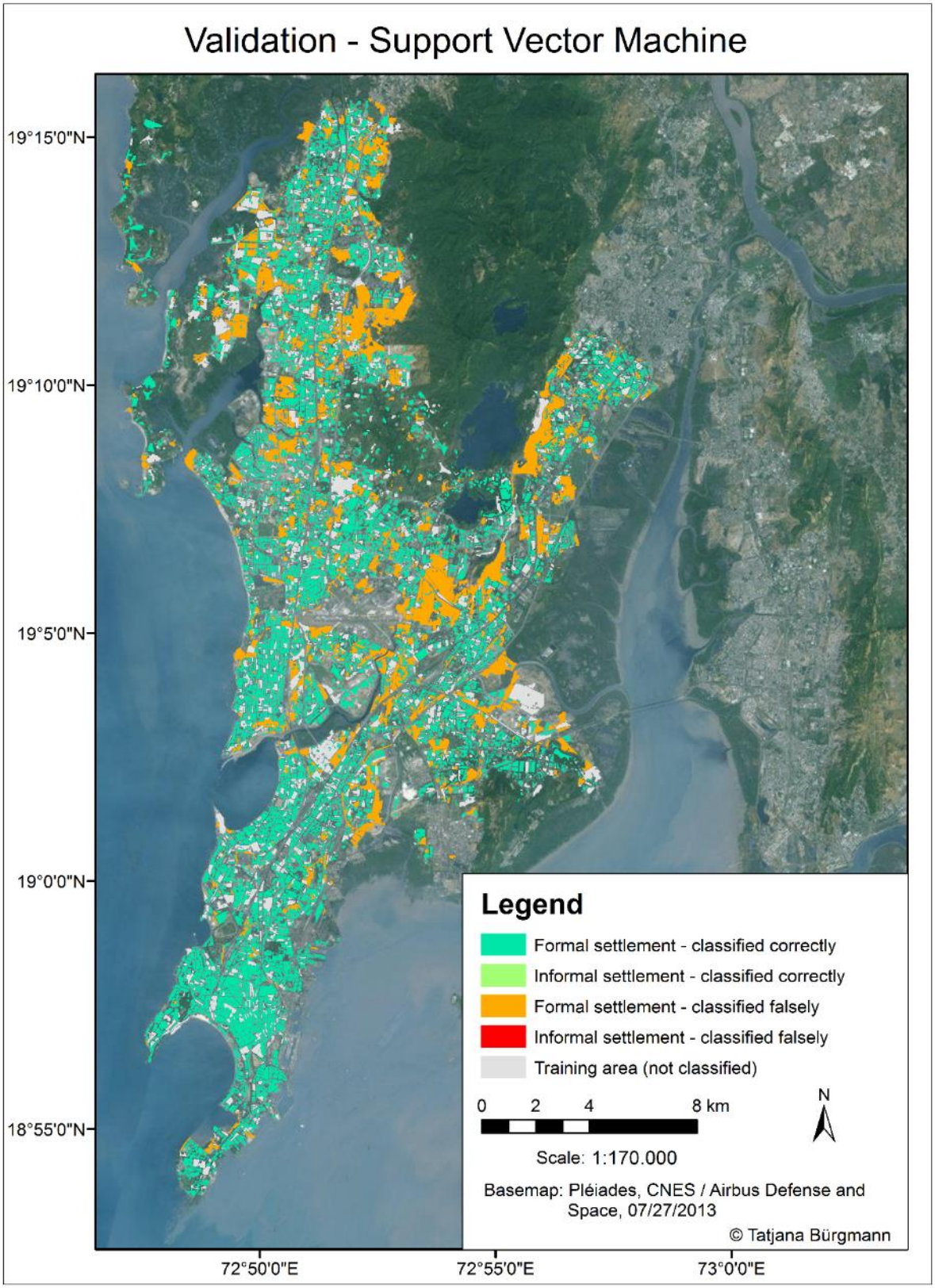


Figure D-18: Validation of the classification result using Support Vector Machine

Figure D-19: Validation of correctly and incorrectly classified blocks using LDA

Figure D-20: Validation of correctly and incorrectly classified blocks using RF

Figure D-21: Validation of correctly and incorrectly classified blocks using SVM:

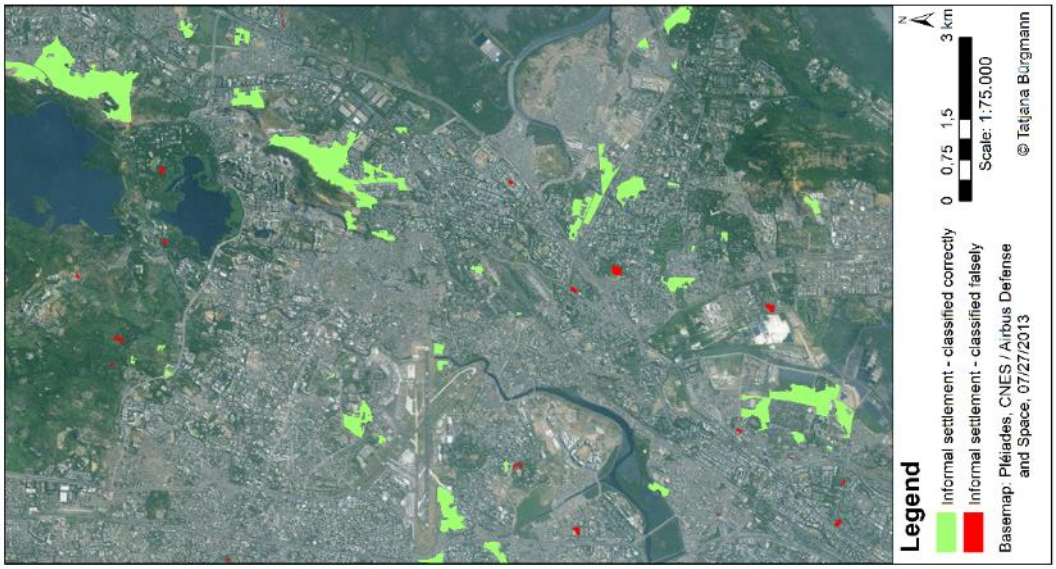


Figure D-22: Validation of Informal Settlements using LDA

Figure D-23: Validation of Informal Settlements using RF

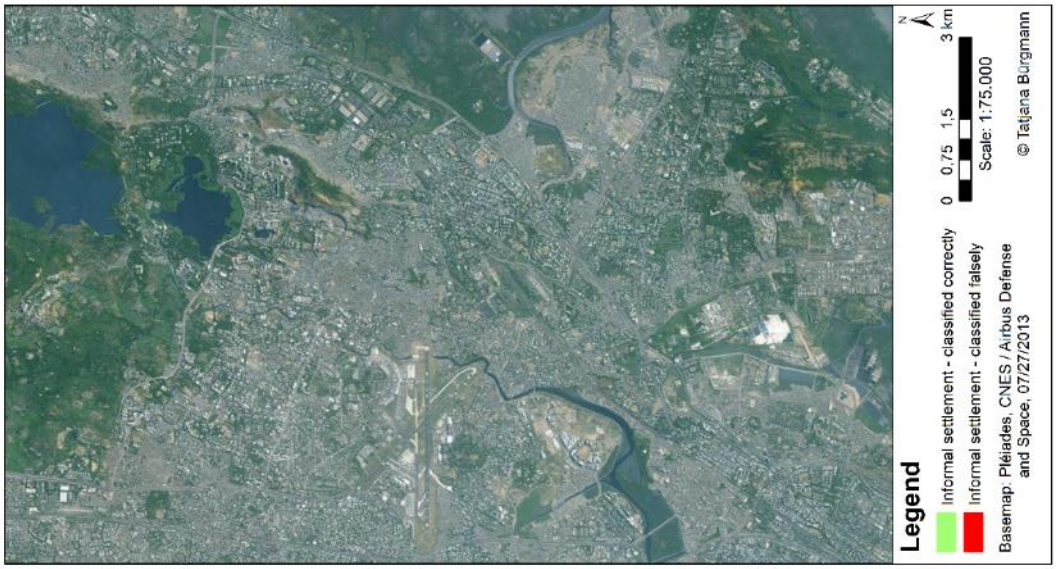


Figure D-24: Validation of Informal Settlements using SVM:

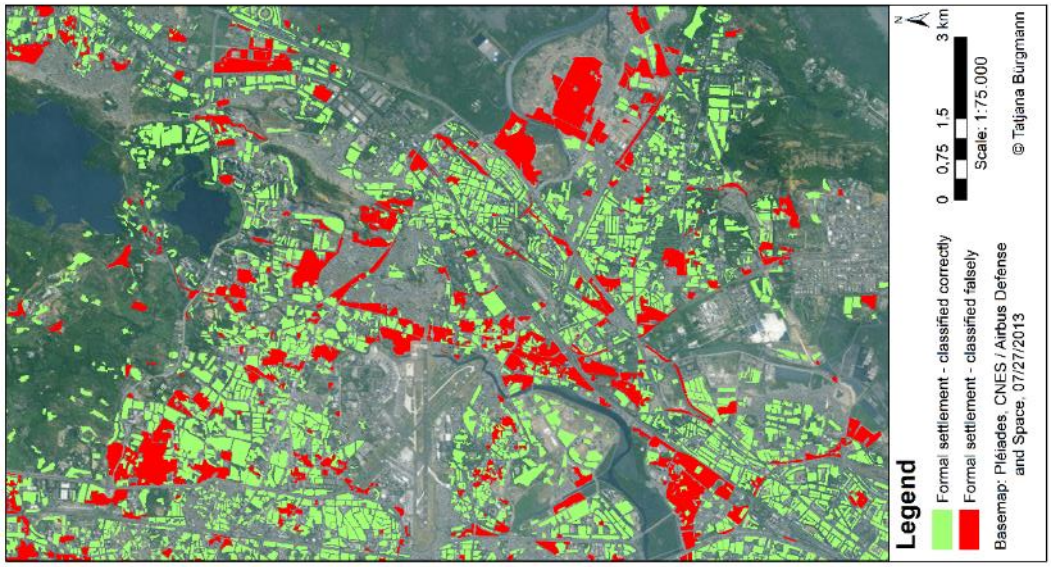


Figure D-25: Validation of Formal Settlements using LDA

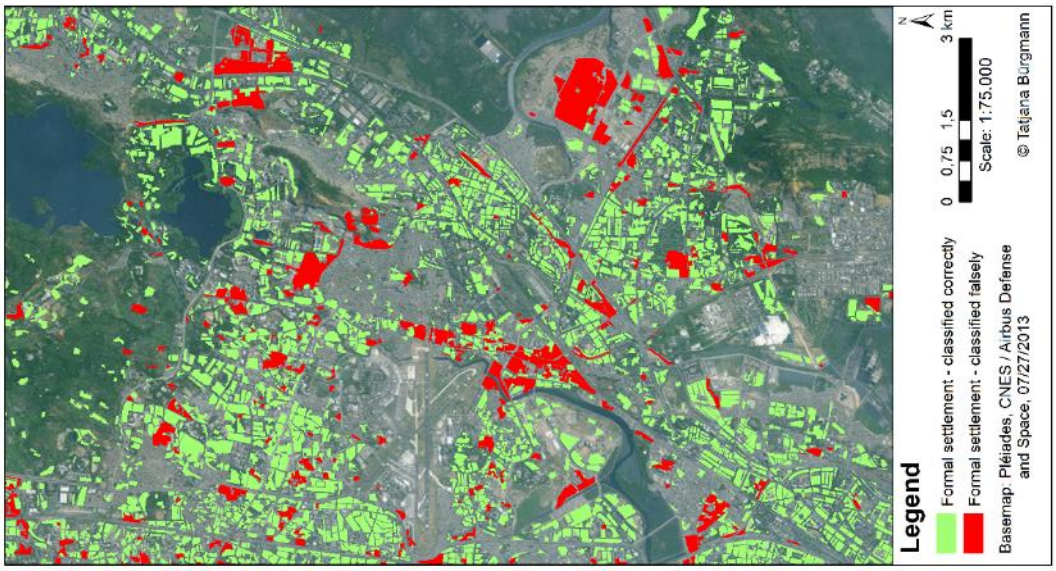


Figure D-26: Validation of Formal Settlements using RF

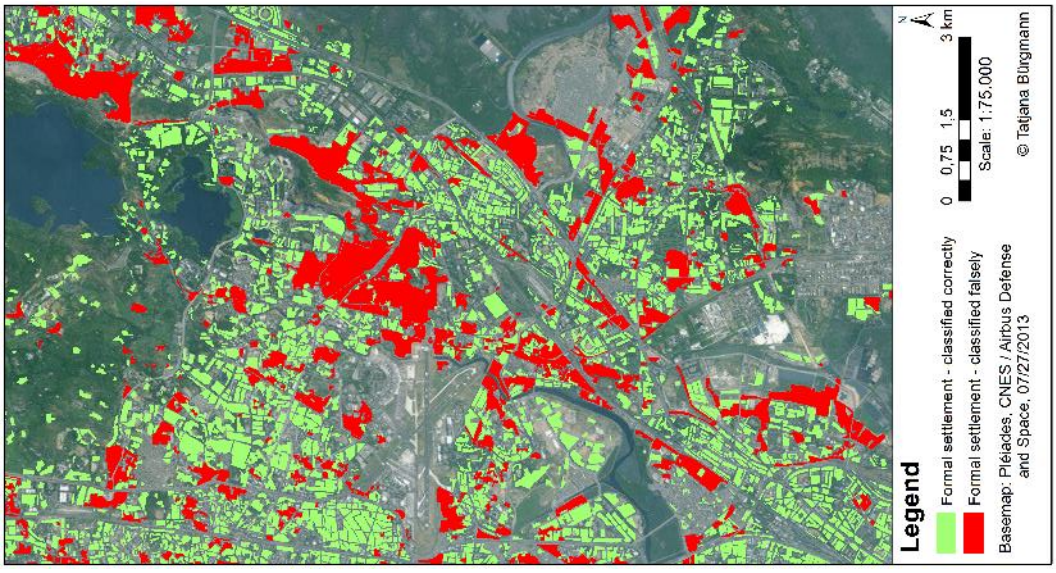


Figure D-27: Validation of Formal Settlements using SVM:

E Appendix V: Accuracy Assessment Results

Table E-1: Accuracy values reached by LDA using a different number of features

No. of features	10	20	30	40	50	60	70	80	90	100	126
Overall	88%	88%	89%	89%	89%	89%	89%	89%	89%	89%	89%
Kappa	0.03	0.05	0.12	0.2	0.23	0.28	0.3	0.32	0.33	0.35	0.36
User's (Formal)	89%	89%	89%	90%	90%	91%	91%	91%	91%	92%	92%
Producer's (Formal)	100%	100%	99%	98%	98%	98%	98%	97%	97%	97%	96%
User's (Informal)	45%	47%	52%	53%	57%	56%	59%	57%	59%	58%	55%
Producer's (Informal)	2%	3%	8%	15%	18%	23%	24%	27%	28%	31%	34%
TSS	0.02	0.03	0.07	0.14	0.16	0.21	0.22	0.24	0.25	0.28	0.3

Table E-2: Accuracy values reached by RF using different numbers of trees

No. of trees	5	10	25	50	125	250	500	1000	1500	2000
Overall	89%	90%	90%	90%	90%	90%	90%	90%	90%	91%
Kappa	0.29	0.31	0.31	0.31	0.31	0.31	0.32	0.32	0.32	0.33
User's (Formal)	91%	91%	91%	91%	91%	91%	91%	91%	91%	91%
Producer's (Formal)	97%	98%	99%	99%	99%	99%	99%	99%	99%	99%
User's (Informal)	51%	60%	72%	75%	78%	79	78%	78%	81%	79%
Producer's (Informal)	27%	26%	24%	23%	22%	22%	23%	23%	22%	23%
TSS	0.23	0.23	0.22	0.22	0.21	0.22	0.23	0.22	0.22	0.23
Runtime (hours)	0.25	0.3	0.5	0.8	1.5 h	3.5	6.5	11.5	21	25

Table E-3: Results of the Accuracy Assessment and performance of LDA, RF and SVM using the 30 best features

	LDA	RF	SVM
Overall	89%	91%	88%
Kappa	0.12	0.32	0
User's (Informal)	52%	82%	12%
Producer's (Informal)	8%	23%	1%
User's (Formal)	89%	91%	89%
Producer's (Formal)	99%	99%	100%
TSS	0.07	0.22	0
Runtime	18 min	6 h	2.5 h