Semantic Annotation and Object Extraction for Very High Resolution Satellite Images

DISSERTATION

zur Erlangung des akademischen Grades eines Doktors
der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von
M.Sc. Wei Yao
eingereicht bei der Naturwissenschaftlichen-Technischen Fakultät
der Universität Siegen
Siegen 2017

Tag der mündlichen Prüfung: 15. Dezember 2017
Gutachter:

- Prof. Dr. Otmar Loffeld
- Prof. Dr. Mihai Datcu

Prüfer:

- Prof. Dr. Otmar Loffeld
- Prof. Dr. Mihai Datcu
- Prof. Dr. Andreas Kolb
- Prof. Dr. Michael Möller (Vorsitz der Prüfungskommission)

Gedruckt auf alterungsbeständigem holz- und säurefreiem Papier.

This dissertation is dedicated to my loving and beloved parents.

Acknowledgements

It has been a long journey to write this dissertation, from trying to solve the problem using signal processing techniques at ZESS in the University of Siegen, till focusing on the classification application from an image processing perspective at the Remote Sensing Institute in DLR. However, I personally quite enjoyed this special experience and treasure this journey of trying various ideas, experimenting, writing articles and, in the end, completing this dissertation. I also treasure the help from all sides: professors, colleagues, secretaries, university administrative people, parents, friends, landlord and landlady, etc.

Particularly, I would like to thank Prof. Otmar Loffeld to provide me the MOSES scholarship at ZESS from the University of Siegen and the project cooperated with IECAS, China to carry out this research work and finish my dissertation. Also special thanks to my group leader Dr. Holger Nies for having research discussions as well as taking care of lots of administrative things, Mrs. Silvia Niet-Wunram for helping me to register at the university and taking charge of travel arrangements, reimbursements, etc. Also thanks to other colleagues and friends: Amaya, Qurat, Florian, Simon and Simon, Jingshan, Xiaolan, Ling, Junchuan, Wennan, Yuhong, Cathy, Rainer, etc.

Moreover, I would also like to thank Prof. Mihai Datcu and Prof. Peter Reinartz to accept me as a guest and provide me a working place at the Remote Sensing Institute in DLR, so that I could continue my research and cooperate with other colleagues who are experts in the field. Many thanks to Sabine for helping me in a lot of administrative stuff in DLR, also thanks to Deniz, Shiyong, Daniela, Ambar, Reza, Kevin, Jiaojiao, Xiangyu, Nina, Mayte, Song, Jian, etc. Particularly, I would like to thank my colleagues Gottfried and Octavian for scientific and technological discussions, regarding remote sensing background, etc. I would also thank Dr. Pierre Blanchart from CNES and was also with the team before sharing his first prototype system of non-locality map generation.

Besides, friendship and the nice communicating with colleagues have also played an important role during my research life. Hence, I would like to express my best thankfulness and wishes to: Marion from KIT, Fang from Gilching, Ning from Uni Siegen, Chen from Uni Siegen, Meng from Huawei Düsseldorf, Jiaxing from Uni Siegen; Johannes and Ingrid from Seefeld Hechendorf who have been my landlord and the wife of my landlord and have helped me to get used to the local life in Bavaria, Germany.

In the end, I would like to express my sincere thankfulness to my loving parents, they are simply great. No matter what kind of difficulties I've encountered, when I got stuck without progress or even failed, they supported and encouraged me as always. Ever since I was a kid, they educated and parented me from primary school through Bachelor and Master; now their love and strict upbringing have nourished me till today to finish this PhD dissertation.

Zusammenfassung

Mit den vielen hochauflösenden SAR- (Radar mit synthetischer Apertur) und optischen Satelliten, die sich im Orbit befinden, werden auch die zugehörigen Bildarchive ständig größer und aktualisiert, da täglich neue hochaufgelöste Bilder aufgenommen werden. Daraus ergeben sich neue Perspektiven und Herausforderungen an eine automatische Interpretation von hochaufgelösten Satellitenbildern zur detaillierten semantischen Annotation und Objekt-Extraktion. Dazu kommt, dass das florierende Gebiet des maschinellen Lernens die Leistungskraft von Computer-Algorithmen gezeigt hat, die ihre "Intelligenz" zur Lösung zahlreicher und verschiedenartiger Anwendungsfälle (wie visuelle Objekterkennung, inhaltsbasierte Bildsuche etc.) bereits allgemein demonstriert haben. Allerdings können die vorgeschlagenen und bereits existierenden Methoden momentan nur eine begrenzte Anzahl von Bildern verarbeiten. Daher wird in dieser Dissertation versucht, Informationen aus großen Mengen von Satellitenbildern zu extrahieren. Wir bieten Lösungen zur halbautomatischen Interpretation von Satellitenbildinhalten auf der Ebene von Bild-ausschnitten und von Pixeln, bis hin zur Objekt-Ebene mit hochaufgelösten Bildern von TerraSAR-X und WorldView-2. Hierbei wird das Analyse-Potential von nicht überwachten Lernverfahren zur Verarbeitung von großen Datenmengen genutzt.

Bei großen Datenmengen versuchen unsere Lösungen, die Problemstellung in einem ersten Schritt aufgrund einer einfachen Annahme zu vereinfachen. Wir nehmen eine Gauß-Verteilung an, um Bild-Cluster zu beschreiben, die sich aus einer Clustering-Methode ergeben. Basierend auf bereits aufgeteilten Clustern von Bildausschnitten schlagen wir für die semantische Annotation von großen Datensätzen eine teil-überwachte Cluster und Klassifizierungs-Vorgehensweise vor.

Dazu entwerfen wir ein Mehr-Ebenen-System, das eine gute Möglichkeit bietet, Bildinhalte von drei Gesichtspunkten her zu beschreiben. Der erste Punkt bezieht sich auf Bildausschnitte in einer hierarchischen Baumstruktur, wo ähnliche Ausschnitte zusammengelegt werden. Der zweite Punkt beschreibt die Bildhelligkeiten und SAR-Speckle-Details, um eine Klassifizierung auf Pixelebene für allgemeine Landbedeckungskategorien zu erhalten. Der dritte Punkt erlaubt die Interpretation auf Objektebene. Dafür werden Informationen zur Ortszuordnung sowie zur ähnlichkeit zwischen Komponenten berücksichtigt. Zur iterativen Berechnung der sog. "Non-Locality"-Karte, die zur Objektextraktion genutzt wird, wurde ein Konzept zum aktiven Lernen auf der Basis einer Support-Vector-Maschine (SVM) implementiert.

Eine weitere Nutzung unseres Ansatzes kann über eine hierarchische Struktur für SAR- und optische Daten erfolgen, nämlich in der Art, wie die Bildinterpretationen auf Ausschnittsebene, Pixelebene und Objektebene untereinander

verbunden werden. Infolgedessen kann man, von einem Vollbild ausgehend, allgemeine und detaillierte Informationsebenen extrahieren. Das Zusammenfügen von verschiedenen Ebenen hat bereits zu vielversprechenden Ergebnissen bei der automatischen semantischen Annotation für große Mengen von hochaufgelösten Satellitenbildern geführt. Diese Dissertation zeigt auch, bis zu welchem Grad Informationen aus jeder Datenquelle extrahiert werden können.

Abstract

With a number of high-resolution Synthetic Aperture Radar (SAR) and optical satellites in orbit, the corresponding image archives are continuously increasing and updated as new high-resolution images are being acquired everyday. New perspectives and challenges for the automatic interpretation of high-resolution satellite imagery for detailed semantic annotation and object extraction have been raised up. What's more, the booming machine learning field has proved the power of computer algorithms by presenting the world their "intelligence" to solve numerous and diverse applications, visual object recognition, content-based image retrieval, etc. However, till now, the proposed and already existing methods are usually able to process only a limited amount of images. Hence, this dissertation tries to extract information from large amounts of satellite imagery. We provide solutions for the semi-automatic interpretation of satellite image content from patch-level and pixel-level to object-level, using the high-resolution imagery provided by TerraSAR-X and WorldView-2. The mining potential of unsupervised learning methods is utilized for the processing of large amounts of data.

With large amounts of data, our solutions try to simplify the problem at the first step based on a simple assumption. A Gaussian distribution assumption is applied to describe image clusters obtained via a clustering method. Based on the already grouped image patch clusters, a semi-supervised cluster-then-classify framework is proposed for the semantic annotation of large datasets.

We design a multi-layer scheme that offers a great opportunity to describe image contents from three perspectives. The first perspective represents image patches in a hierarchical tree structure, similar patches are grouped together, and are semantically annotated. The second perspective characterizes the intensity and SAR speckle information in order to get a pixel-level classification for general land cover categories. The third perspective allows an object-level interpretation. Here, the information of location and similarity among elements are taken into account, and an SVM-based active learning concept is implemented to update iteratively the so-called "non-locality" map which can be used for object extraction.

A further exploitation of our approach could be to introduce a hierarchical structure for SAR and optical data in the way the patch-level, pixel-level and object-level image interpretation are connected to each other. Hence, starting from a whole scene, general and detailed levels of information can be extracted. Such fusions between different levels have achieved promising results towards an automated semantic annotation for large amounts of high-resolution satellite images. This dissertation also demonstrates up to which level information can be extracted from each data source.

Keywords: High-resolution TerraSAR-X data, high-resolution WorldView-2 data, patch-level image interpretation, pixel-level SAR image interpretation, object-level optical image interpretation, hypothesis test, speckle statistics, Bayesian model, active learning.

Contents

Contents ix

List of Figures xiii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Main Goals of This Work . . . 4
- 1.3 Contributions of the Dissertation . . . 4
- 1.4 Outline of the Dissertation . . . 5

2 Data Characteristics and Basic Mathematics 7
- 2.1 Data Characteristics . . . 7
  - 2.1.1 Introduction to Remote Sensing . . . 7
  - 2.1.2 Synthetic Aperture Radar . . . 8
    - 2.1.2.1 Radar Principle . . . 8
  - 2.1.3 SAR Statistical Properties . . . 10
    - 2.1.3.1 Speckle Effect . . . 10
    - 2.1.3.2 Statistical Properties of the Backscattered Signal . . . 10
  - 2.1.4 Speckle Reduction . . . 11
    - 2.1.4.1 Multi-Look Processing . . . 11
- 2.2 Basic Mathematics . . . 12
  - 2.2.1 Probability Distributions . . . 12
    - 2.2.1.1 Copula-based Joint Probability Modeling . . . 14
    - 2.2.1.2 Gaussian Mixture Models (GMM) . . . 14
    - 2.2.1.3 Bayes' Rule . . . 15
  - 2.2.2 Parameter Estimation Methods . . . 15
    - 2.2.2.1 Method of Moments (MoM) . . . 15
    - 2.2.2.2 Method of Maximum Likelihood Estimation (MLE) . . . 15
    - 2.2.2.3 Method of Maximum A Posterior Estimation (MAP) . . . 16
    - 2.2.2.4 Method of Log-Cumulants (MoLC) . . . 16
    - 2.2.2.5 Expectation Maximization (EM) . . . 17
  - 2.2.3 Numerical Optimization Methods . . . 18
    - 2.2.3.1 Gradient Descent . . . 18
    - 2.2.3.2 Newton's Method . . . 19
    - 2.2.3.3 Stochastic Gradient Descent . . . 19
  - 2.2.4 Sampling Methods . . . 19
    - 2.2.4.1 Markov Chain Monte Carlo (MCMC) . . . 19

3 State of the Art **21**
3.1 Earth Observation Meets Computer Vision . . . 22
3.2 Hierarchical Representation . . . 23
3.2.1 Feature Hierarchy . . . 24
3.2.2 Semantic Hierarchy . . . 25
3.3 Description of Images . . . 25
3.3.1 Feature Extraction . . . 26
3.3.1.1 Multi-Spectral Information . . . 27
3.3.1.2 Textural Information . . . 28
3.3.1.3 Geometric Information . . . 29
3.3.2 Feature Encoding . . . 30
3.3.3 The Curse of Dimensionality . . . 31
3.3.4 Distance Metrics . . . 32
3.3.4.1 Fractional and Minkowski Distances . . . 33
3.3.4.2 Distance Metric Learning . . . 33
3.4 Machine Learning . . . 34
3.4.1 Classic Machine Learning Methods . . . 35
3.4.2 New Trends in Semi-supervised Learning . . . 35
3.4.3 Object Extraction-based Semantic Exploration . . . 38
3.5 Conclusions and Proposed Concepts . . . 40
3.5.1 Applied Dataset . . . 42
3.5.2 Semi-supervised Learning . . . 43

4 Application and Evaluation of a Hierarchical Patch Clustering Method for Image Patches **49**
4.1 Approach . . . 49
4.2 Methodology . . . 51
4.2.1 Feature Extraction . . . 52
4.2.2 Hierarchical Clustering . . . 52
4.2.3 Modified G-means Algorithm . . . 53
4.2.3.1 Gaussian Hypothesis Testing . . . 53
4.2.3.2 Anderson-Darling Test . . . 53
4.2.3.3 Feature Vector Projection . . . 54
4.2.4 Comparative Similarity Measures . . . 55
4.2.4.1 Fractional Distance Metric . . . 55
4.2.4.2 Minkowski Distance Metric . . . 56
4.2.5 Evaluation . . . 56
4.2.5.1 Visual Evaluation . . . 56
4.2.5.2 Internal Evaluation . . . 56
4.2.5.3 External Evaluation . . . 57
4.2.6 Comparative Clustering Methods . . . 57
4.3 Results . . . 58
4.3.1 Datasets . . . 58
4.3.2 Experimental Settings . . . 60
4.3.3 Parameter Settings . . . 60
4.3.4 Visual Evaluation . . . 61
4.3.5 Internal Evaluation . . . 62
4.3.6 External Evaluation . . . 63
4.3.6.1 Analysis of Absolute Homogeneity . . . 64

4.3.6.2 Analysis of Relative Homogeneity . . . 64
4.3.6.3 Analysis of Cluster Numbers . . . 65
4.3.7 Comparative Experiments . . . 66
4.3.7.1 Different Clustering Methods . . . 66
4.3.7.2 Different Features . . . 66
4.4 Conclusions . . . 69

5 Semi-supervised Semantic Image Patch Annotation 71
5.1 Methodology . . . 71
5.1.1 Creation of a Reference Dataset . . . 73
5.1.2 Semi-supervised Learning . . . 74
5.1.2.1 Cluster-then-Label . . . 74
5.1.2.2 Supervised Learning Within Clusters . . . 74
5.1.3 K-Medoids Algorithm Implementation . . . 75
5.1.4 Evaluation . . . 75
5.1.4.1 Quantitative Evaluation . . . 75
5.1.4.2 Visual Evaluation . . . 76
5.2 Results . . . 77
5.2.1 Image Data Selection and Subsampling . . . 77
5.2.1.1 Data Selection . . . 77
5.2.1.2 Data Pre-Processing . . . 77
5.2.2 Parameter Settings . . . 79
5.2.3 Quantitative Evaluations . . . 79
5.2.3.1 Internal Evaluation . . . 79
5.2.3.2 External Evaluation . . . 82
5.2.3.3 Additional Evaluations . . . 85
5.2.4 Visual Evaluations . . . 87
5.2.4.1 Tree Structure . . . 87
5.2.4.2 Feature Space Visualization . . . 87
5.2.4.3 Cluster Centroid Patches . . . 88
5.2.4.4 Cluster Homogeneity . . . 91
5.3 Conclusions . . . 94
5.3.1 Clustering . . . 94
5.3.2 Classifiers . . . 94
5.3.3 Semi-supervised Learning and Manually Annotated Reference Data . . . 95
5.3.4 Semi-Annotation/Labeling . . . 95

6 Pixel-Level Bayesian Classification and Active Learning Based Object Extraction 97
6.1 Pixel-Level Bayesian Classification . . . 97
6.1.1 Modeling of Speckle Statistics Feature . . . 98
6.1.2 Image Intensity Modeling . . . 99
6.1.3 Combined Intensity - Speckle Statistics Feature Model . . . 99
6.1.4 Bayesian Classification . . . 100
6.1.5 Experiments . . . 101
6.1.5.1 Brief Dataset Description . . . 101
6.1.5.2 Evaluation . . . 102
6.2 Active Learning Based Object Extraction . . . 108
6.2.1 Definition of Non-Locality . . . 108
6.2.2 SVM-based Active Learning . . . 109

6.2.2.1 Version Space . . . 109
6.2.2.2 Sample Selection Strategies . . . 109
6.2.2.3 Prototype Implementation . . . 110
6.2.2.4 Evaluation and Discussion . . . 111
6.3 Summary . . . 120

7 Conclusions **121**
7.1 Summary . . . 121
7.2 Future Works . . . 123

Appendix A **125**
A.1 Summary of Machine Learning Algorithms . . . 125
A.2 Support Vector Machines (SVMs) . . . 130
A.2.1 Linearly Separable Binary Classifications . . . 130
A.2.2 Non-Linearly Separable Classifications . . . 132
A.2.3 Application . . . 133

Appendix B **135**
B.1 Extraction of Common Objects . . . 135
B.2 Extraction of Specific Objects . . . 140

References **145**

List of Figures

1.1 TerraSAR-X image interpretation of Siegen city. 2
1.2 Worldview-2 image interpretation of Munich city. 3
1.3 First and second contributions of the dissertation. 6
1.4 First part of the fourth contribution 6
1.5 Second part of the fourth contribution 6

2.1 The electromagnetic spectrum and the atmospheric transmittance of a clear cloud-free atmosphere. 8

3.1 Hierarchical representations from semantic and feature perspectives. 24
3.2 Description and analysis of images. 26
3.3 Active learning scheme vs. passive learning scheme. 35
3.4 Semantic annotation framework from patch-level and pixel-level perspectives. 41
3.5 Geographical distribution of our semantically annotated TerraSAR-X data set. 43
3.6 Unsupervised clustering for TerraSAR-X image patches. 45
3.7 Proposed processing and analysis procedure for image patches. 46
3.8 Proposed processing and analysis procedure for object extraction. 47

4.1 Proposed data clustering and analysis scheme for image patches. 51
4.2 An ensemble of odd and even Gabor filters 52
4.3 Feature projection based on k-means clustering, a Gaussian-test will be used later during the construction of a hierarchical clustering structure. 55
4.4 Patch examples for three classes of dataset 1. 58
4.5 Patch examples for three classes of dataset 2. 59
4.6 Patch examples for three classes of dataset 3. 59
4.7 Screen shot of patches as an example of a cluster from dataset 1. 61
4.8 Screen shot of patches as an example of a cluster from dataset 2. 61
4.9 Visualization of patches as an example of a cluster from dataset 3. 62
4.10 RMSSTD index versus distance parameter for three datasets. 62
4.11 RS index versus distance parameter for three datasets. 63
4.12 Homogeneity of the whole dataset. 64
4.13 Homogeneity percentage of different cluster types. 65
4.14 Number of clusters for dataset 1 and 2. 66
4.15 RMSSTD index versus different clustering methods for three datasets. 67
4.16 RS index versus different clustering methods for three datasets. 67
4.17 RMSSTD index versus distance parameter for dataset 3. 68
4.18 RS index versus distance parameter for dataset 3. 68
4.19 Cluster numbers versus distance parameter for dataset 3. 69

5.1 Proposed data classification and analysis scheme for image patches. 72
5.2 Examples of annotated classes. 74
5.3 Examples of scenes from Russia. 77
5.4 Examples of scenes from German speaking countries. 78
5.5 Examples of scenes from North America. 78
5.6 Examples of scenes from Africa. 78
5.7 Accuracy of detailed level classifications for the modified G-means algorithm for the data collection 17. 80
5.8 Micro-average F-score of detailed level classifications for the modified G-means algorithm for the data collection 17. 80
5.9 Accuracy of detailed level classifications for the original G-means algorithm for the data collection 17. 81
5.10 Micro-average F-score of detailed level classifications for the original G-means algorithm for the data collection 17. 81
5.11 Accuracy of general level classifications for the modified G-means algorithm for the data collection 17. 83
5.12 Micro-average F-score of general level classifications for the modified G-means algorithm for the data collection 17. 84
5.13 Accuracy of general level classifications for the original G-means algorithm for the data collection 17. 84
5.14 Micro-average F-score of general level classifications for the original G-means algorithm for the data collection 17. 85
5.15 Evaluation of data collection 17. 86
5.16 Cluster tree structure of the data collection 17. 87
5.17 3D feature space of the data collection 17 with general level reference data. . 88
5.18 3D feature space of the data collection 17 with detailed level reference data. . 89
5.19 3D feature space of the data collection 17 with a distance parameter of 1. . . 89
5.20 Patch examples of the most compact cluster (collection 17). 90
5.21 Patch examples of the mid-compact cluster (collection 17). 90
5.22 Patch examples of the most spread out cluster (collection 17). 90
5.23 Cluster splitting for the data collection 17 with the general level reference data. 91
5.24 Cluster splitting for the data collection 17 with the detailed level reference data. 92
5.25 The class-cluster distribution of data collection 17 with general level reference data. 93
5.26 The class-cluster distribution of data collection 17 with detailed level reference data. 93

6.1 Urban area, $\mathcal{G}^0$ distributions for intensities and the $Ctm2$ feature. 102
6.2 Forest area, $\mathcal{G}^0$ distributions for intensities and the $Ctm2$ feature. 103
6.3 Urban area, $\mathcal{G}^0$ distribution for intensities, and Weibull distribution for the $Ctm2$ feature. 103
6.4 Forest area, $\mathcal{G}^0$ distribution for intensities, and Weibull distribution for the $Ctm2$ feature. 104
6.5 TerraSAR-X GEC product of Aachen city. 106
6.6 Ground truth reference map of Aachen city. 106
6.7 Classification result of Aachen city. 107
6.8 Pseudo-color image of Aachen city. 107
6.9 Active learning based object extraction for optical imagery. 108
6.10 The interface of the active learning based object extraction method. 110

6.11 Training sample selection of the active learning based object extraction method. 111
6.12 WorldView-2 data of Munich city. 112
6.13 Iterations of training samples and non-locality maps for a *river* object. 113
6.14 Non-locality map of a *river* superimposed on a water reference. 114
6.15 Iterations of training samples and non-locality maps for a *tennis court* object. . 115
6.16 Non-locality map of a *tennis court* superimposed by the original image. 116
6.17 Iterations of training samples and non-locality maps for a *gray round building* object. 117
6.18 Non-locality map of a *gray round building* superimposed by the original image. 118
6.19 Iterations of training samples and non-locality maps for a *blue corner building* object. 119

7.1 Detailed level openstreetmap ground truth of Aachen city. 124

A.1 The separating hyperplane of a Support Vector Machine. 131

B.1 Iterations of training samples and non-locality maps for *railway* objects. . . . 136
B.2 Iterations of training samples and non-locality maps for *red building* objects. . 137
B.3 Iteration of training samples and non-locality maps for *trees* objects. 138
B.4 Iterations of training samples and non-locality maps for *white square building* objects. 139
B.5 Iterations of training samples and non-locality maps for *sports field* objects. . . 140
B.6 Iterations of training samples and non-locality maps for a *bridge* object. 141
B.7 Iterations of training samples and non-locality maps for a *round-about* object. 142
B.8 Iterations of training samples and non-locality maps for a *pond* object. 143

Chapter 1

Introduction

> The only true wisdom is knowing that you know nothing.
>
> Socrates

1.1 Motivation

Nowadays, Earth Observation (EO) from space is becoming an increasingly booming and promising scientific research area. According to the Union of Concerned Scientists (UCS) database, till the end of January 2014, there were already 45 organisations with 192 EO satellites in space. Nevertheless, more and more EO satellites are launched into space. For example, until now, there are 20 Chinese Earth observation satellites already in orbit. In the commercial or civilian remote sensing industry, satellite imaging instruments with varying spatial resolutions have been launched into space.

As a consequence, besides already existing large image archives, tens of thousands of new images of various resolutions, imaging modes (active and passive imaging, e.g., SAR and optical) and processing products (e.g., SSC, MGD, GEC modes of TerraSAR-X products Portal [2016], multi-spectral, hyper-spectral and panchromatic products) are also acquired and processed everyday. Simultaneously, the fast development of new processing and analyzing algorithms in machine learning and computer vision (e.g., complex scene understanding, and object extraction) have achieved amazing results. For example, convolutional neural networks (CNNs) with a deep hierarchical structure consisting of several layers have yielded excellent results outperforming human beings on face and cat recognition Khaligh-Razavi [2014], Bengio et al. [2014], Le et al. [2012], by training millions of images.

On the contrary, the interpretation of satellite imagery has not yet reached a very detailed and satisfying level, due to the large amount of data as well as the image content complexity. Hence, motivated by new image data and promising algorithmic methods, and also provoked by the inherent nature of human curiosity, we are no longer content with basic level satellite image analysis and applications in low resolutions (e.g., general land cover classification), but eager to extract new information from high-resolution imagery. Fig. 1.1 and Fig. 1.2 provide possible interpretation schemes for high-resolution satellite data through patch-level semantic annotation and object-level semantic recognition.

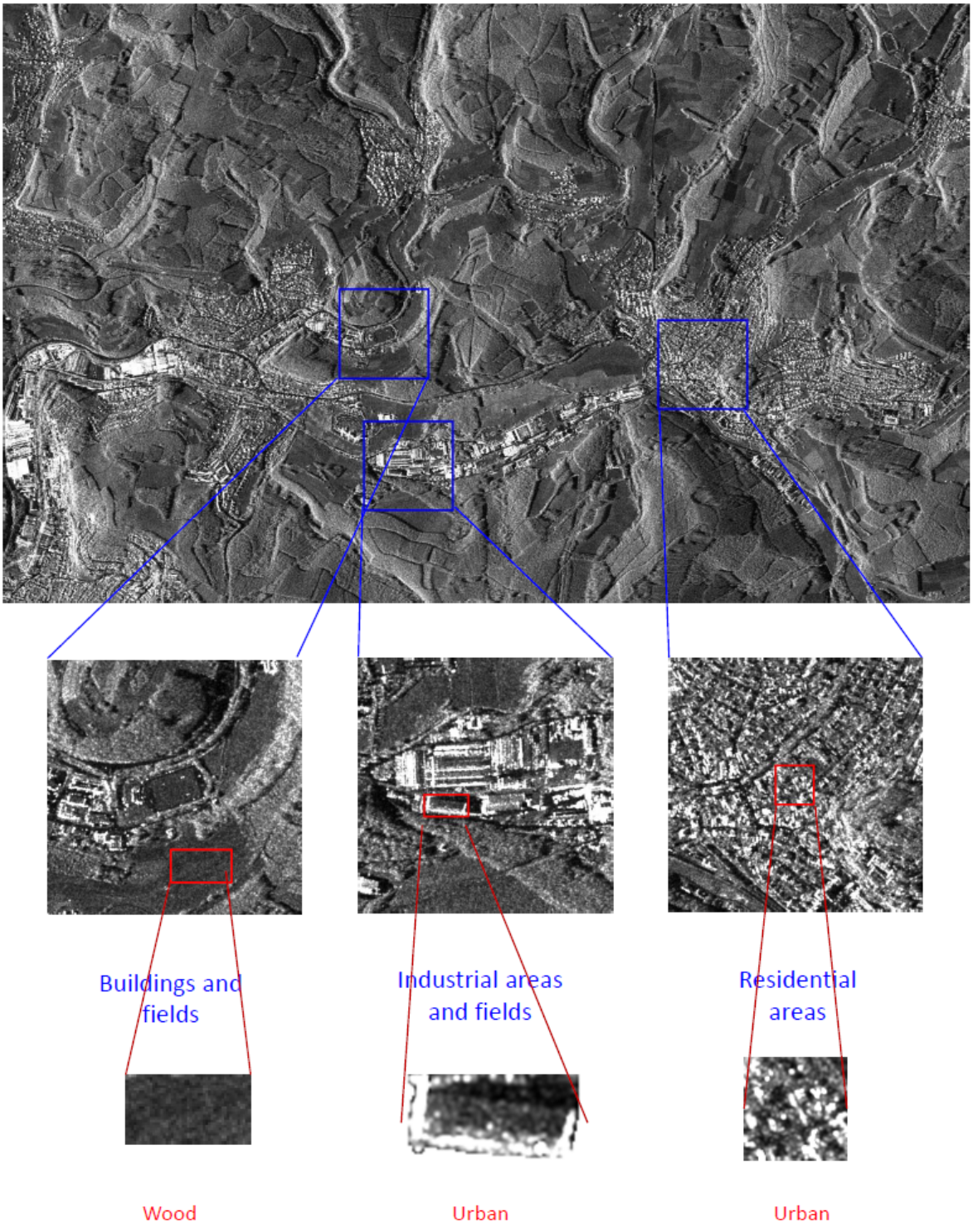


Figure 1.1: TerraSAR-X image interpretation of Siegen city. Three example patches are shown: buildings and fields, industrial areas and fields, and residential areas. Within the patches, forest and urban categories can be classified in pixel-level.

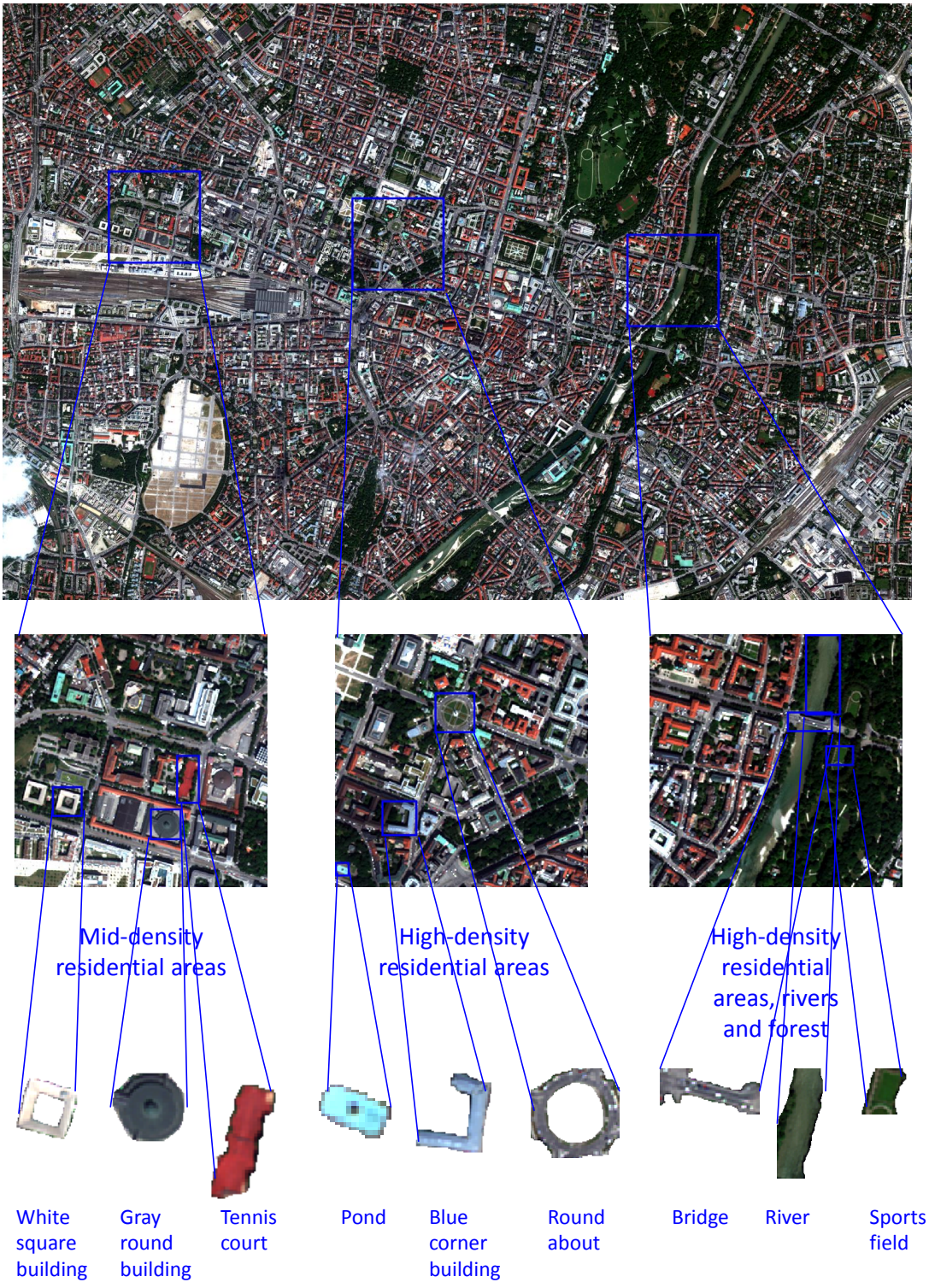


Figure 1.2: Worldview-2 image interpretation of Munich city. Three example patches are shown: middle density residential areas, high density residential area, and high density residential areas, river and forest. Within the patches, various objects can be extracted: white square building, gray round building, tennis court, pond, blue corner building, etc.

1.2 Main Goals of This Work

In this dissertation, the overall objective is to provide solutions for patch-level and pixel-level semi-automatic interpretation and mapping of image content using high-resolution SAR and optical satellite imagery, e.g., TerraSAR-X and WorldView-2 products.

Behind this general objective lie four important goals:

- To represent image patches as a hierarchical structure, and describe image contents with different distance metrics in high-dimensional feature space.
- To develop a patch-level semi-supervised clustering-then-classification method, with respect to two levels of semantic categories, i.e., general and detailed level semantic labels.
- To develop a pixel-level classification method which is based on joint probability distributions and a Bayesian classifier for SAR imagery.
- To develop an object-level extraction method which is based on a non-locality concept and SVM-based active learning for optical imagery.

1.3 Contributions of the Dissertation

The **first contribution** of this dissertation lies in an adaptation and a deep analysis of a hierarchical clustering method for patch-level high-resolution satellite image clustering. In order to simplify the image content interpretation by grouping similar patterns together at the first step, the Gaussian-test-based hierarchical patch clustering method is adapted and modified with a suitable size threshold to get meaningful clusters. The evaluation of the **cluster homogeneity** is qualitatively and quantitatively analyzed in three ways: internal and external evaluation, as well as visual evaluation. The experimental results confirmed that the method is able to obtain homogeneous clusters.

A further exploitation of our approach would be to further process the grouped clusters. In fact, when the first step of problem simplification goal is achieved, the goal of image content interpretation for large-scale image data is not so complex as in the beginning. This introduces our **second contribution** which generates **a semi-supervised learning scheme** by implementing supervised learning within obtained clusters. In here, different feature descriptors, and different classification methods are discussed.

The **third contribution** of this dissertation consists in the evaluation of the proposed semi-supervised methodology using **different distance metrics** under the concept of the curse of dimensionality regarding high-dimensional feature vectors. In fact, although we usually take the Euclidean distance for granted, however, the different distance metrics can shape different feature spaces, so that they have different geometrical characteristics.

Due to the different data characteristics, the **fourth contribution** of this dissertation is the attempt to reach **pixel-level classification** using TerraSAR-X imagery for general semantic categories. It uses a copula function to incorporate image intensity and image feature information to obtain a joint probability distribution which is used as a likelihood PDF in a Bayesian classifier. The next attempt is to obtain **object-level extraction** using WorldView-2 imagery for detailed semantic categories based on the non-locality concept and an SVM-based active learning scheme. The experimental results confirm that currently it is not easy to extract detailed-level categories from SAR imagery; however, it is a reachable goal for optical imagery.

1.4 Outline of the Dissertation

Chapter 2 deals with the data characteristics and preliminary mathematics. Starting from an energy source, the electromagnetic spectrum, and how energy sources interact with objects, the basic characteristics of synthetic aperture radar and optical images are explained and compared. For synthetic aperture radar images, there is a difficult phenomenon that we need to pay attention to: the speckle effect. Then fundamentals of SAR image intensity distribution models and mathematical tools for parameter estimation, numerical optimization, and sampling methods, which are important for machine learning algorithms, are presented.

In Chapter 3 the problems of how to represent the image information and how to describe image contents will be answered. Respectively, a hierarchical representation structure, various feature descriptors, and the influence of different distance metrics in high-dimensional feature space will reviewed. The current trend of machine learning, especially semi-supervised learning, active learning, and object extraction will be discussed. Moreover, the classic machine learning algorithms are summarized into tables in Appendix A. In the end, our proposed concept and the processing framework will be illustrated and explained.

Coming to Chapter 4, a modified Gaussian-test-based hierarchical clustering method is applied and evaluated for high-resolution satellite images. The purpose is to obtain homogeneous clusters within a hierarchical clustering structure which later allow the classification and annotation of image data ranging from single scenes up to large satellite data archives. After cutting a given image into small patches and feature extraction from each patch, k-means is used to split sets of extracted image feature vectors to create a hierarchical structure. As image feature vectors usually fall into a high-dimensional feature space, we test different distance metrics, in order to tackle the curse of dimensionality problem. By using three different SAR and optical image datasets, Gabor texture and Bag-of-Words (BoW) features are extracted, and the clustering results are analyzed via visual and quantitative evaluations. The approach is compared with other classic unsupervised clustering methods.

Based on the evaluations in Chapter 4, Chapter 5 proposes a semi-automated hierarchical clustering and classification framework for SAR image annotation. As an updated framework, it comprises three stages: Firstly, each image is cut into patches and each patch is transformed into a texture feature vector. Secondly, similar feature vectors are grouped into clusters, where the number of clusters is determined by repeated cluster splitting to optimize their Gaussianity. Finally, the most appropriate class (i.e., a semantic label) is assigned to each image patch. This is accomplished by semi-supervised learning. Various validation methods have been used to test and evaluate the proposed framework.

Chapter 6 tries to solve the problem from the pixel-level perspective. Although it is difficult to perform pixel-level classification for SAR imagery, we try to model joint density distributions of different features for general categories: urban, water, etc. Then a Bayesian classifier is used to assign semantic labels to the category that obtains the largest posterior probability. Later, an active learning method based on the non-locality concept is discussed for optical images to extract more detailed objects: buildings, rivers, tennis courts, etc. In order to allow interactive user operations, an SVM-based active learning scheme will be applied.

Finally, Chapter 7 gives some conclusions summarizing the main experimental results.

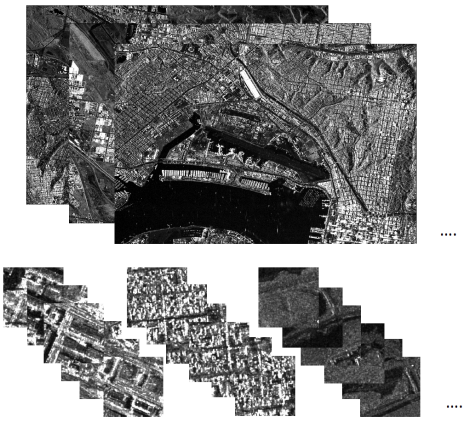


Figure 1.3: **First contribution** of the dissertation: patch-level hierarchical clustering for TerraSAR-X imagery. **Second contribution** of the dissertation: semi-supervised semantic annotation for TerraSAR-X image patches.

(a) Ground truth @Openstreetmap.

(b) Original SAR image.

(c) Final interpretation map.

Figure 1.4: **First part of the fourth contribution** of the dissertation: pixel-level classification for TerraSAR-X images.

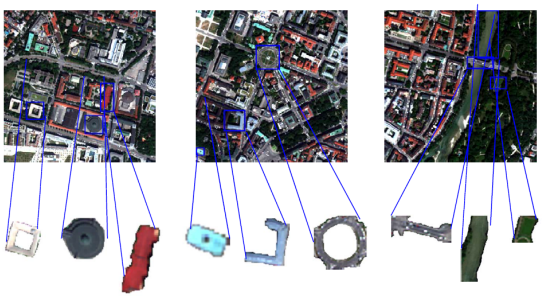


Figure 1.5: **Second part of the fourth contribution** of the dissertation: object extraction for WorldView-2 optical images.

Chapter 2

Data Characteristics and Basic Mathematics

> You see, but you do not observe. The distinction is clear.
>
> Sherlock Holmes

2.1 Data Characteristics

In this chapter, typical data characteristics of satellite images and their basic mathematics are presented. Starting from an energy source, we explain and compare the electromagnetic spectrum, the interaction of radiating energy with objects, and the basic characteristics of SAR and optical imaging. For SAR images, there are difficult phenomena that we need to pay attention to: the speckle effect, shadowing and foreshortening phenomena, etc. In addition, we present the fundamentals of SAR image brightness distribution models, mathematical tools for parameter estimation and numerical optimization, and sample generation methods which are important for machine learning algorithms.

2.1.1 Introduction to Remote Sensing

As defined in the Nomenclature chapter, remote sensing is the acquisition of information about a remote object or phenomenon without making real physical contact with it. In many cases, remote sensing from satellites, airplanes or ground-based stations is used to gather information about the physical, chemical and biological systems of the Earth. This field of applications is called Earth observation. Typical examples of remote sensing applications are: air traffic control, meteorological predictions and early warning, generation of topographic maps via stereographic pairs of aerial photographs, measuring land-use change (e.g., deforestation), as well as monitoring and responding to natural disasters (e.g., fires, floods, earthquakes, etc.). In order to accomplish these tasks, different data sources can be used separately or fused to achieve the expected results. Data processing techniques as well as state-of-the-art theories and algorithms (e.g., image processing, machine learning, etc.) from other research areas are used. In this dissertation, we will mostly rely on state-of-the-art machine learning algorithms to process high-resolution satellite images and generate semantic image annotations.

When we look at our potential energy sources, there are two categories of sensors - passive ones and active ones. As the Sun is a continuous energy source, the majority of

remote sensing data is obtained with **passive sensors**, e.g., our everyday photos, and images taken by airborne or space-borne sensors. Besides sunshine as the main energy source for passive imaging, thermal infrared and passive microwave sensors also measure the natural Earth's energy emissions. In contrast, **active sensors** contain their own source of energy. In the field of remote sensing, the best examples are radar and SAR, which emit energy in the microwave range of the electromagnetic spectrum. The energy being reflected by a selected area of the Earth's surface and acquired by the remote sensing instrument is recorded as a sensed image of that area.

As shown in Fig. 2.1 Drinkwater [2008], with different electromagnetic wavelengths, various kinds of remote sensing images (e.g., SAR, panchromatic, multispectral and hyperspectral images) can be obtained. When electromagnetic waves reach a material, depending on the characteristics of the material, three types of mostly partial interaction will occur: reflection, absorption, and transmission.

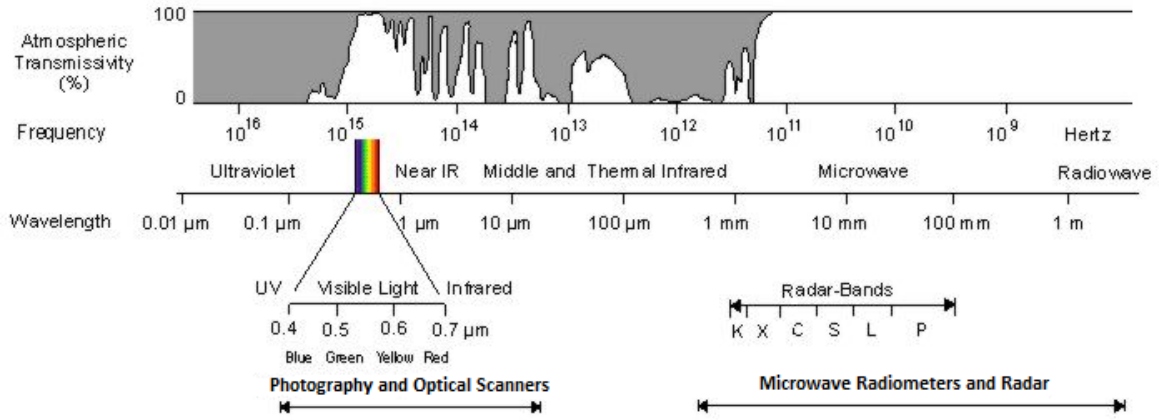


Figure 2.1: The electromagnetic spectrum and the atmospheric transmittance of a clear cloud-free atmosphere. Source: with modifications from Albertz [2007].

As the atmosphere scatters and absorbs radiation during its passage from the Sun to the Earth's surface and from the Earth's surface to the sensor, the wavelength ranges of the electromagnetic spectrum in which the transmission of radiation through the atmosphere is high are known as the "atmospheric windows". Hence, satellite sensors designed to observe land or water surfaces operate within the atmospheric windows which are shown in Fig. 2.1. For us remote sensing researchers, multispectral and hyperspectral images are acquired from the ultraviolet, visible and infrared spectral ranges, while SAR images are acquired from the microwave spectral range.

2.1.2 Synthetic Aperture Radar

Since one main data source in this dissertation is SAR, we will look in the following subsections into the basic radar principle and SAR statistical properties, as well as speckle effects, and some speckle reduction techniques.

2.1.2.1 Radar Principle

Radar is the abbreviation for **Ra**dio **D**etection **a**nd **R**anging. It is a method developed for target acquisition and distance measurement by means of radio waves or microwave

measurements. Radar systems are active illumination systems which operate in the radio wave or microwave domain, which means that radar systems are independent from solar illumination. Radar benefits us with the following two advantages: 1) its imaging capability through clouds; and 2) its imaging capability at night. Table 2.1 gives an overview of the commonly used radar frequency bands and their parameters.

Table 2.1: Radar frequency bands Ulaby et al. [1981].

Band	**Frequency (GHz)**	**Wavelength (cm)**
K	10.9 - 36.0	2.75 - 0.83
X	5.75 - 10.9	5.21 - 2.75
C	4.2 - 5.75	7.14 - 5.21
S	1.55 - 4.2	19.4 - 7.14
L	0.39 - 1.55	76.9 - 19.4
P	0.225 - 0.39	133 - 76.9

Regarding different radar frequency bands, target visibility is the key factor for the wavelength selection. While a long wavelength allows a penetration into vegetation and soil, a small one is more sensitive to surface roughness. Hence, intermediate bands (L, C and X) have been mostly used by radar systems. Moreover, X-band is a suitable frequency-band for high-resolution radar applications, e.g., urban sensing applications. The benefits of X-band lie in its improved resolution, but it also shows a good response to most land cover signatures. A typical X-band instrument is the German TerraSAR-X system launched in June, 2007, with a SAR sensor that can be operated in different modes and polarizations Portal [2016].

SAR systems are special systems which have been developed to overcome the limitations of systems with real aperture concerning their azimuth (i.e., along-track) resolution. In order to improve their azimuth resolution, SAR systems electronically synthesize an extremely long antenna or aperture by using the forward motion of a small antenna and a special recording and processing of the backscattered echoes.

This can be accomplished by an airborne or space-borne SAR instrument that continuously transmits a time-synchronized sequence of pulses; their received echoes are sampled regularly, and recorded on-board; then the data are compressed and downlinked to ground and, during routine processing, sharp ("focused") images are generating via an inverse Fourier transformation. The Fourier transformation generates complex-valued image pixels. These complex-valued samples span a large dynamic range and their magnitudes are called "detected" pixels; in order to reduce the dynamic range of the SAR data, one often converts the initial "intensities" into "amplitudes" via square rooting. Thus, one has to be careful, what product types are available.

In addition, based on SAR image analysis and observations, here are some properties of basic SAR image characteristics: Large incidence angles (shallow angles) result in smoother images and better range resolution Oliver and Quegan [1998]. Buildings often cause multiple reflections: sequential ground-wall or wall-ground reflections induce multi-bounce effects. Different surface types have different backscattering effects, due to smooth or rough surface structures, or different textures.

2.1.3 SAR Statistical Properties

As a coherent imaging system, SAR systems generate images which suffer from speckle. Basically, speckle is a salt-and-pepper pattern in radar imagery due to the coherent nature of the radar wave, which causes random constructive and destructive interference among the electromagnetic waves which are reflected from different scatterers within the imaged footprint area of each pixel. This phenomenon becomes visible only in magnitude ('detected') images, not in complex-valued SAR images. As a consequence of this phenomenon, the interpretation of detected SAR images may become severely disturbed. Hence, many previous research works focus on the topic of speckle reduction Oliver and Quegan [1998].

In this section, the physical origins of the speckle effect as well as its statistical properties are discussed. Afterwards, speckle reduction is briefly investigated although in high-resolution images where we have a lower number of scatterers per pixel, the speckle, which therefore is only partly developed, is less disturbing than in low-resolution images Oliver and Quegan [1998].

2.1.3.1 Speckle Effect

Although the true physical interactions among electromagnetic waves, surfaces and sensors are very complex, a simple multiplicative model is normally applied to explain, for instance, how the Earth's surface appears in a SAR satellite image. For distributed targets, each resolution cell can be considered to contain a number of N discrete scatterers. When an electromagnetic wave interacts with a given target, each scatterer k contributes a reflected wave with an amplitude A_k and a phase ψ_k. Hence, the total reflected return of the incident wave is:

$$u = Ae^{i\psi} = \sum_{k=1}^{N} A_k e^{i\psi_k} \tag{2.1}$$

The summation is made over the number of scatterers which are illuminated.

2.1.3.2 Statistical Properties of the Backscattered Signal

As indicated by Formula 2.1, for large numbers of scatterers N, several properties of the received signal can be derived Oliver and Quegan [1998].

- Due to the Central Limit Theorem, the in-phase and quadrature components of a complex-valued SAR image, $u_i = Acos\psi$ and $u_q = Asin\psi$, are independent identically distributed Gaussian random variables with zero mean, and their spatially-dependent variance $\sigma/2$ within a small local window is determined by the scattering amplitudes A_k. The corresponding joint probability density function (pdf) is a circular Gaussian distribution with variance σ:

$$f_{u_i,u_q}(\xi_i,\xi_q) = \frac{1}{\pi\sigma} e^{-\frac{\xi_i^2+\xi_q^2}{\sigma}}. \tag{2.2}$$

- The observed phase ψ is uniformly distributed between $-\pi$ and π.

- The amplitude $A = |u| = \sqrt{u_i^2 + u_q^2}$ follows a Rayleigh distribution:

$$f_A(a) = \frac{2a}{\sigma} e^{-\frac{a^2}{\sigma}}, \quad a \geq 0, \tag{2.3}$$

  with a mean value of $E(A) = \frac{\sqrt{\pi\sigma}}{2}$ and a variance of $var(A) = (1 - \frac{\pi}{4}\sigma)$.

- The intensity or power $I = A^2$ has a negative exponential distribution:

$$f_I(i) = \frac{1}{\sigma} e^{-\frac{i}{\sigma}}, \quad i \geq 0, \tag{2.4}$$

  with a mean value of $E(I) = \sigma$ and a variance of $var(I) = \sigma^2$.

Except for the phase ψ, for other distributions that are characterized by only a single parameter, σ corresponds to the mean average intensity.

The presented speckle model and the proposed distributions are important in handling SAR data as they can be used for target classification and image segmentation. Hence, in this dissertation, most of the experiments are carried out using amplitude or intensity values as our interest lies in the brightness distributions of "detected" single SAR images. A corresponding example will be discussed in the first part of Chapter 6, where we use the statistical properties of SAR images for general land cover classification.

2.1.4 Speckle Reduction

For low to mid SAR image resolution, speckle is an undesirable effect which hinders the image interpretation. Therefore, speckle reduction may become a critical pre-processing step of different applications such as target detection, object recognition and classification, etc. There are two main despeckling techniques: multi-look processing, and adaptive image restoration techniques (i.e., speckle filtering). In this dissertation, several experiments have been performed by using the statistical information of probability density distributions of image intensities or amplitudes, hence some basic relationships will be explained.

2.1.4.1 Multi-Look Processing

The multi-look approach is to average the detected images over several pixels within a sliding window. This process of averaging is known as multi-looking, the resulting images are known as L-looks, where L denotes the number of incoherently summed pixels or looks.

Multi-look intensity images: The L-look average intensity

$$I = \frac{1}{L} \sum_{k=1}^{L} I_k \tag{2.5}$$

is known to obey a Gamma distribution with order parameter L and $I \geq 0$ according to Oliver and Quegan [1998]:

$$f_I(i) = \frac{1}{\Gamma(L)} (\frac{L}{\sigma})^L i^{L-1} e^{-\frac{Li}{\sigma}}, \tag{2.6}$$

where $\Gamma(.)$ is the Gamma function:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt. \tag{2.7}$$

Moreover, the Equivalent Number of Looks (ENL) within a window is defined as:

$$ENL = \frac{(E(I))^2}{var(I)}, \tag{2.8}$$

where $E(I) = \sigma$ and $var(I) = \frac{\sigma^2}{L}$. The *ENL* is equivalent to the number of independent intensity values averaged per pixel.

Square-rooted multi-look intensity images: Chaabouni-Chouayakh [2009] The square root of the multi-look intensity is preferred to use for displaying purposes:

$$S = \sqrt{I} = \sqrt{\frac{1}{L}\sum_{k=1}^{L} I_k}. \tag{2.9}$$

With $\mu_S = \sqrt{\mu_I}$, $S \geq 0$ follows a conditional square-rooted Gamma distribution:

$$f_S(s|\mu_S) = \frac{2}{\Gamma(L)}(\frac{L}{\mu_S^2})^L s^{2L-1} e^{-\frac{Ls^2}{\mu_S^2}}, \tag{2.10}$$

which is a Rayleigh distribution with $L = 1$ and $\mu_S^2 = \mu_I$.

As statistical modeling has served as a useful tool in SAR image analysis, basic and advanced mathematical tools, which are widely used for remote sensing image processing, will be systematically discussed in the following section.

2.2 Basic Mathematics

In order to better explain different mathematical tools, this section is organized to explain classic analytical and numerical tools to deal with large amounts of data. Special attention will be paid to probabilistic and statistical models due to their powerful data modeling capability.

2.2.1 Probability Distributions

From the perspective of **signal processing**, image histograms can be modeled in the standard form of a Gamma distribution:

$$f_{I|\alpha,\beta}(i|\alpha,\beta) = \frac{\beta^\alpha i^{\alpha-1}\exp\{-\beta i\}}{\Gamma(\alpha)}. \tag{2.11}$$

In SAR processing, I stands for the intensity I, α is used as the number of looks L, β is taken as $\frac{L}{\sigma}$, and σ is the standard deviation.

Since the often used Rayleigh distribution and the negative exponential distribution are two special examples of a Weibull distribution with specific parameters, the definition of the Weibull distribution is shown below:

$$f_{I|\alpha,\beta}(i|\alpha,\beta) = \frac{\beta}{\alpha^\beta} i^{\beta-1}\exp\left[-(\frac{i}{\alpha})^\beta\right], i \geqslant 0. \tag{2.12}$$

Here α is used as the scale parameter, and β is the shape parameter. A Weibull distribution can describe single-look SAR images with high accuracy for either amplitude or intensity values; however, it cannot exactly represent multi-look SAR images exactly Gao [2010].

Particularly, for high-resolution SAR images of urban regions, the $\mathcal{G}^0$ distribution, which is equivalent to the Fisher distributions Tison et al. [2004], is usually used as an empirical model:

$$f_{I|\alpha,\beta,L}(i|\alpha,\beta,L) = \frac{L^L\Gamma(L-\beta)}{\alpha^\beta\Gamma(L)\Gamma(-\beta)}\frac{i^{L-1}}{(\alpha+Li)^{L-\beta}}, \tag{2.13}$$

while α stands for the scale parameter, β is the shape parameter, and L is the number of looks.

From the perspective of **machine learning**, the multinomial distribution and the Dirichlet distribution, which are conjugate distributions, are important tools and often used in the framework of Bayesian statistics. A typical example is the generative statistical Latent Dirichlet Allocation (LDA) model that allows observation sets to be explained by unobserved groups where similar data are grouped together.

In contrast, the multinomial distribution denoted by $Mult(p_1,..,p_K,n)$, is a discrete distribution over k-dimensional non-negative integer vectors $x \in \mathcal{Z}_+^K$ where $\sum_{i=1}^K x_i = n$, and $P = (p_1,...,p_K)$ is an element of the (K-1)-dimensional simplex $\mathcal{S}_K$, $n \geq 1$. The probability mass function (pmf) of this multinomial distribution is:

$$\begin{aligned} f(x_1,...,x_K;n,p_1,...,p_K) &= Pr(X_1 = x_1 \quad and... \quad and X_K = x_K) \\ &= \begin{cases} \frac{n!}{x_1!...x_K!}{p_1}^{x_1}...{p_K}^{x_K}, & when \quad \sum_{i=1}^K x_i = n \\ 0 & otherwise, \end{cases} \end{aligned} \tag{2.14}$$

for non-negative integers $x_1,...,x_K$. It can be expressed using the Gamma function as:

$$f(x_1,...,x_K;n,p_1,...,p_K) = \frac{\Gamma(\sum_i x_i+1)}{\prod_i \Gamma(x_i+1)}\prod_{i=1}^K {p_i}^{x_i}. \tag{2.15}$$

A **Dirichlet distribution** is the conjugate prior for Multinomial distribution in the sense of Bayesian inference, and a generalization of the Beta distribution:

$$f(x_1,...,x_K;\alpha_1,...,\alpha_K) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)}\prod_{i=1}^K x_i^{\alpha_i-1}. \tag{2.16}$$

Using Bayes' Rule which will be explained later, the data $\boldsymbol{X} = X_1,...,X_n$ is generated independently and identically from $Mult(p_1,...,p_K,n)$, its prior information follows $Dir(\alpha)$, and its posterior is:

$$\begin{aligned} P(\theta|\boldsymbol{X}) &\propto P(\boldsymbol{X}) \\ &\propto (\sum_{j=1}^m \theta_j^{N_j})(\sum_{j=1}^m \theta_j^{\alpha_j-1}) \\ &= \sum_{j=1}^m \theta_j^{N_j+\alpha_j-1}, \end{aligned} \tag{2.17}$$

so

$$P(\theta|\boldsymbol{X}) = Dir(N+\alpha). \tag{2.18}$$

Hence, if the prior is a Dirichlet distribution with parameters α, the posterior is a Dirichlet distribution with parameters $N+\alpha$.

Table 2.2: Bivariate Archimedean copula families.

Copula	$C(u,v)$	Parameter θ
Clayton	$(\max\{u^{-\theta}+v^{-\theta}-1;0\})^{-\frac{1}{\theta}}$	$\theta \in (-1,\infty)\backslash\{0\}$
Frank	$-\frac{1}{\theta}\log\left(1+\frac{(\exp\{(-\theta u)\}-1)(\exp\{(-\theta v)\}-1)}{\exp\{(-\theta)-1\}}\right)$	$\theta \in \mathbb{R}\backslash\{0\}$
Gumbel	$\exp\left(-((-\log(u))^{\theta}+(-\log(v))^{\theta})^{\frac{1}{\theta}}\right)$	$\theta \in [1,\infty)$

2.2.1.1 Copula-based Joint Probability Modeling

In most cases, a single random variable is not sufficient to describe a problem, and a joint probability distribution is needed to solve complicated problems. Based on Sklar's theorem, any multivariate joint distribution can be written in terms of uni-variate marginal distribution functions and a copula function, which describes the dependence structure between the variables Nelsen [2006]. Hence, copulas are popular in high-dimensional statistical applications. When two random variables are considered, the bivariate Archimedean family copulas can be used to fit a joint PDF model (cf. Table 2.2). A bivariate copula function is a joint cumulative distribution of two uniform random variables X_1 and X_2, defined as:

$$C(x_1,x_2) = P(X_1 < x_1, X_2 < x_2) \tag{2.19}$$

where $X_i \sim U(0,1)$ for $i = 1,2$.

According to Sklar's theorem Nelsen [2006], the joint cumulative distribution of two random variables X_1 and X_2 is expressed by the copula function of $F_1(x_1)$, $F_2(x_2)$. Therefore, the corresponding joint probability density can be given by the derivative of the following copula function:

$$\begin{aligned} f(x_1,x_2) &= \frac{\partial^2 C(x_1,x_2)}{\partial x_1 \partial x_2} \\ &= c(F_1(x_1),F_2(x_2))f_1(x_1)f_2(x_2). \end{aligned} \tag{2.20}$$

2.2.1.2 Gaussian Mixture Models (GMM)

Sometimes, complicated data can be modeled as a mixture model which is a probabilistic model for representing the presence of subpopulations within an overall population. The simplest example is the popular Gaussian Mixture Model, which is a weighted sum of M component Gaussian densities as given by the following equation:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \sum_{cov}), \tag{2.21}$$

where x is a D-dimensional continuous-valued data vector. A GMM is parameterized by the mean vectors $\mu_i, , i = 1,...,M$, covariance matrices $w_i, , i = 1,...,M$ and mixture weights $\sum_{cov}$ from all component densities. Collectively, they are represented by the notation:

$$\lambda = \{\mu_i, w_i, \sum_{cov}\}, i = 1,...,M. \tag{2.22}$$

Each component density is a D-variate Gaussian function with the form:

$$g(x|\mu_i, \sum_{cov}) = \frac{1}{(2(\pi)^{D/2})|\sum_{cov}|^{1/2}} exp\Big\{ -\frac{1}{2}(x-\mu_i)'\sum_{cov}{}^{-1}(x-\mu_i)\Big\}, \tag{2.23}$$

with μ_i as the mean vector, $\sum_{cov}$ as the covariance matrix. The mixture weights satisfy the $\sum_{i=1}^{M} w_i = 1$ constraint.

GMM parameters can be estimated from training data using the iterative Expectation-Maximization (EM) algorithm or the Maximum A Posteriori (MAP) methods from a well-trained prior model Reynolds [2008].

2.2.1.3 Bayes' Rule

Bayesian approaches use Bayes' rule to update beliefs in hypotheses in response to data:

$$P(Hypothesis|Data) = \frac{P(Data|Hypothesis)P(Hypothesis)}{P(Data)} \tag{2.24}$$

$P(Hypothesis|Data)$ is the posterior distribution, $P(Hypothesis)$ is the prior distribution, $P(Data|Hypothesis)$ is the likelihood, and $P(Data)$ is a normalizing constant also called "evidence".

2.2.2 Parameter Estimation Methods

In order to properly model the data, parameter estimation techniques are needed to estimate the parameters of the selected distributions. Classic methods are method of moments, method of maximum likelihood estimation, method of maximum a posterior estimation, method of log-cumulants, expectation maximization and the Bayesian rule. All these will be presented in this section.

2.2.2.1 Method of Moments (MoM)

The method of moments estimation is based on the law of large numbers, which has the property that the average of the results obtained from a large number of trials shall be close to the expected value, and will tend to converge when more trials are performed:

$$\widetilde{M}_n = \frac{1}{n}\sum_{i=1}^{n} M_i \rightarrow \mu_M \quad \text{as} \quad n \rightarrow \infty. \tag{2.25}$$

MoM has the advantages of being easy to compute, to always work, and the estimate is consistent; its disadvantages are that it is usually not the best estimator which achieves the minimum Mean Squared Error (MSE) and sometimes obtains meaningless estimates.

2.2.2.2 Method of Maximum Likelihood Estimation (MLE)

The idea behind the method of maximum likelihood is that it is preferable to obtain parameter estimates which would most likely produce the data we actually observe.

The definition of likelihood is: Let $f(x_1, ..., x_n; \theta)$, $\theta \in \mathbb{R}^k$, be the joint probability density function of n random variables $X_1, ..., X_n$ with sample values $x_1, ..., x_n$. The likelihood function of the samples is given by:

$$L(\theta; x_1, ..., x_n) = f(x_1, ..., x_n; \theta), \tag{2.26}$$

where L is a function of θ for fixed sample values.

The maximum likelihood estimators (MLEs) are those values of the parameters that maximize the likelihood function with respect to the parameter θ. That is,

$$L(\tilde{\theta}; x_1, ..., x_n) = \max_{\theta \in \Theta} L(\theta; x_1, ..., x_n), \tag{2.27}$$

where Θ is the set of possible values of the parameter θ.

MLE has the advantages that when the sample size n is large ($n > 30$), the MLE is unbiased, consistent, normally distributed, and efficient in the sense that it produces the minimum MSE compared with other methods; Moreover, it is more useful in statistical inference. At the same time, it also has the disadvantages that it can be highly biased for small samples; sometimes there is no closed-form solution; MLE is sensitive to initial values, which might not result in a global optimum.

2.2.2.3 Method of Maximum A Posterior Estimation (MAP)

The method of maximum a posterior estimation estimates θ as the mode of the posterior distribution of this random variable:

$$\tilde{\theta}_{MAP}(x) = \arg\max_{\theta} f(\theta|x) \quad = \arg\max_{\theta} \frac{f(x|\theta g(\theta))}{\int_{\theta} f(x|\theta) g(\theta) d\theta} = \arg\max_{\theta} f(x|\theta) g(\theta). \tag{2.28}$$

Given the MAP formulation, three key points need to be addressed: the choice of the prior distribution family, the specification of the parameters for the prior densities, and the evaluation of the maximum a posteriori. Thus, an appropriate choice of the prior distribution can greatly simplify the MAP estimation process Gauvain and Lee [1994].

Usually, a MAP estimation can be computed in the following four ways:
Analytically, the estimator can be given in closed form because conjugate priors are used; Via numerical optimization, such as a conjugate gradient method or Newton's method. It usually requires first or second derivatives that need to be evaluated analytically or numerically; Via the application of an expectation-maximization algorithm. This does not require derivatives; Or via the Monte Carlo method using simulated annealing.

2.2.2.4 Method of Log-Cumulants (MoLC)

Proposed by Nicolas [2002], and applied by Moser et al. [2006], MoLC has been introduced to SAR data analysis for image distribution estimation. A Mellin transform instead of a Fourier transform is used to analyze random variables which are defined over $\mathbb{R}^+$. For a function f defined over $\mathcal{R}^+$ only, the integral MT[f]

$$MT[f](s) = \int_0^{+\infty} u^(s-1) f(u) du \tag{2.29}$$

is called a Mellin transformation, where s is a complex number whose norm is equal to 1.

There are known relationships between common statistics and the Fourier transform: the first characteristic function of a function f is the Fourier transform of its density function. The nth moment is the nth derivative of the characteristic function; the nth cumulant is the nth derivative of the logarithm of the characteristic function.

Main Second-Kind Statistics: By mimicking common statistics based on a Fourier transformation, with a Mellin transformation, the main functions are summarized by Tison et al. [2004]; here X is a random variable:

- 1st second-kind characteristic function:

$$\phi_X(s) = MT(p_X) = \int_0^{+\infty} x^{s-1} p_X(x) dx. \tag{2.30}$$

- 2nd second-kind characteristic function:

$$\psi_X(s) = log(\phi_X(s)). \tag{2.31}$$

- rth-order second-kind characteristic moment:

$$m_r(s) = \frac{d^r \phi_X(s)}{ds^r}\bigg|_{s=1} = \int_0^{\infty} (logx)^r p_X(x) dx. \tag{2.32}$$

- rth-order second-kind characteristic cumulant which is also called "log-cumulant":

$$k_r(s) = \frac{d^r \psi_X(s)}{ds^r}\bigg|_{s=1}. \tag{2.33}$$

- For regular moments, we have:

$$\hat{k_1} = \hat{m_1}. \tag{2.34}$$

$$\hat{k_2} = \hat{m_2} - (\hat{m_1})^2. \tag{2.35}$$

$$\hat{k_3} = \hat{m_3} - 3\hat{m_1}\hat{m_2} + 2(\hat{m_1})^3. \tag{2.36}$$

- Based on Equations 2.34, 2.35 and 2.36, the first three log-cumulant estimators for N samples y_i are defined as follows:

$$\hat{k_1} = \frac{1}{N}\sum_{i=1}^{N}[log(y_i)]. \tag{2.37}$$

$$\hat{k_2} = \frac{1}{N}\sum_{i=1}^{N}[(log(y_i) - \hat{k_1})^2]. \tag{2.38}$$

$$\hat{k_3} = \frac{1}{N}\sum_{i=1}^{N}[(log(y_i) - \hat{k_1})^3]. \tag{2.39}$$

Therefore, like for MoM, parameter estimation is used to solve the above-mentioned equations.

2.2.2.5 Expectation Maximization (EM)

The EM algorithm is an efficient iterative procedure to compute the maximum likelihood or maximum a posteriori estimate in the presence of missing or hidden data.

Here is an example of calculating a maximum likelihood estimate. Consider a generative model with parameter θ. This model generates a set of data D, which consists of two parts: the observed data X, and the latent data Z. Under this model, the complete likelihood of D is given by:

$$p(D|\theta) = p(X, Z|\theta). \tag{2.40}$$

The marginal likelihood of X is given by:

$$p(X|\theta) = \sum_z p(X, z|\theta). \tag{2.41}$$

Here, the variable Z represents the summation over all possible assignments z. If Z is a continuous variable, then the sum operation is changed to an integral over the domain of Z.

1. Initialize $\tilde{\theta}^{(0)}$, which can be set to some random value in Θ, or by an initial guess derived from problem-specific heuristics.

2. For t = 1,2,..., repeat:

a. **E-step**. The missing data are estimated given the observed data, and the current estimates of the model parameters are obtained by using the conditional expectation. Compute the posterior of the distribution of Z given X and $\theta^{(t-1)}$:

$$q^{(t)}(Z) = p(Z|X; \tilde{\theta}^{(t-1)}). \tag{2.42}$$

Here, $q^{(t)}$ is used to denote the posterior distribution of Z obtained at the t-th iteration.

b. **M-step**. Under the assumption that the missing data are known, the likelihood function is maximized. The estimates of the missing data from the E-step are used to replace the actually missing data. Solve the optimal $\tilde{\theta}^{(t)}$ by maximizing the expectation of the complete log-likelihood with respect to $q^{(t)}$:

$$\tilde{\theta}^{(t)} = \arg\max_{\theta \in \Theta} \sum_z q^{(t)}(z) \log p(X, z|\theta). \tag{2.43}$$

By taking advantage of the independence between the variables, the computation of this sum can be greatly simplified.

3. The iteration stops when a convergence criterion is met: for instance, when the difference between $\theta^{(t)}$ and $\theta^{(t-1)}$ is below a given threshold, etc.

2.2.3 Numerical Optimization Methods

We can solve equations analytically like aforementioned, but sometimes no closed-form solution can be obtained, then numerical optimization methods can be considered. In order to better discuss the problem, a simple supervised learning setup is considered. Each example pair $z = (x, y)$ is composed of an arbitrary input x and a scalar output y. A loss function $l(\tilde{y}, y)$ measures the cost of predicting $\tilde{y}$ when the actual answer is y. A family $\mathcal{F}$ of functions $f_w(x)$ is parametrized by a weight vector w. We seek the function $f \in \mathcal{F}$ which minimizes the loss $Q(z, w) = l(f_w(x), y)$ averaged on the examples $z_1, ..., z_n$:

$$E_n(f) = \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i). \tag{2.44}$$

The empirical risk $E_n(f)$ measures the training set performance.

2.2.3.1 Gradient Descent

Gradient descent, also known as steepest descent, is a first-order iterative optimization algorithm. In order to find a local minimum of a function, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point.

Each iteration updates the weights w on the basis of the gradient of $E_n(f_w)$:

$$w_{t+1} = w_t - \gamma \frac{1}{n} \sum_{i=1}^{n} \nabla_w Q(z_i, w_t), \tag{2.45}$$

where γ is an adequately chosen gain. Under sufficient regularity assumptions, when the initial estimate w_0 is close enough to the optimum, and the gain γ is sufficiently small, the algorithm achieves *linear convergence* Bottou [2010].

2.2.3.2 Newton's Method

By replacing the scalar gain γ by a positive definite matrix Γ_t which approaches the inverse of the Hessian of the cost at the optimum, we obtain:

$$w_{t+1} = w_t - \Gamma_t \frac{1}{n} \sum_{i=1}^{n} \nabla_w Q(z_i, W_t). \tag{2.46}$$

This *second-order gradient descent* is a variant of the well-known Newton's method. Under sufficiently optimistic regularity assumptions, when w_0 is sufficiently close to the optimum, this algorithm achieves *quadratic convergence* Bottou [2010].

2.2.3.3 Stochastic Gradient Descent

Instead of computing the gradient of $E_n(f_w)$ exactly, in the stochastic gradient descent algorithm, each iteration estimates this gradient on the basis of a single randomly picked example z_t:

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t). \tag{2.47}$$

The stochastic process $w_t, t = 1, \ldots$ depends on the examples randomly picked at each iteration, and this is a drastic simplification.

Similarly, the *second-order stochastic gradient descent* (2SGD) multiplies the gradients by a positive definite matrix Γ_t to approach the inverse of the Hessian Bottou [2010]:

$$w_{t+1} = w_t - \gamma_t \Gamma_t \nabla_w Q(z_t, w_t). \tag{2.48}$$

2.2.4 Sampling Methods

In machine learning applications, the integration and optimization problems play a fundamental role. Hence, sampling methods are often applied to solve these problems in high-dimensional spaces.

2.2.4.1 Markov Chain Monte Carlo (MCMC)

As a posterior probability is in most of the cases difficult to compute, the **MCMC** approximation generates samples from the posterior distribution by constructing a reversible Markov-chain as an equilibrium distribution to match the target posterior distribution. In this sub-section, the details of how inference is actually performed will not be explained; however, in the sense of practical applications, the **Metropolis-Hastings algorithm**, the most popular MCMC method, will be explained as an example.

1. Initialise $x^{(0)}$.

2. For $i = 0$ to $N - 1$
   - Sample $u \quad U_{[0,1]}$.
   - Sample $x^* \quad q(x^* \| x^{(i)})$.
   - If $u < \mathcal{A}(x^(i), x^*) = min(1, \frac{p(x^*)q(x^{(i)}|x^*)}{p(x^(i))q(x^*|x^(i))})$

$$x^{(i+1)} = x^* \tag{2.49}$$

   else

$$x^{(i+1)} = x^{(i)} \tag{2.50}$$

Algorithm 2.1. Metropolis-Hastings algorithm.

A Metropolis-Hastings step of an invariant distribution $p(x)$ and proposal distribution $q(x^*|x)$ includes sampling a candidate value x^* given the current value x according to $q(x^*|x)$. Then the Markov chain moves towards x^* with acceptance probability $A(x, x^*) = min(1, \frac{p(x^*q(x|x^*))}{p(x)q(x^*|x)})$, otherwise it remains at x Andrieu et al. [2003].

Chapter 3

State of the Art

> To know that we know what we know, and that we do not know what we do not know, that is true knowledge.
>
> Confucius

With the increased availability of Earth Observation (EO) data, due to new satellite missions, their various sensors, and the interoperability of data archives, the remote sensing community is facing today a dramatic increase in both data volume and data content details. Many EO applications which stem from the early days of the geoscience and remote sensing discipline, such as measuring land-use characteristics, monitoring and responding to natural disasters, managing natural resources, etc. are nowadays outdated and have been complemented by more detailed innovative application-oriented techniques. Further, remote sensing image analysis models, methods and algorithms, especially in urban areas, have been largely developed for medium-resolution images. With regard to their data volume as well as their data content, the current abundant high-resolution satellite data are far from being well processed, analyzed, and utilized. They, for example, hyperspectral images taken from space, are stimulating new research ideas, and driving future research trends towards new models and algorithms for object analysis, pixel level segmentation, etc.

To fully explore the value of these data, their essential information has to be extracted, converted and presented in tangible form which can be utilized in conjunction with other data sets, for example, the widely used Geographic Information Systems (GIS). Particularly, in order to browse and index image content information efficiently in large-scale image databases, usually the information is compressed and complemented by textual keywords. A common method is to pre-process, classify and, in the end, annotate the image databases. Despite its high level of correctness, manual annotation requires a considerable amount of human effort, because labeling an image forces us to become aware of the detailed image content which is very time-consuming Barriuso and Torralba [2012]. As a result, some researchers process data with the help of further professional GIS software, some process data on pixel level, and some process them on polygon level Benz et al. [2004]. However, these users process data only for certain specialized applications, and their functionalities need to be enhanced.

In this dissertation, we focus on developing various kinds of fully automatic or semi-automatic techniques which can efficiently and correctly interpret an image database, without reliance on commercial software. By the nature of image annotation, there are three major intrinsic problems: (a) **how to describe an image mathematically**, (b) **how to assess the similarity of images Datta et al. [2008]**, and (c) **How to extract information**

Table 3.1: Ranges of spatial resolution with different satellites or instruments and corresponding levels of expected object recognition.

Type of satellite or instrument scale	Approximate range of spatial resolution (meters)	General level of object discrimination
Low-resolution satellite images (Landsat 1,2,5)	30 m or worse	Broad land-cover patterns (regional to global mapping) DigitalGlobe [2015]
Medium-resolution satellite images (Spot 1,2,3,5, Rapid Eye, ESA's Sentinel-2)	5 m to 30 m	Separation of between major land use classes (e.g., urban, agricultural, forest, water, barren, vegetation) DigitalGlobe [2015]
High-resolution satellite images (e.g., SPOT 5,6, TerraSAR-X)	1 m to 5 m	Recognition of urban elements (buildings, houses, sport grounds), natural forest stands and orchards DigitalGlobe [2015]
Very high resolution satellite images (e.g., WorldView-1,2,3, Quickbird)	1 m or better	More finer objects are observable (cars, smaller trees, and large animals in grassland)
Airborne multispectral scanners	$>$ 0.3 m	Identification of very small objects (windows, doors and large individual trees)
Drone-mounted digital frame camera Mostegel et al. [2016]	$>$ 0.04 m	Identification of humans, faces, phones and books

within an image which corresponds to the following chapters.

3.1 Earth Observation Meets Computer Vision

In the sense of different resolutions, as shown by Table 3.1, the spatial resolution trends fall into 6 classes, while very detailed urban elements (buildings, houses, etc.) can be recognized from high-resolution satellite imagery.

Hence, in this section, we briefly review the development of Earth observation, especially when it meets computer vision. We are under the illusion that seeing is effortless, but frequently our visual system is cheated by false perceptions which make us believe we understand something but in fact we don't. The task of labeling an image forces us to become aware of the difficulties underlying scene understanding. Suddenly, the act of seeing is not effortless anymore Barriuso and Torralba [2012]. Hence, appropriate techniques are needed to overcome the difficulties, and the consideration of the booming computer vision area is probably a good choice.

As photogrammetry has developed in parallel to remote sensing, researchers also have noticed the growing relations of photogrammetry with computer vision Foerstner [2009]. In the early days of high-resolution imaging, as opposed to pixel-level based analysis, computer

vision based image analysis has been performed mainly through the recognition of objects, for example, coastlines, roads, rivers, etc., as objects allow for a meaningful interpretation of an image which is close to human perception. In this case, textural and line features were extracted as image primitives, probabilistic graphs were used as model-based processing, and shape information was used to analyze a multi-sensor dataset Gautama et al. [2000].

Nowadays, computer vision experts are still making efforts for understanding parts of larger images that we probably neglect at a first glance Barriuso and Torralba [2012]. They have collected many typical image datasets and their understanding is supported by image annotation. Most of the available semantically annotated image collections, which contain a large variety of retrievable objects and classes, have been built by individual groups with the intention to solve specific problems. Typical examples are LabelMe Russell et al. [2008] and Scene UNderstanding (SUN) Xiao et al. [2010].

Compared with computer vision applications that mainly focus on ordinary optical images which cover specific objects within a small field of view, remote sensing images are often used for EO purposes covering large areas which contain rich and detailed information. However, remote sensing image analysis applications haven't yet reached the same level of automation as computer vision applications, and is way more complicated. In order to fast browse and index image content in a large-scale remote sensing image dataset, it is critical to develop techniques which are able to semantically annotate a given dataset efficiently and with high quality. Thus, one of the urgent but not yet well-solved tasks in remote sensing is **the image annotation in large-scale remote sensing datasets**. More specifically, the purpose of semantic image annotation in remote sensing is to reach a better understanding not only of the natural environment that surrounds us, but also of the built-up environment that we live in.

3.2 Hierarchical Representation

Here we deal with the first question: **How to represent the image information?** In remote sensing, a large complex image scene is usually highly informative with multiple levels of concepts, it is then preferable to logically represent the image content as a hierarchical structure with abstract and detailed levels. For example, urban areas can be categorized into sub-classes such as densely built-up areas or sparsely built-up areas. In addition, a hierarchical structure provides us a visual intuition to describe the connections between different layers.

As images are usually highly informative with multiple levels of concepts, image semantics is represented as a hierarchical structure with different levels. Some researchers try to provide descriptive information, e.g., image attributes or tags in addition to image classes or categories. Gao et al. [2014] propose a multi-layer group based tag propagation method which combines the class labels and subgroups of instances with similar tag distributions to annotate test images. A hierarchical tree-structured semantic unit provides not only the class and subclass an image belongs to, but also the attributes an image has Han et al. [2014]. In Yuan et al. [2015], hierarchical features are learnt based on a layer-wise tag-embedded deep learning model to correspond to hierarchical semantic structures in the tag hierarchy; some researchers adjust topic model concepts which have been successfully applied in the natural language processing to image processing domains, e.g., Latent Dirichlet Allocation (LDA); here the image information is described as documents, topics, and words. In Hoo and Chan [2015], a zero-shot object recognition system which integrates a topic model and a hierarchical class concept is proposed, and comparable performance with

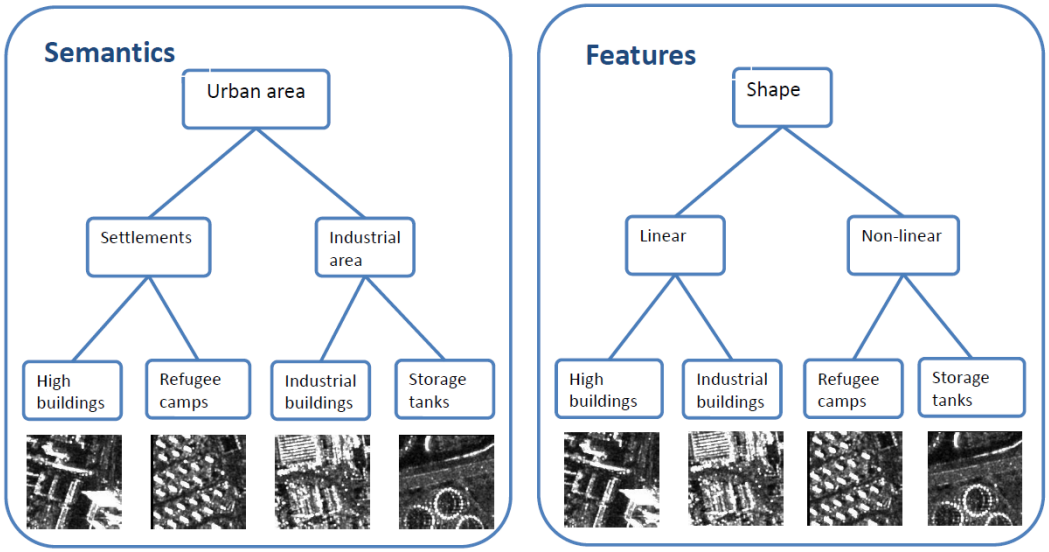


Figure 3.1: Hierarchical representations from semantic and feature perspectives.

state-of-the-art algorithms is achieved. A novel author-genre-topic hierarchical generative model is proposed in Luo et al. [2014]. In addition to a cascaded category-aware visual search model, weak category clues are utilized to achieve better retrieval accuracy, efficiency, and memory consumption Zhang et al. [2014]. WordNet[1] is a typical example of hierarchical representation. It is an English lexical database where nouns, verbs, adjectives, and adverbs are grouped into distinct concepts that are interlinked via conceptual-semantic and lexical relations. There is also psycholinguistic theory behind, which attributes human memory with hierarchical semantics, as complex scenes are difficult to be fully described by a single label for each image patch. Some researchers propose a hierarchical Multi-Instance Multi-Label semantic learning (MIML) method which is a variation of the traditional semisupervised framework for high-resolution remote sensing image annotation via a Gaussian process Chen et al. [2013].

We can build a hierarchical representation from different perspectives: feature hierarchy and semantic hierarchy, Fig. 3.1 shows that, when we consider general-level semantic labels of "settlements" and "industrial area", "high buildings" and "refugee camps" to belong to "settlements", "industrial buildings" and "storage tanks" belong to "industrial area"; when we consider the shape features of "linear" and "non-linear", now "high buildings" and "industrial buildings", "refugee camps" and "storage tanks" should be grouped together. However, in this dissertation, we prefer to construct a semantic hierarchy rather than a feature hierarchy.

3.2.1 Feature Hierarchy

Features can be extracted hierarchically. From a biological perspective, a hierarchical process seems to happen during the fast visual recognition in the mammalian cortex, and is also consistent with the inherent characteristics of human cognition. Hence, a number of weakly inspired computer vision models and unsupervised learning algorithms are described and presented in LeCun [2012], the effectiveness of these algorithms for learning various feature hierarchies are demonstrated with practical tasks such as scene parsing, pedestrian

[1] http://wordnet.princeton.edu

detection, and object classification.

The hierarchical features can represent objects in a very exquisite way. In Epshtein and Uliman [2005], an automated informative feature hierarchy extraction method is described for object classification, and according to the authors, these hierarchical features are more informative and better for classification compared with similar non-hierarchical features. Moreover, a feature hierarchy organizes image representations into multiple levels which also correspond to different semantic levels. Hence, some authors propose a novel Layer-wise Tag-embedded Deep Learning (LTDL) model to learn hierarchical features that relate to hierarchical semantic structures in the tag hierarchy Yuan et al. [2015].

3.2.2 Semantic Hierarchy

In most cases, semantic labels are formed hierarchically. In the survey of semantic hierarchies for image annotation Tousch et al. [2012], the use of an unstructured vocabulary and structured vocabulary are analyzed, and it is argued that the use of structured vocabularies is critical to the success of image annotation.

Regarding the application of detecting and recognizing object parts, e.g., face recognition, a semantic hierarchical representation is proposed in Epshtein and Ullman [2007]. In this method, a minimal feature hierarchy is constructed at the beginning, then semantically equivalent representatives are added to each node, and the entire hierarchy is used as a context to determine the identity and locations of added features. Particularly, object parts are detected and learned for all levels in the hierarchy. Similarly, a lexical semantic network is proposed in Marszalek and Schmid [2007] to extend the current object recognition techniques. The semantics of image labels are used to integrate prior knowledge about inter-class relationships. Thus, a semantic hierarchy of discriminative classifiers is built and trained. The results demonstrate that high-level categories can be classified due to the extension with semantic inference tools, e.g., animals or car windows. Moreover, the topic model can be used for the application of object recognition Hoo and Chan [2015]; in this paper, a probabilistic Latent Semantic Analysis (pLSA) topic model is used. The idea is that an object is represented via a hierarchical concept, which includes feature primitives, bag of words features, and semantic topics in the image.

Most of the available image classification works are not able to deal with large-scale image search; hence, a cascaded category-aware visual search method is proposed in Zhang et al. [2014], where noisy local features are discarded, and visual and category image clues are extracted and recorded in a hierarchical index structure; the algorithm shows good performance although the introduced training information is weak. Armed with a sparse coding technique, a multi-layer group-based tag propagation method which combines class labels and subgroups of instances is proposed in Gao et al. [2014], where the Reproducing Kernel Hilbert Space (RKHS) is used for non-linearly separable features; in this case, a kNN strategy is integrated which greatly improves the computational efficiency.

In summary, as the world is so complex and contains a lot of different objects, a semantic hierarchy is viewed as a practical and useful tool to model and represent it. Thus, the second goal of this research is **to represent semantic labels hierarchically.**

3.3 Description of Images

When we want to represent images hierarchically, at first, we need to solve a problem: **How to describe image contents?** In both the remote sensing and computer vision communities,

this is usually done by extracting and encoding image features. Fig. 3.2 illustrates different

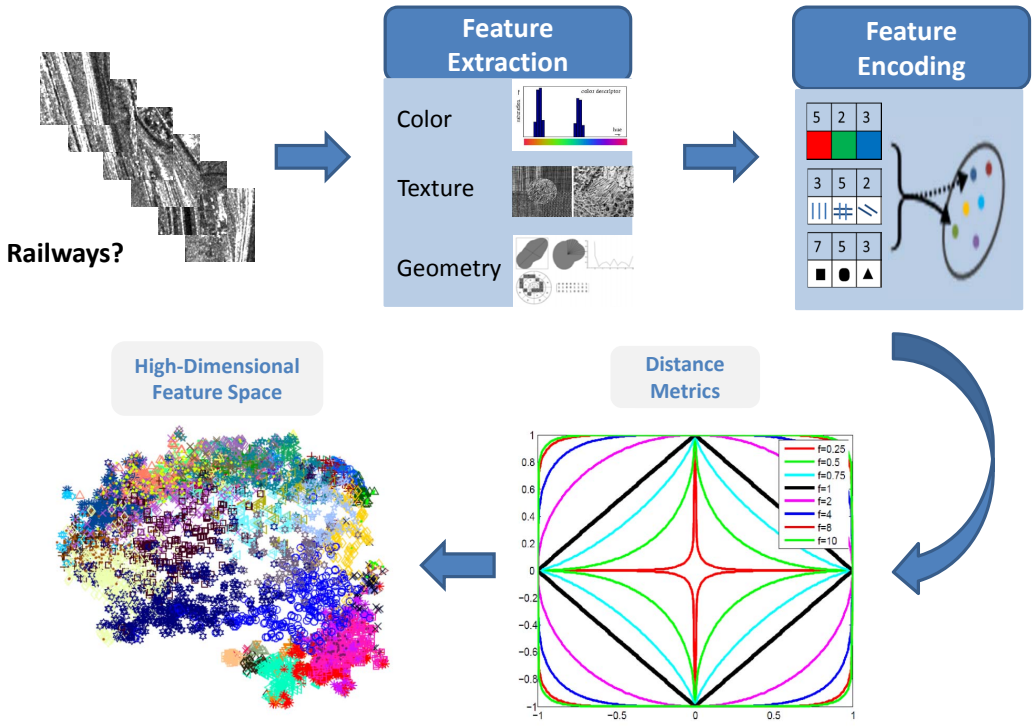


Figure 3.2: Description and analysis of images. The procedure consists of feature extraction, feature encoding, and different distance metric analyses in high-dimensional feature space.

complexities of feature descriptors, the common image description procedure consists of two levels: the first step deals with feature extraction which is the extraction of image texture, shape and color information as a primitive feature descriptor; the next step is feature encoding in which the primitive descriptors are aggregated as a feature vector, for example, as a bag-of-words feature.

3.3.1 Feature Extraction

Image feature extraction can be viewed from many perspectives, traditionally, features are extracted based on human expert knowledge. Take finger print classification for example: the image gradient direction, the local dominant orientation, and minutia points are extracted and described as an image feature vector. Regarding remote sensing images, and the special properties of satellite images, for example, the speckle effect which is inherent in SAR images, should be analyzed and taken into account when necessary.

Without domain-specific knowledge, features can be extracted based on our general knowledge. One important image property is its color or spectral information, which can be identified by different color descriptors. With respect to image structure, we

concentrate on shape features: edges, lines, bridges, image region contours, corners, and blobs which represent local structures. For the global image properties, we usually extract texture features: the statistical properties of an image such as its gray-level histogram, mean, variance, Fourier transform, and various moments. For example, Newsam et al. [2003] use Gabor filter based homogeneous texture descriptors to annotate remotely sensed datasets. Furthermore, from a machine learning perspective, feature extraction can be made by analyzing the variables after a statistical processing procedure: principal component analysis, discriminant analysis, etc. Jiang [2009].

These primitive features, i.e., color, textual and shape features, have been integrated or evaluated in a number of different research works. Image retrieval from large and heterogeneous image databases is evaluated in Howe and Huttenlocher [2000] by exploiting a diverse and expandable set of image properties, for instance, color, texture, and location information. Various primitive features, for example, gray-level co-occurrence matrices, Gabor filters, quadrature mirror filters, and nonlinear short-time Fourier transforms have also been adapted for SAR images and evaluated, see Dumitru and Datcu [2013]. Furthermore, for general land cover classification, for example, urban, natural land and water classification, Aytekin et al. [2013] propose a new SAR image feature vector by combining the intensity information of pixels with spatial information and structural relationships. This work also shows the importance of taking account of both the intensity information of pixels as well as their spatial-structural relationships when generating an image feature vector.

3.3.1.1 Multi-Spectral Information

Spectral information is one of the most commonly used features of remote sensing images; it is especially useful for multi-spectral or hyper-spectral images. In RGB color space, spectral information is considered as color information. The traditional color descriptors are extracted to represent image color distributions and are often expressed in the form of a color histogram. The simplest color descriptor is obtained by calculating a color histogram in the original RGB space Swain and Ballard [1991]. The corresponding discrete distributions can be summarized to their first- and second-order statistical moments Stricker and Orengo [1995]. The performance of the color descriptors which include a histogram descriptor coded via a Haar transform, a color structure histogram, a dominant color descriptor, and a color layout descriptor that are defined in the MPEG-7 standard, are documented in Manjunath et al. [2001]. However, the drawback of distribution-based color histogram descriptors is their lack of spatial relationships. Hence, in order to incorporate spatial color information among image pixels, color coherence vectors Pass et al. [1996], color correlograms Huang et al. [1997], and weighted color histograms are proposed and analyzed in Vertan and Boujemaa [2000]. In principle, the authors try to weight each pixel distribution into its corresponding color distribution.

Another direction goes into color space transformations. There are different color spaces; the Lab color space is considered to better coincide with human perception than the RGB space; the YUV color space is a linear transform of RGB which separates intensity from color information. It is primarily used in video applications; one luminance (Y) and two chrominance (UV) components are defined in this model Tokarczyk et al. [2015]; Salembier and Sikora [2002] describe a set of color descriptors using the YUV color space.

When we look at object and scene recognition applications, the invariance properties and the distinctiveness of color descriptors are studied in a structured way in Sande et al. [2010]. Based on their theoretical and experimental results, the invariance to light intensity

changes and light color changes affects category recognition. Furthermore, the usefulness of invariant properties turns out to be category-specific.

As for color object recognition, there is a discussion in Gevers and Smeulders [1999], which concludes that for multicolored man-made objects, the highest object recognition accuracy is achieved by $l1$, $l2$, $l3$ and a hue color model.

Color information extraction from multi-spectral satellite images is different from the extraction from common color spaces, which requires specific techniques. Multi-spectral image analysis is usually made by combining different spectral bands. In the case of SAR or optical satellite images, the color descriptor is reduced to a single channel which represents the intensity value. The commonly used color descriptors described above can be adjusted to gray level images.

3.3.1.2 Textural Information

Image texture information is a set of metrics designed to describe the spatial arrangement of intensities in an image or selected regions of an image. Texture features are characterized to describe granular and repetitive patterns inside images; hence, they have direct relations to image semantics: categories such as forest, grassland and urban areas can be well distinguished using very simple texture attributes.

Haralick et al. Haralick et al. [1973] propose to use quantities based on gray level co-occurrence matrices to characterize image structures. The co-occurrence matrices represent the distribution of co-occurring pixel values at a certain offset, and correspond to second-order statistics. Based on these matrices, statistical measures such as contrast, uniformity, mean, variance, inertia moments, etc. and up to fifth-order statistical moments are computed on theses matrices.

The performance of the texture descriptors defined in the MPEG-7 standard which include one that characterizes homogeneous texture regions, another that represents the local edge distribution, and a compact descriptor that facilitates texture browsing, are documented in Manjunath et al. [2001].

Besides, geometric and photometric transformation invariance are desired characteristics of texture features; hence, they are very helpful to the segmentation of natural scenes or object recognition in remote sensing images. In Lazebnik et al. [2003], an affine-invariant texture descriptor is extracted from a sparse set of regions using a Laplacian blob detector. In Mikolajczyk and Schmid [2004] a Harris detector is used to locate highly textured points and the derivatives of the Laplacian allows the computation of the scale and affine shape of the corresponding local structures. This kind of affine- and scale-invariant texture information is very useful to represent local structures and regions.

Later, many approaches which rely on transforms to capture textual information have been proposed. Most of these transforms are made in the spatial-frequency domain as the frequency and the orientation of repetitive patterns which describe how textual information can be captured. As a classic image feature technique which originated from signal processing, Gabor filter-based features proposed by Manjunath and Ma [1996] for two-dimensional signals are used as a stable tool for image feature extraction. These Gabor texture features are based on multichannel filtering which covers a limited range of frequencies and orientations, thus emulating some characteristics of the human visual system. The filters are defined by various radial frequencies, standard deviations, and orientations. In Kamarainen et al. [2006], the authors study the most significant results of Gabor filtering, its invariance properties, and restrictions on the use of Gabor filters in feature extraction.

Similar ideas are also applied to wavelet transforms in which the wavelet coefficients are used as the final texture descriptor Randen and Husoy [1994], Do and Vetterli [2002], and Ruiz et al. [2004]. The wavelet transform based texture features decompose a signal using a series of elementary functions which are created by scalings and translations of a base function as wavelets via high-pass filters and scalings via low-pass filters.

Other popular texture descriptors are Tamura's six texture features selected via psychological experiments: coarseness, contrast, directionality, linelikeness, regularity, and roughness Tamura et al. [1978]. They perform well on general databases. It is also noticed that they can outperform traditional texture features, e.g., first- and second-order statistics on Gabor filter bank outputs Deselaers et al. [2008]. Moreover, different texture features, and their use in content-based image retrieval applications are reviewed in Howarth and Rueger [2004]. Recent research for SAR images has shown that a combination of texture features comprising mean, variance, wavelet components, semivariogram, lacunarity, and weighted-rank fill ratio yields a better classification accuracy than single textural measures Chamundeeswari et al. [2009].

3.3.1.3 Geometric Information

Shape information is another powerful attribute in describing image content, and is strongly linked to image semantics, because humans can easily recognize objects simply from their shapes, which demonstrates their special distinguishability compared to color and texture features. The purpose of shape features is to encode simple geometric forms such as straight lines in different directions. Basically, the shape features can be categorized into the following classes: polygonal approximations which include merging methods and splitting methods; spatial interrelation features which include convex hulls, chain codes, smooth curve decompositions, shape matrices, and shape context; moments which include boundary moments and region moments comprising invariant moments, algebraic moment invariants, homocentric polar-radius moments, and orthogonal Fourier-Mellin moments; scale-space methods which include curvature scale spaces, and intersection point maps; and shape transform domains which include Fourier descriptors, wavelet transforms, angular radial transforms, and shapelet descriptors.

In the computer vision field, Hoiem et al. [2005] propose a multiple hypothesis framework for robustly estimating scene structures from a single image and obtaining confidence levels for each geometric label; the coarse geometric properties of a scene can be estimated by learning appearance-based models of geometric classes, even in cluttered natural scenes. The method is useful in two domains: object detection and automatic single-view reconstruction. Moreover, the implementation procedures of different shape representation techniques have been examined and their advantages and disadvantages are discussed in Zhang and Lu [2004].

Furthermore, in computer vision, in contrast to traditional geometrical features, researchers have proposed several shape features which consider spatial information between pixels and their neighborhoods. Lowe [2004] presents a method to extract image distinctive features which are invariant to image scale and rotation by transforming image data into scale-invariant coordinates relative to local features. This method was originally proposed to be used to perform reliable matching between different views of an object or scene. Later, Bay et al. [2008] present a another scale- and rotation-invariant interest point detector and descriptor, called SURF (Speeded-up robust features) which approximates or even outperforms previous schemes regarding repeatability, distinctiveness and robustness; however, it can be computed and compared much faster. The method relies on integral

images for image convolutions by using a Hessian matrix-based measure for the detector, and a distribution-based descriptor. For robust visual object recognition, Histograms of Oriented Gradient (HOG) descriptors are proposed in Dalal and Triggs [2005]. The method is based on the idea that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions. As shown in Cai et al. [2013], Wang and Wang [2015], Bouchiha and Besbes [2013], and Torrione et al. [2014], these shape features have also been applied and adapted in the remote sensing area and they perform well.

3.3.2 Feature Encoding

Many state-of-the-art scene classification techniques re-process low-level image descriptors into richer feature encoding representations of intermediate complexity Boureau et al. [2010]. Feature encoding gives us a more compact feature representation based on primary feature descriptors. According to Chatfield et al. [2011], this includes the following methods: histogram encoding which is actually vector quantization, kernel codebook encoding, locality-constrained linear coding, Fisher encoding, and super-vector encoding.

In particular, as first described by Li and Perona [2005] and Li et al. [2007], bag-of-visual-words features yield more satisfactory and cogent experimental classification results than other state-of-the-art features Feng et al. [2011], Yang et al. [2007], and Yang and Newsam [2010]. When constructing a bag-of-visual-words representation, Cui et al. [2015] discovered that the direct use of the pixel values from a local window as low level features and the introduction of a random dictionary to divide the feature space leads to very competitive results, and can achieve rather good performance for both optical and SAR satellite images.

Regarding human action recognition, Fisher vector feature encoding is used in combination with efficient linear kernels in Kantorov and Laptev [2014], and shows better performance compared to histogram encoding.

In Boureau et al. [2010], several types of coding modules, such as hard and soft vector quantization, and sparse coding, are comprehensively cross-evaluated. A more recent and thorough discussion concerning different coding methods as well as the connections between them, especially their motivations and mathematical representations, and how they have evolved, are surveyed in Huang et al. [2014]. According to these authors, when considering the whole feature space, voting-based coding and Fisher coding follow different ways of describing the probability density distribution of features. Considering local coding with careful feature representation, reconstruction-based coding achieves a more precise description of each feature than voting-based coding. For example, Locality-constrained Linear Coding (LLC) performs better than other classic methods such as sparse coding, and also runs faster than most reconstruction-based coding methods. Compared with reconstruction-based coding, saliency-based coding has two advantages: it is directly derived from the definition of saliency which avoids the underdetermined problem in the least-squares-based reconstruction procedure, and it performs much faster due to its easy implementation without iterative optimization.

Due to lots of parameters, resulting from the feature encoding process, the final feature vectors usually form a high-dimensional feature space, which thus brings out the famous "curse of dimensionality" problem.

3.3.3 The Curse of Dimensionality

Richard Bellman apparently coined the term "the curse of dimensionality" in a book on control theory Bellman [1961], by saying "In view of all that we have said in the foregoing sections, the many obstacles we appear to have surmounted, what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from the earliest days."

Generally, the problem with high dimensionality is that a certain number of data points becomes increasingly "sparse" as the dimensionality increases. Here is an example of one of the special topology characteristics in high dimension given by Bishop [2006]: Considering a sphere of radius α together with the concentric hypercube of side 2α, so the sphere touches the hypercube at the centers of each of its sides, then by calculating the ratio of the volume of the sphere to the volume of the cube, the results show that in a high-dimensional space, most of the volume of a cube is concentrated in the large number of corners, in other words, "spikes".

In remote sensing, usually people extract and combine various kinds of feature descriptors to describe an original image; thus, a feature vector typically lies in a very high-dimensional feature space, which brings out the well-known "curse of dimensionality" problem. For example, in the case of cluster analysis, the original data is divided into clusters for the purposes of summarizing or improving understanding. The effect of increasing dimensionality of distances or similarity is a new challenge which needs to be taken up to tackle the curse of dimensionality problem Steinbach et al. [2003]. Other researchers claim that the commonly used Euclidean norm and Gaussian kernels are inappropriate in high-dimensional spaces. Hence, alternative distance measures and kernels together with geometrical methods are developed by Verleysen and Francois [2005] to reduce the dimension of the space. Efficient feature coding can shrink a high-dimensional feature space to a relatively modest dimension. Regarding the problem of large-scale image search, a joint dimensionality reduction and indexing approach is proposed in Hervé et al. [2012] who claim that the Fisher kernel gives better feature coding than the reference bag-of-visual-words approach for any given vector dimension. The evaluation also concludes that the image representation can be reduced to a few dozen bytes while preserving high accuracy.

From a mathematical point of view, the curse of dimensionality term usually refers to the apparent intractability of searching in a high-dimensional space while approximating and integrating a high-dimensional function Donoho [2000]. As discrete function integration is used very frequently in machine learning algorithms, e.g., in Bayesian graphical models, some experts propose algorithms to tame the problem by applying randomized algorithms which decompose the problem into a small number of discrete combinatorial optimization problems subject to constraints used as a hash function Ermon et al. [2013]. For better searching and indexing in high-dimensional data spaces, the pyramid-technique is proposed by Berchtold et al. [1998] towards breaking the curse of dimensionality. It is highly adapted to range query processing using a so-called maximum metric. The original data is mapped into a one-dimensional space which is managed via a B+ tree. Moreover, Donoho [2000] mentions it as both a curse and a blessing of dimensionality which indicates the high-dimensionality is actually a double-edged sword.

In contrast to the traditional negative impression of the high-dimensionality curse, there is also positive aspect of high dimensionality. This is the concentration of a measured phenomenon, for example, in the Banach space, in which random fluctuations are well controlled in high dimensions. Thus, some problems will be easier to solve than in lower dimensions. In our case, if the blessings of dimensionality are properly utilized, these might

bring us unexpected and surprising performance gains.

Hence, one goal of this dissertation is **to study features in a high-dimensional feature space,** especially what is a good distance metric to describe the pair-wise relationships between feature points. Therefore, the definition of distance metrics plays an important role in exploring features in a given feature space which usually has more than three dimensions Aggarwal et al. [2001]). Exploratory data analysis which aims to identify the main data characteristics will be explained in following chapters, in order to verify statistical assumptions, to select appropriate models, and to determine the hidden relationships among the variables.

Like atoms behave differently in tiny molecules, for example, the wave-particle duality property, classic geometrical characteristics also change in a high-dimensional space, for example, the volume of a cube case where most of the mass are in the corners. The classic Euclidean distance does not suit for a high-dimensional space anymore, and other distance metrics are needed to measure the distance between two elements. In this dissertation, we have a closer look at this phenomenon.

3.3.4 Distance Metrics

From a mathematical perspective, a set of points X and a metric d defines a metric space (X, d), and a metric function is also known as the distance between two points Burago et al. [2001]. These distance metrics reduce multi-dimensional distances to simple scalars which can be used for later processing. Regarding digital images, the similarity between image feature vectors is measured by some practical distances: L_p metrics which are usually used for image compression to select a lossy compression scheme, weighted editing metrics, or cosine distance, Mahalanobis distance which is histogram quadratic distance, Kullback-Leibler distance, Kolmogorov-Smirnov distance, Hamming distance which calculates pixel misclassification error rate, and Hausdorff distance which measures the distance between two point-sets Deza and Deza [2009].

As a similarity measurement, the Hausdorff distance is used to compare different images Huttenlocher et al. [1993]. The method is tolerant of small position errors and it extends naturally to the problem of comparing a portion of a model against an image. In Yu et al. [2006], a distance metric study is made, and the authors propose a novel boosted distance metric which not only finds the best distance metric that fits the distribution of the underlying elements but also selects the most important feature elements with respect to similarity. Concerning the application of image retrieval, distances are calculated between feature vectors of a query and reference data in order to query and index new images Deza and Deza [2009].

Moreover, for the unsupervised clustering and supervised classification methods, due to their computational efficiency and ease of use, Euclidean distance, Manhattan and Minkowski distance metrics have been embedded in the k-means algorithm for comparison and discussion Singh et al. [2013]. As proposed by Von Luxburg and Bousquet [2004], a Lipschitz classifier can be used for metric spaces and a corresponding notion of margin is defined such that the classifier separates the training points with a large margin.

In a broader sense, distance metrics have also been used for the indexing of a digital library of popular music Francu and Nevill-Manning [2000], for pollen records Gavin et al. [2003], for duplicate detection in databases Bilenko and Mooney [2002], for the comparison of strings for name-matching Cohen et al. [2003], and social networks Tang et al. [2009].

3.3.4.1 Fractional and Minkowski Distances

Especially, in a high-dimensional space, the typical characteristics are the undesired effects of distance concentration and the emergence of hub and anti-hub objects. A hub object has a small distance to an exceptionally large number of data points; an anti-hub object lies far from all other data points. An unsupervised approach is proposed in Schnitzer and Flexer [2014] for an empirical examination of concentration and hubness, and choosing an L_p norm by minimizing hubs and miximizing nearest neighbor classification. Particularly in high-dimensional continuous landscape analysis, distance metrics together with sampling techniques are discussed in Morgan and Gallagher [2014], and their limitations have also been studied and improved.

For a specific feature descriptor, different distance metrics can lead to large discrepancies in clustering performances. For example, in Aggarwal et al. [2001], the fractional distance metric is claimed to outperform classic Euclidean and Manhattan distance metrics with clustering algorithms. Previous work has also resulted in the use of fractional L_p distance metrics instead of the universal Euclidean distance metric Aggarwal et al. [2001]. Howarth and Rueger [2005] provide more meaningful results both from the theoretical and empirical perspectives. However, researches contained in Francois et al. [2007] conclude that fractional metrics are always more concentrated than the Euclidean metric, contrary to what is generally assumed. This indicates that it is worthwhile and intriguing to study fractional distances, and to analyze the clustering or classification performances of different distance metrics, e.g., fractional, L1, L2, and Minkowski distance metrics, etc.

The Minkowski distance is a metric in a normalized vector space which can be considered as a generalization of both the Euclidean distance which is known as the L2 distance, and the Manhattan distance which is known as the L1 distance. In this dissertation, the Minkowski distance is used to represent distance metrics.

The Minkowski distance was originated by Kruskal [1964]. It is a natural extension of fractional distances, as they share a similar distance metric form, except that the norms are different. For a more detailed explanation of Minkowski distances, see Van de Geer [1995]. There the author discusses how to use the Minkowski model to solve problems. In Amorim and Mirkin [2012], the use of Minkowski metric based k-means is proposed to tackle irrelevant features; the method appears to be competitive compared to other distance metrics based on k-means algorithms, e.g., the Euclidean metric based approach.

3.3.4.2 Distance Metric Learning

Besides, there is a new trend which learns the optimal metric by using state-of-the-art machine learning algorithms. In a survey by Bellet et al. [2013] on metric learning for feature vectors and structured data, various metric learning algorithms are categorized and summarized, and recent trends and extensions are discussed. One can easily find a suitable algorithm for one's own application based on the main features of metric learning methods.

Basically, distance metrics can be learnt for similarity measurement, e.g., for image similarity measurements. The classic SVM approach is to develop an algorithm which provides a flexible way of describing qualitative training data as a set of constraints Schultz and Joachims [2003]. Very recently, a regularized distance metric framework called Semantic Discriminative Metric Learning (SDML) is proposed in Wang et al. [2016]. It ensures the consistency between dissimilarities and semantic distinctions, and avoids inaccuracy similarities; in the end, excellent performance on benchmark image datasets is obtained.

For unsupervised clustering and supervised classification applications, Xing et al. [2003] propose distance metric learning with an application to clustering with side-information. In order to obtain large margin nearest neighbor classifiers, Weinberger and Saul [2009] also propose to learn a Mahalanobis distance metric which significantly improves the classification performance of the k nearest neighbor algorithm. Moreover, Park et al. [2011] focus on solving the large margin nearest neighbor optimization problem more efficiently, while learning a Mahalanobis distance metric. For image retrieval, Discriminative Component Analysis (DCA) and kernel DCA which outperform the common relevant component analysis are proposed for learning distance metrics with contextual constraints Hoi et al. [2006]. Propelled by the rapid development of machine learning algorithms, Hoi et al. [2010] implemented a semi-supervised learning scheme for collaborative image retrieval, and compared it with standard metrics, unsupervised metrics, and supervised metrics with side-information.

In order to get a clear idea about the advantages and disadvantages of different distance learning methods, Yang [2006] made a very comprehensive and thorough survey about distance metric learning. The topic is reviewed under different learning conditions: unsupervised learning versus supervised learning; learning in a global sense versus in a local sense; and linear kernel based distance metrics versus nonlinear kernel based distance metrics. More recently, a systematic review of the metric learning literature is compiled in Bellet et al. [2013], in which the Mahalanobis distance metric learning is discussed, as it emerged as a powerful alternative. Other than calculating the similarity between points, Zhu et al. [2013] propose to extend the idea to evaluating the similarities between one point and one set, and between sets. The corresponding experiments on gender classification, digit recognition, object categorization and face recognition show that the performance of point-to-set distance and set-to-set distance based classifications have improved. In remote sensing applications, almost no related researches have been preformed.

3.4 Machine Learning

As is mentioned in Jiang [2009], feature extraction reduces the original 2-D image data into a high-dimensional feature vector while annotation or classification then guarantee the numerical feature vector to a binary one, which corresponds to image annotations with regard to remote sensing image annotation.

The statistical properties of remote sensing images become complicated because of the multi-spectral properties of a pixel location, the presence of different kinds of noise sources and uncertainties, the inherent non-linear data model, and the high spatial and spectral redundancies. The special characteristics of remote sensing data and the variety of application objectives give rise to the application of a wide range of machine learning and signal processing algorithms.

There are specific technical terms both used in the machine learning and statistics communities. A comparison which shows typical terms is shown in Table 3.2. This table is based on Bob Tibshirani's amusing comparison between machine learning and statistics[1]. It is interesting to note that, when reviewing the literature of machine learning or computer vision, it is easy to get confused by different sets of terminology used by the authors. In this dissertation, technical terms will follow the direction of machine learning, which appears to be more practically orientated and better to understand.

[1] http://statweb.stanford.edu/~tibs/stat315a/glossary.pdf

Table 3.2: Glossary of machine learning and statistics.

Machine Learning	**Statistics**
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression or classification
unsupervised learning	density estimation or clustering

3.4.1 Classic Machine Learning Methods

Nowadays, a number of machine learning techniques which include classification, regression, and clustering are successfully applied in different fields. The commonly used methods are: statistical methods, Bayesian methods, probabilistic graph models, etc.. Tables A.1, A.2, A.3, A.4 in the Appendix give an overview of basic and advanced machine learning algorithms which range from unsupervised learning to supervised learning regarding aspects of algorithm descriptions, models, objective functions, and training algorithms.

Table A.1 describes three types of unsupervised learning algorithms which cluster data into groups based on chosen metrics, e.g., different distance metrics, Kullback-Leibler (KL) divergence, information gain, etc.; Table A.2 explains three basic supervised learning algorithms which make simple assumptions about clean and organized data; in Table A.3, the classic perceptron algorithm is shown at first, then the details of more advanced convolutional neural networks are explained, via models, objective functions, and training algorithms; Table A.4 lists three advanced learning algorithms: decision trees, random forests, and SVM. For a wide range of applications, they yield robust and efficient performance, for example, content based image retrieval, facial expression classification, personal recommendation systems, handwritten character categorization, etc.

3.4.2 New Trends in Semi-supervised Learning

In order to prevent making EO analyses in a very laborious way, in Datcu and Seidel [2005] a remote sensing information processing system Knowledge-driven Information Mining (KIM) is proposed which is based on human-centered concepts to solve difficult tasks in EO image interpretation. In this context, image annotation can be performed with supervision, without supervision, or with "intermediate-level" supervision. It includes supervised learning

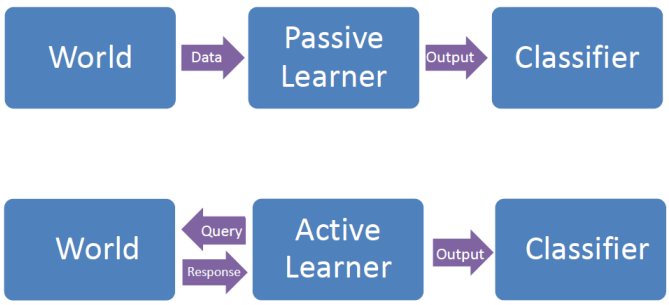


Figure 3.3: Active learning scheme vs. passive learning scheme.

Carneiro et al. [2007], unsupervised learning Lienou et al. [2010], Su et al. [2011], as well as the "intermediate" form of supervision, i.e., semi-supervised learning Blanchart and Datcu [2010], or active learning Blanchart et al. [2014]. As shown in Fig. 3.3, active learning is a special case of semi-supervised machine learning in which a learning algorithm is able to interactively query the user or some other information source to obtain the desired outputs at new data points Settles [2014], Olsson [2009]. Driven by the motivation of applying high-quality state-of-the-art algorithms, many researchers dealing with remote sensing image annotation have exploited a full supervision for image content understanding:

In a 20 meter resolution pixel-level land cover classification Rodriguez-Galiano et al. [2012], a Random Forest (RF) algorithm (see Table A.4 in the Appendix) which is robust to training data reduction and noise yields accurate land cover classifications. Those variables that the RF algorithm identified as most important for classifying land cover also coincided with our expectations.

In a review of SVM (see Table A.4 in the Appendix) applied in remote sensing Mountrakis et al. [2011], SVMs are particularly appealing in the remote sensing field due to their ability to generalize well even with limited training samples, a common limitation for remote sensing applications. However, they also suffer from parameter assignment issues that can significantly affect the obtained results. A summary of empirical results is provided for various applications of over one hundred published works. It is their hope that this survey will provide guidelines for future applications of SVMs and potential areas of algorithm enhancement.

Latent Dirichlet Allocation (LDA), as a generative model, is often used for classification in natural language processing. The original algorithm and its variations have been introduced and extended into the remote sensing domain by Lienou et al. [2010], Luo et al. [2012], and Zhang et al. [2010]. Similarly, inspired by the relationships among chromosomes, DNA, and genes in biological systems, an image-to-concept distribution model is proposed by Su et al. [2011] to obtain reliable semantic annotations.

Moreover, in the context of dealing with huge amounts of data, unsupervised clustering methods (see Table A.1 in the Appendix) which group visually similar results together Cao et al. [2009], Choi et al. [2010], Wang et al. [2014], Lu et al. [2010], and Wang et al. [2008] usually reduce the time required to generate an annotated dataset Chen and Ellis [2015]. In order to better evaluate the clustering results, research done in Varga and Nedevschi [2013] formulates a new definition of compactness which can be used as a similarity measure between feature descriptors.

As both supervised learning and unsupervised learning have their own advantages and disadvantages, a probabilistic formulation, which combines the advantages of the two methods via a reformulation of the supervised approach, is proposed by Carneiro et al. [2007] for semantic image annotation and retrieval. A semi-supervised algorithm which trains a hierarchical latent variable model with both labeled and unlabeled data is proposed by Blanchart and Datcu [2010] for auto-annotation and unknown structure discovery in satellite images, and a multi-scale coarse-to-fine cascaded active learning method to retrieve patterns in large image datasets, is proposed by Blanchart et al. [2014]. As complex scenes are difficult to describe with a single label for each image patch, a hierarchical semantic multi-instance multi-label learning (MIML) framework for high resolution remote sensing image annotation via a Gaussian process is proposed by Chen et al. [2013].

In two review papers of content based image retrieval Gevers and Smeulders [2003], and Datta et al. [2008], the authors discuss generally and thoroughly about the learning techniques that are traditionally used in image retrieval. As described in Datta et al. [2008],

three different machine learning techniques and their applications in image retrieval are available: clustering, classification, and relevance feedback. Clustering has the advantage of exploring unknown information but with poor user adaptability; classification makes better predictions but is limited to existing labeled training classes; relevance feedback provides more user involvement. The authors mention the importance of user interaction which shows a positive trend for future image retrieval applications. Both active learning and semi-supervised learning are important techniques when scarce labeled data are available. A Gaussian random field model is used to combine them, while the semi-supervised learning problem is formulated in terms of a Gaussian random field on a weighted graph, the mean value of which is characterized in terms of harmonic functions. Then active leaning is performed on top of this semi-supervised learning Zhu et al. [2003]. In an overview of image information mining in the exploration of EO archives by Datcu and Seidel [2003], the authors clarify the trend from a computer-centered approach, for example, in content based image retrieval to a human-centered approach, for example, in image information mining. The huge volume of data, the variability and heterogeneity of the image data which include a diversity of sensors, times or conditions of acquisition, etc., has made image information mining a task with high complexity. In this review, a theoretical concept for image information representation and adaptation to the user conjecture is developed, in which the image data is represented via image features and meta features, then a clustering and learning algorithm is applied to generate a cluster model; in the end, a semantic representation is obtained.

Furthermore, when adapting learning methods to user experience, Blanchart proposes in his PhD dissertation a semi-supervised algorithm which exploits "unknown" semantic structures and tries to solve the problem of learning in interactive image search engines which relates to active learning Blanchart [2012]. The results show that the introduction of active learning is able to handle large volumes of data while preserving a satisfactory level of accuracy. Sometimes, there is a confusion between content-based image retrieval and image information mining Quartulli and Olaizola [2013]; however, this term is explained in the Nomenclature chapter. In order to minimize the total annotation cost within an active learning process, Persello et al. [2014] address the active sample selection problem in the framework of a Markov decision process, in which one is allowed to plan the next labeling action based on an expected long-term cumulative reward. Hence, we can see that there is a clear trend of integrating semi-supervised learning and active learning methods in the framework of classification or annotation applications.

Additionally, in some cases, extra information can also be considered for image annotation, e.g., visual contexts can be learned for image annotation based on Flickr group labels Ulges et al. [2011]. Regarding the high-dimensional feature space, a semantic image browser which applies existing information visualization techniques to semantic image analysis is proposed in Yang et al. [2006]. There the high-dimensional feature space is shown without dimension reduction, and also a rich set of interaction tools are integrated. For geospatial analysis, this is a common approach to process data from heterogeneous geospatial databases. In Shyu et al. [2007], new techniques with shape and visual characteristics, multiobject relationships and semantic models which link low-level image features with high-level visual descriptors are proposed. The system is able to answer image analysts' questions in seconds which allows image analysts to identify relevant imagery more rapidly. A query method which combines data metadata, image content and image semantics is presented in Espinoza-Molina and Datcu [2013], and Dumitru et al. [2014]. The system is focused on processing TerraSAR-X data and is able to analyze sets of

high-resolution TerraSAR-X image scenes with more than 300 semantic annotations.

3.4.3 Object Extraction-based Semantic Exploration

In the sense of semantic exploration, it is desirable to obtain information (i.e., annotation labels) up to the object-based pixel level. An object-oriented analysis can contribute to most remote sensing applications as pixel-level processing explores richer information contents Benz et al. [2004]. For SAR images, due to their image properties, it is still difficult to process the data up to the pixel level; however, researchers have already tried to extract various objects from high-resolution optical images. For example, in Guo et al. [2009], a semantics-aware two-stage image segmentation approach integrated with a hyperclique pattern discovery method is proposed. In Blaschke [2010], the authors give an overview of the development of object-based image analysis methods for remote sensing, which aim to delineate readily usable objects from imagery, and combine image processing and GIS functionalities in order to utilize spectral and contextual information in an integrative way.

As the object-based segmentation for remote sensing imagery requires pixel-level precision, most of the applications are done with a multi-level or multi-scale approach which allows refinement of segmentation regarding different levels of objects. The authors of Chaabouni-Chouayakh and Datcu [2010] propose to fuse different automatic object extractors from coarse to fine level which is able to provide reliable pieces of interpretation. In order to extract semantically meaningful objects for object-based remote sensing image analysis, a multi-scale analysis is always needed during segmentation. In Yi et al. [2012], the multi-scale segmentation first divides the whole image area into multiple regions where each region consists of ground objects within similar optimal segmentation scales. Then those suboptimal segmentation regions are synthesized to get the final result. The proposed scale-synthesis method can generate accurate segmentation results that benefit latter classification procedures. For object-based classification, a multiagent object-based segmentation algorithm is proposed to optimally control the procedure of object merging Zhong et al. [2014]. The contextual information includes strong interaction, high flexibility, and parallel global control capability from the surrounding objects. The experimental results show that the algorithm yields a stable and competitive performance for high-resolution remote sensing images.

Moreover, the segmentation results are obtained in a iterative manner or within a scalable tile-based framework. In Mylonas et al. [2015], the image is segmented in an iterative manner; at each iteration, a single object is extracted via a genetic algorithm-based object extraction method. As the objective of remote sensing applications is often to adapt image segmentation algorithms for large amounts of data, large images are usually divided into smaller image tiles to overcome the memory problem; in this case, each tile is processed independently. A scalable tile-based framework for region-merging algorithms to segment large images is proposed in Lassalle et al. [2015]. The results show the benefits of this framework and demonstrate the scalability of this approach by applying it to really large images.

The first segmentation results are not always optimal, and a lot of post-processing methods are proposed to refine the segmentation results. For high-resolution remote sensing imagery, a novel texture-preceded segmentation algorithm is proposed by Li et al. [2010], in which texture clustering is carried out first as a loose constraint for subsequent segmentation. The method can merge homogeneous texture segments easily and also well detect real object boundaries. In Johnson and Xie [2011], an optimal image segmentation is identified using an unsupervised evaluation method of segmentation quality, the under- and over-segmented

regions are refined accordingly, and a multi-scale approach is used for segmentation refinement. In Stefanski et al. [2013], a strategy for a semi-automatic optimization of object-based classification of multi-temporal data is presented by applying a Random Forest (RF) approach. Troya-Galvis et al. [2015] propose a novel unsupervised metric, which evaluates the local quality per segment by analyzing segment neighborhoods and a specific homogeneity criterion is given to quantify under- and over-segmentation. The behavior of the proposed metric is analyzed and validated. A detail-preserving smoothing classifier based on conditional random fields with a competitive performance is proposed in Zhao et al. [2015]. Moreover, in Geiss and Taubenboeck [2015], the authors propose a supervised classification model which is trained for the refinement of second time segmentation results; here, additional information is provided from the initial classification outcome.

When multi-level processing and specific learning methods may not provide globally acceptable solutions for remote sensing applications, some authors seek support from extra information or build very domain-specific models. In Bouziani et al. [2010], as high-resolution urban areas usually contain many objects which form a very complicated segmentation task, the authors use an existing digital map of the same area within a classification process where an automated multi-spectral segmentation algorithm is applied. In particular, for building detection, a domain-specific segmentation method is proposed in Karadag et al. [2015], which integrates information related to the building detection problem into the detection system during the segmentation step. Although this approach may obtain very precise segmentation results regarding a specific type of object segmentation, the drawback of it is that the data model is so specific and limited to the usage of a given application, and is not robust enough to be extended to other applications.

Other machine learning methods were also applied to the semantic segmentation of remote sensing images. Some authors propose an automated selection of a single segmentation level with the Hierarchical SEGmentation (HSEG) algorithm Tarabalka et al. [2012], which gives a good performance for multi- and hyper-spectral image analysis by combining region object finding with region object clustering. A total variation formulation which is an effective tool for image restoration, enhancement, reconstruction and diffusion is proposed in Zhang et al. [2013]. The relations among different objects in high-resolution remote sensing images are discussed in Vanegas et al. [2013], namely the alignment and parallelism for image description; illustrative examples on optical satellite images show the descriptive power. A morphological profile-based attribute profile method with special emphasis on remote sensing image classification is contained in Ghamisi et al. [2015]. A superpixel-based graphical model is proposed in Zhang et al. [2015], where gradient fusion, probabilistic fusion, and boundary penalties between the super-pixel methods are introduced in the procedure of classification. The results outperform the other state-of-the-art methods.

Similar to image annotation, regarding image segmentation for high-resolution remote sensing imagery, the importance of user interaction is also emphasized. Some authors propose to apply Graph Cut (GC) theory within a labeling problem on a Markov Random Field (MRF) constructed on the image grid Bai et al. [2012]. In this case, a large number of interactive user strikes are needed to obtain satisfactory segmentation results. In a review particularly for hyper-spectral image classification Camps-Valls et al. [2014], the authors discuss the performance difference among traditional purely supervised solutions, semi-supervised solutions and active learning solutions, then address the emphasis of user interaction for future hyper-spectral image classification.

3.5 Conclusions and Proposed Concepts

We start from a huge amount of available **data**, then extract **information** via data analysis, and, in the end, **knowledge** is acquired by using different data processing techniques. In the Motivation and State of the Art chapters, three research goals have already been brought out, namely:

1. To annotate images in large-scale remote sensing datasets.
2. To study the extracted features in their high-dimensional feature space.
3. To obtain object-based extraction for high-resolution optical satellite images.

Based on the above literature review, we propose an intuitive rough idea of how to decompose the big complex problem into smaller parts and represent image content on three levels: scene, patch, and object. The full scene level covers a large area with an abundance of detailed information; the patch level covers local semantics ranging from general to detailed levels, for example, from urban areas to high-density residential areas; finally, the object level depicts very detailed object information by extracting object profiles.

As indicated by Sinha and Poggio [1996], human vision relies heavily on context. For optical and SAR images, image content can be represented via different levels. As shown in Fig. 1.2, the optical image content can be represented as a hierarchical structure of a scene, patches, and objects. On the full scene level, the figure shows a high-density urban area in Munich, Germany, where the typical city elements (e.g., residential areas, industrial areas, railways, rivers, etc.) are clearly visible. On patch level, three patches represent different scales of a residential area, together with other land cover types. On object level, high-resolution optical satellite imagery provides us with a good opportunity to visually recognize detailed objects: a white square building, a gray round building, a tennis court, a pond, etc.

Similar to Fig. 1.2, SAR image content can also be represented on scene-patch-objects levels, as shown in Fig. 1.1. On the full scene level, the whole scene depicts the city of Siegen, Germany within a large area of forests and agricultural fields; On patch level, three patches depict a building within a field, an industrial area within a field, and a residential area. On object level, the objects are visually very difficult to identify, as they have different visual characteristics compared with optical images. Among the given examples, the football playground is dark because its smooth surface generates low backscattering towards the observer; the industrial building has very bright lines along its edges because of metallic material on the roof and double scattering due to the given incidence angle; residential houses are represented by clusters of bright spots. More detailed SAR image properties are explained in the Data Characteristics and Basic Mathematics chapter.

Fig. 3.4 describes our proposed overall semi-supervised framework from two perspectives: patch-level classification, and pixel-level segmentation. Both of them consist of three basic modules: determination of image characteristics, machine learning algorithms, and semantic annotation. After that, either hierarchically grouped patches or extracted objects are obtained. Then different semantic labels are attached to patches and objects. More solution details will be explained and further developed in the following chapters.

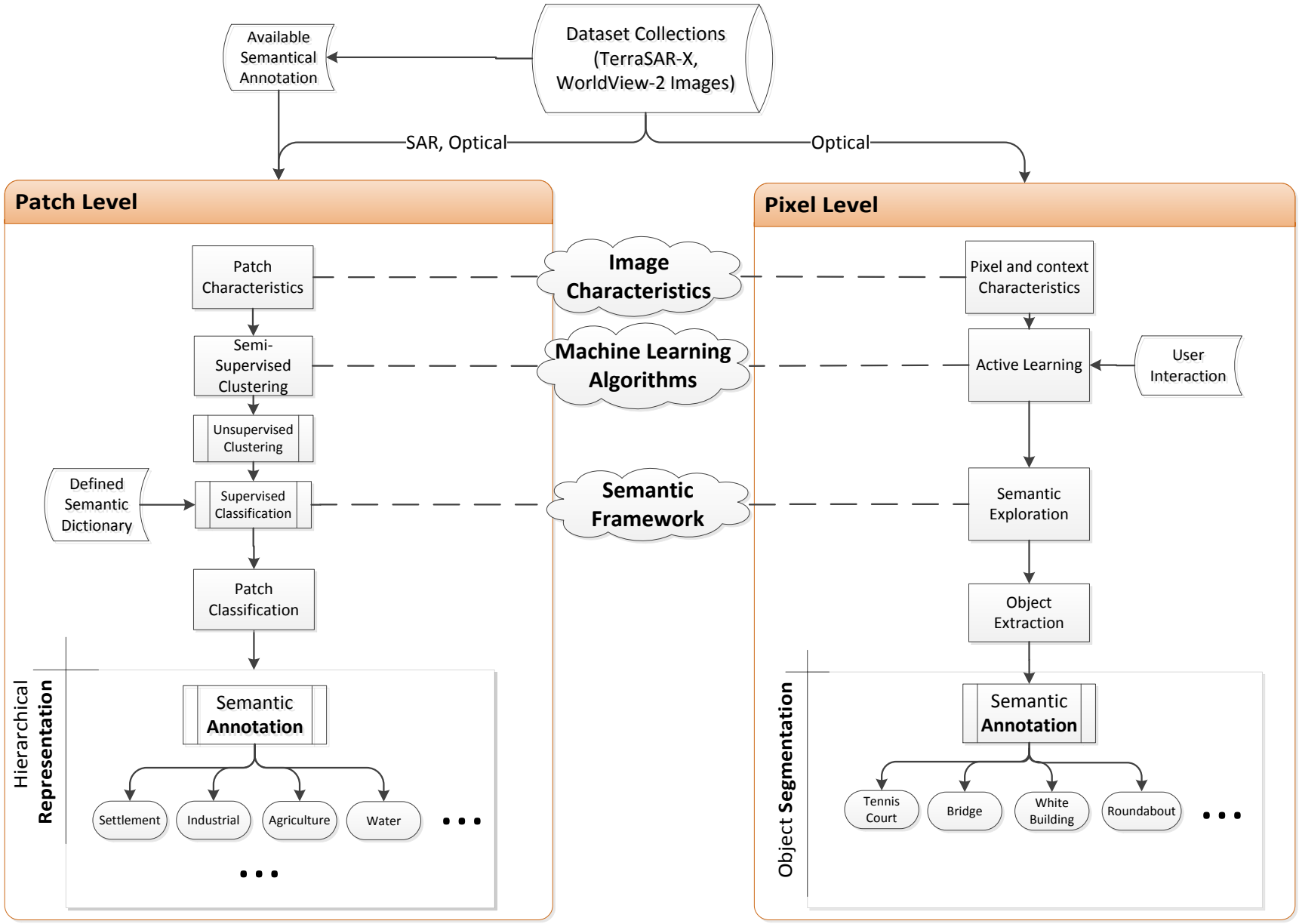


Figure 3.4: Semantic annotation framework from patch-level and pixel-level perspectives. It consists of three main modules: image characteristics, machine learning algorithms, and semantic framework.

3.5.1 Applied Dataset

Since its launch, the German TerraSAR-X satellite has already acquired tens of thousands of high-resolution Synthetic Aperture Radar (SAR) images that have been processed into different product types. Their resolution is usually 1-5 meters/pixel; thus, very detailed structures (e.g., industrial constructions, residential buildings, etc.) are clearly visible in the images Fritz [2013].

Based on the available data products that we have at our disposal, we focus in this dissertation on Multi-look Ground Detected (MGD) and Radiometrically Enhanced (RE) high resolution TerraSAR-X SpotLight mode products. In this mode, the resolution of TerraSAR-X images is about 2.9 m with a pixel spacing of 1.25 m. The average size of the given images is 4,500×6,500 pixels. We generated a semantically annotated dataset consisting of more than 100 scenes covering different urban and non-urban areas from around the world (see Fig. 3.5). Each image was tiled into patches of 160×160 pixels; thus, about 110,000 patches were generated.

Due to the specific characteristics of SAR imaging systems, there are some unique phenomena that can be observed in a SAR image, which make the interpretation of a single SAR image more challenging compared to an optical image. For instance, there is the notorious speckle phenomenon which has the so-called "pepper and salt" noise effect. Apart from their specific data characteristics, there are also other bottlenecks for SAR satellite image interpretation: artifacts due to ortho-rectification, highly unbalanced classes, a small number of training data, and many labeling errors caused by human factors, etc. Hence, in this dissertation, we focus on individual SAR images with single polarization.

Besides SAR products, we also analyzed optical images of WorldView-2, a commercial EO satellite which is owned by DigitalGlobe. WorldView-2 provides commercially available panchromatic imagery with a resolution of 0.46 m, and eight-band multispectral imagery of 1.84 m resolution. It was launched on October 8th, 2009 as the third DigitalGlobe satellite in orbit, joining WorldView-1 which was launched in 2007 and QuickBird which was launched in 2001. The satellite revisit period for any place on Earth is 1.1 days Wikipedia [2016]. In August 2014, WorldView-3 was launched complementing the fleet of WorldView satellites.

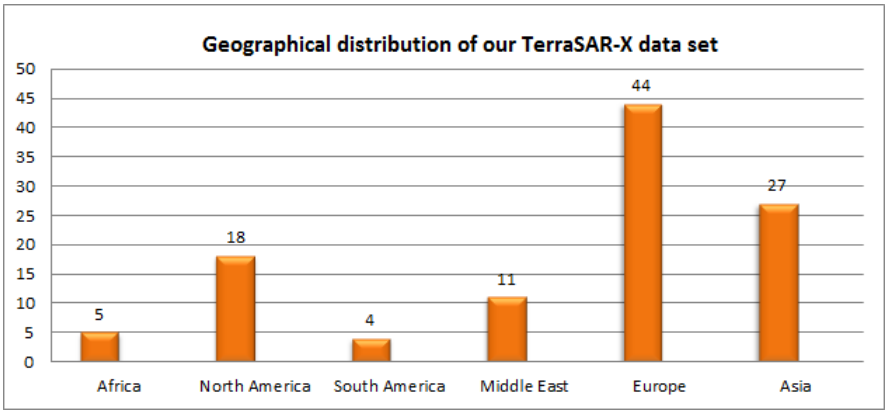


Figure 3.5: Geographical distribution of our semantically annotated TerraSAR-X data set.

3.5.2 Semi-supervised Learning

Due to the large data volume of satellite images, their scene content complexity, and the computational effort of classification and labeling algorithms, we had to cut large image scenes into small patches for subsequent object extraction. Hence, hundreds of scenes become hundreds of thousands of patches. In order to save image interpreter time by avoiding manual labeling of training data, which is necessary for supervised learning, an intuitive idea comes to our mind: Why not simplify this large-scale data problem at the beginning by grouping similar patterns together? We know that unsupervised learning methods are able to group similar patterns together. Fig. 3.6 shows a rough idea of this concept. Here, similar patches can be grouped together based on different patch characteristics such as square-like patterns, homogeneous textures, scattered short and long lines, etc. To be independent from the actual statistical distribution of the image data, we aim at a **homogeneous** clustering of feature vectors which allows us to group similar patches within the dataset as proposed by Datcu and Schwarz [2010]. We assume a Gaussian distribution for homogeneous cluster content because it is rather general and robust (not case-specific). Thus, we hope to obtain homogeneous clusters prior to final semantic annotation based on suitable cluster splitting.

In hierarchical clustering, the main problem is to find a suitable criterion for cluster splitting. Based on the method proposed in Hamerly and Elkan [2003], the Anderson-Darling Gaussian test Anderson and Darling [1952] performs better than the Bayesian Information Criterion (BIC) in finding a good stopping rule during clustering. Therefore, the Gaussian-test was used for creating our hierarchical structures. In order to make the whole process as simple as possible, we applied an adjusted Gaussian test based clustering method.

During the construction of the hierarchical structure, we used K-means clustering of image patch feature vectors for cluster splitting due to its simplicity and efficiency. We incorporated a fractional distance metric and the Minkowski distance metric into k-means while calculating the distances to tackle the "curse of dimensionality" problem. In order to better evaluate the homogeneity of the grouped clusters, the evaluation was made from two aspects: image patch visualization as well as feature space cluster density analysis.

However, an unsupervised learning method alone cannot provide reliable semantic information to users. As known, it is hard to link unsupervised learning results to physically meaningful semantic labels. It would be a good solution to add some prior information and

Table 3.3: Table of semantic categories.

No.	Abstract Level	Detailed Level
1	Agriculture	Agricultural land
2	Transportation	Airport
3	Transportation	Airport building area
4	Transportation	Boat
5	Water bodies	Breaking waves
6	Transportation	Bridge
7	Water bodies	Channel
8	Bare ground	Cliff
9	Bare ground	Desert and Sand
10	Forest	Forest mixed
11	Forest	Forest mixed and Stubble
12	Urban area	High-density residential area
13	Bare ground	Hill
14	Urban area	House in residential area
15	Industrial area	Industrial area
16	Urban area	Low-density residential area
17	Urban area	Medium-denisty residential area and 2
18	Urban area	Medium-density residential area and 1
19	Military facilities	Military aerospace facilities
20	Urban area	Mixed urban area
21	Urban area	Mixed urban area and Stubble
22	Water bodies	Ocean and 1
23	Water bodies	Ocean and 2
24	Transportation	Port
25	Transportation	Railway
26	Transportation	Road
27	Urban area	Skyscraper
28	Urban area	Sport area
29	Industrial area	Storage tank
30	Agriculture	Stubble and Road

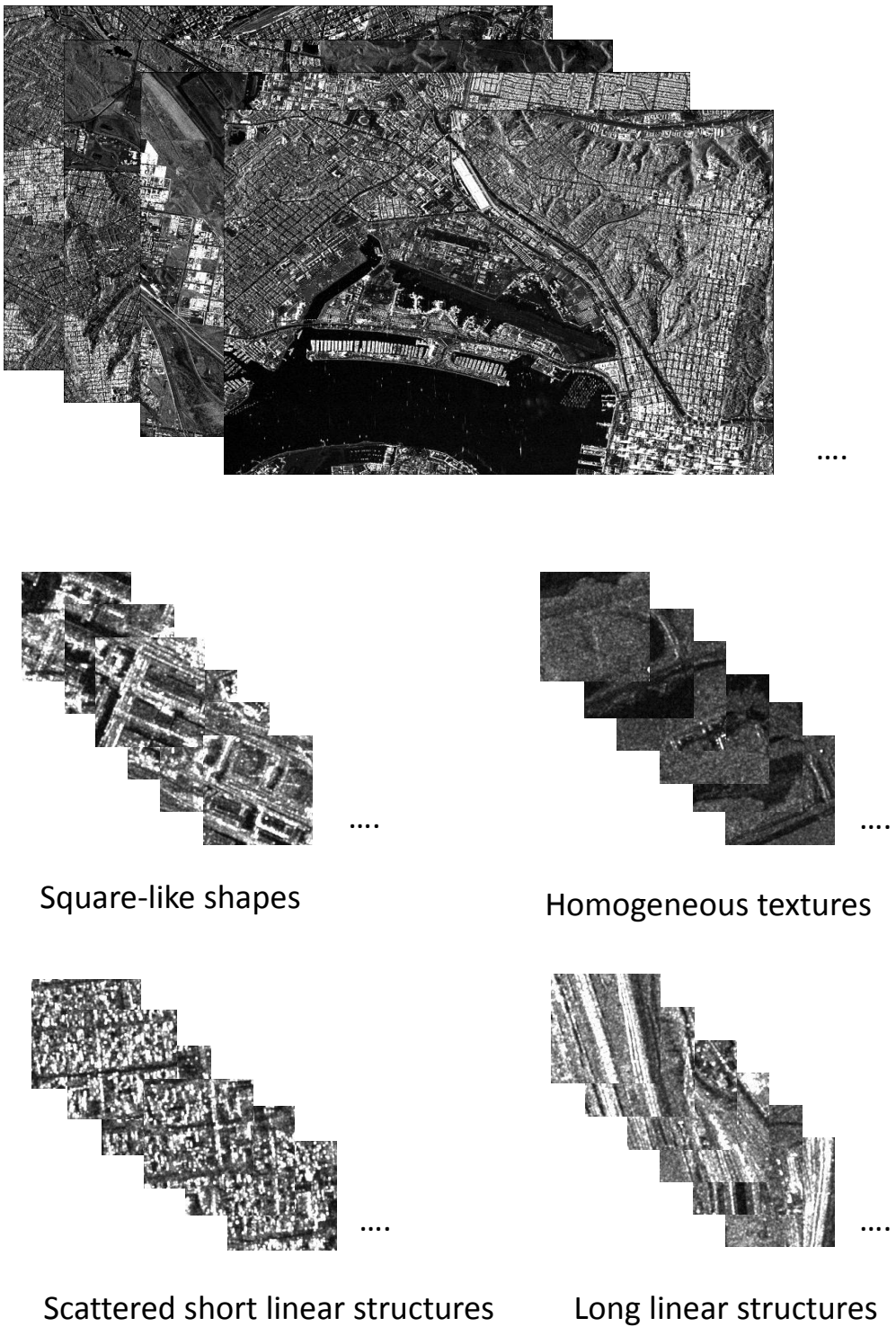


Figure 3.6: Unsupervised clustering for TerraSAR-X image patches.

human interaction, which leads to the use of semi-supervised learning methods: A small amount of labeled data is used for training with a large amount of unlabeled data remaining for subsequent analysis. Thus, a cluster-then-label semi-supervised learning method Zhu and Goldberg [2009] was adapted and modified. Different supervised learning methods were tested within clusters, in particular, methods related to nearest neighbors have been paid attention to, as we are interested in the effects of distance metrics on semantic annotation. Table 3.3 lists the semantic categories that we use in this dissertation.

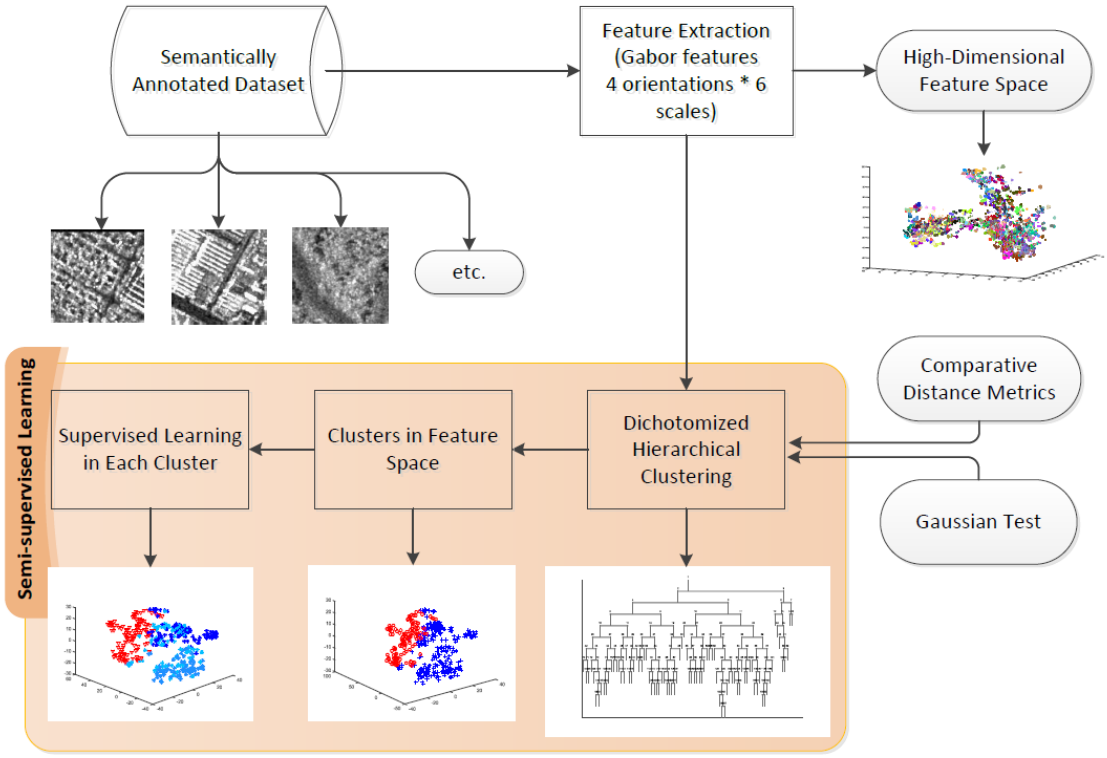


Figure 3.7: Proposed processing and analysis procedure for image patches.

Fig. 3.7 illustrates the whole concept of our proposed patch level processing and analysis procedure. We start from a semantically annotated dataset of image patches, from which we select a number of collections. Then we perform feature extraction for each image patch. This results in a high-dimensional feature space. A clustering algorithm with a Gaussian test and distance metric are used to generate a dichotomized hierarchical cluster tree structure. After applying supervised learning in each cluster, predefined semantic labels are assigned to the feature vector points. Since the object-level is for semantic exploration, so there are no predefined semantic labels.

Fig. 3.8 demonstrates the whole concept of our proposed pixel level processing and analysis procedure. Optical images are taken as input, and a user interface is set up for handling positive and negative training samples. Based on pixel distance and similarity information, a non-locality map is calculated, which is used for iterative active learning. At the end, user-selected objects are extracted via object segmentation.

Chapters 4,5 and 6 follow the procedure proposed in this chapter.

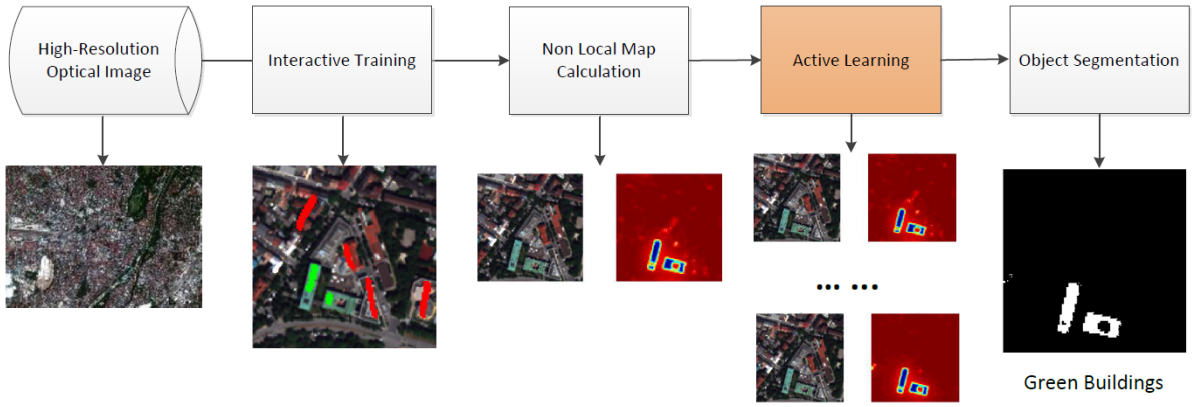


Figure 3.8: Proposed processing and analysis procedure for object extraction.

Chapter 4

Application and Evaluation of a Hierarchical Patch Clustering Method for Image Patches

A setback may turn out to be a blessing in disguise.

Chinese proverb

In this chapter, we apply and evaluate a modified Gaussian-test-based hierarchical clustering method for high-resolution satellite images. The purpose is to obtain homogeneous clusters within each hierarchy level which later allow the classification and annotation of image data ranging from single scenes up to large satellite data archives. After cutting a given image into small patches and feature extraction from each patch, k-means is used to split sets of extracted image feature vectors to create a hierarchical structure. As image feature vectors usually fall into a high-dimensional feature space, we test different distance metrics, in order to tackle the "curse of dimensionality" problem. By using three different SAR and optical image datasets, Gabor texture and Bag-of-Words (BoW) features are extracted, and the clustering results are analyzed via visual and quantitative evaluations. We also compared our approach with other classic unsupervised clustering methods. The most important contributions of this chapter are the discussion and evaluation of cluster homogeneity by comparing various datasets, feature descriptors, evaluation measures and clustering methods, as well as the analysis of the clustering performances under various distance metrics. The results show that the Gaussian-test-based hierarchical patch clustering method is able to obtain homogeneous clusters, while Gabor texture features performs better than the BoW features. In addition, it turns out that a distance parameter ranging from 1.2 to 2 performs best. Also indicated by Yao et al. [2016a], our modified G-means algorithm is faster than the original algorithm.

4.1 Approach

Driven by the motivation to apply high-quality state-of-the-art algorithms, many researchers dealing with remote sensing image annotation prefer a full supervision for image content understanding. However, due to the large data volume and the scene content complexity in our applications, we shifted our attention towards an alternative solution: Why not simplify

the problem at the beginning by grouping similar data together as a first step to generate sets of homogeneous clusters.

So in this dissertation, we apply an unsupervised clustering method as a preliminary step to obtain homogeneous clusters of similar patches within image databases. Specifically, our method intends to answer the following questions: How do we define homogeneity, how do we obtain homogeneous groups, and how do we evaluate the results?

For a large complex image scene, it is preferable to represent the image content as a hierarchical structure with abstract and detailed levels. For example, urban areas can be categorized into sub-classes such as densely built-up areas or sparsely built-up areas. In addition, a hierarchical structure will provide a lot of information describing the connections between different layers. However, in hierarchical clustering, the main problem is to find a suitable criterion for cluster splitting. Based on the method proposed in Hamerly and Elkan [2003], the Anderson-Darling Gaussian test performs better than the Bayesian Information Criterion (BIC) in finding a good stopping rule during clustering. Hence, the Gaussian-test was used for creating our hierarchical structures. Further details will be explained in the Methodology section.

Prior to clustering, we cut all given images into small patches and extract feature vectors from each patch. For high-dimensional image feature vectors, the well-known "curse of dimensionality" problem exists. In addition, for a specific feature descriptor, different distance measures can lead to large discrepancies in clustering performances. For example, in Aggarwal et al. [2001], the fractional distance metric is claimed to outperform classic Euclidean and Manhattan distance metrics. Thus, we have to analyze the performances of different distance metrics. Therefore, the fractional, L1, L2, and Lp (Minkowski) distance metrics are also studied in the Methodology section.

In order to better evaluate the homogeneity of the grouped clusters, the evaluation is done from two aspects: image patch visualization as well as feature space cluster density analysis. Then we continue with the preparation of various datasets that we need for the evaluation and validation of our method. This includes SAR and optical datasets, with semantic references, as well as the internal clustering validation, analysis of annotation credibility, and representativeness of the dataset.

To deal with the above issues, we present a Gaussian-test-based hierarchical clustering method for space-borne TerraSAR-X and airborne optical images. K-means clustering of image patch feature vectors is used to split the clusters during the construction of the hierarchical structure due to its simplicity and efficiency. The fractional distance metric is incorporated in k-means while calculating the distance for tackling the "curse of dimensionality" problem.

For cluster creation, it is like creating a new lexicon, each cluster corresponds to a letter, which we can look up in the dictionary. Creating a dictionary is useful for indexing and organizing a scalable database.

Since the annotation can be made with a big variety of different datasets, the class distributions in the feature space are different, and the annotation labels become quite different, too. In order to simplify the problem and make it less ambiguous, we use a Gaussian assumption for homogeneous cluster content because it is rather general and robust (not case-specific). Thus, we can obtain homogeneous clusters prior to final annotation.

4.2 Methodology

As depicted in Fig. 4.1 and suggested by Datcu and Schwarz [2010] and Newsam et al. [2003], after obtaining homogeneous image patch clusters, they can be used as a basic component of an image retrieval system to find similar images and to exploit hidden information. Hence, in this section, a detailed description of our proposed method, the Gaussian-test-based hierarchical clustering and the related evaluation metrics are given.

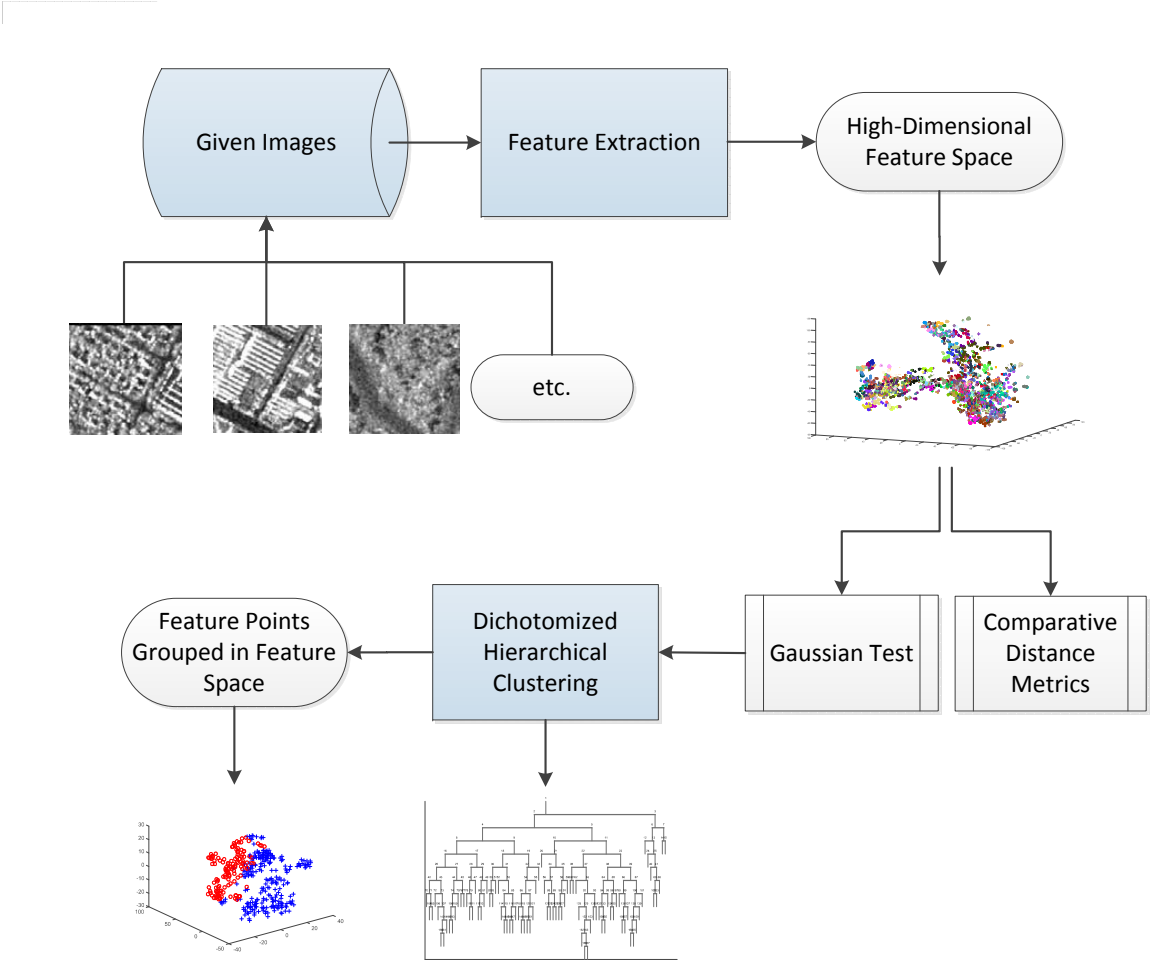


Figure 4.1: Proposed data clustering and analysis scheme for image patches.

A problem of most off-the-shelf clustering algorithms is the need of selecting the appropriate number of clusters. With a hierarchical clustering structure, this can be approached alternatively by a proper splitting rule. In order to get homogeneous clusters, we chose to use the Gaussian-means algorithm proposed in Hamerly and Elkan [2003] that splits clusters based on their actual feature vector contents. The splitting rule in this algorithm is to check whether the cluster follows a Gaussian distribution and to perform splitting if necessary. Thus, there is no need to preset a fixed number of clusters. To prevent too small clusters, we add a constraint on the minimum size of clusters. In the following, we first present the Gaussian-means algorithm and then describe the fractional distance metric.

4.2.1 Feature Extraction

In order to analyze the performance of the modified G-means method (see below) with different remote sensing image sources, SAR and optical image datasets have been used for comparison. As a typical texture feature extractor, Gabor coefficients represent an established standard tool for feature extraction from SAR data as they support robust texture recognition. In contrast, optical images can also be well classified by the Bag-of-Words (BoW) method Yang et al. [2007] since it describes the general as well as the detailed information contained in an image patch.

In the experiments, we have used 2D Gabor wavelet coefficients as texture features. As a typical texture feature extractor, Gabor coefficients represent an established standard tool for feature extraction from SAR data as they support robust texture recognition.

$$g_{\lambda,\theta,\psi,\sigma,\eta}(x,y) = \exp\left\{-\frac{x'^2+\lambda^2 y'^2}{2\sigma^2}\right\} cos(2\pi\frac{x'}{\lambda}+\psi), \tag{4.1}$$

where $x' = xcos\theta + ysin\theta$, $y' = -xsin\theta + ycos\theta$. Different θ values represent different directions, different σ values control different scale of Gabor filters.

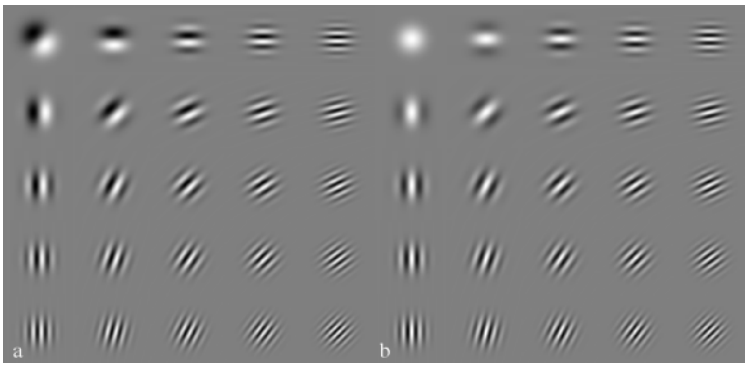


Figure 4.2: An ensemble of odd (a) and even (b) Gabor filters Lee [1996]. This figure shows Gabor filters of 5 scales and 10 orientations.

4.2.2 Hierarchical Clustering

In routine operations, the same Gabor feature extraction steps, which are used to create the reference dataset, are executed for each newly acquired image to be analyzed, however, we had to adapt the feature clustering step.

Since the classification performance of each reference class is unpredictable, it is difficult to expect a direct correspondence between clusters and classes. Each class can spread out into several clusters and we have to generate a hierarchical cluster structure to overcome this difficulty. Due to its simplicity and efficiency, k-means clustering can be used to split the initial clusters to construct a hierarchical structure. As a result, convex clusters are grouped in feature space and homogeneous clusters are obtained. A problem of k-means is to select the optimal number of clusters. This can be reached, however, by a proper splitting rule. Therefore, we use the Gaussian-means (G-means) algorithm proposed in Hamerly and Elkan [2003]. The Central Limit Theory tells us that the arithmetic mean of a sufficiently large number of iterates of independent random variables, each with a well-defined expectation value and a well-defined variance, will be approximately normally distributed, regardless of the underlying distribution.

Our splitting rule is to check whether a cluster follows a Gaussian distribution. Thus, there is no need to preset a fixed number of clusters. To prevent cluster generation with too few patches, we define an additional constraint for a minimum cluster size and we modify the G-means algorithm accordingly.

4.2.3 Modified G-means Algorithm

For cluster generation, we replace the classic k-means algorithm with a modified Gaussian-means (G-means) variant that checks the Gaussianity of the cluster content and performs a cluster splitting in case of non-Gaussianity. The modified G-means algorithm starts with the whole image patch dataset. During each iteration of the algorithm, a cluster of data which does not follow a Gaussian distribution is split into two new clusters, while all clusters that already obey a Gaussian distribution are preserved in the generated hierarchical structure. For computational efficiency, the Gaussian hypothesis test is simplified to process only an n -dimensional vector as the original $n \times p$-dimensional feature vectors are projected onto the vector direction formed by the two new cluster centers. Because of the way the Gaussianity test is performed, a hierarchical clustering structure is naturally created.

4.2.3.1 Gaussian Hypothesis Testing

The Gaussian hypothesis is chosen because of its ability to obtain homogeneous clusters as well as its simplicity of implementation. Based on the method proposed in Hamerly and Elkan [2003], the Anderson-Darling Gaussian-test performs better than the Bayesian information criterion (BIC) in finding a good splitting rule for clustering. Hence, the Anderson-Darling Gaussian-test is used to construct a hierarchical cluster structure, where an existing cluster is split into two new clusters when the null hypothesis is rejected, and the cluster splitting is stopped when the null hypothesis is accepted. For the definition of the null hypothesis, see Hamerly and Elkan [2003]. For hypothesis testing, a confidence level refers to the percentage of all possible samples that can be expected to include the true population parameter. Algorithm 1 describes our modified G-means algorithm, the algorithm is initialized with two clusters by applying k-means once.

4.2.3.2 Anderson-Darling Test

The Anderson-Darling test is a statistical test of whether a given sample of data is drawn from a given probability distribution. When applied to testing whether a normal distribution adequately describes a set of data, it is one of the most powerful statistical tools for detecting most departures from normality Stephens [1974], Stephens [1986]. The following procedure shows how the Anderson-Darling test works:

- Define two hypotheses:
  H_0: the data points around the data set center are sampled from a Gaussian.
  H_1: the data points around the data set center are not sampled from a Gaussian.
- Normalize the data points so that their mean value becomes 0 and their variance equals 1. Let $x(i)$ be the ith ordered value. Let $F(x(i))$ be the $N(0,1)$ cumulative distribution function of the data points, with $z_i = F(x(i))$.
- In our case, the mean value μ and standard value σ are estimated from the data, the corresponding test statistic is:

Data: Feature vector matrix $X_{n\times p}$, a confidence level α, and a minimum cluster size s. n is the number of patches, p is the feature vector dimension.
Result: A hierarchical cluster structure.
Initialize C, D, G, S; //cluster centers, cluster data, Gaussianity booleans, cluster sizes
while *!($g_i = 1$ or $s_i \leq s$)* **do**
  Use k-means with k=2 to split cluster i;
  Obtain vector X' by projecting the feature matrix of cluster i onto the direction defined by the two newly generated cluster centers;
  Use an Anderson-Darling Gaussian test for vector X' under the confidence level α to check whether cluster i follows a Gaussian distribution;
  if *cluster i follows a Gaussian* **then**
    Keep c_i;
    g_i = 1;
  else
    Update C, D, G, S with new cluster parameters;
    Reorder $X_{n\times p}$ according to clusters.
  end
  i = i + 1;
end

Algorithm 1: Modified G-means (X, α, $size$) Algorithm Yao et al. [2016b].

$$
\begin{aligned}
A^2(Z) &= (-\frac{1}{n}\sum_{i=1}^{n}(2i-1)*[\log z_i + \log(1-z_{n+1-i})] - n) \\
A^2_\star(Z) &= A^2(Z)*(1+4/n-25/(n^2)).
\end{aligned}
\tag{4.2}
$$

4.2.3.3 Feature Vector Projection

The commonly used Gaussian hypothesis test is designed for data points along one dimension. In the case of high-dimensional data, the computational complexity increases dramatically. Therefore, the feature vectors are re-projected to make them amenable to one-dimensional Gaussian hypothesis testing Hamerly and Elkan [2003]. The idea of feature projection is explained in the following and Fig. 4.3.

- For a data set X with $n \times p$ dimensions, initialize two centers using the k-means++ algorithm with k=2 Arthur and Vassilvitskii [2007], and run k-means on these two centers in X until convergence is reached.
- The cluster centers c_1, c_2 are obtained. Let $v = c_1 - c_2$ be a p-element vector which connects the two centers. This is the cardinal direction that k-means exploits for clustering.
- Project all the data points X onto this preferred direction v, obtain $x_i'=<x_i, v>/\|v\|^2$. X' is thus a one-dimensional representation of the dataset X projected onto v.
- Normalize X' so that its mean value equals 0 and its variance equals 1. The normalized X' is the projected feature vector which will be used and tested for Gaussianity.

The linear projection works as a reverse case of Projection Pursuit and Independent Component Analysis (ICA). A linear projection of a random vector is a linear combination of

its components. As indicated by the idea of ICA, indirectly due to the central limit theorem, such a projection (i.e., signal mixture) tends to appear to be “more Gaussian” than the components of the original vector Kruskal [1969], Stone [2004].

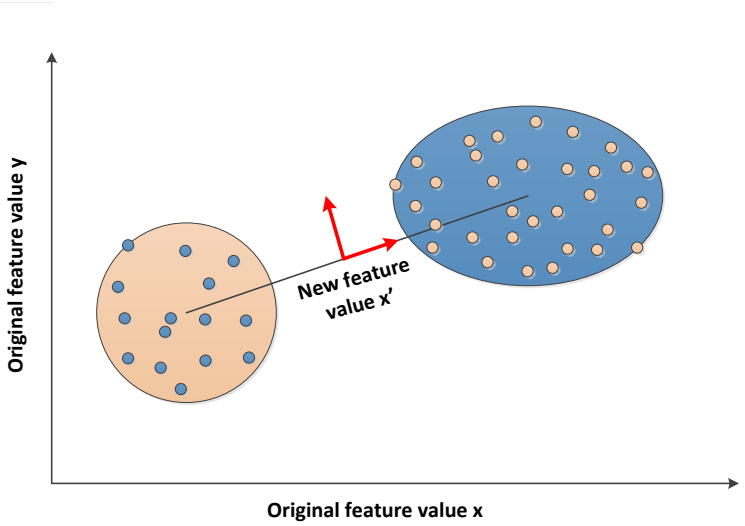


Figure 4.3: Feature projection based on k-means clustering, a Gaussian-test will be used later during the construction of a hierarchical clustering structure.

For the implementation, various distance metrics were experimented with and we used a tree data structure to store the overall clustering structure. The tree structure not only enables us to observe how the entire dataset splits into smaller clusters at different levels, but also provides us the possibility to explore the connection among general and detailed classes.

4.2.4 Comparative Similarity Measures

Since the relationships among feature points in the feature space are measured by pairwise distances, the definition of a distance metric hence plays an important role in exploring structures in feature space. In this research, we also analyzed the performance of different distance metrics. Besides Euclidean space, the fractional distance and the L1 (i.e., Manhattan), L2 (i.e., Euclidean), and Lp (i.e., Minkowski) distance metrics were included in our modified G-means clustering algorithm and studied.

4.2.4.1 Fractional Distance Metric

The fractional distance metric is claimed to be able to tackle the "curse of dimensionality" problem Aggarwal et al. [2001], as an extension of Euclidean distance, the distance parameter f lies within the range of $(0, 1)$:

$$dist_d^f(x, y) = [\sum_{i=1}^{d} |x^i - y^i|^f]^{1/f}, \tag{4.3}$$

where d is the length of the feature vectors. It has been demonstrated in Aggarwal et al. [2001] that this metric performs better than the common Euclidean and Manhattan distance

metrics (i.e., L2 and L1) in high-dimensional feature spaces, which is the usual case in image classification.

4.2.4.2 Minkowski Distance Metric

Similar to the fractional distance metric, the Minkowski distance Kruskal [1964] is a metric defined in Euclidean space which can be viewed as a generalization form besides the Euclidean and the Manhattan distances:

$$dist_d^p(x,y) = [\sum_{i=1}^{d} |x^i - y^i|^p]^{1/p}, \tag{4.4}$$

where d is the length of the feature vector and p is a parameter with $p \geq 1$. The Minkowski distance is typically used when p equals 1 or 2 (i.e., Manhattan distance, Euclidean distance). When p approaches infinity, the so-called Chebyshev distance is obtained.

4.2.5 Evaluation

For unsupervised clustering analysis, there are two major categories, the internal evaluation which takes account of the compactness of the data, e.g., the spreadness of the data which is usually calculated by the average ratio of the cluster diameter compared to the distance between clusters; and the external evaluation which uses reference labels to calculate the accuracy of the clustering method.

Apart from the above two methods, in this article, an intuitive visualization evaluation is also performed. It can give us a direct impression of the clustering quality as the goal of the clustering method is to group similar patches together and obtain homogeneous clusters.

4.2.5.1 Visual Evaluation

The visual evaluation includes visualizing patches within Gaussian-distributed clusters.

4.2.5.2 Internal Evaluation

Suggested by Kovacs et al. [2006] and Halakidi et al. [2002], hierarchical clustering algorithms usually use the root-mean-squared standard deviation (RMSSTD) index and the R Squared (RS) index for internal evaluation, but they can also be used to evaluate the results of any clustering algorithm. The RMSSTD index Sharma [1996] is the variance of the clusters which measures the homogeneity of the clusters, and is formally defined by Equation 4.5. Since the aim of the clustering process is to identify homogeneous groups, a lower RMSSTD value stands for better clustering Kovacs et al. [2006].

$$RMSSTD = \sqrt{\frac{\sum_{\substack{i=1\ldots nc \\ j=1\ldots d}} \sum_{k=1}^{n_{ij}} (x_k - \overline{x_{ij}})^2}{\sum_{\substack{i=1\ldots nc \\ j=1\ldots d}} (n_{ij} - 1)}}. \tag{4.5}$$

On the contrary, the RS (R Squared) index Kovacs et al. [2006] measures the dissimilarity of clusters as it measures the degree of homogeneity between groups. The values of RS range from 0 to 1 where 0 means there is no difference among the clusters and 1 indicates that there are significant differences among the clusters. The definition is as follows:

$$\begin{aligned} RS &= \frac{SS_t - SS_w}{SS_t}, \text{where} \\ SS_t &= \sum_{j=1}^{d} \sum_{k=1}^{n_j} (x_k - \overline{x_j})^2, \\ SS_w &= \sum_{\substack{i=1\ldots nc \\ j=1\ldots d}} \sum_{k=1}^{n_{ij}} (x_k - \overline{x_{ij}})^2, \end{aligned} \tag{4.6}$$

The parameter nc represents the number of clusters, the parameter d stands for the dimensionality of the feature vector, the parameter $\overline{x_{ij}}$ is the mean of the data in variable j and cluster i, and the parameter n_{ij} describes the number of elements in variable j and cluster i.

4.2.5.3 External Evaluation

In our research, the purpose of clustering is to prepare for subsequent semantic annotation; instead of using traditional standard clustering evaluation methods (e.g., F-measure, confusion matrix), a new external analysis method to measure the homogeneity of clustering results is defined which is able to emphasize the following points: 1. the homogeneity in each cluster, as the more homogeneous one cluster is, the more easily we can annotate the cluster later; 2. the number of clusters; 3. the size of the resulting clusters. The detailed calculation is defined as follows:

- A cluster is considered as homogeneous, when more than 90% of the patches belong to one class, denoted by C_{homo}.
- A patch is considered to be labeled correctly as homogeneous, when it belongs to the major class of the homogeneous cluster C_{homo}, denoted by P_{homo}.
- Count the number of homogeneous patches, denoted by $num(P_{homo})$.
- Calculate the homogeneity percentage of the clustering result, which is defined as the number of homogeneous patches divided by the total number of patches in the dataset: $num(P_{homo})/num(P)$.

4.2.6 Comparative Clustering Methods

In order to prove the method's performance in simplifying the image annotation problem by obtaining homogeneous clusters, we have also made a set of comparative experiments using related traditional unsupervised clustering methods. The following three methods were chosen for the comparison:

- k-means: as the G-means method uses k-means to split the data.
- Gaussian mixture model (GMM): as the cluster splitting criterion is based on a Gaussian test.
- Agglomerative hierarchical clustering: as the method builds a hierarchical structure.

The results are shown below.

4.3 Results

In this section, our prepared datasets, the experimental settings, and the results will be described. Then the clustering results of the proposed methodology will be analyzed, including quantitative evaluations in terms of different distance metrics, together with some visual evaluations which provide detailed insights into the data characteristics.

4.3.1 Datasets

In order to obtain a more thorough analysis with the G-means method, we have prepared three datasets for the evaluation. All of them are semantically annotated, two datasets are cut from SAR images, and one is collected from optical data.

Dataset 1 consists of 15 classes of SAR image patches, with over 100 patches in each class and more than 3000 patches in total. The patches are collected from Spotlight TerraSAR-X radiometrically enhanced Multilook Ground Range detected products, with a pixel spacing of 1.25 m, a resolution about 2.9 m and a size of 160×160 pixels Fritz [2013]. As shown in Fig. 4.4, each class has a relatively large intra-class variance.

Dataset 2 consists of 11 classes of SAR image patches, with 100 patches in each class and 1100 patches in total. The patches are collected from Spotlight TerraSAR-X Single Look Complex products which have been converted to magnitude data, with a resolution of 1 m/pixel and a size of 200×200 pixels. As shown in Fig. 4.5, compared to dataset 1, dataset 2 is very homogeneous, with a relatively small intra-class variance.

Dataset 3 is the UC Merced Land Use Dataset Yang and Newsam [2010], which consists of 21 classes of optical image patches, with 100 patches in each class and 2100 patches in total. The patches are extracted from the USGS National Map Urban Area Imagery collection, with a resolution of 1 foot and a size of 256×256 pixels.

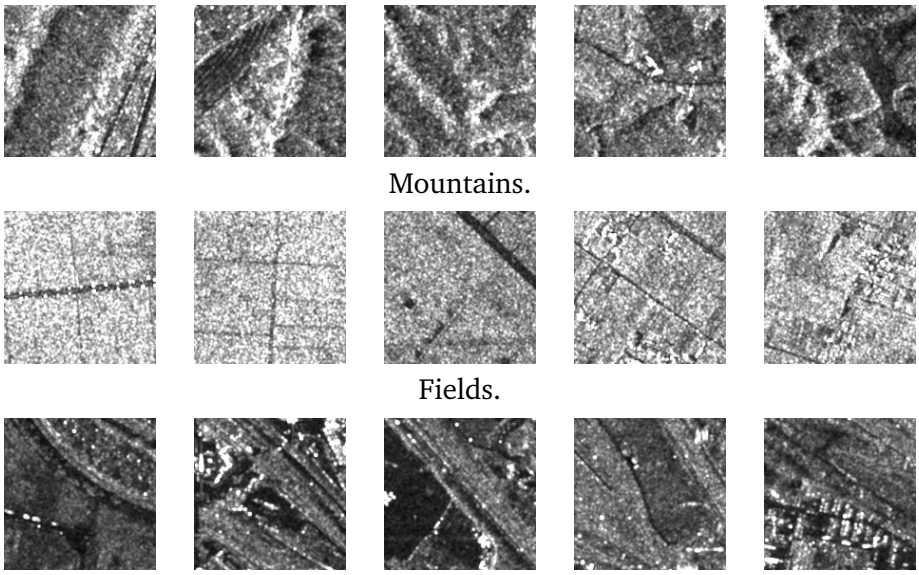
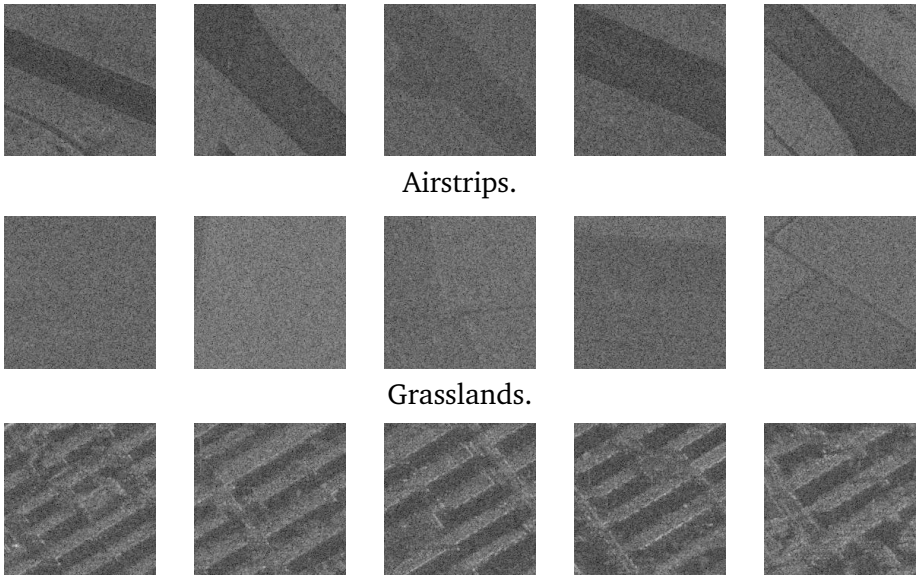


Figure 4.4: Patch examples for three classes of dataset 1.

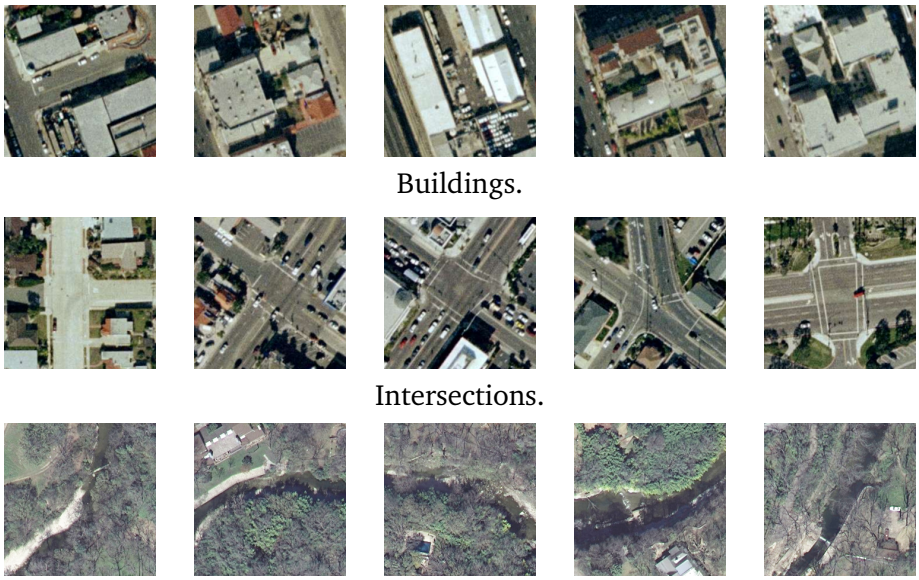


Airstrips.

Grasslands.

Urban areas.

Figure 4.5: Patch examples for three classes of dataset 2.



Buildings.

Intersections.

Rivers.

Figure 4.6: Patch examples for three classes of dataset 3.

4.3.2 Experimental Settings

The image features are extracted using a Gabor filter bank with 4 scales and 6 orientations. The feature values are the mean and variance of the filtered responses of each patch and they form a 48-dimensional feature vector. The choice of Gabor features resulted from comparisons with other feature extraction methods Dumitru and Datcu [2013].

Before we extract feature descriptors, each image patch will be subsampled from its original size of 160×160 to 80×80 pixels. This procedure decorrelates neighboring pixels, reduces the computational effort, and typically increases the recall during image retrieval by 1% Dumitru and Datcu [2013].

Alternatively, the BoW features are constructed by using the raw pixel values as the primitive feature descriptors, as suggested by Cui et al. [2015]. For a better comparison, two methods of building the visual-word vocabulary have been experimented with, namely vector quantization as well as random dictionaries.

As a conventional approach, the vector quantization (VQ) technique clusters the primitive feature descriptors in their feature space into a large number of clusters (e.g., using the k-means clustering algorithm) and encodes each feature descriptor by the index of the cluster to which it belongs to Yang et al. [2007]. Each cluster is conceived as a visual word that represents a specific local pattern shared by the feature descriptors in that cluster.

Recently, random dictionaries have been claimed to yield good classification results in addition to their computational efficiency Yang et al. [2007]. In our case, the required visual words are created by generating a random dictionary which introduces the randomness during feature extraction. However, we will not deal with the effect of randomness in this dissertation, as it is only being used as a specific type of feature descriptor.

4.3.3 Parameter Settings

Table 4.1 lists the detailed parameter settings of our experiments.

Table 4.1: Table of parameter settings.

Parameters	Values	Remarks
Gabor filters	4 scales and 6 orientations	These parameters resulted from comparisons with other feature extraction methods and are also used in the MPEG-7 standard MPEG-7.
Bag-of-Words	Window size 3×3, patch size 5×5, 250 visual words	These parameters are based on their excellent performance published in Cui et al. [2015].
Confidence level	$\alpha = 0.0001$	Critical value = 1.8692.
Cluster size limit	40 patches	The number 40 is chosen based on two reasons: (1) based on the Central Limit Theorem, to obtain a meaningful Gaussian distribution; (2) to obtain suitable cluster sizes.
Distance metrics	0.2 to 5	From 0.2 to 1 with a step size of 0.2 for fractional distance, from 1 to 5 with a step size of 1 for Minkowski distance.

4.3.4 Visual Evaluation

In this section, in order to have a direct visual impression of the clustering performance, one clustering example comprising 40 patches is shown for each dataset. The clustering results are obtained under the condition of a distance metric equal to 1.

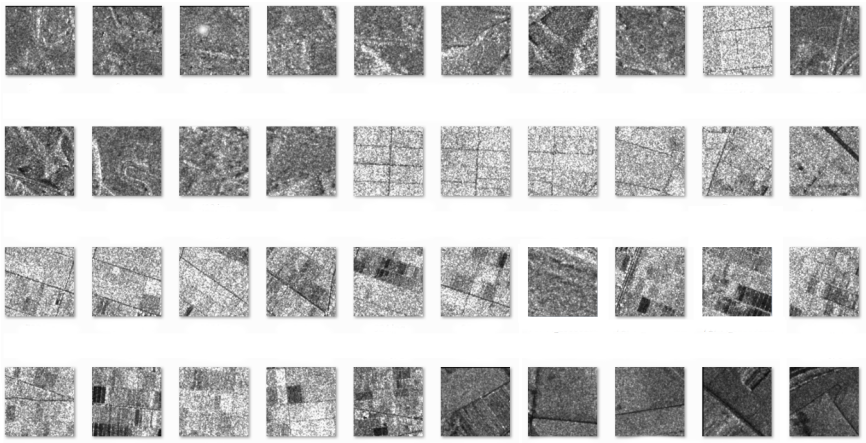


Figure 4.7: Screen shot of patches as an example of a cluster from dataset 1.

Fig. 4.7 shows a screen shot of patches as an example of a cluster from dataset 1. It consists of: 86 cropland patches, 1 highway patch, 12 grass patches, 21 field patches, and 2 mountain patches. Although some are light patches, some are dark patches. The majority of patches refers to vegetation.

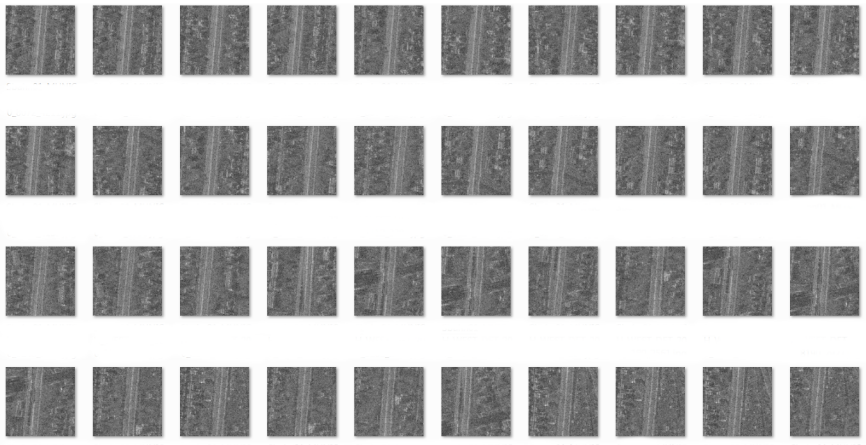


Figure 4.8: Screen shot of patches as an example of a cluster from dataset 2.

Fig. 4.8 shows a screen shot of patches as an example of a cluster from dataset 2. All 61 patches are railway train patches.

Fig. 4.9 shows a screen shot of patches as an example of a cluster from dataset 3. It contains 1 airplane patch, 1 baseball diamond patch, 8 building patches, 19 dense residential patches, 1 golf course patch, 55 harbor patches, 3 intersection patches, 4 medium residential patches, 21 mobile home park patches, 28 parking lot patches, and 1 tennis court patch. The common characteristic of the patches is that most of them contain regular patterns.

Figure 4.9: Visualization of patches as an example of a cluster from dataset 3.

All of the three examples are typical clustering results from each dataset, the visualization results represent the method's ability in arranging similar image patches. Furthermore, as a side product, the results also reflect the individual subjective effect of semantic image annotation, since two patches might be annotated as belonging to one class or to two classes, depending on the level of image details and the annotation level.

4.3.5 Internal Evaluation

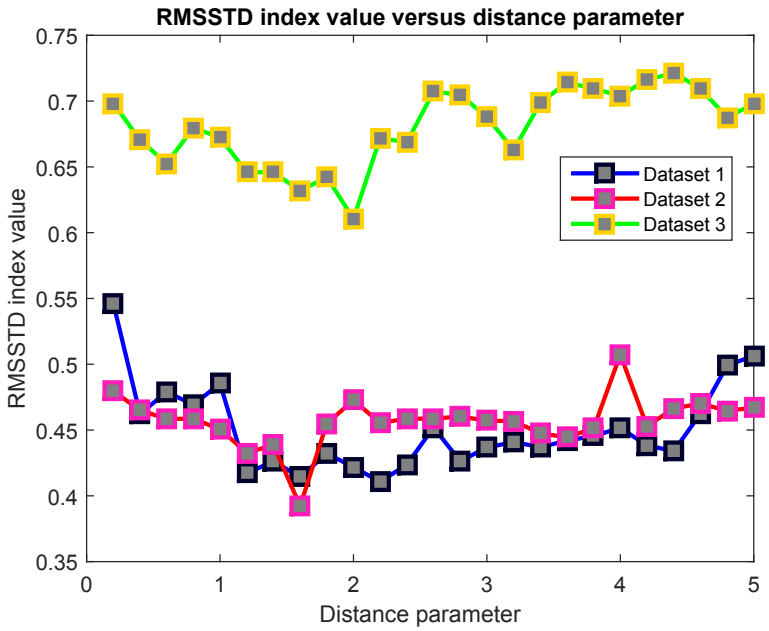


Figure 4.10: RMSSTD index versus distance parameter for three datasets. Different colors show the results for different datasets.

In this section, all of the three datasets are used for comparative experiments. Fig. 4.10 and Fig. 4.11 illustrate the RMSSTD index and the RS index versus distance parameter for three datasets, respectively. The RMSSTD index values first decrease then increase as the distance parameter increases; on the contrary, the RS index values first increase then

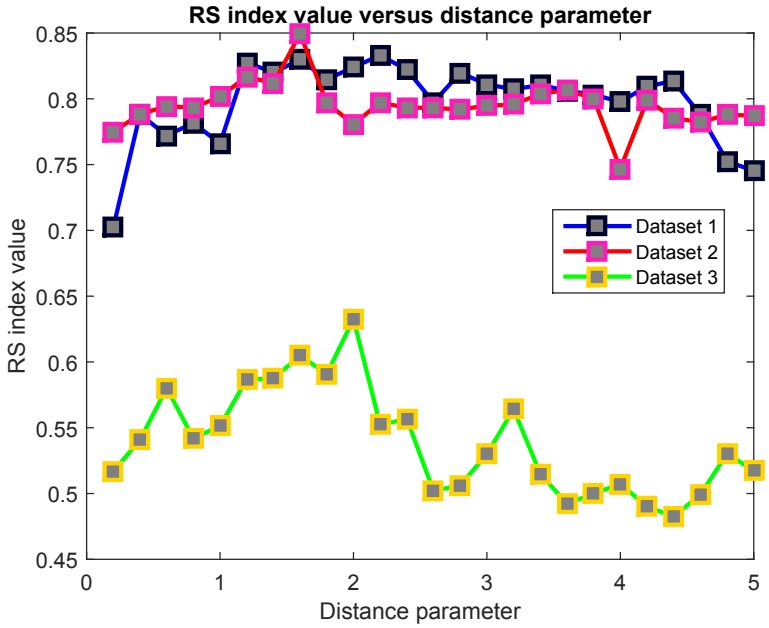


Figure 4.11: RS index versus distance parameter for three datasets. Different colors show the results for different datasets.

decrease as the distance parameter increases. Generally, a distance parameter that ranges from 1.2 to 2 performs best.

For dataset 1, the best distance parameter is 1.2 due to the high feature vector variance. For dataset 2, the best distance parameter is 1.6 due to the lower feature vector variance. For dataset 3, the best distance parameter is 2 because the optical data are characterized by a low noise level. Also, the two SAR datasets result in more homogeneous clusters as their RMSSTD and RS index values are similar, although the obtained clusters in dataset 1 contain more mixed annotations than in dataset 2. There are two possible explanations: Either the feature vector variances are different or the applied feature descriptor performs worse for dataset 1. We prefer the first explanation, as shown in Fig. 4.7 and Fig. 4.8; the patches are more diverse for dataset 1.

Fig. 4.10 and Fig. 4.11 demonstrate that a distance parameter larger than 2 does not improve the performance, so a distance metrics range from 0.2 to 2 has been used for the following experiments.

4.3.6 External Evaluation

In this section, the properties of cluster homogeneity will be discussed. As dataset 1 and dataset 2 are both SAR datasets with big and small within-class invariance labeling, it is reasonable to concentrate on a comparison between dataset 1 and dataset 2. We show experimental results and an analysis of the proposed hierarchical patch clustering method for dataset 1 and dataset 2. In order to get a more thorough understanding, we analyze the results from different perspectives.

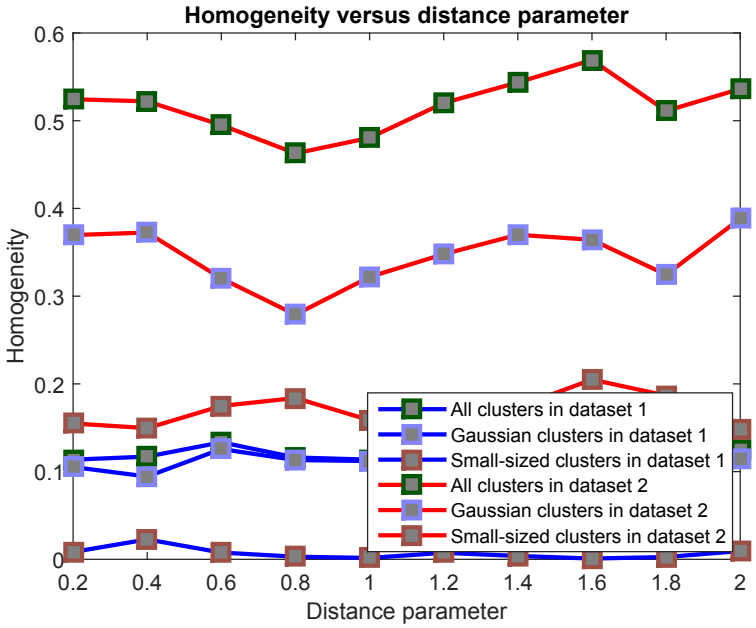


Figure 4.12: Homogeneity of the whole dataset. The blue line represents results for dataset 1. The red line represents results for dataset 2. Here we consider two types of clusters: Gaussian clusters and small-sized clusters, and all clusters.

4.3.6.1 Analysis of Absolute Homogeneity

Fig. 4.12 depicts the homogeneity of the whole dataset versus different distance parameters. We observe:

- For dataset 1 (shown in red) with large intra-class variances, around 10 percent of the patches are grouped in homogeneous clusters; for dataset 2 (shown in green) with small intra-class variances, around 50 percent of the patches are grouped in homogeneous clusters.
- Gaussian clusters comprise the majority of the homogeneous clusters, as the green curves and the corresponding red curves have similar shapes.
- Small-size clusters also contain some homogeneous clusters, shown in blue. In particular, dataset 2 contains more homogeneous small-size clusters.
- The best distance parameter for dataset 1 is 1.8; the best distance parameter for dataset 2 is 1.6.

Based on these observations, the Gaussian-test-based hierarchical patch clustering method is able to group visually homogeneous image patches (e.g., dataset 2).

4.3.6.2 Analysis of Relative Homogeneity

Fig. 4.13 depicts the homogeneity percentage of each cluster type (Gaussian clusters and small-sized clusters) versus different distance parameters:

- Gaussian clusters have a similar homogeneity percentage to overall clusters, as the green curves and the corresponding red curves have similar shapes.

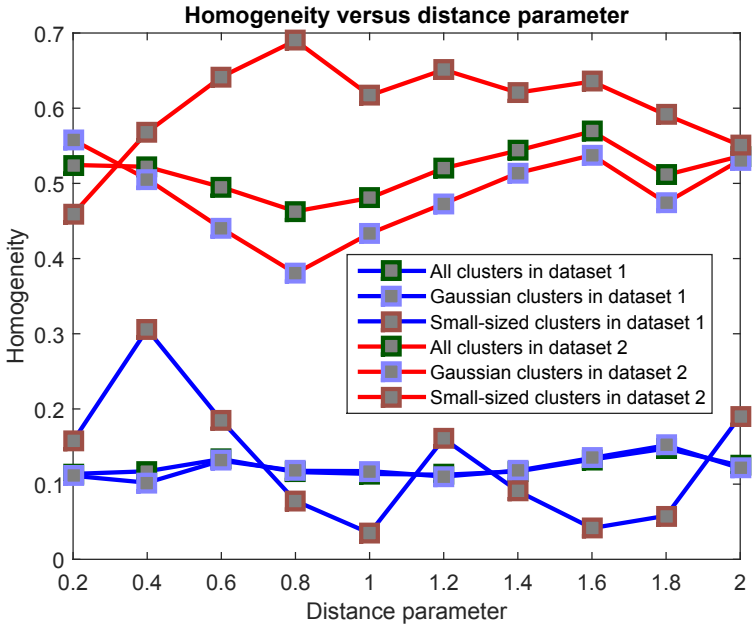


Figure 4.13: Homogeneity percentage of different cluster types. The blue line represents results for dataset 1. The red line represents results for dataset 2. Here we consider two types of clusters: Gaussian clusters and small-sized clusters, and all clusters.

- For dataset 2 (shown in green), the small-size clusters are more homogeneous, as the green line with "+" symbols has higher values than the green line with "o" symbols.
- The best distance parameter for dataset 1 is 1.8; the best distance parameter for dataset 2 is 1.6.

As a consequence, we have to set a minimum cluster size threshold during the clustering procedure, since compared to Gaussian clusters, the small-size clusters are already homogeneous.

4.3.6.3 Analysis of Cluster Numbers

Fig. 4.14 shows the number of clusters:

- For dataset 1 (shown in red), the number of clusters changes drastically and is unstable.
- For dataset 2 (shown in green), the number of clusters does not change much and is relatively stable. The number of clusters equals about twice the actual number of classes.

The different number of clusters for the different datasets is due to the variance of the image patches.

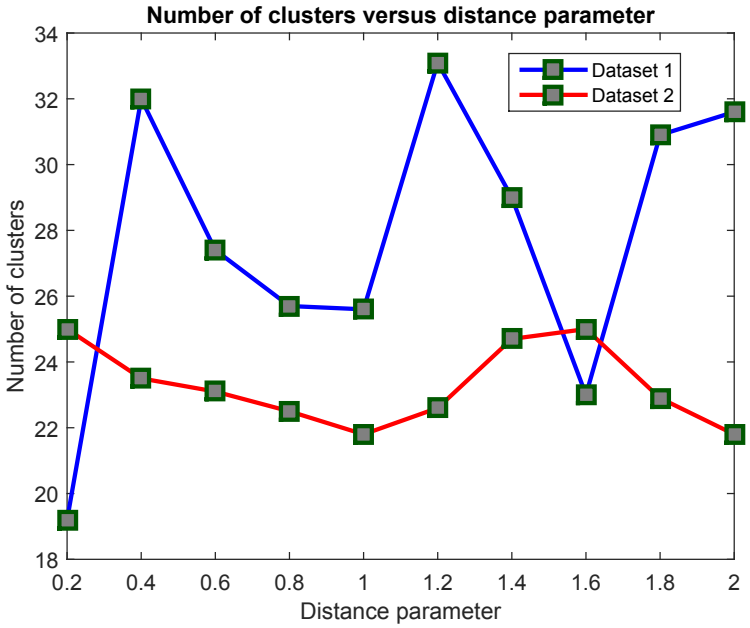


Figure 4.14: Number of clusters for dataset 1 and 2. Different colors represent the results for different datasets.

4.3.7 Comparative Experiments

All of the comparative experiments were conducted by applying a Matlab environment.

4.3.7.1 Different Clustering Methods

For the previously mentioned different clustering methods, we used all three datasets.

The number of clusters for k-means, GMM, and agglomerative hierarchical clustering methods is set according to the number of clusters obtained by the G-means method; for the sake of simplicity, the distance metric is set to Euclidean.

Fig. 4.15 and Fig. 4.16 demonstrate the RMSSTD and the RS index values for all three datasets when using different clustering methods: G-means, k-means, and agglomerative hierarchical clustering.

The GMM algorithm did not converge correctly due to the fitting of too many parameters. Obviously, the agglomerative hierarchical clustering method yields the worst result. Not surprisingly, the classic k-means performs somewhat better than G-means, as k-means groups the feature vectors in a global way. However, due to the Gaussian-test-based splitting method, G-means can build a hierarchical tree structure for clusters, and there is no need to preset the number of clusters, which is an obvious limitation to the application of k-means.

4.3.7.2 Different Features

Because dataset 3 is a popular publicly available optical dataset which has been used more frequently than the other two datasets; all of our comparative experiments using different features were made by using dataset 3.

Fig. 4.17, Fig. 4.18 and Fig. 4.19 depict the RMSSTD index value, the RS index value, and the number of clusters for dataset 3 when applying three different feature extractors: Gabor filtering, BoW based on clustering, and BoW based on a random dictionary.

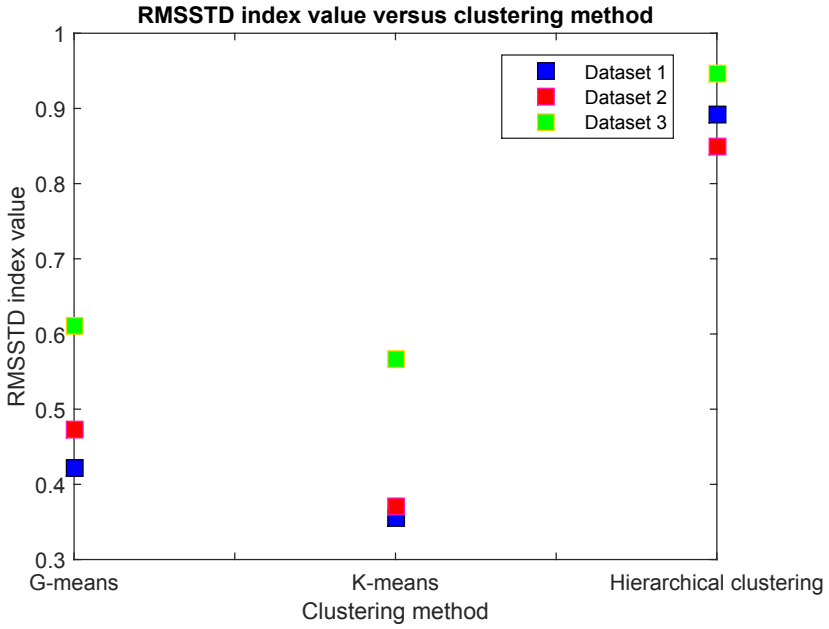


Figure 4.15: RMSSTD index versus different clustering methods for three datasets.

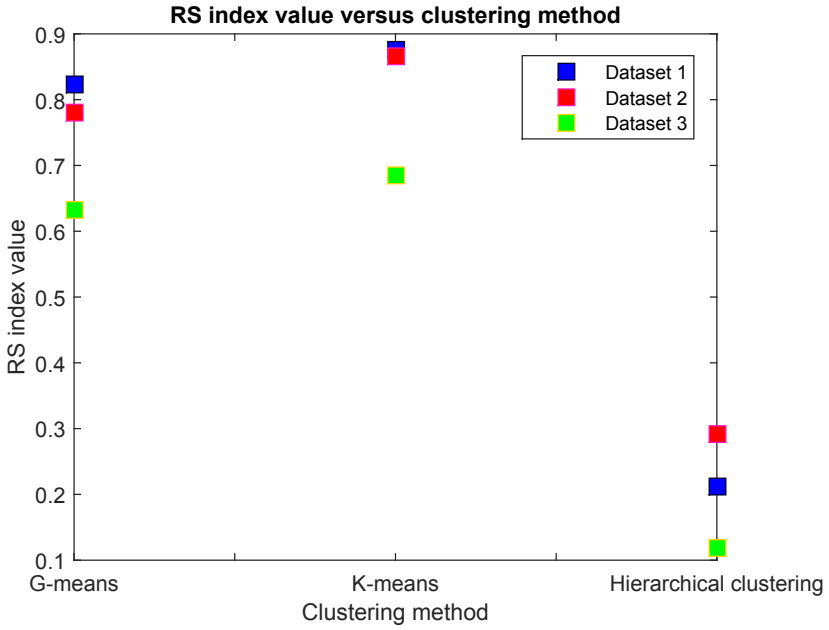


Figure 4.16: RS index versus different clustering methods for three datasets.

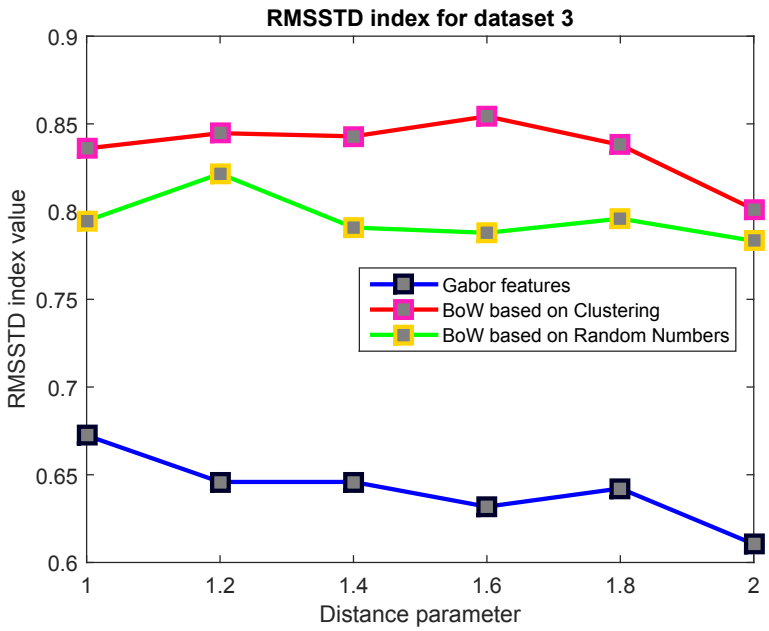


Figure 4.17: RMSSTD index versus distance parameter for dataset 3. Different colors show the results for different features.

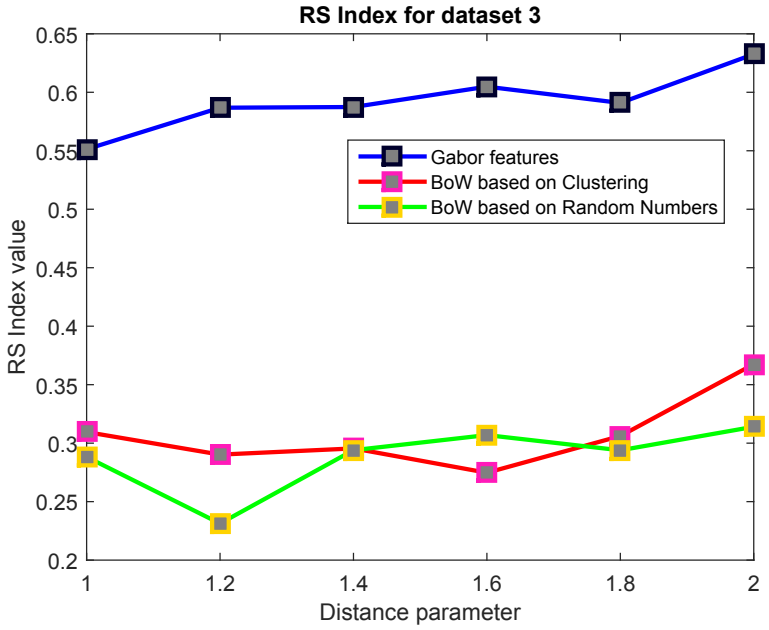


Figure 4.18: RS index versus distance parameter for dataset 3. Different colors show the results for different features.

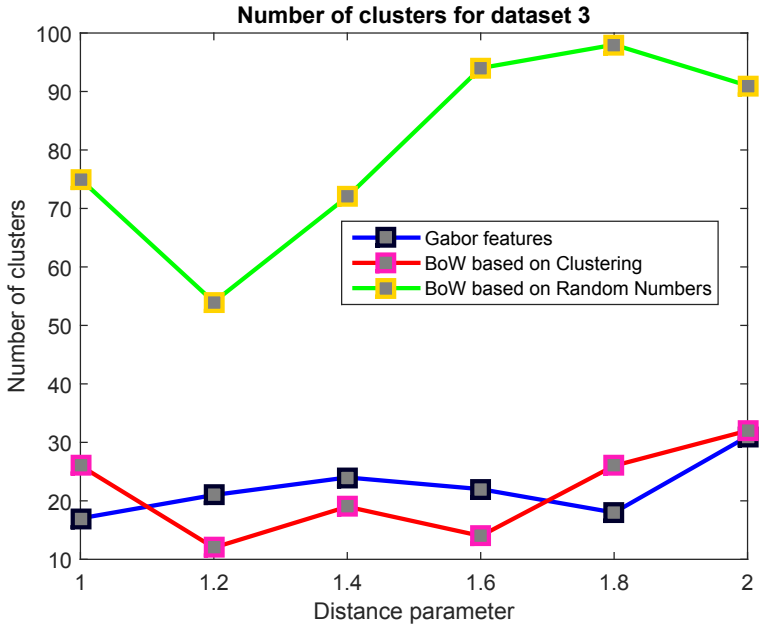


Figure 4.19: Cluster numbers versus distance parameter for dataset 3. Different colors show the results for different features.

The best distance metric for dataset 3 is the Euclidean distance. Gabor features turn out to give the best performance, while BoW features with a random dictionary yield better results for the RMSSTD index and slightly inferior results for the RS index compared to BoW features with clustering. However, as shown in Fig. 4.19, BoW features with a random dictionary result in too many clusters, which is not practical as we prefer a lower number of clusters as well as cluster homogeneity. Hence, clustering based on BoW features is more practical than BoW features with a random dictionary.

4.4 Conclusions

In this chapter, we presented a hierarchical patch clustering method for high-resolution remote sensing images, with the purpose of obtaining homogeneous clusters. A Gaussian goodness test (Anderson-Darling test) is used to split feature vectors into homogeneous clusters; in addition, we define a threshold for the minimum cluster size. Thus, Gaussian clusters and small-sized clusters are obtained during the clustering procedure. We compared various distance metrics in order to obtain robust clustering. The results show that the Gaussian-test-based hierarchical patch clustering method is able to obtain homogeneous clusters, while Gabor texture features perform better than the BoW features. In addition, it turns out that a distance parameter ranging from 1.2 to 2 performs best. Also indicated by Yao et al. [2016b], our modified G-means algorithm is faster than the original algorithm.

The clustering method was applied to two space-borne SAR datasets and one airborne optical dataset. The clustering results were evaluated in three ways: visually, and by internal and external index calculations. The label annotations of all three datasets were used to assist in the evaluations. Visualizations of typical clusters give us an intuitive feeling of how homogeneous feature vector clusters look like. Based on the internal and external evaluations of the experimental results, we compared relations among the clustering results,

under different distance metrics, and we calculated the homogeneity percentage against the whole dataset or for each category (Gaussian clusters and small-sized clusters) with the annotated reference data. It turns out that our presented method is able to obtain homogeneous data, the best distance parameter lies within the range from 1.2 to 2.

We also compared the results using different clustering methods and different feature descriptors. The results show that G-means can obtain relatively homogeneous clusters, although compared to k-means, the cluster homogeneity is slightly degraded because of the repeated dichotomous cluster splitting. Moreover, Gabor features gave the best performance for the optical dataset 3.

Chapter 5

Semi-supervised Semantic Image Patch Annotation

> The journey of a thousand miles begins with one step.
>
> Lao Tzu

In this chapter, we propose a semi-automated hierarchical clustering and classification framework for SAR image annotation. The implementation of the framework allows the classification and annotation of image data ranging from single scenes up to large satellite data archives. As an updated version of our framework Fig. 4.1 of Chapter 4, the new framework comprises three stages: Firstly, each image is cut into patches and each patch is transformed into a texture feature vector. Secondly, similar feature vectors are grouped into clusters, where the number of clusters is determined by repeated cluster splitting to optimize their Gaussianity. Finally, the most appropriate class (i.e., a semantic label) is assigned to each image patch. This is accomplished by semi-supervised learning. For the testing and validation of our implemented framework, a concept for a two-level hierarchical semantic image content annotation was designed and applied to a manually annotated reference data set consisting of various TerraSAR-X image patches with meter-scale resolution. Here, the upper level contains general classes, while the lower level provides more detailed sub-classes for each parent class. For a quantitative and visual evaluation of the proposed framework, we compared the relationships between the clustering results, the semi-supervised classification results, and the two-level annotations. It turns out that this proposed method is able to obtain reliable results for the upper level (i.e., general class) semantic classes; however, due to the too many detailed sub-classes versus the few instances of each sub-class, the proposed method generates inferior results for the lower level. The most important contributions of this chapter are the integration of modified Gaussian-means and modified cluster-then-label algorithms, for the purpose of large-scale SAR image annotation, as well as the measurement of the clustering and classification performances of various distance metrics.

5.1 Methodology

We follow the work of Chapter 4 by presenting a semi-supervised Gaussianity-based hierarchical clustering method for remote sensing image annotation. Specifically, our method intends to answer the following two questions: How do we explore information

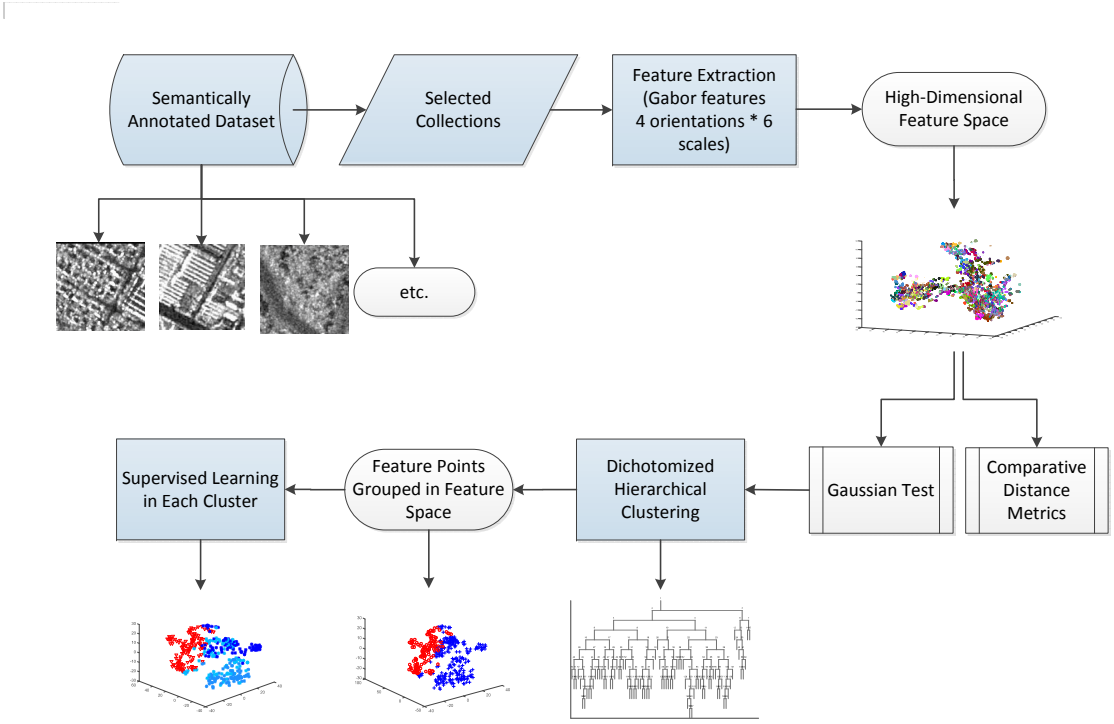


Figure 5.1: Proposed data classification and analysis scheme for image patches.

in the feature vector space, and how do we link our semi-supervised results with an already annotated reference dataset?

Hence, one goal of this research is to study the extracted features in their high-dimensional feature space, i.e., how do they behave (including their computational effort) and what is a good distance metric to describe the pair-wise relationships between the feature points. Therefore, the definition of distance metrics plays an important role in exploring features in a given feature space (that, as a rule, has more than three dimensions Aggarwal et al. [2001]).

The second goal is to find the relationships between the clustering results, the semi-supervised classification results, and the two-level annotations. When we group the feature points based on their similarity, we are able to generate homogeneous clusters. Then a cluster-then-label semi-supervised learning method is performed Zhu and Goldberg [2009].

We start with the preparation and application of reference data that we need for the evaluation and validation of our method. This includes the performance analysis of clustering as well as classification, where we apply quantitative tests. The hierarchical clustering method which have been explained in detail in Chapter 4 will be used within the notion of semi-supervised classification. In the end, we compare the clustering results of the above-mentioned hierarchical clustering method and the cluster-then-label semi-supervised results with the reference data annotations, in order to analyze the correspondences between the learning results and manually annotated reference data. The results are evaluated by visual and quantitative analyses.

Fig. 5.1 illustrates the whole concept of the proposed data classification and analysis scheme. We start from a semantically annotated dataset of image patches, from which we select a number of collections. Then we perform feature extraction for each image patch. This results in a high-dimensional feature space. A clustering algorithm with a Gaussian test

and distance metric are used to generate a dichotomized hierarchical cluster tree structure. After applying supervised learning in each cluster, predefined semantic labels are assigned to the feature vector points.

In this chapter, the range of distance parameters is set to [0.2, 2] with a step size of 0.2, and to [3, 13] with a step size of 1. In case of the original G-means algorithm, the range of distance parameters is modified to [0.8, 2] with a step size of 0.2, and to [3, 12] with a step size of 1, due to the long computational time.

5.1.1 Creation of a Reference Dataset

In this sub-section, we present the methodology used to classify and semantically annotate our TerraSAR-X images being used as a reference dataset. Our general approach during the annotation scheme was to tile each TerraSAR-X image into a number of non-overlapping patches, extract features, then classify and annotate each individual patch. The main steps of the processing chain were:

- Select TerraSAR-X images and tile them into patches with a size of 160×160 pixels. Subsample each patch into a smaller patch of 80×80 pixels to generate decorrelated pixels Dumitru and Datcu [2013].
- Extract a 48-dimensional feature vector from each patch using Gabor filters with 4 scales and 6 orientations (take the mean and variance of the patch coefficients) Dumitru and Datcu [2013].
- Classify the feature vectors into classes using a Support Vector Machine (SVM) with relevance feedback Cui [2014]. Each patch is assigned to a single class based on the dominant content of the patch.
- Annotate each class by giving an appropriate semantic meaning to each class Dumitru et al. [2014]. Google Earth is used as ground truth for visual support.

The annotation chain was semi-automated. The first two steps were automated, while the last two steps required manual interaction. For classification, an operator had to rank the given positive and negative examples and grouped them into classes of relevance; for annotation, the operator selected a proper semantic label for each class from a list of available labels. We defined a nomenclature adapted to TerraSAR-X images with a two-level hierarchical scheme that consists of a total of 150 semantic classes MPEG-7. The upper level semantic annotation contained 8 general classes (settlements, industrial production areas, military facilities, transport, agriculture, natural vegetation, bare ground, and water bodies) which were later split into lower level detailed sub-classes. For example, agriculture was split into the following sub-classes: cropland, stubble, bare land, ploughed agricultural land, rice paddies, pasture, plantations and vegetables, greenhouses, and vineyards. Fig. 5.2 gives some examples of the annotated classes.

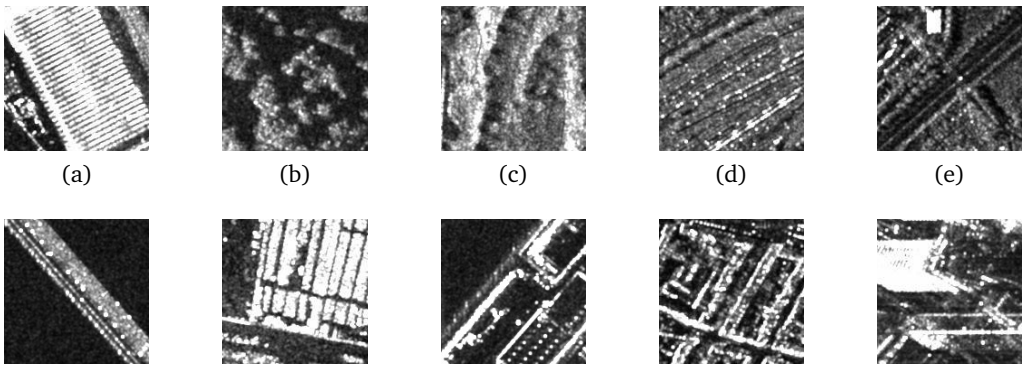


Figure 5.2: Examples of annotated classes. (a) Industrial Buildings (b) Mixed Forest (c) Mountains (d) Railways (e) Roads (f) Bridges (g) Depots (h) Harbor Infrastructure (i) High-Density Residential Areas (j) Skyscrapers.

5.1.2 Semi-supervised Learning

5.1.2.1 Cluster-then-Label

With already homogeneously clustered data points on hand, we follow a cluster-then-label procedure within each cluster which then makes the whole procedure a semi-supervised classification Zhu and Goldberg [2009].

To this end, the original cluster-then-label algorithm is modified to accelerate the image annotation. After the clusters are obtained, we label the training data within each cluster, which guarantees there is no cluster without training labels. The algorithm 2 describes the modified cluster-then-label algorithm.

Data: Labeled patches (x_1, y_1),...,(x_l, y_l), unlabeled patches x_{l+1},...,x_{l+u}, a clustering algorithm A, and a supervised learning algorithm L
Result: Labels for unlabeled patches y_{l+1},...,y_{l+u}
Use A to cluster x_l,...,x_l, x_{l+1},...,x_{l+u}, obtain clusters c_1,...,c_n.
for *each cluster* $c_i = c_1 : c_n$ **do**
  Let S_{c_i} be the labeled patches in the cluster c_i;
  Learn a supervised predictor from S_{c_i}: $f_{S_{c_i}} = L(S_{c_i})$;
  Apply $f_{S_{c_i}}$ to all unlabeled patches within the cluster c_i.
end

Algorithm 2: Modified cluster-then-label semi-supervised algorithm.

5.1.2.2 Supervised Learning Within Clusters

For our studies, three algorithms are chosen and evaluated to perform supervised learning within clusters: an SVM and a k Nearest Neighbor (KNN), as well as a Naive Bayes Nearest Neighbor (NBNN) algorithm. We assume that the readers are already familiar with the SVM and KNN algorithms that rely on the image-to-image distance. Here, we explain the NBNN algorithm which obtains classifications from the perspective of image-to-class distances.

The NBNN image classifier is formalized in Algorithm 3, which is a highly accurate

approximation of the optimal maximum a posteriori Naive-Bayes image classifier. A detailed theoretical derivation can be found in Boiman et al. [2008].

Data: Feature descriptor types f_1,...,f_n of an image patch P.
Result: Class label $\tilde{C}_P$.
for *every* f_i **do**
  for *every defined* C_j **do**
    Compute the NN of f_i in class C_j: $NN_{C_j}(f_i)$.
  end
end
$\tilde{C}_P = \arg\min_C \sum_{i=1}^{n} \|f_i - NN_C(f_i)\|^2$.

Algorithm 3: The NBNN algorithm.

5.1.3 K-Medoids Algorithm Implementation

Although k-means is the most popular clustering algorithm due to its simplicity and efficiency Hastie et al. [2009], it has shortcomings as well, such as:

- No recommendation on the initial location of cluster centroids;
- No criterion for selecting the number of clusters Amorim and Mirkin [2012].

With the alternative distance metrics in place of the traditional Euclidean distance metric, k-means turns into k-medoids. Thus, the cluster center is no longer the mean of the data points and we have to find the cluster center whose components are minimizers of the summed distances:

$$dist(C_k) = \sum_{i \in C_k} |y_i - c_k|^p. \tag{5.1}$$

In order to calculate the corresponding Minkowski center, a steepest descent algorithm is proposed in Amorim and Mirkin [2012], and is claimed to converge much faster than a nature-inspired evolutionary method. Since the Minkowski distance is very similar to a fractional distance (with $p >= 1$ or $0 < p < 1$), we extend the steepest descent algorithm to the entire real-valued parameter space (i.e., to $p > 0$).

5.1.4 Evaluation

Several tests are performed to evaluate the behavior of the proposed clustering method. We use quantitative measurements like internal criteria (e.g., the Dunn index) and external evaluations (i.e. precision/recall, F-score, etc.); we also make visual evaluations by analyzing feature space plots and the patches of selected cluster centroids.

5.1.4.1 Quantitative Evaluation

Since the purpose of clustering is for later semantic annotation, instead of using the traditional standard clustering evaluation methods, we need to define evaluation methods which are able to emphasize the following points: the homogeneity in each cluster, as the more homogeneous a cluster is, the easier it becomes that we can annotate the cluster later; the number of clusters; and the size of clusters.

The clustering results can be evaluated in two ways:

- Internal evaluation: the clustering results are evaluated without reference data. The Dunn index that identifies dense and well-separated clusters is used in this chapter. It is defined as the ratio between the minimal inter-cluster distance to the maximal intra-cluster distance Dunn [1973]:

$$D = \min_{1\leq i\leq n} \min_{1\leq j\leq n, i\neq j} \frac{d(i,j)}{\max_{1\leq k\leq n} d'(k)}. \tag{5.2}$$

- External evaluation: the clustering results are evaluated based on given reference data. In our case, the widely used confusion matrix, overall accuracy, and F-score are used to evaluate the classification accuracy. In addition, the overall accuracy and F-scores are also calculated for the original G-means algorithm. Equation 5.3 defines precision and recall; Equation 5.4 shows the definition of F-score Sokolova and Lapalme [2009]:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \tag{5.3}$$

$$F_\beta = \frac{((\beta)^2 + 1) \times P \times R}{(\beta)^2 \times P + R} \tag{5.4}$$

Here, P is precision, R is recall. TP means true positive error, FP means false positive error, FN means false negative error. β is a positive real number which enables F_β to "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision" Van Rijsbergen [1979].

Because each class generates a separate F-score value, a micro F-score (obtained by weighted averaging the class-specific F-scores) is used in this chapter.

Finally, we present in this chapter for each distance metric its individual computational time and the resulting number of clusters.

5.1.4.2 Visual Evaluation

Visual evaluation is used to get an intuitive idea about the clusters in feature space as it is very difficult to understand how features spread out in high-dimensional spaces Aggarwal et al. [2001]. Therefore, we have to apply a dimensionality reduction method to facilitate the understanding of the clusters. The following list describes our 5 visual evaluation methods.

- Tree structure: The hierarchical clustering structure can be represented as a dichotomy tree, which explains the splitting of the clusters.

- Feature space visualization: In order to visually evaluate the clustering results, the t-distributed stochastic neighbor embedding (t-SNE) algorithm can be used for dimensionality reduction; it retains the local data structure as well as the main global structure of the feature space Van der Maaten and Hinton [2008]. For the reference data, the clusters are labeled using the available annotation classes; for the obtained clustering results, each cluster is shown in a different color.

- Cluster centroid patches: For analyzing the compactness of clusters, the closest patch, the median distance patch, and the farthest patch from the cluster centroid are chosen to show the patch-feature relationships of the clusters.

- Cluster homogeneity: The cluster homogeneity is presented via pie charts, using the general and detailed level reference data, to provide the quantitative class percentages within each cluster.

5.2 Results

In this section, the selection of the image data collections, the subsampling of the pixels as well as the detailed parameter settings will be described. Then the clustering results of the proposed methodology will be analyzed, including a quantitative evaluation to find the optimal distance metric, and also some visual evaluations which provide detailed insights into the data characteristics.

5.2.1 Image Data Selection and Subsampling

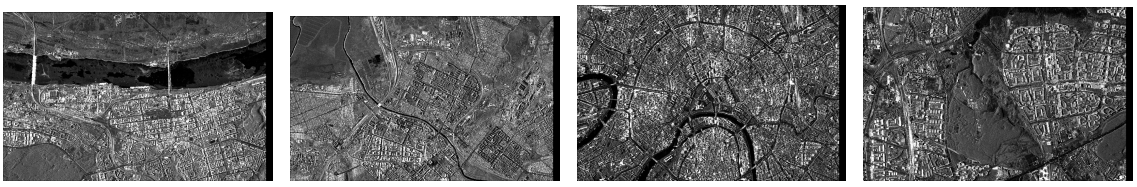
Prior to any data analysis, we had to select appropriate image data collections and to pre-process the data.

5.2.1.1 Data Selection

As typical examples, we selected four image collections from our database which cover different areas of the world from North America, Africa, and Europe. Table 5.1 contains detailed information about the selected collections. Figures 5.3 and 5.6 show some quick-look examples of images taken from the selected collections.

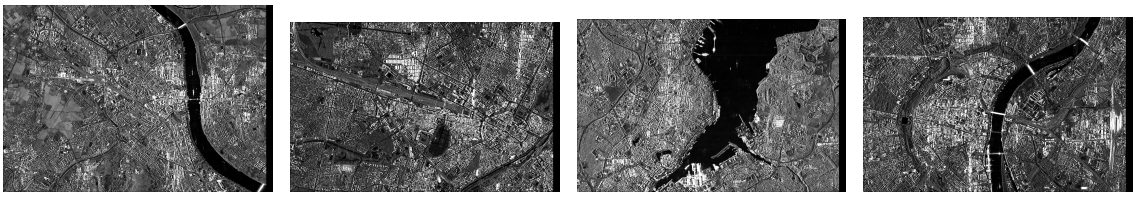
5.2.1.2 Data Pre-Processing

Before we extracted feature descriptors, each image patch was subsampled from its original size of 160×160 to 80×80 pixels. This procedure decorrelated neighboring pixels, reduced the computational effort, and typically increased the recall by 1% Dumitru and Datcu [2013].



(a) Perm (b) Tula (c) Center of Moscow (d) North Moscow

Figure 5.3: Examples of scenes from Russia Fritz [2013], Dumitru et al. [2014].

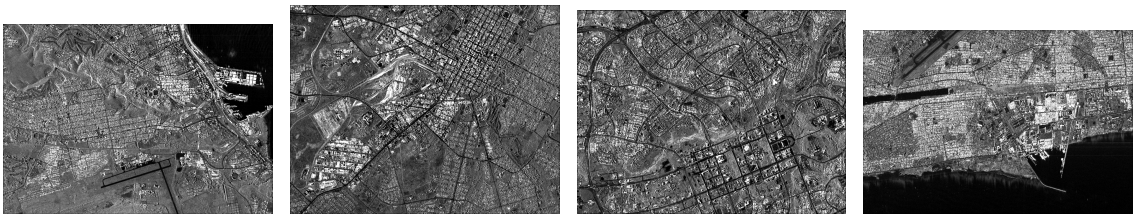


(a) Bonn (b) Munich (c) Kiel (d) Cologne

Figure 5.4: Examples of scenes from German speaking countries. Here the selected examples are from Germany Fritz [2013], Dumitru et al. [2014].

(a) Ciudad Juarez, Mexico (b) Tucson, USA (c) San Diego North, USA (d) Sun Lakes, USA

Figure 5.5: Examples of scenes from North America Fritz [2013], Dumitru et al. [2014].



(a) Port Elizabeth, South Africa (b) Bulawayo, Zimbabwe (c) Abuja, Nigeria (d) Lome, Togo

Figure 5.6: Examples of scenes from Africa Fritz [2013], Dumitru et al. [2014].

Table 5.1: Image data collection parameters.

Dataset	Continent	Country	Resolution	No. of Scenes	No. of Patches	No. of General Classes	No. of Detailed Classes
Collection 02	Asia	Russia	2.9 m	7	7187	8	39
Collection 03	Europe	Germany, Switzerland	2.9 m	7	7176	8	41
Collection 17	North America	United States, Mexico	2.9 m	9	6975	8	30
Collection 27	Africa	South Africa, Zimbabwe, Nigeria, Togo	2.9 m	4	3536	8	22

5.2.2 Parameter Settings

Table 5.2 lists the detailed parameter settings of our experiment.

Table 5.2: Table of parameter settings.

Parameter	Value	Remark
Patch size	160×160 pixels	Depends on the imaging parameters of the TerraSAR-X data acquisitions (e.g., resolution and pixel spacing) Dumitru and Datcu [2013].
Gabor filters	4 scales and 6 orientations	These parameters resulted from comparisons with other feature extraction methods and are also used in the MPEG-7 standard MPEG-7.
Confidence level in Anderson-Darling test	$\alpha = 0.0001$	Critical value = 1.8692.
Cluster size limit	$\geq$ 40 patches	The number 40 is chosen based on two reasons: (1) based on the Central Limit Theorem, to obtain a meaningful Gaussian distribution; (2) to obtain suitable cluster sizes (see Table 5.1).
Supervision percentage	30 percent	Normally 30 percent is sufficient for experiments.

5.2.3 Quantitative Evaluations

We started our performance evaluation experiment with image data collection 17 and our detailed level reference data. In this case, for urban areas, the given images contained high-density residential areas, medium-density residential areas, low-density residential areas, skyscrapers, etc. Dumitru et al. [2014]). In total, collection 17 contained 30 semantic classes.

From the experimental results, we presented not only the internal and external evaluations, but also the computational time and the resulting number of clusters. In order to explain the annotation results and to get a better understanding, we conducted a set of comparative tests to prove the effectiveness and limits of our proposed method. Finally, based on the quantitative results, the optimal distance metric is chosen for later experiments.

5.2.3.1 Internal Evaluation

Dunn's Index yielded 0.0047 for all distance metrics. Since it was defined as the ratio of the minimal inter-cluster distance to the maximal intra-cluster distance, the constant result reflects that the obtained clustering structures are similar, even when applying different distance metrics.

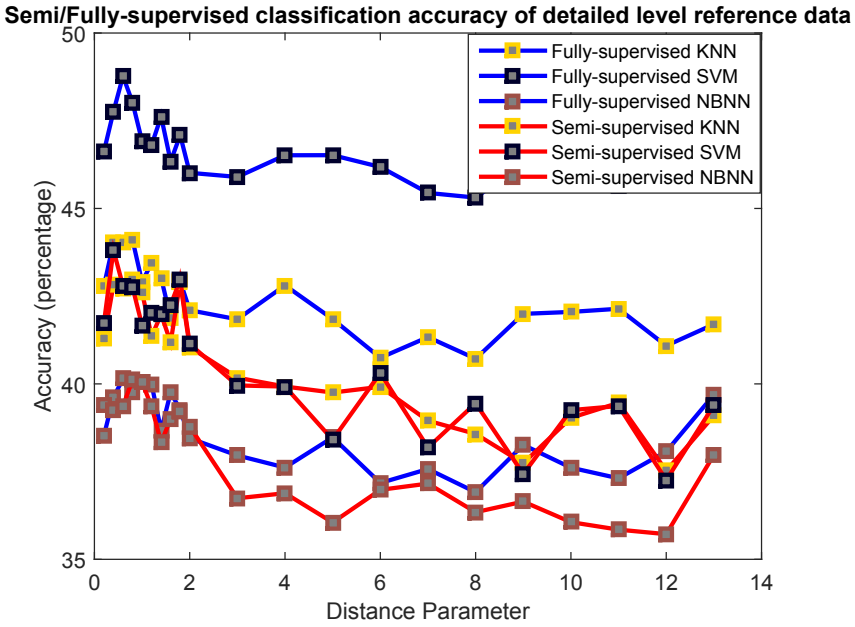


Figure 5.7: Accuracy of detailed level classifications for the modified G-means algorithm for the data collection 17. The blue line shows results for fully-supervised classification. The red line shows results for semi-supervised classification. Different colors of squares represent different algorithms.

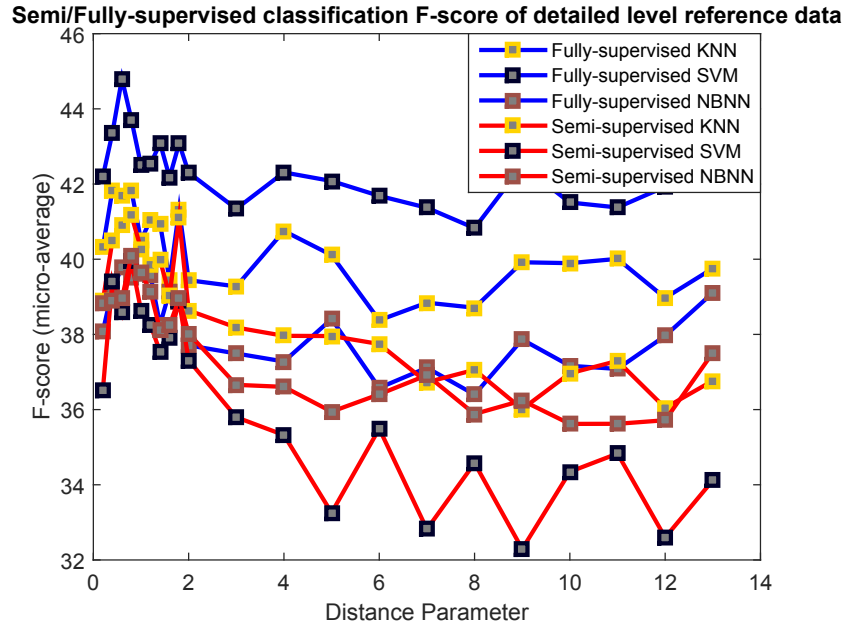


Figure 5.8: Micro-average F-score of detailed level classifications for the modified G-means algorithm for the data collection 17. The blue line shows results for fully-supervised classification. The red line shows results for semi-supervised classification. Different colors of squares represent different algorithms.

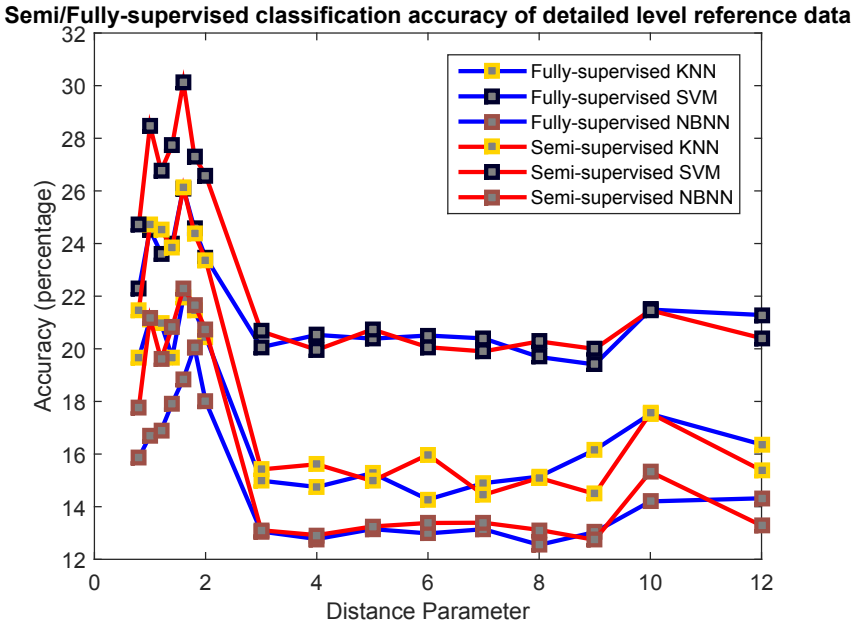


Figure 5.9: Accuracy of detailed level classifications for the original G-means algorithm for the data collection 17. The blue line shows results for fully-supervised classification. The red line shows results for semi-supervised classification. Different colors of squares represent different algorithms.

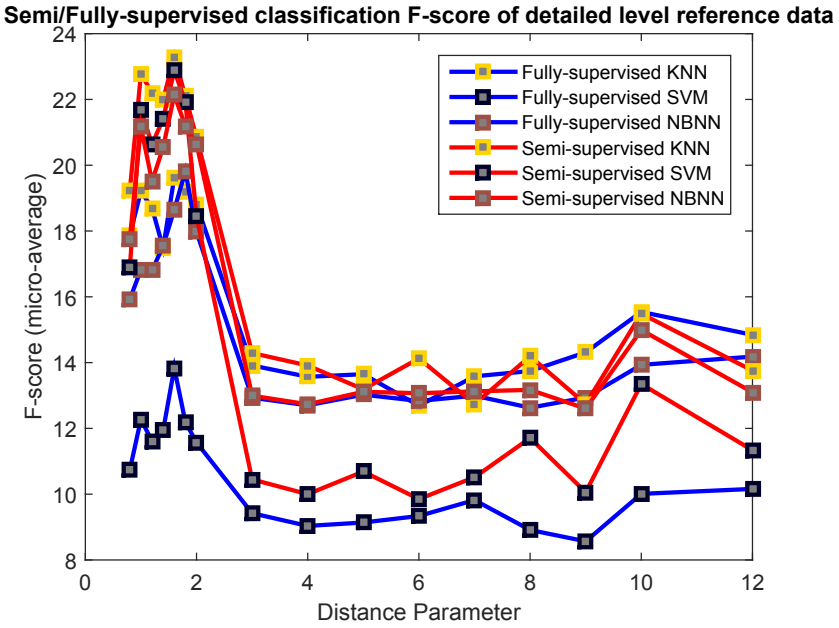


Figure 5.10: Micro-average F-score of detailed level classifications for the original G-means algorithm for the data collection 17. The blue line shows results for fully-supervised classification. The red line shows results for semi-supervised classification. Different colors of squares represent different algorithms.

5.2.3.2 External Evaluation

For the data collection 17, Fig. 5.7 and Fig. 5.9 depict the overall classification accuracy of our detailed level reference data for each distance metric obtained with different supervised and semi-supervised classifiers (KNN, SVM, and NBNN), regarding the original and modified G-means algorithms.

In case of the modified G-means algorithm, as shown by Fig. 5.7, we observe that:

- When $p < 2$, all (supervised and semi-supervised) classification accuracies are decreasing versus distance; when $p > 2$, the accuracy does not change much.
- When we compare the semi-supervised and supervised classifications, the supervised classifications are more accurate.

In case of the original G-means algorithm, as shown by Fig. 5.9, we observe that:

- When $p < 2$, all (supervised and semi-supervised) classification accuracies are increasing versus distance; when $2 \leq p < 3$, all (supervised and semi-supervised) classification accuracies are decreasing versus distance; when $p \geq 3$, the accuracy does not change much.
- When we compare the semi-supervised and supervised classifications, the semi-supervised classifications are more accurate.

Similarly, Fig. 5.8 and Fig. 5.10 show the overall classification F-score of our detailed level reference data for each distance metric obtained with different supervised and semi-supervised classifiers (KNN, SVM, and NBNN), for the original and modified G-means algorithms, respectively. For the F-score plots, the remarks made to Fig. 5.7 and Fig. 5.9 apply, too.

Overall, the accuracy and F-score values of the original G-means algorithm are lower than the values of the modified G-means algorithm.

Fig. 5.3 shows the confusion matrix for a selected number of classes with a sufficient number of patch samples. The urban classes are often mixed up, which also reflects the "semantic gap" between manually annotated semantics and computer-based calculations.

Table 5.3: Confusion matrix of image data collection 17 with detailed level reference data for selected classes.

Actual \ Predicted	Airport	Channel	Forest mixed	High-density residential area	Hill	Industrial area	Medium density residential area	Aerospace facilities	Mixed urban area	Ocean	Road	Skyscraper
Airport	66	6	2	0	17	1	0	6	0	22	3	0
Channel	13	23	4	0	12	11	5	13	4	0	17	0
Forest mixed	0	6	24	0	11	2	10	2	8	0	15	0
High-density residential area	0	0	0	165	0	16	29	0	3	0	3	15
Hill	5	9	7	0	84	0	0	3	1	4	15	0
Industrial area	0	4	0	41	0	96	63	12	14	0	34	9
Medium density residential area	0	2	5	35	0	40	703	0	92	0	38	8
Aerospace facilities	5	11	1	0	3	5	11	51	6	0	9	0
Mixed urban area	0	5	7	10	2	47	265	12	124	0	42	0
Ocean	47	2	0	0	8	0	0	1	0	207	0	0
Road	0	13	11	13	10	69	98	16	35	0	102	6
Skyscraper	0	0	0	31	0	34	19	0	3	0	10	102

In general, with regard to detailed level reference data, semi-supervised classifiers deliver worse results than supervised classifiers. For the general level reference data, we have 8

semantic classes with more than 120 patches per class; for the detailed level reference data, we have 30 semantic classes with around 15 patches per class. Hence, there are too few samples for each detailed class, so that the trained classifiers are too weak to make correct decisions.

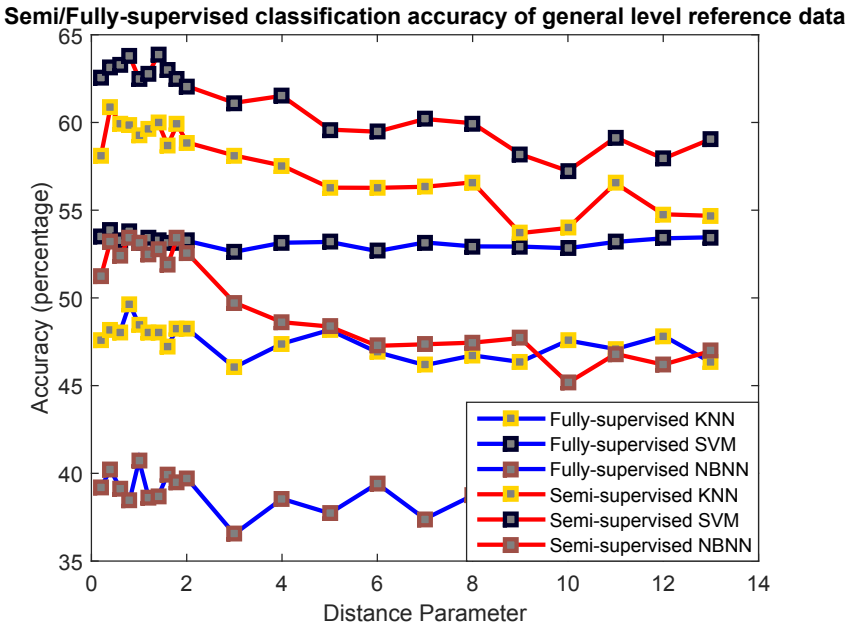


Figure 5.11: Accuracy of general level classifications for the modified G-means algorithm for the data collection 17. The blue line shows results for fully-supervised classification. The red line shows results for semi-supervised classification. Different colors of squares represent different algorithms.

For the data collection 17, Fig. 5.11 and Fig. 5.13 depict the overall classification accuracy of general level data for each distance metric, with KNN, SVM, and NBNN as classifiers, for the original and modified G-means algorithms. We notice for both algorithms:

- For the semi-supervised classifiers, the highest classification accuracy is obtained by SVM, followed by KNN. The accuracy versus distance parameter is decreasing with higher distance parameter values.
- For the supervised classifiers, we obtain a curve ordering similar to the semi-supervised curves. The tendency of the accuracy is fluctuating but stays relatively constant.
- When we compare the semi-supervised classifiers with the supervised classifiers, the semi-supervised classifiers perform better.

Fig. 5.12 and 5.14 depict the overall classification F-score of the general level reference data for each distance metric, using different supervised and semi-supervised classifiers (KNN, SVM, and NBNN), for the original and modified G-means algorithms. We observe that:

- For the semi-supervised classifiers, we notice that the F-score decreases with higher values of Lp. The highest F-score value is obtained by KNN, followed by NBNN.
- For the supervised classifiers, we obtain a curve ordering similar to the semi-supervised curves. The F-score value is fluctuating but stays relatively constant.

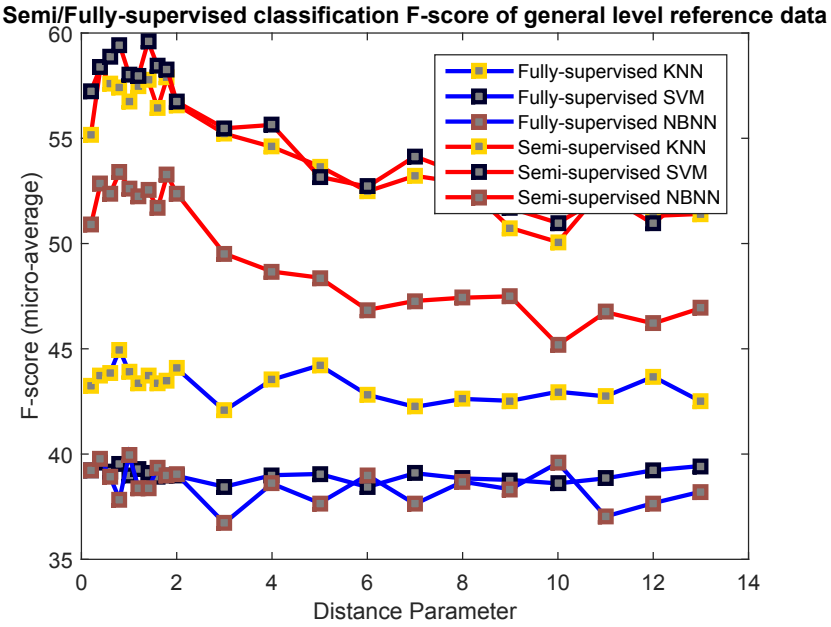


Figure 5.12: Micro-average F-score of general level classifications for the modified G-means algorithm for the data collection 17. The blue line shows results for fully-supervised classification. The red line shows results for semi-supervised classification. Different colors of squares represent different algorithms.

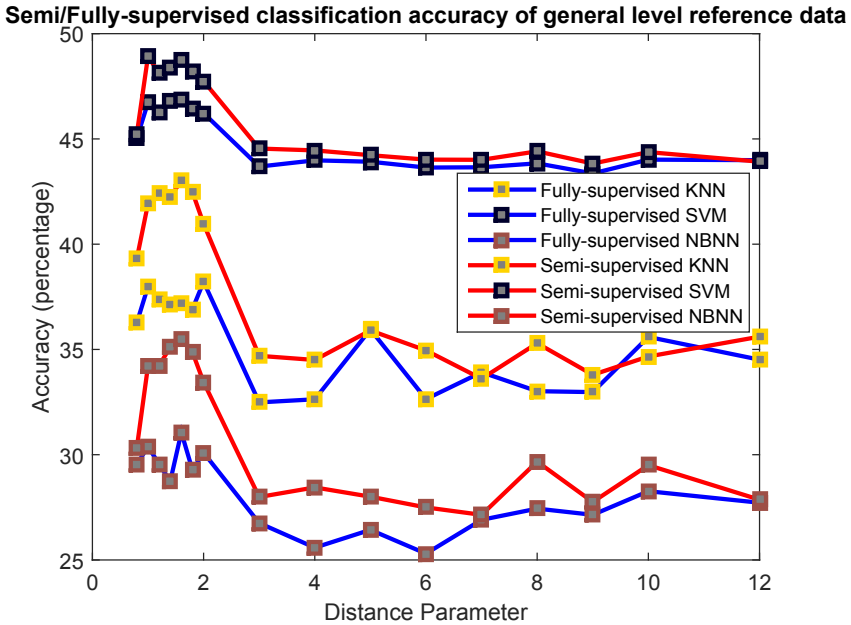


Figure 5.13: Accuracy of general level classifications for the original G-means algorithm for the data collection 17. The blue line shows results for fully-supervised classification. The red line shows results for semi-supervised classification. Different colors of squares represent different algorithms.

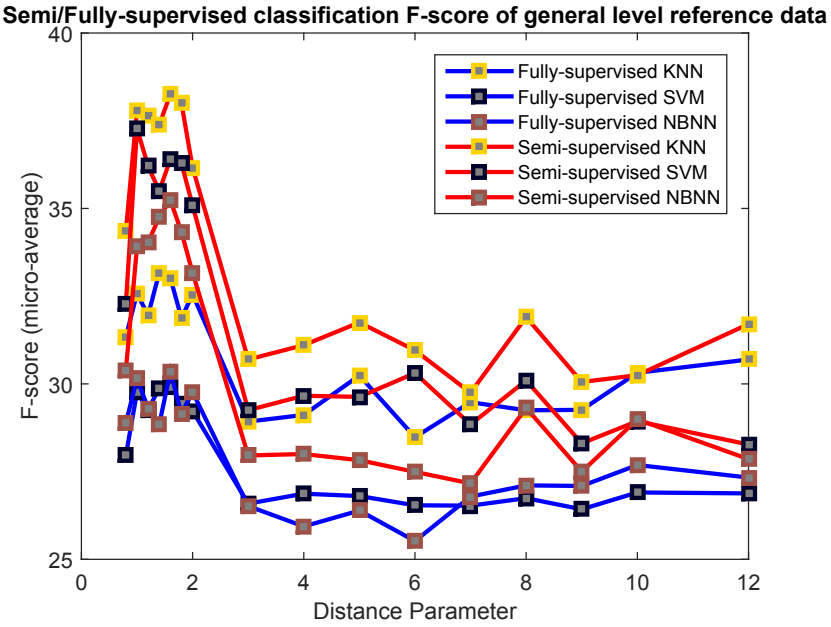


Figure 5.14: Micro-average F-score of general level classifications for the original G-means algorithm for the data collection 17. The blue line shows results for fully-supervised classification. The red line shows results for semi-supervised classification. Different colors of squares represents different algorithms.

- When we compare the semi-supervised classifiers with the supervised classifiers, the semi-supervised classifiers perform better.

For the accuracy and F-scores, the values of the modified G-means algorithm are higher than the original ones.

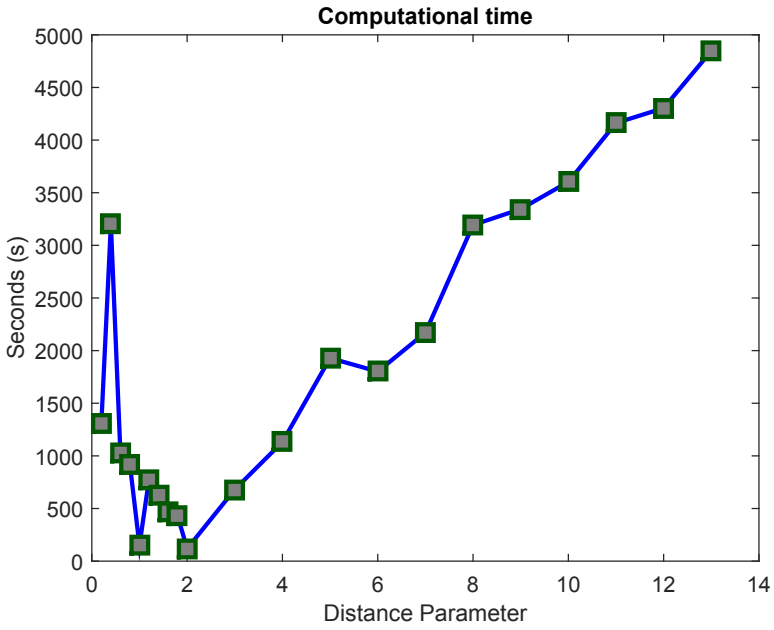
On the one hand, the results verified our guess that with more samples in each class, the cluster-then-label semi-supervised learning performs better than fully supervised learning. On the other hand, the need for a relatively large number of samples limits the applicability of the former method. To sum up, in terms of learning accuracy and F-score value, it is obvious that fractional distances give better performance. In terms of computational time, the classical Euclidean distance as well as the Manhattan distance cost much less time, while the fractional distances take around 100 times more time than the Euclidean and Manhattan norms. In terms of the number of clusters, Minkowski distances yield fewer clusters than the number of clusters for detailed level reference data, while fractional distances generate more and thus smaller clusters. As a consequence of the compromise between efficiency and performance, we chose $L = 1$ as our optimal distance parameter for subsequent analyses.

5.2.3.3 Additional Evaluations

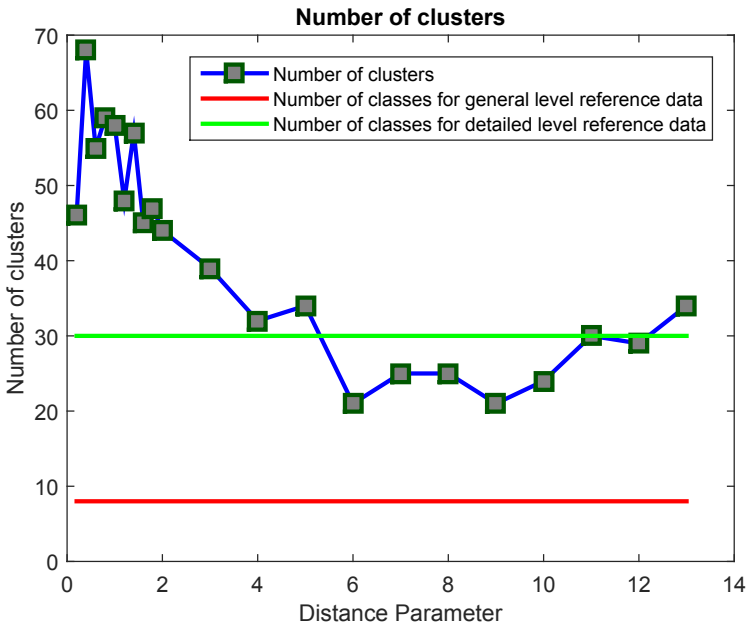
Besides the internal and external evaluations of the clustering results, we also considered the computational time and the number of generated clusters. Fig. 5.15 shows the computational time and the number of generated clusters for different distance metrics for the image data from collection 17.

Fig. 5.15(a) shows the computational time for all Lp distance results. We can observe that:

- For $p = 1$ and $p = 2$, we obtain the lowest computational time.



(a)



(b)

Figure 5.15: Evaluation of data collection 17. (a) Computational time. (b) Number of clusters. The number of annotated classes for the general level reference data is 8, the number of annotated classes for the detailed level reference data is 30.

- For $p < 1$ and $1 < p < 2$, the computational time is higher than for $p = 1$ and $p = 2$.
- For $p > 2$, the computational time is increasing.

Fig. 5.15(b) presents the number of clusters for all Lp distance results. When we look at the classification accuracies of the generated clusters, we need to take into account that the general level reference data contain 8 classes, while the full number of manually referenced data is 30. The results show that:

- For $p < 6$, the number of clusters is fluctuating but the tendency is decreasing versus distance.
- For $6 < p < 12$, the number of clusters is lower than the attainable maximum (i.e., 30).

5.2.4 Visual Evaluations

All our visual results were generated with a distance metric of $L = 1$ for data collection 17.

5.2.4.1 Tree Structure

Fig. 5.16 shows the hierarchical tree structure with 4 to 10 layers by using our proposed method. A cluster that follows a Gaussian distribution and contains a sufficient number of patches is labeled as "1"; otherwise, it is labeled as "0". The homogeneities within the clusters (i.e., the zoomed area shown in Fig. 5.16) are analyzed in the Cluster Homogeneity sub-section.

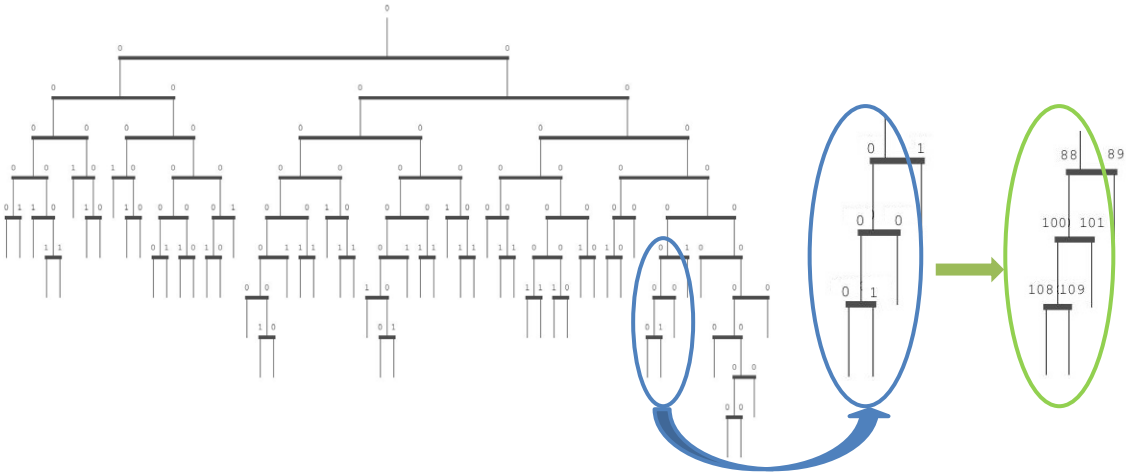


Figure 5.16: Cluster tree structure of the data collection 17. The blue ellipse shows some zoomed clusters, the green ellipse shows the corresponding cluster numbering.

5.2.4.2 Feature Space Visualization

By using the t-SNE algorithm Van der Maaten and Hinton [2008] which is claimed to preserve the complete local structure and some global structure of the data points, the original 48-dimensional feature space was reduced to three dimensions.

- Fig. 5.17 and Fig. 5.18 show that due to the human interaction, the semantic annotations change the separating surfaces among the different classes, which then result in the spreading of a single class across the whole feature space.
- Fig. 5.19 demonstrates the effectiveness of the cluster stopping criterion (i.e., the Gaussian hypothesis test), each cluster is visually compactly grouped.
- Many classes are so spread out that without human interaction or supervision, it is impossible to rely on unsupervised learning methods to separate classes.

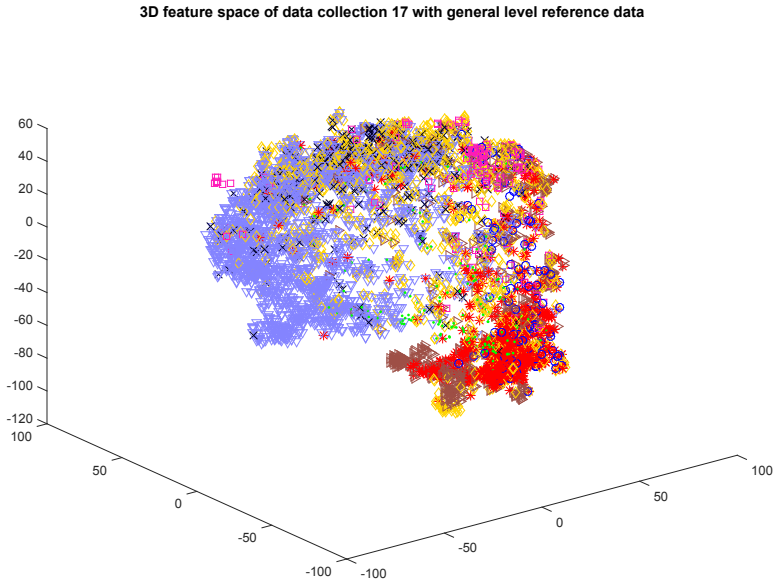


Figure 5.17: 3D feature space of the data collection 17 with general level reference data. The different colors stand for classes with different semantic annotations. The axis scaling refers to projected t-SNE results Van der Maaten and Hinton [2008].

5.2.4.3 Cluster Centroid Patches

We chose three scales of cluster distances: most compact, mid-compact and spread out. For each cluster, the distance-based feature-patch correspondence is illustrated by three representative patches: the closest patch, the mid-distance patch, and the farthest patch from the cluster center. Fig. 5.20 to Fig. 5.22 illustrate how the distance influences the representative patches of the clusters.

- Fig. 5.20 shows the patches with very low intensity which correspond to water body classes.
- Fig. 5.21 shows the patches with mid-intensity which correspond to transportation or agriculture classes.
- Fig. 5.22 shows the patches with very high intensity which correspond to urban areas.
- Fig. 5.20 to Fig. 5.22 demonstrate that high-intensity urban area classes tend to yield spread out clusters; in contrast, water bodies and agriculture classes with low intensity tend to result in compact clusters.

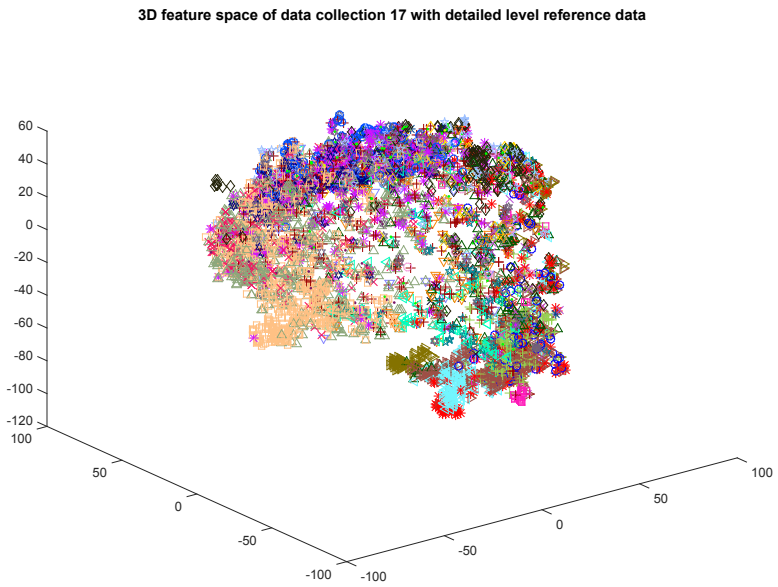


Figure 5.18: 3D feature space of the data collection 17 with detailed level reference data. The different colors stand for classes with different semantic annotations. The axis scaling refers to projected t-SNE results Van der Maaten and Hinton [2008].

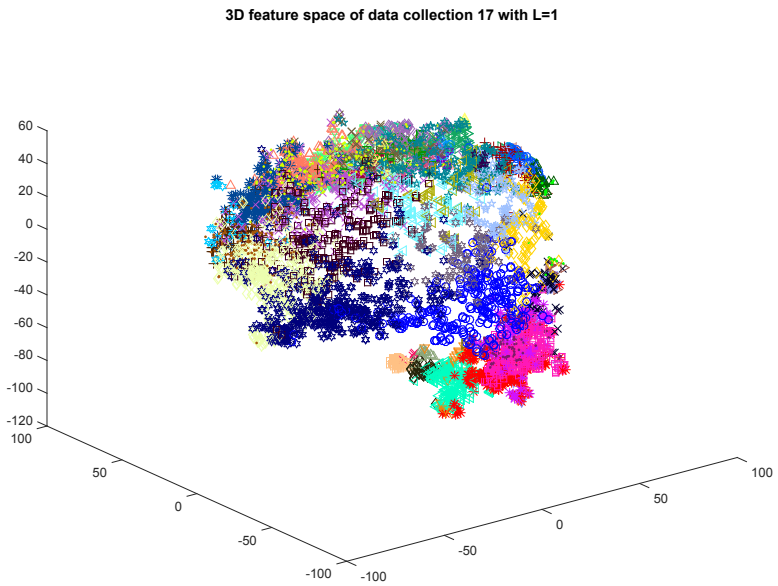
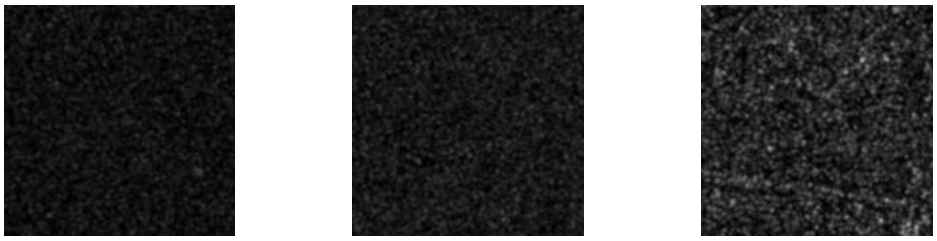
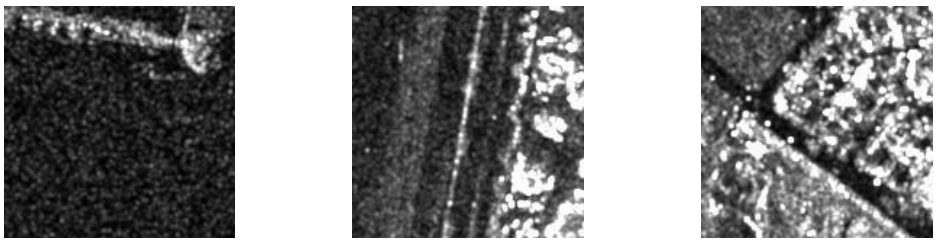


Figure 5.19: 3D feature space of the data collection 17 with a distance parameter of 1. The different colors stand for different obtained clusters. The axis scaling refers to projected t-SNE results Van der Maaten and Hinton [2008].



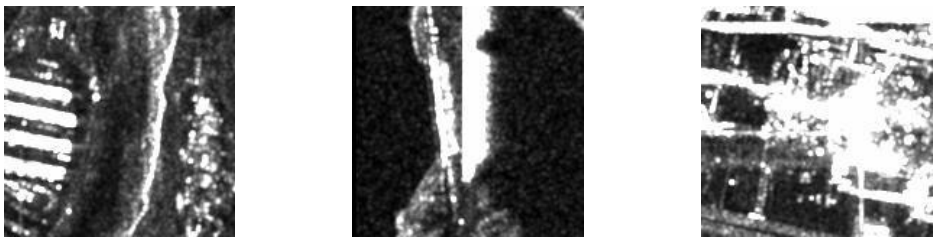
(a) (b) (c)

Figure 5.20: Patch examples of the most compact cluster (collection 17). (a) The patch closest to the cluster center. (b) The mid-distance patch from the cluster center. (c) The patch farthest from the cluster center.



(a) (b) (c)

Figure 5.21: Patch examples of the mid-compact cluster (collection 17). (a) The patch closest to the cluster center. (b) The mid-distance patch from the cluster center. (c) The patch farthest from the cluster center.



(a) (b) (c)

Figure 5.22: Patch examples of the most spread out cluster (collection 17). (a) The patch closest to the cluster center. (b) The mid-distance patch from the cluster center. (c) The patch farthest from the cluster center.

5.2.4.4 Cluster Homogeneity

- Fig. 5.23 and Fig. 5.24 show the layer-wise splitting of clusters for general and detailed level reference data, which corresponds to the zoomed clusters in Fig. 5.16. The size of the pie-diagrams reflects the number of patches within each cluster. Clusters 88, 100 and 109 represent the main route of cluster splitting, where urban areas are the dominant class in the general level reference data, and high-density residential areas are the dominant class in the detailed level reference data.

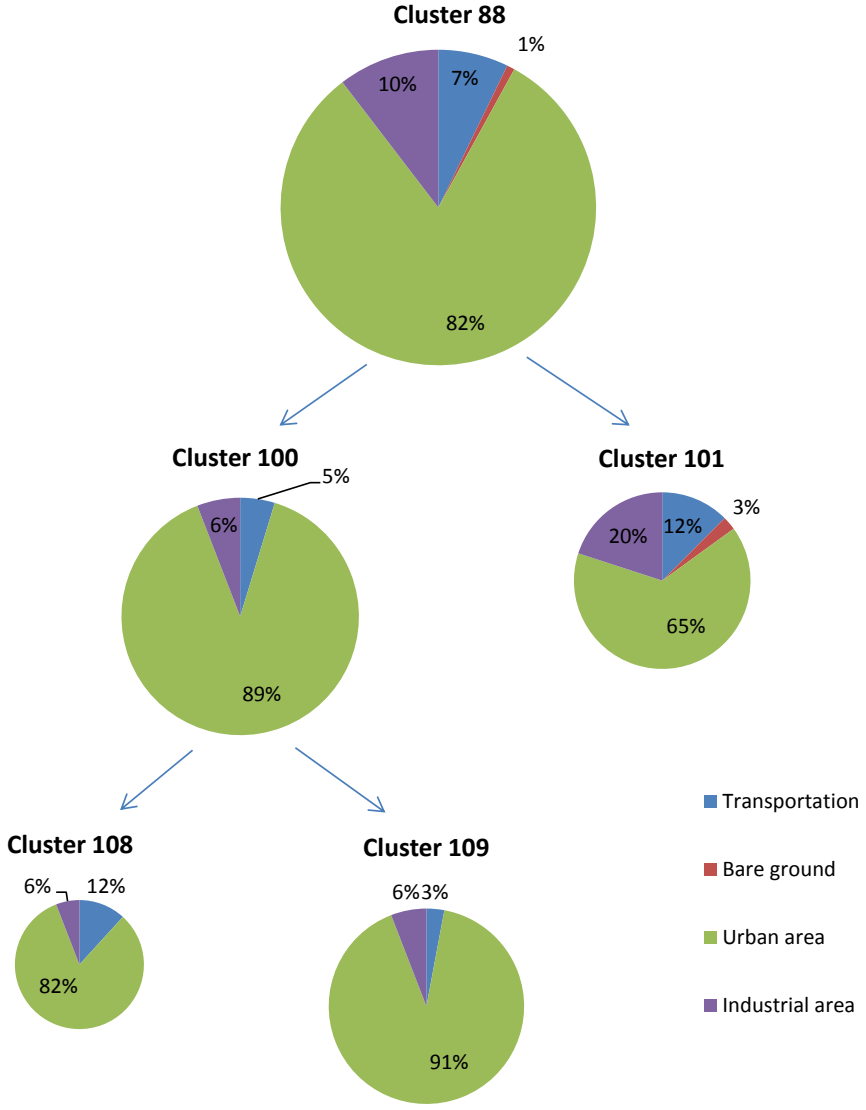


Figure 5.23: Cluster splitting for the data collection 17 with the general level reference data.

- Fig. 5.25 and Fig. 5.26 show the overall class-cluster distribution for the general and detailed level reference data, respectively. For the general level reference data, most of the clusters contain one or two dominant classes; for the detailed level reference data, due to the increased number of classes, the distributions tend to be more disordered than for the general level reference data. However, there are a number of clusters with highly dominant classes, which appear to be very homogeneous.

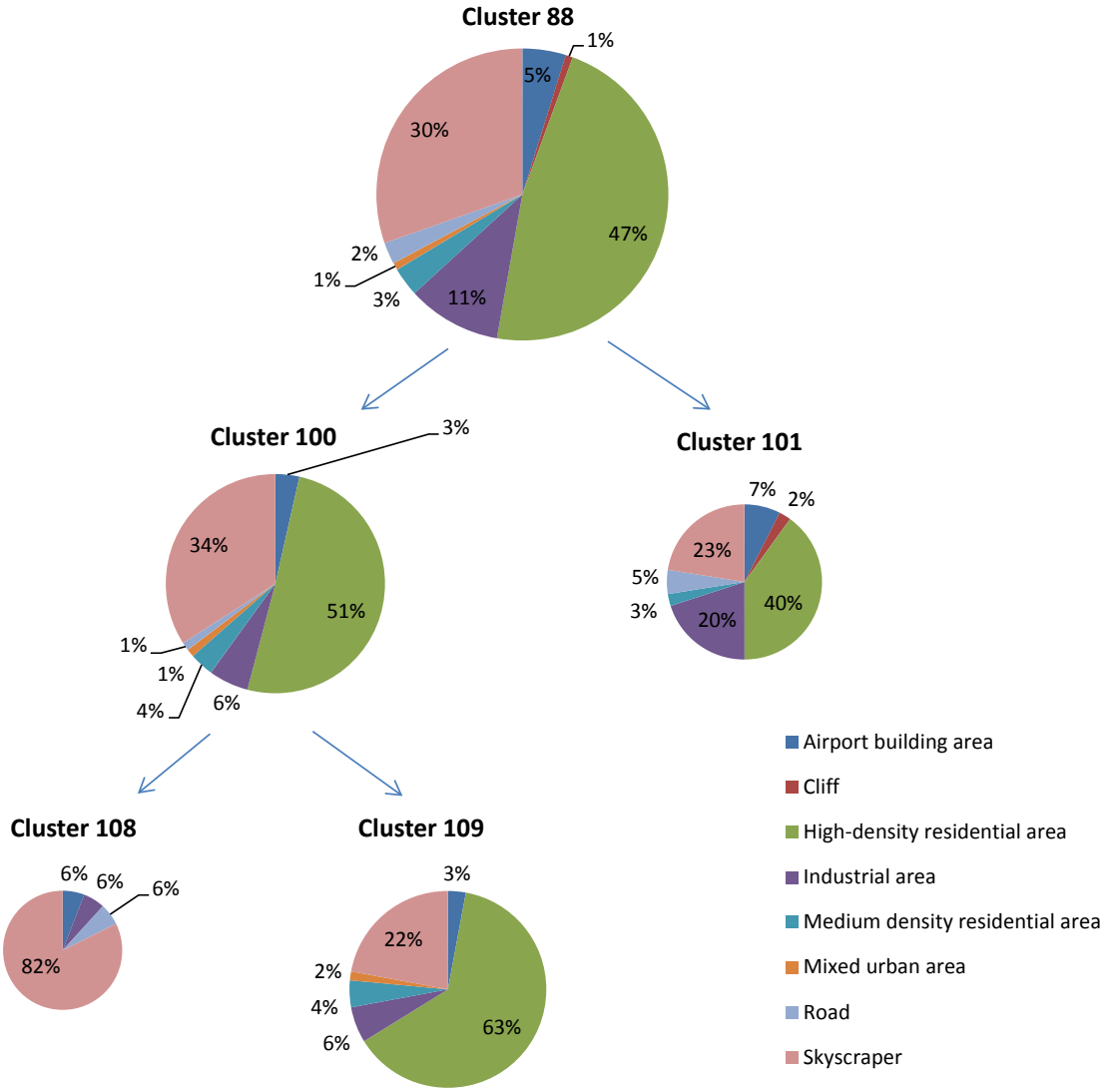


Figure 5.24: Cluster splitting for the data collection 17 with the detailed level reference data.

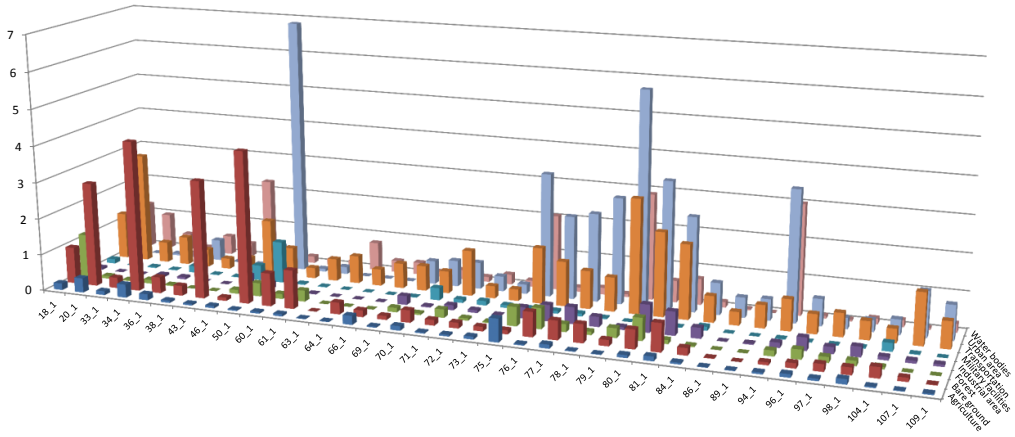


Figure 5.25: The class-cluster distribution of data collection 17 with general level reference data. The X axis represents the last layer Gaussian-distributed cluster number; the Y axis shows the general level reference class labels; the Z axis represents the percentage of the number of patches versus the total number of patches.

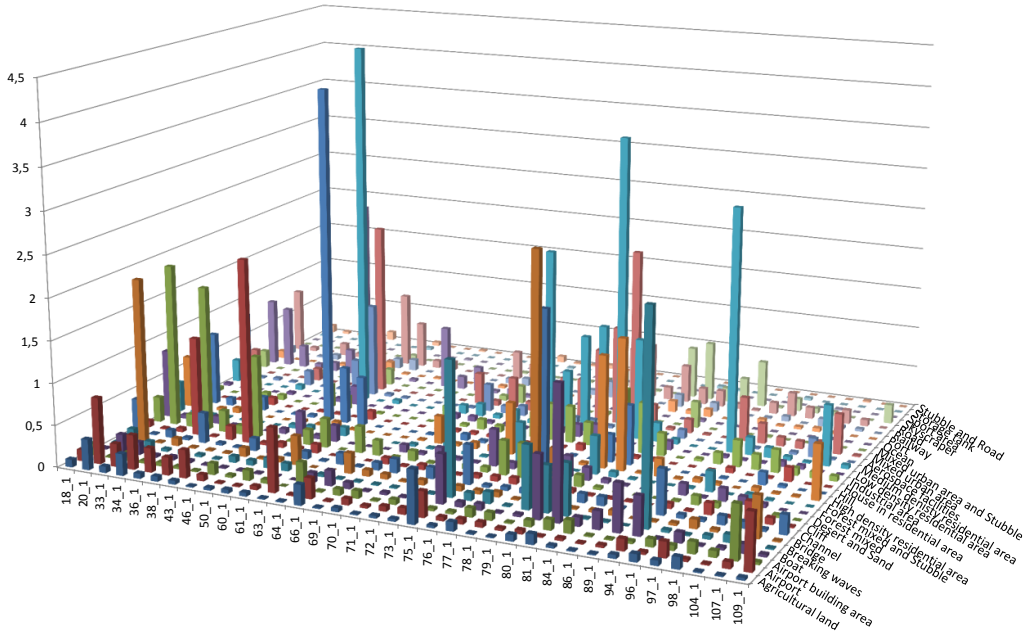


Figure 5.26: The class-cluster distribution of data collection 17 with detailed level reference data. The X axis represents the last layer Gaussian-distributed cluster number; the Y axis shows the detailed level reference class labels; the Z axis represents the percentage of the number of patches versus the total number of patches.

5.3 Conclusions

In this chapter, we proposed a processing and analyzing procedure for large-scale SAR image annotation, which is illustrated in Fig. 5.1. For the purpose of testing and evaluation, four semantically annotated data collections with two-level reference data was prepared. We used them for the analysis of two methods proposed in the introduction, a hierarchical cluster splitting method and various distance metrics (fractional and Minkowski distances) in order to explore the information contained in the feature space. We compared the semi-supervised results and the annotated reference data and analyzed the relations between the clustering results and the general level reference as well as the relations between the semi-supervised classification results and the detailed level reference data. Based on quantitative and visual evaluations of the experimental results, we compared relations among the clustering results, the semi-supervised classification results, and the general and detailed level reference data. It turned out that our proposed method is able to obtain reliable results for the general level reference data; however, due to the too many detailed sub-classes and their few instances, the proposed method generates inferior results for the detailed level reference data. Similar results were obtained for all data collections.

5.3.1 Clustering

There were two main issues when we analyzed unsupervised clustering: The distance metrics (i.e., fractional distance, Minkowski distance), and the termination criterion for the cluster splitting (i.e., the Gaussian hypothesis test).

Regarding the overall classification accuracy and F-score, fractional distances outperform Minkowski distances. The stop criterion works well; when we visualized the feature space, the clusters were grouped compactly; during cluster splitting, the clusters tend to become homogeneous with one or two dominant classes. This can be seen in the pie diagrams. Moreover, by observing the quick-look patches of the clusters it turned out that most of the clusters are very homogeneous, although they may have different semantic labels.

As demonstrated by Figures 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13 and 5.14 the modified G-means algorithm performs better than the original G-means algorithm.

5.3.2 Classifiers

For the classification accuracy and F-score value, we investigated the performances of different supervised learning methods that were used within the clusters (i.e., SVM, KNN, and NBNN).

As for the overall classification accuracy, SVM always achieves the best results both for general and detailed level reference data. In the case of the F-score value, the nearest neighbor methods (i.e., KNN and NBNN) performs better than SVM. The reason behind it is that SVM obtains better classification accuracies for some dominant classes, but lower classification accuracies for the other minority classes.

When the number of classes within a cluster is increasing (e.g., semi-supervised NBNN compared to semi-supervised KNN, for F-score values of detailed level reference data), NBNN performs better than KNN. This indicates that NBNN tends to provide good results when we have a limited number of samples in each class.

5.3.3 Semi-supervised Learning and Manually Annotated Reference Data

As mentioned in Carneiro et al. [2007], unsupervised clustering makes weaker demands on the quality of the manual annotations which largely reduced the human effort. On the other hand, it does not explicitly treat semantics as image classes; therefore, it is not guaranteed that the semantic annotations are optimal in a recognition or retrieval sense.

In order to bridge this "semantic gap", the relationships between the unsupervised clusters and the general and detailed level reference data have to be discussed:

When we look at the relations between clustering results and general level reference data, and the relations between clustering results and detailed level reference data, usually a cluster comprised 5 to 6 classes, with one or two of them being dominant. Of course, the correspondence is better for the general level reference data due to their lower number of classes. With a good classifier, each patch is labeled correctly with a probability of 0.6 for the general level reference data, and with a probability of 0.4 for the detailed level reference data.

5.3.4 Semi-Annotation/Labeling

In the end, we provide a general framework which can be used as a reference by other practitioners who are also interested in large-scale SAR image dataset annotations. We recommend the following procedure:

- Tile the data set into image patches and extract patch features.
- Use clustering methods to group feature data into clusters based on distance similarity. It is better to use fractional distances or Manhattan distances rather than the traditional Euclidean distances.
- Within each cluster, label some general classes manually; then use a supervised learning method to label the remaining unlabeled patches.

Chapter 6

Pixel-Level Bayesian Classification and Active Learning Based Object Extraction

> Our life is endless in the way that our visual field is without limit.
>
> Ludwig Wittgenstein

Currently, most remote sensing classification and recognition researches and applications are interested in large-scale areas, e.g., image patches. However, in addition to patch-level methods, which are presented in Chapters 4 and 5, in this chapter, we will present pixel-level and object-level methods for high-resolution satellite imagery. Due to the special inherent imaging mechanisms of SAR images, e.g., the speckle effect, it is difficult to perform pixel-level classification. Hence, in the first part of this chapter, we try to model joint density distributions of different features for very general categories: urban, water, etc. Later, an active learning method based on the non-locality concept proposed by Pierre Blanchart in Blanchart and Ferecatu [2015], and Blanchart and Ferecatu [2014], is discussed for optical images to extract more detailed objects: buildings, rivers, sports fields, etc. The most important contributions of this chapter are the attempts to reach pixel-level classification using TerraSAR-X imagery for general semantic categories, and object-level extraction using WorldView-2 imagery for detailed semantic categories. The experimental results show that, currently it is rather difficult to extract detailed-level categories from SAR imagery; on the contrary, it is a reachable goal for optical imagery.

6.1 Pixel-Level Bayesian Classification

A novel speckle statistics feature for the characterization of speckle development has been presented by Esch et al. [2011]. It shows significant differences for certain basic land cover types: urban, forest, and water. As well, the $\mathcal{G}^0$ model has been proposed to model SAR image intensities by Frery et al. [1997] for extremely heterogeneous regions, such as urban areas. Hence, in this research, we focus on pixel-based image classifications of high-resolution SAR images which statistically model the joint probability densities of speckle statistics feature and image intensities under a Bayesian classification framework, regarding three specific land cover types: urban, forest, and water.

6.1.1 Modeling of Speckle Statistics Feature

As explained in Esch et al. [2011], the speckle statistics feature modeling is implemented by the following steps: The description of the local development of speckle, also known as image texture in a SAR image, is the local image heterogeneity expressed by the coefficient of variation:

$$C_S = \frac{\sigma_S}{\mu_S} \tag{6.1}$$

where μ_S describes the mean **amplitude value** and σ_S stands for the standard deviation of the amplitude values within an image window. C_S has a relationship between true texture C_T and fading texture C_F induced by speckle:

$$C_S^2 = C_T^2 C_F^2 + C_T^2 + C_F^2 \tag{6.2}$$

where the speckle-induced heterogeneity C_F can be estimated via the number of looks N

$$C_F = \frac{\sigma_F}{\mu_F} \tag{6.3}$$

with

$$\mu_F = \frac{\Gamma(N + 1/2)}{N^{1/2}\Gamma(N)} \tag{6.4}$$

and

$$\sigma_F^2 = 1 - \mu_F^2. \tag{6.5}$$

Here, by reading the metadata file of a TerraSAR-X products, the number of looks is calculated as a product of azimuth looks and range looks.

The discrepancy between C_S and C_F can be used to quantify the true texture C_T:

$$C_T^2 = \frac{C_S^2 - C_F^2}{1 + C_F^2}. \tag{6.6}$$

It is recommended to add a majority procedure by calculating the mean of these local measurements in a larger window, which can reduce the feature's sensitivity towards local noise:

$$C_{Tm}^2 = \frac{1}{M} \sum_{i=1}^{n} C_{Ti}^2. \tag{6.7}$$

Because the speckle statistics distribution is unknown, we use a Weibull distribution as suggested in Esch et al. [2011], as well as a $\mathcal{G}^0$ distribution and kernel density estimator for comparison to model the distribution. Based on the definition of the Weibull distribution in Chapter 2, the probability density of the speckle statistics feature is expressed as:

$$f_{C_{Tm}^2}(c_{Tm}^2) = \frac{\beta}{\alpha^\beta} {c_{Tm}^2}^{\beta-1} \exp\left[-(\frac{c_{Tm}^2}{\alpha})^\beta\right], c_{Tm}^2 \geqslant 0. \tag{6.8}$$

6.1.2 Image Intensity Modeling

In addition to the classic statistical properties modeling of SAR images, which are presented in Chapter 2, alternatively, a multiplicative model is also used as a parametric model for SAR images which explains the stochastic behavior of data obtained with coherent illumination. With the assumption of fully developed speckle, a pixel y in a SAR images is the outcome of the product of two independent random variables: the terrain backscatter x and the speckle noise z.

$$y = x \cdot z, \tag{6.9}$$

where y is the observed image pixel, x is the noise free pixel, and z is the speckle noise. The most common multiplicative noise models are the $\mathcal{G}$ and the $\mathcal{K}$ distributions.

Extremely heterogeneous regions in high-resolution SAR image follow a generalized inverse Gaussian distribution Frery et al. [1997], therefore, we use the $\mathcal{G}^0$ distribution to model the probability density of SAR intensities which is defined as:

$$f_{I|\alpha,\beta,L}(i|\alpha,\beta,L) = \frac{L^L\Gamma(L-\beta)}{\alpha^\beta\Gamma(L)\Gamma(-\beta)}\frac{i^{L-1}}{(\alpha+Li)^{L-\beta}}. \tag{6.10}$$

6.1.3 Combined Intensity - Speckle Statistics Feature Model

In the recent literature about statistical SAR image modeling, the method of log cumulant (MoLC) estimation is being widely used; it is well adapted to urban areas, and less sensitive to large values Tison et al. [2004]. Hence, in this research, we estimate the marginal PDF of the speckle statistics feature, and the marginal PDF of intensities by solving the MoLC equations of Table 6.1 Cui et al. [2014]. The marginal PDFs for the categories urban, water, and forest are expressed by $f_{C^2_{Tm}}(c^2_{Tm}|urban)$, $f_{C^2_{Tm}}(c^2_{Tm}|water)$, $f_{C^2_{Tm}}(c^2_{Tm}|forest)$, $f_I(i|urban)$, $f_I(i|water)$ and $f_I(i|forest)$.

As two features (SAR intensities and speckle statistics feature) are considered, a joint distribution model of the speckle statistics feature, and image intensities can be obtained via Copula functions which have been explained in Chapter 2 Voisin et al. [2013]. The definition of a bivariate copula function is a joint cumulative distribution of two uniform random variables X_1 and X_2,

$$C_{X_1,X_2}(x_1,x_2) = P(X_1 < x_1, X_2 < x_2) \tag{6.11}$$

where $X_i \sim U(0,1)$ for $i = 1,2$.

In order to construct a land cover specific statistical model for urban, forest and water land covers, their joint probability distribution is defined by the following formula. The bivariate Archimedean family copulas which have been defined in Table 2.2 are chosen to fit the joint PDF model. Moreover, as the probability density function rather than the cumulative density function is what we are interested in, the problem is simplified via Sklar's theorem as follows:

$$\begin{aligned} f_{C^2_{Tm},I}(c^2_{Tm},i|class_i) &= \frac{\partial^2 C_{C^2_{Tm},I}(c^2_{Tm},i|class_i)}{\partial(C^2_{Tm}|class_i)\partial(I|class_i)} \\ &= c(F_{C^2_{Tm}}(c^2_{Tm}|class_i), F_I(i|class_i)) f_{C^2_{Tm}}(c^2_{Tm}|class_i) f_I(i|class_i), \end{aligned} \tag{6.12}$$

where c is the density of the copula, $class_i$ is the label for class i, with $i \in [1,3]$ since we consider three categories: urban, forest, and water. The $F_{C^2_{Tm}}(c^2_{Tm})$ and $F_I(i)$ are the

Table 6.1: PDFs and MoLC equations for the $\mathcal{G}^0$ and Weibull distributions

Model	Probability Density Function	MoLC Equations	Parameters
$\mathcal{G}^0$	$f_{X\|\alpha,\beta,L}(x\|\alpha,\beta,L) = \frac{L^L\Gamma(L-\beta)}{\alpha^\beta\Gamma(L)\Gamma(-\beta)}\frac{x^{L-1}}{(\alpha+Lx)^{L-\beta}}$	$k_1 = \frac{\ln(\alpha)}{L} + \phi(L)\phi(-\beta)$ $k_i =$ $\phi(i-1,L) + (-1)^i\phi(i-1,-\beta)$ $i = 2,3$	α: scale parameter β: shape parameter L: number of looks
$Weibull$	$f_{X\|\alpha,\beta}(x\|\alpha,\beta) = \frac{\beta}{\alpha^\beta}x^1\exp\left[-(\frac{x}{\alpha})^\beta\right], x \geqslant 0$	$k_1 = \ln(\alpha) + \beta^{-1}\phi(0,1)$ $k_2 = \beta^{-2}\phi(1,1)$	α: scale parameter β: shape parameter

cumulative distributions of C^2_{Tm} and I. Thus, the likelihood PDFs $f_{C^2_{Tm},I}(c^2_{Tm}, i|urban)$, $f_{C^2_{Tm},I}(c^2_{Tm}, i|water)$ and $f_{C^2_{Tm},I}(c^2_{Tm}, i|forest)$ are obtained.

In Formula 6.12, the cumulative distribution is calculated via the following definition:

$$F_X(x) = P(X \leq x) = \sum_{x_i \leq x} P_X(x = x_i) = \sum_{x_i \leq x} P_X(x_i). \tag{6.13}$$

As already explained in Chapter 2, we are using the MoLC estimation method to estimate the probability density functions for image intensities and the speckle statistics feature. In analogy with the classic moment generating functions and cumulant generating functions, which are used in probability theory and statistics, MoLC uses second kind statistics which consist of a second kind moment generating function, and a second kind cumulant generating function. In terms of computation, the second kind cumulant generating functions can be estimated by observed samples:

$$\tilde{k}_1 = \frac{1}{n}\sum_{i=1}^{n}\log x_i, \tilde{k}_i = \frac{1}{n}\sum_{i=1}^{n}(\log x_i - \tilde{k}_1)^i. \tag{6.14}$$

6.1.4 Bayesian Classification

For classification, a Bayesian classifier which is explained in Chapter 2 is used to model and classify different land cover types in high-resolution TerraSAR-X images. A Bayesian classifier obeys Bayes' Rule, the posterior probability is proportional to the product of prior probability and likelihood probability:

$$\begin{aligned} f_{C^2_{Tm},I}(class_i|c^2_{Tm}, i) &= \frac{f(class_i)f_{C^2_{Tm},I}(c^2_{Tm}, i|class_i)}{f_{c^2_{Tm},i}(c^2_{Tm}, i)} \\ &\propto f(class_i)f_{C^2_{Tm},I}(c^2_{Tm}, i|class_i), \end{aligned} \tag{6.15}$$

here $class_i$ is the label for class i, with $i \in [1,3]$ as we consider three categories: urban, forest and water. Hence, based on 6.15, the posterior probabilities for the three categories $f_{C^2_{Tm},I}(urban|c^2_{Tm}, i)$, $f_{C^2_{Tm},I}(forest|C^2_{Tm}, I)$, and $f_{C^2_{Tm},I}(water|C^2_{Tm}, I)$ are generated.

In the end, a classification label is assigned to the class that gives the largest class conditional posterior probability:

$$label = \arg\max_{class} f(class)f_{C^2_{Tm},I}(c^2_{Tm}, i|class). \tag{6.16}$$

The computational procedure is as follows:

- Compute the image intensities I.
- Compute the speckle statistics feature $Ctm2$, according to the formulas in Section 6.1.1.
- Take the logarithms of I and $Ctm2$.
- For the three given categories,

  compute the second kind moments m_1, m_2, and m_3 based on a moment generating function.

  compute the second kind cumulants k_1, k_2, and k_3 using a cumulant generating function which is based on second kind moments.

  according to Table 6.1, solve the MoLC equations of the Weibull and $\mathcal{G}0$ distributions, so that the model parameters β, α, and L are obtained.

  with the MoLC estimated distribution parameters, the marginal **PDFs** and **CDFs** are obtained for image intensities and the speckle statistics feature.

  based on the equation 6.12, model the **copula density** of two cumulative distributions of two random variables, which is also the marginal **likelihood probability distribution**.
- A classification label is assigned to the category that gives the largest **posterior probability distribution**.

6.1.5 Experiments

6.1.5.1 Brief Dataset Description

A dataset consisting of 15 TerraSAR-X images containing cities of North Rhine-Westphalia (NRW), Germany is used to create our training and test database. The images are GEC products with a ground resolution of 2.9m, their incidence angles lie between 20 and 45 degrees, with various acquisition dates and orbits branches. The reason of choosing GEC products rather than SSC products lies in that we need geographic coordinates for comparisons to a ground truth map. With the help of geocoded coordinates, we use the open source data from OpenStreetMap to obtain the corresponding ground truth data. These data largely reduce the human labeling effort and can easily create ground truth for a number of products.

In order to generate the ground truth reference map, we use the coordinate information from the metadata file provided by GEC product, and the polygon position information from the shape file provided by OpenStreetMap. Hence, each pixel of the ground truth reference map has an exact correspondence with the shape file; the image-to-OpenStreetMap coregistration is relatively accurate expect for the geometrical and radiometric distortions of the GEC products which may occur. Moreover, there is one inevitable problem in the OpenStreetMap based reference map generation method: There are some pixels without specific map class information, due to the lack of geographical information from the open source data. In order to overcome this difficulty, in the current experiment setting, we focus our attention only on those pixels which have geographical information, rather than the whole scene.

6.1.5.2 Evaluation

For classifying the three land cover classes: urban, forest, water, a set of experiments for comparison and analysis has been carried out.
In the intermediate step of fitting feature marginal distributions to each land cover class, we noticed that, as expected, a $\mathcal{G}^0$ distribution is promising in modeling highly heterogeneous regions, while the estimated $\mathcal{G}^0$ model can fit data histograms very well; however, for speckle statistics feature, the estimated Weibull model, and the $\mathcal{G}^0$ model show minor differences with respect to histogram shapes and amplitudes; in contrast, the kernel density estimator can exactly fit the data.
Hereinafter three experiments with different models for speckle statistics feature are presented. We used three copula functions (Clayton, Frank, and Gumbel) to model the joint probability. The results using different copula functions are similar; therefore, the most representative results are presented. In the confusion matrix table, each column contains the classification rate between the actual TSX-based result compared to the ground truth reference map.

Histogram fitting Fig. 6.1 and Fig. 6.2 show $\mathcal{G}^0$ distribution fitting for intensities and $\mathcal{G}^0$ distribution fitting for the $Ctm2$ feature, for urban and forest areas. Fig. 6.3 and Fig. 6.4 show $\mathcal{G}^0$ distribution fitting for intensities and $Weibull$ distribution fitting for the $Ctm2$ feature, for urban and forest areas. As can be observed from the figures, a $\mathcal{G}^0$ distribution fits better than a $Weibull$ distribution for the $Ctm2$ feature, which is also proved by the classification results below.

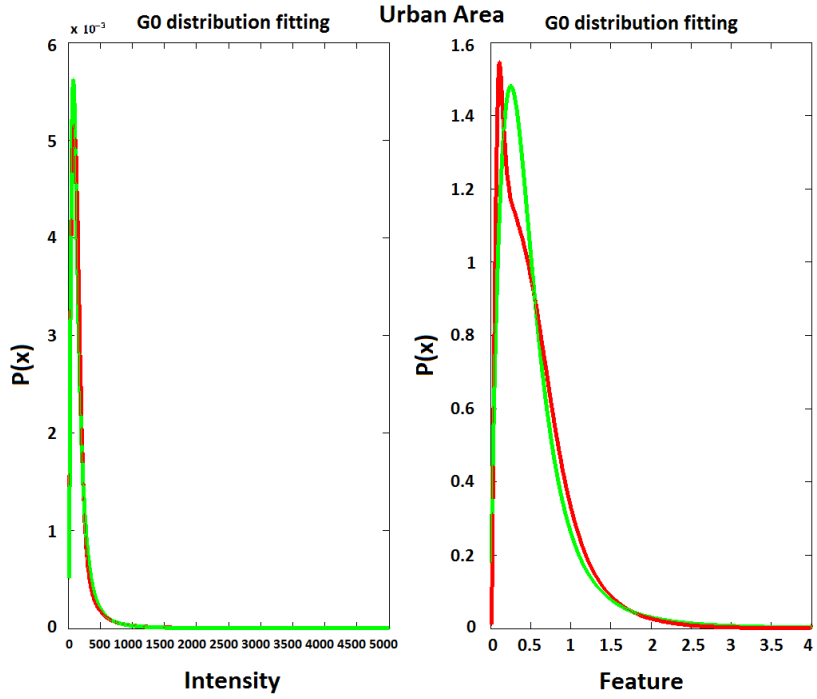


Figure 6.1: Urban area, $\mathcal{G}^0$ distributions for intensities and the $Ctm2$ feature. The color green shows the fitted model, the color red shows the real data.

Experiment 1 We model the intensities with a $\mathcal{G}^0$ distribution, and the speckle statistics feature with a kernel density estimator. By using a Frank copula function, the overall classification accuracy is 72.31%.

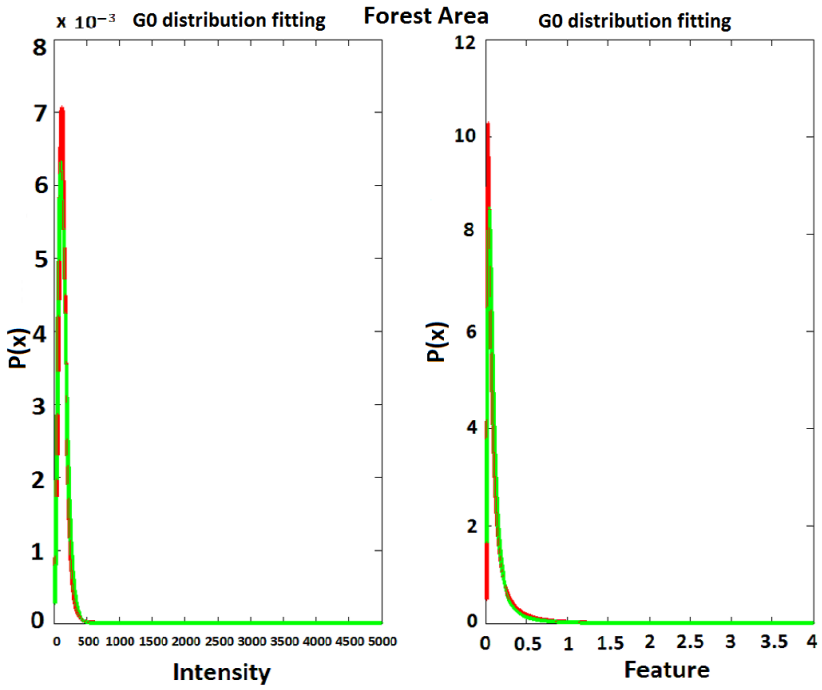


Figure 6.2: Forest area, $\mathcal{G}^0$ distributions for intensities and the $Ctm2$ feature. The color green shows the fitted model, the color red shows the real data.

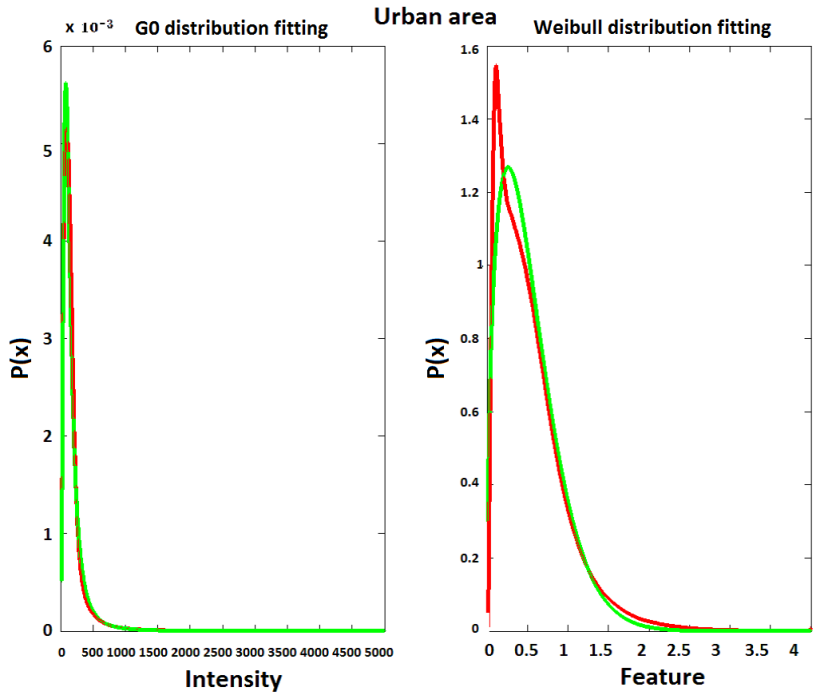


Figure 6.3: Urban area, $\mathcal{G}^0$ distribution for intensities, and Weibull distribution for the $Ctm2$ feature. The color green shows the fitted model, the color red shows the real data.

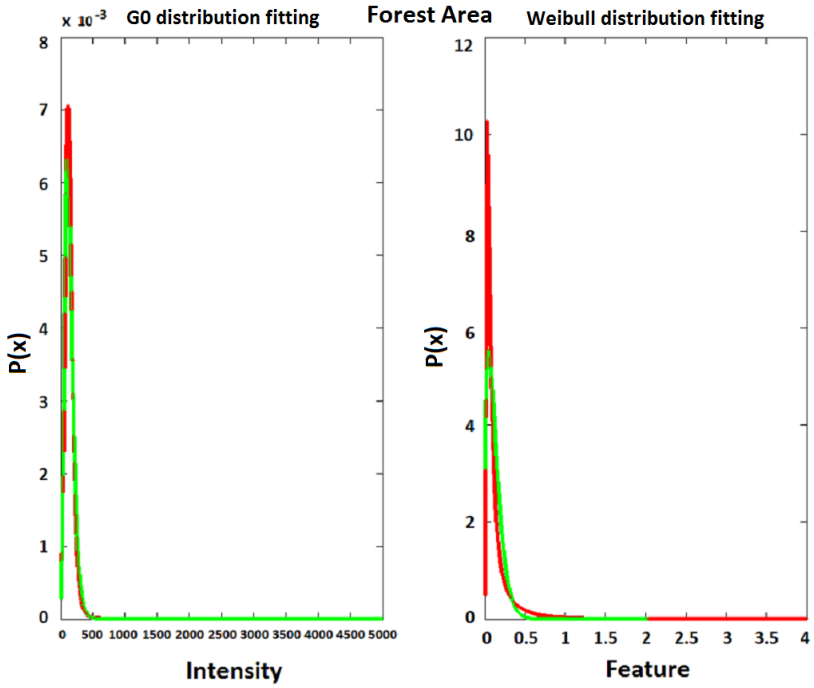


Figure 6.4: Forest area, $\mathcal{G}^0$ distribution for intensities, and Weibull distribution for the $Ctm2$ feature. The color green shows the fitted model, the color red shows the real data.

Table 6.2: Confusion matrix of experiment 1.

	Urban	Forest	Water
Urban	**0.9234**	0.3198	0.5931
Forest	0.0659	**0.6739**	0.3300
Water	0.0107	0.0062	**0.0769**

As stated above, the kernel density estimator can exactly model the texture feature, the overall accuracy is the highest among the three experiments, but the specific class accuracy rate lies in the middle among all of the experiments.

Experiment 2 We model the intensities with a $\mathcal{G}^0$ distribution, and the speckle statistics feature with a Weibull distribution. By using a Frank copula function, the overall classification accuracy is 71.81%.
A Weibull distribution is proposed by Esch et al. [2011] for the speckle statistics feature. The overall accuracy lies in the middle, while the forest area accuracy is the lowest among the three experiments. When looking at the PDF fitting step, the automatic Weibull model did not fit the observed data very well, which may be the main reason for its inferior performance.

Experiment 3 We model intensities and the speckle statistics feature with $\mathcal{G}^0$ distribution. When using a Clayton copula function, the overall classification accuracy is 71.22%.
A $\mathcal{G}^0$ distribution is proposed by Frery et al. [1997] to model heterogeneous areas; our classification results prove this idea, since the urban and forest classification accuracies yield the highest values, while the water accuracy results in the lowest value among the three experiments.

Table 6.3: Confusion matrix of experiment 2.

	Urban	Forest	Water
Urban	**0.9401**	0.3594	0.5903
Forest	0.0488	**0.6332**	0.3256
Water	0.0111	0.0074	**0.0841**

Table 6.4: Confusion matrix of experiment 3.

	Urban	Forest	Water
Urban	**0.9430**	0.2788	0.6408
Forest	0.0462	**0.7160**	0.2977
Water	0.0108	0.0052	**0.0615**

By looking at experiment 1,2, and 3, compared to the method of Esch et al. [2011], whose urban classification accuracy is 0.8261, our proposed classification method has a better performance.
Fig. 6.5 shows a GEC product of Aachen, Germany, whose brightness and contrast has been adjusted in order to show the image details. Fig. 6.6 depicts the corresponding ground truth reference map generated by using geographic information from OpenStreetMap. Fig. 6.7 contains a typical classification result for Aachen, by using the proposed classification procedure. Red represents urban areas, green represents forest areas, gray represents water, and white represents those areas without geographic information. Fig. 6.8 shows an example of a color composite image for Aachen, by overlapping the intensity value and the $Ctm2$ feature.

Figure 6.5: TerraSAR-X GEC product of Aachen city.

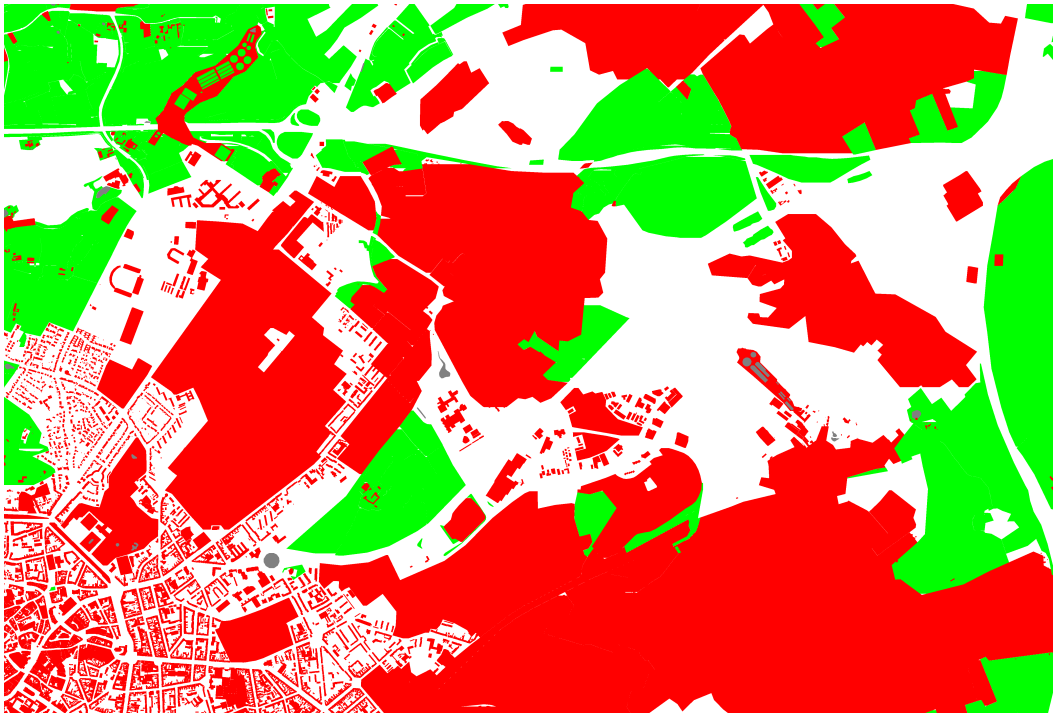


Figure 6.6: Ground truth reference map of Aachen city.

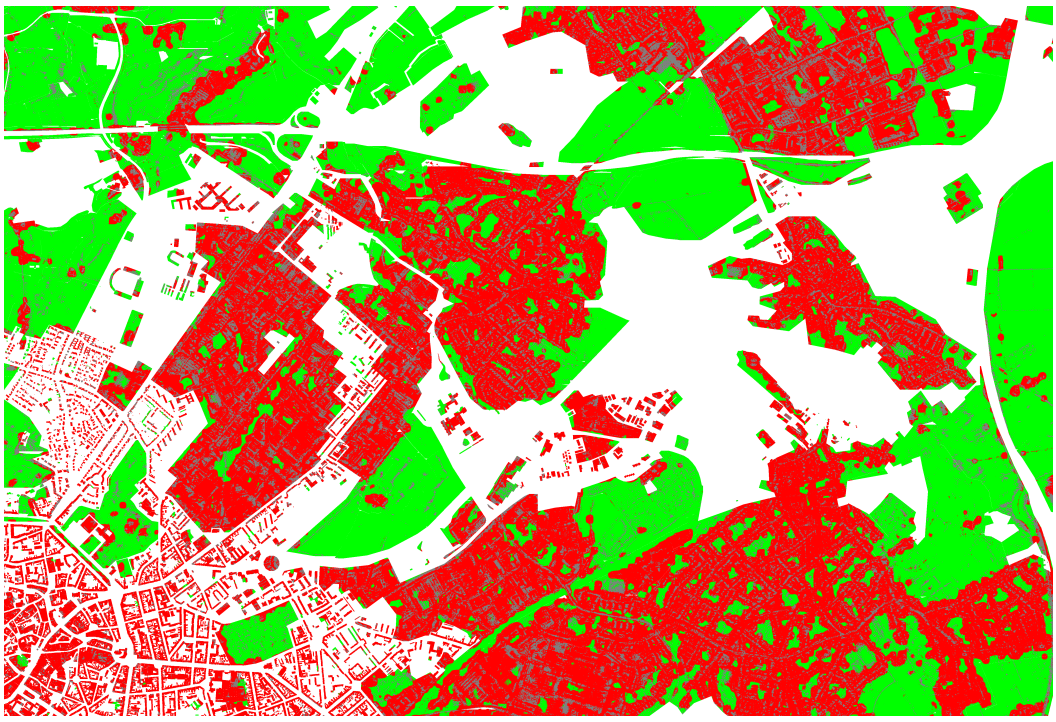


Figure 6.7: Classification result of Aachen city.

Figure 6.8: Pseudo-color image of Aachen city.

6.2 Active Learning Based Object Extraction

Due to the small amount of given training samples, the assumption has been made that the potential corresponding objects in a big scene either visually look similar to or are locationally close to the given samples or similar objects Blanchart and Ferecatu [2014]. As verified by Blanchart and Ferecatu, their non-locality method is able to extract and discover semantic categories within optical imagery; however, in most cases, there is still potential for improving the first results; hence, active learning has been incorporated in this method for improving results via iterations.

Since an SVM is used for the definition of the non-locality map, the corresponding active learning concept is then based on the support vectors which are suggested by the classifier. The proposed active learning procedure is shown in Fig. 6.9.

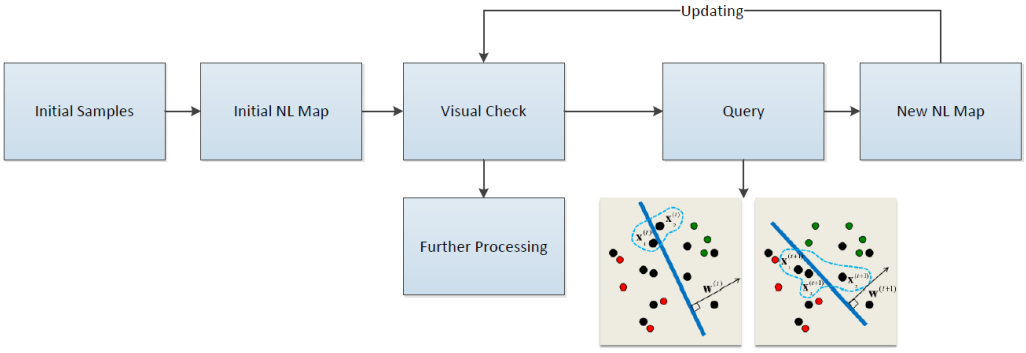


Figure 6.9: Active learning based object extraction for optical imagery.

6.2.1 Definition of Non-Locality

The principle of non-locality has been proposed based on the observation that objects are localized elements in an image with strong intra-part similarity Blanchart and Ferecatu [2015], This similarity is denoted by locality / non-locality: an element from our target object will be surrounded by many other elements of the same object, and thus can be paired with many similar elements inside its neighborhood. This phenomenon is referred as *locality*. Hence, a high "non-locality" refers to the scene background, and a low "non-locality" refers to objects. Thus, similarity and closeness are two important phenomena to define objects that belong to the same category. The *non-locality* NL-score is thus defined as:

$$NL(x) = \sum_{y \in Img} nl(x,y) = \sum_{y \in Img} s(x,y) * d(x,y) \tag{6.17}$$

where $s(x,y)$ is the similarity between elements x and y and $d(x,y)$ is their Euclidean distance in the given image Blanchart et al. [2014].

A Support Vector Machine (SVM) based Supervised Pairwise Target Similarity (SPTS) measure has been proposed to define the similarity part in $NL(x)$, so 6.17 is computed as:

$$NL(x) = \sum_{y \in Img} nl(x,y) = \sum_{y \in Img} (1 - SPTS(x,y)) * f(d_n(x,y)) \tag{6.18}$$

where $f(d) = exp(-\frac{d^2}{h^2})$ is a function that characterizes the object scale to the whole image, and $d_n(x,y) = \frac{d(x,y)}{max(d(x,y))}$ is the normalized Euclidean distance. According to Blanchart et al. [2014], $f(d) = exp(-\frac{d^2}{h^2})$ is taken where h is a spatial bandwidth. The map is normalized between 0 and 1 using the affine transformation which relies on the minimum and maximum value in the non-locality map.

6.2.2 SVM-based Active Learning

Since the calculation of similarity in the non-locality map is based on a support vector machine (SVM), the basic theory of support vector machines (SVMs) is explained in Appendix A as a background knowledge. Generally, there are two basic types of machine learning models: discriminative models and generative models. The former ones estimate conditional probability distributions which are typically used for tasks such as classification and regression; the latter ones rely on the joint distributions which are a full probabilistic model of all variables and require a full understanding of the problem and domain specific expertise in advance. The setback of these models is that once the model type is chosen, the rest is to estimate its parameters; hence, there is seldom a chance for us to upgrade the selected model. This is the typical drawback of classic machine learning.

In analogy to human learning, in the domain of educational research, Prince [2004] mentions, "In the classroom-based learning process, active learning requires students to do meaningful learning activities and think about what they are doing. It is often contrasted to the traditional lecture where students passively receive information from the instructor." Comparatively, there is potential for improvement space that machine learning can also learn from human learning. Namely, when we see our machine (our computer) as a student, a passively learning student (classic machine learning methods) only gets information (an empirical model) from the teacher (us), once being taught knowledge (a fixed model) is composed (built), parameters are set, with no freedom at all; however, on the contrary, an actively learning student (active learning methods) uses his own ideas or judgment to actively propose a possible solution (a sample selection) that boosts a learning procedure, which is with more freedom and learns faster.

6.2.2.1 Version Space

In order to better understand the idea of active learning, there is an important notion called version space that needs to be explained. Given a set of labeled instances $x_i, i = 1, 2, ..., N$ and a kernel function $K(x,y)$, many different hyperplanes $y = w^T\psi(x) + b$ in the mapped higher-dimensional feature space $\mathcal{F}$ can separate the actual classes. This set of hyperplanes is called *version space* Tong [2001 (accessed October 17, 2016]. It is defined as:

$$\mathcal{W} = w|\|w\| = 1, y_i(w^T\psi(x)) > 0, i = 1, 2, ..., N \tag{6.19}$$

The instances that correspond to these hyperplanes are the support vectors.

6.2.2.2 Sample Selection Strategies

The objective of active learning is then to select the samples that can reduce the size of the version space as much as possible. Intuitively to speed up the query procedure, the

instance selected in each iteration should split the current version space into two equal parts. However, this is not practically feasible. It is assumed that the version space is symmetric; hence, the normal vector w^* of the optimal decision surface is often roughly in the center of the version space which is able to separate the version space into two approximately equal parts. Thus, the unlabeled instances that are closest to the current decision surface should be queried as the most informative ones.

Besides that, similarity metrics are also used for sample selection with the assumption that the selected samples should be diverse enough from the currently labeled instances. The most popular approach is to compute the angular diversity between two samples. Another direction is density sampling which aims to select samples from dense unlabeled regions in the feature space under the assumption that the samples in dense regions are more representative than samples from rare regions. In our experiments, we make the first assumption, and the samples are chosen as the support vectors to the current decision surface.

6.2.2.3 Prototype Implementation

Our original prototype was implemented by Blanchart and Ferecatu [2014]. Fig. 6.10 shows the interface of the active learning method without any user interaction, Fig. 6.11 shows the positive and negative training sample selection via mouse operations. Here, green stands for positive training samples, red stands for negative training samples. An improved version makes it an iterative algorithm based on an active learning concept.

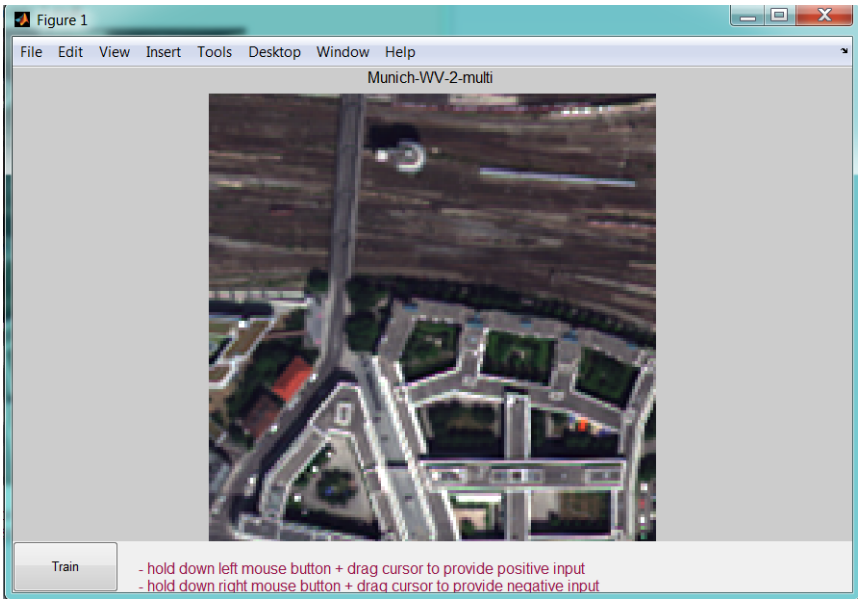


Figure 6.10: The interface of the active learning based object extraction method.

The WorldView-2 multi-spectral satellite image of Munich city which is shown in Fig. 6.12 is used for extracting objects based on a pixel-level non-locality map. In the following experiments, the non-locality map is calculated by using the red, green and blue channel data of the WorldView-2 multi-spectral image.

The most informative and ambiguous points are those support vectors that are close to the classification hyperplane surface, which is shown in the middle column of Fig. 6.13, Fig. 6.15 and the other examples. The white points are the queried pixels that are proposed by

Figure 6.11: Training sample selection of the active learning based object extraction method.

classifier, please note that most of the points lie on the edges of our target objects or false alarm objects which look very similar to target objects. The Matlab color bar 'jet' routine has been used to colorize the non-locality map, where blue stands for "positive", and red means "negative". The results are evaluated either with available reference data based on OpenStreetMap, or by visually checking the original image. In future, region growing and morphological operations can be used to get a better object profile.

6.2.2.4 Evaluation and Discussion

Our approach has two general expectations: one is to explore unknown objects or new categories based on the given training samples, the other is to obtain the exact objects that we are interested in; this usually applies to only one instance.

In the following, the results of extracting different objects are shown for visual evaluation. In order to evaluate the algorithm, the experimental results are divided into three subgroups: **common objects** with several occurrences in the image, **specific objects** with only one occurrence in the image, and **difficult objects** that we discuss for further analysis.

During the iterations of the non-locality map, the added training samples chosen by a user are very critical to obtain the final results; at the same time, the user is guided by the queried informative pixels which are shown in the middle column, and the non-locality map to add training samples.

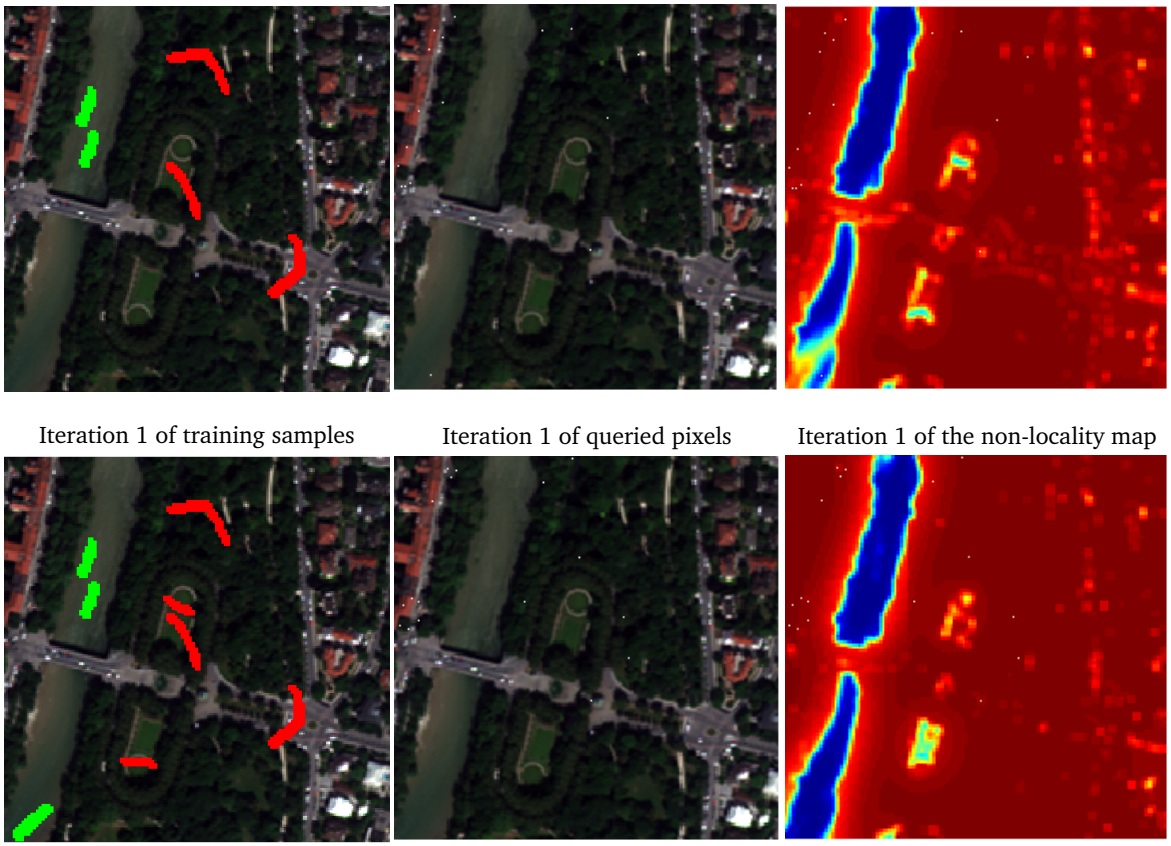
Common Objects: In this sub-section, different common object classes are discussed: railways, a red building, a river, trees, a white square building, and a sports field. Each category contains more than one instance within the test image. However, only the river object will be discussed here, the results of the other objects are contained in Appendix B.

This river example and other common object examples in Appendix B show that the proposed algorithm is able to discover new instances and after 3 to 4 iterations the potential object instances are all extracted. The extracted river results is evaluated by superimposing the OpenStreetMap based water reference. The main water body is extracted, while the

Figure 6.12: WorldView-2 data of Munich city.

bridge is left without creating a false alarm. There is a false negative on the left river bank, the reason might be the bare land rather than the trees-covered land on the river bank.

River case: The *river* extraction is complete when new positive samples are retrieved.



Iteration 1 of training samples | Iteration 1 of queried pixels | Iteration 1 of the non-locality map

Iteration 2 of training samples | Iteration 2 of queried pixels | Iteration 2 of the non-locality map

Figure 6.13: Iterations of training samples and non-locality maps for a *river* object. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

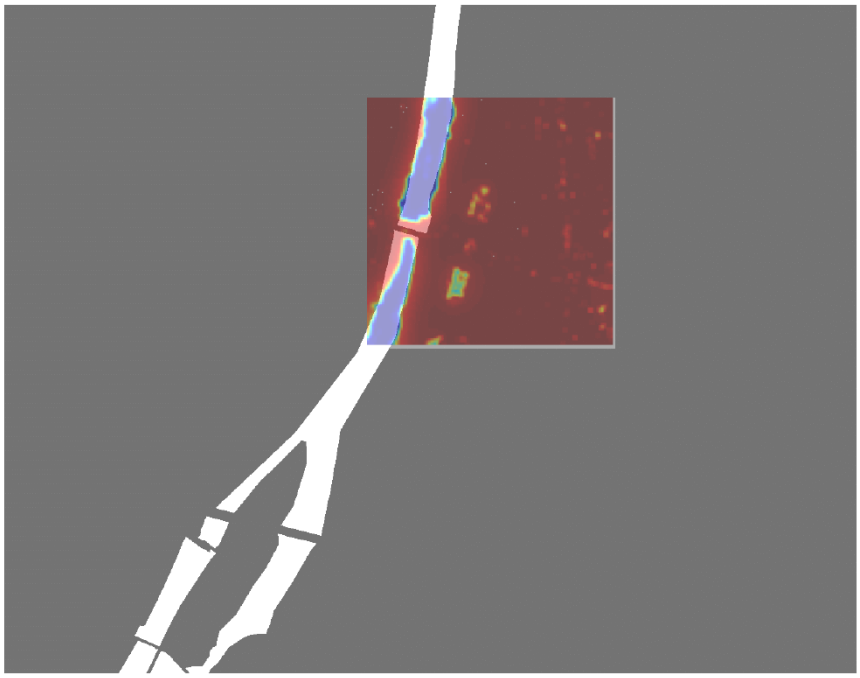
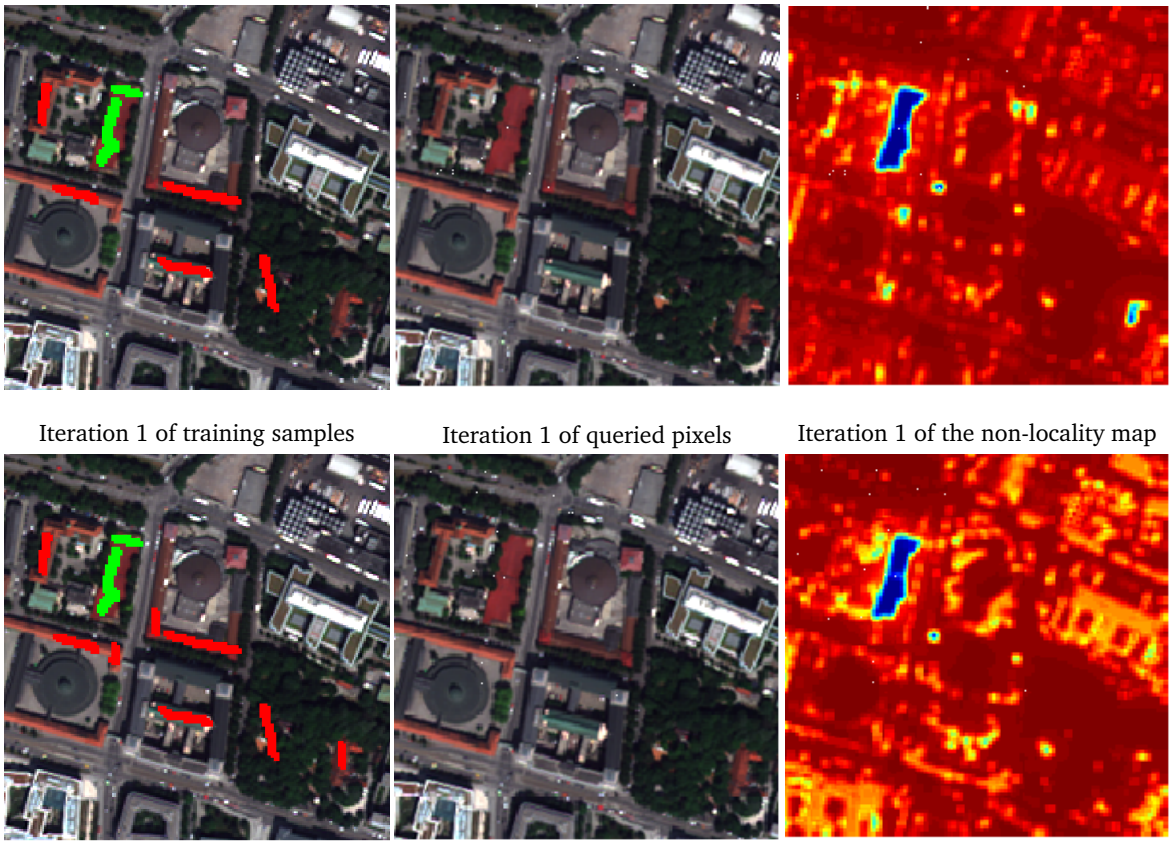


Figure 6.14: Non-locality map of a *river* superimposed on a water reference.

Specific Objects: In this sub-section, extraction of specific objects: such as a bridge, a round-about, a pond, and a tennis court, which all have only one instance within the image, will be discussed. Hence, the user demand is very specific. However, only the tennis court object will be discussed here, the results of the other objects will be shown in Appendix B.

This tennis court example and other specific object examples in Appendix B show that the proposed algorithm is able to discover new instances, and after 3 to 4 iterations the potential object instances are all extracted. Also due to the unique properties of the specific objects, remarkably more negative samples need to be given for an accurate extraction of the objects. The extracted tennis court is evaluated by superimposing of the original optical image. The main tennis court body is extracted correctly.

Tennis court case: The *tennis court* can be extracted when enough negative samples which look similar are given.



Iteration 1 of training samples | Iteration 1 of queried pixels | Iteration 1 of the non-locality map

Iteration 2 of training samples | Iteration 2 of queried pixels | Iteration 2 of the non-locality map

Figure 6.15: Iterations of training samples and non-locality maps for a *tennis court* object. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

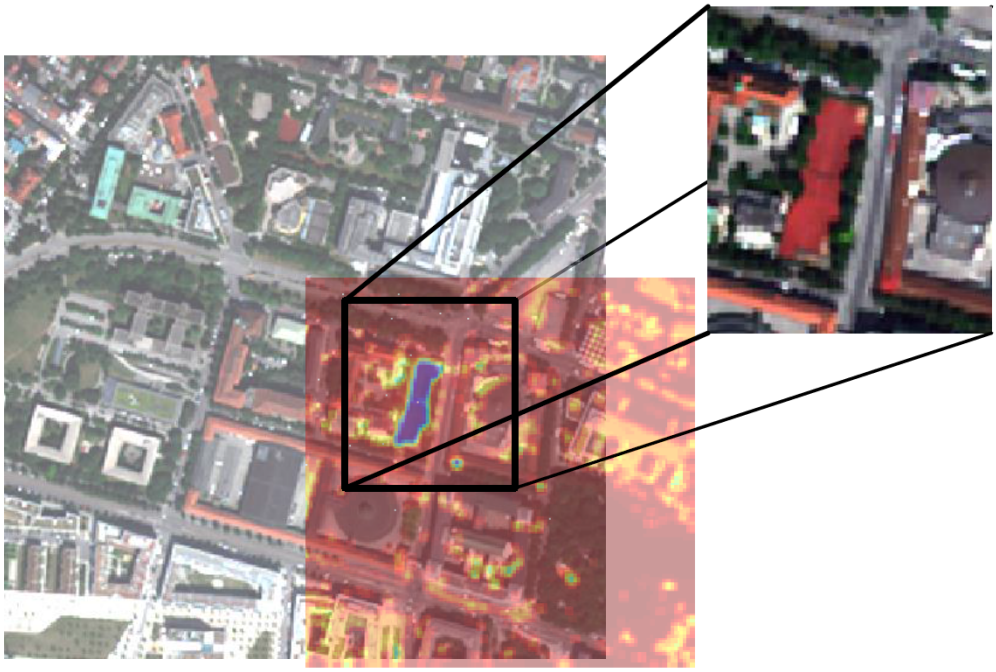


Figure 6.16: Non-locality map of a *tennis court* superimposed by the original image.

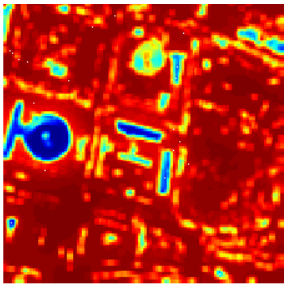
Discussion of Difficult Objects: In this sub-section, difficult cases with specific objects are discussed and analyzed. Particularly, we try to figure out the reasons why the method is inferior or fails to extract the objects.

For very specific objects, advanced features are needed to describe the distinctiveness of the objects. Otherwise, more human input is needed to distinguish between positive and negative, especially the negative samples. The extracted gray round building result is evaluated by superimposing the original optical image. The main building bodies are extracted, while the centers of the buildings are left out as false negatives.

Gray round building case: Lots of training samples are needed to specify this user-selected object, as the given positive training samples are not sufficiently representative for its circular shape. The similarity concept of the non-locality map is not good enough to represent circular shape.

Iteration 1 of training samples

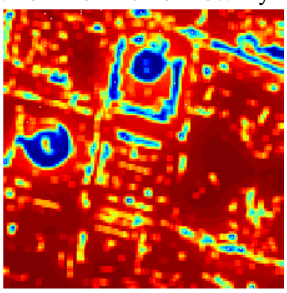
Iteration 1 of queried pixels



Iteration 1 of the non-locality map

Iteration 2 of training samples

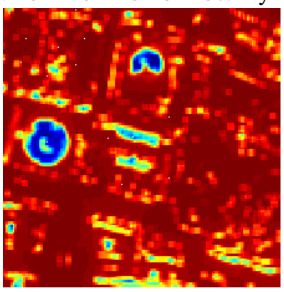
Iteration 2 of queried pixels



Iteration 2 of the non-locality map

Iteration 3 of training samples

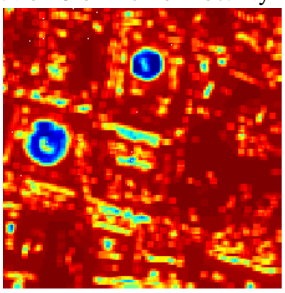
Iteration 3 of queried pixels



Iteration 3 of the non-locality map

Iteration 4 of training samples

Iteration 4 of queried pixels



Iteration 4 of the non-locality map

Figure 6.17: Iterations of training samples and non-locality maps for a *gray round building* object. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

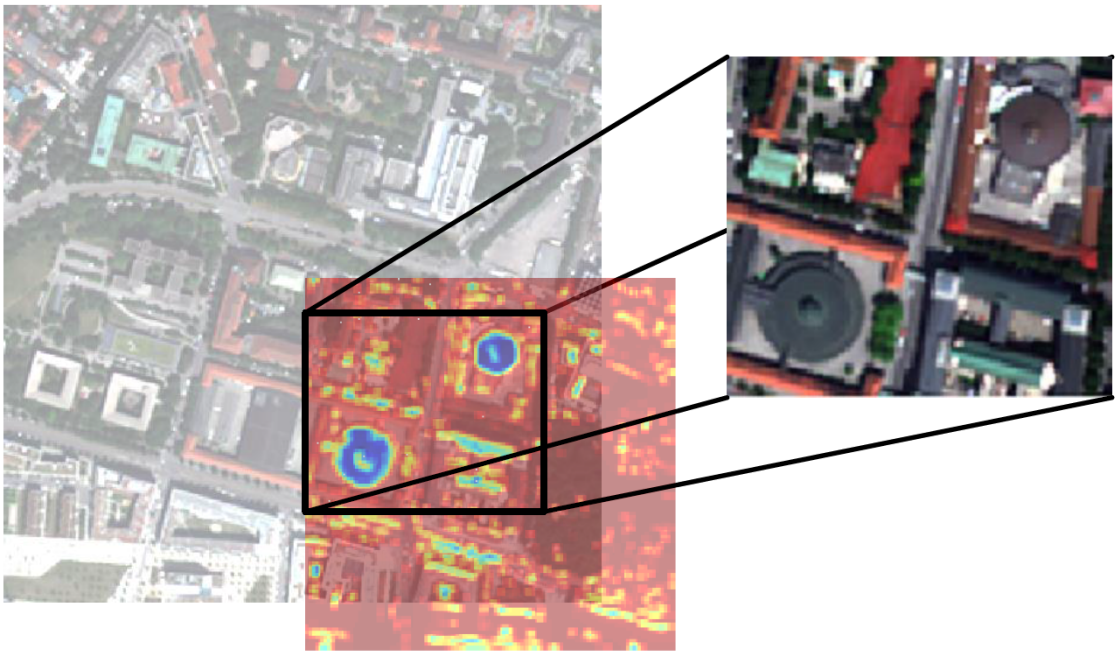
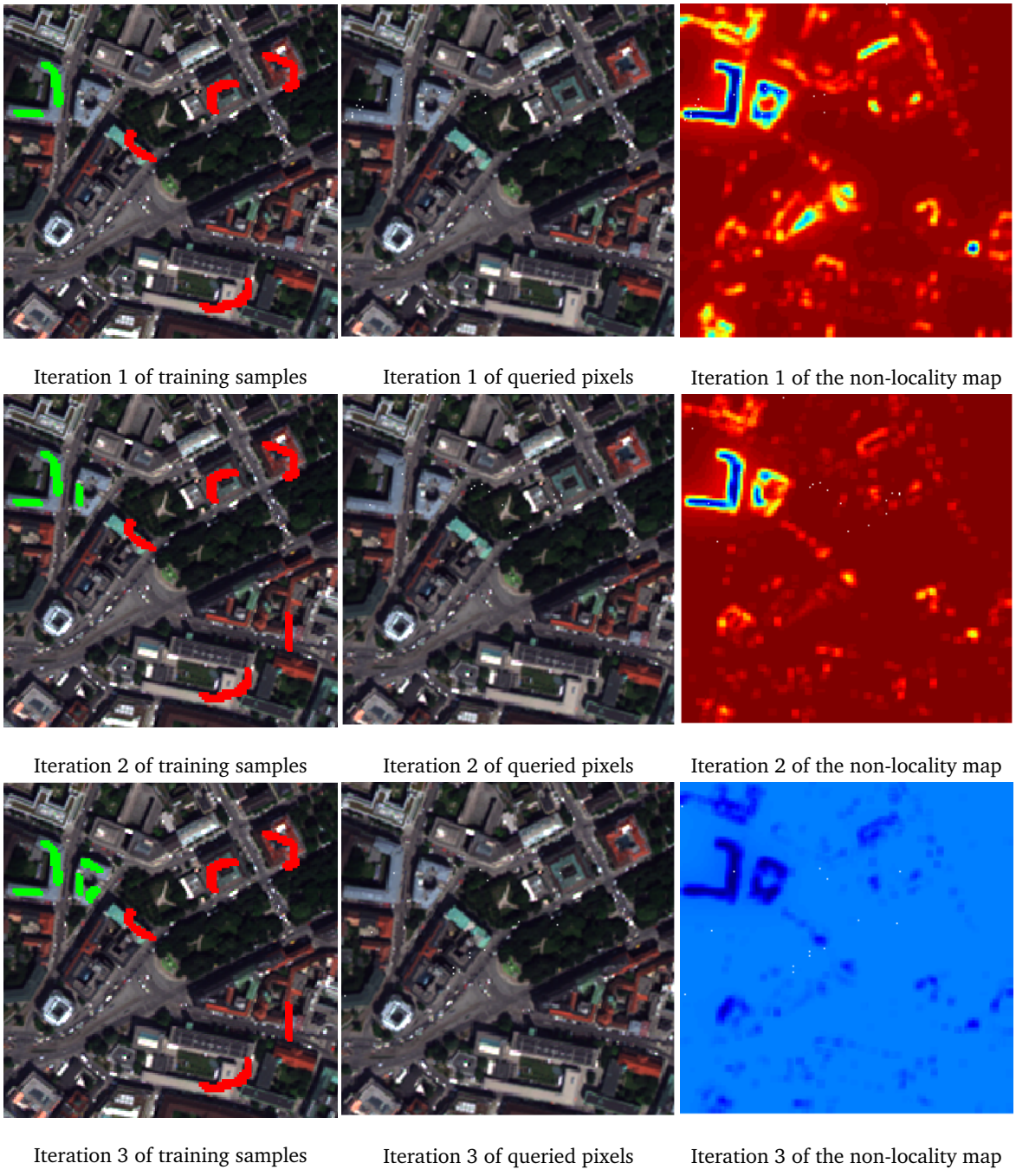


Figure 6.18: Non-locality map of a *gray round building* superimposed by the original image.

Blue corner building case: The non-locality map indicates that the buildings close to the user-selected buildings are similar to the target objects. In the third iteration, when too many samples are given, the classifier generates an inferior result in the end. The reason might be that the calculated features are not able to distinguish between positive and negative training samples.



Iteration 1 of training samples | Iteration 1 of queried pixels | Iteration 1 of the non-locality map

Iteration 2 of training samples | Iteration 2 of queried pixels | Iteration 2 of the non-locality map

Iteration 3 of training samples | Iteration 3 of queried pixels | Iteration 3 of the non-locality map

Figure 6.19: Iterations of training samples and non-locality maps for a *blue corner building* object. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

6.3 Summary

In this chapter, Bayesian rule based pixel classification for SAR imagery and active learning based object extraction for optical imagery have been discussed.

For high-resolution TerraSAR-X imagery, an alternative approach to using the speckle statistics was proposed and evaluated. The $\mathcal{G}^0$ distribution was validated using second-kind statistics for modeling SAR image intensities. For comparison, the Weibull distribution, the $\mathcal{G}^0$ distribution, and a kernel density estimator was used for modeling speckle statistics. To model joint distributions, three bivariate copula functions from the Archimedean family were applied. Based on a copula function, joint probabilities were used for classification by a Bayesian classifier. For urban and forest areas, the $\mathcal{G}^0$ distribution is more appropriate for modeling speckle statistics of SAR images. Regarding the low classification accuracy of water, we conclude that the main reason is the lack of sufficient sample pixels in this category. Furthermore, the urban classification accuracy outperforms the threshold method proposed in Esch et al. [2011]. In the end, the three copulas (Clayton, Frank, and Gumbel) give similar performances.

For high-resolution WorldView-2 imagery, we implemented an active learning concept have been implemented to improve the object extraction which is based on a non-locality map. The support vectors generated by the SVM classifiers during the creation of the non-locality maps are used as the queried pixels in the active learning iterations and are assumed to be informative. We performed various experiments with the purpose to extract various objects. The results show that our adapted method is able to extract common objects via 3 to 4 iterations, with a few more user inputs also specific objects can be extracted via 3 to 4 iterations. However, there are some very specific objects that cannot be extracted reliably what might improved via the use of advanced features.

In general, the pixel-level classification in SAR imagery is suitable for very general semantic categories; on the contrary, object extraction in optical imagery can reach a very detailed level and with relatively good results.

Chapter 7

Conclusions

> Do not go where the path may lead, go instead where there is no path and leave a trail.
>
> Ralph Waldo Emerson

7.1 Summary

In this dissertation, we have provided solutions for high-resolution satellite image information extraction and content interpretation using TerraSAR-X and WorldView-2 data. As an image scene is usually of big size with complicated contents, in order to reduce the processing time and the complexity of problem, we remote sensing researchers often cut the whole scenes into patches as a pre-processing step in order to reduce the processing time and the complexity of problem. However, as so much information and interesting details become visible in high-resolution imagery, it is not satisfactory to stop at patch level processing. Hence, processing and analyzing on pixel level is needed. Due to the large amount of data, we start the analysis by using an unsupervised learning methods, in the hope of simplifying our interpretation task to group similar patches together at the first step. However, as known, the accuracy of unsupervised learning methods is insufficient to obtain satisfactory semantic annotation results; therefore, supervised learning methods are applied to those grouped data clusters. In the sense of pixel level processing, different methods have been proposed for different data sources: speckle statistics feature and image intensities are considered for SAR data to obtain joint probability distributions; similarity and closeness have been considered to extract objects based on active learning concepts for optical data.

We presented **a hierarchical patch clustering method** for high-resolution remote sensing images:

- As evaluated by internal cluster and external cluster indices as well as patch visualization, the Gaussian-test-based hierarchical patch clustering method is able to obtain homogeneous clusters. A Gaussian goodness test (Anderson-Darling test) is used to split feature vector sets into homogeneous clusters which largely reduces the clustering time and also preserves the homogeneity. With a minumum cluster size threshold, our modified G-means algorithm is faster than the original algorithm.

- However, for different datasets, external results show that the human defined semantic labels influence the homogeneity level evaluation, which indicates that some human subjective factor is involved in the process of semantic annotation.
- Moreover, fractional and Minkowski distance metrics have been analysed; it turns out that a distance parameter ranging from 1.2 to 2 performs best. Different feature descriptors have been evaluated during the experiments, the results show that classic Gabor texture features perform better than the BoW features. By comparing with different clustering methods, our results show that G-means can obtain relatively homogeneous clusters; the cluster homogeneity is slightly degraded compared to k-means because of the repeated dichotomous cluster splitting.

Then, **a semi-supervised classification scheme for image patches** has been formulated within the framework of a hierarchical clustering method. Large-scale datasets have been used, each with more than thousands of image patches and two-level semantic annotations: these serve as general and detailed level reference data. It turns out that our proposed method is able to obtain reliable results for the general level reference data; however, due to the too many detailed sub-classes and their few instances, the proposed method generates inferior results for the detailed level reference data.

- **Concerning classifiers**: For the classification accuracy and F-score value, the performances of different applied supervised learning methods (i.e., SVM, KNN, and NBNN) were investigated.
  - As for the overall classification accuracy, SVM always achieves the best results both for general and detailed level reference data. In the case of the F-score value, the nearest neighbor methods (i.e., KNN and NBNN) perform better than SVM. The reason behind it is that SVM obtains better classification accuracies for some dominant classes, but lower classification accuracies for the other minority classes.
  - When the number of classes within a cluster is increasing (e.g., semi-supervised NBNN compared to semi-supervised KNN, for F-score values of detailed level reference data), NBNN performs better than KNN. This indicates that NBNN tends to provide good results when we have a limited number of samples in each class.
- **Concerning semi-supervised learning and manually annotated reference data**: After supervised classification within clusters, the relationships between the unsupervised clusters and the general and detailed level reference data were discussed:
  - When we look at the relations between clustering results and general level reference data, and the relations between clustering results and detailed level reference data, usually a cluster comprises 5 to 6 classes, with one or two of them being dominant. Of course, the correspondence is better for the general level reference data due to their lower number of classes.
  - With a good classifier, each patch is labeled correctly with a probability of 0.6 for the general level reference data, and with a probability of 0.4 for the detailed level reference data.

Regarding pixel-level segmentation and object extraction, a method based on the concept of Bayes' rule and joint probability densities has been analyzed for SAR data.

Moreover, an active learning concept was implemented with the idea of non-locality for various detailed object extractions in high-resolution optical imagery. In general, the pixel-level classification in SAR imagery is suitable for very general semantic categories; on the contrary, object extraction in optical imagery can reach very detailed levels and with relatively good results.

- **For high resolution TerraSAR-X imagery**, considering image intensities and speckle statistics feature, three copula functions have been studied to model the joint probability density:
  - The urban classification accuracy outperforms the threshold method proposed in Esch et al. [2011].
  - Concerning urban and forest areas, the $\mathcal{G}^0$ distribution is more appropriate for modeling the speckle statistics feature of SAR images. Regarding the low classification accuracy of water, the main reason could be the lack of sufficient sample pixels in this category.
  - The three copulas (Clayton, Frank, and Gumbel) give similar performances.
- **For high resolution WorldView-2 imagery**, an active learning concept has been implemented to improve the object extraction method which is based on a non-locality map.
  - Various experiments were performed with the purpose to extract various objects. The results show that the modified method is able to extract common objects via 3 to 4 iterations. With a few more user inputs, specific objects can also be extracted via 3 to 4 iterations.

7.2 Future Works

This dissertation has put efforts in solving high-resolution satellite image interpretation on patch level and pixel level, by using semi-supervised learning, active learning, and Bayes' rule. The summary above demonstrates the attained semantic annotations for different data sources. However, there are still some open questions within this dissertation: the optimal distance metrics in the processing of SAR data, the local and global minimum of G-means clustering, better histogram fitting in modelling different land covers in SAR data, and the possibilities of more detailed semantic class extractions via active learning. Hence, as shown in Fig. 7.1, more detailed semantic categories encourage us to develop still more advanced semantic annotations and object recognition algorithms.

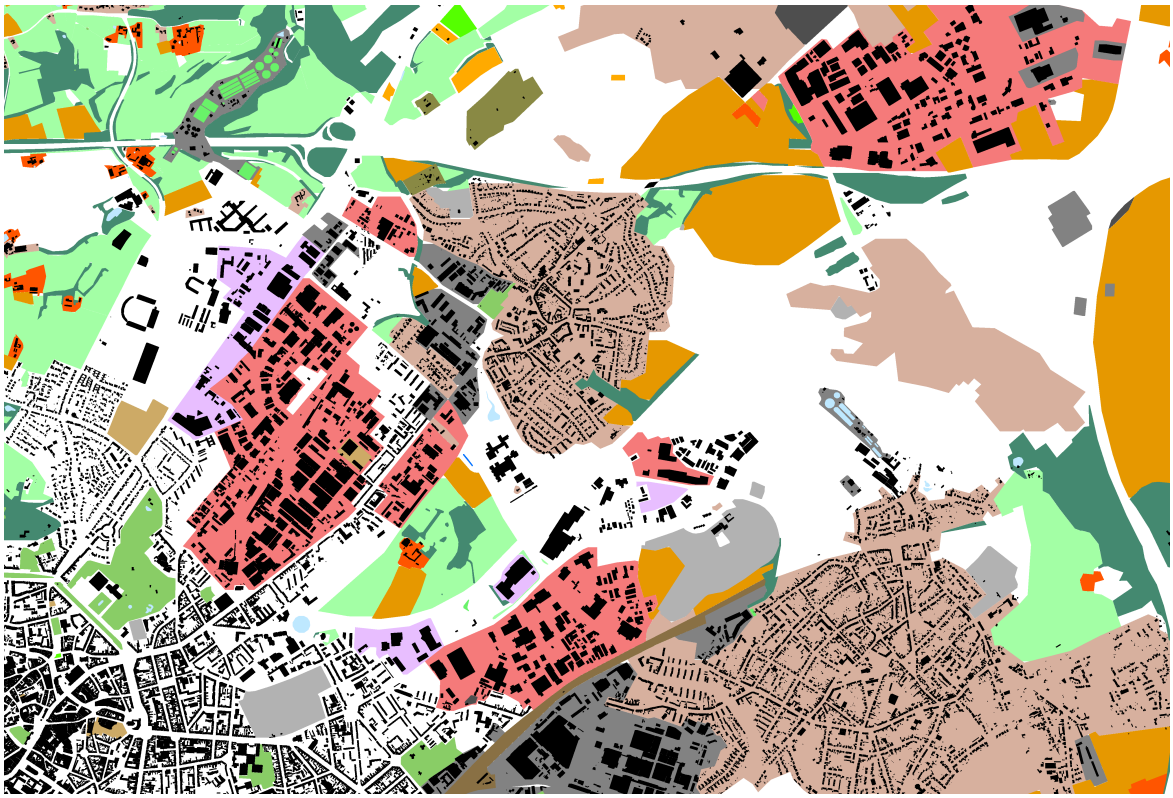


Figure 7.1: Detailed level openstreetmap ground truth of Aachen city.

Appendix A

A.1 Summary of Machine Learning Algorithms

Table A.1: Overview of classic machine learning algorithms.

Algorithm	**Description**	**Model**	**Objective**	**Training Algorithm**
K-Means	A hard-margin, geometric **clustering** algorithm, where each data point is assigned to its closest centroid.	Hard assignments $$r_{nk} \in 0,1 \; s.t. \; \forall n \sum_k r_{nk} = 1, \quad (1)$$ i.e., each data point is assigned to one cluster k. The geometric distance is the l^2 norm Euclidean distance: $$\|x_{ni} - \mu_{ki}\|_2 = \sqrt{\sum_{i=1}^{D}(x_{ni} - \mu_{ki})^2}. \quad (2)$$	$$\arg\min_{r,\mu} \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|x_n - \mu_k\|_2^2, \quad (3)$$ ...i.e. minimize the distance from each cluster center to each of its points.	**E**xpectation: $$r_{nk} = \begin{cases} 1 & if \|x_n - \mu_k\|^2 \; minimal for \; k, \\ 0 & o/w. \end{cases}$$ **M**aximisation: $$\mu_{MLE}^{(k)} = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}, \quad (4)$$ ...where $\mu^{(k)}$ is the centroid of cluster k.
Mean Shift Clustering	A non-parametric **clustering** algorithm for locating the maxima of a density function (mode finding procedure).	Starting on the data points, find the stationary points of the density function. Then prune these points by retaining only the local maxima. The set of all locations that converge to the same mode defines the basin of attraction of that mode. The points which are in the same basin of attraction are associated with the same cluster. Given n data points $x_i, i = 1, ..., n$ on a d-dimensional space R^d, the multivariate kernel density estimate obtained with kernel $K(x)$ and window radius h is: $$f(x) = \frac{1}{nh^d}\sum_{i=1}^{n} K(\frac{(x - x_i)}{h}),$$ $$K(x) = c_{k,d}k(\|x\|^2). \quad (5)$$	The gradient of the density estimator is $$\nabla f(x) = \frac{2c_{k,d}}{nh^{d+2}}\sum_{i=1}^{n}(x_i - x)g(\|\frac{x - x_i}{h}\|^2)$$ $$= \frac{2c_{k,d}}{nh^{d+2}}[\sum_{i=1}^{n} g(\|\frac{x - x_i}{h}\|^2)]* \quad (6)$$ $$[\frac{\sum_{i=1}^{n} x_i g(\|\frac{x-x_i}{h}\|^2)}{\sum_{i=1}^{n} g(\|\frac{x-x_i}{h}\|^2)} - x],$$ ...where $g(x) = k'(s)$. The second term is the mean shift $$m_h(x) = \frac{\sum_{i=1}^{n} x_i g(\|\frac{x-x_i}{h}\|^2)}{\sum_{i=1}^{n} g(\|\frac{x-x_i}{h}\|^2)} - x. \quad (7)$$	The mean shift procedure, obtained by successive • computation of the mean shift vector $m_h(x^t)$, • translation of the window $x^{t+1} = x^t + m_h(x^t)$.
Gaussian Mixture Model	A probabilistic **clustering** algorithm, where clusters are modeled as latent Gaussians, and each data point is assigned the probability of being drawn from a particular Gaussian.	Assignments to clusters by specifying probabilities: $$p(x^{(i)}, z^{(i)}) = p(x^{(i)}|z^{(i)})p(z^{(i)}), \quad (8)$$ ...with $z^{(i)}$ follows a multinomial(γ) distribution, and $$\gamma_{nk} \equiv p(k|x_n) s.t. \sum_{j=1}^{k} \gamma_{nj} = 1, \quad (9)$$ i.e., maximize the probability of the observed data x.	$$\mathcal{L}(x, \pi, \mu, \sum) = \log p(x|\pi, \mu, \sum)$$ $$= \sum_{n=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}_k(x_n|\mu_k, \sum_k)\right). \quad (10)$$	**E**xpectation: For each n, k set: $$\gamma_{nk} = p(z^{(i)} = k|x^{(i)}; \gamma, \mu, \sum) \; (= p(k|k_n))$$ $$= \frac{p(x^i|z^{(i)} = k; \mu, \sum)p(z^{(i)} = k; \pi)}{\sum_{j=1}^{K} p(x^{(i)}|z^{(i)} = l; \mu, \sum)p(z^{(i)} = l; \pi)} \quad (11)$$ $$= \frac{\pi_k \mathcal{N}(x_n|\mu_k, \sum_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \sum_j)}.$$ **M**aximisation: $$\sum_k = \frac{\sum_{n=1}^{N} \gamma_{nk}(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma_{nk}},$$ $$\mu_k = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}}, \; \pi_k = \frac{1}{N}\sum_{n=1}^{N} \gamma_{nk}. \quad (12)$$

Table A.2: Overview of classic machine learning algorithms (continued)

Algorithm	**Description**	**Model**	**Objective**	**Training Algorithm**
Logistic Regression	A **classification** algorithm, which predicts a dichotomous outcome like ordinary least squares regression.	$$p(y=\pm 1\|x,w)=\sigma(yw^Tx) = \frac{1}{1+\exp(-yw^Tx)}, \quad (13)$$ used for binary classification or for predicting the certainty of a binary outcome. $\sigma(x)$ is the logistic function which is defined as: $$\sigma(x)=\frac{1}{1+\exp(x)} \quad (14)$$	$$\arg\max_w = -\sum_{i=1}^{n}\log\Big(1+\exp\Big(-y_i w^T x_i\Big)\Big), \quad (15)$$ ...i.e. find the parameter vector w which maximizes the log-likelihood.	Newton's method based on iteratively reweighted least squares algorithms: $$\mathbf{w}_{new}=\mathbf{w}_{old}+ (\mathbf{XAX}^T)^{-1}\sum_i(1-\sigma(y_i\mathbf{w}^T\mathbf{x}_i))y_i\mathbf{x}_i. \quad (16)$$ We define $$z_i=\mathbf{x}_i^T\mathbf{w}_{old}+\frac{1-\sigma(y_i\mathbf{w}^T\mathbf{x}_i)y_i}{a_{ii}}. \quad (17)$$
Naive Bayes Classifier	A **classification** algorithm which learns $p(C_k\|x)$ by modeling $p(x\|C_k)$ and $p(C_k)$, using Bayes' rule to infer the class conditional probability. It assumes each feature independent of all others, therefore 'Naive'.	$$y(x)=\arg\max_k p(C_k\|x) = \arg\max_k p(x\|C_k)*p(C_k) = \arg\max_k \prod_{i=1}^{D} p(x_i\|C_k)*p(C_k) = \arg\max_k \sum_{i=1}^{D}\log(p(x_i\|C_k))+\log(p(C_k)). \quad (18)$$	No optimisation needed.	**Multivariate likelihood** $$p(x\|C_k)=\sum_{i=1}^{D}\log(p(x_i\|C_k)),$$ $$p_{MLE}(x_i=v\|C_k)= \frac{\sum_{j=1}^{N}\sigma(t_j=C_k\Sigma x_{ji}=v)}{\sum_{j=1}^{N}\sigma(tj=C_k)}. \quad (19)$$ **Multinomial likelihood which is used in LDA** $$p(x\|C_k)=\prod_{i=1}^{D}p(word_i\|C_k)^{x_i}. \quad (20)$$
K Nearest Neighbors	A simple **classification** algorithm, in which the label of a new point $\hat{x}$ is classified with the most frequent label $\hat{t}$ of the k nearest training instances.	$$\hat{t}=\arg\max_C \sum_{i:x_i\in N_k(x,\hat{x})}\delta(t_i,C),$$ $N_k(\mathbf{x},\hat{x})\leftarrow k$ points in **x** closest to $\hat{x}$, Euclidean distance formula: $\sqrt{\sum_{t=1}^{D}(x_i-\hat{x_i})^2}$, $\delta(a,b)\leftarrow 1$ if a=b; 0 otherwise. (21)	No optimisation needed.	Use cross-validation to learn the appropriate k; otherwise no training, classification based on existing points.

Table A.3: Overview of classic machine learning algorithms (continued).

Algorithm	**Description**	**Model**	**Objective**	**Training Algorithm**
Perceptron	A **classification** algorithm which directly estimates the linear function $y(x)$ by iteratively updating the weight vector when incorrectly classifying a training instance.	Binary, linear classifier: $$y(x) = sign(\mathbf{w}^T x), \quad (22)$$ ...where: $$sign(x) = \begin{cases} +1 & if\ x > 0, \\ -1 & if\ x < 0. \end{cases}$$ Multiclass perceptron: $$y(x) = \arg\max_{C_k} \mathbf{w}^T \phi(x, C_k). \quad (23)$$	Tries to minimise the error function (the number of incorrectly classified input vectors): $$\arg\min_{\mathbf{w}} Ep(\mathbf{w}) = \arg\min_{\mathbf{w}} - \sum_{n \in \mathcal{M}} \mathbf{w}^T x_n t_n, \quad (24)$$ ...where $\mathcal{M}$ is the set of misclassified training vectors.	Iterate over each training example x_n, and update the weight vector if misclassification occurs: $$\begin{aligned} \mathbf{w}^{i+1} &= \mathbf{w}^i + \eta \Delta Ep(\mathbf{w}) \\ &= \mathbf{w}^i + \eta x_n t_n, \end{aligned} \quad (25)$$ ...where typically $\eta = 1$. For the multiclass perceptron: $$\mathbf{w}^{i+1} = \mathbf{w}^i + \phi(x, t) - \phi(x, y(x)). \quad (26)$$
Convolutional Neural Network	A problem of determining the network weights to approximate a specific target mapping g, which makes it a **classification** algorithm.	A (L+1)-layer perceptron, consists of D input units, C output units, and several so called hidden units. The i^{th} unit within layer l computes the output $$\begin{aligned} y_i^{(l)} &= f(z_i^{(l)}) \text{ with} \\ z_i^{(l)} &= \sum_{k=1}^{m^{(l-1)}} w_{i,k}^{(l)} y_k^{(l-1)} + w_{i,0}^{(k)}, \end{aligned} \quad (27)$$ ...where $w_{i,k}^{(l)}$ denotes the weighted connection from the k^{th} unit in layer $(l-1)$ to the i^{th} unit in layer l, and $w_{i,0}^{(l)}$ can be regarded as external input to the unit and is referred to as bias. $m^{(l)}$ denotes the number of units in layer l, such that $D = m^{(0)}$ and $C = m^{(L+1)}$	Minimise a chosen objective function which can be interpreted as an error measure between the network output $y(x_n)$ and the desired target output t_n. The sum-of-squared error measures is given by: $$\begin{aligned} E1(w) &= \sum_{n=1}^{N} E1_n(w) \\ &= \sum_{n=1}^{N} \sum_{k=1}^{C} (y_k(x_n, w) - t_{n,k})^2, \end{aligned} \quad (28)$$ and the cross-entropy error measure is given by: $$\begin{aligned} E2(w) &= \sum_{n=1}^{N} E2_n(w) \\ &= \sum_{n=1}^{N} \sum_{k=1}^{C} t_{n,k} \log(y_k(x_n, w)), \end{aligned} \quad (29)$$ ...where $t_{n,k}$ is the k^{th} entry of the target value t_n.	**Stochastic training:** An input value is chosen at random and the network weights are updated based on the error $E_n(w)$. **Batch training:** All input values are processed and the weights are updated based on the overall error $E(w) = \sum_{n=1}^{N} E_n(w)$. **Online training:** Every input value is processed only once and the weights are updated using the error $E_n(w)$. **Mini-batch training:** A random subset $M \subseteq 1, ..., N$ (mini-batch) of the training set is processed and the weights are updated based on the cumulative error $E_M(w) := \sum_{n \in M} E_n(w)$.

Table A.4: Overview of classic machine learning algorithms (continued).

Algorithm	**Description**	**Model**	**Objective**	**Training Algorithm**
Decision Trees	A hierarchical piecewise **classifier** for splitting complex problems into a hierarchy of simpler ones.	Features are thought of as being randomly sampled from the set of all possible features, with a function $\phi(v)$ selecting a subset of the features of interest. $$\phi : \mathcal{R}^d \rightarrow \mathcal{R}^{d'}, with\ d' << d, \quad (30)$$ ...where d is the feature dimensionality, and $\phi(\mathbf{v})$ is a function which selects a subset of features of interest.	The optimal parameters $\boldsymbol{\theta}$ of the j^{th} split node need to be computed, which is done here by maximizing an information gain objective function: $$\boldsymbol{\theta}_j^* = \arg\max_{\boldsymbol{\theta}_j} I_j, \quad (31)$$ with $$I_j = I(S_j, S_j^L, S_j^R, \theta_j), \quad (32)$$ ...where the symbols S_j, S_j^L, S_j^R denote the sets of training points before and after the split.	The gain of information achieved by splitting the data is computed as: $$I = H(S) - \sum_{i \in 1,2} \frac{\lvert S^i \rvert}{\lvert S \rvert} H(S^i), \quad (33)$$ the differential entropy of a d-variate Gaussian density is defined as: $$H(S) = \frac{1}{2} \log\Big((2\pi)^d \lvert \Lambda(\mathcal{S}) \rvert\Big). \quad (34)$$ Here, $\Lambda(\mathcal{S})$ stands for the gradient of S.
Random Forests	A **classification** algorithm which is an ensemble of randomly trained decision tree methods.	Each split node j is associated with a binary split function: $$h(\boldsymbol{v}, \boldsymbol{\theta}_j) \in 0, 1. \quad (35)$$ The ensemble model is obtained via combining all tree predictions into a single forest prediction by a simple averaging operation: $$p(c\lvert\boldsymbol{v}) = \frac{1}{T} \sum_{t=1}^{T} p_t(c\lvert\boldsymbol{v}). \quad (36)$$ Alternatively, one could also multiply the tree output together (though the trees are not statistically independent) $$p(c\lvert\boldsymbol{v}) = \frac{1}{Z} \prod_{t=1}^{T} p_t(c\lvert\boldsymbol{v}). \quad (37)$$	The optimal parameters $\boldsymbol{\theta}$ of the j^{th} split node need to be computed, which is done here by maximizing an information gain objective function: $$\boldsymbol{\theta}_j^* = \arg\max_{\boldsymbol{\theta}_j \in \mathcal{T}_j} I_j, \quad (38)$$ with $$I_j = I(S_j, S_j^L, S_j^R, \theta_j), \quad (39)$$ ...where the symbols S_j, S_j^L, S_j^R denote the sets of training points before and after the split.	The gain of information achieved by splitting the data is computed as: $$I_j = H(S_j) - \sum_{i \in L,R} \frac{\lvert S_j^i \rvert}{\lvert S_j \rvert} H(S_j^i), \quad (40)$$ ...where the Shannon entropy is defined mathematically as: $$H(S) = -\sum_{c \in C} p(c) \log p(c). \quad (41)$$
Support Vector Machine	A maximum margin **classifier**: finds the separating hyperplane with the maximum margin to its closest data points.	$$y(x) = \sum_{n=1}^{N} \lambda_n t_n x^T x_n + w_0. \quad (42)$$	**Primal** $$\arg\min_{\mathbf{w}, w_0} \frac{1}{2} \lVert \mathbf{w} \rVert^2, \quad s.t. t_n(\mathbf{w}^T x_n + w_0) \geq 1\ \forall n. \quad (43)$$ **Dual** $$\hat{\mathcal{L}}(\bigwedge) = \sum_{n=1}^{N} \lambda_n - \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n \lambda_m t_n t_m x_n^T x_m, \quad s.t. \lambda_n \geq 0, \sum_{n=1}^{N} \lambda_n t_n = 0,\ \forall n. \quad (44)$$	• Quadratic Programming (QP) • SMO, Sequential Minimal Optimisation (chunking).

A.2 Support Vector Machines (SVMs)

An SVM can be used to solve tasks such as linearly separable binary classifications, non-linearly separable binary classifications, data regressions, non-linearly separable classifications or regression problems. Since in most cases, our real world problems are non-linear but separable, not only the basic support vector machines for linearly separable binary classifications, but also for non-linearly separable problems will be explained in this section.

A.2.1 Linearly Separable Binary Classifications

We have L training points, where each input $\mathbf{x}_i$ has D attributes (i.e., with D dimensions) and belongs to one of two classes y_i=-1 or +1:

$$\mathbf{x}_i, y_i \quad where \quad i = 1...L,\ y_i \in -1, 1,\ \mathbf{x} \in \mathcal{R}^D. \tag{45}$$

We assume there is a hyperplane which can separate the two classes. This hyperplane can be described by $w \cdot x + b = 0$ where w is normal to the hyperplane; $\frac{b}{\|w\|}$ is the perpendicular distance from the origin to the hyperplane.

Support Vectors are the training points closest to the separating hyperplane, and the aim of an SVM is to orientate this hyperplane to be as far as possible from the closest members of both classes, which is also called as **maximize the minimum margin**.

Referring to Fig. A.1, an SVM needs to select the variables w and b so that our training data can be described by:

$$\mathbf{x}_i \cdot w + b \geq +1 \quad for \quad y_i = +1 \tag{46}$$

$$\mathbf{x}_i \cdot w + b \leq -1 \quad for \quad y_i = -1 \tag{47}$$

These equations can be combined into:

$$y_i(\mathbf{x}_i \cdot w + b) - 1 \geq 0 \quad \forall \tag{48}$$

We now consider the points that lie closest to the separating hyperplane, i.e. the Support Vectors. Then the two planes H_1 and H_2 that these points lie on can be described by:

$$\mathbf{x}_i \cdot w + b = +1 \quad for \quad H_1 \tag{49}$$

$$\mathbf{x}_i \cdot w + b = -1 \quad for \quad H_2 \tag{50}$$

The hyperplane's equidistance from H_1 to H_2 is known as the SVM $margin$. The margin is to be maximized in order to orientate the hyperplane H to be as far away from the Support Vectors as possible.

Vector geometry shows that the margin is equal to $\frac{1}{\|w\|}$, maximizing it and subjecting to constraint 48 is equivalent to find:

$$min\|w\| \quad \text{such that} \quad y_i(\mathbf{x}_i \cdot w + b) - 1 \geq 0 \quad \forall_i \tag{51}$$

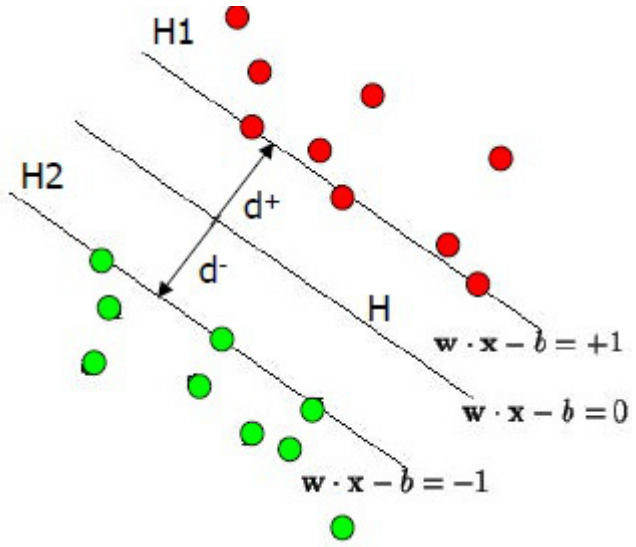


Figure A.1: The separating hyperplane of a Support Vector Machine.

Minimizing $\|w\|$ is equivalent to minimizing $\frac{1}{2}\|w\|^2$; this term enables us to perform a Quadratic Programming (QP) optimization later on. Therefore, we need to find:

$$min\frac{1}{2}\|w\|^2 \quad \text{s.t.} \quad y_i(\mathbf{x}_i \cdot w + b) - 1 \geq 0 \quad \forall_i \tag{52}$$

When caring for the constraints in this minimization, Lagrange multipliers α have to be assigned, where $\alpha_i \geq 0 \; \forall_i$:

$$L_P \equiv \frac{1}{2}\|w\|^2 - \alpha[y_i(\mathbf{x}_i \cdot w + b) - 1 \; \forall_i] \tag{53}$$

$$\equiv \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L} \alpha_i y_i(\mathbf{x}_i \cdot w + b) + \sum_{i=1}^{L} \alpha_i \tag{54}$$

By differentiating L_P with respect to w and b and setting the derivatives to zero we obtain:

$$\frac{\partial L_P}{\partial w} = 0 \; \Rightarrow \; w = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x}_i \tag{55}$$

$$\frac{\partial L_P}{\partial b} = 0 \; \Rightarrow \; \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{56}$$

So we need to maximize:

$$L_D \equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \; s.t. \; \alpha_i \geq 0 \; \forall, \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{57}$$

$$\equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T H\alpha \; s.t. \; \alpha_i \geq 0 \; \forall, \; \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{58}$$

The new formulation L_D is referred as the *Dual form of the Primary* L_P. It is worth to note that the Dual form requires only dot product values of each input vector $\mathbf{x}_i$, which is known as the *Kernel Trick*. Hence, instead of minimizing L_P, we need to maximize L_D:

$$\max_{\alpha}[\sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T H\alpha] \quad s.t. \quad \alpha_i \geq 0 \quad \forall \quad and \quad \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{59}$$

A QP solver is used to obtain α and w for this convex quadratic optimization problem. Only parameter b then remains to be calculated.

Any data point satisfying 56 is a support vector $\mathbf{x}_s$ that will have the following form:

$$y_s(\mathbf{x}_s \cdot w + b) = 1 \tag{60}$$

Substituting in 55 yields:

$$y_s(\sum_{m \in S} \alpha_m y_m \mathbf{x}_m \cdot \mathbf{x}_s + b) = 1 \tag{61}$$

where S denotes the set of indices of support vectors, which is determined when $\alpha_i > 0$. Multiplication by y_s and use of $y_s^2 = 1$ leads to:

$$y_s^2(\sum_{m \in S} a_m y + mx + m \cdot \mathbf{x}_s + b) = y_s \tag{62}$$

$$b = y_s - \sum_{m \in S} a_m y_m \mathbf{x}_m \cdot \mathbf{x}_s \tag{63}$$

Instead of using an arbitrary support vector $\mathbf{x}_s$, b is calculated by taking an average over all of the support vectors in S:

$$b = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{m \in S} \alpha_m y_m \mathbf{x}_m \cdot \mathbf{x}_s) \tag{64}$$

With the variables w and b that define the optimal orientation of the separating hyperplane, the support vector machine is then defined Fletcher [2009].

A.2.2 Non-Linearly Separable Classifications

In this case, create a matrix H from the dot product of our input variables:

$$H_{ij} = y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j = \mathbf{x}_i^T \mathbf{x}_j \tag{65}$$

where $k(\mathbf{x}_i, \mathbf{x}_j)$ is an example of a family of functions called *Kernel Functions* ($k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$ is known as a *Linear Kernel*). Kernel functions are all based on calculating inner products of two vectors. If these functions can be recast into a higher dimensionality space by some potentially non-linear feature mapping functions $\mathbf{x} \rightarrow \psi(\mathbf{x})$, only inner products of the mapped inputs in the feature space need to be determined rather than to explicitly calculate ψ. This is also called *Kernel Trick*; it is useful for non-linearly separable or regressable problems when a suitable mapping is given to map the input vectors $\mathbf{x}$ to a higher dimensionality feature space Burges [1998]. The most commonly used kernels are listed in Table A.5:

Table A.5: Commonly used kernel functions.

Kernel	**Function Formula**
Radial Basis Kernel	$k(\mathbf{x}_i, \mathbf{x}_j) = exp^{-(\frac{(\|\mathbf{x}_i - \mathbf{x}_j\|)^2}{2\sigma^2})}$
Polynomial Kernel	$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + a)^b$
Sigmoid Kernel	$tanh(a\mathbf{x}_i \cdot \mathbf{x}_j - b)$

A.2.3 Application

Regarding the classification problem for non-linearly separable data, we can obtain a support vector machine via following steps Fletcher [2009]:

- Create H, where $H_ij = y_i y_j \psi(\mathbf{x}_i) \cdot \psi(\mathbf{x}_j)$.
- Depending on how significantly misclassifications should be treated, a suitable threshold value of parameter C is selected.
- Find α so that

$$\sum_{i=1}^{L} \alpha_i - \frac{1}{2} \alpha^T H \alpha \tag{66}$$

  is maximized, subject to the constraints

$$0 \leq \alpha_i \leq C \quad \forall_i \quad and \quad \sum_{i=1}^{L} \alpha_i y_i = 0. \tag{67}$$

  This is solve via a quadratic programming (QP) solver.
- Calculate $w = \sum_{i=1}^{L} \alpha_i y_i \psi(\mathbf{x}_i)$.
- Determine the set of support vectors S by finding the indices such that $0 < \alpha_i \leq C$.
- Calculate $b = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{m \in S} \alpha_m y_m \psi(\mathbf{x}_m) \cdot \psi(\mathbf{x}_s))$.
- Each new point x' is then classified by evaluating $y' = sgn(w \cdot \psi(\mathbf{x}') + b)$.

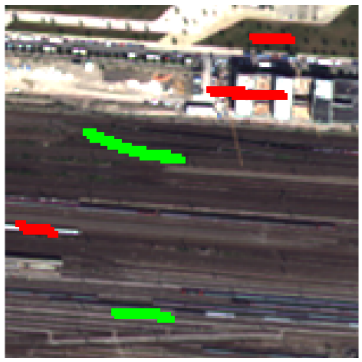
Appendix B

In this appendix, more experimental results are reported.

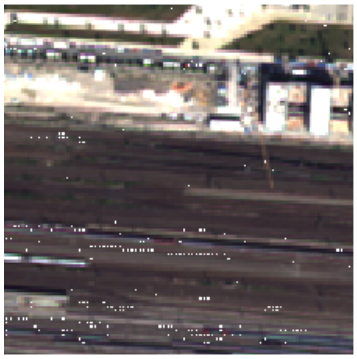
B.1 Extraction of Common Objects

For active learning results, the following shows results of common objects: railways, a red building, trees, and a white square building.

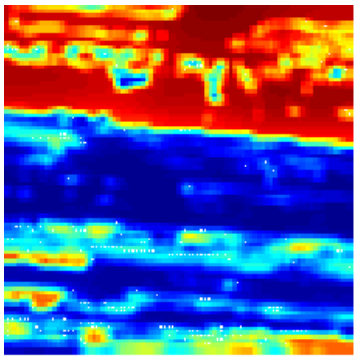
Railway case: the lighter part of *railways* looks different from the darker part of *railways*, assumably there were train cabins, so at the beginning the lighter part is described as light blue. After giving more training samples, the algorithm learns what the user wants and then returns better results later.



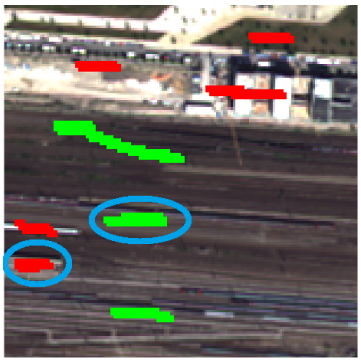
Iteration 1 of training samples



Iteration 1 of queried pixels

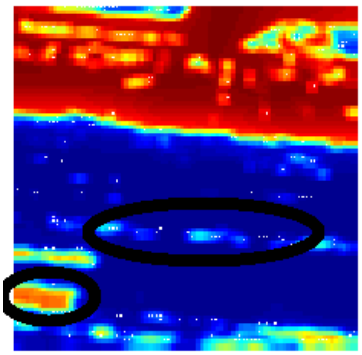


Iteration 1 of the non-locality map

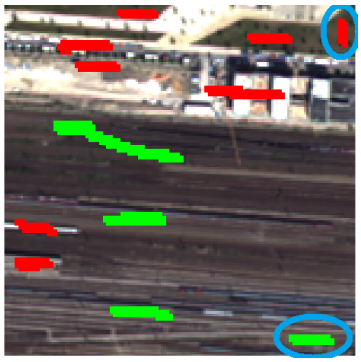


Iteration 2 of training samples

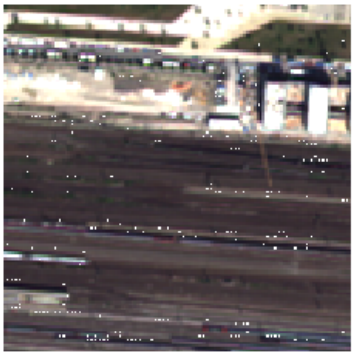
Iteration 2 of queried pixels



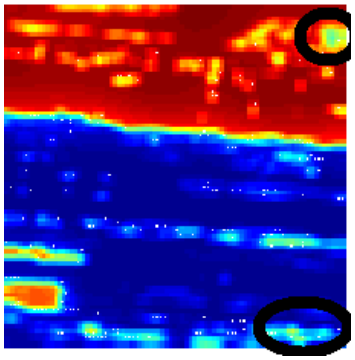
Iteration 2 of the non-locality map



Iteration 3 of training samples



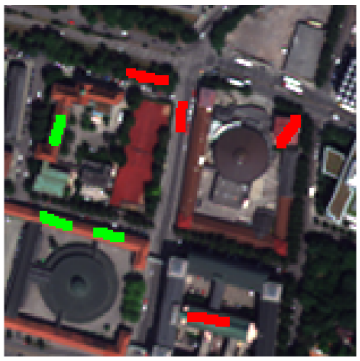
Iteration 3 of queried pixels



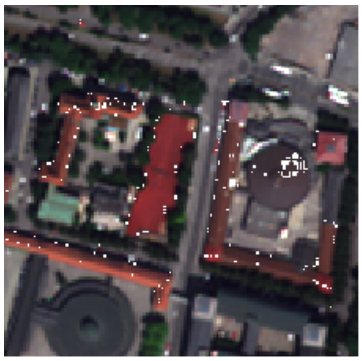
Iteration 3 of the non-locality map

Figure B.1: Iterations of training samples and non-locality maps for *railway* objects. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

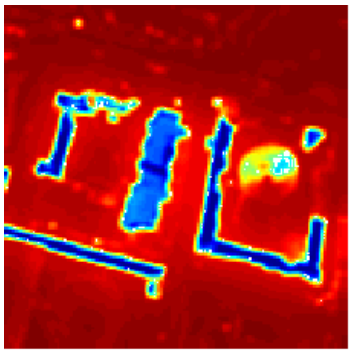
Red building case: Due to the used color descriptor, a *tennis court* has also been considered as a potential target at the beginning; however, after giving more negative training samples, the trained model learns the difference between *red buildings* and a *tennis court*.



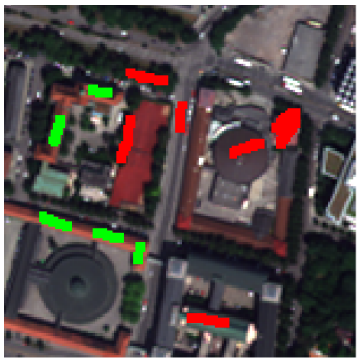
Iteration 1 of training samples



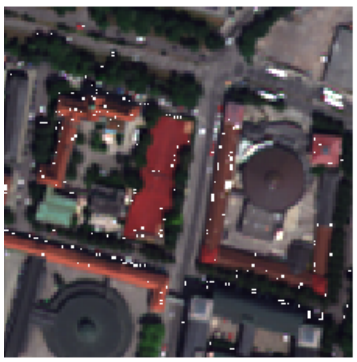
Iteration 1 of queried pixels



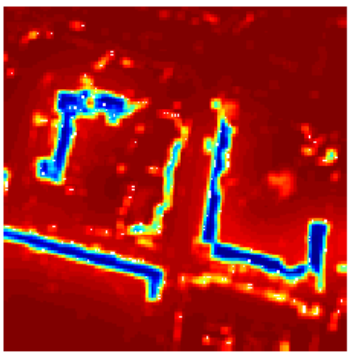
Iteration 1 of the non-locality map



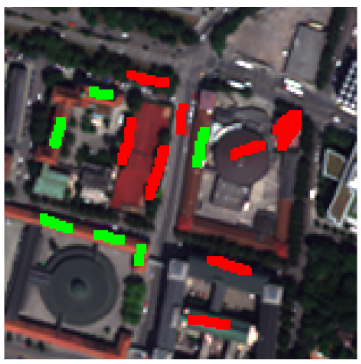
Iteration 2 of training samples



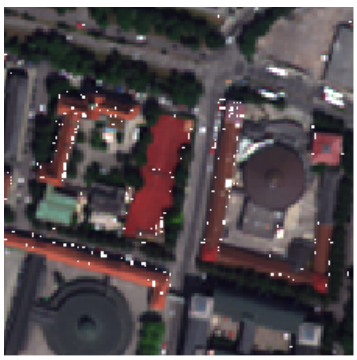
Iteration 2 of queried pixels



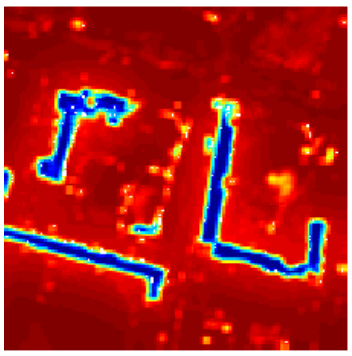
Iteration 2 of the non-locality map



Iteration 3 of training samples



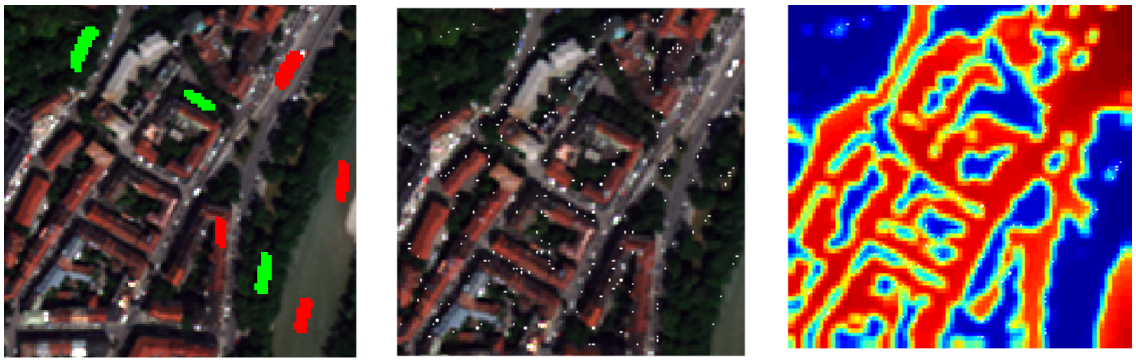
Iteration 3 of queried pixels



Iteration 3 of the non-locality map

Figure B.2: Iterations of training samples and non-locality maps for *red building* objects. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

Trees case: In this image, *trees* are very different from the rest of the image content, with a single training iteration the objects are extracted.



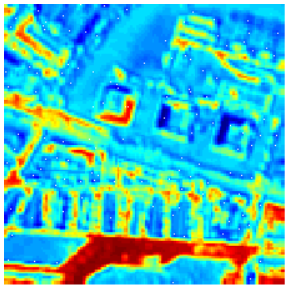
Iteration 1 of training samples | Iteration 1 of queried pixels | Iteration 1 of the non-locality map

Figure B.3: Iteration of training samples and non-locality maps for *trees* objects. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

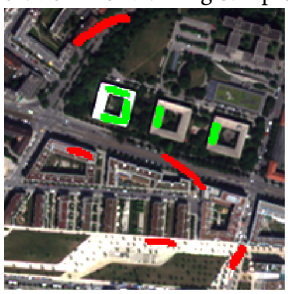
White square building case: With many similar objects and user-selected specific target objects, it takes 4 iterations to extract the *white square buildings*.

Iteration 1 of training samples

Iteration 1 of queried pixels

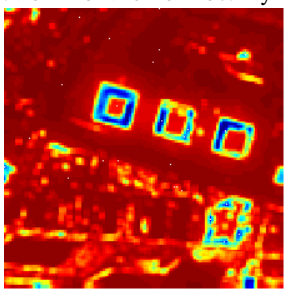


Iteration 1 of the non-locality map



Iteration 2 of training samples

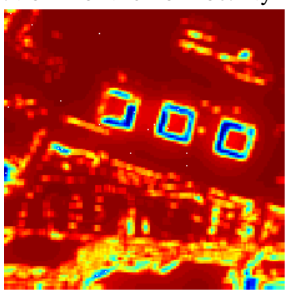
Iteration 2 of queried pixels



Iteration 2 of the non-locality map

Iteration 3 of training samples

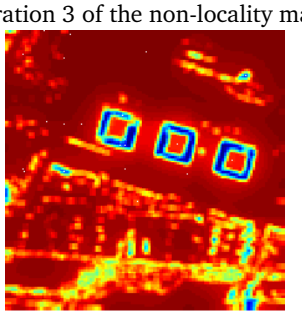
Iteration 3 of queried pixels



Iteration 3 of the non-locality map

Iteration 4 of training samples

Iteration 4 of queried pixels



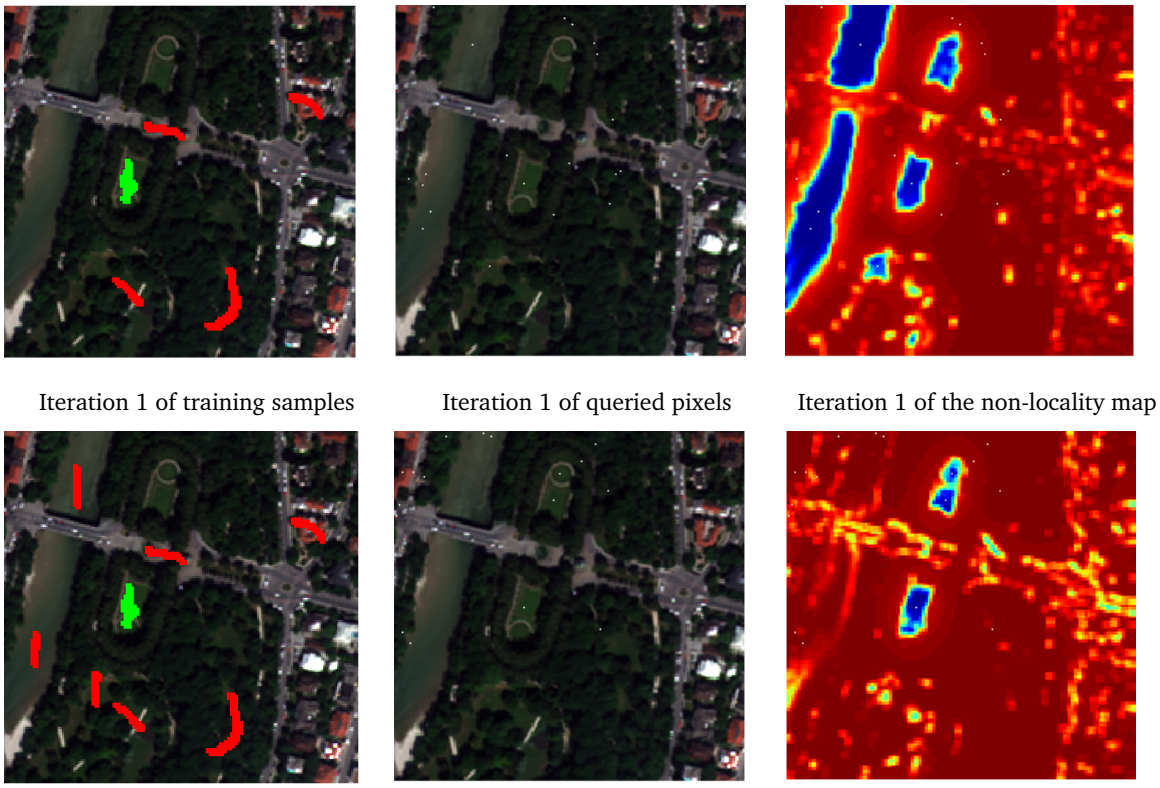
Iteration 4 of the non-locality map

Figure B.4: Iterations of training samples and non-locality maps for *white square building* objects. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

B.2 Extraction of Specific Objects

The following results of specific objects are shown: a sports field, a bridge, a round-about, and a pond.

Sports field case: The result of the first iteration indicates that *river* and *sports field* are similar. In fact, they do visually look similar, however, semantically they are different objects. After selecting *river* as negative samples in the second iteration, the user-preferred *sports fields* are extracted.

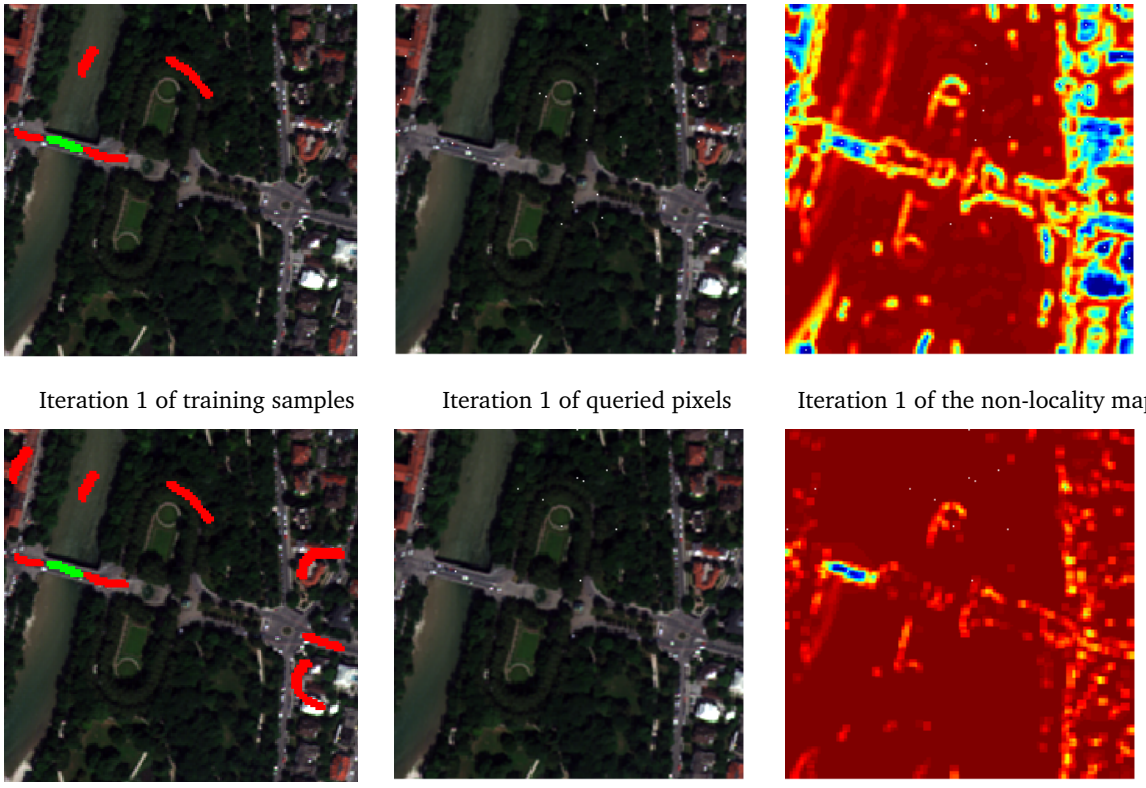


Iteration 1 of training samples | Iteration 1 of queried pixels | Iteration 1 of the non-locality map

Iteration 2 of training samples | Iteration 2 of queried pixels | Iteration 2 of the non-locality map

Figure B.5: Iterations of training samples and non-locality maps for *sports field* objects. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

Bridge case: A *bridge* is an object which is confused with roads on land; however, it refers to bridges over water. Hence, roads need to be given as negative samples.



Iteration 1 of training samples | Iteration 1 of queried pixels | Iteration 1 of the non-locality map

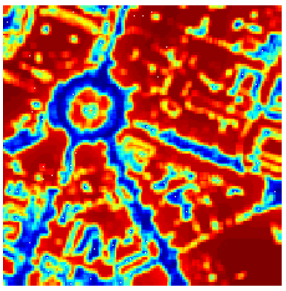
Iteration 2 of training samples | Iteration 2 of queried pixels | Iteration 2 of the non-locality map

Figure B.6: Iterations of training samples and non-locality maps for a *bridge* object. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

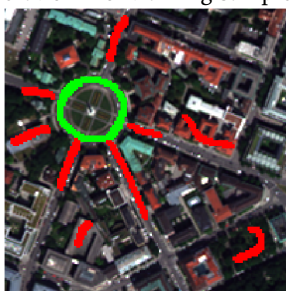
Round-about case: A round-about contains roads with a specific shape, so normal roads need to be given as negative samples during some iterations.

Iteration 1 of training samples

Iteration 1 of queried pixels

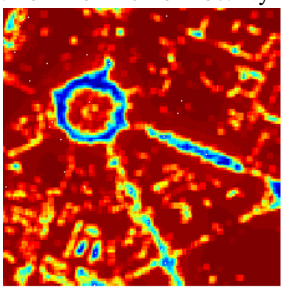


Iteration 1 of the non-locality map

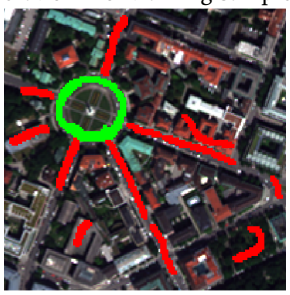


Iteration 2 of training samples

Iteration 2 of queried pixels

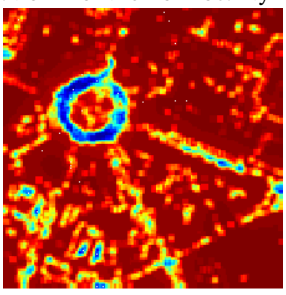


Iteration 2 of the non-locality map



Iteration 3 of training samples

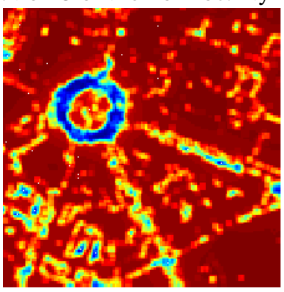
Iteration 3 of queried pixels



Iteration 3 of the non-locality map

Iteration 4 of training samples

Iteration 4 of queried pixels



Iteration 4 of the non-locality map

Figure B.7: Iterations of training samples and non-locality maps for a *round-about* object. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

Pond case: *Pond*-like small objects can be extracted when enough negative samples are given.

Iteration 1 of training samples | Iteration 1 of queried pixels | Iteration 1 of the non-locality map

Figure B.8: Iterations of training samples and non-locality maps for a *pond* object. Regarding training samples, green stands for positive training samples, and red stands for negative training samples; regarding results, blue stands for object, and red stands for non-object.

References

C. Aggarwal, D. Keim, and A. Hinneburg. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the International Conference on Database Theory (ICDT)*, pages 420–434, 2001. 32, 33, 50, 55, 72, 76

J. Albertz. *Einführung in die Fernerkundung. Grundlagen der Interpretation von Luft and Satellitenbildern*. WBG (Wissenschaftliche Buchgesellschaft), 2007. 8

R.C. Amorim and B. Mirkin. Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. *Pattern Recognition*, 45:1061–1075, 2012. ISSN 00313203. doi: 10.1016/j.patcog.2011.08.012. 33, 75

T.W. Anderson and D.A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 23:193–212, 1952. ISSN 0003-4851. doi: 10.1214/aoms/1177729437. 43

C. Andrieu, N. De Freitas, A. Doucet, and Jordan. M. I. An introduction to mcmc for machine learning. *Machine Learning*, pages 5–43, 2003. 20

D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. ISBN 978-0-898716-24-5. doi: 10.1145/1283383.1283494. 54

O. Aytekin, M. Koc, and I. Ulusoy. Local primitive patterns for the classification of SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 51:2431–2441, April 2013. 27

J. Bai, S.M. Xiang, and C.H. Pan. A graph-based classification method for hyperspectral images. In *2012 Spring Congress on Engineering and Technology (S-CET)*, S-CET'12, pages 1–4. IEEE, 2012. 39

A. Barriuso and A. Torralba. Notes on image annotation. *Computing Research Repository (CoRR)*, abs/1210.3448, 2012. 21, 22, 23

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: speeded up robust features. *Journal of computer vision and image understanding*, pages 346–359, June 2008. 29

A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013. URL http://arxiv.org/abs/1306.6709. 33, 34

R. Bellman. Princeton University Press, 1961. 31

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *CoRR*, 2014. 1

U.C. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder, and M. Heynen. Multi-resolution, object-oriented fuzzy analysis of remote sensing data for gis-ready information. *ISPRS journal of Photogrammetry and Remote Sensing*, pages 239–258, 2004. 21, 38

S. Berchtold, C. Boehm, and H.P. Kriegel. The pyramid-technique: towards breaking the curse of dimensionality. *Proceedings of the International Conference on Management of Data (ACM SIGMOD 1998)*, 1998. 31

M. Bilenko and R.J. Mooney. Learning to combine trained distance metrics for duplicate detection in databases, February 2002. Artificial Intelligence lab, University of Texas at Austin. 32

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738. 31

P. Blanchart. *Fast learning methods adapted to the user specificities: Application to Earth observation image information mining*. PhD thesis, Télécom ParisTech, Ecole Doctorale d'Informatique, 2012. 37

P. Blanchart and M. Datcu. A semi-supervised algorithm for auto-annotation and unknown structures discovery in satellite image databases. *IEEE journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 3:698–717, 2010. 36

P. Blanchart and M. Ferecatu. Non-locality maps for interactive foreground extraction and object detection. September 2014. 97, 108, 110

P. Blanchart and M. Ferecatu. Local integrity constraints for structure detection and segmentation in high-resolution earth observation images. In *Proceedings of the International Conference on Image Processing 2015 (ICIP)*, pages 373–377, 2015. 97, 108

P. Blanchart, M. Ferecatu, S.Y. Cui, and M. Datcu. Pattern retrieval in large image databases using multiscale coarse-to-fine cascaded active learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7:1127–1141, 2014. 36, 108, 109

T. Blaschke. Object based image analysis for remote sensing. *ISPRS journal of Photogrammetry and Remote Sensing*, 65:2–16, 2010. doi: 10.1016/j.isprsjprs.2009.06.004. 38

O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–8, 2008. 75

L. Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, pages 177–187, Paris, France, August 2010. Springer. 19

R. Bouchiha and K. Besbes. Automatic remote-sensing image regostration using surf. *International journal of computer theory and engineering*, February 2013. 30

Y.L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proceedings of the 23th IEEE conference on computer vision and pattern recognition*, pages 2559–2566, June 2010. 30

M. Bouziani, K. Goita, and D.C. He. Rule-based classification of a very high resolution image in an urban environment using multispectral segmentation guided by cartographic data. *IEEE Transaction on Geoscience and Remote Sensing*, 48(8):3198–3211, 2010. 39

D. Burago, Y. Burago, and S. Ivanov. A course in metric geometry. *Graduate studies in mathematics*, 2001. Department of Mathmetics, Pennsylvania State University Steklov Institute for Mathematics at St. Petersburg. 32

C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, pages 121–167, 1998. 133

G.R. Cai, P.M. Jodoin, S.Z. Li, Y.D. Wu, S.Z. Su, and Z.K. Huang. Perspective-sift: an efficient tool for low-altitude remote sensing image registration. *Journal of signal processing*, pages 3088–3110, 2013. 30

G. Camps-Valls, D. Tuia, L. Bruzzone, and J.A. Benediktsson. Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. *IEEE Signal Processing Magazine*, 31:45–54, 2014. 39

L.L. Cao, J.B. Luo, H. Kautz, and T.S. Huang. Image annotation within the context of personal photo collections using hierarchical event and scene models. *IEEE Transactions on Multimedia*, 11(2):208–219, 2009. 36

G. Carneiro, A.B. Chan, P.J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:394–410, 2007. ISSN 01628828. doi: 10.1109/TPAMI.2007.61. 36, 95

H. Chaabouni-Chouayakh. *Multi-Layer interpretation of high resolution SAR images: application to urban area mapping by means of information fusion*. PhD thesis, TELECOM ParisTech, 2009. 12

H. Chaabouni-Chouayakh and M. Datcu. Coarse-to-fine approach for urban area interpretation using TerraSAR-X data. *IEEE Geoscience and Remote Sensing Letters*, 7:78–82, 2010. 38

V.V. Chamundeeswari, D. Singh, and K. Singh. An analysis of texture measures in pca-based unsupervised classification of sar images. *IEEE Geoscience and Remote Sensing Letters*, 6 (2):214–218, 2009. 29

Ken Chatfield, Victor Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011. 30

K.M. Chen, P. Jian, Z.X. Zhou, J.E. Guo, and D.B. Zhang. Semantic annotation of high-resolution remote sensing images via Gaussian process multi-instance multilabel learning. *IEEE Geoscience Remote Sensing Letters*, 10:1285–1289, 2013. 24, 36

Z.Z. Chen and T. Ellis. Semi-automatic annotation samples for vehicle type classification in urban environments. *IET Intelligent Transport Systems*, 9(3):240–249, March 2015. 36

J.Y. Choi, W. De Neve, Y.M. Ro, and K.N. Plataniotis. Automatic face annotation in personal photo collections using context-based unsupervised clustering and face information fusion. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(10):1292–1309, 2010. 36

W.W. Cohen, P. Ravikumar, and S.E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI-03 workshop on information integration*, pages 73–78, August 2003. 32

S. Cui, G. Schwarz, and M. Datcu. Remote sensing image classification: no features, no clustering. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(11):5158–5170, November 2015. 30, 60

S.Y. Cui. *Spatial and Temporal SAR image information mining*. PhD thesis, UniversitÃd't Siegen, Naturwissenschaftlich-Technischen FakultÃd't, 2014. 73

S.Y. Cui, G. Schwarz, and M. Datcu. A comparative study of statistical models for multilook sar images. *IEEE Geoscience and Remote Sensing Letters*, 11:1752–1756, 2014. 99

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *in CVPR 05' Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 886–893, 2005. 30

M. Datcu and G. Schwarz. Image information mining methods for exploring and understanding high resolution images. In *Proceedings of International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 33–35, 2010. ISBN 9781424495658. doi: 10.1109/IGARSS.2010.5654442. 43, 51

M. Datcu and K. Seidel. Image information mining: Exploration of earth observation archives. *Geographica Helvetica*, 58(2):154–168, 2003. doi: 10.5194/gh-58-154-2003. URL http://www.geogr-helv.net/58/154/2003/. 37

M. Datcu and K. Seidel. Human-centered concepts for exploration and understanding of earth observation images. In *IEEE Transactions on Geoscience and Remote Sensing*, volume 43, pages 601–609, 2005. ISBN 0196-2892. doi: 10.1109/TGRS.2005.843253. 35

R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2), May 2008. 21, 36

T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval: An experimental comparison. *Information Retrieval*, 11(2):77–107, April 2008. 29

M.M. Deza and E. Deza. *Encyclopedia of distances*. Springer-Verlag Berlin Heidelberg, 2009. 32

DigitalGlobe. Remote sensing technology trends and agriculture, 2015. URL https://global.digitalglobe.com/sites/default/files/DG-RemoteSensing-WP.pdf. White paper, Remote Sensing. 22

M.N. Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler diatance. *IEEE Transactions on image processing*, pages 146–158, February 2002. 29

D.L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. In *AMS Conference on Math Challenges of the 21st Century*, 2000. 31

Drinkwater. Coordinating satellite observations during the international polar year 2007-2008, March 2008. URL http://earthzine.org/2008/03/09/coordinating-satellite-observations-during-the-international-polar-year-2007-2008/. Accessed: 2016-07-15. 8

C.O. Dumitru and M. Datcu. Information content of very high resolution sar images: Study of feature extraction and imaging parameters. *IEEE Transactions on Geoscience and Remote Sensing*, 51:4591–4610, 2013. 27, 60, 73, 77, 79

C.O. Dumitru, Shiyong Cui, and M. Datcu. Information content of very high resolution sar images: Semantics, geospatial context, and ontologies. *IEEE booktitle of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1635–1650, November 2014. 37, 73, 77, 78, 79

J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1973. 76

B. Epshtein and S. Uliman. Feature hierarchies for object classification. In *Proceedings of the tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 220 – 227, October 2005. 25

B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*. IEEE Computer Society, 2007. 25

S. Ermon, Gomes C. P., Sabharwal A., and B. Selman. Taming the curse of dimensionality: Discrete integration by hashing and optimization. *CoRR*, abs/1302.6677, 2013. 31

T. Esch, A. Schenk, T. Ullmann, M. Thiel, A. Roth, and S. Dech. Characterization of land cover types in TerraSAR-X images by combined analysis of speckle statistics and intensity information. *IEEE Transactions on Geoscience and Remote Sensing*, 49:1911–1925, 2011. 97, 98, 104, 105, 120, 123

D. Espinoza-Molina and M. Datcu. Earth-observation image retrieval based on content, semantics, and metadata. *IEEE Transactions on Geoscience and Remote Sensing*, 51:5145–5159, 2013. 37

J. Feng, L.C. Jiao, X.R. Zhang, and D.D. Yang. Bag-of-visual words based on clonal selection algorithm for sar image classification. *IEEE Geoscience and Remote Sensing Letters*, 8(4): 691–695, 2011. 30

T. Fletcher. Support vector machines explained, March 2009. 132, 133

W. Foerstner. Computer vision and remote sensing - lessons learned. University of Stuttgart, Department of Photogrammetry, 2009. 22

D. Francois, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, 2007. 33

C. Francu and C.G. Nevill-Manning. Distance metrics and indexing strategies for a digital library of popular music. In *Proceedings on 2000 IEEE international conference on multimedia and expo, ICME2000*, August 2000. 32

A.C. Frery, H.J. Mueller, C.C.F. Yanasse, and S.J.S. Sant'Anna. A model for extremely heterogeneous clutter. *IEEE Transactions on Geoscience and Remote Sensing*, 35(3):648–659, 1997. 97, 99, 104

T. Fritz. TerraSAR-X level 1b product format specification, 2013. URL http://www2.geo-airbusds.com/files/pmedia/public/r460_9_030201_level-1b-product-format-specification_1.3.pdf. 42, 58, 77, 78

G. Gao. Statistical modeling of sar images: A survey. *Sensors*, pages 775–795, January 2010. 12

S.H. Gao, L.T. Chia, I.W.H. Tsang, and Z.X. Ren. Concurrent single-label image classification and annotation via efficient multi-layer group sparse coding. *IEEE Transactions on Multimedia*, 16(3):762–771, 2014. doi: 10.1109/TMM.2014.2299516. URL http://dx.doi.org/10.1109/TMM.2014.2299516. 23, 25

S. Gautama, Heene G., R. Pires, J. D'Haeyer, and I. Bruyland. Computer vision techniques for remote sensing. University of Ghent, Department of Telecommunication and Information Processing, 2000. 23

J.L. Gauvain and C.H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, 1994. 16

D.G. Gavin, W.W. Oswald, E.R. Wahl, and J.W. Williams. A statistical approach to evaluating distance metrics and analog assignments for pollen records. *Quaternary research*, pages 243–428, November 2003. 32

C. Geiss and H. Taubenboeck. Object-based postclassification relearning. *IEEE Geoscience and Remote Sensing Letters*, 12:2336–2340, 2015. 39

T. Gevers and A.W.M. Smeulders. Color-based object recognition. *Pattern recognition*, pages 453–464, March 1999. 28

T. Gevers and A.W.M. Smeulders. Image search engines: An overview, 2003. 36

P. Ghamisi, M.D. Mura, and J.A. Benediktsson. A survey on spectral-spatial classification techniques based on attribute profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 53:2335–2353, 2015. 39

D.H. Guo, H. Xiong, V. Atluri, and N.R. Adam. Object discovery in high-resolution remote sensing images: a semantic perspective. *Under consideration for publication in knowledge and information systems*, pages 211–233, May 2009. 38

M. Halakidi, Y. Batistakis, and M. Vazirgiannis. Clustering validity checking methods: Part II. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data (SIGMOD'02)*, volume 31, pages 19–27, 2002. 56

G. Hamerly and C. Elkan. Learning the k in k-means. In *Proceedings of the Seventh Annual Conference on Neural Information Processing Systems (NIPS)*, pages 281–288, 2003. 43, 50, 51, 52, 53, 54

Y.H. Han, X.X. Wei, X.C. Cao, Y. Yang, and X.F. Zhou. Augmenting image descriptions using structured prediction output. *IEEE Transactions on Multimedia*, 16(6):1665–1676, 2014. 23

R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on systems, man and cybernetics*, pages 610–621, November 1973. 28

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, February 2009. 75

J. Hervé, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, September 2012. doi: 10.1109/TPAMI.2011.235. URL https://hal.inria.fr/inria-00633013. 31

S.C.H. Hoi, W. Liu, M.R. Lyu, and W.Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition, CVPR 2006*, pages 2072 – 2078, 2006. 34

S.C.H. Hoi, W. Liu, and S.F. Chang. Semi-supervised distance metric learning for collaborative image retrieval. *ACM transactions on multimedia computing, communications, and applications (TOMM)*, August 2010. 34

D. Hoiem, A.a. Efros, and M. Hebert. Geometric context from a single image. *Proceedings on 2005 International conference on computer vision ICCV'05*, pages 654–661, 2005. 29

W.L. Hoo and C.S. Chan. Zero-shot object recognition system based on topic model. *IEEE Transactions on Human-Machine Systems*, 45:518–525, 2015. 23, 25

P. Howarth and S. Rueger. Evaluation of texture features for content-based image retrieval. *Proceedings of the international conference on image and video retrieval*, 2004. 29

P. Howarth and S. M. Rueger. Fractional distance measures for content-based image retrieval. In D. E. Losada and J. M. FernÃąndez-Luna, editors, *Proceedings of the 27th European Conference on Information Retrieval (ECIR 2005)*, volume 3408 of *Lecture Notes in Computer Science*, pages 447–456. Springer, 2005. 33

N.R. Howe and D.P. Huttenlocher. Integrating color, texture, and geometry for image retrieval. *Proceedings of IEEE conference on computer vision and pattern recognition*, pages 239–246, 2000. 27

J. Huang, S.R. Kumar, Mitra. M., and W.J. Zhu. Image indexing using color correlograms. In *Proceedings of the 1997 conference on computer vision and pattern recognition (CVPR 1997)*, pages 762–768, June 1997. 27

Y.Z. Huang, Z.F. Wu, L. Wang, and T.N. Tan. Feature coding in image classification: a comprehensive study. *IEEE Transactions on pattern analysis and machine intelligence*, pages 493 – 506, March 2014. 30

D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transaction of Pattern Analysis and Machine Intelligence*, pages 850–863, September 1993. 32

Xudong Jiang. Feature extraction for image recognition and computer vision. In *Proceedings of 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009)*, pages 1 – 15, August 2009. 27, 34

B. Johnson and Z.X. Xie. Unsupervised image segmentation evaluation and refinement using a multi-scale approach. *ISPRS journal of Photogrammetry and Remote Sensing*, 66:473–483, 2011. doi: 10.1016/j.isprsjprs.2011.02.006. 38

Joni-Kristian Kamarainen, Ville Kyrki, and Heikki Kaelviaeinen. Invariance properties of gabor filter-based features - overview and applications. *IEEE Transactions on Image Processing*, 15, May 2006. 28

V. Kantorov and I. Laptev. Efficient feature extraction encoding and classification for action recognition. In *Proceedings of 2014 IEEE conference on computer vision and pattern recognition*, pages 2593 – 2600, June 2014. 30

O.O. Karadag, C. Senaras, and F.T.Y. Vural. Segmentation fusion for building detection using domain-specific information. *IEEE journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8:3305–3315, 2015. 39

S.M. Khaligh-Razavi. What you need to know about the state-of-the-art computational models of object-vision: a tour through the models. *CoRR*, 2014. 1

F. Kovacs, C. Legany, and A. Babos. Cluster validity measurement techniques. In *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'06)*, pages 388–393, 2006. 56

J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964. ISSN 00333123. doi: 10.1007/BF02289565. 33, 56

J.B. Kruskal. Toward a practical method which helps uncover the structure of a set of observations by finding the line transformation which optimizes a new "index of condensation". *Statistical Computation*, pages 427–440, 1969. 55

P. Lassalle, J. Inglada, J. Michel, and M. Grizonnet. A scalable tile-based framework for region-merging segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 53: 5473–5485, 2015. 38

S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proceedings of the ninth IEEE international conference on computer vision*, pages 649–655, October 2003. 28

Q. Le, M.A. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In *International Conference in Machine Learning*, 2012. 1

Y. LeCun. Learning invariant feature hierarchies. In *Proceedings of the 12th European Conference on Computer Vision*, pages 496 – 505. Springer-Verlag Berlin Heidelberg, October 2012. 24

T. S. Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on pattern analysis and mahcine intelligence*, October 1996. 52

F.F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. *Proceedings on 2005 IEEE Computer Societry Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005. 30

F.F. Li, R. Fergus, and A. Torralba. Recognizing and learning object categories, 2007. CVPR 2007 short course. 30

N. Li, H. Huo, and T. Fang. A novel texture-preceded segmentation algorithm for high-resolution imagery. *IEEE Transaction on Geoscience and Remote Sensing*, 48(7):2818–2828, 2010. 38

M. Lienou, H. Maitre, and M. Datcu. Semantic annotation of satellite images using latent Dirichlet allocation. *IEEE Geoscience and Remote Sensing Letters*, 7:28–32, 2010. 36

D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International booktitle of computer vision*, pages 91–110, November 2004. 29

Yijuan Lu, Lei Zhang, Jiemin Liu, and Qi Tian. Constructing concept lexica with small semantic gaps. *IEEE Transactions on Multimedia*, 12(4):288–299, 2010. 36

W. Luo, H. Li, and G. Liu. Automatic annotation of multispectral satellite images using author-topic model. *IEEE Geoscience and Remote Sensing Letters*, 9:634–638, 2012. 36

W. Luo, H.L. Li, G.H. Liu, and L.Y. Zeng. Semantic annotation of satellite images using author-genre-topic model. *IEEE Transactions on Geoscience and Remote Sensing*, 52:1356–1368, 2014. 24

B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, pages 837 – 842, August 1996. 28

B.S. Manjunath, J.R. Ohm, V.V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE transactions on circuits and systems for video technology*, pages 703–715, June 2001. 27, 28

M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007).*, pages 1–7, June 2007. 25

K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International journal of Computer Vision*, 60:63–86, October 2004. 28

R. Morgan and M. Gallagher. Sampling techniques and distance metrics in high dimensional continuous landscape analysis: limitations and improvements. *IEEE transaction of evolutionary computation*, pages 456 – 461, June 2014. 33

G. Moser, J. Zerubia, and Serpico S.B. Sar amplitude probability density function estimation based on a generalized gaussian model. *IEEE Transaction of Image Processing*, pages 1429–1442, June 2006. 16

C. Mostegel, M. Rumpler, F. Fraundorfer, and H. Bischof. Uav-based autonomous image acquisition with multi-view stereo quality assurance by confidence prediction. *IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, 2016. 22

G. Mountrakis, J. Im, and C. Ogole. Support vector machines in remote sensing: A review. *ISPRS journal of Photogrammetry and Remote Sensing*, 66:247–259, May 2011. doi: 10.1016/j.isprsjprs.2010.11.001. 36

MPEG-7. URL http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm. 60, 73, 79

S.K. Mylonas, D.G. Stavrakoudis, D.B. Theocharis, and P.A. Mastorocostas. Classification of remotely sensed images using the genesis fuzzy segmentation algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, 53:5352–5376, 2015. 38

R.B. Nelsen. *An Introduction to Copulas (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 14

S. Newsam, L. Wang, S. Bhagavathy, and B.S. Manjunath. Using texture to annotate remote sensed datasets. In *Proceedings of the 3rd International Sympopisum Image Signal Process Analysis (ISPA 2003)*, volume 1, pages 72–77, 2003. 27, 51

J.M. Nicolas. Introduction to second kind statistics application of log moments and log cumulants to the analysis of radar image distributions. *Traitement du Signal*, pages 139–167, 2002. In French. 16

C. Oliver and S. Quegan. *Understanding Synthetic Aperture Radar Images*. Artech House, 1998. 9, 10, 11

F. Olsson. A literature survey of active machine learning in the context of natural language processing, April 2009. Swedish Institute of Computer Science. 36

K. Park, C. Shen, Z.H. Hao, and J. Kim. Efficiently learning a distance metric for large margin nearest neighbor classification. In *Procedings of the twenty-fifty AAAI conference on artificial intelligence*, August 2011. 34

G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proceedings of the Fourth ACM International Conference on Multimedia*, MULTIMEDIA '96, pages 65–73. ACM, 1996. 27

C. Persello, A. Boularias, M. Dalponte, T. Gobakken, E. Naesset, and B. Schölkopf. Cost-sensitive active learning with lookahead: Optimizing field surveys for remote sensing data classification. *IEEE Transactions on Geoscience and Remote Sensing*, 10(52):6652–6664, 2014. 37

Earth Observation Portal. TerraSAR-X Mission, 2016. URL https://directory.eoportal.org/web/eoportal/satellite-missions/t/terrasar-x. [Online; accessed 25-December-2016]. 1, 9

M. Prince. Does active learning work? a review of the research. *Journal of Engineering Education*, pages 223–231, 2004. 109

M. Quartulli and I.G. Olaizola. A review of EO image information mining. *ISPRS journal of Photogrammetry and Remote Sensing*, 75:11–28, 2013. 37

T. Randen and J.H. Husoy. Multichannel filtering for image texture segmentation. *Optical Engineering*, August 1994. 29

D. Reynolds. Gaussian mixture models. *Encyclopedia of Biometric Recognition*, February 2008. 15

V.F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J.P. Rigol-Sanchez. An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS journal of Photogrammetry and Remote Sensing*, 67:93–104, 2012. doi: 10.1016/j.isprsjprs.2011.11.002. 36

L.A. Ruiz, A. Fdez-Sarria, and J.A. Recio. Texture feature extraction for classification of remote sensing data using wavelet decomposition: a comparative study. In *International Archives of Photogrammetry and Remote Sensing. Vol.XXXV, ISSN*, pages 1682–1750, 2004. 29

B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008. ISSN 09205691. doi: 10.1007/s11263-007-0090-8. 23

P Salembier and T Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York, NY, USA, 2002. 27

K. Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on pattern analysis and machine intelligence*, pages 1582 – 1596, September 2010. 27

D. Schnitzer and A. Flexer. Choosing the metric in high-dimensional spaces based on hub analysis. In *22th European Symposium on Artificial Neural Networks, ESANN 2014, Bruges, Belgium, April 23-25, 2014*, 2014. URL http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2014-16.pdf. 33

M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *In proceedings of advances in neural information processing systems, NIPS 2003*. MIT Press, 2003. 33

B. Settles. Active learning literature survey, Januray 2014. Computer Sciences Technical report 1648, University of Wisconsin-Madison. 36

S. Sharma. *Applied multivariate techniques*. John Wiley and Sons, Inc., New York, January 1996. 56

C.R. Shyu, M. Klaric, G.J. Scott, A.S. Barb, C.H. Davis, and K. Palaniappan. Geoiris: Geospatial information retrieval and indexing system - content mining, semantics modeling, and complex queries. *IEEE Transactions on Geoscience and Remote Sensing*, 45: 839–852, 2007. ISSN 01962892. doi: 10.1109/TGRS.2006.890579. 37

A. Singh, A. Yadav, and A. Rana. K-means with three different distance metrics. *International journal of computer applications*, pages 13–17, April 2013. 32

P. Sinha and T. Poggio. Role of learning in three-dimensional form perception. In *Letters to nature*, volume 384, December 1996. 40

M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45:427–437, 2009. 76

J. Stefanski, B. Mack, and B. Waske. Optimization of object-based image analysis with random forests for land cover mapping. *IEEE journal of Selected Topics in Applied Earth Observation and Remote Sensing*, 6:2492–2504, 2013. 39

M. Steinbach, L. ErtÃűz, and V. Kumar. The challenges of clustering high-dimensional data. In *In New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recognition*. Springer-Verlag, 2003. 31

M.A. Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, 69(347):730–737, September 1974. 53

M.A. Stephens. Tests based on edf statistics. *Goodness-of-fit techniques*, pages 97–193, 1986. 53

J.V. Stone. *Independent component analysis: A tutorial introduction*. The MIT Press Cambridge, 2004. 55

M. Stricker and M. Orengo. Similarity of color images. pages 381–392, 1995. 27

J. Su, C. Chou, C. Lin, and V. Tseng. Effective semantic annotation by image-to-concept distribution model. *IEEE Transactions on Multimedia*, 13:530–538, 2011. ISSN 15209210. doi: 10.1109/TMM.2011.2129502. 36

M.J. Swain and D.H. Ballard. Color indexing. *International booktitle of computer vision*, pages 11–32, November 1991. 27

H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on systems, man, and cybernetics*, pages 460–473, June 1978. 29

J. Tang, M. Musolesi, C. Mascolo, and V. Latora. Temporal distance metrics for social network analysis. In *Proceedings of the 2nd ACM workshop on online social networks, WOSN 2009*, pages 31–36, 2009. 32

Y. Tarabalka, J.C. Tilton, J.A. Benediktsson, and J. Chanussot. A marker-based approach for the automated selection of a single segmentation from a hierarchial set of image segmentations. *IEEE journal of Selected Topics in Applied Earth Observation and Remote Sensing*, 5:262–272, 2012. 39

C. Tison, J.M. Nicolas, F. Tupin, and H. Maitre. A new statistical model for markovian classification of urban areas in high resolution sar images. *IEEE transactions on Geoscience and Remote Sensing*, 42:2046–2057, 2004. 13, 16, 99

P. Tokarczyk, J.D. Wegner, S. Walk, and Schindler. K. Features, color spaces, and boosting: new insights on semantic classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, January 2015. 27

S. Tong. *Active learning: theory and applications*. PhD thesis, Stanford university, 2001 (accessed October 17, 2016). URL http://www.robotics.stanford.edu/~stong/papers/tong_thesis.pdf. 109

P.A. Torrione, K.D. Morton, R. Sakaguchi, and L.M. Collins. Histograms of oriented gradients for landmine detection in ground-penetrating radar data. *IEEE transactions on geoscience and remote sensing*, March 2014. 30

A.M. Tousch, S. Herbin, and J.Y. Audibert. Semantic hierarchies for image annotation: A survey. *Pattern Recognition*, 45(1):333–345, January 2012. ISSN 0031-3203. doi: 10.1016/j.patcog.2011.05.017. URL http://dx.doi.org/10.1016/j.patcog.2011.05.017. 25

A. Troya-Galvis, P. Gançarski, N. Passat, and L. Berti-Équille. Unsupervised quantification of under- and over-segmentation for object-based remote sensing image analysis. *IEEE journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(5):1936–1945, 2015. 39

F. Ulaby, R. Moore, and A. Fung. *Microwave remote sensing: active and passive, volume 1, Microwave remote sensing fundamentals and radiometry*. Addison-Wesley, 1981. 9

A. Ulges, M. Worring, and T. Breuel. Learning visual contexts for image annotation from flickr groups. *IEEE Transactions on Multimedia*, 13:330–341, 2011. ISSN 15209210. doi: 10.1109/TMM.2010.2101051. 37

J.P. Van de Geer. Some aspects of minkowski distances, March 1995. department of data theory, Leiden University. 33

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 76, 87, 88, 89

C.J. Van Rijsbergen. Butterworth, 2nd ed. edition, 1979. 76

M.C. Vanegas, I. Bloch, and J. Inglada. Alignment and parallelism for the description of high-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 51: 3542–3557, 2013. 39

R. Varga and S. Nedevschi. Label transfer by measuring compactness. *Image Processing, IEEE Transactions on*, 22(12):4711–4723, December 2013. 36

M. Verleysen and D. Francois. The curse of dimensionality in data mining and time series prediction. *Proceedings of the 8th International Work-Conference on Artificial Neural Networks (IWANN 2005, LNCS 3512)*, pages 758–770, 2005. 31

C. Vertan and N. Boujemaa. Upgrading color distributions for image retrieval can we do better? In *Proceedings of the 4th international conference, VISUAL 2000*, November 2000. 27

A. Voisin, V.A. Krylov, G. Moser, S.B. Serpico, and J. Zerubia. Classification of very high resolution sar images of urban areas using copulas and texture in a hierarchical markov random field model. *IEEE Geoscience and Remote Sensing Letters*, 10:96–100, 2013. 99

U. Von Luxburg and O. Bousquet. Distance-based classification with lipschitz function. *Journal of machine learning research*, pages 669 – 695, 2004. 32

D.Y. Wang, S.C.H. Hoi, Y. He, and J.K. Zhu. Mining weakly labeled web facial images for search-based face annotation. *IEEE Transactions on Knowledge and Data Engineering*, 26 (1):166–179, 2014. 36

H.B. Wang, L. Feng, J. Zhang, and Y. Liu. Semantic discriminative metric learning for image similarity measurement. *IEEE Transactions on Multimedia*, August 2016. 33

J.X. Wang and Y.W. Wang. Modified surf applied in remote sensing image stitching. *International journal of signal processing, image processing and pattern recognition*, pages 1–10, 2015. 30

X.J. Wang, L. Zhang, X.R. Li, and W.Y. Ma. Annotating images by mining image search results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1919–1932, 2008. 36

K.Q. Weinberger and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, pages 207–244, June 2009. 34

Wikipedia. Worldview-2 — Wikipedia, the free encyclopedia, 2016. URL https://en.wikipedia.org/wiki/WorldView-2. [Online; accessed 24-July-2016]. 42

J.X. Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, 2010. ISBN 9781424469840. doi: 10.1109/CVPR.2010.5539970. 23

E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in neural information processing systems 15*, pages 505–512. MIT Press, 2003. 34

J. Yang, J.P. Fan, D. Hubball, Y.L. Gao, H.Z. Luo, W. Ribarsky, and M.O. Ward. Semantic image browser: Bridging information visualization with automated intelligent image analysis. In *IEEE Symposium On Visual Analytics Science And Technology, IEEE, VAST 2006, October 31-November 2, 2006, Baltimore, Maryland, USA*, pages 191–198, 2006. doi: 10.1109/VAST.2006.261425. URL http://dx.doi.org/10.1109/VAST.2006.261425. 37

J. Yang, Y. Jiang, A. Hauptmann, and C. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the International Workshop on Multimedia Information Retrieval*, pages 197–206, 2007. 30, 52, 60

L. Yang. Distance metric learnig: a comprehensive survey, 2006. Department of computer science and engineering, Michigan state university. 34

Y. Yang and S. Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Informaiton Systems*, pages 270–279, 2010. 30, 58

W. Yao, C.O. Dumitru, O. Loffeld, and M. Datcu. Semi-supervised hierarchical clustering for semantic sar image annotation. *IEEE journal of selected topics in applied earth observations and remote sensing*, 9:1993–2008, 2016a. 49

W. Yao, O. Loffeld, and M. Datcu. Application and evaluation of a hierarchical patch clustering method for remote sensing images. *IEEE journal of selected topics in applied earth observations and remote sensing*, 9:2279–2289, 2016b. 54, 69

L.N. Yi, G.F. Zhang, and G.C. Wu. A scale-synthesis method for high spatial resolution remote sensing image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 50: 4062–4070, 2012. 38

J. Yu, J. Amores, N. Sebe, and Q. Tian. A new study on distance metrics as similarity measurement. In *Proceedings of the 2006 IEEE international conference on multimedia and expo, ICME 2006*, pages 533–536, July 2006. 32

Z.Q. Yuan, C.S. Xu, J.T. Sang, and S.C. Yan. Learning feature hierarchies: A layer-wise tag-embedded approach. *IEEE Transactions on Multimedia*, 17(6):816–827, 2015. doi: 10.1109/TMM.2014.2299516. URL http://dx.doi.org/10.1109/TMM.2014.2299516. 23, 25

D.S. Zhang and G.J. Lu. Review of shape representation and description techniques. *Pattern Recognition*, pages 1–19, 2004. 29

G.Y. Zhang, X.P. Jia, and J.K. Hu. Superpixel-based graphical model for remote sensing image mapping. *IEEE Transaction on Geoscience and Remote Sensing*, 53(11):5861–5871, 2015. 39

Q. Zhang, X. Huang, and L.P. Zhang. An energy-driven total variation model for segmentation and classification of high spatial resolution remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, 10:125–129, 2013. 39

S.L. Zhang, Q. Tian, Q.M. Huang, and W. Gao. Cascade category-aware visual search. *IEEE Transactions on Image Processing*, 23:2514–2527, 2014. 24, 25

Z. Zhang, M.Y. Yang, M. Zhou, and X.Z. Zeng. Simultaneous remote sensing image classification and annotation based on the spatial coherent topic model. In *Proceedings of International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1698–1701, 2010. 36

J. Zhao, Y.F. Zhong, and L.P. Zhang. Detail-perserving smoothing classifier based on conditional random fields for high spatial resolution remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 53:2440–2452, 2015. 39

Y.F. Zhong, B. Zhao, and L.P. Zhang. Multiagent object-based classifier for high spatial resolution imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 52:841–857, 2014. 38

P.F. Zhu, L. Zhang, and D Zuo, W.M. Zhang. From point to set: extend the learning of distance metrics. In *Proceedings of the 2013 IEEE international conference on computer vision, ICCV 2013*, pages 2664 – 2671, December 2013. 34

X.J. Zhu and A.B. Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3:31–33, 2009. 46, 72, 74

X.J. Zhu, J. Lafferty, and Z.B. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003. 37