Geospatial Object Detection Through Topological Data Analysis

by

Meirman Syzdykbayev

B.S. in Geography, Eurasian National University, L.N.Gumilyov, 2012

PRO.MS. in Geographical Information Systems and Remote Sensing, University of Pittsburgh, 2017

Submitted to the Graduate Faculty of the

School of Computing and Information in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

SCHOOL OF COMPUTING AND INFORMATION

This thesis was presented

by

Meirman Syzdykbayev

It was defended on

December 7, 2023

and approved by

Dr. Daqing He, Professor, Department of Informatics and Networked Systems, School of Computing and Information
Dr. Konstantinos Pelechrinis, Associate Professor, Department of Informatics and Networked Systems, School of Computing and Information
Dr. Xiaowei Jia, Assistant Professor, Department of Computer Science, School of Computing and Information

Thesis Advisor: Dr. Hassan A. Karimi, Professor, Department of Informatics and Networked Systems, School of Computing and Information

Copyright © by Meirman Syzdykbayev

2023

Geospatial Object Detection Through Topological Data Analysis

Meirman Syzdykbayev, PhD

University of Pittsburgh, 2023

Geospatial object detection plays a key role in geospatial data analysis and is used in a variety of applications. Geospatial objects can be detected through computer vision and machine learning (ML) models and algorithms by focusing on geometrical and/or contextual information. There are challenges with geospatial object detection models and algorithms including geospatial data noise, geospatial data labeling, and feature representation. In addition to geometrical and contextual information, geospatial datasets contain topological information, which is often not considered for detection. The hypothesis in this thesis is that incorporating information on the shape of a geospatial objects, i.e., topological information, may address these challenges and improve detection accuracy. One way to incorporate topological information, along with geometrical or contextual information, is by using methods from topological data analysis such as Persistent Homology (PH) and Mapper.

The research in this thesis is focused on the development of methods that utilize topological information to detect geospatial objects. Explored in this research is incorporation of topological information into two geospatial object detection methods, knowledge-based, where topological information is merged with geometrical and contextual information, and ML-based, where topological information is transformed into a multichannel image as an additional feature. To test the results of these two methods after topological information is incorporated into them, their performances were evaluated by detecting landslide boundaries.

The main contributions of this research are evaluation of PH in the detection of landslides using LiDAR data; development of an algorithm to extract linear terrain features from digital terrain models; development of a topological knowledge-based geospatial object detection method; development of a topological ML-based geospatial object detection method; and development two algorithms to transform topological information into a multichannel image.

Table of Contents

List of Tables

List Of List of Figures

Preface

This thesis would not have been possible without the guidance, support, and encouragement of many individuals who have generously shared their wisdom, resources, and time with me.

Firstly, my deepest appreciation goes to my advisor, Dr. Hassan Karimi. Throughout my Ph.D. journey, his unwavering dedication, meticulous guidance, and enthusiasm for research have been my compass. His uncompromising commitment to academic excellence and his ability to always encourage me to explore beyond the known horizons have left an indelible mark on my academic trajectory. Over the course of our six-year collaboration, he has consistently challenged me to reevaluate accepted norms and adopt methodical and innovative approaches to scientific challenges. I am deeply grateful for the countless hours, insights, and resources he has generously invested, ensuring that my Ph.D. experience was not only intellectually rewarding but also personally enriching. To be mentored by a luminary such as Dr. Hassan Karimi is an honor I will cherish forever.

I would also like to express my sincere gratitude to my thesis committee members: Dr. Daqing He, Dr. Konstantinos Pelechrinis, and Dr. Xiaowei Jia. Each of them has provided astute feedback, critical evaluation, and constructive criticism that has substantially enhanced the quality of my research. Their encouragement and consistent push for excellence expanded the scope and depth of my research horizons.

Special recognition is extended to Dr. Dara Mendez and Dr. Clive D'Souza. Collaborating with their research groups has provided me with invaluable insights and opportunities. Their steadfast support, profound expertise, and guidance have significantly enriched my Ph.D. experience.

Lastly, but by no means the least, my heartfelt gratitude is reserved for my family. I owe a debt of gratitude to my parents, Isatay Syzdykbayev and Zhamal Syzdykbayeva. Their boundless love, unwavering belief in my capabilities, and steadfast encouragement have been the bedrock upon which all my accomplishments stand. Their teachings, both explicit and implicit, have guided me throughout my life and have been instrumental in shaping who I am today.

In this journey of research and self-discovery, every mentioned individual has played a pivotal role, and for that, I remain eternally grateful.

1.0 Introduction

1.1 Problem Statement

The increasing volume of geospatial data continues to surpass the processing capabilities of current computing systems. The McKinsey Global Institute reports that the accumulation of geospatial data reached around 1 petabyte in 2009, with an annual growth rate of 20% (Lee & Kang, 2015). This data, sourced from various platforms at impressive frequency, presents unique challenges necessitating rigorous study and examination. This includes studies related to data storage, quality assurance, analysis, and visualization (Praveen, Babu, & Rama, 2016). Dealing with this vast amount of geospatial data requires processing data more efficiently and with advanced analytical skills and tools.

Geospatial object detection plays a key role in geospatial data analysis and is used in a variety of applications, such as natural hazards monitoring, agriculture, and urban management, among others (Wang et al., 2022). The main task of geospatial object detection is to determine if a geospatial dataset includes objects of interest and locate their positions in the dataset and on the Earth's surface (Long, Gong, Xiao, & Liu, 2017). Current geospatial object detection approaches can be categorized into four types (Cheng & Han, 2016): template matching-based, knowledge-based, object-based image analysis, and ML-based. Of these four, this thesis focuses on knowledge-based detection and ML-based detection approaches because our two proposed geospatial object detection methods fit into these types.

The knowledge-based geospatial object detection approach translates the geospatial object detection task into hypothesis testing by establishing different knowledge and rules. There are two types of information used in hypothesis testing: geometrical and contextual. The hypothesis testing based on geometrical information uses knowledge about the geometrical characteristics of the geospatial object and the hypothesis testing based on contextual information employs relationships between geospatial objects and the background environment (Cheng & Han, 2016).

The ML-based geospatial object detection approach identifies classes of geospatial objects and provides their locations in the form of bounding boxes. This approach also includes ML-based instance segmentation which identifies classes of geospatial objects and provides their locations at a pixel level (Hafiz & Bhat, 2020). With an increase in the number of ML models and the expansion of geospatial benchmark datasets with labels (Helber, Bischke, Dengel, & Borth, 2019; Van Etten, Lindenbaum, & Bacastow, 2018), using ML models to detect geospatial objects has become popular, mainly due to the fact that recent ML-based geospatial object detection methods have achieved significant improvements (Hoeser, Bachofer, & Kuenzer, 2020; Hoeser & Kuenzer, 2020).

The term ‘geospatial object’ refers to the generalized form of landscape objects, such as water bodies, landslides, forests, and grasslands (Cheng & Han, 2016). Geospatial objects contain unclear boundaries and are part of the background environment, a reference to geospatial objects located in neighboring regions that do not belong to the class of the object of interest. Current geospatial object detection approaches suffer from several challenges such as scalability, which occurs due to large variations in size and shape of geospatial objects and occlusion, background clutter, illumination, and shadow, which may stem from the data collection process (Pham, Courtrai, Friguet, Lefèvre, & Baussard, 2020; Pun, Xia, & Lee, 2018; L. Zhou, Pan, Wang, & Vasilakos, 2017).

In this thesis we address the three following challenges out of those identified by Zhou et al., (2017) :

- Data noise. Noise can be introduced to traditional physics-based or ML models due to data sparsity issues, missing or inaccurate values, and outliers. Geospatial datasets typically contain noise due to various factors such as properties of remote sensing sensors and environmental, among other factors. For example, noise caused by environmental factors in satellite images can be caused by the presence of cirrus clouds (Qiu, Zhu, & Woodcock, 2020).
- Data labeling. Geospatial data labeling is a time-consuming and laborious process that needs expert knowledge in the geoinformatics field. To cope with this challenge, several alternative solutions, such as crowdsourcing, and special ML methods, such as active, transfer, and semi-supervised learning, have been developed (Settles, 2012). In addition, having complete and correct ground truth data will allow us properly evaluate performance of traditional physics-based models as well.
- Feature representation and selection. The process of selecting and aggregating different subsets of features at various levels, also known as feature engineering. The performance of traditional physics-based or ML models in geospatial object detection is dependent on the choice of the geospatial data representation or features. The selection of appropriate features is necessary when building models with improved performance. However, like geospatial data labeling, geospatial feature engineering is a laborious process that needs expert knowledge (Bengio, Courville, & Vincent, 2013).

1.2 Topological Data Analysis

Topological data analysis (TDA) is a field that has emerged from research in computational topology and data analytics (Chazal & Michel, 2021). Given a dataset as a set of points, TDA studies the structure and can give information about the shape, i.e., topological information, which includes shape of the geospatial object of interest. In comparison to the geometrical information in a dataset that is localized and more rigid, topological information in a dataset is multi-scale and global (Hensel, Moor, & Rieck, 2021). TDA includes various methods, the two most important of which are Persistent Homology (PH) and Mapper (Chazal & Michel, 2021). PH can obtain topological information from a dataset by studying the corresponding data connections and gaps. In addition, PH can describe the complicated structures of a dataset, such as loops and voids, that are not visible with other geometrical-based methods (Otter et al., 2017). Mapper can obtain topological information by representing a dataset using a graph (Singh et al., 2007). PH and Mapper are based on topological concepts, where large-scale distances are ignored but nearness between points are preserved.

PH has demonstrated its effectiveness in increasing ML-based model accuracy (Hensel et al., 2021). Currently, many works seek to use outputs of PH to achieve a fundamentally different view of datasets. These works are mostly focused on representing topological information with structured features that can be used as input features in ML models (Pun et al., 2018). With these works, the term 'topological machine learning' (Topological ML) has lately begun to appear as a field at the intersection of TDA and ML (Hensel et al., 2021). However, one shortcoming of current Topological ML models is that they do not take advantage of information about the location of the subset of points that form topological information in their representations. In other words, the representations in current Topological ML models can be used to state whether the dataset contains a circle or not with no information about the locations of the subset of points that form the circle. Information about the locations of the subset of points that form topological information can be vital in tasks such as object detection.

Focusing on the shape of geospatial objects is one of the key steps to improve geospatial data analysis (Lü et al., 2019). For example, the set of images in Figure 1.1 shows the shapes of geospatial objects such as lakes, landslide boundaries, and rivers. The set of points for each geospatial object in the figure was sampled from satellite images at different scales. Through these points, one can see that in addition to geometrical properties, geospatial datasets have unique shapes and contain topological information. This topological information is often not considered in geospatial data analysis.

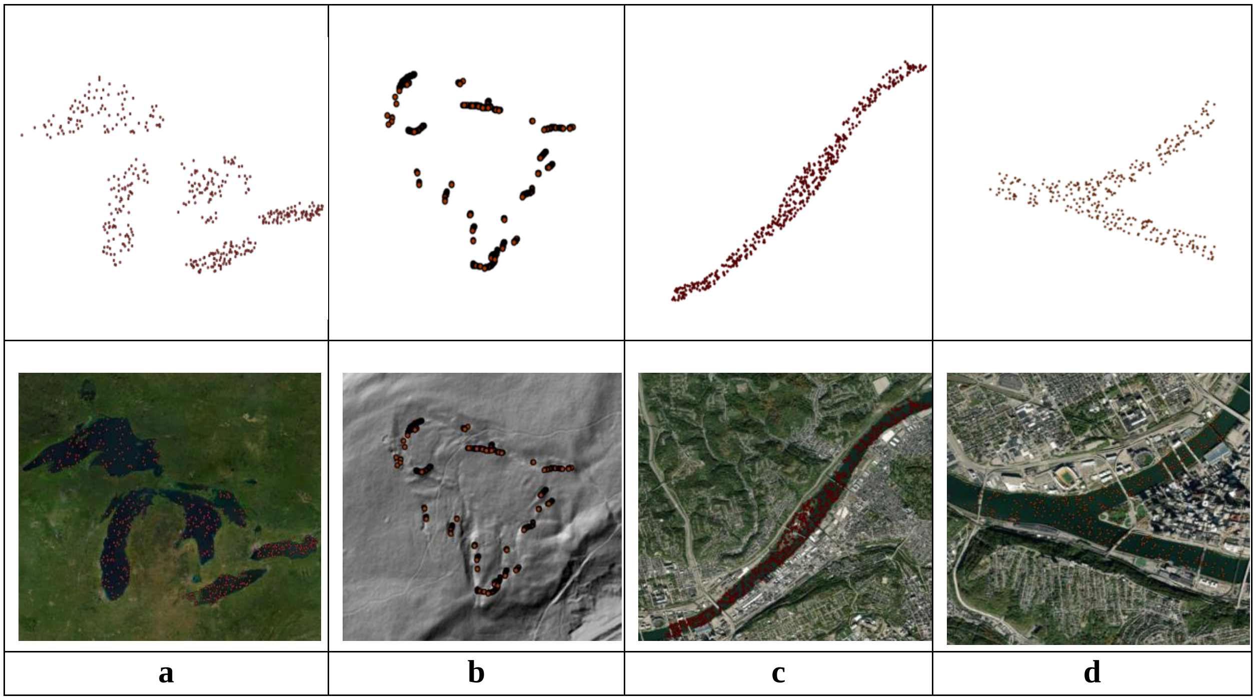


Figure 1.1 Geospatial objects with unique shapes: (a) the Great Lakes, shown as separate clusters of points; (b) a boundary of a landslide, shown as separate clusters of points with a circular shape; (c) a river, shown as a cluster of points with a straight-line shape

1.3 Proposed Research

This thesis explores the hypothesis that integrating topological information, which describes the shape of a geospatial dataset, can address the existing challenges in geospatial object detection and enhance detection accuracy. To test this hypothesis, three research questions are posed:

Research Question 1 (RQ1): Do filters based on topological information applied to candidate polygons resulted from PH improve geospatial object detection accuracy?

Research Question 2 (RQ2): Do filters based on combined topological, geometrical, and contextual information improve accuracy of topological knowledge-based (Topological KB) geospatial object detection methods?

Research Question 3 (RQ3): Can topological information derived from a geospatial dataset be represented such that:

- the output of the representation is a three-dimensional array or a multichannel image that can be input to ML-based geospatial object detection methods,
- the representation is not sensitive to geospatial data noise, and
- the representation, which includes location of the subset of points that form topological information, improve object detection accuracy?

1.4 Contributions

The thesis's contributions are:

- An evaluation of PH in the detection of landslides using LiDAR data.
- An algorithm to extract linear terrain features (LTFs) from Light Detection and Ranging (LiDAR)-derived digital terrain model (DTM).
- A Topological KB geospatial object detection method that utilizes topological information in addition to geometrical and contextual information.
- A Topological ML based geospatial object detection method where topological information is used as an additional input feature to ML models.
- Two algorithms to transform topological information into a multichannel image.

1.5 Thesis Structure

Chapter 2 presents an overview of current geospatial object detection approaches and TDA. Chapter 3 outlines the entire proposed research and the three research questions that will be addressed in subsequent chapters. In Chapter 4, RQ1, by examining the capabilities of PH applied to LiDAR data on landslides, is addressed. Chapter 5 presents a new algorithm for extracting LTFs. Chapter 6 addresses RQ2 by developing and evaluating a Topological KB geospatial object detection method. Chapter 7 addresses RQ3 by developing and evaluating a Topological ML. Chapter 8 provides conclusions and limitations of the thesis along with possible future research directions.

2.0 Background

This chapter provides an overview of two geospatial object detection approaches and TDA, including PH and Mapper.

2.1 Geospatial Object Detection

Geospatial datasets can be analyzed to detect geospatial objects of interest in them and locate their positions. Cheng & Han, (2016) divided object detection approaches into four categories: template matching-based, knowledge-based, object-based image analysis (OBIA-based), and ML-based. Of these four categories, our proposed geospatial object detection methods belong to knowledge-based object detection and ML-based object detection approaches, both of which are detailed further below.

2.1.1 Knowledge-Based Object Detection

The knowledge-based object detection approach, which includes various methods, transforms the geospatial object detection problem into the hypothesis-testing problem by creating different knowledge and rules (Cheng & Han, 2016). Hypothesis testing in current methods is based on geometrical information and contextual information (Figure 2.1). Knowledge-based object detection methods that employ geometrical information encode prior knowledge into shape models. For example, farmlands have straight boundaries and specific sizes. Hence, a hypothesis is that this geometrical information, i.e., boundary properties and sizes, can be used to detect farmlands. Knowledge-based object detection methods that use contextual information employ the relationships between geospatial objects and their background environment. In other words, the context is knowledge about how geospatial objects interact with their neighboring regions and other geospatial objects. For example, a hypothesis about landslides, which are located on a specific slope, is that the existence and degree of a slope can be utilized as a clue for landslide detection. Martha, Kerle, Van Westen, Jetten, & Kumar, (2011) detected landslides by creating rules based on prior knowledge about landslides. First, they identified candidate landslides, then performed a selection based on rules using geometrical and contextual information. In short, the main idea of knowledge-based object detection methods is to transform knowledge about detected geospatial objects to detection rules. If the rules are too general, they will cause false positives, otherwise, if the rules are too specific, they will cause false negatives (Cheng & Han, 2016).

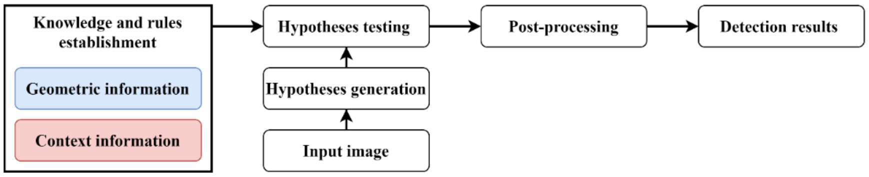


Figure 2.1 Knowledge-based object detection pipeline (Cheng & Han, 2016)

2.1.2 ML-Based Object Detection

The ML-based object detection approach, which includes various ML methods, uses features and labels to train a classifier that captures vital information from the input dataset. The inputs to the classifier are multichannel images and the outputs are predicted labels and boundaries of the geospatial objects (Cheng & Han, 2016). ML-based object detection methods provide classes of objects and location of each object in the form of bounding boxes (Figure 2.2a). Existing ML-based object detection methods are of two types: region proposal-based and regression-based (Li, Wan, Cheng, Meng, & Han, 2020).

Region proposal-based object detection methods consist of two steps. The first step creates a series of candidate regions that could include geospatial objects. The second step classifies the candidate regions into object classes or background environments and additionally fine-tune the coordinates of the bounding boxes (Li et al., 2020). Regression-based methods, designed to identify an object's location, address the task as a regression problem. In comparison with region proposal-based methods, regression-based methods apply one-stage object detectors and do not need to create candidate regions. For that reason, regression-based methods are efficient and simple (Li et al., 2020).

Besides ML-based object detection methods, ML-based instance segmentation methods provide classes of geospatial objects and locations of the detected geospatial objects at a pixel level (Figure 2.2b) (Hafiz & Bhat, 2020). In other words, the instance segmentation methods are applied to predict the mask and the class for each geospatial object in an input dataset. The difference between instance segmentation and semantic segmentation is that in semantic segmentation every pixel in the image is marked into a category class while in instance segmentation every pixel in the image is marked into instances of multiple categories. In other words, if an image contains several objects of one category, for example, several lakes, in semantic segmentation the pixel values for all lakes will be marked as "lake" while in instance segmentation the pixel values for each will be marked as "lake_1","lake_2", etc.

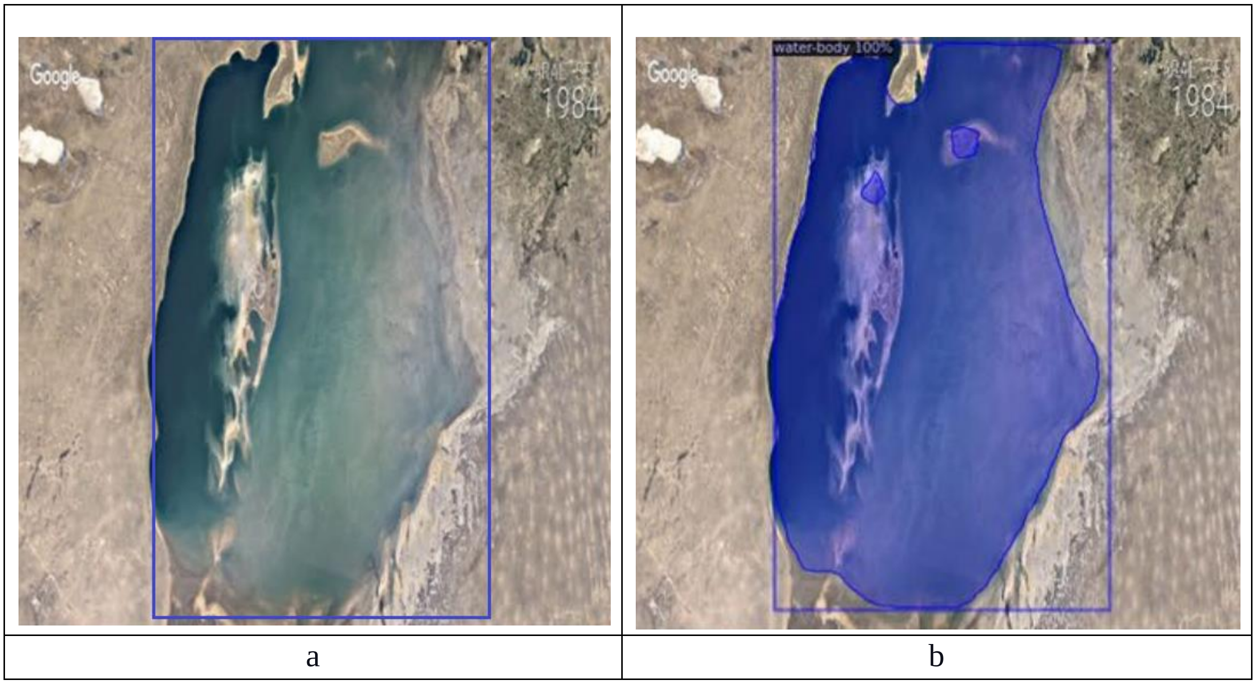


Figure 2.2 ML-based methods: (a) results of the ML-based object detection method; (b) results of the ML-based instance segmentation method

2.2 Topological Data Analysis

TDA encompasses a collection of effective algorithms that can be used for the investigation and quantification of the shape and structure of datasets to answer questions in a specific domain (Chazal & Michel, 2021). Instead of using only statistical descriptors in the analysis, that can mislead when analyzing objects with different shapes (Figure 2.3) (Matejka & Fitzmaurice, 2017), TDA analyzes data in a fundamentally unique way by exploring the underlying shape (Chazal & Michel, 2021). The two widely used methods in TDA are PH and Mapper; the theory and mathematics behind these two methods are presented in Appendix A.

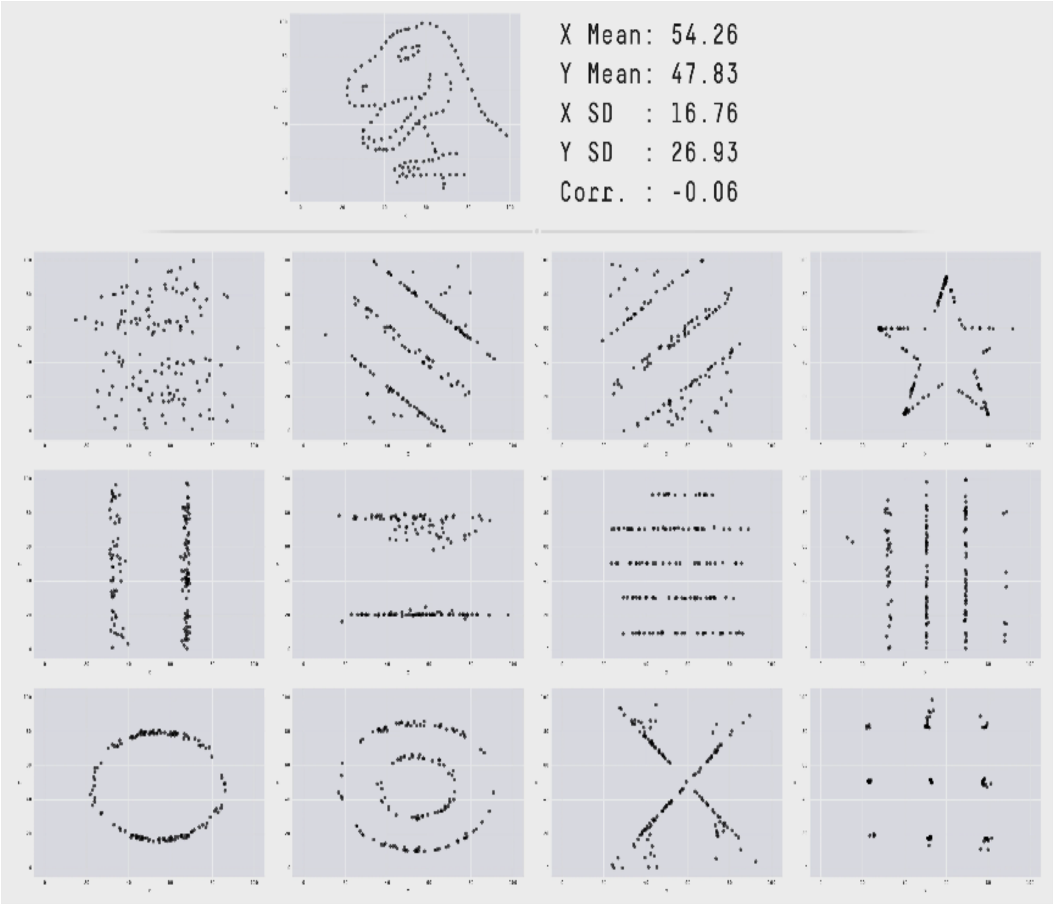


Figure 2.3 The point set datasets: each point set has different shape, but has same statistical descriptors (mean, standard deviation, and Pearson's correlation) (Matejka & Fitzmaurice, 2017)

PH is applied to obtain topological information from data by studying the corresponding data connections and gaps (Otter et al., 2017). PH utilizes all three properties of simplicial complexes (see Appendix A) to derive topological information at different dimensions. Chazal & Michel, (2021) presented the basic PH pipeline with four steps as follows (Figure 2.4) to derive topological information from a given dataset:

1. Input to PH a finite set of points with corresponding distance information (Carlsson, 2009). The distance metric depends on the application, and the choice of the correct metric is

essential. For example, protein data metrics can be measured in nanometers, and metrics for satellite image analysis can be measured in meters.

2. Construct a nested sequence of simplicial complexes from the set of points using different values of r .
3. Derive topological information from the nested sequence of simplicial complexes. This step consists of two functions:

   A. *homology group* returns topological information given the simplicial complex that was constructed using r.

   B. *persistent homology* utilizes *homology group* with a different value of r, and records each change. In other words, when r is changed, the topological information associated with the newly created simplicial complexes is also changed, and this second function records these changes.
4. Use the extracted topological information as a feature or descriptor for the dataset to assist in better understanding the dataset. This topological information can be visualized or can be used as a feature in ML models.

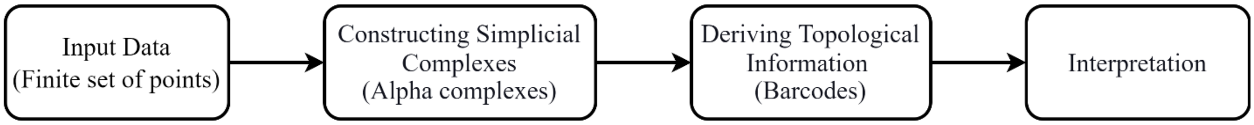
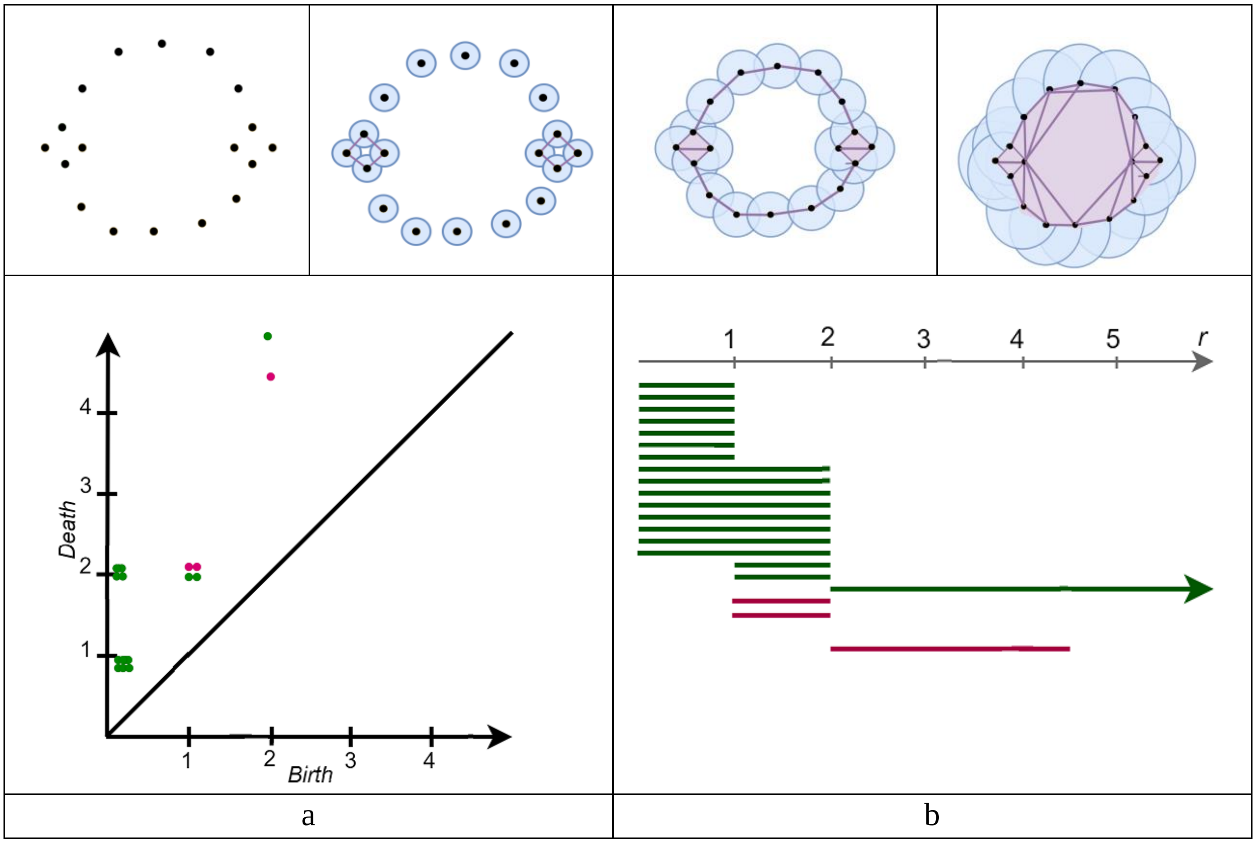


Figure 2.4 PH pipeline (Chazal & Michel, 2021)

Mapper is based on the idea of preserving information at close distances and discarding information on large distances. In comparison with PH, where a series of nested subcomplexes is created and analyzed to derive topological information, Mapper is designed to produce a single low-dimensional simplicial complex in a graph from which information about the underlying data can be extracted (Singh et al., 2007). The input to Mapper, similar to PH, is a set of points P; detailed information about Mapper can be found in (Ristovska & Sekuloski, 2019; Singh et al., 2007).

2.3 Topological ML

Topological ML uses topological features as input which includes information about the shape. The most common representations of output of PH are persistence diagrams (PD) and barcodes. A PD is a plot with the birth and death times (Figure 2.5a), and the barcode is a series of lines, one for each feature, stretching from birth to death times (Figure 2.5b). However, PD and barcodes are not structured and cannot be used directly as input data to ML models (Otter et al., 2017; Pun et al., 2018). To use output of PH, i.e., topological information, as input data to ML models, the information needs to be converted into structured features; detailed information about how topological information is converted into structured features for Topological ML can be found in Hensel et al., 2021; Pun et al., (2018). Below, we discuss some ways of converting topological information into structured features as input to ML models.



a

b

Figure 2.5 The output of PH: (a) PD, (b) barcode

A simple way of using topological information in ML models is summary statistics, such as total persistence, mean, median, and standard deviation of a persistence diagram (Pereira & De Mello, 2015; Syzdykbayev & Karimi, 2020). These representations are useful in simple ML models, but they are often not directly applicable to complex ML models and require more expressive representations. Hensel et al., (2021) discussed two methods that help use topological information as input to ML models: vector-based and kernel-based. Of these two methods, we focus on vector-based methods because they are used to transform topological information into a multichannel image in this thesis. Vector-based methods transform topological information into vectors of different dimensions. The most common vector-based methods are persistence landscape (Bubenik, 2015) and persistence images (Adams et al., 2017).

Persistence landscape is a vector representation of topological information (Bubenik, 2015). To calculate persistence landscape, first, a rotation of persistence diagram needs to be performed (Figure 2.6a and Figure 2.6b). Next, the persistence landscape can be calculated by using the following equation:

$$\lambda(k,t) = \ kmax\{f_{(b_i d_i)}(t)\}_{i\in I} \tag{2.1}$$

where

$$f_{(b,d)}(t) = max\ \{0, min\ \{t-b, d-t\}\} \tag{2.2}$$

and $kmax$ is the k-th largest value of $f_{(b,d)}(t)$ (Bubenik, 2015). In other words, persistence landscapes involve computing the area of influence of each point in a persistence diagram. The basis of the k-th persistence landscape can be formed by using a connected shaded region with at least k intersections (Figure 2.6c).

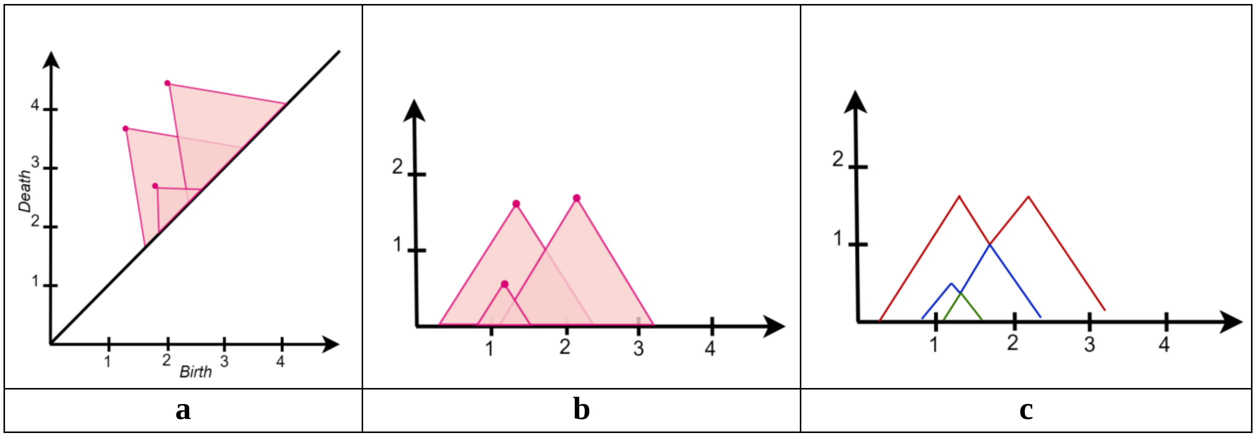


Figure 2.6 Persistence landscape: (a) PD, (b) rotated PD, (c) persistence landscape

The idea of persistence images is to convert a PD into vectors while maintaining an interpretable connection to the original persistent diagram. In comparison to persistence landscapes, persistence images transform a diagram into a matrix. As its name states, this representation can be represented as a single-channel image (Adams et al., 2017). To get persistence images similar to persistence landscapes, first, a transformation needs to be performed on a persistent diagram (Figure 2.7a) from birth-death pairs into birth-persistence coordinates (x, y), where $x = b$ and $y = d - b$ (Figure 2.7b). Each point is then represented by a Gaussian function g, centered at (x, y):

$$g(x,y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{[(x-b)^2+(y-d)^2]}{2\sigma^2}\right)} \tag{2.3}$$

Then for each birth-death pair, a persistence surface needs to be computed:

$$p(x,y) = \sum_{i=1}^{N} f(x,y)\, g_i(x,y) \tag{2.4}$$

where the weight function $f(x,y)$ is necessary to account for the diagonal and $f(x,y) = (y/y_{max})$, where y_{max} is the maximum value of the filtration parameter used in the original persistence calculation. Next, the persistence surface p is transformed into the persistence image by discretizing it into a grid and integrating persistence surface p into a grid (Figure 2.7c).

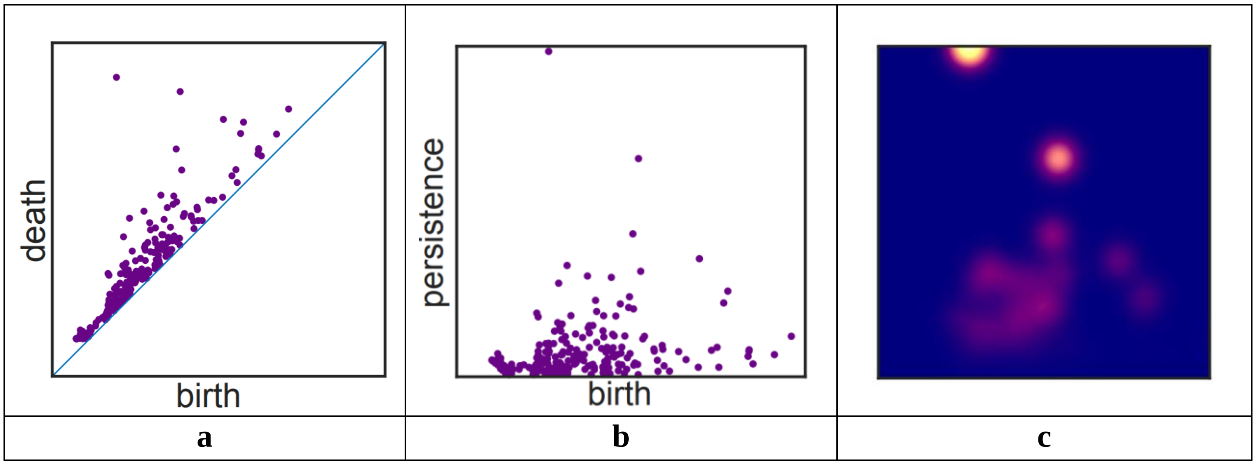


Figure 2.7 Persistence image (a) PD, (b) rotated PD, (c) persistence image (Adams et al., 2017)

3.0 Proposed Research

This thesis is focused on testing the hypothesis that integrating topological information, which describes the shape of a geospatial object, can overcome the challenges with existing geospatial object detection approaches and enhance detection accuracy. To test this hypothesis, research that addresses three research questions is proposed.

3.1 Geospatial Object Detection: PH-Based Method

In Chapter 4, we explore the role of topological information in detecting geospatial objects by addressing the following research question:

> **RQ1:** Do filters based on topological information applied to candidate polygons resulted from PH improve geospatial object detection accuracy?

We will conduct experiments where PH is used to detect landslides, characterized by circular structures. The contributions of Chapter 4 are:

- Development of a PH-based method designed specifically to be used on LiDAR-derived DTM to detect landslides.
- Evaluation of the PH-based method on detecting landslides.

There are different geospatial objects that have clear circular structures such as craters or hurricanes and we selected landslides for the following reasons:

- Landslides are potentially catastrophic geologic events that can take lives, cause economic loss, and have a negative environmental impact.
- Boundaries of existing landslides are usually acquired manually by using remote sensing datasets such as satellite imagery or LiDAR, a time-consuming and labor-intensive process. While there are automatic and semi-automatic techniques available for landslide detection, these techniques often come with their own set of limitations and do not guarantee high accuracy across diverse terrains and conditions.
- Landslides leave discernible signs on the landscape, such as altered slope shape, position, or surface appearance (see Figure 3.1).

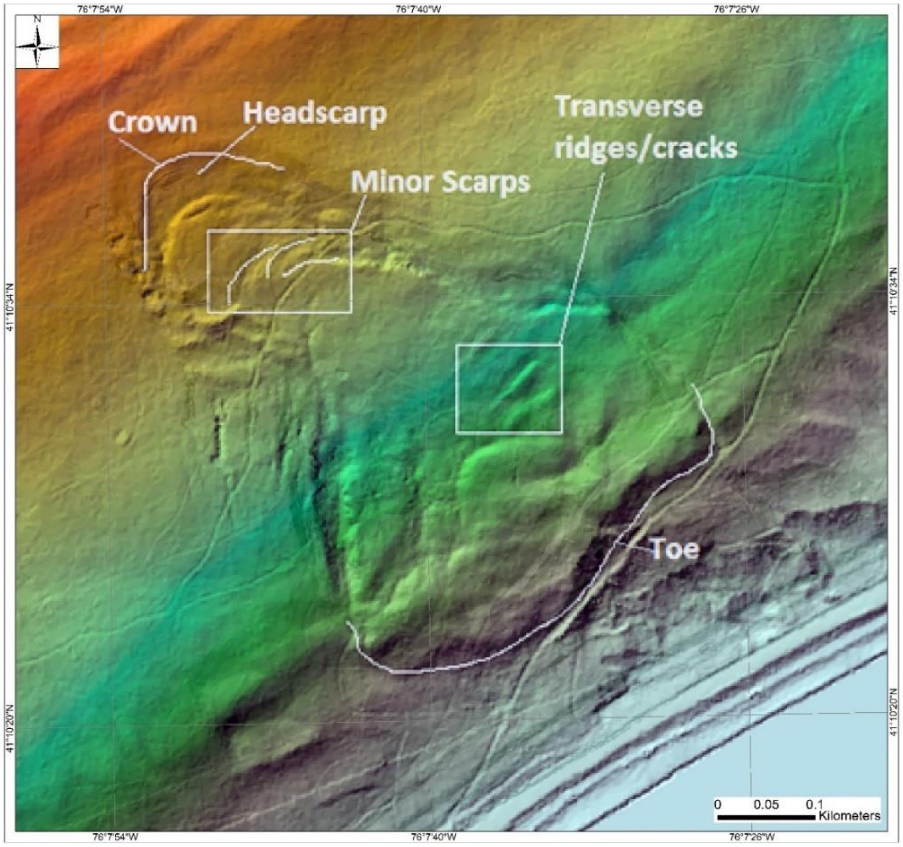


Figure 3.1 LiDAR-derived 1-meter DTM shaded relief map from Pennsylvania with annotations identifying morphological expressions of landslides (Syzdykbayev et al., 2020b)

3.2 New LTFs Extraction Algorithm

In Chapter 5, we present a new LTFs extraction algorithm considering that LTFs are required to create PH results, i.e., candidate polygons, and that current LTFs extraction algorithms have shortcomings. This new algorithm emulates the human ability to perceive and decipher terrain feature information through perceptual cues like light angles, shadows, patterns, and textures. Such cues, rooted in shaded relief which offers a visual portrayal of real-world terrain, are instrumental in pinpointing terrain features, especially linear ones as highlighted by Syzdykbayev et al., (2020a). The contributions of Chapter 5 are:

- Development of a new LTFs extraction algorithm.
- Evaluation of a new LTFs extraction algorithm using synthetically-generated and real-word datasets.

3.3 Geospatial Object Detection: Topological KB Method

In Chapter 6, we explore the role of topological information on knowledge-based geospatial object detection methods by addressing the following research question:

> **RQ2:** Do filters based on combined topological, geometrical, and contextual information improve accuracy of Topological KB geospatial object detection methods?

We will conduct a set of experiments where topological, geometrical, and contextual information is used as input in a Topological KB geospatial object detection method for selecting candidate polygons. The contributions of Chapter 6 are:

- Development and evaluation of a Topological KB geospatial object detection method that uses topological information alongside geometrical and contextual information.
- Creation of a list of geometrical- and contextual-based rules tailored for landslide detection using the Topological KB geospatial object detection method.
- An in-depth analysis to identify the specific rule from the created list that is best suited for landslide detection. This is achieved through extensive experimentation, applying all possible rules to ascertain the most effective one for landslide detection.

3.4 Geospatial Object Detection: Topological ML-Based Method

In Chapters 7, we explore the role of topological information, represented as features, in ML-based geospatial object detection methods by addressing the following research question:

> **RQ3:** Can topological information derived from a geospatial dataset be represented such that:
>
> - the output of the representation is a three-dimensional array or a multichannel image that can be input to ML-based geospatial object detection methods,
> - the representation is not sensitive to geospatial data noise, and
> - the representation, which includes location of the subset of points that form topological information, improve object detection accuracy?

We will develop algorithms to transform topological information into a multichannel image and conduct experiments where the multichannel image is used as a feature for a Topological ML geospatial object detection method. The contributions of Chapter 7 are:

- Development of a new Topological ML geospatial object detection method, where topological information is harnessed as an auxiliary input feature.
- Development of two new algorithms designed to transform topological information into a multichannel image.

4.0 Geospatial Object Detection: PH-Based Method

4.1 Methodology

The method for geospatial object detection using PH takes data in the form of LiDAR-derived DTM as input and produces the boundaries of detected geospatial objects as output. The method uses only one type of information, topological, and follows the workflow shown in Figures 4.1 and 4.2. The first step of the workflow involves extracting linear features from DTM. The second step involves creating candidate polygons from the LTFs using PH. The third step involves selecting candidate polygons, from those found by PH, whose boundaries satisfy the detection rules that are based only on topological information.

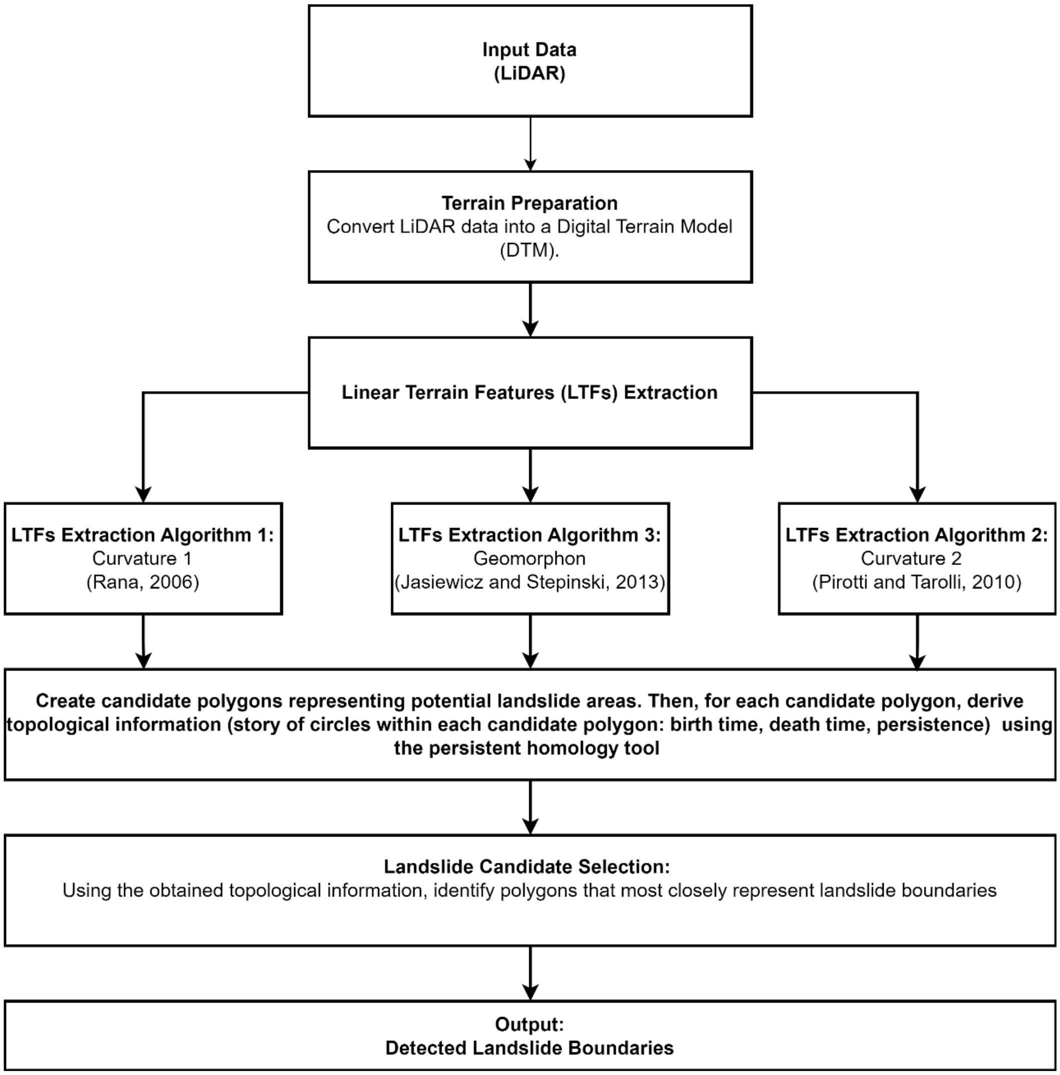

Input Data
(LiDAR)
Terrain Preparation
Convert LiDAR data into a Digital Terrain Model (DTM).
Linear Terrain Features (LTFs) Extraction
LTFs Extraction Algorithm 1:
Curvature 1
(Rana, 2006)
LTFs Extraction Algorithm 3:
Geomorphon
(Jasiewicz and Stepinski, 2013)
LTFs Extraction Algorithm 2:
Curvature 2
(Pirotti and Tarolli, 2010)
Create candidate polygons representing potential landslide areas. Then, for each candidate polygon, derive topological information (story of circles within each candidate polygon: birth time, death time, persistence) using the persistent homology tool
Landslide Candidate Selection:
Using the obtained topological information, identify polygons that most closely represent landslide boundaries
Output:
Detected Landslide Boundaries

Figure 4.1 Workflow of the PH method of geospatial object detection

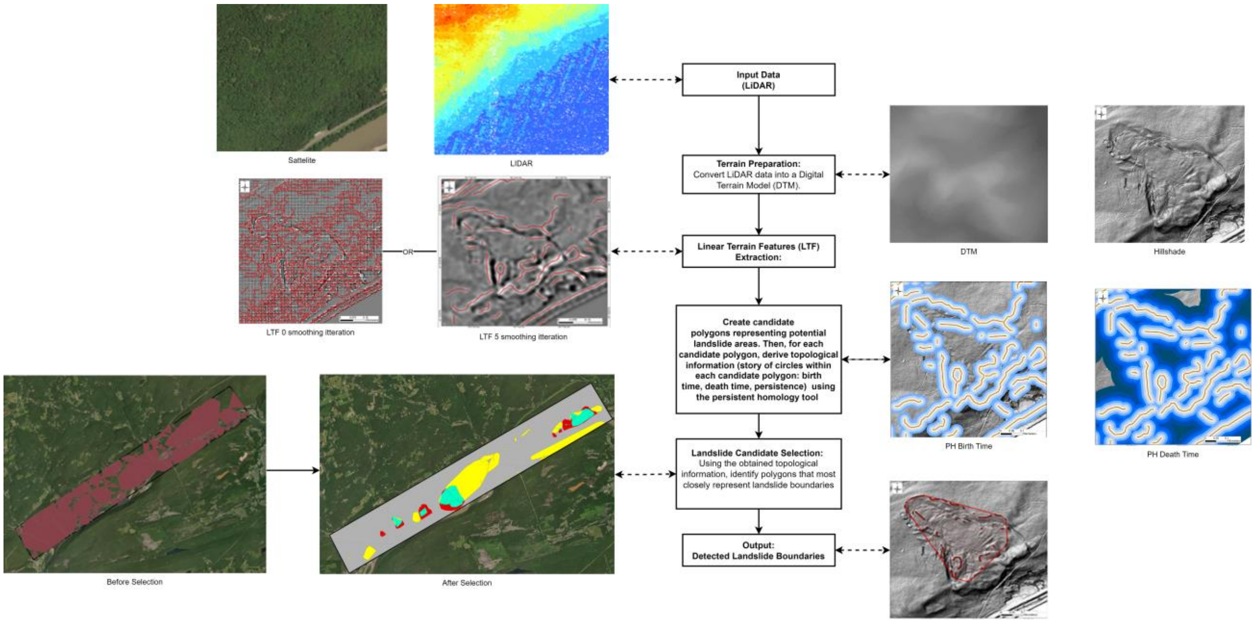

Sattelite
LiDAR
Input Data (LiDAR)
Terrain Preparation: Convert LiDAR data into a Digital Terrain Model (DTM).
DTM
Hillshade
OR
LTF 0 smoothing itteration
LTF 5 smoothing itteration
Linear Terrain Features (LTF) Extraction:
Create candidate polygons representing potential landslide areas. Then, for each candidate polygon, derive topological information (story of circles within each candidate polygon: birth time, death time, persistence) using the persistent homology tool
PH Birth Time
PH Death Time
Before Selection
After Selection
Landslide Candidate Selection: Using the obtained topological information, identify polygons that most closely represent landslide boundaries
Output: Detected Landslide Boundaries

Figure 4.2 Workflow of the PH method of geospatial object detection, illustrated with images representing data at each step

4.1.1 Extracting LTFs from DTM

The process of extracting linear features, often referred to as "edge detection" or "line-finding", is frequently used in computer vision (Szeliski, 2010a). Linear features typically form the boundary lines between areas of different textures, intensities, or colors in an image. These lines usually highlight rapid intensity changes within a small region of the image and are crucial for conveying significant visual information such as shapes of objects. They often provide essential semantic cues related to surface alterations, depth transitions, changes in surface reflectance, and illumination discontinuities (Szeliski, 2010a)

Morphological expressions of landslides can be characterized as a collection of small LTFs such as ridges and scarps (Figure 3.1) and extracting them from DTM would identify landslide boundaries. Considerable research has been conducted on detection of terrain morphology features including ridges and scarps. Rana (2006) proposed a curvature-based semi-automated iterative channel and ridge identification algorithm that is simple and provides reliable results. The algorithm can identify ridges and scarps but requires determination of a threshold value (Curvature 1). Pirotti & Tarolli, (2010) applied multiplication of the standard deviation as a threshold value to identify ridges and channels (Curvature 2). Jasiewicz & Stepinski, (2013) proposed a new algorithm to identify landform elements called "Geomorphon". This algorithm does not require a threshold value and is based on the principle of pattern recognition.

To extract LTFs related to landslides, a LiDAR-derived DTM was used as input. As in any computer vision (CV) based object detection task, the accuracy of any LTFs extraction algorithm depends on the DTM pixel size. Another parameter is the number of smoothing iterations. With a high number of smoothing iterations, LTFs may be averaged out and treated as flat surfaces, without smoothing iterations, geospatial data noise in DTM can increase false-positive rates, resulting in extracting LTFs that do not exist (see Figures 4.2 b1 and c1). To address these issues, we conducted an experiment where different pixel sizes and various numbers of smoothing iteration were used in Curvature 1, Curvature 2, and Geomorphon LTFs extraction algorithms. The total number of combined parameters used in the experiment is 54, as shown in Table 4.1.

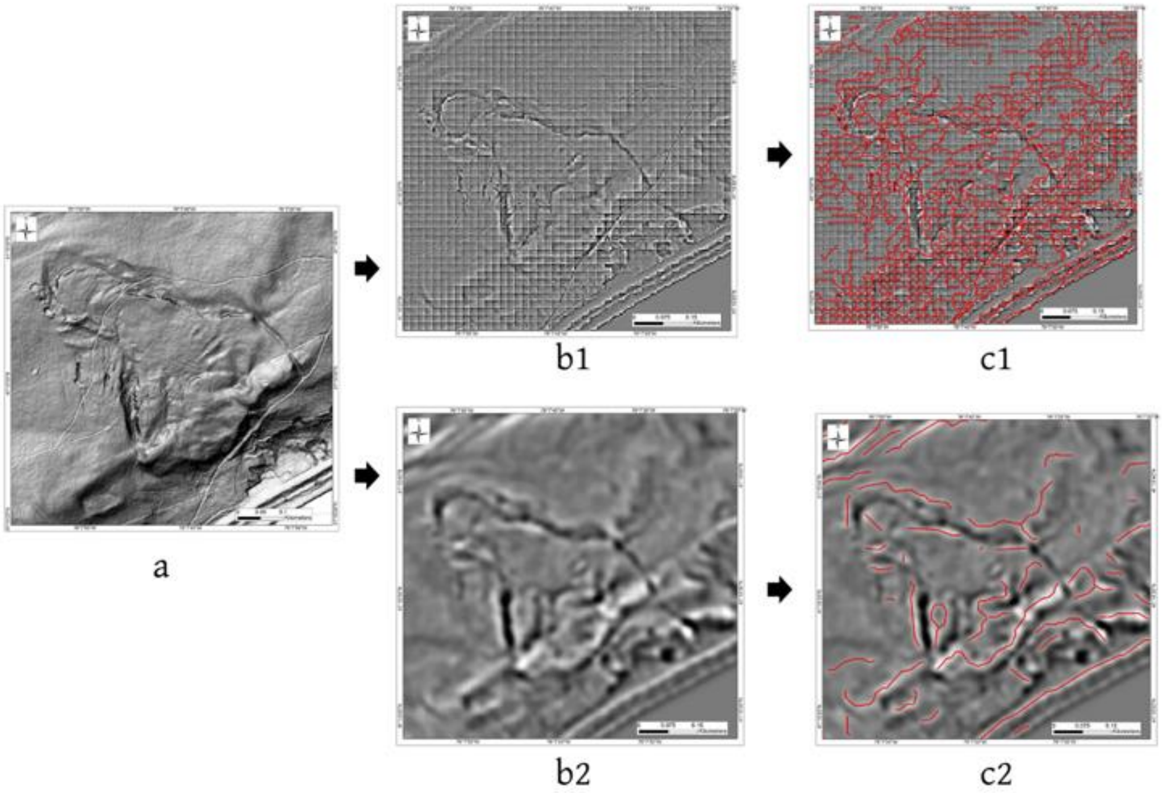


Figure 4.3 (a) shaded relief surface, (b1) curvature of the surface with pixel size 5 meters, (b2) curvature of the surface with 5 times smoothing iterations and with pixel size 5 meters, (c1, c2) extracted LTFs overlaying the curvature of the surface (Syzdykbayev et al., 2020b)

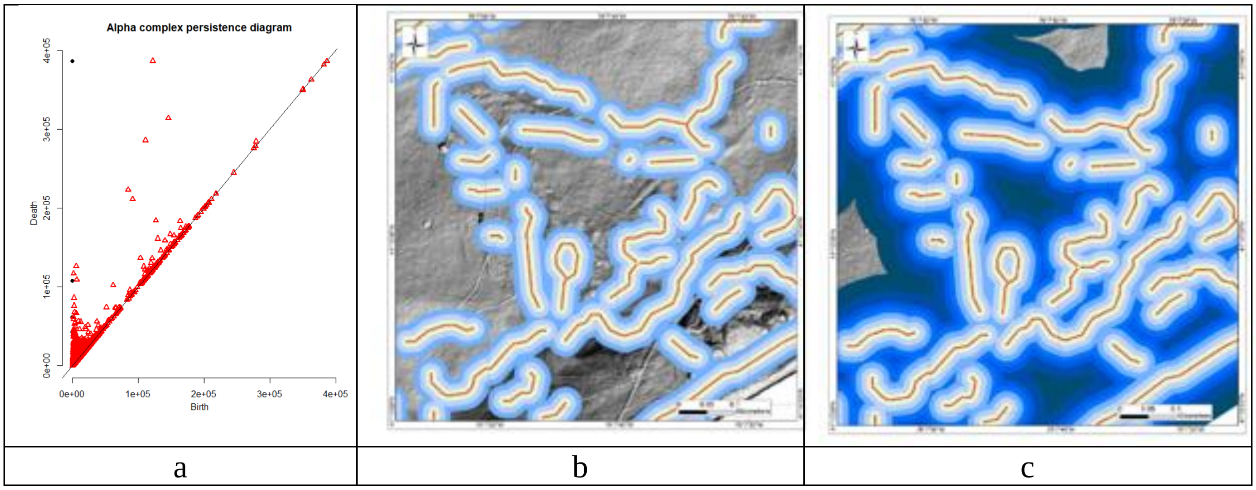
Table 4.1 Parameters used in the experiment

Parameter name	**Parameters**
DTM pixel size	1m, 5m, 10m
Smoothing iteration	0, 2, 5, 10, 15, 20
LTFs extraction algorithm	Curvature using a threshold (Rana, 2006) (Curvature 1)
	Curvature using a threshold calculated from the multiplication of standard deviation (Pirotti & Tarolli, 2010) (Curvature 2)
	Geomorphon (Jasiewicz & Stepinski, 2013)

The difference between Curvature 1 and Curvature 2 algorithms is in how their threshold values to derive LTFs are obtained. For Curvature 1 the threshold value was manually set for each study area. For Curvature 2 the threshold value was calculated by using a standard deviation of the all-pixel values (Pirotti & Tarolli, 2010).

4.1.2 Creating Candidate Polygons through PH

In the second step of the workflow, each of the 54 extracted LTFs was separately converted into a set of points and used as input to PH, as shown in Figure 4.3 (c1, c2). The output of PH is a set of points with topological information that includes the birth and death times of connected components and circles. This topological information is visualized in a PD, as presented in Figure 4.4 (a), and shows the birth time (Figure 4.4 b, light blue) and death time (Figure 4.4 c, dark blue) of the detected circle.



a

b

c

Figure 4.4 (a) persistence diagram of the points, (b) birth time (appearance) of the circles (light blue), (c) death time of the circle (dark blue) (Syzdykbayev et al., 2020b)

4.1.3 Selection Based on Topological Information

In the third step of the workflow, rules based on topological information were created to select landslide boundaries. Landslides typically have a circular shape and come in various sizes. LTFs are formed around a landslide boundary, and these LTFs usually are not connected (Figure 4.4 b). The detection rules are based on threshold values for birth time and lifetime. The birth time value is associated with the distance between LTFs that form landslide boundaries, and the lifetime value is associated with the size and shape of a landslide.

According to Cheng & Han, (2016) in object detection, if rules and threshold values are too general, they will cause false positives; on the other hand, if rules are too specific, they will cause false negatives. Since landslides have various sizes, the rules created to detect large landslides will fail to detect small landslides. The three rules created to identify landslides based on size are as follows:

- Small landslides with a small distance between LTFs (Figure 4.5 a), meaning that the threshold value that was set for both the lifetime of the circle and the distance between points that create the circle are small.
- Medium landslides with medium birth times and lifetimes, see Figure 4.5 (b).
- Large landslides with large birth times and lifetimes, see Figure 4.5 (c).

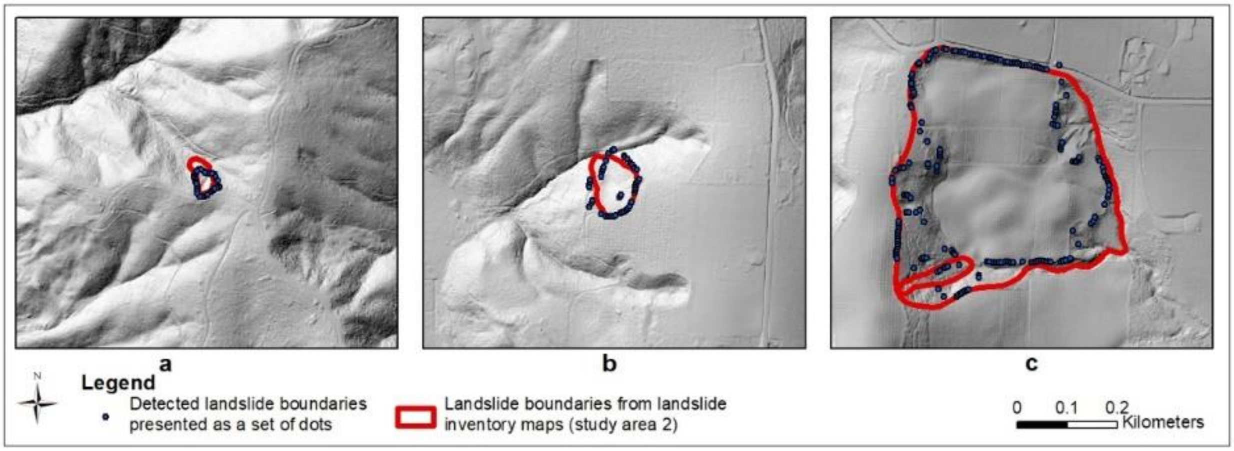

a
b
c
N
Legend
● Detected landslide boundaries presented as a set of dots
Landslide boundaries from landslide inventory maps (study area 2)
0
0.1
0.2
Kilometers

Figure 4.5 Detected landslides: (a) small, (b) medium, and (c) large. All overlaid-on landslides from landslide inventory maps (Syzdykbayev et al., 2020b)

4.2 Datasets and Study Area

The input dataset used in the experiment was LiDAR-derived DTM. Airborne LiDAR is a remote-sensing method that is used to acquire digital representations of a topographic surface. LiDAR can penetrate terrain that is covered in vegetation and thus can provide quantitative descriptions of a topographic surface in heavily vegetated areas (Figures 4.6 a and b). This ability provides an advantage over other methods, such as optical aerial or satellite images, which are based on visual understanding and are not able to penetrate vegetation canopy (Figure 4.6 c) and is specifically important when attempting to detect landslides in heavily vegetated areas (Guzzetti et al., 2012a). DTM is derived from LiDAR using filters to remove all the points that do not return from the ground, since DTM is a bare-earth representation of a terrain that does not include natural or man-made objects, such as vegetation and buildings (Figure 4.6 a).

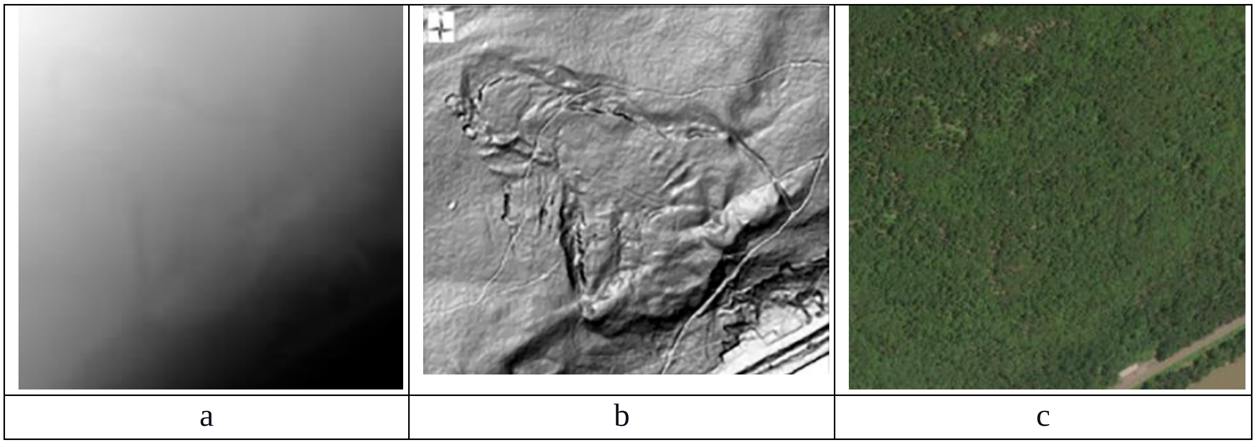


Figure 4.6 a) LiDAR-derived 1-meter DTM, b) shaded relief derived from DTM, and c) RGB satellite image of the same area (Syzdykbayev et al., 2020b)

The output of the experiment, detected landslides, was compared to a landslide inventory map, which is a collection of recorded locations of landslides with additional information such as date of occurrence and types of landslides (Guzzetti et al., 2012b).

The experiment was conducted in five study areas. The criteria for selection of the study areas were high landslide density, availability of LiDAR data, and availability of a landslide inventory map. The study areas are located in four states, Pennsylvania, Oregon, Colorado, and Washington in which both LiDAR data and a landscape inventory map in the regions with high landslide susceptibility exist (Figure 4.7). The dataset for each study area was obtained from the respective state government official website for Pennsylvania, Oregon, Colorado, and Washington. Most of the data is open source, except for the data for Colorado, which requires a formal request to download data.

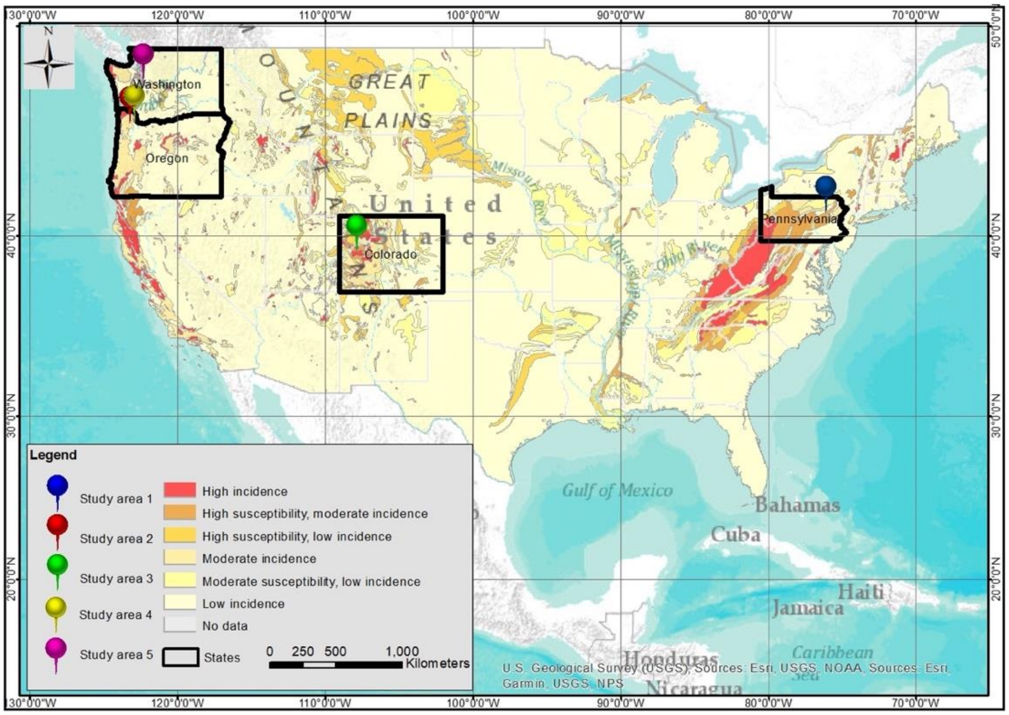


Figure 4.7 Landslide susceptibility maps in four states and for five study areas: Pennsylvania, Oregon, Colorado, and Washington (Syzdykbayev et al., 2020b)

Table 4.2 Characteristics of study areas (Syzdykbayev et al., 2020b)

	Study Area 1	**Study Area 2**	**Study Area 3**	**Study Area 4**	**Study Area 5**
Location (latitude, longitude)	41°12′39″N 76°03′44″W	45°34′0″N 123°11′0″W	39°10'44.6"N 107°50'58"W	45°42′0″N 122°53′0″W	47°36′28″N 122°20′6″W
State	Pennsylvania	Oregon	Colorado	Oregon	Washington
Area (square meters)	25,572,981	135,587,076	167,225,478	134,963,016	216,603,450
Landslide Area (square meters)	2,140,041	26,283,520	42,828,063	61,104,222	21,470,606
Percentage of Landslide Area	8.37%	19.38%	25.61%	45.27%	9.91%
Number of Landslides	7	738	206	1664	783
Elevation range (meters)	from 151 to 471	from 48 to 548	from 1,985 to 3,174	from 7 to 521	from 0 to 159
Slope range (degrees)	from 0 to 71.8	from 0 to 84.22	from 0 to 74.7	from 0 to 89.41	from 0 to 88.6

Table 4.3 Characteristics of input data (Syzdykbayev et al., 2020b)

	Study Area 1	Study Area 2	Study Area 3	Study Area 4	Study Area 5
LIDAR (or LIDAR derived DTM)					
Acquisition time	2006 - 2008	2007	2015 - 2016	2007	2000-2005
Source	Pennsylvania Spatial Data Access	State of Oregon Department of Geology and Mineral Industries	Colorado Geological Survey	State of Oregon Department of Geology and Mineral Industries	Puget Sound LiDAR Consortium
Horizontal ground resolution	1 meter	1 meter	1 meter	1 meter	1.8 meters
Existing landslides					
Acquisition time	2019	2019	2015	2019	2017
Source	(Karimi et al., 2019)	Gales Creek quadrangle Oregon’s State-wide Landslide Information Database	Colorado Geological Survey	Dixie Mountain quadrangle Oregon’s State-wide Landslide Information Database	Washington State Department of Natural Resources web portal
Acquisition method	Detecting visually LiDAR-derived DTM. Mimics protocol by Burns et al., (2008)	Compiling landslide inventory data created by using LiDAR and protocol by Burns et al., (2008)	Compiling landslide information digitized from 1:24 000-scale maps published in geologic hazard maps of Colorado	Compiling landslide inventory data created by using LiDAR and protocol by Burns et al., (2008)	Compiling landslide inventory data through different methods and scales

Study Area 1, marked with a blue pin in Figure 4.7 and shown also in Figure 4.8, is located in Luzerne County, Pennsylvania (latitude 41°12′39″N and longitude 76°03′44″W). In an area of 25,572,981 square meters, there are seven identified landslides, and the combined area of mapped landslides is 2,140,041 square meters. The elevation ranges from 151 meters to 471 meters, with a standard deviation of 100 meters. The slope inclination in the area ranges from 0 to 71 degrees (Table 4.2). The input data for Study Area 1, LiDAR data for Pennsylvania from 2006 to 2008, are publicly available through Pennsylvania Spatial Data Access (Access, n.d.). The data on existing landslides were obtained from the landslide inventory database for north-eastern Pennsylvania (Table 4.3).

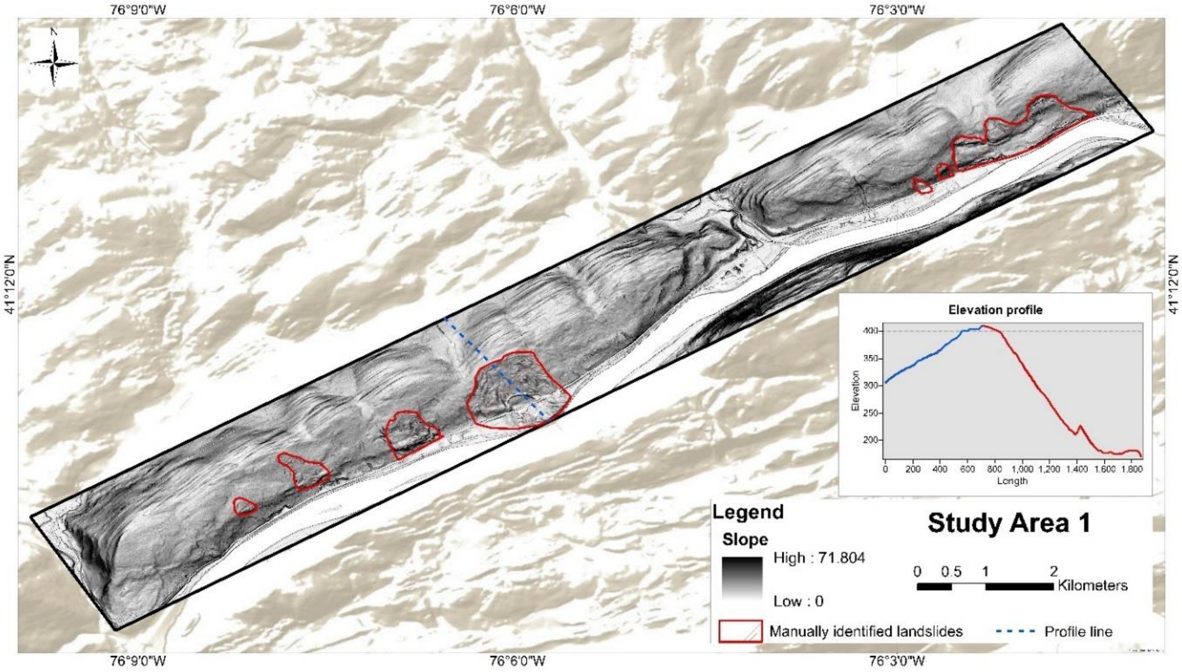


Figure 4.8 Study Area 1(Pennsylvania) with locations of manually identified landslides (Syzdykbayev et al., 2020b)

Study Area 2, marked with a red pin in Figure 4.7 and shown also in Figure 4.9, is located in Oregon (latitude 45°34′0″N and longitude 123°11′0″W). The total area is 135,587,076 square meters. There are 738 documented landslides, and the combined area of mapped landslides is 26,283,520 square meters. The elevation ranges from 48 meters to 548 meters, with a standard deviation of 110 meters. The slope inclination in the area ranges from 0 to 84 degrees (Table 4.2). The data for Study Area 2 and Study Area 4 were obtained from the State of Oregon Department of Geology and Mineral Industries public file transfer protocol (FTP) site. The data on existing landslides were obtained from Oregon's Statewide Landslide Information Database (Burns & Madin 2009)

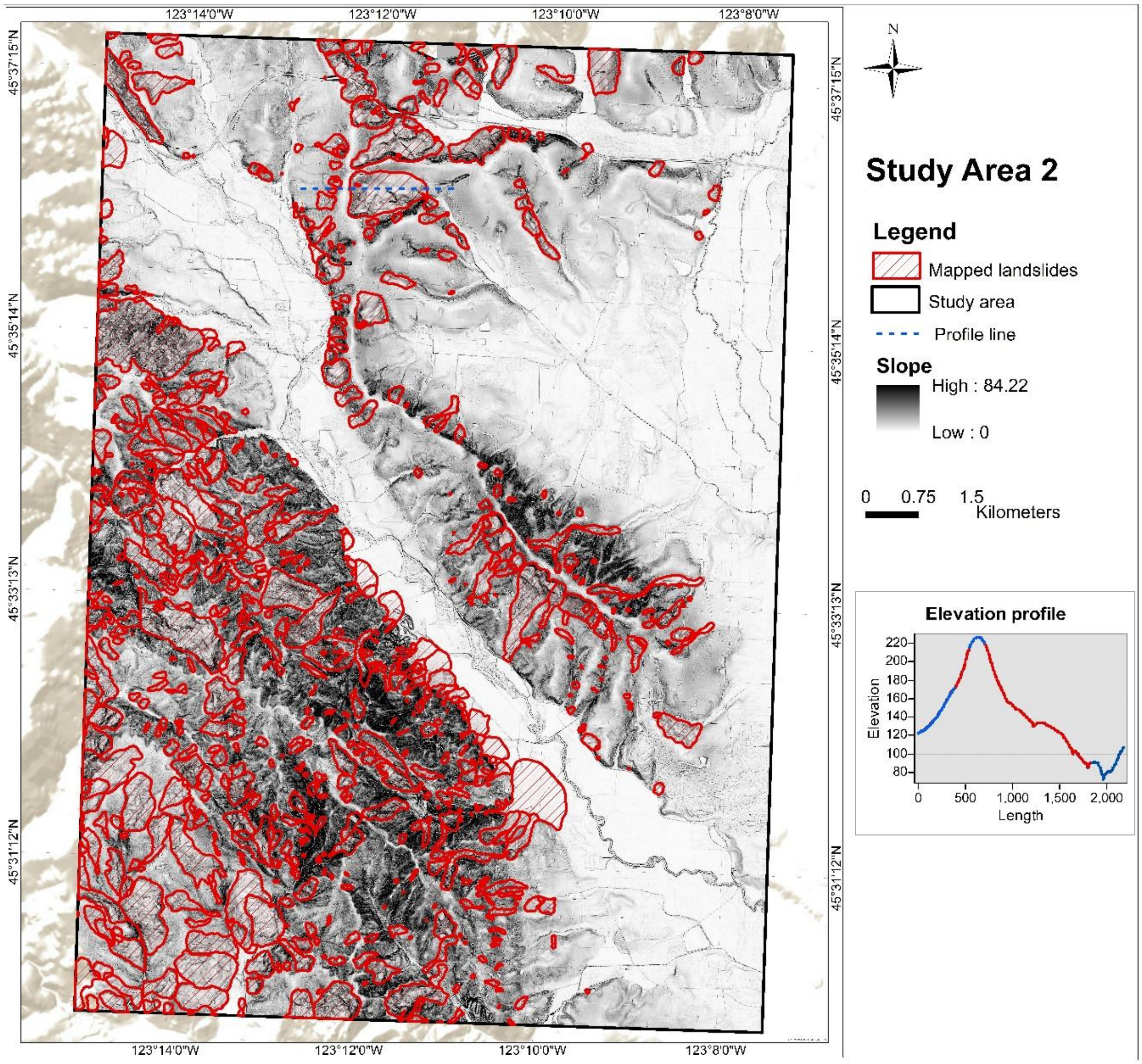


Figure 4.9 Study Area 2 (Oregon) with locations of previously mapped landslides (Syzdykbayev et al., 2020b)

Study Area 3, marked with a green pin in Figure 4.7 and shown also in Figure 4.10, is located in Mesa County, Colorado (latitude 39°10'44.6"N and longitude 107°50'58.0"W). The total area is 167,225,478 square meters. There are 206 documented landslides, and the combined area of landslides is 42,828,063 square meters. The elevation ranges from 1,985 meters to 3,174 meters, with a standard deviation of 269 meters. The slope inclination in the area ranges from 0 to 74 degrees (Table 4.2). The data for Study Area 3 was obtained by request/permission from the Colorado Geological Survey (CGS). The data on existing landslides with scarps and deposits were obtained from the CGS web portal (Table 4.3).

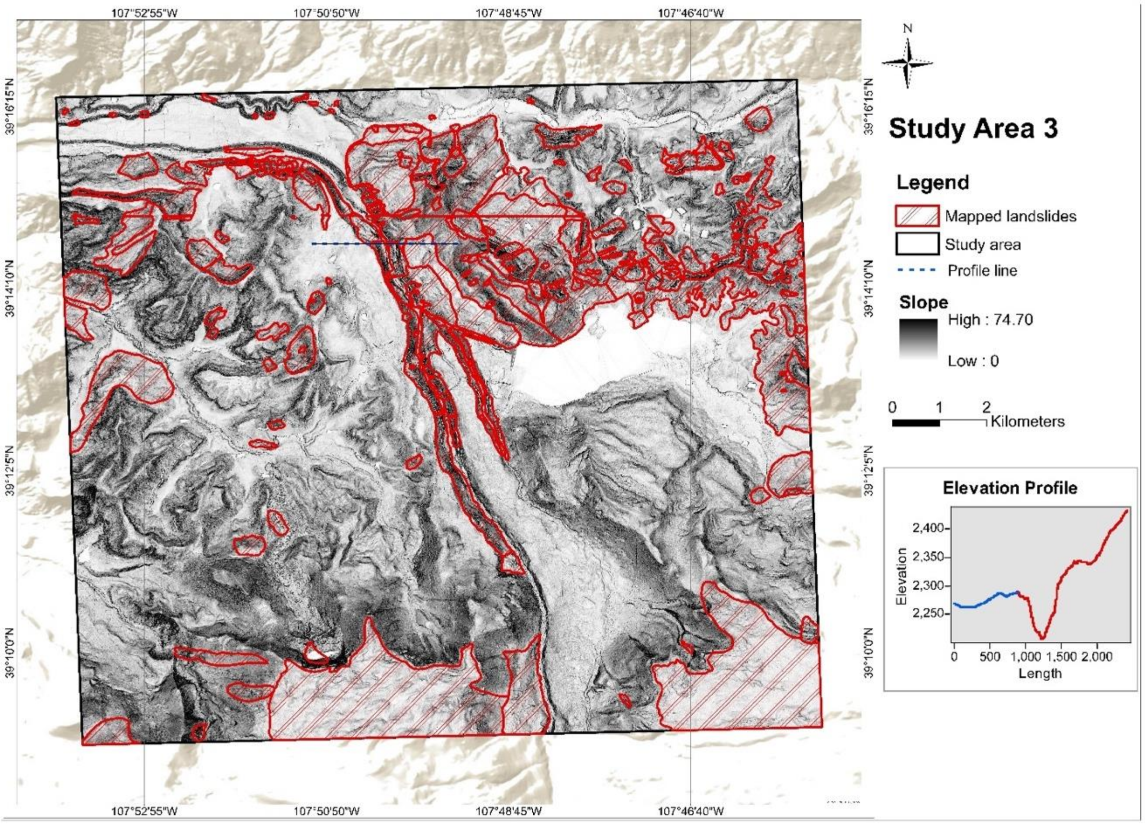


Figure 4.10 Study Area 3 (Colorado) with locations of previously mapped landslides (Syzdykbayev et al., 2020b)

Study Area 4, marked with a yellow balloon in Figure 4.7 and shown also in Figure 4.11, is located in Oregon (latitude 45°42′0″N and longitude 122°53′0″W). The total area is 134,963,016 square meters. There are 1,664 documented landslides and the combined area of mapped landslides is 61,104,222 square meters. The elevation ranges from 7 meters to 521 meters, with a standard deviation of 123 meters. The slope inclination in the area ranges from 0 to 89 degrees (Table 4.2).

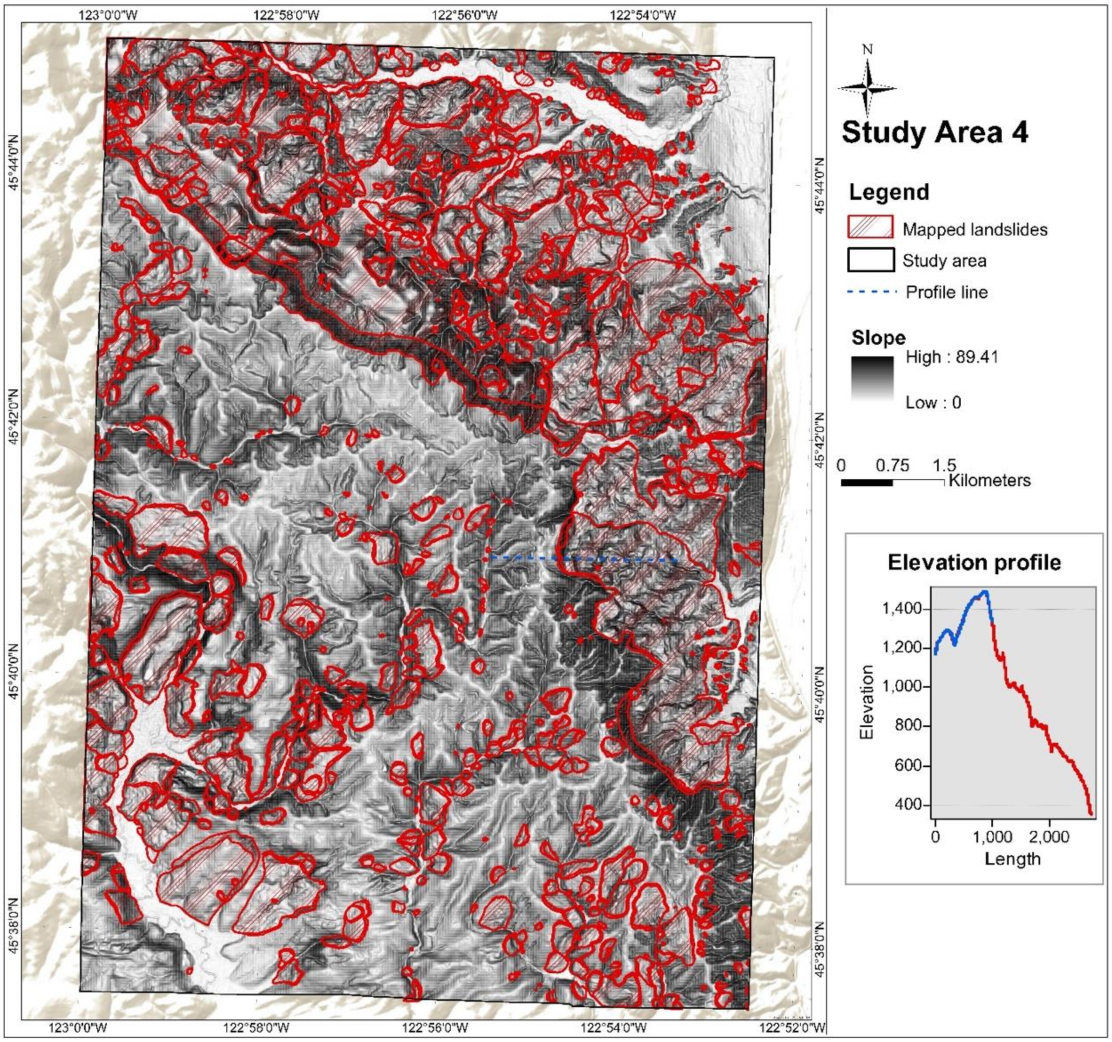


Figure 4.11 Study Area 4 (Oregon) with locations of previously mapped landslides (Syzdykbayev et al., 2020b)

Study Area 5, marked with a purple balloon in Figure 4.7 and shown also in Figure 4.12, is located in Washington (latitude 47°36′28″N and longitude 122°20′6″W). The total area is 216,603,450 square meters. There are 783 documented landslides, and the combined area of mapped landslides is 21,470,606 square meters. The elevation ranges from 0 meters to 159 meters, with a standard deviation of 39 meters. The slope inclination in the area ranges from 0 to 88 degrees (Table 4.2). The data for Study Area 5 was obtained from the Puget Sound LiDAR Consortium. The data on existing landslides were obtained from the Washington State Department of Natural Resources web portal (Table 4.3).

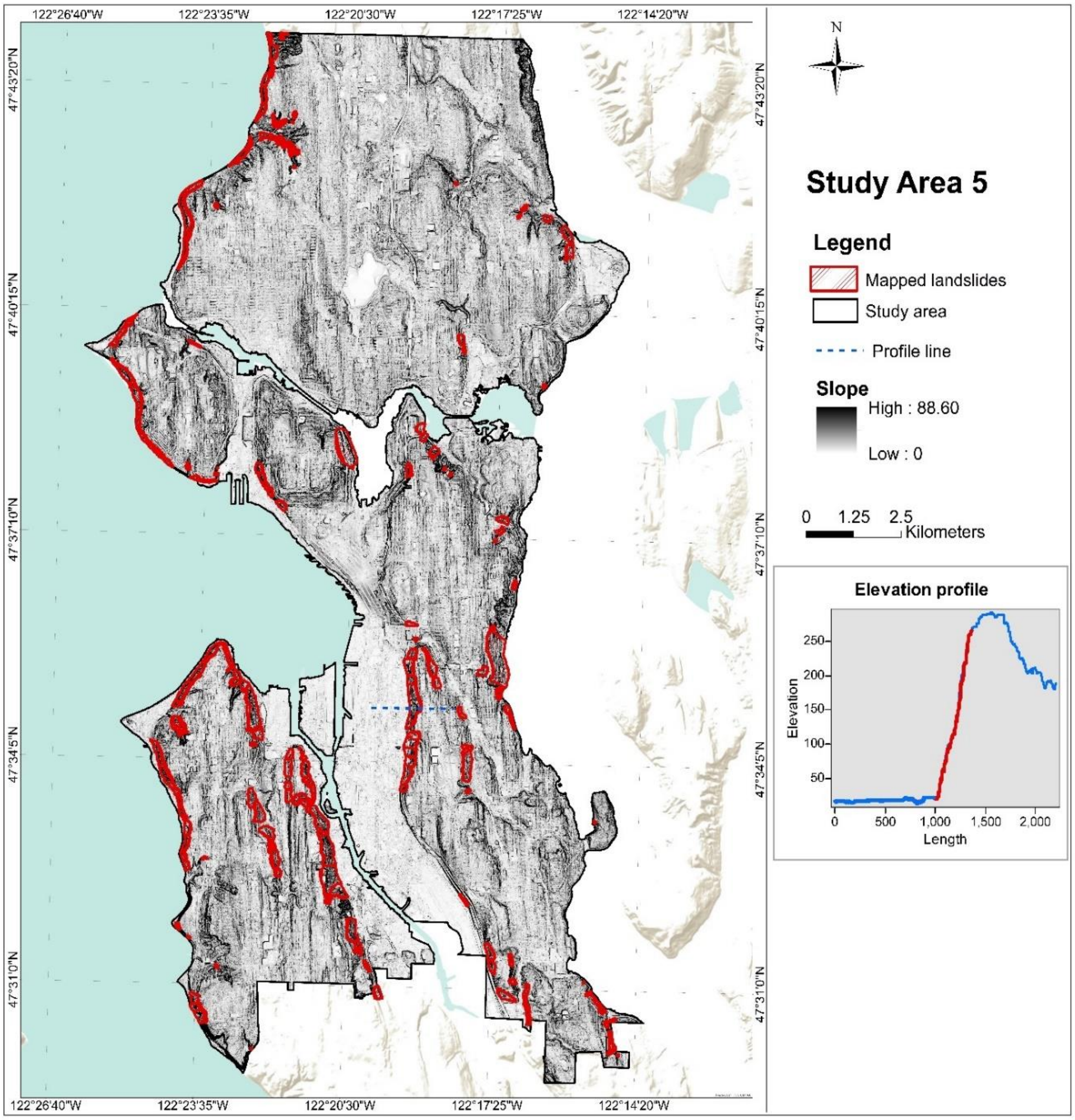


Figure 4.12 Study Area 5 (Washington) with locations of previously mapped landslides (Syzdykbayev et al., 2020b)

4.3 Experiment

To address RQ1, we designed an experimentation to detect geospatial objects, which includes a comparative analysis of the results, the established ground truth, and the findings of the knowledge-based method proposed by Bunn et al., (2019). We used the results from study areas located in Oregon (Study Area 2 and Study Area 4) for comparison since the same study areas were used in Bunn et al., (2019).

Bunn et al., (2019) introduced a method known as Scarp Identification and Contour Connection Method (SICCM) for landslide detection. SICCM is a two-step process that can utilize LiDAR-derived DTM as input data. The first step, scarp identification, involves the detection of landslide scarp lines. This detection is facilitated by applying a threshold value, which is determined through rules that utilize contextual information. The second step, deposit mapping, identifies landslide deposits. This identification is carried out by applying rules based on the geometrical information of the previously identified scarp lines. In essence, SICCM leverages both geometrical and contextual information from the terrain data to systematically detect and map landslides.

We used accuracy, precision, recall, Cohen's Kappa coefficient, and F-1 score as validation metrics.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

$$Cohen's\ Kappa\ coefficient = \frac{P_c - P_{exp}}{1 - P_{exp}} \tag{4.4}$$

where

$$P_c = Accuracy$$

$$P_{exp} = \frac{((TP + FN)*(TP + FP) + (FP + TN)*(FN + TN)}{sqrt(TP + TN + FN + FP)} \tag{4.5}$$

$$F_1 Score = \frac{2*Precision* Recall}{Precision + Recall} \tag{4.6}$$

TP stands for true positive; TN stands for true negative; FP stands for false positive; FN stands for false negative. Precision is the ratio of correctly detected areas to all detected areas. Recall is the ratio of correctly detected areas to all existing areas. Accuracy is a measure of how detection was correct overall. Cohen's Kappa coefficient (Cohen, 1960; Tsangaratos & Ilia, 2016) is a measure of agreement between detection and reality or a measure of how the result is significantly better than random (Jensen, 1996). The results of these evaluations, based on different pixel sizes, smoothing iterations, and LTFs algorithms across the five study areas, are presented in Figures 4.14 - 4.22.

4.3 Results

Detected landslides were compared to landslide inventory maps in all five study areas. For each study area, the above-mentioned three LTFs extraction algorithms, several smoothing iterations, and different pixel sizes were applied, and accuracy, precision, recall, Cohen's Kappa coefficient, and F1 score evaluation metrics were calculated. Table 4.4 shows the results of these metrics with the highest Cohen's Kappa coefficient.

Table 4.4 LTFs extraction algorithms, confusion matrix values, number of existing, detected, and intersected landslides, accuracy, precision, recall, F1 score, and Cohen's Kappa coefficient for each study area (Syzdykbayev et al., 2020b)

	Study Area 1	**Study Area 2**	**Study Area 3**	**Study Area 4**	**Study Area 5**
LTFs extraction algorithm	Curvature 2	Curvature 1	Geomorphon	Curvature 2	Curvature 2
Pixel size	5 meters	1 meter	1 meter	1 meter	10 meters
Number of smoothing iterations	5	2	5	15	1
True positive	5.17%	12.55%	14.51%	29.2%	6.%
False Positive	17.16%	22.21%	34.82%	14.66%	20.06%
False Negative	3.22%	6.54%	11.37%	15.96%	2.33%
True negative	74.44%	58.68%	39.28%	40.16%	71.59%
Accuracy	0.79	0.71	0.53	0.69	0.77
Precision	0.23	0.36	0.29	0.66	0.22
Recall	0.61	0.66	0.56	0.64	0.72
F1 score	0.33	0.46	0.38	0.65	0.34
Cohen's Kappa coefficient	**0.24**	**0.29**	**0.07**	**0.38**	**0.25**

For Study Area 1 in Pennsylvania, 55 polygons with the highest Cohen's Kappa coefficient that were deemed potential locations of existing landslides were detected, as shown in Figure 4.13. The highest Cohen's Kappa coefficient was derived by Curvature 2 (see Table 4.4), with a pixel size of 5 meters and with 5 smoothing iterations as shown in Figure 4.14 (k). The accuracy, precision, and recall are 0.79, 0.23, and 0.61, respectively. The Cohen's Kappa coefficient is 0.24, which indicates fair agreement (Landis & Koch, 1977).

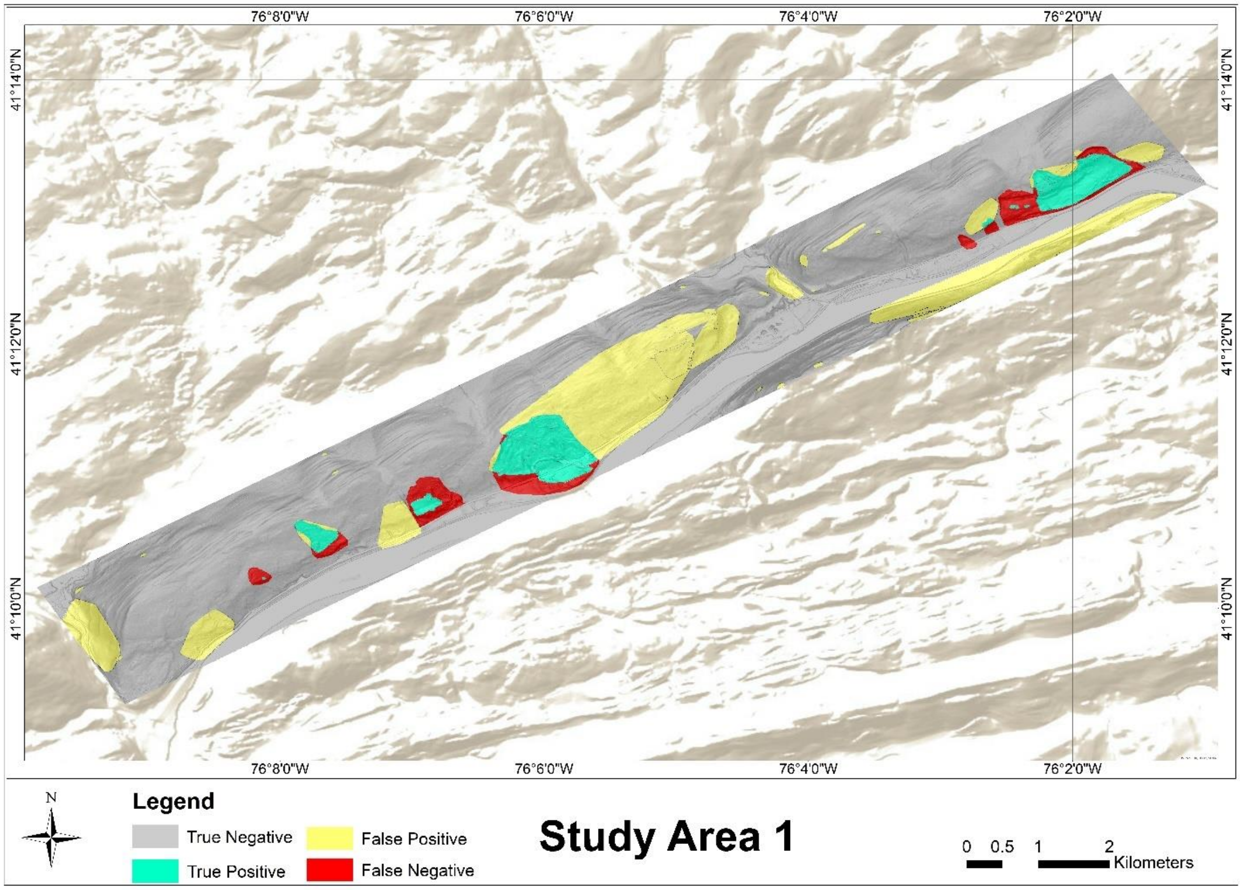


Figure 4.13 Locations of detected and mapped landslides; Study Area 1: Pennsylvania (Syzdykbayev et al., 2020b)

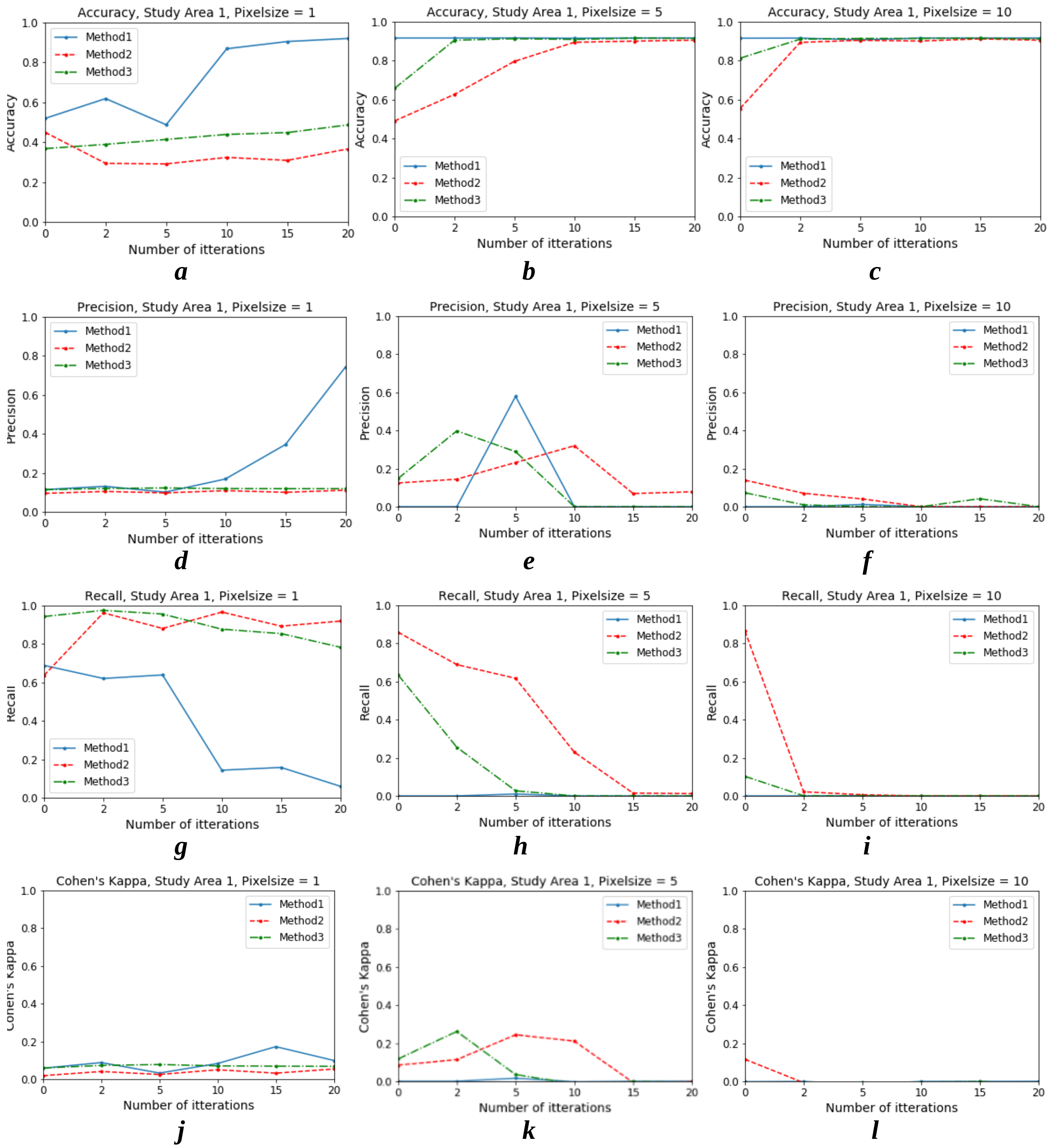


Figure 4.14 Results of the three LTFs extraction algorithms with different pixel sizes and smoothing iterations as a measure of accuracy, precision, recall, and Cohen's Kappa coefficient; Study Area 1: Pennsylvania (Syzdykbayev et al., 2020b)

For Study Area 2 in Oregon, 1514 polygons with the highest Cohen's Kappa coefficient that were deemed potential locations of existing landslides were detected, as shown in Figure 4.15. The highest Cohen's Kappa coefficient was derived by using Curvature 1 (see Table 4.4), with a pixel size of 1 meter and with 2 smoothing iterations, as shown in Figure 4.16 (j). The accuracy, precision, and recall are 0.71, 0.36, and 0.66, respectively. The Cohen's Kappa coefficient is 0.29, which indicates fair agreement (Landis & Koch, 1977).

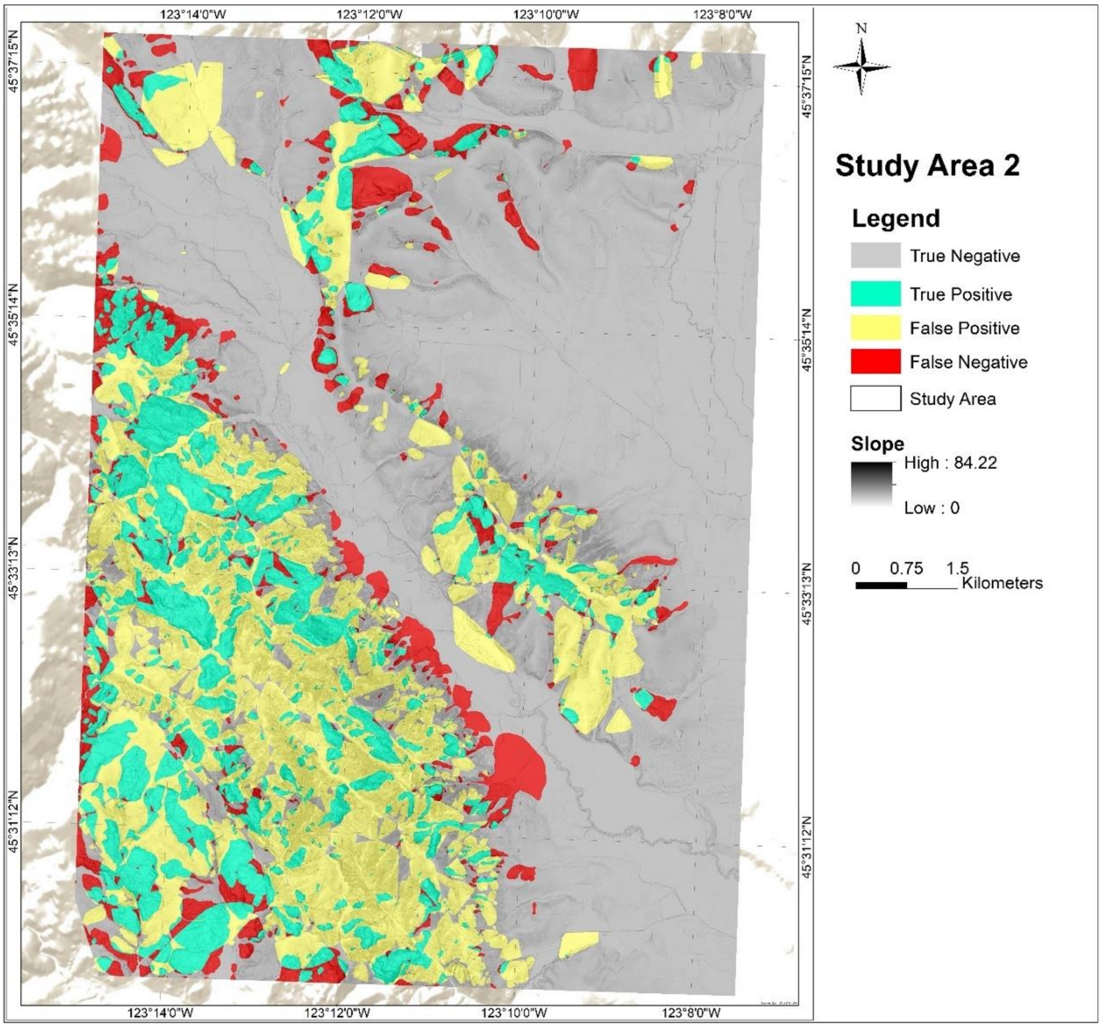


Figure 4.15 Locations of detected and mapped landslides; Study Area 2: Oregon (Syzdykbayev et al., 2020b)

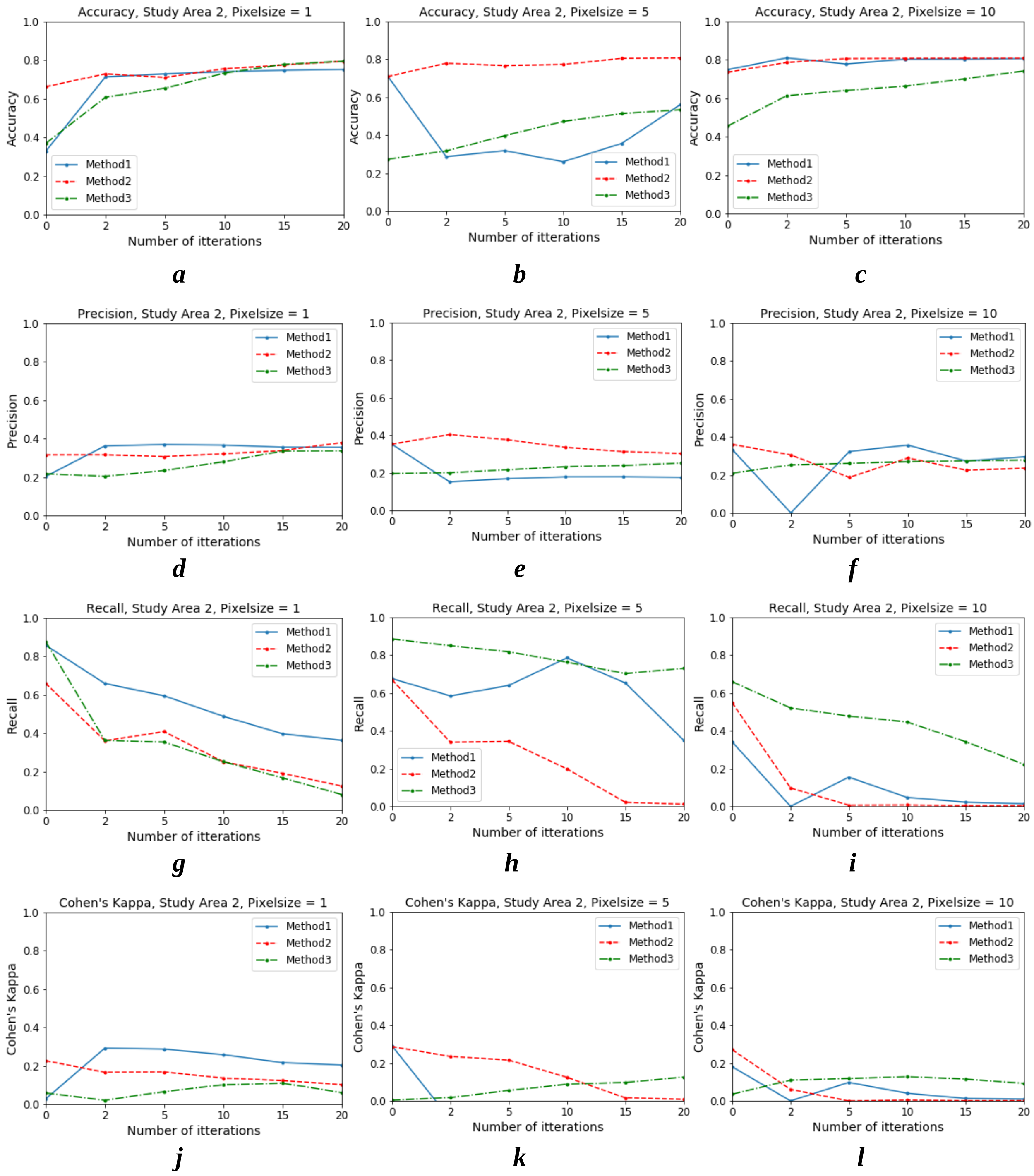


Figure 4.16 Results of the three LTFs extraction algorithms with different pixel sizes and smoothing iterations as a measure of accuracy, precision, recall, and Cohen's Kappa coefficient; Study Area 2: Oregon (Syzdykbayev et al., 2020b)

For Study Area 3 in Colorado, 494 polygons with the highest Cohen's Kappa coefficient that were deemed potential locations of existing landslides were detected, as shown in Figure 4.17. The highest Cohen's Kappa coefficient was derived by using Geomorphon (see Table 4.4), with a pixel size of 1 meter and with 5 smoothing iterations, as shown in Figure 4.18 (j). The accuracy, precision, and recall are 0.53, 0.29, and 0.56, respectively. The Cohen's Kappa coefficient is 0.07, which indicates slight or no agreement (Landis & Koch, 1977).

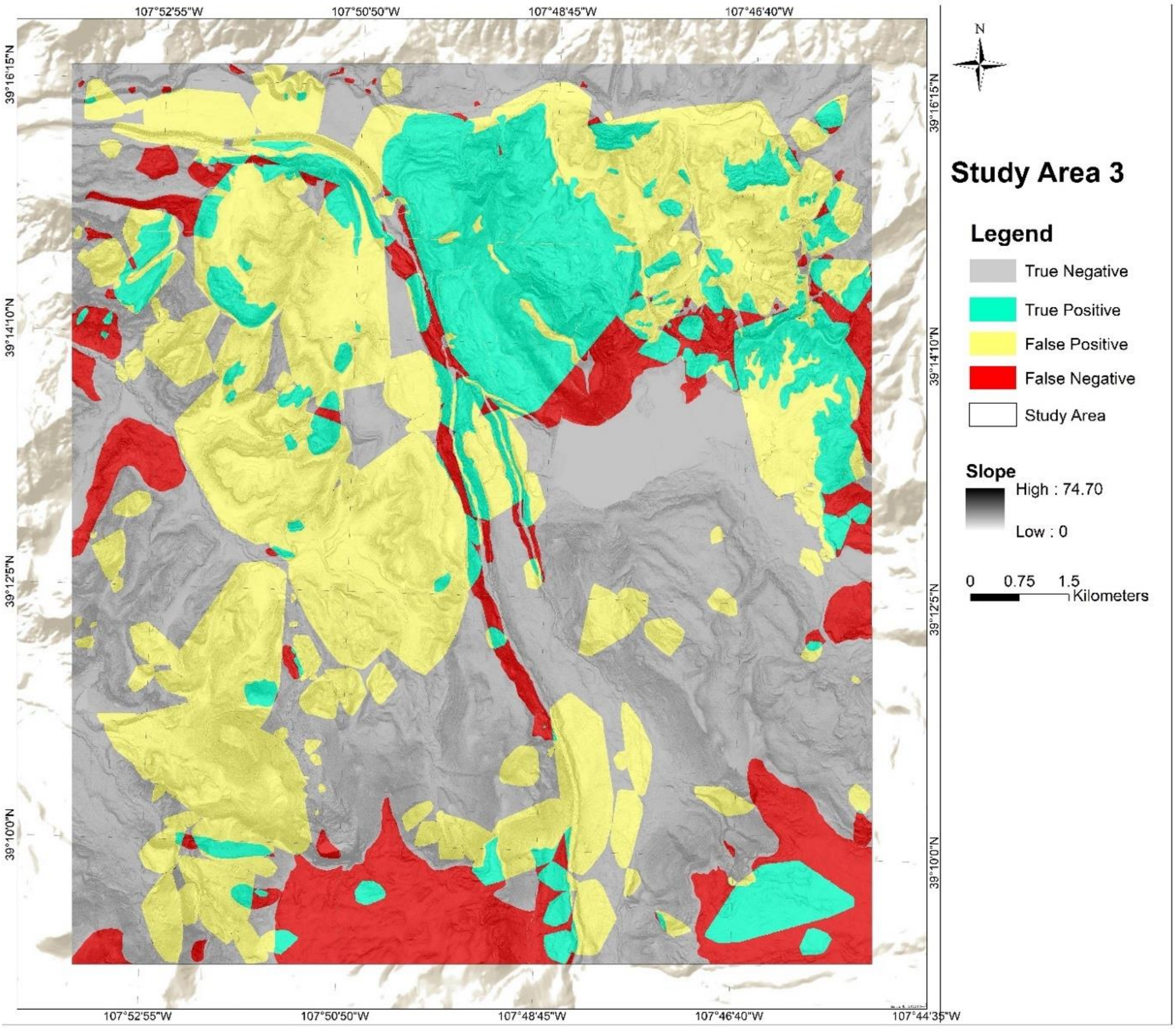


Figure 4.17 Locations of detected and mapped landslides; Study Area 3: Colorado (Syzdykbayev et al., 2020b)

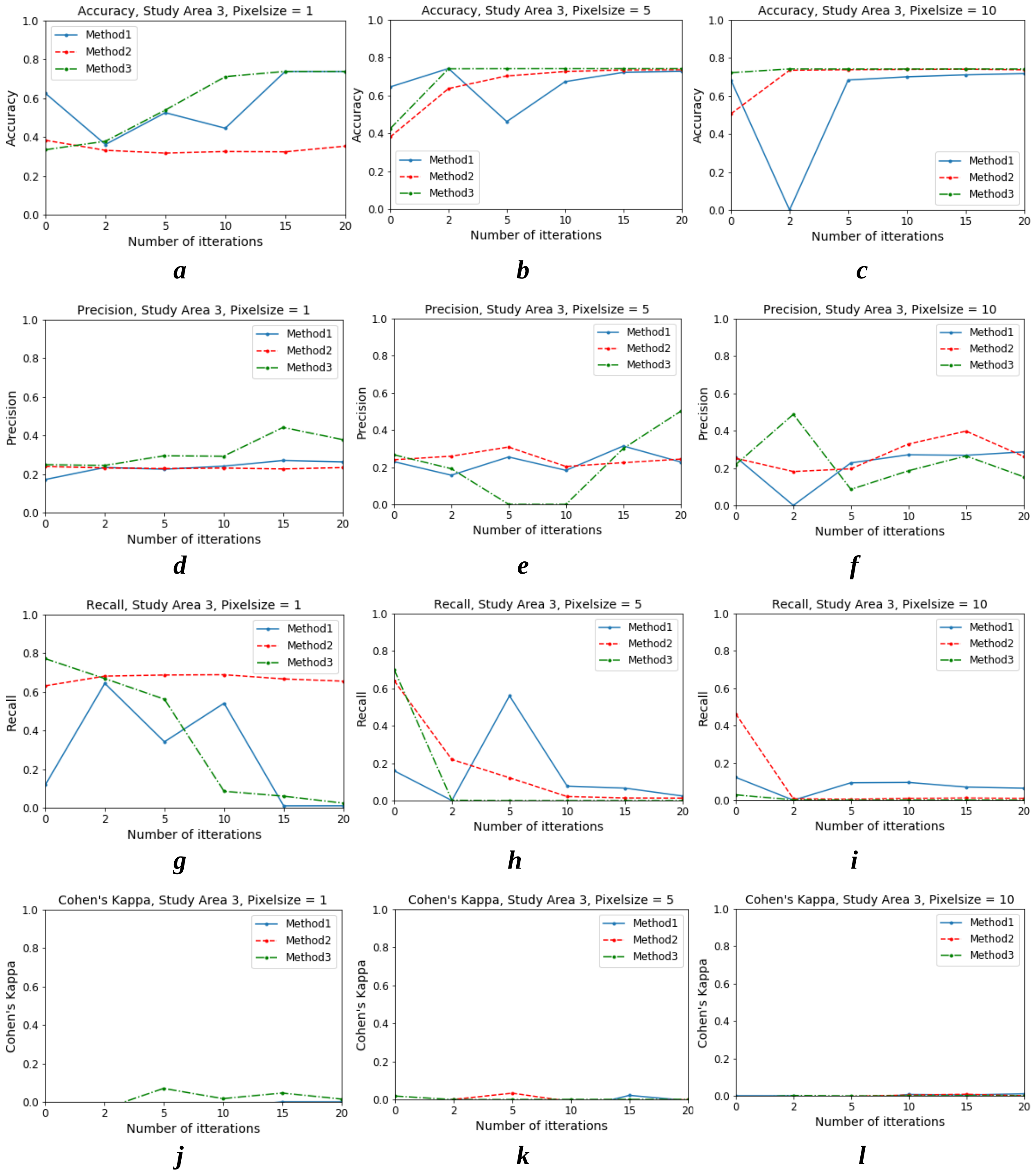


Figure 4.18 Results of the three LTFs extraction algorithms with different pixel sizes and smoothing iterations as a measure of accuracy, precision, recall, and Cohen's Kappa coefficient; Study Area 3: Colorado (Syzdykbayev et al., 2020b)

For Study Area 4 in Oregon, 1514 polygons with the highest Cohen's Kappa coefficient that were deemed potential locations of existing landslides were detected, as shown in Figure 4.19. The highest Cohen's Kappa coefficient was derived by using Curvature 2 (see Table 4.4), with a pixel size of 1 meter and with 15 smoothing iterations, as shown in Figure 4.20 (k). The accuracy, precision, and recall are 0.69, 0.66, and 0.64, respectively. The Cohen's Kappa coefficient is 0.38, which indicates fair agreement (Landis & Koch, 1977).

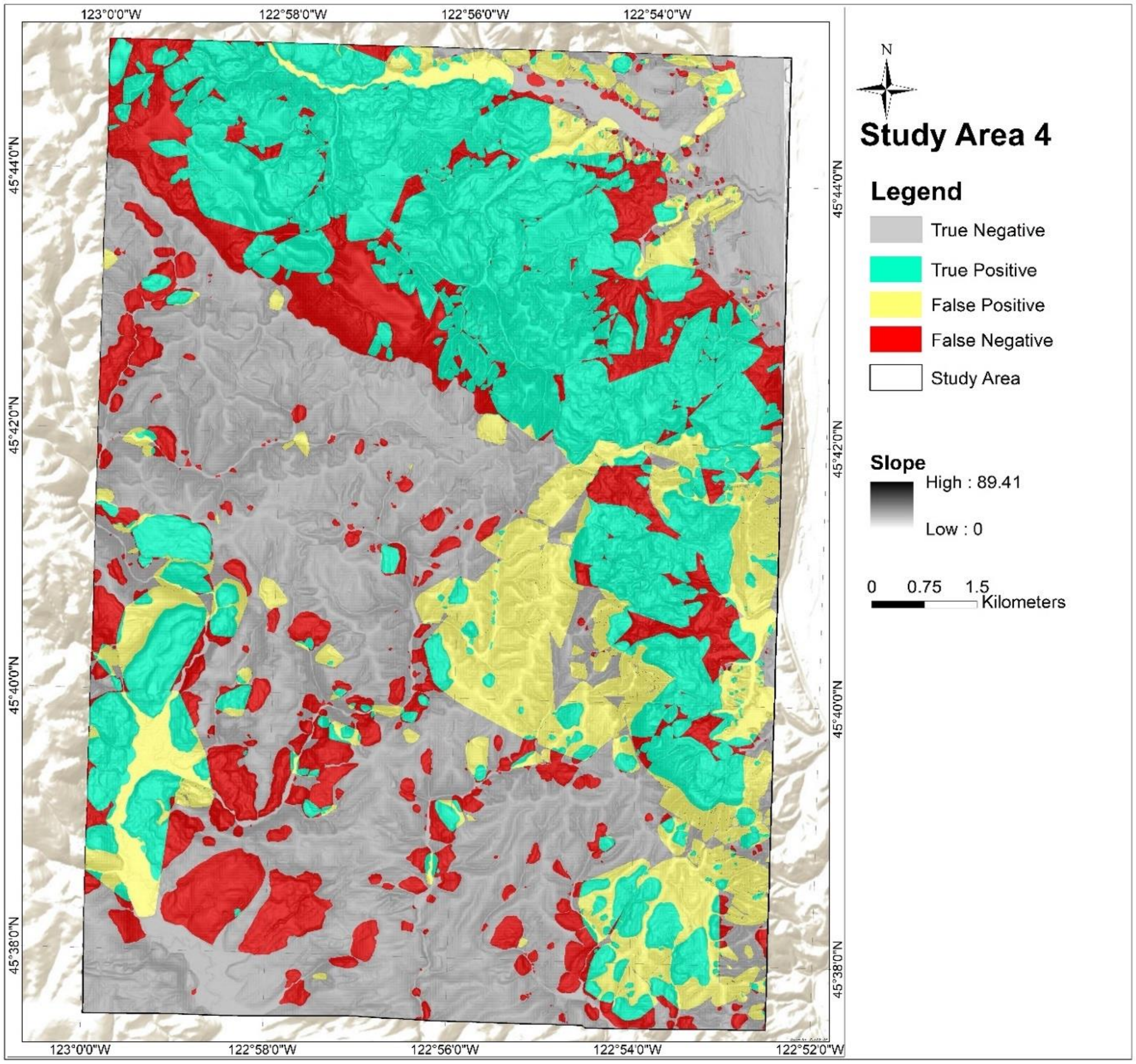
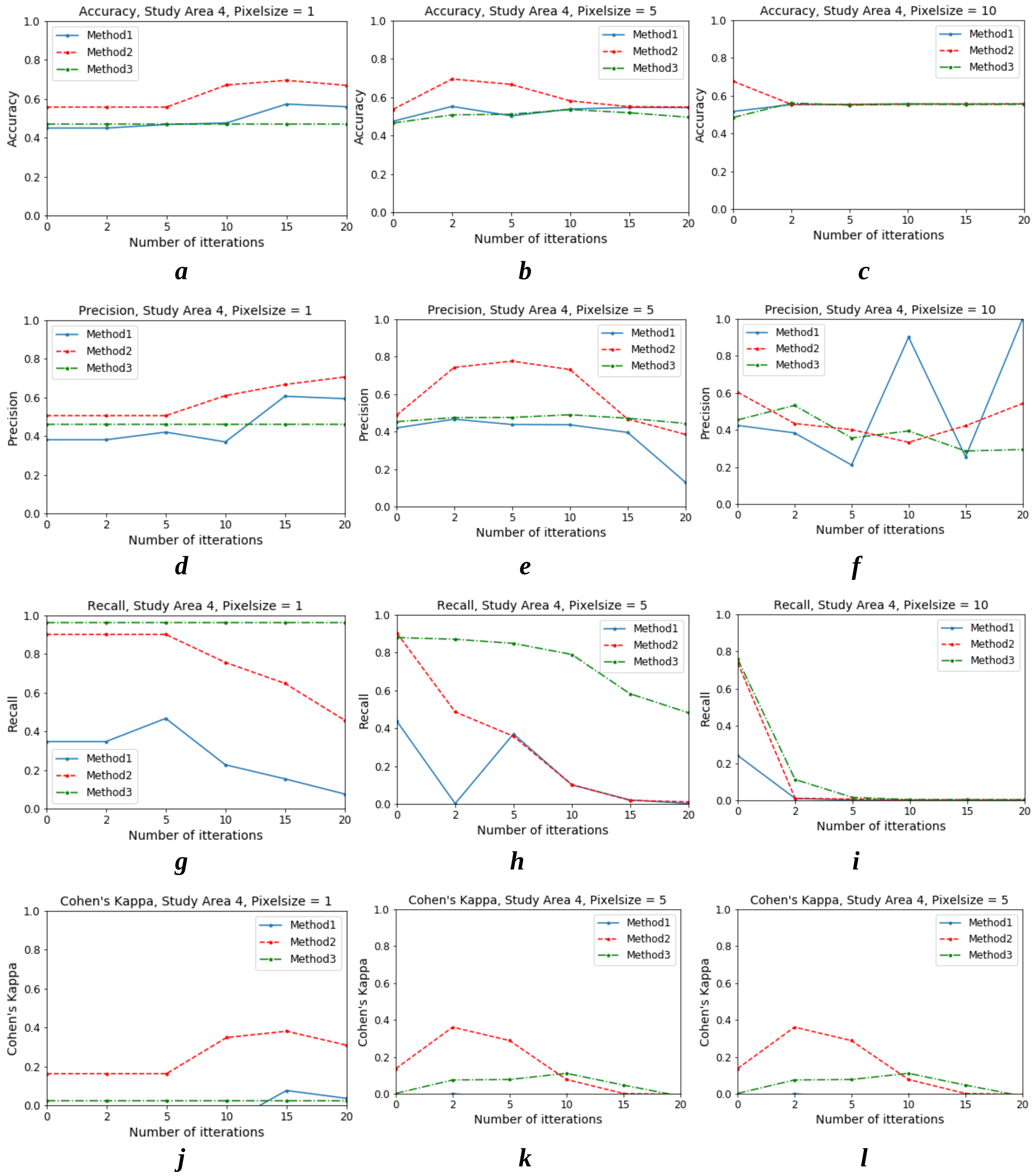


Figure 4.19 Locations of detected and mapped landslides; Study Area 4: Oregon (Syzdykbayev et al., 2020b)



a *b* *c*

d *e* *f*

g *h* *i*

j *k* *l*

Figure 4.20 Results of the three LTFs extraction algorithms with different pixel sizes and smoothing iterations as a measure of accuracy, precision, recall, and Cohen's Kappa coefficient; Study Area 4: Oregon (Syzdykbayev et al., 2020b)

For Study Area 5 in Washington, 5274 polygons with the highest Cohen's Kappa coefficient that were deemed potential locations of existing landslides were detected, as shown in Figure 4.21. The highest Cohen's Kappa coefficient was derived by using Curvature 2 (see Table 4.4), with a pixel size of 10 meters and with 1 smoothing iteration, as shown in Figure 4.22 (k). The accuracy, precision, and recall are 0.77, 0.22, and 0.72, respectively. The Cohen's Kappa coefficient is 0.25, which indicates fair agreement (Landis & Koch, 1977).

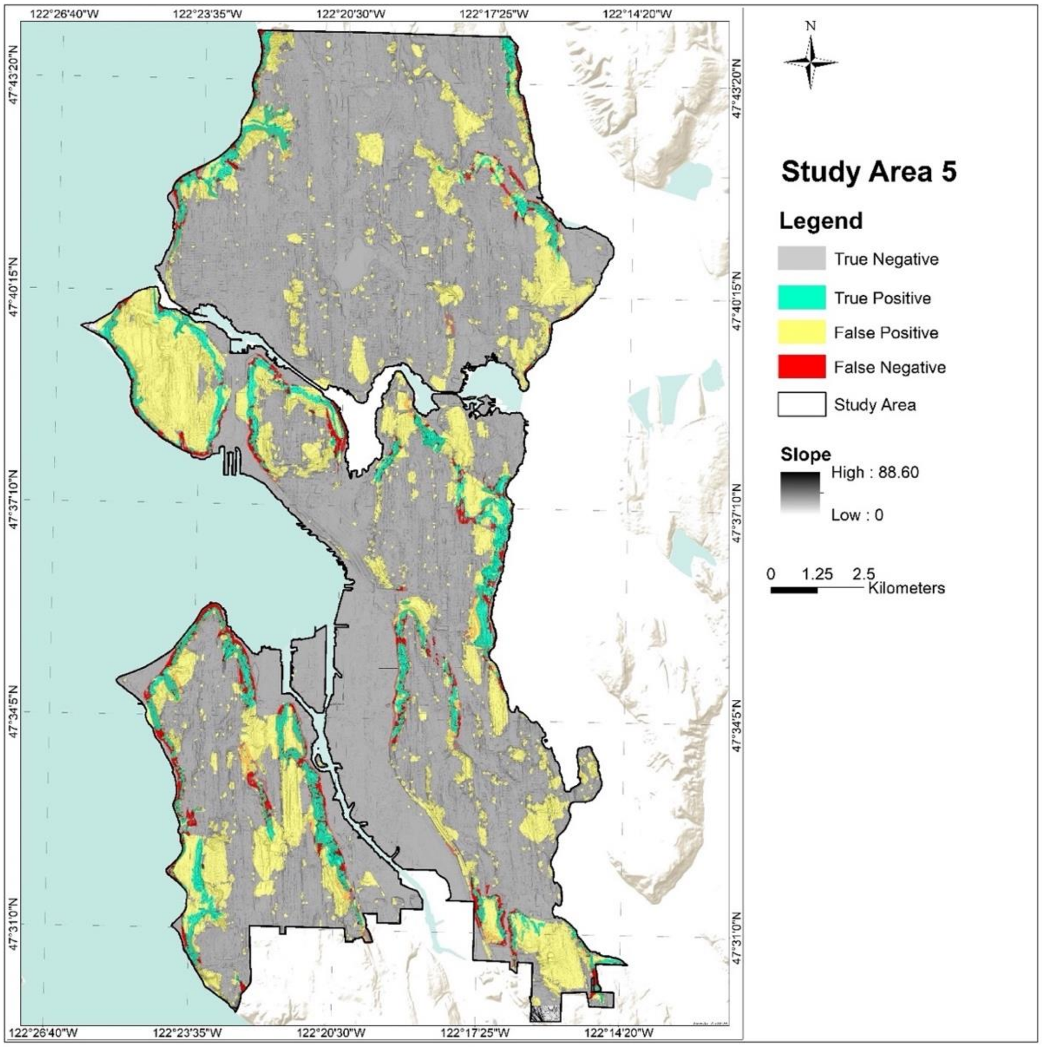
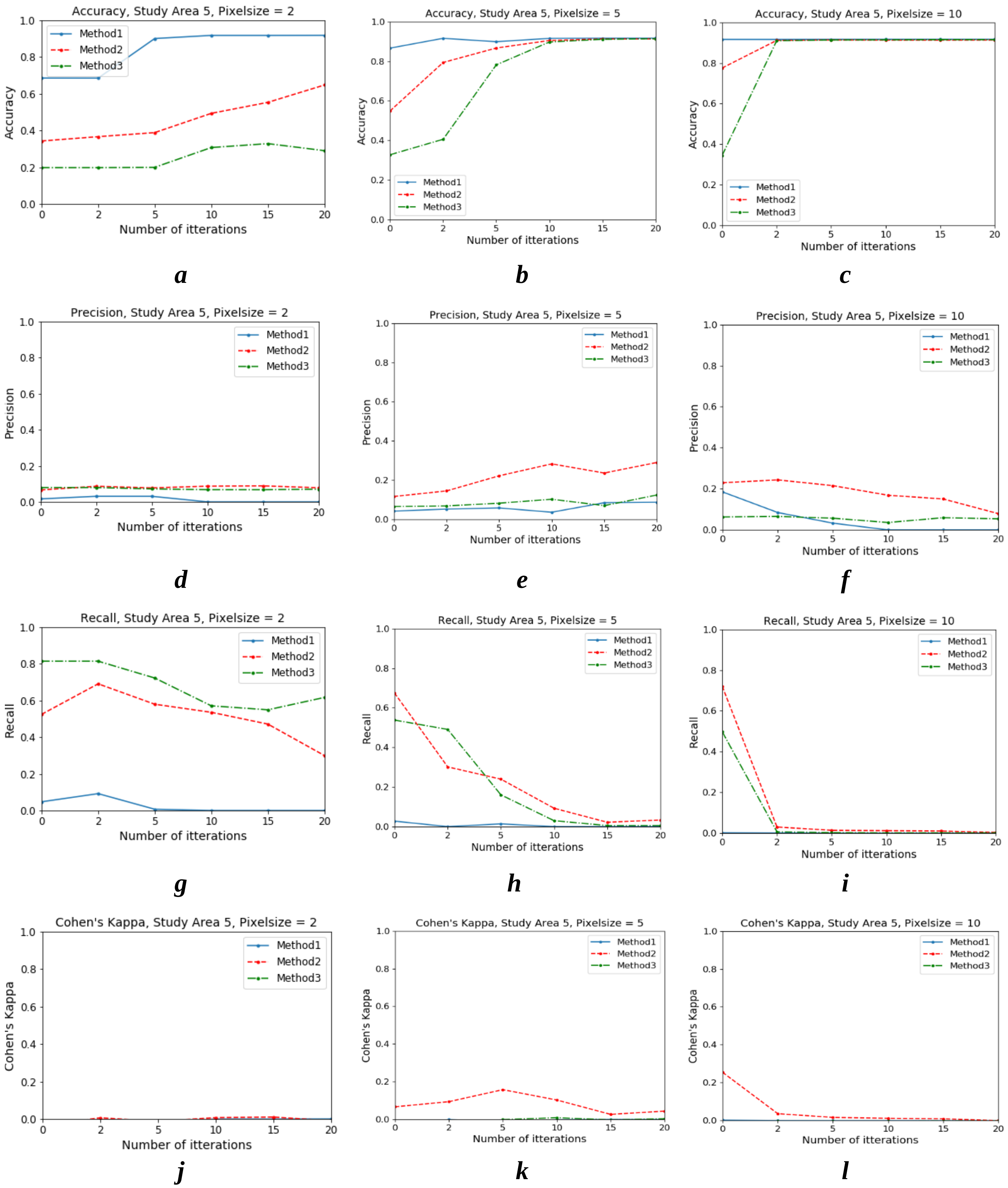


Figure 4.21 Locations of detected and mapped landslides; Study Area 5: Washington (Syzdykbayev et al., 2020b)



a *b* *c*

d *e* *f*

g *h* *i*

j *k* *l*

Figure 4.22 Results of the three LTFs extraction algorithms with different pixel sizes and smoothing iterations as a measure of accuracy, precision, recall, and Cohen's Kappa coefficient; Study Area 5: Washington (Syzdykbayev et al., 2020b)

4.4 Discussion

The results of the experiment indicate that the accuracy, precision, recall, and Cohen's Kappa coefficient vary depending on the study area, LTFs extraction algorithm, pixel size, and smoothing iteration (Table 4.4). For Study Areas 1 and 5, the highest accuracy was 0.91, and occurred when large pixel sizes and several smoothing iterations were used (Figures 4.14 and 4.22). Study Area 2 had the highest Cohen's Kappa coefficient, 0.29, and the accuracy score was 0.80. Study Area 3 had the lowest Cohen's Kappa coefficient, 0.07, and had the highest percentage of false positives, 35%.

It is worth mentioning that the area of landslides and the area of background environment are not equal. This property of the input dataset creates a foreground-to-background imbalance problem (Oksuz et al., 2020) which happens when number of positives and number of negatives are extremely unequal, hence impairing detection accuracy. Therefore, the high accuracy value can be misleading. For example, for Study Area 1, the accuracy of 0.91 with high smoothing iterations and both small and large pixel sizes (Figure 4.14 (a,b,c)) could be due to the fact that the unbalanced data detects fewer landslides increasing the values of TN and decreasing the values of FP. With a pixel size of 10 meters and 20 smoothing iterations, the accuracy dropped from 0.9 to 0, because no landslides were detected, see Figure 4.14 (c).

Pixel size and number of smoothing iterations affect the results of LTFs extraction algorithms. A large pixel size and a large number of smoothing iterations will result in a decrease in the number of detected landslides. This trend can be seen from the recall graph where the recall score drops with the increase in pixel size and number of smoothing iterations (see Figures 4.14, 4.16, 4.18 (g, h, i)).

In summary, the experiment showed that landslide boundaries can be detected by using only topological information at different scales. The overall low accuracy is due to a combination of the following factors:

- type and location of landslides,
- accuracy of landslide map inventories used for evaluation, and
- existence of FP (see Table 4.4)

FP were caused by two reasons. The first is that the LTFs extraction algorithm was sensitive to the geospatial data noise that originated during the process of deriving DTM from LiDAR data. The noise can be addressed by using a larger pixel size and a large number of smoothing iterations. However, this solution will cause data loss where, in addition to the noise, LTFs around small landslides can also disappear. To address this issue, we developed a new LTFs extraction algorithm that does not need threshold values (see Chapter 5). The second reason is that there exist other objects (geospatial or man-made) that have circular shapes similar to those of landslides. Examples of such objects are craters or farm areas. Such objects were also detected in the experiment and caused FP. To address this issue, we present a Topological KB geospatial object detection method (see Chapter 6).

5.0 New LTFs Extraction Algorithm

In Chapter 4, we demonstrated that it is feasible to generate candidate polygons using PH, thereby detecting geospatial objects by selecting from these candidate polygons, relying solely on topological information embedded in them. However, the results indicated a considerable number of false positives in certain study areas. Consequently, we postulated the following future research direction (Syzdykbayev et al., 2020b):

"Considering the importance of accurately determining ridges and scarps for detection of landslides, one future work is to take a convolutional filters approach and apply it to DTM."

5.1 Methodology

We developed a new LTFs extraction algorithm that detects terrain linear features by using shaded relief, controlling the altitude and azimuth of the illumination source, and implementing edge detection filters. The algorithm is called Shade-relief and takes a DTM as input and returns an image with LTFs. Figure 5.1 shows the steps of Shade-relief. The first step involves applying shaded relief to a DTM using several altitudes and azimuth parameters. The second step involves applying edge detection filters, shown in Equation 5.1, to each azimuth variable to obtain lines between shaded and non-shaded areas. The third step involves removing LTFs that do not persist under slight (± 44◦) azimuth shifts. The fourth step involves combining outputs of all azimuth parameters into one image by summing images and classifying the outcome.

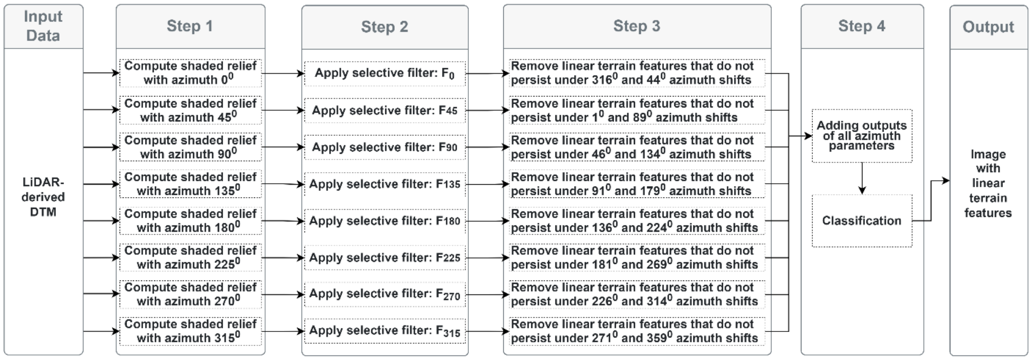

Input Data
Step 1
Step 2
Step 3
Step 4
Output
LiDAR-derived DTM
Compute shaded relief with azimuth 0⁰
Compute shaded relief with azimuth 45⁰
Compute shaded relief with azimuth 90⁰
Compute shaded relief with azimuth 135⁰
Compute shaded relief with azimuth 180⁰
Compute shaded relief with azimuth 225⁰
Compute shaded relief with azimuth 270⁰
Compute shaded relief with azimuth 315⁰
Apply selective filter: F0
Apply selective filter: F45
Apply selective filter: F90
Apply selective filter: F135
Apply selective filter: F180
Apply selective filter: F225
Apply selective filter: F270
Apply selective filter: F315
Remove linear terrain features that do not persist under 316⁰ and 44⁰ azimuth shifts
Remove linear terrain features that do not persist under 1⁰ and 89⁰ azimuth shifts
Remove linear terrain features that do not persist under 46⁰ and 134⁰ azimuth shifts
Remove linear terrain features that do not persist under 91⁰ and 179⁰ azimuth shifts
Remove linear terrain features that do not persist under 136⁰ and 224⁰ azimuth shifts
Remove linear terrain features that do not persist under 181⁰ and 269⁰ azimuth shifts
Remove linear terrain features that do not persist under 226⁰ and 314⁰ azimuth shifts
Remove linear terrain features that do not persist under 271⁰ and 359⁰ azimuth shifts
Adding outputs of all azimuth parameters
Classification
Image with linear terrain features

Figure 5.1 The steps of Shade-relief LTFs extraction algorithm

In Step 1, the shaded relief operation by Horn (1981) was used, where a different azimuth orientation was applied to the input DTM eight times. These azimuth orientations were north (0°), northeast (45°), east (90°), southeast (135°), south (180°), southwest (225°), west (270°), and northwest (315°). The slopes p and q are computed in two directions in each cell:

$$p = \frac{[(z_9 \;+\; z_8 + z_7) - (z_3 \;+\; z_2 + z_1)]}{8d} \tag{5.1}$$

$$q = \frac{[(z_9 \;+\; z_6 + z_3) - (z_7 \;+\; z_4 + z_1)]}{8d} \tag{5.2}$$

where d is the distance between pixel centers and z_i is the height at location i (Figure 5.2 a).

Next, these values were converted into a reflectance value by using an appropriate reflectance map R (Horn, 1981):

$$R(p,q) = \frac{1}{2} + \frac{\frac{1}{2}(\acute{p} \;+\; a)}{\sqrt{b^2 \;+\; (\acute{p} \;+\; a)^2}} \tag{5.3}$$

$$\acute{p} = \frac{(p_0 p \;+\; q_0 q)}{\sqrt{(p_0^2 \;+\; q_0^2)}} \tag{5.4}$$

where $p_0 = \frac{1}{\sqrt{2}}$ and $q_0 = -\frac{1}{\sqrt{2}}$ for a light source at a standard cartographic position, with an azimuth angle of 315° and an altitude of 45° for the illumination source (Figure 5.2 b). Parameters a and b allow for control of the intensity of gray values for horizontal continuous surfaces for visualization purposes, (Horn, 1981) recommended $a = 0$ and $b = \frac{1}{\sqrt{2}}$.

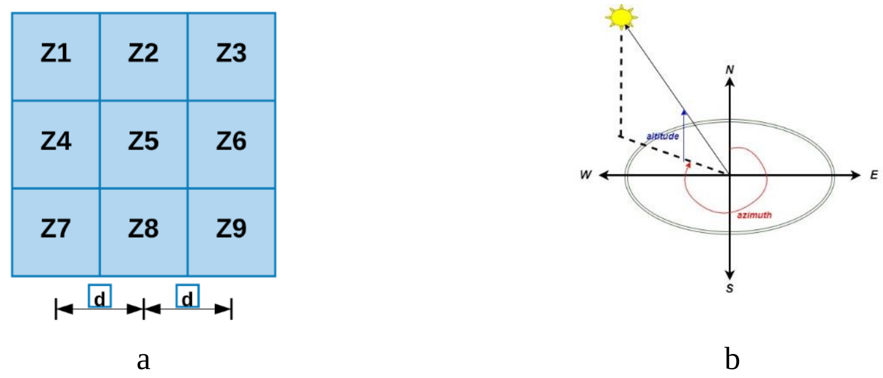


Figure 5.2 (a) 3 * 3 cell window representation of a surface; (b) illustration of light source altitude (45°) and azimuth (315°) angles (Syzdykbayev et al., 2020a)

Since the goal was to identify lines that serve as illumination boundaries, shaded areas with values equal to 0 were highlighted. Consequently, the output of the reflectance map was reclassified by using the following equation (Figure 5.3c):

$$f_{ij}(x) = \begin{Bmatrix} 0, for\ x = 0 \\ 1_{,}\ for\ x > 0 \end{Bmatrix} \qquad (5.5)$$

Step 2 involves applying edge detection filters shown in Equation 5.6 on each reclassified shaded relief operation. These filters were chosen based on a specific azimuth orientation that could be used for border (edge) detection in eight azimuthal orientations (Cesar & da Fontoura Costa, 1995). For example, to detect terrain features from shaded relief with an azimuth of 45°, filter F_{45} was used. For shaded relief with an azimuth of 90°, filter F_{90} was used, and so on. The results of using filter F_{90} (Equation 5.6) that captures border lines where a light source for the shaded relief coming from azimuth 90◦ are shown in Figure 5.3.

$$\begin{gathered}
F_{45}=\begin{Bmatrix}0&0&-1\\0&0&0\\1&0&0\end{Bmatrix}\ F_{90}=\begin{Bmatrix}0&0&0\\1&0&-1\\0&0&0\end{Bmatrix}\ F_{135}=\begin{Bmatrix}1&0&0\\0&0&0\\0&0&-1\end{Bmatrix}\ F_{180}=\begin{Bmatrix}0&1&0\\0&0&0\\0&-1&0\end{Bmatrix}\\
F_{225}=\begin{Bmatrix}0&0&1\\0&0&0\\-1&0&0\end{Bmatrix}\ F_{270}=\begin{Bmatrix}0&0&0\\-1&0&1\\0&0&0\end{Bmatrix}\ F_{315}=\begin{Bmatrix}-1&0&0\\0&0&0\\0&0&1\end{Bmatrix}\ F_{0/360}=\begin{Bmatrix}0&-1&0\\0&0&0\\0&1&0\end{Bmatrix}
\end{gathered} \tag{5.6}$$

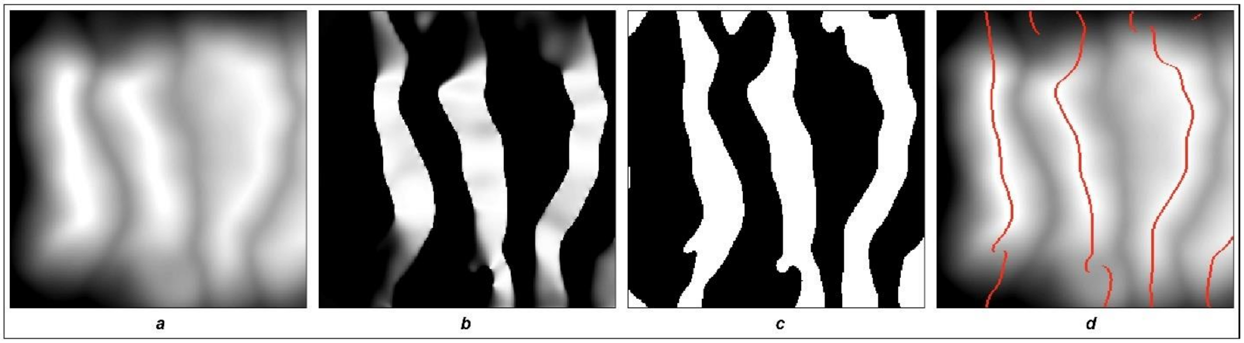


Figure 5.3 (a) DTM of synthetically-generated lines, (b) shaded relief (azimuth: 90° and altitude: 0°) of synthetically-generated DTM, (c) reclassification results, (d) filter F_{90} results overlaid on DTM (Syzdykbayev et al., 2020a)

Step 3 involves removing LTFs that do not persist under slight (± 44◦) azimuth shifts. This operation is needed to remove false LTFs. For flatter terrain or terrain that has a shape that resembles a cone or half-sphere, Shade-relief detected spur lines in fixed (every 45°) directions (Figure 5.4a). To remove false LTFs, two tasks were implemented. In the first task, for each azimuth orientation, Steps 1 and 2 were repeated by shifting azimuth orientation to ±44°. This is the maximum allowable azimuth value because filters were designed to be compatible with specific shaded relief that is performed every 45°. If the value exceeds ±44°, these filters will not be able to capture border lines for specific shaded relief. Several experiments with different azimuth values were performed and the results show that smaller values lead to a higher rate of false spur line detection (the results of these experiments are presented in the discussion section). The output of this task is three images with terrain features (TF) for each azimuth degree; the main azimuth degree TF_i and two additional created images are TF_{i-44} and TF_{i+44}. For example, if $i = 45°$ then three images, TF_1 , TF_{45}, and TF_{89}, would be created. In the second task, for each azimuth degree, true terrain features that persist under slight azimuth changes were selected. The selection was performed by using Equation 5.7:

$$TF_{i\_true} = \ (TF_{i-44} + TF_i + TF_{i+44}) \tag{5.7}$$

$$TF_{i\_true}(x) = \begin{Bmatrix} 0, for\ x = 1 \\ 1, for\ x > 1 \end{Bmatrix} \tag{5.8}$$

By applying this equation, the true terrain features that persist under slight azimuth changes received a high weight (pixel value), whereas false spur lines did not persist under slight azimuth changes and had a pixel value equal to 1 (Figure 5.4b). Thus, by deleting values that were equal to 1, false spur lines were removed (Figure 5.4c).

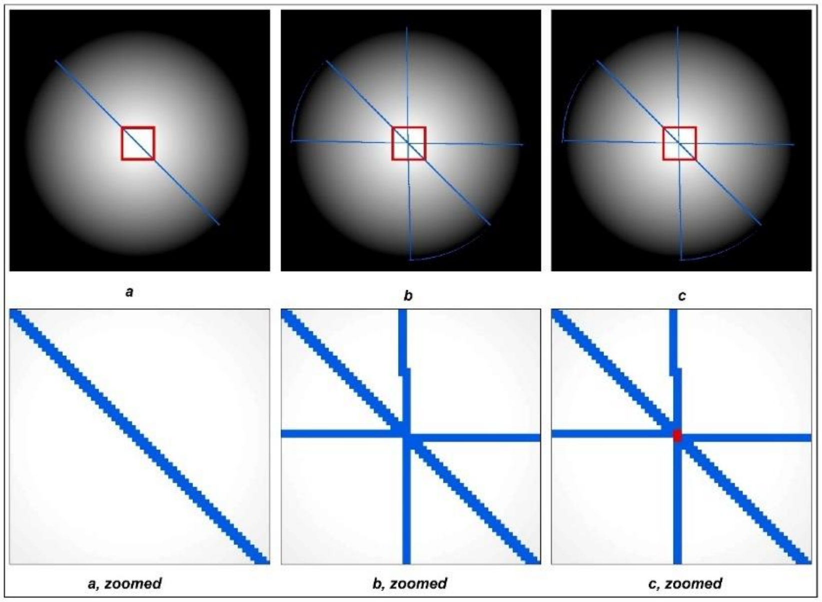


Figure 5.4 DTM of cone overlaid with (a) borderline 45°, (b) addition of three borderlines 1°, 45°, 89°, and (c) detected terrain features (peak) (Syzdykbayev et al., 2020a)

Step 4 involves combining the outputs of all azimuth parameters into one image by summing all corresponding pixels of all images and classifying the outcome. After using filters and removing falsely detected spur lines, images with LTFs from eight azimuth orientations were generated. In other words, each image contains LTFs extracted from a specific azimuth orientation. To obtain LTFs from all azimuth orientations, these eight images were combined resulting in one image with a maximum pixel value equal to 8 and a minimum pixel value equal to 0.

$$TF_{total} = \sum_{1}^{8} TF_{i\_true} \tag{5.9}$$

From the results, a pixel whose value was smaller than 2 was considered to be noise and pixels with values larger than 2 were classified as LTFs.

$$TF_{total}(x) = \begin{Bmatrix} linear\ terrain\ features \quad for\ x > 2 \\ noise \quad for\ x \leq 2 \end{Bmatrix} \tag{5.10}$$

5.2 Datasets

A synthetically-generated DTM and a real-world DTM were used as input for Shade-relief, Geomorphon, Curvature, and Probabilistic, and their performances were compared. The use of synthetically-generated datasets for this experiment is grounded in several reasons: (a) non-existent pre-established datasets for LTFs; (b) flexibility to create unique shapes and construct a variety of scenarios, enabling a thorough evaluation of the LTFs extraction algorithms; and (c) possibility of predefining knowledge for the ground truth, known number, length, and location of extracted LTFs, providing a reliable benchmark for assessing algorithmic performance.

The synthetically-generated DTM included common shapes such as cone (Figure 5.5), half-sphere (Figure 5.6), square pyramid (Figure 5.7), and two terrains with known straight and curved LTFs. Each of these terrains has seven (Figure 5.8) and eight (Figure 5.9) separate LTFs, respectively, with a different level of sharpness. These datasets were generated by using the World Machine Basic (World Machine, n.d.) software. For the real-world DTM, two locations were used: the southern half of the Lackawanna synclinorium to the southeast of the city of Wilkes-Barre in Luzerne County, Pennsylvania (latitude 41° 10’ 25” N and longitude −75° 54’1” W) and Crater Lake in Klamath County, Oregon (latitude 42°95′ N and longitude 122°10′ W). The input data for the first location was obtained from Pennsylvania Spatial Data Access (Access, n.d.). The input data for the second location were obtained from the State of Oregon Department of Geology and Mineral Industries public file transfer protocol (FTP) site. The LTFs from these real-world data were manually extracted.

5.3 Experiment

For evaluation, the performance of Shade-relief was compared with the performances of Geomorphon (Jasiewicz & Stepinski, 2013), Curvature (Pirotti & Tarolli, 2010), and Probabilistic (X. Zhou et al., 2019). Both synthetically-generated and real-world datasets were used in these comparisons.

Geomorphon uses the principle of pattern recognition to classify landforms from DTM (Jasiewicz & Stepinski, 2013). It detects common local morphological elements such as flats, peaks, ridges, shoulders, spurs, slopes, hollows, footslopes, valleys, and pits, using the concept of local ternary patterns (Liao, 2010). To extract these local ternary patterns, instead of using a fixed size neighborhood, Geomorphon uses a neighborhood with a size and shape suitable for the local topography. Hence, it can identify landforms at various spatial scales and is computationally efficient (Jasiewicz & Stepinski, 2013).

Curvature extracts LTFs by creating surface curvatures and uses a certain threshold; a multiplication of the curvature's standard deviation. This algorithm includes a combination of the first derivative of an elevation, which is slope, and the second derivative, which is curvature. Curvature is one of the basic terrain parameters and is commonly used in terrain analysis (Pirotti & Tarolli, 2010). It defines the orientation of a slope and quantifies morphologies, where a positive pixel value of the output image is convex and a negative pixel value is concave; a ridge pixel has a positive value, while a channel has a negative value (Rana, 2006).

Probabilistic extracts LTFs by using aspects and slopes that are derived from DTM. This algorithm uses contextual information and multiple neighborhood analysis in combination with a probability model to extract LTFs (X. Zhou et al., 2019).

5.4 Results

Extracted LTFs were compared to true LTFs in both synthetically-generated and real-world datasets. For each dataset, accuracy, precision, recall, and Cohen's Kappa coefficient evaluation metrics were calculated. Tables 5.1 and 5.2 show the results of these metrics. Overall, the results show that Shade-relief outperformed all three other algorithms. Specifically, Shade-relief more often resulted in the highest evaluation metric values, such as Cohen's Kappa coefficient, suggesting that it outperformed, or was at the very least competitive against, the other three algorithms.

5.4.1 Synthetically-generated Dataset

Table 5.1 shows the evaluation metrics of the results on synthetically-generated dataset produced by Shade-relief and those produced by Geomorphon, Curvature, and Probabilistic and compared to known terrain features.

Table 5.1 Cohen's Kappa coefficient, accuracy, precision, recall for a synthetically-generated dataset with each algorithm's results compared to known terrain features (Syzdykbayev et al., 2020a)

Dataset	Algorithm	Cohen's Kappa	Accuracy	Precision	Recall
Cone	Shade-relief	0.133	0.999	0.071	1
	Geomorphon	0.0003	0.978	0.0001	1
	Curvature	0.011	0.9993	0.005	1
	Probabilistic	0.0097	0.9992	0.004	1
Half sphere	Shade-relief	0.333	0.999	0.2	1
	Geomorphon	0.0003	0.979	0.0001	1
	Curvature	−0.0007	0.993	0	0
	Probabilistic	−0.0007	0.999	0	0
Square Pyramid	Shade-relief	1	1	1	1
	Geomorphon	0.589	0.997	0.979	0.422

	Curvature	1	1	1	1
	Probabilistic	0.497	0.993	0.334	0.994
Lines	Shade-relief	0.653	0.992	0.488	1
	Geomorphon	0.137	0.931	0.081	0.863
	Curvature	0.691	0.995	0.671	0.7165
	Probabilistic	−0.0072	0.985	0	0
Lines Curved	Shade-relief	0.735	0.994	0.594	0.972
	Geomorphon	0.133	0.898	0.079	0.99
	Curvature	0.274	0.982	0.216	0.404
	Probabilistic	0.225	0.981	0.18252	0.326

For the synthetically-generated dataset, the number, length, and location of the terrain features were already known. Shapes such as a cone or half-sphere have only one peak that can be extracted as one point, as shown in Figures 5.5 and 5.6.

For the conical shape (Figure 5.5), all four algorithms were able to detect the known peak, though with varying levels of success, resulting in a recall value of 1 (Table 5.1). The area representing a detected peak from Curvature and Probabilistic was much larger, Cohen's Kappa coefficients were 0.011 and 0.009, respectively (Table 5.1). Geomorphon correctly detected the location of the peak. However, it falsely detected LTFs and since the area of falsely detected LTFs was large, Cohen's Kappa coefficient is 0.0003 (Table 5.1). In comparison with the other three algorithms, the area representing a detected peak by Shade-relief was smaller, with Cohen's Kappa coefficient at 0.133.

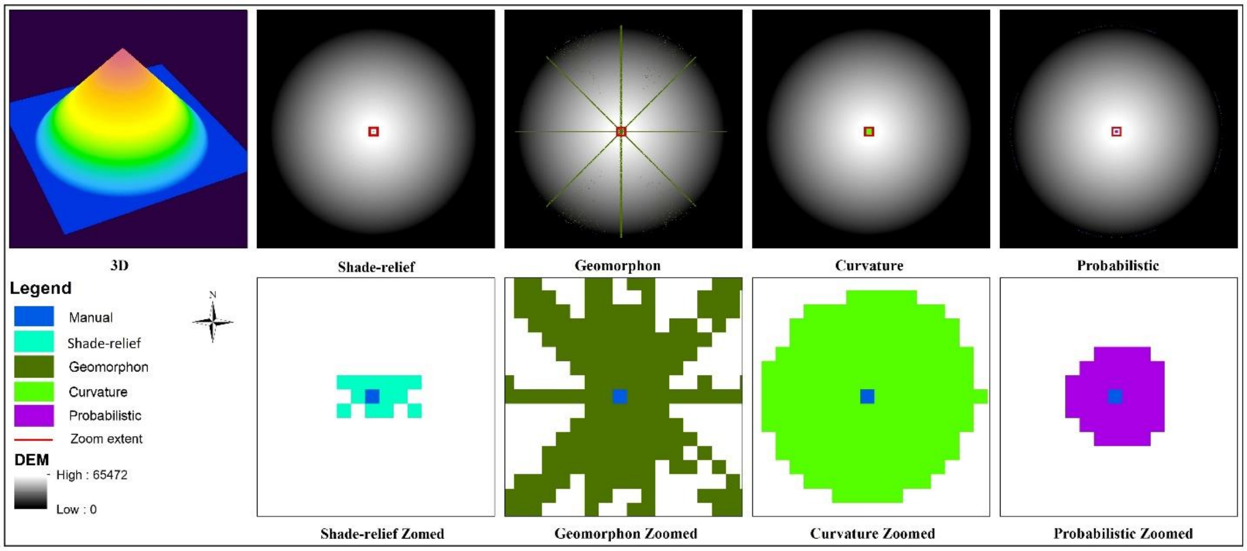


Figure 5.5 Synthetically-generated conical topographic dataset with only one peak (dark blue pixel) and no spurs or ridges. Outputs of the four terrain feature detection algorithms overlaid on the DTM and shown at two different scales (Syzdykbayev et al., 2020a)

For the half-sphere shape (Figure 5.6), Shade-relief had the highest values for Cohen's Kappa coefficient at 0.333 (Table 5.1). Both Shade-relief and Geomorphon were able to detect the peak with a recall value of 1 (Table 5.1), while Curvature and Probabilistic failed to detect the peak (Figure 5.6). Similarly with the conical shape, Geomorphon falsely detected LTFs that do not exist in the dataset, with Cohen's Kappa coefficient at 0.0003 (Table 5.1).

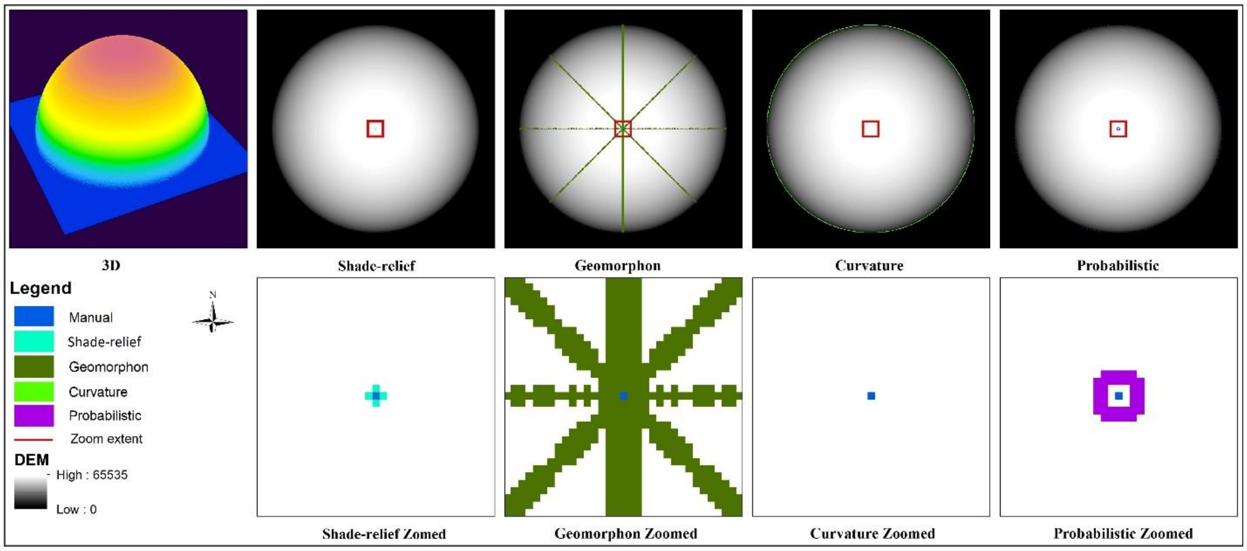


Figure 5.6 Synthetically-generated half-sphere topographic dataset with only one peak (dark blue pixel) and no spurs or ridges. Outputs of the four terrain feature detection algorithms overlaid on the DTM and shown at two different scales (Syzdykbayev et al., 2020a)

For the square pyramid shape (Figure 5.7), all four algorithms were able to detect LTFs accurately with a recall value of 1. Comparing the results of all four algorithms with the manually-extracted known features shows that Shade-relief and Curvature had a value of 1 for all four metrics, while Geomorphon had a particularly low Cohen’s Kappa coefficient (0.589) and recall (0.422) and Probabilistic had a low Cohen’s Kappa coefficient (0.497) and precision (0.334) (Table 5.1).

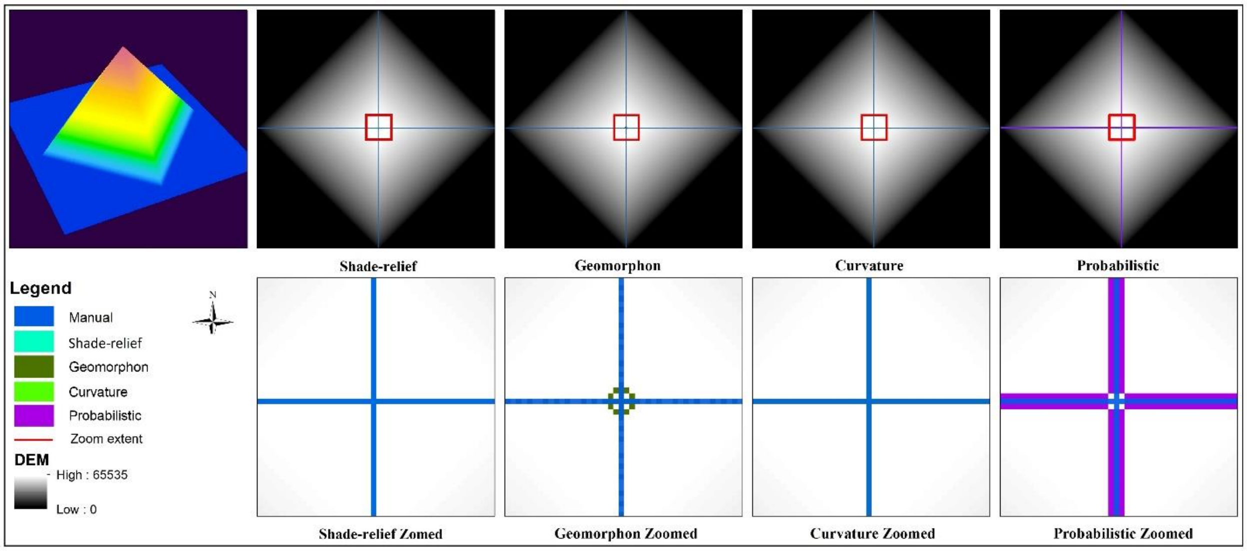


Figure 5.7 Synthetically-generated square pyramid topographic dataset with only one peak and four ridges radiating out from the peak (dark blue pixels). Outputs of the four terrain feature extraction algorithms overlaid on the DTM and shown at two different scales (Syzdykbayev et al., 2020a)

The DTM derived from synthetically-generated terrain with seven LTFs (straight ridges) of equal length (Figure 5.8) exhibits variation from sharp ridges to curved LTFs. Comparing the results of the four algorithms against the known features (Table 5.1) shows that Curvature had the highest Cohen’s Kappa coefficient (0.691), accuracy (0.995), and precision (0.671), while Shade-relief had the highest recall. Overall, all four algorithms had a high (>0.9) accuracy, and Cohen’s Kappa coefficient (0.653) for Shade-relief was similar to that of Curvature’s results (0.671).

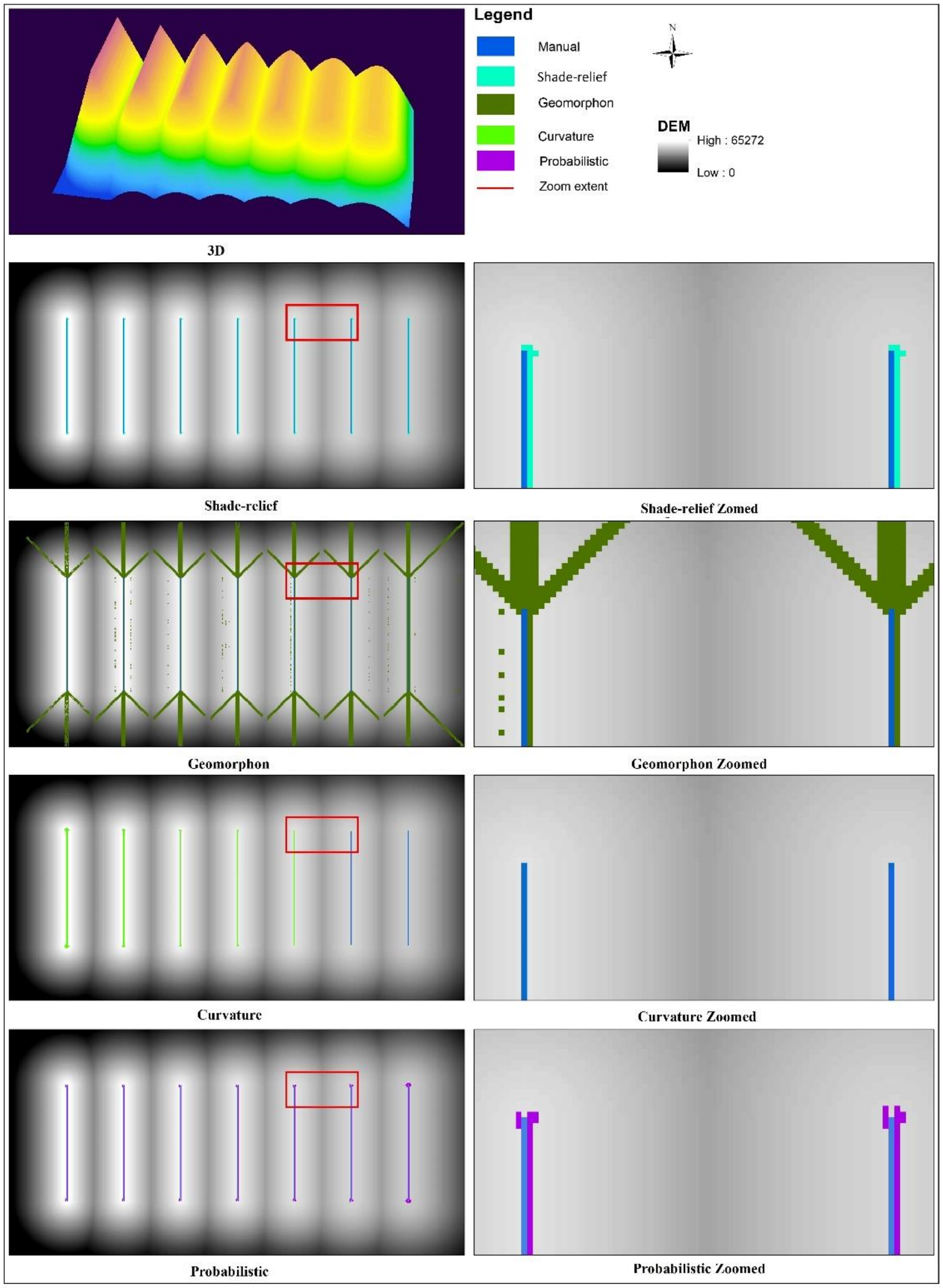


Figure 5.8 Synthetically-generated topographic dataset with seven straight ridges exhibiting a decreasing peak sharpness from left to right. Dark blue pixels represent known ridges that were equal in length for all seven ridges. Outputs of the four terrain feature extraction algorithms overlaid on the DTM and shown at two different scales (Syzdykbayev et al., 2020a)

The DTM derived from terrain with eight curvilinear LTFs (Figure 5.9) also exhibits variation from sharp to curved LTFs. Comparing the results of the four algorithms against the known features (Table 5.1) shows that Shade-relief outperformed the other three algorithms on all metrics, with a high Cohen's Kappa coefficient (0.735), accuracy (0.994), and precision (0.972), except for recall, for which Geomorphon had the highest value (0.99). However, the recall for both Shade-relief (0.972) and Geomorphon (0.99) was high.

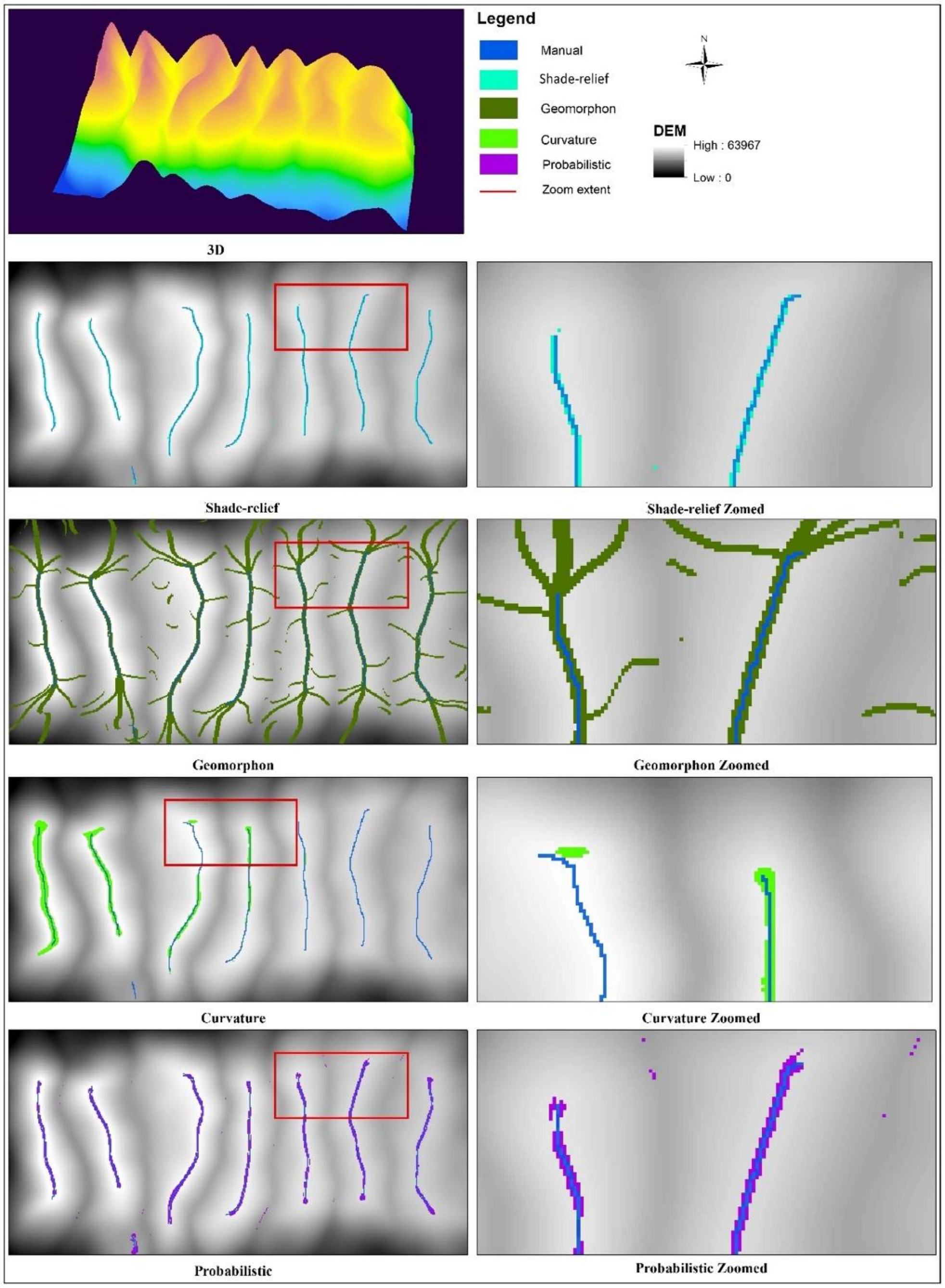
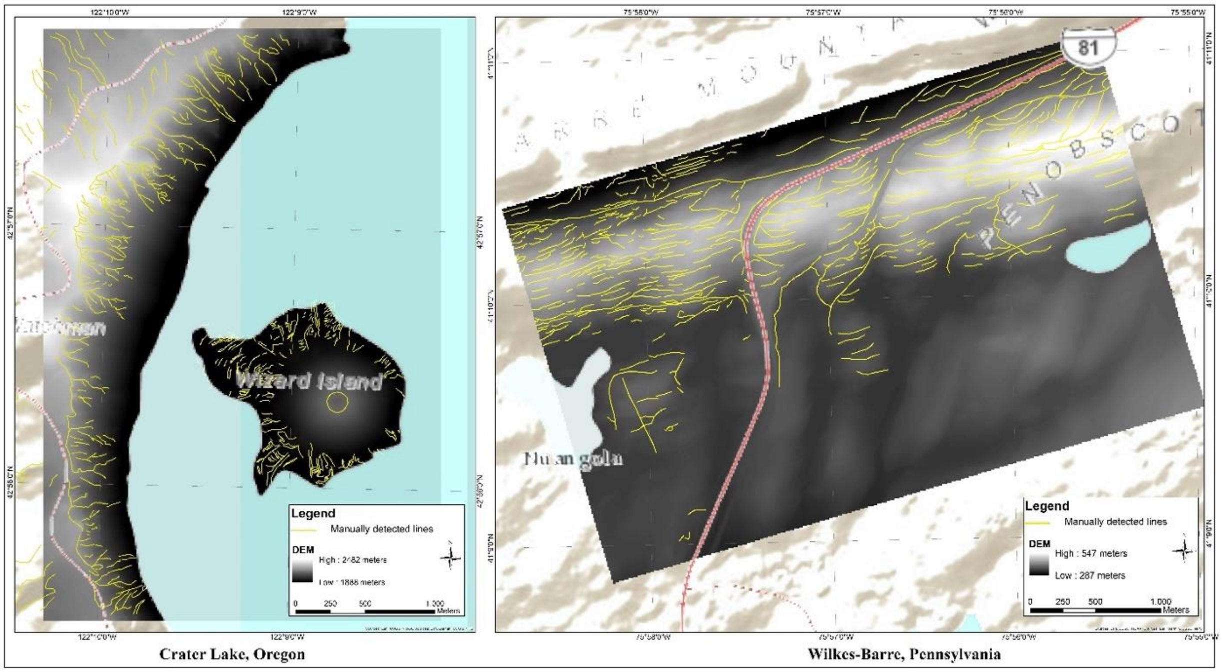


Figure 5.9 Synthetically-generated topographic dataset with seven curvilinear ridges exhibiting a decreasing peak sharpness from left to right. Dark blue pixels represent known ridges that were equal in length for all seven ridges. Outputs of the four terrain feature extraction algorithms overlaid on the DTM and shown at two different scales (Syzdykbayev et al., 2020a)

5.4.2 Real-world Dataset

For the real-world data, two regions with different patterns of terrain features were selected. The first is a linear topographic system south of Wilkes-Barre, Pennsylvania that is part of the Ridge and Valley Province. The second is a radial topographic system for Crater Lake, Oregon (Figure 5.10). For these two regions, there is no known baseline, so LTFs were manually extracted to be used as a baseline. Table 5.2 shows the accuracy, precision, recall, and Cohen's Kappa coefficient (evaluation metrics) of the results for the real-world datasets produced by Shade-relief and the three other algorithms and compared to manually-extracted terrain features.



Crater Lake, Oregon

Wilkes-Barre, Pennsylvania

Figure 5.10 Real-world datasets with manually extracted LTFs (Syzdykbayev et al., 2020a)

Table 5.2 Cohen's Kappa coefficient, accuracy, precision, recall for a real-world dataset with each algorithm's results compared to manually extracted LTFs (Syzdykbayev et al., 2020a)

Dataset	Algorithm	Cohen's Kappa	Accuracy	Precision	Recall
Wilkes-Barre	Shade-relief	0.056	0.940	0.063	0.125
	Geomorphon	0.007	0.811	0.026	0.209
	Curvature	0.061	0.951	0.073	0.1
	Probabilistic	0.058	0.944	0.066	0.117
Crater Lake	Shade-relief	0.115	0.961	0.109	0.172
	Geomorphon	0.123	0.905	0.088	0.481
	Curvature	0.116	0.968	0.126	0.138
	Probabilistic	0.113	0.927	0.86	0.333

For Wilkes-Barre (Figure 5.11), a comparison of the results of the four algorithms against the manually-extracted LTFs shows a similar pattern where Geomorphon had the lowest values for all four metrics (Table 5.2). Curvature resulted in the highest values of Cohen's Kappa coefficient (0.061), accuracy (0.951), and precision (0.073), while Geomorphon had the highest recall (0.209) value.

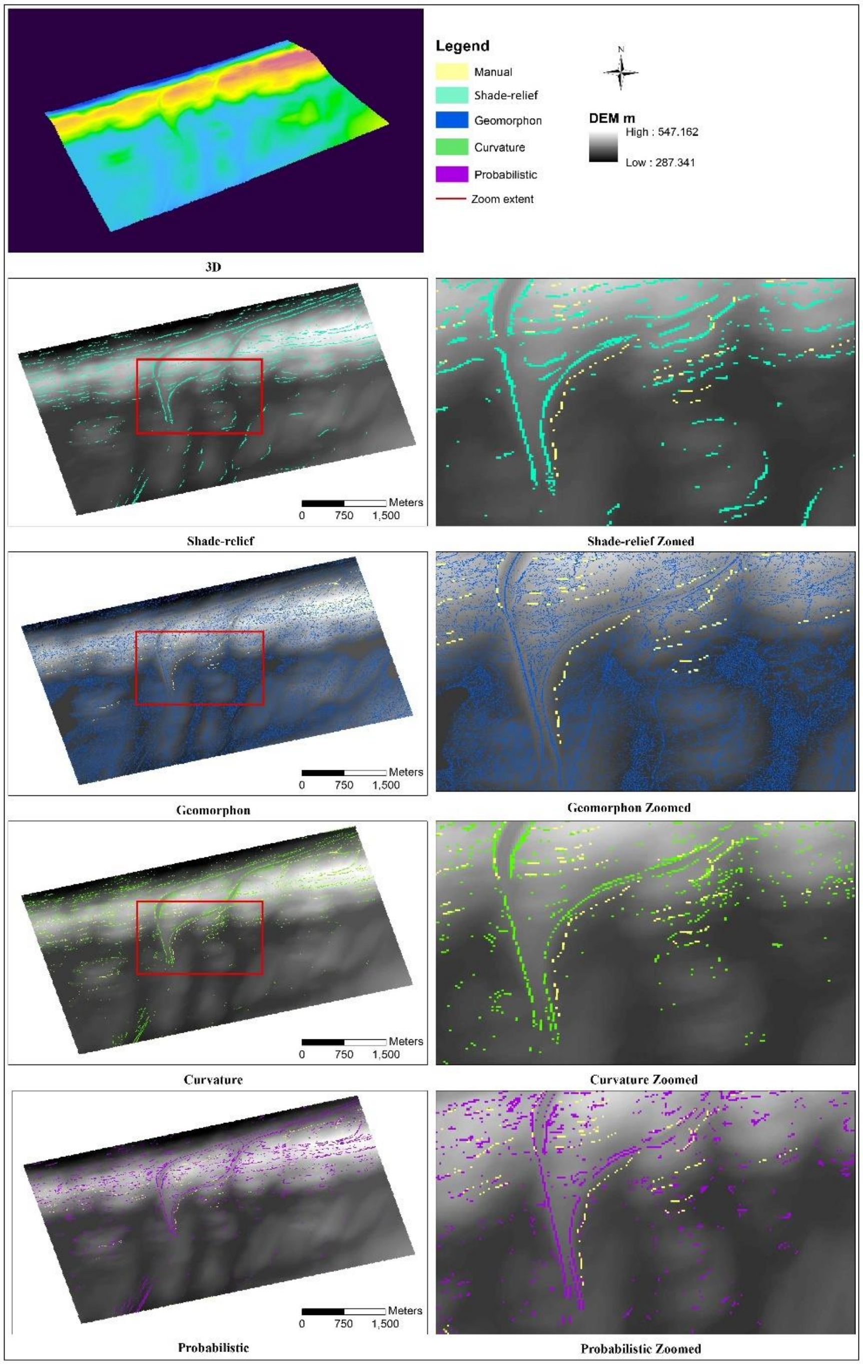


Figure 5.11 Wilkes-Barre, 3D representation, outputs of the four terrain features extraction algorithms overlaid on manually-extracted terrain features and DTM (Syzdykbayev et al., 2020a)

For the topographic dataset of Crater Lake (Figures 5.12 and 5.13), there was far more variability in the results. Comparing the results of the four algorithms against the manually-extracted features shows that Geomorphon had the highest Cohen's Kappa coefficient (0.123) and recall (0.481) (Table 5.2). Shade-relief, Probabilistic, and Curvature resulted in identical evaluation metric values where Cohen's Kappa coefficient is equal to 0.115, 0.113, and 0.116, respectively.

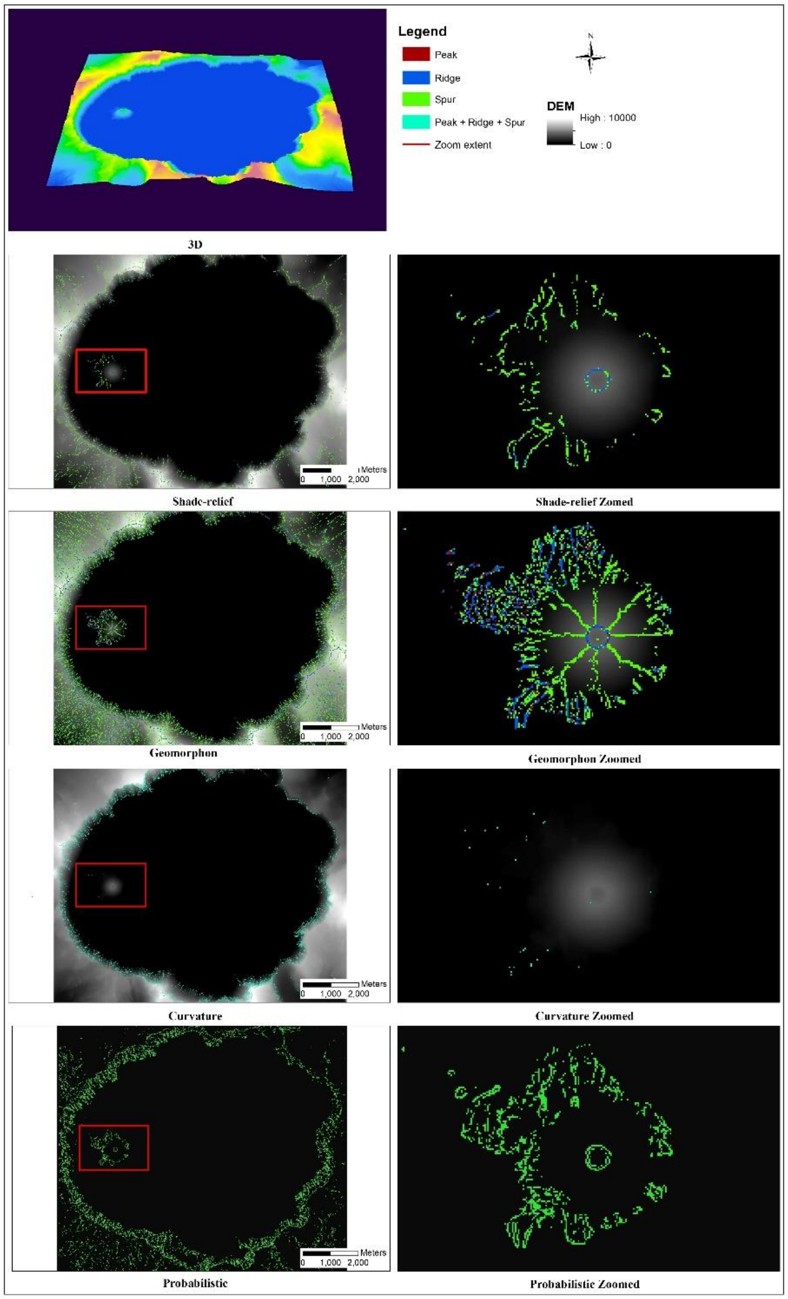


Figure 5.12 Crater Lake, 3D representation, outputs of the four terrain features extraction algorithms overlaid on the DTM, with a distinction between feature types where possible (Syzdykbayev et al., 2020a)

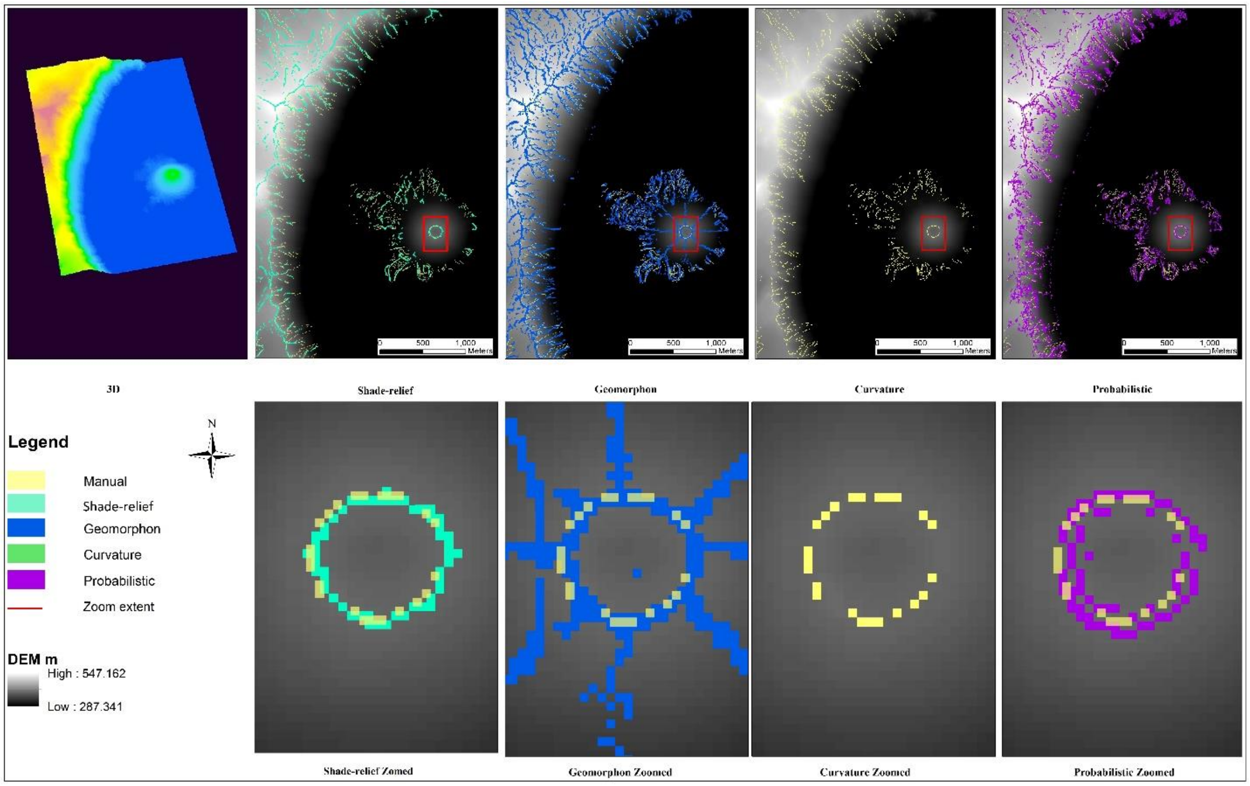


Figure 5.13 Crater Lake, 3D representation, outputs of the four terrain features extraction algorithms overlaid on manually-extracted terrain features and the DTM (Syzdykbayev et al., 2020a)

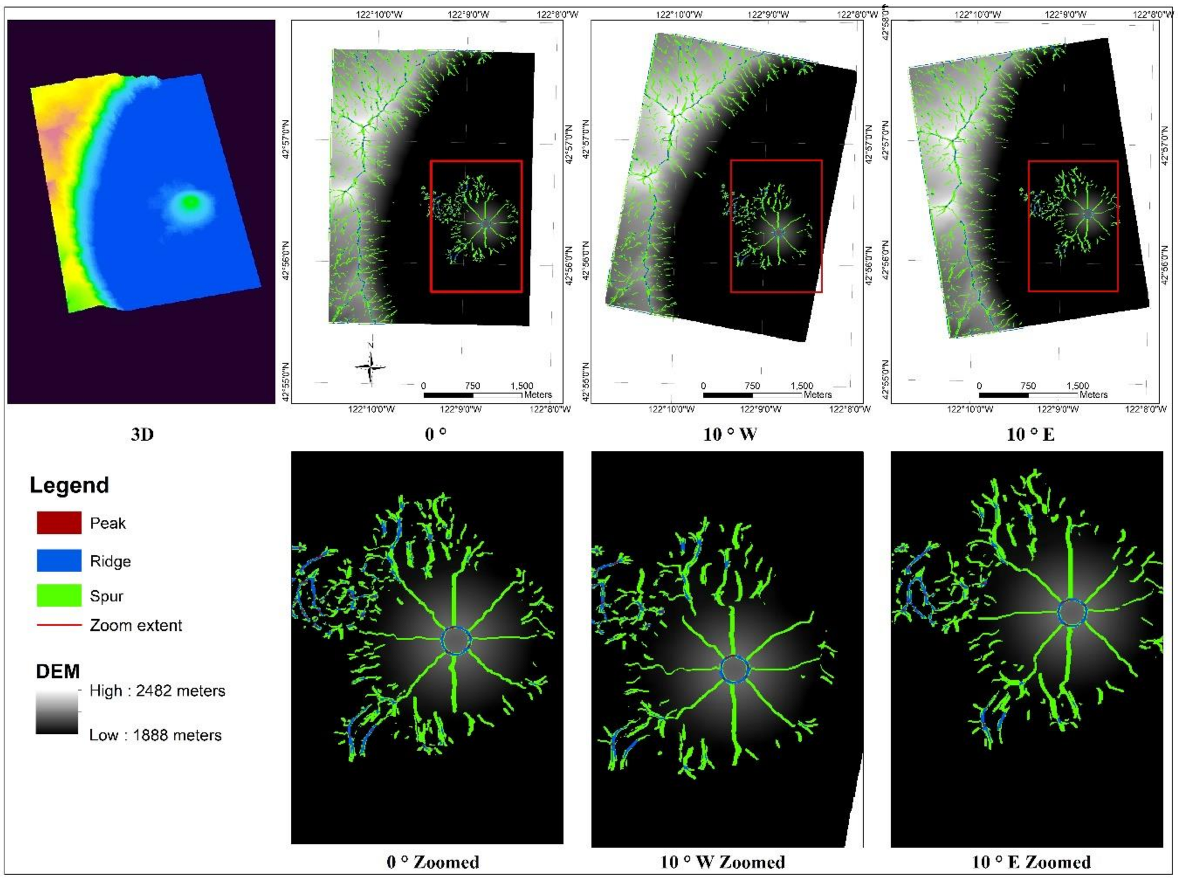
5.5 Discussion

Extraction of LTFs based on geo-morphometry and DTM is a challenging process. The main drawbacks of existing LTFs extraction algorithms are that they are sensitive to the threshold value or can have FP. The results of the four algorithms were compared against the manually-extracted linear features using specific metrics (see Tables 5.1 and 5.2). The performance of each algorithm depends on factors such as dataset used, synthetically-generated or real-world data, and type of terrain.

The results of the synthetically-generated dataset (Table 5.1) show that Shade-relief more often resulted in the highest evaluation metric values, suggesting that it outperforms, or is at the very least competitive against, the other three comparable algorithms. In one instance, for a synthetically-generated square pyramid (Figure 5.7), Shade-relief and Curvature both precisely identified four ridges, Geomorphon identified ridges in every other pixel, and Probabilistic identified what should have been a 1-pixel width linear feature as having a 3-pixel width. In one complex synthetically-generated ridge systems (Figures 5.8 and 5.9), Shade-relief, Probabilistic, and Geomorphon were able to identify all known features. Curvature failed to detect more rounded topographic ridges, likely due to a limiting threshold value, which in this case is a multiple of the curvature's standard deviation. For Curvature, more rounded ridges/peaks resulted in a wider line or larger circle representing the detected features. In the half-sphere example (Figure 5.6), Curvature completely failed to detect the peak, which is why Cohen's Kappa coefficient is so close to 0 (Table 5.1). It should be noted that in Curvature the multiplication of the curvature's standard deviation was used as a threshold to extract LTFs. The threshold value is sensitive to the smoothing filters because smoothing decreases noise in the DTM that affects the derivative values of the DTM (Rana, 2006). For complex real-world data, an optimal threshold value is usually determined through a manual process where the value is iteratively selected and compared either visually or by checking against a known dataset; this process was used in Pirotti & Tarolli, (2010), Syzdykbayev et al., (2020b), and X. Zhou et al., (2019), and showed that an optimal threshold value is two times the curvature's standard deviation.

For the real-world datasets (Figures 5.10–5.13), Curvature resulted in the highest Cohen's Kappa coefficient, accuracy, and precision (Table 5.2). However, Shade-relief and Probabilistic both had values in a narrow range compared to Curvature's results. This suggests that Curvature, Probabilistic, and Shade-relief all perform similarly and that Geomorphon results include far too many poorly detected (incomplete) and FP terrain features.

The main difference between Shade-relief and Geomorphon is that the former can detect spurs with higher accuracy compared with the latter. Geomorphon tends to fail when applied to a dataset with multiple topographic expressions (from smooth to highly variable). In order to further test FP spurs that were generated by Geomorphon, the input DTM was rotated by 10° to the west and east. The detected spurs from Geomorphon did not change their direction even after rotation (Figure 5.14).


3D

0 °

10 ° W

10 ° E

0 ° Zoomed

10 ° W Zoomed

10 ° E Zoomed

Figure 5.14 Raster image (representing Crater Lake) was rotated by 10° to the west and east and overlaid with the ridges, spurs, and peaks detected by Geomorphon (Syzdykbayev et al., 2020a)

The high accuracy of spurs detection by Shade-relief is due to the removal of the detected features that do not persist under slight azimuth changes. An experiment was conducted with several azimuth values (Figures 5.15 and 5.16), and it was revealed that without a slight azimuth change, the spur lines detected by Shade-relief were identical to the spur lines detected by Geomorphon.

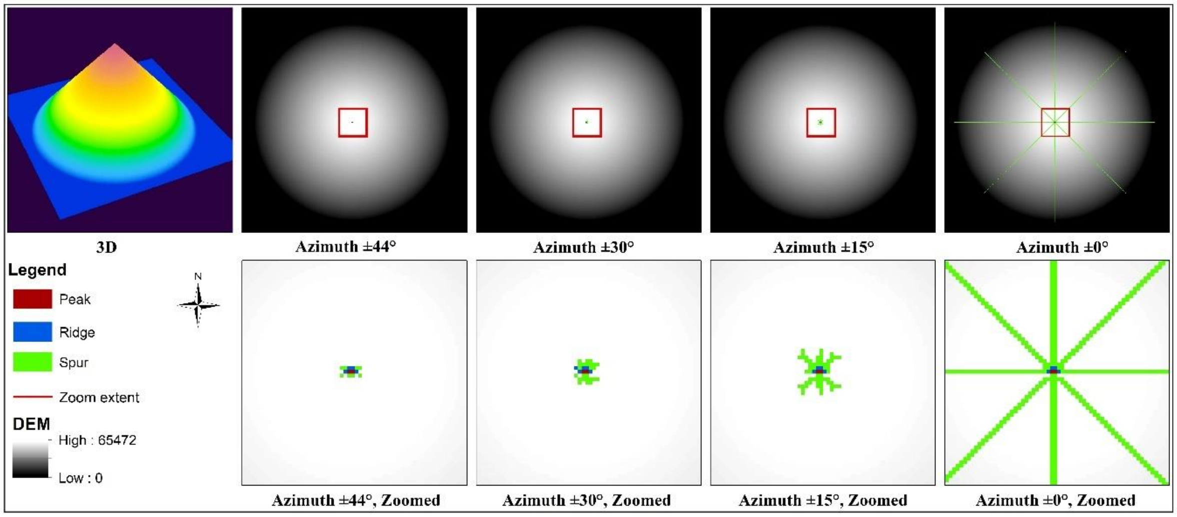


Figure 5.15 Synthetically-generated conical DTM with only one peak with 3D representation (left image) and the outputs of Shade-relief with different azimuth values overlain on the DTM (four right images) (Syzdykbayev et al., 2020a)

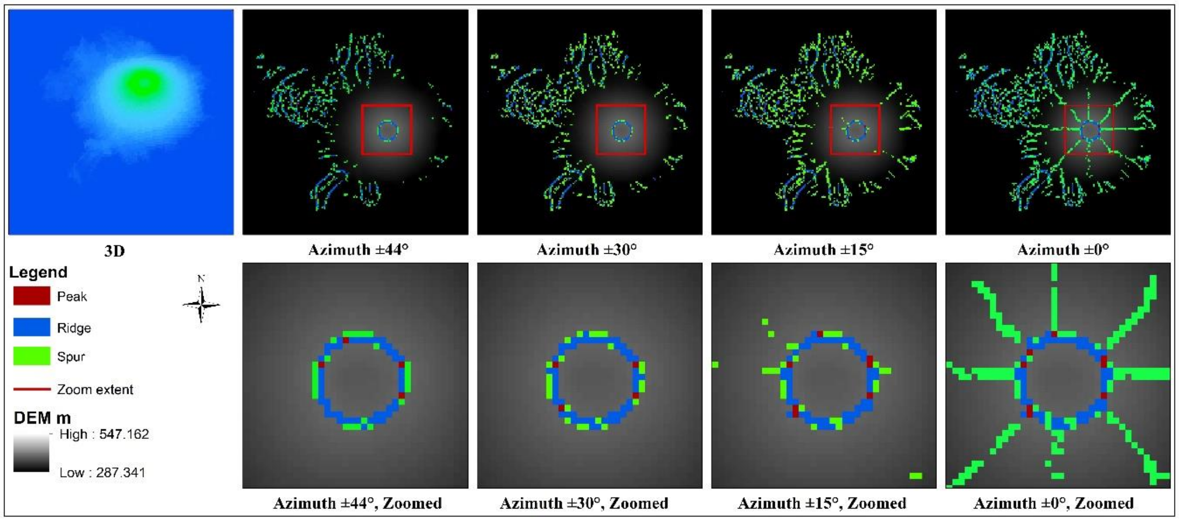


Figure 5.16 Crater Lake DTM with 3D representation (top left image) and the outputs of Shade-relief with different azimuth values overlaid on the DTM (top four right and four bottom images) (Syzdykbayev et al., 2020a)

6.0 Geospatial Object Detection: Topological KB Method

In chapter 4, we demonstrated that it is feasible to generate candidate polygons using PH, thereby detecting geospatial objects by selecting from these polygons, relying solely on topological information embedded in them. However, the results indicated a moderate number of FP in certain study areas. Consequently, we postulated the following research direction (Syzdykbayev et al., 2020b):

> *"Another future work is to compare and analyze the result of this current work with the result of an extended version of the proposed PH where it takes fused data from LiDAR and other datasets, such as NDVI, visual/NIR satellite imagery, apparent thermal inertia, vegetative cover, bedrock lithology, and/or soil types to detect landslides."*

6.1 Methodology

We developed a Topological KB method where we use additional topological information along with geometrical and contextual information on candidate polygons derived from PH. The method is designed to tackle RQ2 by reframing the geospatial object detection challenge into a hypothesis testing problem. This transformation involves formulating various knowledge structures and rules using information about the detected geospatial object. Figure 6.1 illustrates the modification to the knowledge-based approach by Cheng & Han, (2016) with the integration of topological information alongside geometrical and contextual information.

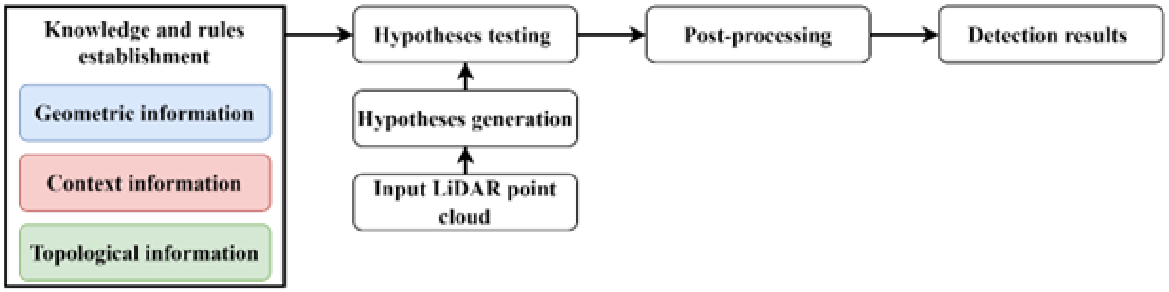


Figure 6.1 Topological KB method

Our method takes LiDAR-derived DTM data as input and the process is illustrated in Figure 6.2 and in Figure 6.3. First, we extract LTFs from DTM and then, using PH, form candidate polygons from these LTFs. The last step involves applying rules for detection which are based on topological, geometrical, and contextual information.

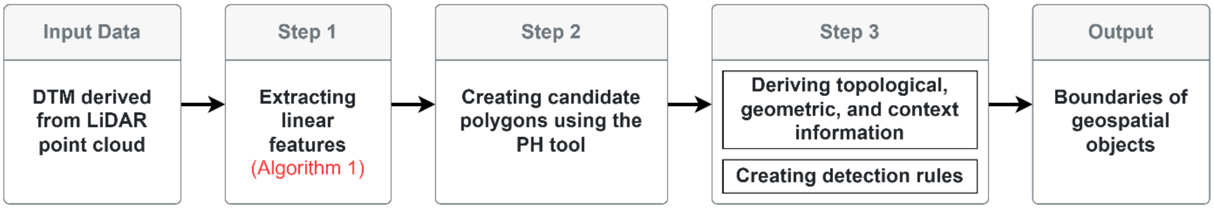

Input Data
DTM derived from LiDAR point cloud
Step 1
Extracting linear features
(Algorithm 1)
Step 2
Creating candidate polygons using the PH tool
Step 3
Deriving topological, geometric, and context information
Creating detection rules
Output
Boundaries of geospatial objects

Figure 6.2 Workflow of the Topological KB geospatial object detection method

6.1.1 Using Topological, Geometrical, and Contextual Information for Detection Rules

Before formulating detection rules, it is essential to derive and embed topological, geometrical, and contextual information for each candidate polygon. We performed a comprehensive literature review to find works where the knowledge-based approach was used to detect landslides and compiled a list of landslide detection rules (Table 6.1). From this list, we selected and implemented the rules in each information category that are common in most works (Table 6.2).

Table 6.1 List of works on landslide detection rules

	Parameter	(Martha et al., 2010)	(Martha et al., 2011)	(Hölbling et al., 2012)	(Liu et al., 2012)	(Rau et al., 2013)	(Blaschke et al., 2014)	(Althuwaynee et al., 2014)	(Hong et al., 2015)	(Mezaal et al., 2017)	(Fanos et al., 2018)	(Bunn et al., 2019)	(Bacha et al., 2020)
Morphological	Slope (degrees: min: 0 and max: 90)	>12	>10	10-60	>10	>15	>7	<72		>20.3	4-75	>30	>20
	Hill-shade (pixel value: min: 0 and max: 255)	<92	<57.9										
	Brightness (pixel value: min: 0 and max: 255)	65<90	<40.1			μ−2.5σ	>104						>255
	Elevation (meters)	>30	>30				1600						
	Curvature (pixel value: min: -1 and max: 1)				-1&1							-1&1	
	Surface roughness (index value min: 0 and higher)				<2.4					>0.27		High	
Geometrical	Length (meters)							6-61				<500	
	Width (meters)							6-20					
	Length/width (ratio)	>3	>3										>5
Hydrological	Stream order (8-bit pixel value: 0-255)	>5											
	Distance from water body (meters)	100	50										
Spectral	Red band (8-bit pixel value: 0-255)	60-90	>64										
	NIR band (8-bit pixel value: 0-255)								>255				

	NDVI (pixel value: min: -1 and max: 1)	>0.12	>0.18						>0.18	>0.176			<1

The topological information, which can be obtained and embedded into candidate polygons using PH, includes:

1. Birth time of the circle, denoting the instance when topological information starts to emerge.
2. Death time of the circle, denoting the moment when topological information ceases to exist.
3. Lifetime of the circle, denoting the interval between the birth time and the death time.

The geometrical information, which can be obtained and embedded into candidate polygons using tools like Geographic Information Systems (GIS), includes:

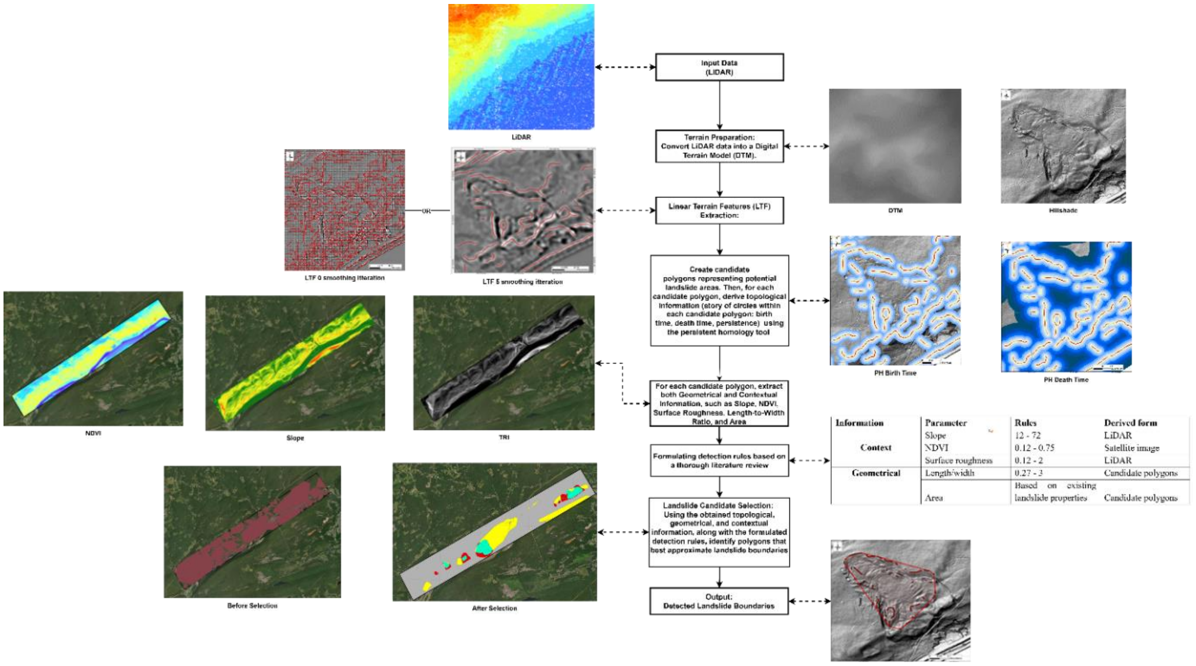
1. Size, overall magnitude, or dimensions of a geospatial object.
2. Ratio between length and width of a geospatial object, a measure of the object's proportion or aspect ratio.

The contextual information, which can also be derived and embedded into candidate polygons using GIS tools, includes:

1. The slope of a region, the angle or steepness of the terrain.
2. Roughness of a terrain, a measure of the terrain's irregularity or complexity.
3. Normalized Difference Vegetation Index (NDVI) score of a region, an indicator of live green vegetation density and health.

Table 6.2 List of geometrical and contextual rules implemented to detect landslides

Information	Parameter	Rules	Derived form
Context	Slope (degrees: min:0 and max: 90)	12 – 72	LiDAR
	NDVI (pixel value: min: -1 and max: 1)	0.12 - 0.75	Satellite image
	Surface roughness (index value min: 0 and higher)	0.12 - 2	LiDAR
Geometrical	Length/width (unit)	0.27 - 3	Candidate polygons
	Area (meters)	Based on existing landslide properties	Candidate polygons


LiDAR
Input Data (LiDAR)
Terrain Preparation: Convert LiDAR data into a Digital Terrain Model (DTM).
DTM
Hillshade
LTF 0 smoothing iteration
OR
LTF 5 smoothing iteration
Linear Terrain Features (LTF) Extraction:
Create candidate polygons representing potential landslide areas. Then, for each candidate polygon, derive topological information (story of circles within each candidate polygon: birth time, death time, persistence) using the persistent homology tool
PH Birth Time
PH Death Time
NDVI
Slope
TRI
For each candidate polygon, extract both Geometrical and Contextual information, such as Slope, NDVI, Surface Roughness, Length-to-Width Ratio, and Area
Formulating detection rules based on a thorough literature review
Before Selection
After Selection
Landslide Candidate Selection: Using the obtained topological, geometrical, and contextual information, along with the formulated detection rules, identify polygons that best approximate landslide boundaries
Output: Detected Landslide Boundaries

Information	Parameter	Rules	Derived form
Context	Slope	12 - 72	LiDAR
	NDVI	0.12 - 0.75	Satellite image
	Surface roughness	0.12 - 2	LiDAR
Geometrical	Length/width	0.27 - 3	Candidate polygons
	Area	Based on existing landslide properties	Candidate polygons

Figure 6.3 Workflow of the Topological KB geospatial object detection method with visualization of each step

6.2 Datasets and Experiment

The characteristics of the input data for the experiment are shown in Table 4.1. A DTM offers a bare-earth depiction of the terrain, devoid of natural or artificial elements such as vegetation or buildings. The output of the experiment, detected landslides, were compared against landslide inventory maps which record landslide locations along with supplementary data, including occurrence dates and landslide types (Guzzetti et al., 2012b).

We conducted experiments to test our Topological KB. The results were evaluated by comparing them with the ground truth, the results of the work by (Syzdykbayev et al., 2020b), where only topological filters were used on candidate polygons, and the findings of the knowledge-based method implemented by Bunn et al (2019) who introduced a method known as SICCM for landslide detection. The same five study areas described in Chapter 4 were used in the experiment (see Figure 4.7 and Table 4.2).

Three experiments were conducted as follows:

- Experiment 1: Using no filters, only raw, unfiltered candidate polygons.
- Experiment 2: Using only geometrical and contextual information as filters on candidate polygons.
- Experiment 3: Using topological information alongside geometrical and contextual information.

These experiments collectively provided a holistic view of the implications of different filtering approaches in the task of geospatial object detection.

We ran three LTFs algorithms (Shade-relief, Curvature, Geomorphon) on three different pixel sizes and used six smoothing iterations (Table 6.3). Furthermore, to identify the most effective geometrical and contextual information, we undertook a comprehensive analysis of all possible subsets. This analysis encompassed cases ranging from utilization of all possible combinations of geometrical and contextual information with and without addition of topological information.

The total number of comparisons were: 3 (LTFs extraction algorithm) * 3 (pixel size) *6 (Smoothing iteration) * 4! (Geometrical information) * 6! (Contextual information) = 933120

Table 6.3 Parameters used in the experiment

Parameter name	**Parameters**
DTM pixel size	1m, 5m, 10m
Smoothing iteration	0, 2, 5, 10, 15, 20
LTFs extraction algorithm	Shade-relief (Syzdykbayev et al., 2020a), Curvature (Pirotti & Tarolli, 2010), Geomorphon (Jasiewicz & Stepinski, 2013)
Topological information	Birth of the circle, Lifetime of the circle
Geometrical information	Min length-width ratio of the candite polygon, Max Min length-width ratio of the candite polygon, Min area, Max area.
Contextual information	Min slope, Max slope, Min TRI, Max TRI, Min NDVI, Max NDVI

6.3 Results

Detected landslides were compared to landslide inventory maps and with the results of the work by (Syzdykbayev et al., 2020b), in all five study areas. In addition, for Study Areas 2 and 4, the results were also compared with the results of the work by Bunn et al. (2019). The three LTFs extraction algorithms, six smoothing iterations, and different three-pixel sizes were applied, and accuracy, precision, recall, Cohen’s Kappa coefficient, and F-1 score evaluation metrics were calculated. Table 6.4. shows the results of these metrics with the highest F-1 score for each study area.

Table 6.4 LTFs extraction algorithms, number of existing, detected, and intersected landslides, accuracy, precision, recall, Cohen's Kappa coefficient and F1 score for each study area

	Study Area 1	**Study Area 2**	**Study Area 3**	**Study Area 4**	**Study Area 5**
LTFs extraction algorithm	Geomorphon	Geomorphon	Shade-relief	Curvature	Curvature
Pixel size	1	1	1	5	5
Number of smoothing iterations	1	10	20	5	1
Accuracy	0.95	0.66	0.47	0.59	0.97
Precision	0.38	0.33	0.28	0.51	0.36
Recall	0.80	0.86	0.85	0.82	0.58
Cohen's Kappa coefficient	0.50	0.31	0.12	0.24	0.43
F1 Score	**0.52**	**0.48**	**0.43**	**0.64**	**0.45**

Findings for Study Area 1:

- *Experiment 1:* The best F1 score (0.34) was obtained with pixel sizes of 5 using Shade-relief (Figure 6.4 (b)). Compared with Syzdykbayev et al., (2020c), the unfiltered results were superior at a pixel size of 1 (Figure 6.4 (a)) but deteriorated at pixel size 5 (Figure 6.4 (b)). Results with pixel size 10 lacked consistency across smoothing iterations (Figure 6.4 (c)).
- *Experiment 2:* An F1 score peak of 0.5248 was observed at pixel size 1 using Geomorphon (Figure 6.4 (d)). When compared with Syzdykbayev et al., (2020c), a Topological KB geospatial object detection method outperformed in all configurations except for the pixel size of 10 subjected to 20 smoothing iterations (Figures 6.4 (d, e, f)).

- *Experiment 3:* The highest F1 score was 0.5247, again at a pixel size of 1 (Figure 6.4 (g)). Compared with Syzdykbayev et al., (2020c), this combined approach outperformed solely at pixel size 1 (Figures 6.4 (g, h, i)).

In summary, the most favorable results for Study Area 1 were obtained when only geometrical and contextual filters were applied to pixel size of 1 and with Geomorphon (Figure 6.4 (d)). From the original set of 10 rules (comprising 4 geometrical and 6 contextual rules), a subset of 7 rules (consisting of 2 geometrical and 5 contextual rules) was chosen to attain these results (Table 6.5).

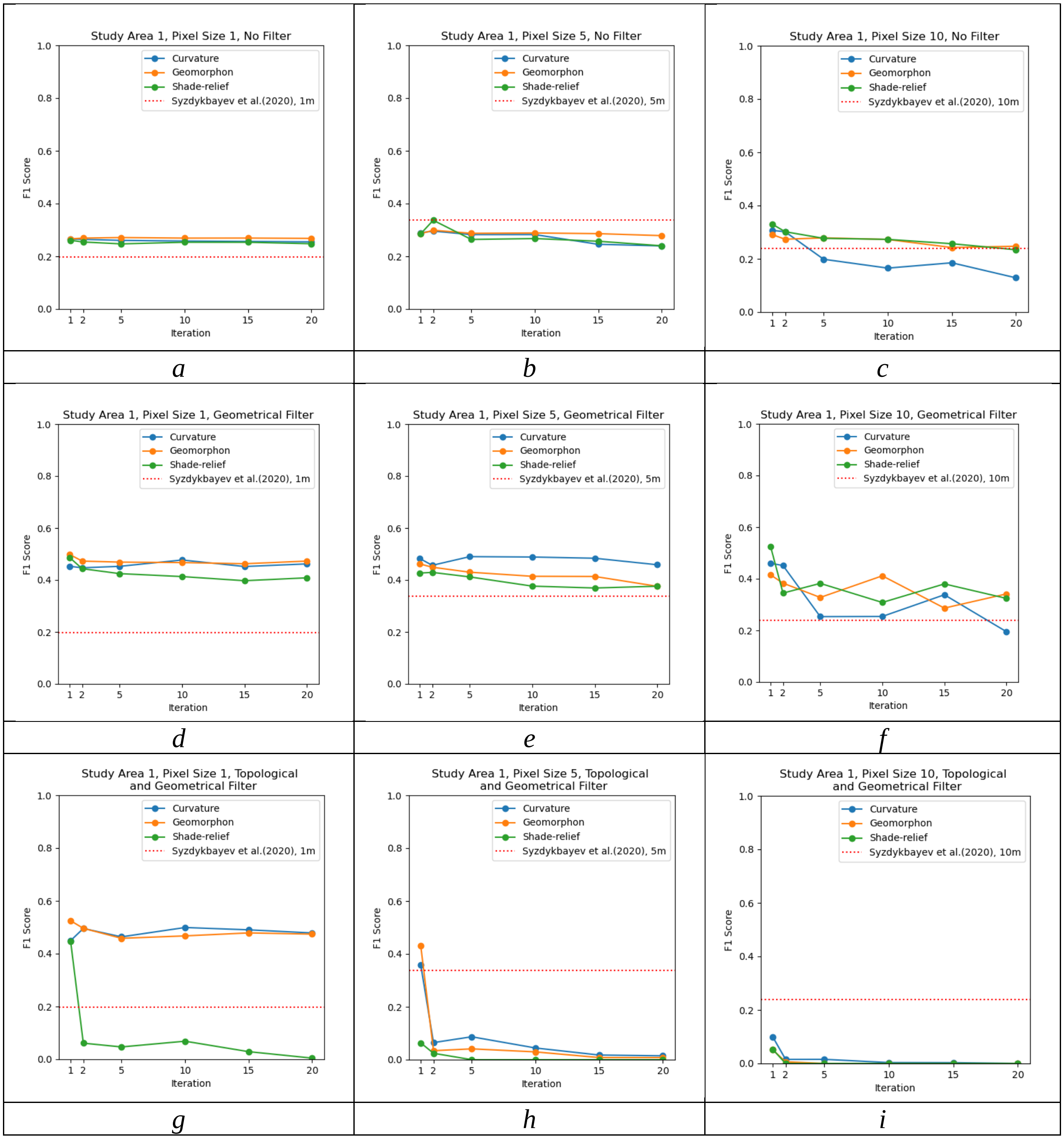


Figure 6.4 Results of the three LTFs extraction algorithms with different pixel sizes, smoothing iterations and three evaluations as a measure of F1 score; Study Area 1: Pennsylvania

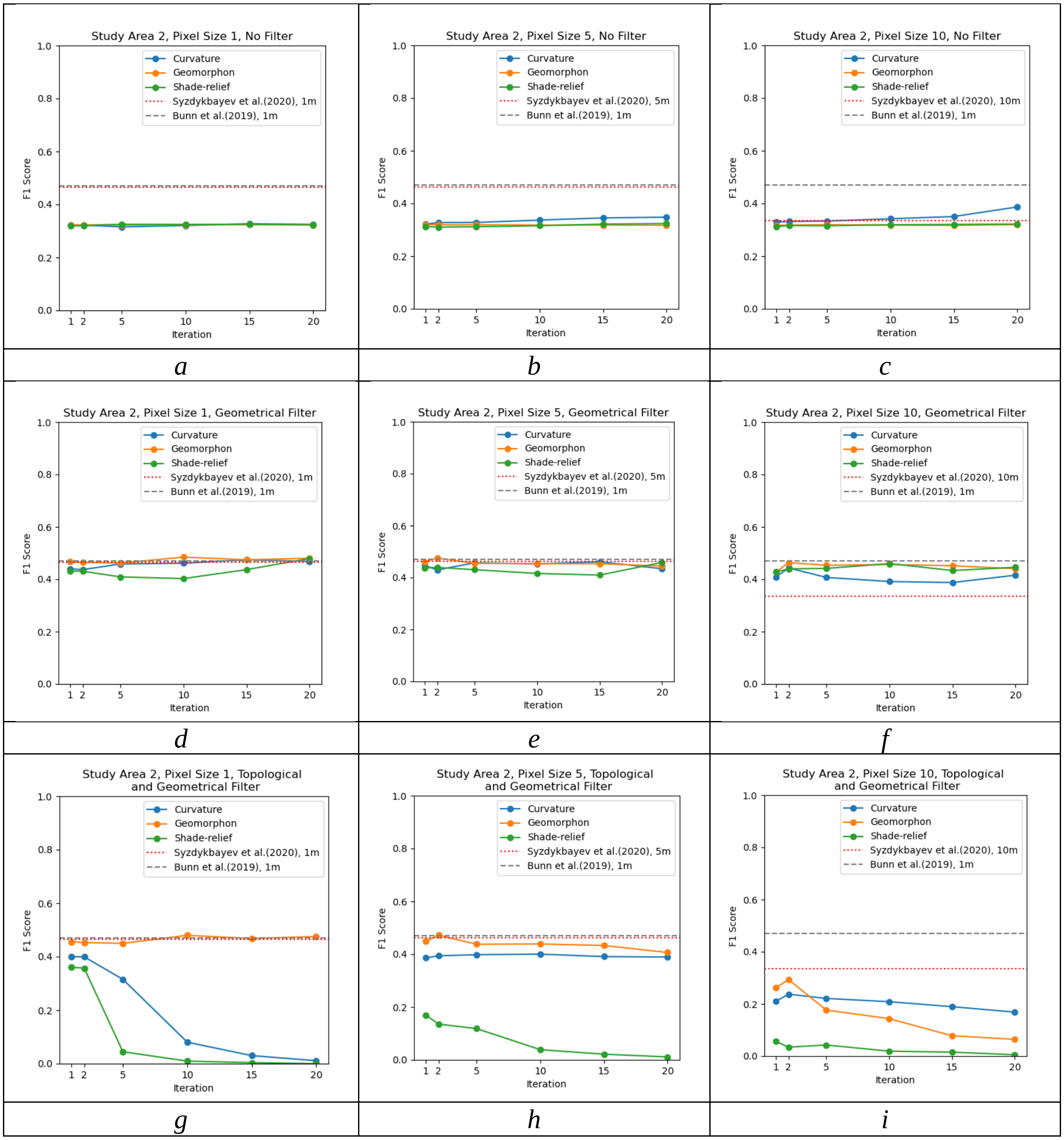
Table 6.5 List of geometrical and contextual rules used to obtain the highest F1 score; Study Area 1: Pennsylvania

Information	Parameter	Rules	Rules Used
Context	Slope	12 - 72	12 - 72
	NDVI	0.12 - 0.75	0.12 - 0.75
	Surface roughness	0.12 - 2	0 - 2
Geometrical	Length/width	0.27 - 3	0 - 3
	Area	261-9746736	0-9746736

Findings for Study Area 2:

- *Experiment 1:* The F1 score peaked at 0.39 with a pixel size of 10 using Curvature (Figure 6.5 (c)). When compared with Bunn et al. (2019), F1 scores for a Topological KB geospatial object detection method were consistently lower across all pixel sizes (Figures 6.5 (a, b, c)). Against Syzdykbayev et al., (2020c), we observed reduced F1 scores at pixel sizes 1 and 5, while the results closely matched at pixel size 10 (Figures 6.5 (a, b, c)).
- *Experiment 2:* The highest F1 score reached 0.48 at pixel size 1 with Geomorphon (Figure 6.5 (d)). When compared with Bunn et al. (2019), results across all pixel sizes were closely matched, with a slight edge in a Topological KB geospatial object detection method in some instances (Figures 6.5 (d, e, f)). Against Syzdykbayev et al., (2020c), we matched their F1 scores at pixel sizes 1 and 5 but surpassed them at pixel size 10 (Figures 6.5 (d, e, f)).
- *Experiment 3:* The highest F1 score was 0.47 at pixel size 10 using Geomorphon (Figure 6.5 (g)). In comparison with both Bunn et al. (2019) and Syzdykbayev et al., (2020c), outcomes of a Topological KB geospatial object detection method were closely matched for pixel sizes 1 and 5 but lagged at pixel size 10 (Figures 6.5 (g, h, i)).

In summary, the most favorable results for Study Area 2 were obtained when only geometrical and contextual filters were applied to pixel sizes of 1 and with Geomorphon (Figure 6.5 (d)). From the original set of 10 rules (comprising 4 geometrical and 6 contextual rules), a subset of 5 rules (consisting of 1 geometrical and 4 contextual rules) was chosen to attain these results (Table 6.6).



a

b

c

d

e

f

g

h

i

Figure 6.5 Results of the three LTFs extraction algorithms with different pixel sizes, smoothing iterations and three evaluations as a measure of F1 score; Study Area 2: Oregon

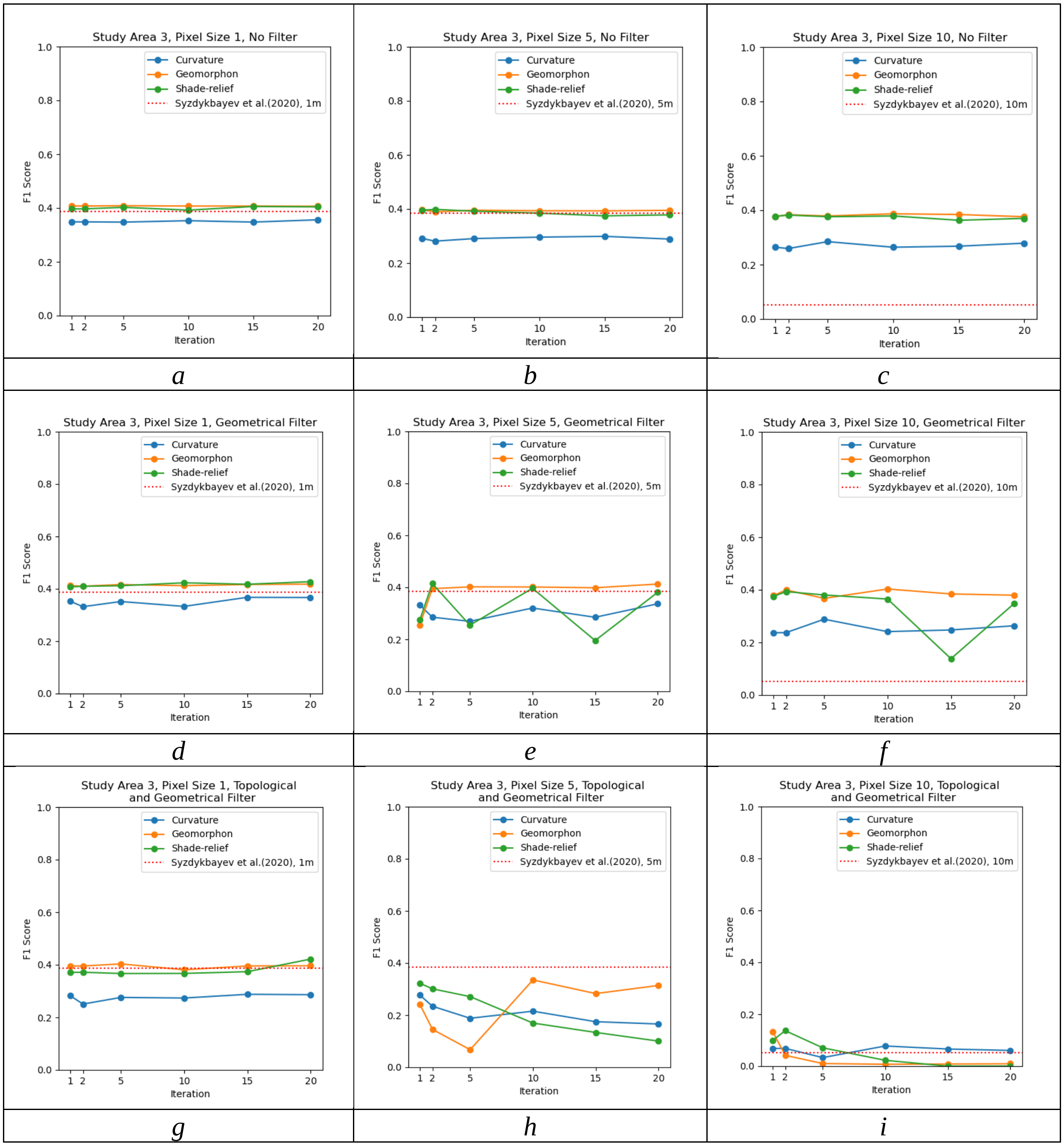
Table 6.6 List of geometrical and contextual rules used to obtain the highest F1 score; Study Area 2: Oregon

Information	Parameter	Rules	Rules Used
Context	Slope (degrees: min:0 and max: 90)	12 - 72	12 - 72
	NDVI (pixel value: min: -1 and max: 1)	0.12 - 0.75	0 - 0.75
	Surface roughness (index value min: 0 and higher)	0.12 - 2	0 - 2
Geometrical	Length/width (unit)	0.27 - 3	0 - ∞
	Area (meters)	261-12443892	261 - ∞

Findings for Study Area 3:

- *Experiment 1:* An optimal F1 score of 0.40 was achieved at pixel size 1 using Geomorphon (Figure 6.6 (a)). Compared with Syzdykbayev et al., (2020c), the absence of filters in this study showed improved results at pixel size 1 (Figures 6.6 (a, b, c)).
- *Experiment 2:* The highest F1 score achieved was 0.43, observed at pixel size 1 when employing Shade-relief (Figure 6.6 (d)). Results comparison with Syzdykbayev et al., (2020c) indicated that using geometrical and contextual filters improved results across all pixel sizes and smoothing iterations (Figures 6.6 (d, e, f)).
- *Experiment 3:* The highest F1 score observed was 0.42 at pixel size 1 (Figure 6.6 (g)). When compared with the results of Syzdykbayev et al., (2020c), the use of this combined filter approach led to better results, but exclusively at pixel size 1 (Figures 6.6 (g, h, i)).

In summary, the best results for Study Area 3 were achieved when only geometrical and contextual filters were applied to pixel sizes of 1 and with Shade-relief (Figure 6.6 (d)). From the original set of 10 rules (comprising 4 geometrical and 6 contextual rules), a subset of 9 rules (consisting of 3 geometrical and 6 contextual rules) was chosen to attain these results (Table 6.7).



a

b

c

d

e

f

g

h

i

Figure 6.6 Results of the three LTFs extraction algorithms with different pixel sizes, smoothing iterations and three evaluations as a measure of F1 score; Study Area 3: Colorado

Table 6.7 List of geometrical and contextual rules used to obtain the highest F1 score; Study Area 3: Colorado

Information	Parameter	Rules	Rules Used
Context	Slope (degrees: min:0 and max: 90)	12 - 72	0 - 72
	NDVI (pixel value: min: -1 and max: 1)	0.12 - 0.75	0.12 - 0.75
	Surface roughness (index value min: 0 and higher)	0.12 - 2	0.12 - 2
Geometrical	Length/width (unit)	0.27 - 3	0 - 3
	Area (meters)	1377-6231540	1377-6231540

Findings for Study Area 4:

- *Experiment 1:* The optimal F1 score was 0.63 at pixel size 5 using Geomorphon (Figure 6.7 (b)). Compared with Bunn et al. (2019), this experiment consistently yielded a higher F1 score across all cases. When compared with Syzdykbayev et al., (2020c), F1 score was lower for pixel sizes 1 and 10, but closely matched their results at pixel size 5 (Figures 6.7 (a, b, c)).
- *Experiment 2:* The highest F1 score of 0.64 was observed at pixel size 5 using Curvature (Figure 6.7 (e)). When compared with Bunn et al. (2019), F1 score was higher across all pixel sizes. Comparing with Syzdykbayev et al., (2020c), the results indicated a lower F1 score at pixel sizes 1 and 10, while the score at pixel size 5 was almost identical to their findings (Figures 6.7 (d, e, f)).

- *Experiment 3:* An F1 score of 0.63 was achieved at pixel size 1 using Geomorphon (Figure 6.7 (g)). Compared with Bunn et al. (2019), the scores were higher for pixel sizes 1 and 5 but lower for pixel size 10. When compared with Syzdykbayev et al., (2020c), the scores at pixel size 5 were similar, while the results for pixel sizes 1 and 10 were lower (Figures 6.7 (g, h, i)).

In summary, the most favorable results for Study Area 4 were obtained when only geometrical and contextual filters were applied to pixel sizes of 5 and with Curvature (Figure 6.7 (e)). From the original set of 10 rules (comprising 4 geometrical and 6 contextual rules), a subset of 5 rules (consisting of 2 geometrical and 3 contextual rules) was chosen to attain these results (Table 6.8).

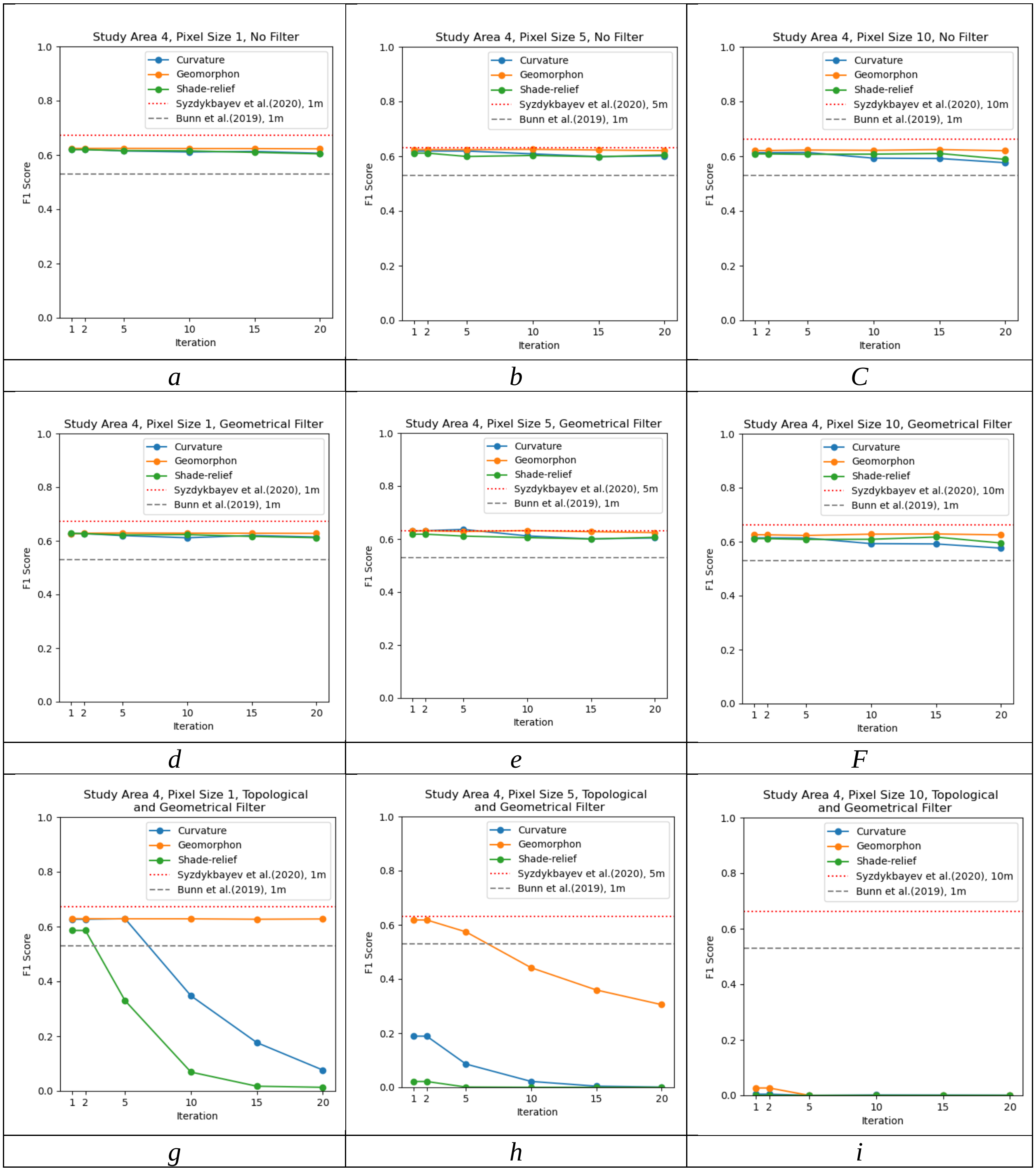


Figure 6.7 Results of the three LTFs extraction algorithms with different pixel sizes, smoothing iterations and three evaluations as a measure of F1 score; Study Area 4: Oregon

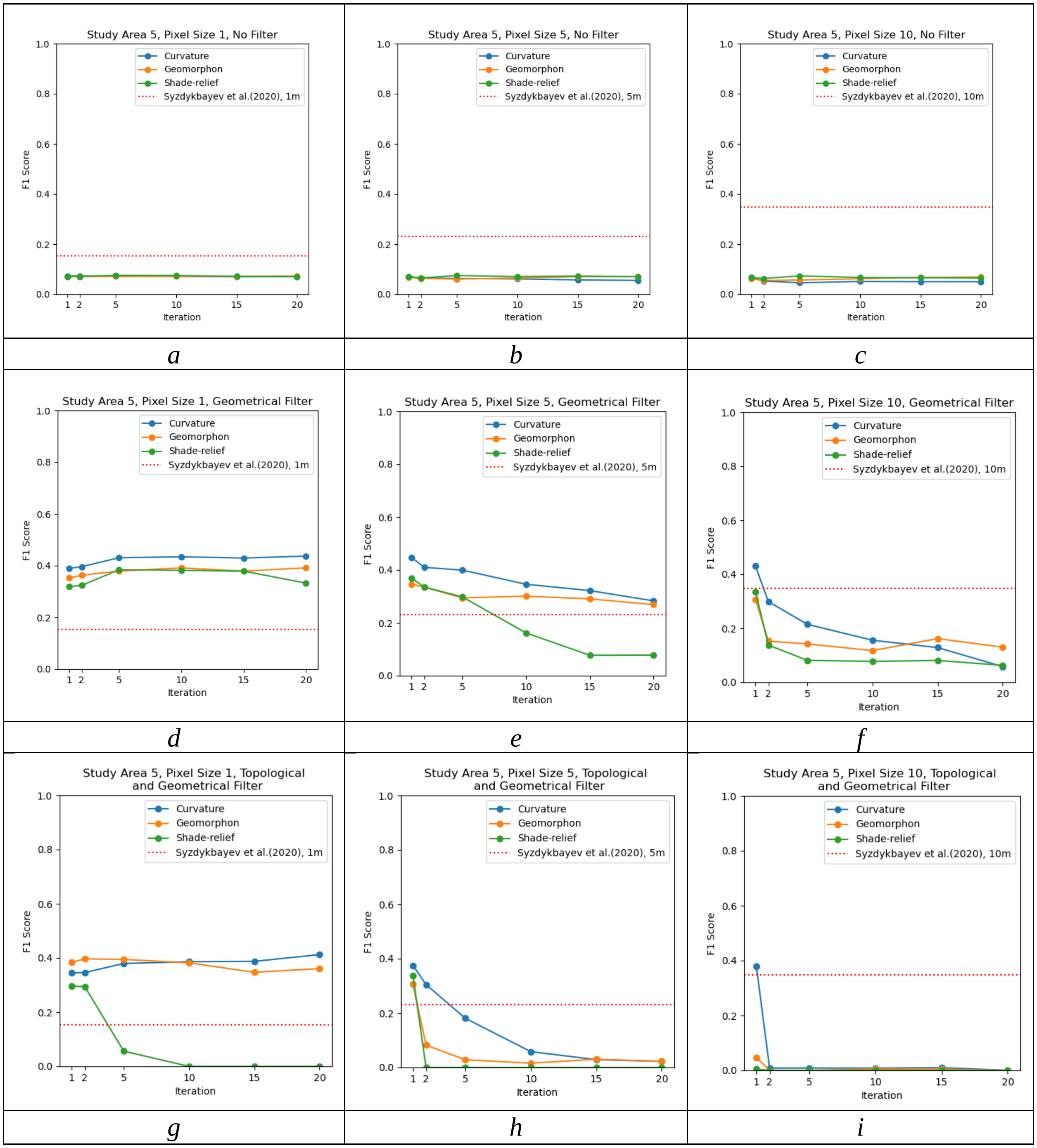
Table 6.8 List of geometrical and contextual rules used to obtain the highest F1 score; Study Area 4: Oregon

Information	Parameter	Rules	Rules Used
Context	Slope (degrees: min:0 and max: 90)	12 - 72	0 - 72
	NDVI (pixel value: min: -1 and max: 1)	0.12 - 0.75	0 - 0.75
	Surface roughness (index value min: 0 and higher)	0.12 - 2	0.12 - ∞
Geometrical	Length/width (unit)	0.27 - 3	0 - 3
	Area (meters)	18- 314297870	0- 314297870

Findings for Study Area 5:

- *Experiment 1:* Optimal F1 scores of 0.07 were achieved with pixel sizes of 10 using Shade-relief. In comparison with Syzdykbayev et al., (2020c), all results were inferior regardless of pixel size or smoothing iterations (Figure 6.8 (a, b, c)).
- *Experiment 2:* The best F1 scores, reaching 0.45, were observed at pixel size 5, employing Curvature (Figure 6.8 (e)). When compared with Syzdykbayev et al., (2020c), the inclusion of geometrical and contextual filters resulted in enhanced outcomes across all pixel sizes and smoothing iterations (Figure 6.8 (d, e, f)).
- *Experiment 3:* Peak F1 scores of 0.41 were observed at pixel size 1 with Curvature (Figure 6.8 (g)). Compared with Syzdykbayev et al., (2020c), the combined filtering approach led to improved results across all pixel sizes and smoothing iterations (Figure 6.8 (g, h, i)).

In summary, the most favorable results for Study Area 5 were obtained when only geometrical and contextual filters were applied to pixel sizes of 5 and with Curvature (Figure 6.8 (e)). From the original set of 10 rules (comprising 4 geometrical and 6 contextual rules), a subset of 4 rules (consisting of 1 geometrical and 3 contextual rules) was chosen to attain these results (Table 6.9).



a *b* *c*

d *e* *f*

g *h* *i*

Figure 6.8 Results of the three LTFs extraction algorithms with different pixel sizes, smoothing iterations and three evaluations as a measure of F1 score; Study Area 5: Washington

Table 6.9 List of geometrical and contextual rules used to obtain the highest F1 score; Study Area 5: Washington

Information	Parameter	Rules	Rules Used
Context	Slope (degrees: min:0 and max: 90)	12 - 72	12 - 72
	NDVI (pixel value: min: -1 and max: 1)	0.12 - 0.75	0 - 0.75
	Surface roughness (index value min: 0 and higher)	0.12 - 2	0 - ∞
Geometrical	Length/width (unit)	0.27 - 3	0 - ∞
	Area (meters)	137- 5689077	0-5689077

6.4 Discussion

Our analysis discerns a notable trend relating to pixel size and number of smoothing iterations: as the pixel size increases from 1 to 10 and the smoothing iterations are augmented, there is a corresponding decrease in F1 score. This observation can be intuitively understood when considering the impact of increased pixel size and smoothing iterations on the quality and detail of the landslide boundary information. Increasing pixel size and number of smoothing iterations is analogous to reducing the resolution of the image or applying a strong blurring effect. In the context of geospatial object detection, this can be likened to examining a landscape from a greater distance or through a foggy lens. While larger pixel sizes and more smoothing iterations simplify the image and can aid in the detection of large-scale patterns or structures, they also hide the finer details that are often critical for accurate object detection. In the case of landslide detection, the intricate details of landslide boundaries, which often contain key indicators of landslide's characteristics and potentially its causes, are lost when viewed at lower resolutions. Consequently, while larger pixel sizes and increased smoothing iterations might expedite the processing and analysis of geospatial data, our results highlight the inherent trade-off between the simplification of data and the preservation of crucial details. Our findings underscore the importance of carefully selecting pixel size and number of smoothing iterations in geospatial object detection tasks. Striking a balance between data simplification for efficiency and preservation of detail for accuracy is key to optimizing the results of such analyses.

Analyzing the results of the three sets of experiments conducted in each of the five study areas, we observe that Experiment 2, using filters based on only geometrical and contextual information, shows better results in comparison with Experiment 1, not using any filters. The reason for this performance improvement becomes clear when we examine the limitations of the first experiment, which inherently carries a high risk of generating FP. Without applying filters, all polygons possessing a circular shape are identified as detected geospatial objects, leading to an overestimation of the actual object count. Experiment 2's results are also better than Experiment 3's results. The exceptions are a few cases where the outcomes from Experiments 2 and 3 are close (as shown in Figure 6.6 (d, g)). The underlying factor here is that Experiment 3 incorporates topological filters, i.e., birth and lifespan of circles, which correlate with the size of the circle, a characteristic already accounted for in the geometrical selection phase of Experiment 2.

With these results it can be concluded that using geometrical and contextual filters applied on candidate polygons yields the highest accuracy. Using filters with combined topological geometrical and contextual information did not improve accuracy, but rather increased the complexity of the detection rules, potentially obstructing object detection. Importantly, it is worth mentioning that the candidate polygons were derived using topological properties. In essence, topological information from the raw input played a crucial role in deriving these candidate polygons.

Our comparative analysis with prior studies yields promising outcomes. When compared with the knowledge-based method utilized in the work by Bunn et al. (2019), our method shows superior results in both Study Areas 2 and 4. Similarly, when these results are compared against the results from the work by Syzdykbayev et al., (2020c), which exclusively employed filters based only on topological information on candidate polygons, we observe an improvement in all study areas with one exception, Study Area 4. In Study Area 4, our results are closely aligned with the results of the work by Syzdykbayev et al., (2020c). The slight discrepancy underscores the inherent complexity of geospatial analysis and attests to the necessity of employing diverse, complementary strategies in different contexts to optimize outcomes.

Table 6.10 Results of Topological KB method, Syzdykbayev et al., (2020c), and Bunn et al. (2019) in each study area

	Study Area 1	**Study Area 2**	**Study Area 3**	**Study Area 4**	**Study Area 5**
Title	**Chapter 6: Geospatial Object Detection: Topological KB Method**				
F1 score	0.52	0.48	0.43	0.64	0.45
Title	**Chapter 4: Geospatial Object Detection: PH-Based Method (Syzdykbayev et al., 2020b)**.				
F1 score	0.334	0.466	0.382	0.65	0.337
Title	**Bunn et al. (2019)**				
F1 score		0.47		0.53	

7.0 Geospatial Object Detection: Topological ML-Based Method

Including additional spectral channels beyond the conventional Red, Blue, and Green (RGB) is an emerging and prevalent practice in ML models, particularly in geospatial object detection tasks utilizing remote sensing datasets. Using diverse spectral bands provides a richer, more detailed representation of the Earth's surface, enhancing the discriminatory power of algorithms in identifying specific features or objects. With multispectral and hyperspectral remote sensing, the addition of channels like Near-Infrared (NIR), Shortwave Infrared (SWIR), and others, helps in capturing subtle differences that are indistinguishable in the traditional RGB space (Wieland et al., 2023). The heterogeneity of spectral reflectance across various land cover types, vegetation health statuses, water bodies, or built-up areas, can be effectively captured with a broader spectral range, leading to improved detection accuracy.

The fusion of Digital Elevation Models (DEM), slope, other derivatives, and landform boundaries, can create a useful combination for enhancing the precision and accuracy of ML models in geospatial object detection tasks (Wieland et al., 2023). DEM and slope provide critical topographical insights, enabling models to differentiate between varying landforms and land uses based on terrain height and gradient information. Other derivatives, such as aspect, curvature, or ruggedness, supplement these data by offering further details on surface characteristics, assisting in the interpretation of complex landform processes and patterns (Wieland et al., 2023). Implementing edge detection, particularly using a Laplacian filter, can introduce another level of precision in geospatial object detection (Tchinda et al., 2021). The Laplacian filter is a second-order derivative operator known for its proficiency in highlighting areas of rapid intensity change in an image, which typically correspond to edges or boundaries (Szeliski, 2010b). These can be leveraged to delineate distinct ecological zones, transition areas, or human-made features, providing valuable contextual information to help ML algorithms in feature extraction and object identification (Xiao et al., 2021).

By integrating these edges or boundaries derived from Laplacian filter onto the existing spectral and topographic data, we introduce another dimension to our model. This enhancement allows the model to account for sharp changes or gradients that are typical of boundary areas (Wieland et al., 2023). This integration can lead to a richer, more nuanced understanding of the geospatial landscape, thereby enhancing the capability of ML models.

In this research task, we incorporate PH and Mapper, as an additional channel into existing multichannel layer, as both are expected to enhance the detection and representation of geospatial boundaries derived from the outputs of Laplacian filter. PH is proficient at highlighting boundaries that exhibit circular patterns, thereby capturing geographic transitions that might be overlooked by traditional edge detection methods. Mapper provides a graph-based visualization of area boundaries, offering a detailed and intuitive understanding of the geographical context, and unveiling the intricacies of the terrain. Both PH and Mapper rely on area boundaries derived from Laplacian filter on RGB image, offering unique capabilities to interpret these boundaries with greater precision and context.

To address RQ3 (see Chapter 3), we present a Topological ML-based geospatial object detection method focused on the shape and location of geospatial objects in geospatial datasets.

7.1 Methodology

The proposed method transforms topological information, derived by using PH and Mapper, into a three-dimensional array, i.e., a multichannel image while preserving its location by a subset of points. The multichannel image can be represented in a three-dimensional array and combined directly with ML models with a CNN architecture. In other words, multichannel images with topological information are used as feature inputs to ML-based models to detect geospatial objects. The workflow of the method is shown in Figure 7.1. The first step of the workflow involves extracting topological information by using PH and Mapper. The second step involves implementing algorithms that transform the extracted topological information into a multichannel image. The third step involves implementing Topological ML to detect geospatial objects where the input feature is a multichannel image derived from the second step.

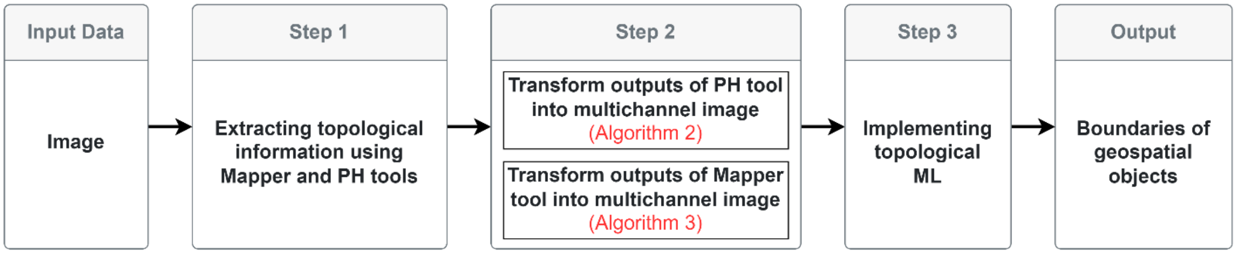

Input Data
Image
Step 1
Extracting topological information using Mapper and PH tools
Step 2
Transform outputs of PH tool into multichannel image (Algorithm 2)
Transform outputs of Mapper tool into multichannel image (Algorithm 3)
Step 3
Implementing topological ML
Output
Boundaries of geospatial objects

Figure 7.1 Workflow of the Topological ML-based geospatial object detection method

7.1.1 Extracting Topological Information from PH and Mapper

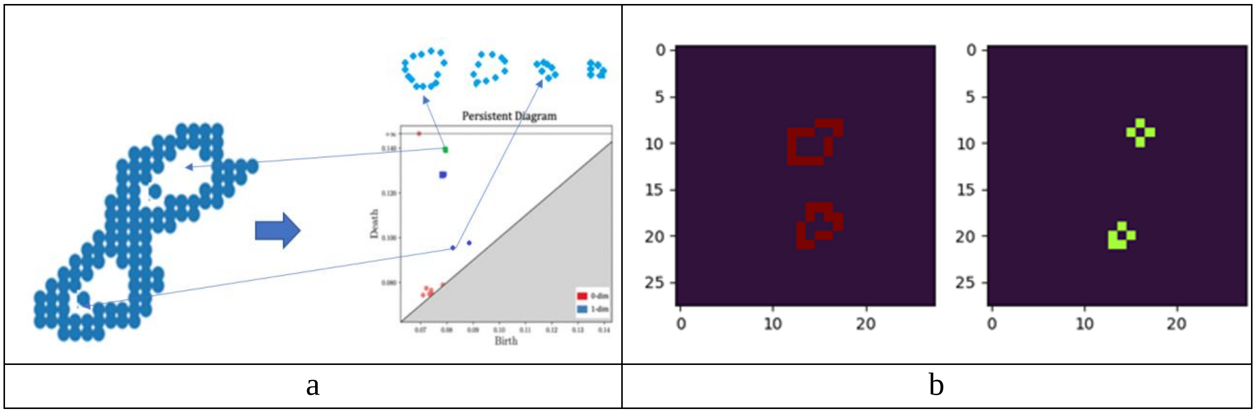
There are many ways to convert an image into a set of points such as using threshold values or extracting linear features. With threshold values, a binary image is created where nonzero values are converted into a set of points (Tymochko, Munch, Dunion, Corbosiero, & Torn, 2020) and to be able to focus on boundaries of geospatial objects, an LTFs extraction algorithm is needed. If the input data is a DTM, an LTFs extraction algorithm is implemented (Syzdykbayev et al., 2020a). If the input data is an RGB image, or a multichannel satellite image, an edge detection algorithm, commonly used in CV, such as Gauss (Laplacian) filters (Szeliski, 2010b), is implemented.

Before using a set of points in PH and Mapper as input, the coordinates of the set of points, which are in geographic coordinate system, must be transformed into a cartesian coordinate system where the origin is the upper left corner of the input image. This transformation allows us to keep track of the location of topological information and to overlay points onto images with corresponding coordinates. Next, PH and Mapper are applied to the set of points to derive topological information. For PH, the location of the subsets of points are derived and stored in addition to the topological information. For Mapper, the location of the nodes and the number of points inside each cluster in addition to the general information about the graph is derived and stored.

7.1.2 Transforming Topological Information into a Multichannel Image

Both ML-based object detection and ML-based instance segmentation methods achieve good performance in detecting geospatial objects. However, both are based on CNN architecture and require a multichannel image as an input (Cheng & Han, 2016; Hafiz & Bhat, 2020). We transformed topological information derived by using PH and Mapper into a multichannel image while preserving the location of the subset of points that form topological information. Such a transformation requires the shape of the multichannel image identical to the size of the input image. Since the outputs of PH and Mapper are different, we implemented two algorithms, one to transform the output of PH into a multichannel image (Algorithm 1), and one to transform the output of Mapper into a multichannel image (Algorithm 2).

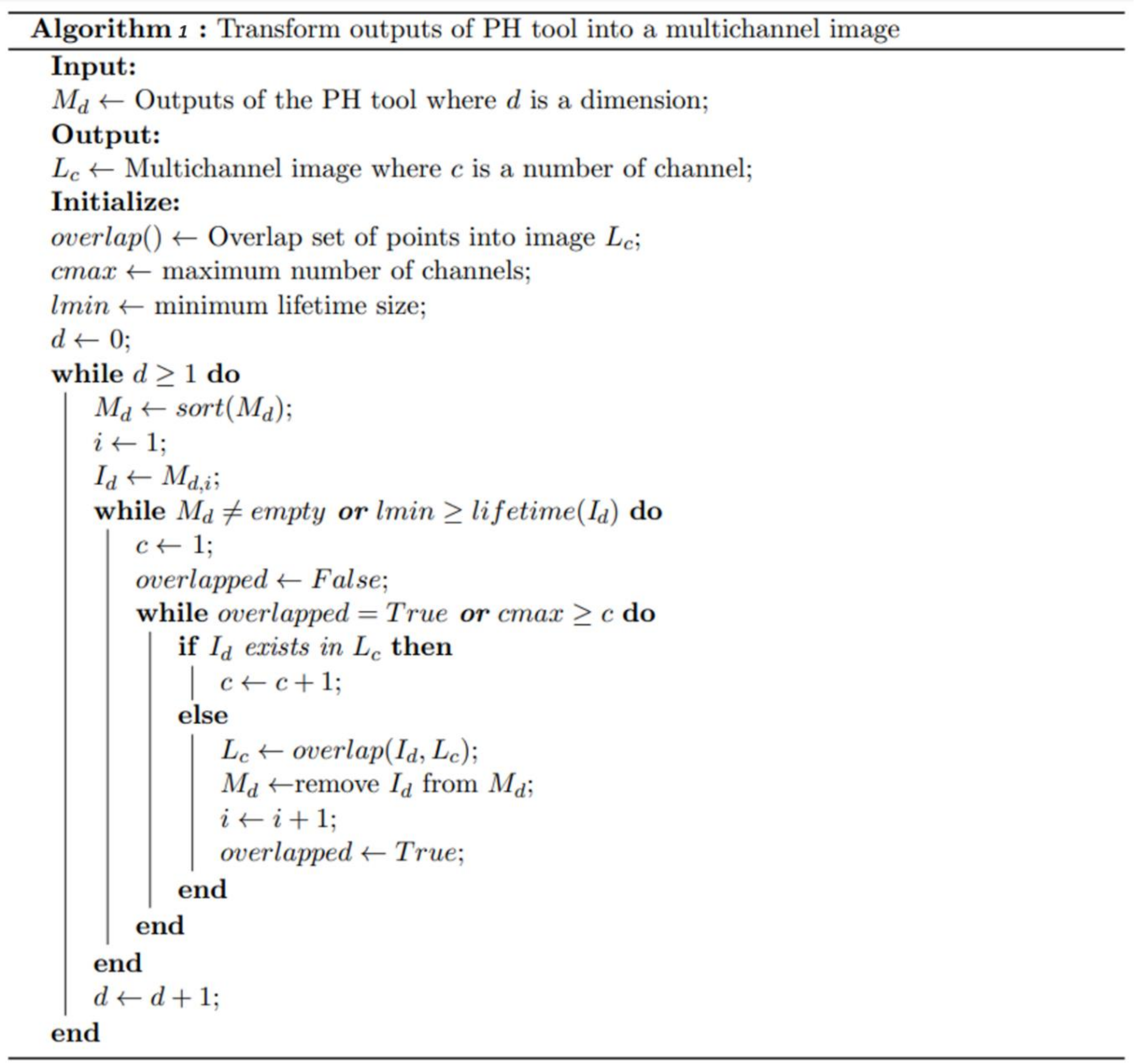
Algorithm 1 transforms the output of PH, specifically connected components and circles, into a multichannel image (Figure 7.2). The general logic of Algorithm 1 is to keep track of the subset of points along with topological information. First, an empty multichannel image, and two threshold values are initialized, one for maximum number of channels in the multichannel image, and the other for minimum lifetime value. The size of each channel of the empty image is equal to the size of the input image, and the number of channels is equal to the first threshold value. The output of PH is sorted as lifetime value. Each point in the subset of points (dimension, birth time, death time, lifetime) outputted by PH (shown in Figure 7.2 a) is overlaid onto the first channel of the empty image with corresponding coordinates (Figure 7.2 b). If the points in a subset of points intersect, each point in the subset is overlaid onto the next channel. The set of pixel values of an empty image that are overlaid with the subset of points represent topological information such as birth time or lifetime.



a

b

Figure 7.2 Algorithm 1: (a) set of points with four circles and PD with subset of points that form topological information (b) multichannel image with topological information



Algorithm 1 : Transform outputs of PH tool into a multichannel image

```
Input:
$M_d \leftarrow$ Outputs of the PH tool where $d$ is a dimension;
Output:
$L_c \leftarrow$ Multichannel image where $c$ is a number of channel;
Initialize:
$overlap() \leftarrow$ Overlap set of points into image $L_c$;
$cmax \leftarrow$ maximum number of channels;
$lmin \leftarrow$ minimum lifetime size;
$d \leftarrow 0$;
while $d \geq 1$ do
    $M_d \leftarrow sort(M_d)$;
    $i \leftarrow 1$;
    $I_d \leftarrow M_{d,i}$;
    while $M_d \neq empty$ or $lmin \geq lifetime(I_d)$ do
        $c \leftarrow 1$;
        $overlapped \leftarrow False$;
        while $overlapped = True$ or $cmax \geq c$ do
            if $I_d$ exists in $L_c$ then
                $c \leftarrow c + 1$;
            else
                $L_c \leftarrow overlap(I_d, L_c)$;
                $M_d \leftarrow$remove $I_d$ from $M_d$;
                $i \leftarrow i + 1$;
                $overlapped \leftarrow True$;
            end
        end
    end
    $d \leftarrow d + 1$;
end
```

The output of Algorithm 2 is a graph with no overlapped nodes or edges (Figure 7.3). First, an empty multichannel image is initialized, where the shape of each channel is equal to the shape of the input image. Next, each node and each edge of the graph is overlaid onto an empty image with corresponding coordinates (Figure 7.3 b). The set of pixel values of an empty image that are overlaid with nodes or edges represent information such as degree values of nodes or lengths of edges (Figure 7.3 b).

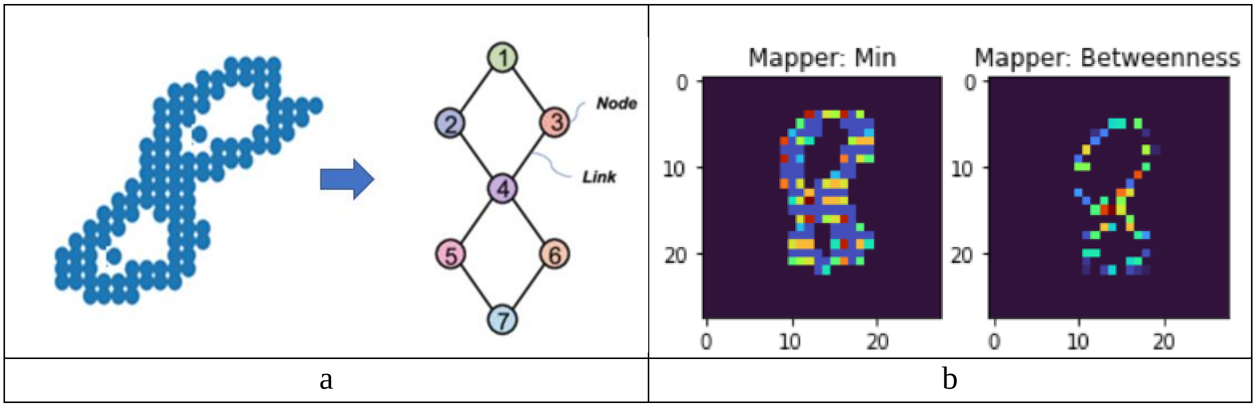
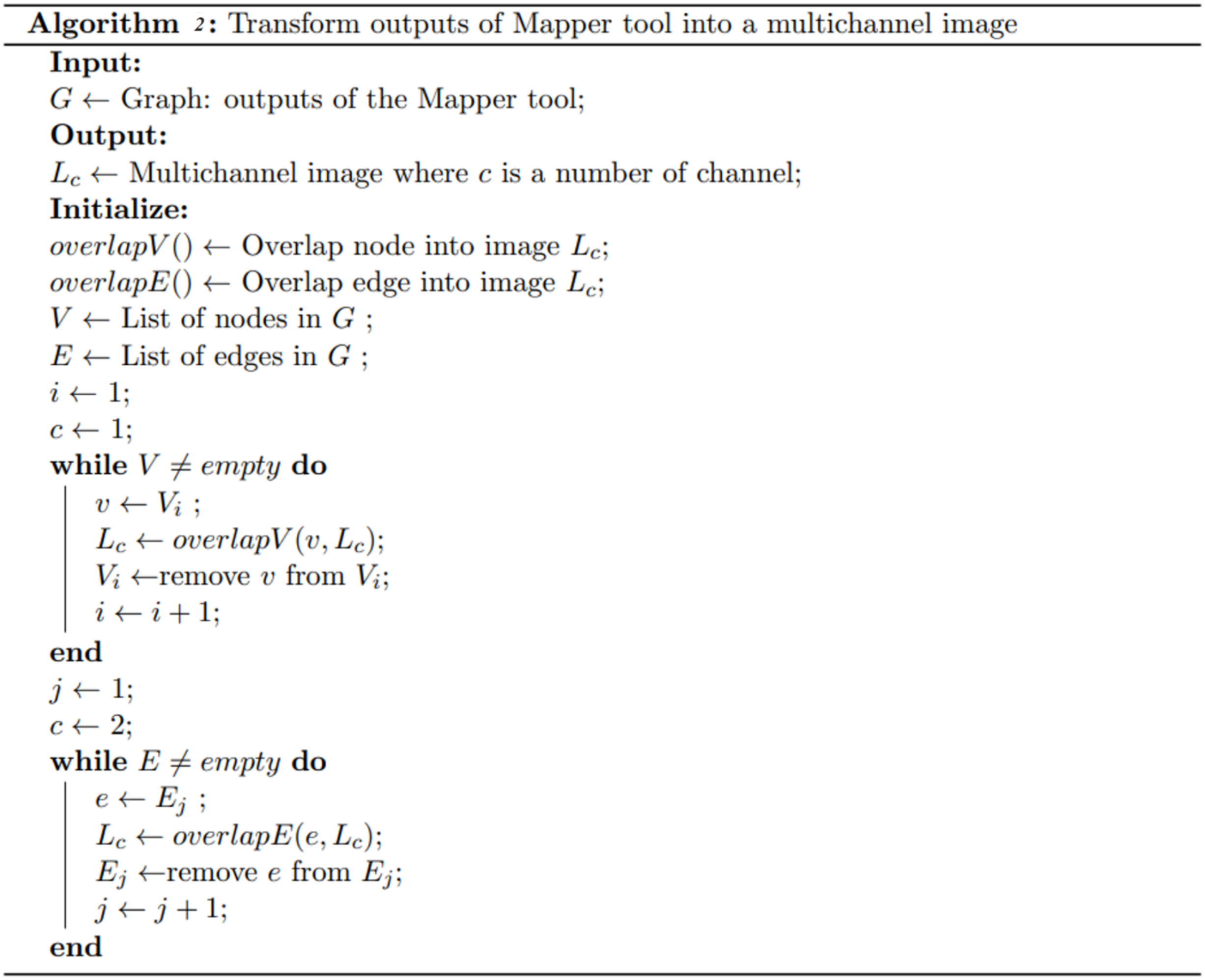


Figure 7.3 : Algorithm 2: (a) set of points with four circles and a graph that was created using Mapper (b) multichannel image with topological information



```
Algorithm 2: Transform outputs of Mapper tool into a multichannel image
    Input:
    $G \leftarrow$ Graph: outputs of the Mapper tool;
    Output:
    $L_c \leftarrow$ Multichannel image where $c$ is a number of channel;
    Initialize:
    $overlapV() \leftarrow$ Overlap node into image $L_c$;
    $overlapE() \leftarrow$ Overlap edge into image $L_c$;
    $V \leftarrow$ List of nodes in $G$ ;
    $E \leftarrow$ List of edges in $G$ ;
    $i \leftarrow 1$;
    $c \leftarrow 1$;
    while $V \neq empty$ do
        $v \leftarrow V_i$ ;
        $L_c \leftarrow overlapV(v, L_c)$;
        $V_i \leftarrow$remove $v$ from $V_i$;
        $i \leftarrow i + 1$;
    end
    $j \leftarrow 1$;
    $c \leftarrow 2$;
    while $E \neq empty$ do
        $e \leftarrow E_j$ ;
        $L_c \leftarrow overlapE(e, L_c)$;
        $E_j \leftarrow$remove $e$ from $E_j$;
        $j \leftarrow j + 1$;
    end
```

7.2 Datasets

To address RQ3, the derived multichannel image was used as a feature input to the ML-based instance segmentation method. We used a geospatial dataset from the LandSlide4Sense competition provided by the Institute of Advanced Research in Artificial Intelligence (LandSlide4Sense, n.d.) as input.

The Landslide4Sense dataset is derived from a diverse set of landslide-affected areas around the world from 2015 through 2021(LandSlide4Sense, n.d.). The dataset consists of training, validation, and test sets containing around 3799 image patches. These image patches consist of input images and labels. All bands in the dataset are resized to the resolution of ~10m per pixel. The labels are masked with landslide boundaries where each label has 128 x 128 pixels. The input images are multi-source satellite imagery with 14 channels that include (see Figure 7.4):

- Multispectral data from Sentinel-2 (B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11, B12) (see Table 7.1)
- Slope and DEM data from ALOS PALSAR

Table 7.1 Sentinel-2 Spectral Bands: Channel Specifications and Descriptions

Channels	Pixel size	Wavelength	Description
B1	60 m	443 nm	Ultra Blue (Coastal and Aerosol)
B2	10 m	490 nm	Blue
B3	10 m	560 nm	Green
B4	10 m	665 nm	Red
B5	20 m	705 nm	Visible and Near Infrared (VNIR)
B6	20 m	740 nm	Visible and Near Infrared (VNIR)
B7	20 m	783 nm	Visible and Near Infrared (VNIR)
B8	10 m	842 nm	Visible and Near Infrared (VNIR)
B8a	20 m	865 nm	Visible and Near Infrared (VNIR)
B9	60 m	940 nm	Short Wave Infrared (SWIR)
B10	60 m	1375 nm	Short Wave Infrared (SWIR)
B11	20 m	1610 nm	Short Wave Infrared (SWIR)
B12	20 m	2190 nm	Short Wave Infrared (SWIR)

The dataset was divided into training (2849) and testing (950) sets where training set was used to train segmentation model and testing set was used to evaluate model’s performance. In order to train the model, PyTorch, an open-source ML framework (Paszke et al., 2019) was used.

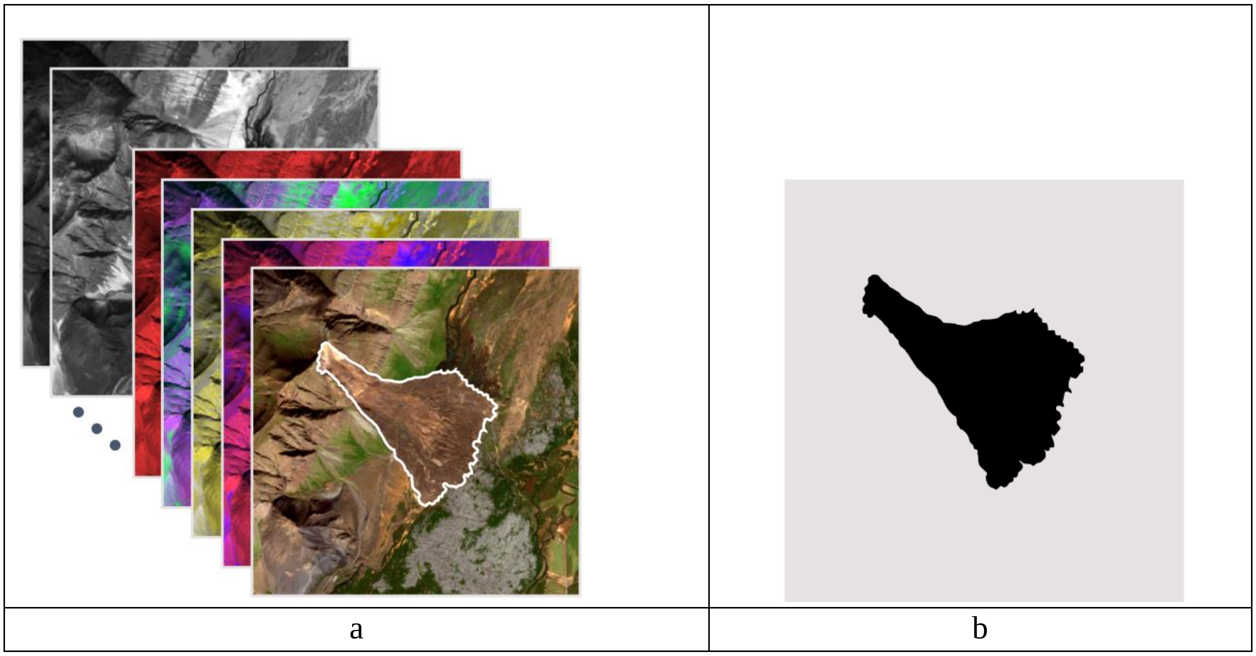


Figure 7.4 (a) multichannel satellite imagery and (b) label pair (mask) of the geospatial dataset from the LandSlide4Sense competition (LandSlide4Sense, n.d.)

7.3 Experiment

We experimented with training and testing of various ML models under diverse parameters and differing numbers of input channels (see Table 7.2). To ensure comprehensive model learning and generalization, we trained each model iteration for a total of 300 epochs. In the ML-based segmentation model, we utilized UNET as the encoder, VGG11 as the decoder, and Jaccard Loss as the loss function. The models were not pretrained.

Table 7.2 Experimental setups and configurations for image encoding and decoding

Encoder Name	Decoder Name	Loss function	Number of Epochs	Channel Name	Number of Channels
UNET	**VGG11**	Jaccard Loss	300	RGB	3
UNET	**VGG11**	Jaccard Loss	300	RGB, Slope, DEM	5
UNET	**VGG11**	Jaccard Loss	300	ALL Channels	14
UNET	**VGG11**	Jaccard Loss	300	RGB, + Edge	3
UNET	**VGG11**	Jaccard Loss	300	RGB, Slope, DEM+ Edge	6
UNET	**VGG11**	Jaccard Loss	300	ALL Channels +Edge	15
UNET	**VGG11**	Jaccard Loss	300	RGB + PH	7
UNET	**VGG11**	Jaccard Loss	300	RGB, Slope, DEM + PH	9
UNET	**VGG11**	Jaccard Loss	300	ALL Channels+ PH	18
UNET	**VGG11**	Jaccard Loss	300	RGB + Mapper	7
UNET	**VGG11**	Jaccard Loss	300	RGB, Slope, DEM + Mapper	9
UNET	**VGG11**	Jaccard Loss	300	ALL Channels + Mapper	18
UNET	**VGG11**	Jaccard Loss	300	RGB + PH + Mapper	11
UNET	**VGG11**	Jaccard Loss	300	RGB, Slope, DEM+ PH + Mapper	13
UNET	**VGG11**	Jaccard Loss	300	ALL Channels + PH + Mapper	22
UNET	**VGG11**	Jaccard Loss	300	ALL Channels + PH + Mapper + Edge	23

Our methodology specifically leverages three types of information derived from the source data, divided into distinct subsets. The first subset solely consists of RGB (Red, Blue, and Green) channels, providing a baseline performance with just the primary color channels. The second subset extends this to include topographic features, namely, slope and DEM data, in addition to RGB channels, as depicted in Figure 7.5. Lastly, we used a more comprehensive approach by utilizing all available channels from the source data as model inputs. It is important to note that these scenarios do not involve any embedding of additional information into the source data. They provide a comprehensive understanding of how inclusion of different data subsets as inputs affects the performance of ML models in geospatial object detection tasks. In this first part of the experiment, rather than augmenting the source data, we focused on experimenting with various subsets of it. The outcomes of these experiments provide insights into both the advantages and potential limitations of incorporating increasingly complex input data into the models.

The second part of the experiment is anchored on the above three subsets of multichannel images. We performed several additions into each subset of multichannel image by incorporating diverse layers of information into the source data. These layers, sourced from Laplacian filter, PH, and Mapper, are described as follows:

- Laplacian Filter is a single-channel addition emphasizing area boundaries. The effects of this layer and its contribution are visualized in Figure 6.5.
- PH contains a four-channel layer. The primary function of this layer is to accentuate areas marked by circular boundaries, revealing underlying topological nuances.
- Mapper contains a four-channel layer. This layer provides a graph-based perspective of the area boundaries, offering a unique analytical angle.
- Combining the outputs of PH and Mapper results in an eight-channel layer.

Overall, we executed 16 experiments (Table 7.2), each adding complexity and depth layers to the geospatial object detection task. The results of these experiments provide valuable insights into the impact of these layered augmentations on the performance of the ML models, and how they contribute to the efficacy and accuracy of geospatial object detection.

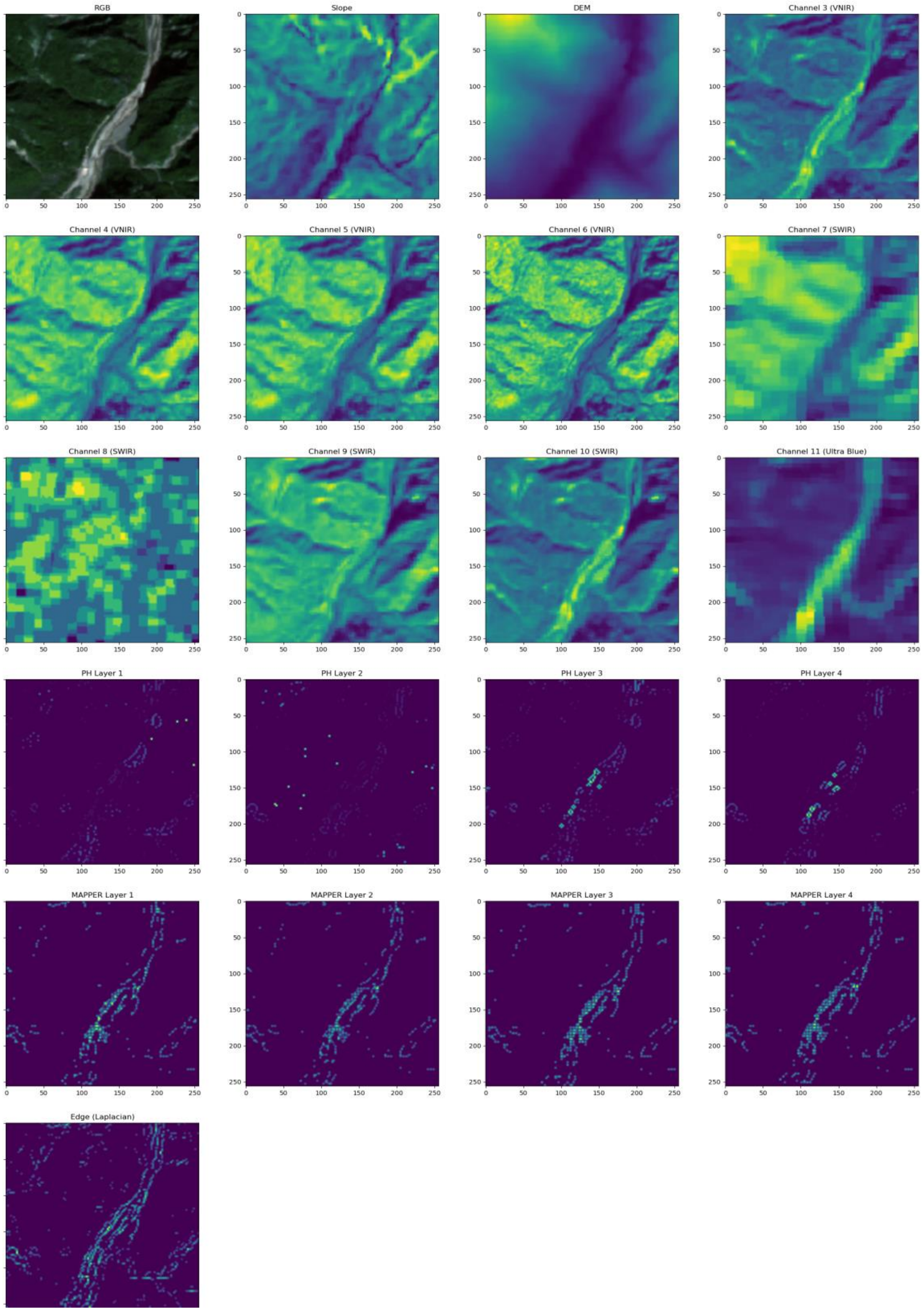


Figure 7.5 RBG, slope and DEM subsets of the source data

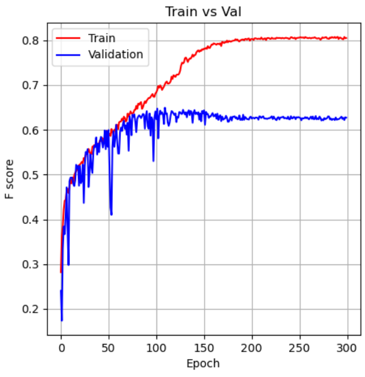
The performances of predicted images against ground truth provided by LandSlide4Sense competition were evaluated by using four common validation metrics: accuracy, precision, recall, and F1 score. In addition, F1 score was used to find a winner in the LandSlide4Sense competition. Hence, by using F1 score, we can compare results of the Topological ML-based method with the winners of the LandSlide4Sense competition. We only show the results of F1 score and the results of the other validation metrics are available in Appendix C.

7.4 Results

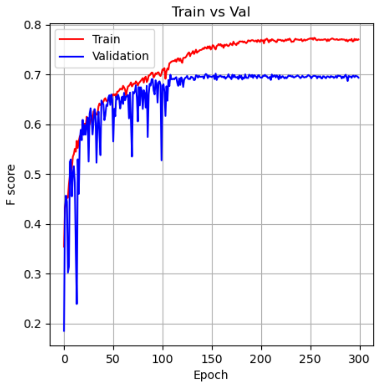
Our experimental design aimed to provide a comprehensive understanding of how the inclusion of different data subsets, specifically adding layers with topological information, as inputs affects the performance of ML models in geospatial object detection tasks. We conducted 16 distinct sets of evaluations, and the results are shown in Table 7.3 and in Figure 7.6.

Table 7.3 Experimental configurations and F1 scores for encoder-decoder models after 300 Epochs

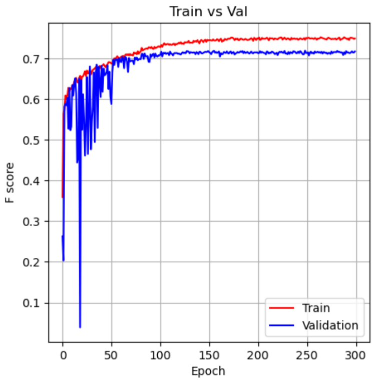
Encoder Name	Decoder Name	Loss function	Channel Name	Number of Channels	F1 score of Training set after 300 epochs	F1 score of Test set after 300 epochs
UNET	VGG11	Jaccard Loss	RGB	3	0.805	0.627
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM	5	0.771	0.694
UNET	VGG11	Jaccard Loss	ALL Channels	14	0.749	0.717
UNET	VGG11	Jaccard Loss	RGB, + Edge	4	0.788	0.628
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM+ Edge	6	0.792	0.678
UNET	VGG11	Jaccard Loss	ALL Channels +Edge	15	0.792	0.717
UNET	VGG11	Jaccard Loss	RGB + PH	7	0.779	0.623
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM + PH	9	0.804	0.666
UNET	VGG11	Jaccard Loss	ALL Channels+ PH	18	0.767	0.717
UNET	VGG11	Jaccard Loss	RGB + Mapper	7	0.813	0.601
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM + Mapper	9	0.774	0.684
UNET	VGG11	Jaccard Loss	ALL Channels + Mapper	18	0.790	0.717
UNET	VGG11	Jaccard Loss	RGB + PH + Mapper	11	0.812	0.604
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM+ PH + Mapper	13	0.807	0.662
UNET	VGG11	Jaccard Loss	ALL Channels + PH + Mapper	22	0.794	0.721
UNET	VGG11	Jaccard Loss	ALL Channels + PH + Mapper + Edge	23	0.775	0.716



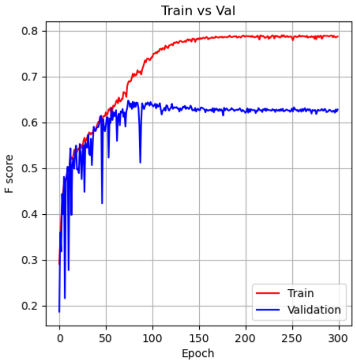
(a) RGB



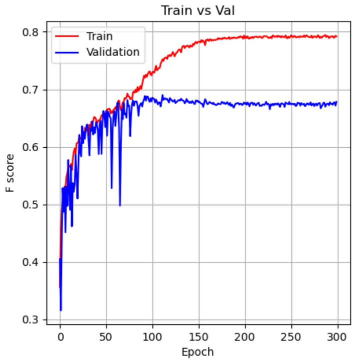
(b) RGB, Slope, DEM



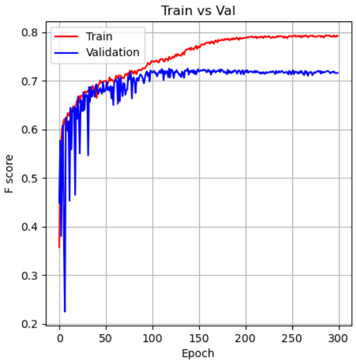
(c) ALL Channels



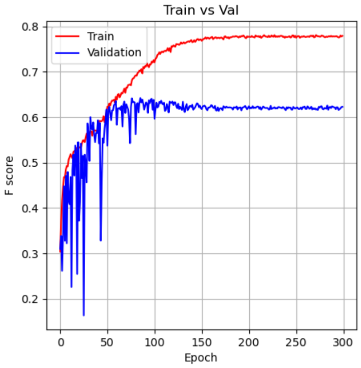
(d) RGB, + Edge



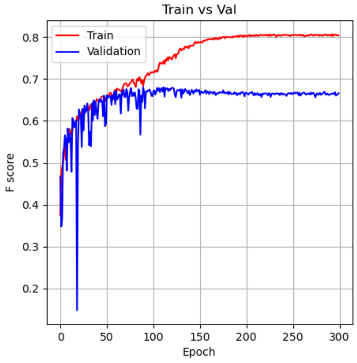
(e) RGB, Slope, DEM+ Edge



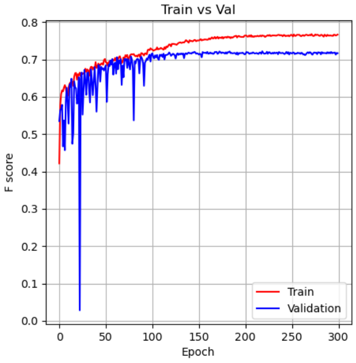
(f) ALL Channels +Edge



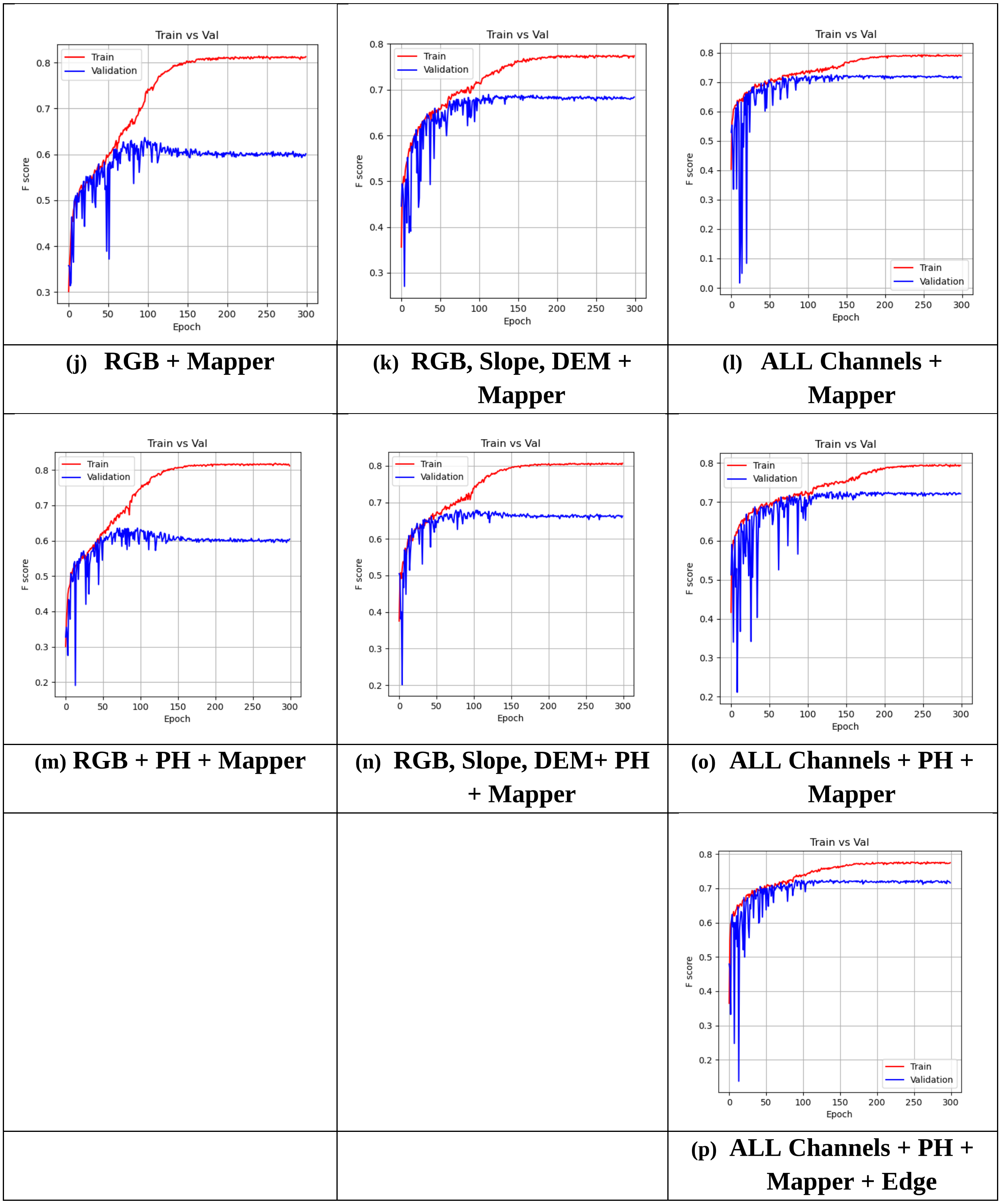
(g) RGB + PH



(h) RGB, Slope, DEM + PH



(i) ALL Channels+ PH



(j) RGB + Mapper

(k) RGB, Slope, DEM + Mapper

(l) ALL Channels + Mapper

(m) RGB + PH + Mapper

(n) RGB, Slope, DEM+ PH + Mapper

(o) ALL Channels + PH + Mapper

(p) ALL Channels + PH + Mapper + Edge

Figure 7.6 Graph with F1 score of training and testing results for each epoch

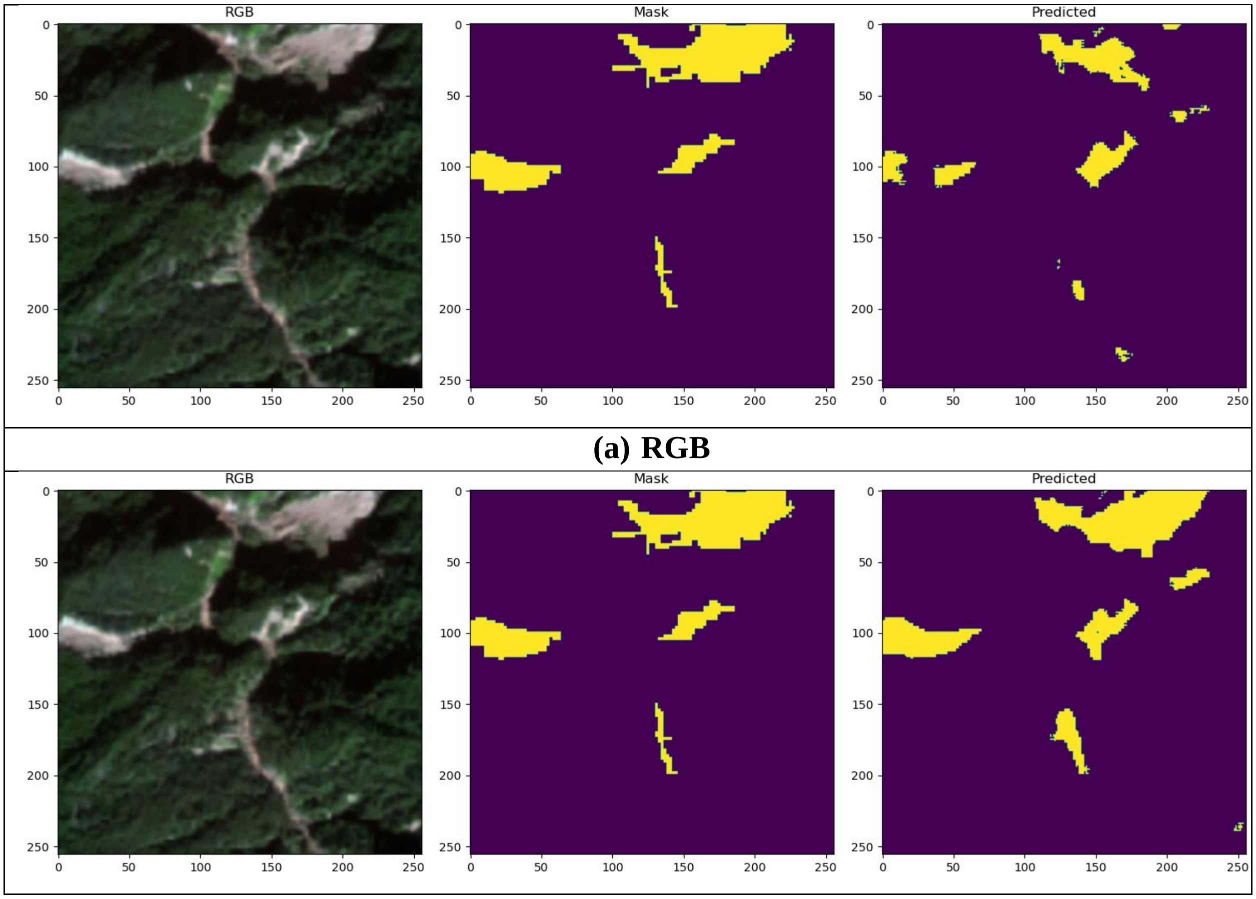
7.5 Discussion

From the analysis presented in Figure 7.6, two notable trends can be observed in the testing results:

- Horizontal Trend: Across all cases, there is a consistent increase in F1 score from left to right on the testing set. This upward trend signifies that the inclusion of additional channels like DEM, slope, NIR, and SNIR effectively enhances the detection accuracy of geospatial objects.
- Vertical Trend: When observing the results from top to bottom, F1 scores remain unchanged. This stability indicates that while the addition of channels such as DEM, slope, NIR, and SNIR improves accuracy, the incorporation of extra layers like Laplacian filter and those with topological information does not provide a corresponding increase in detection accuracy.

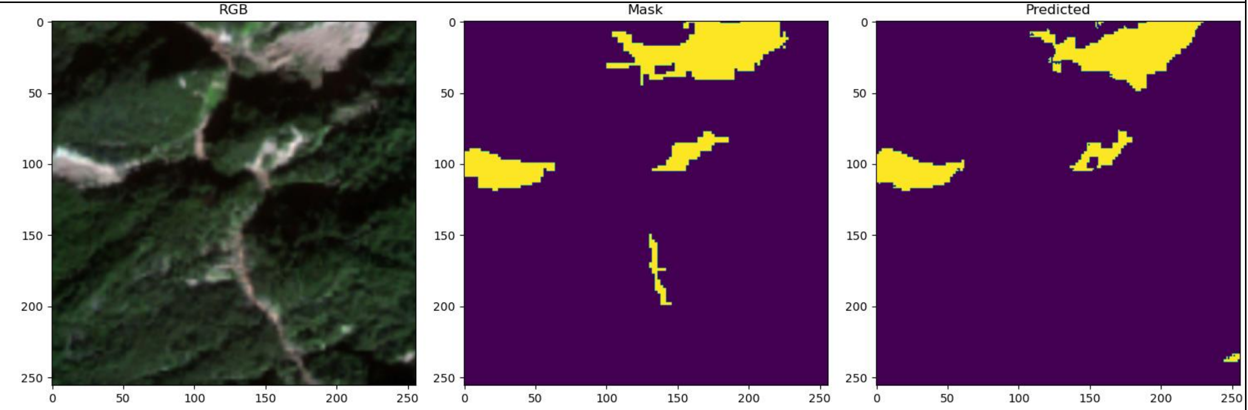
Further insights into the effectiveness of topological layers were gained through an in-depth analysis of a selected image from the testing set, processed after 300 training epochs and presented in Figure 7.7 (a, d, g, j). In these images, the modeling was performed using only RGB data, with and without additional layers such as Laplacian filter and topological layers, and without including features such as DEM, slope, NIR, or SNIR. Upon careful examination of these images, distinctive patterns emerged, illuminating the nuanced impact of incorporating topological layers into the model. The contrast between the images with and without these topological enhancements offers a tangible demonstration of how specific layers interact with the underlying RGB data, casting new light on the potential value of topological information in the detection and characterization of geospatial objects.

With topological layers, the positional results were able to accentuate the exact location of loops present in the RGB image, even if it was generalized in the mask (Figure 7.7 g). Comparatively, in the absence of the topological layer, the model failed to highlight the loops and instead, merely attempted to conform to the mask results (Figure 7.7 a). This observation emphasizes that the inclusion of topological layers contributes to a more precise representation of detected objects' shapes, particularly when those objects possess an internal void or hole. The analysis illustrates the intricate relationship between different layers and channels, emphasizing the complexity of geospatial object detection. While certain additional channels prove beneficial, the use of topological layers, despite their capacity to refine object shapes, does not necessarily translate to improved overall detection accuracy.

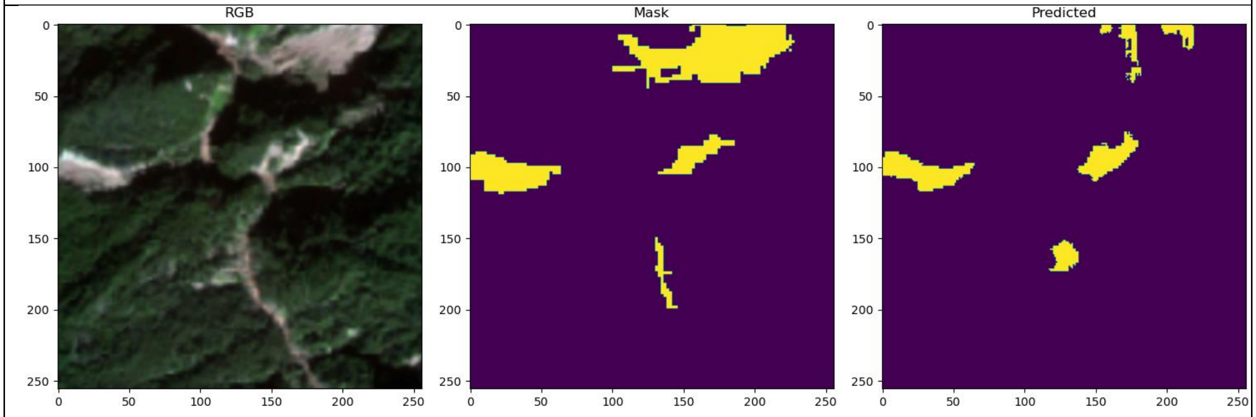


(a) RGB

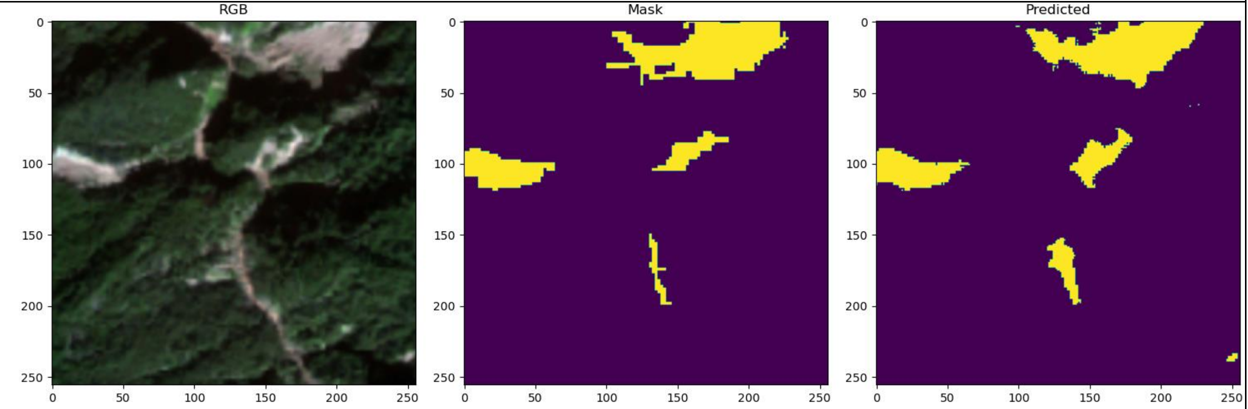
(b) RGB, Slope, DEM



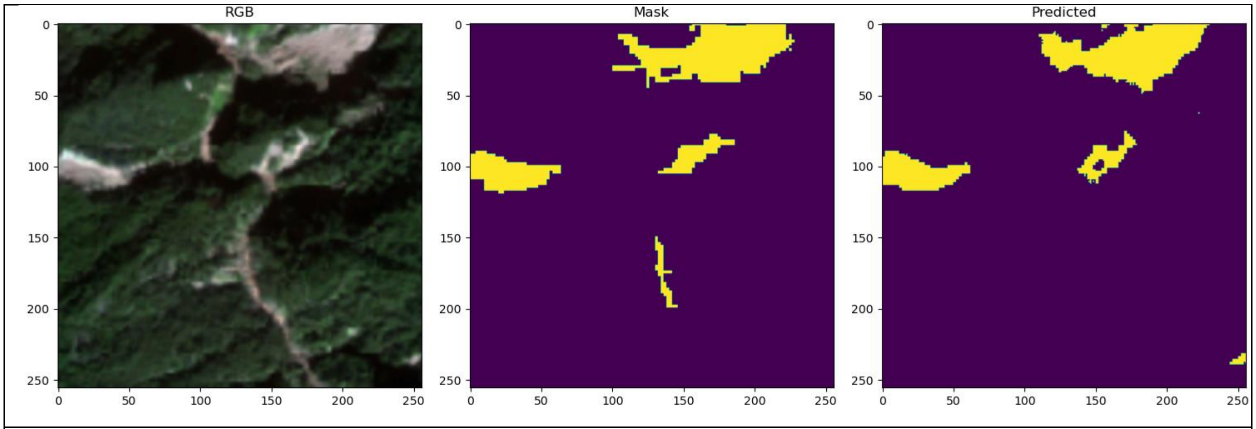
(c) ALL Channels



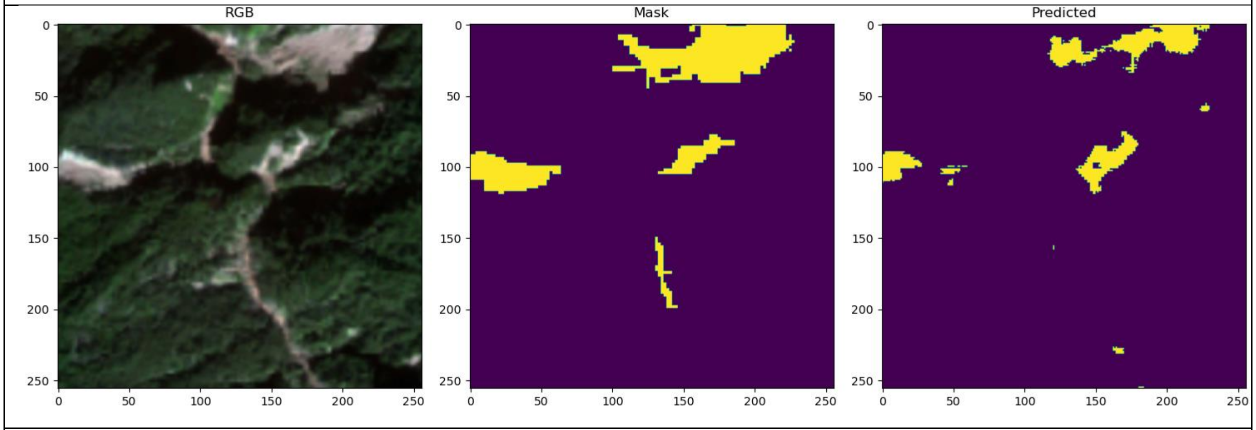
(d) RGB, + Edge



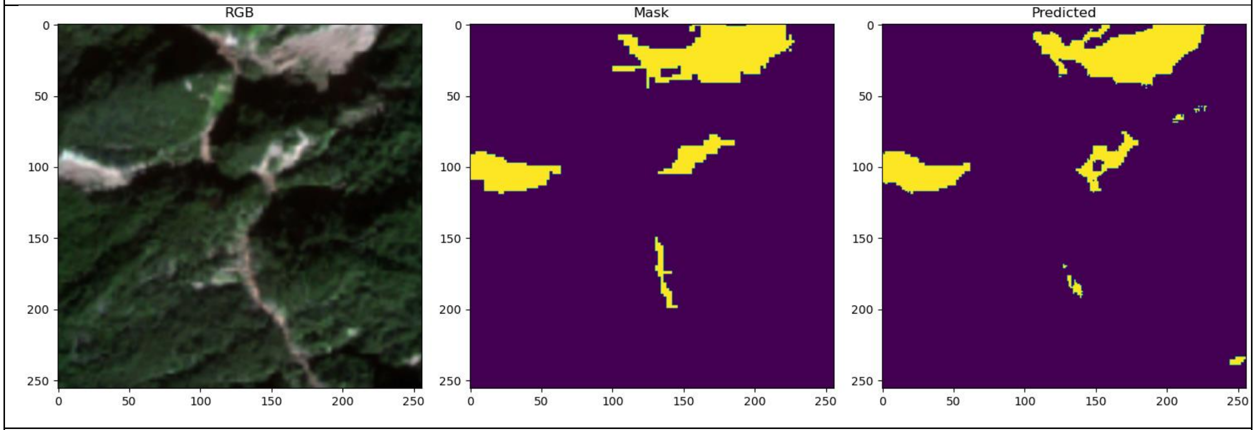
(e) RGB, Slope, DEM+ Edge



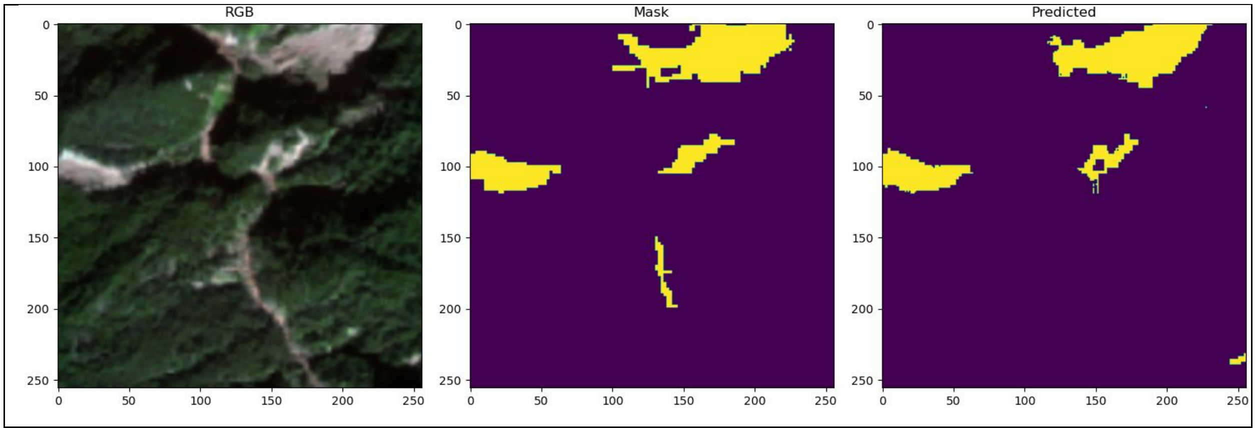
(f) ALL Channels +Edge



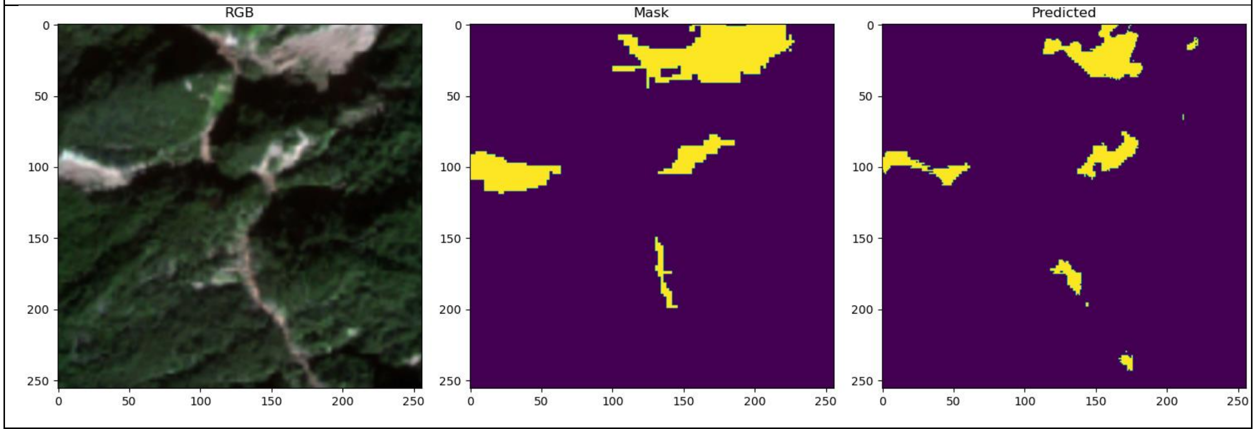
(g) RGB + PH



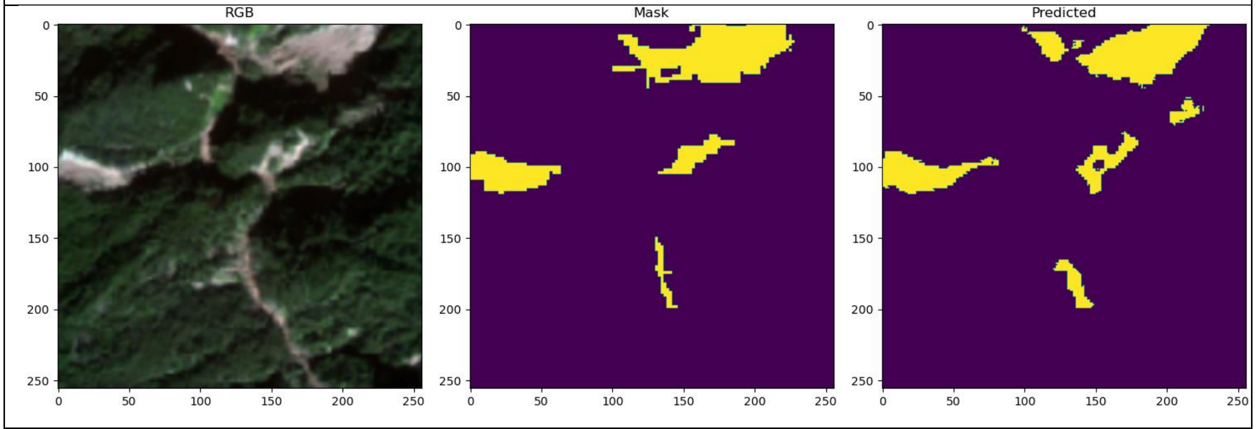
(h) RGB, Slope, DEM + PH



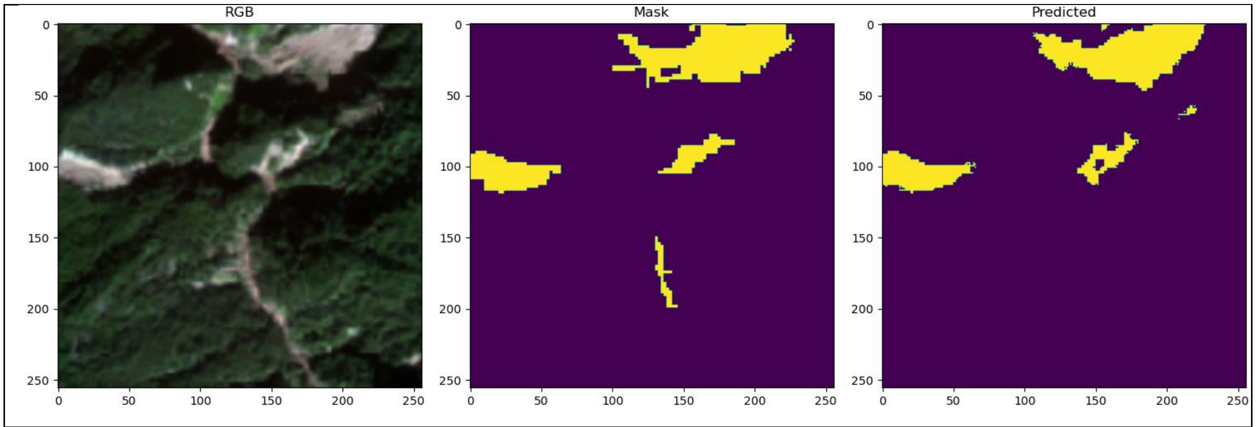
(i) ALL Channels+ PH



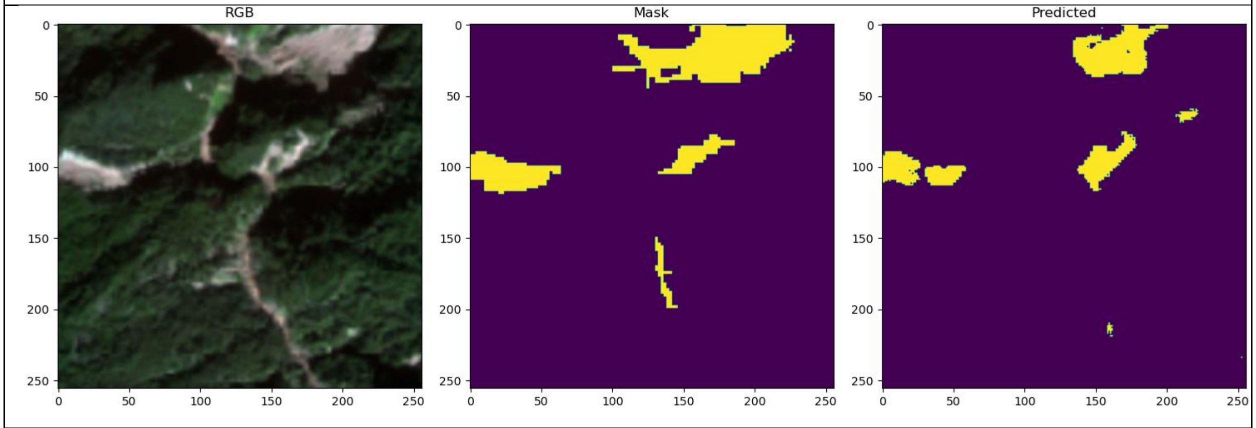
(j) RGB + Mapper



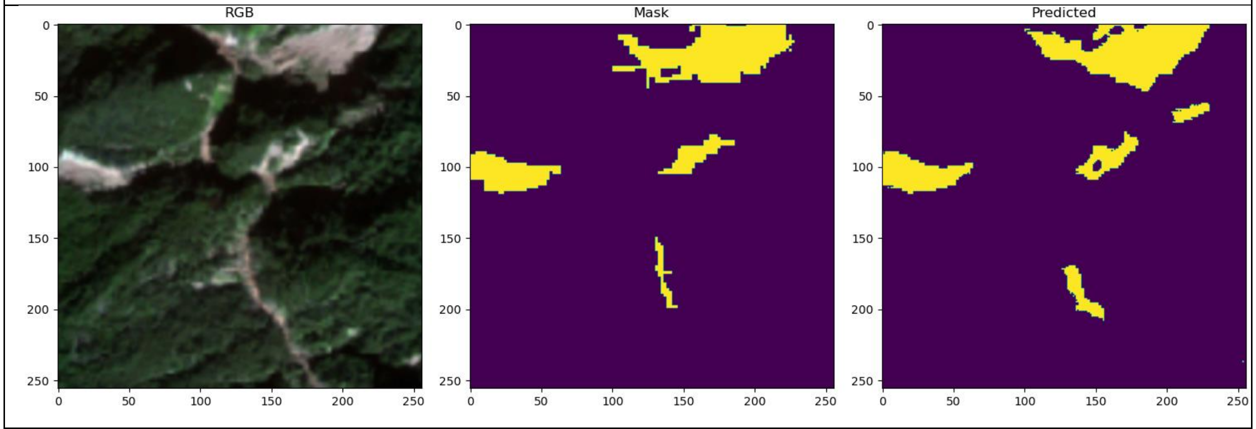
(k) RGB, Slope, DEM + Mapper



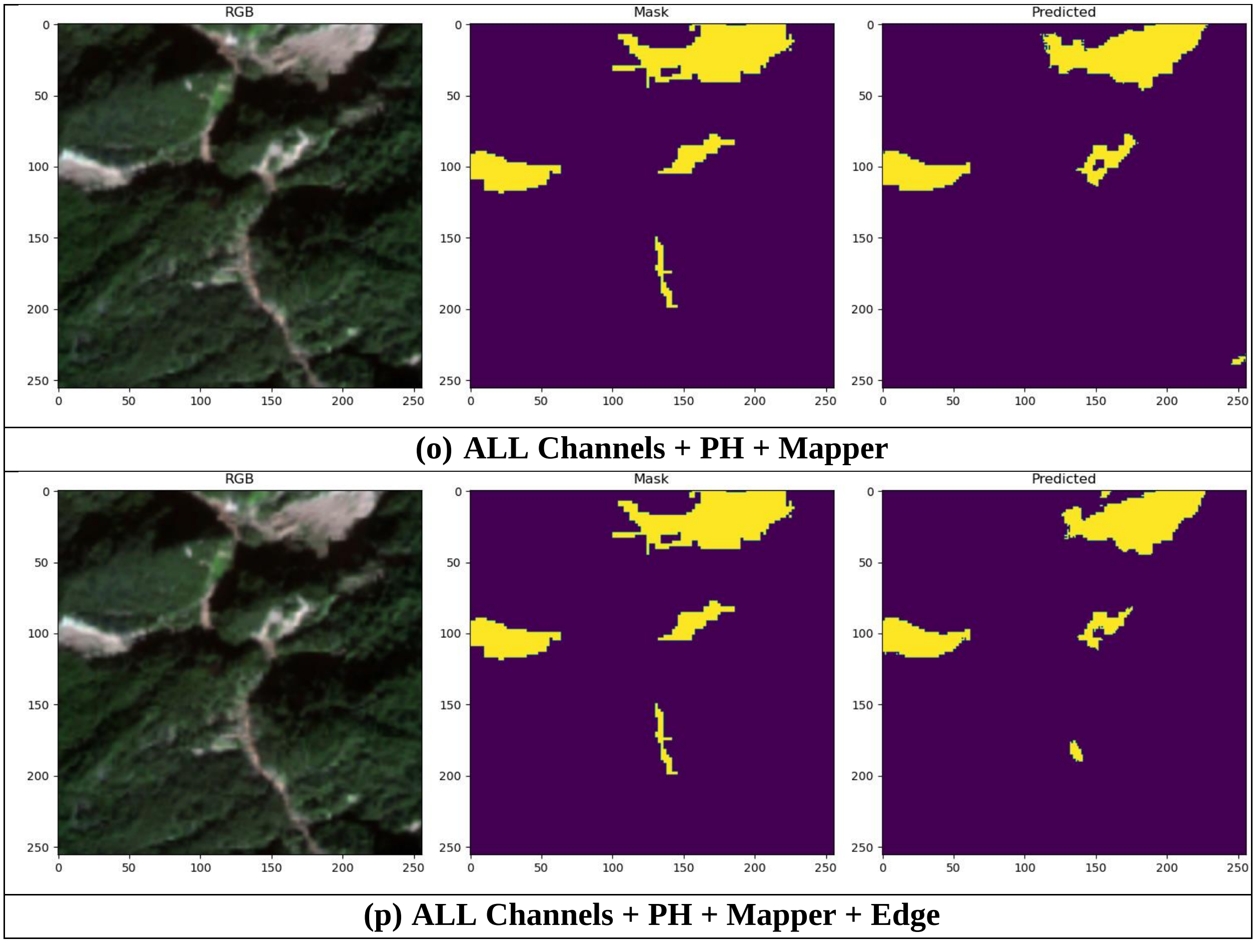
(l) ALL Channels + Mapper



(m)RGB + PH + Mapper



(n) RGB, Slope, DEM+ PH + Mapper



(o) ALL Channels + PH + Mapper

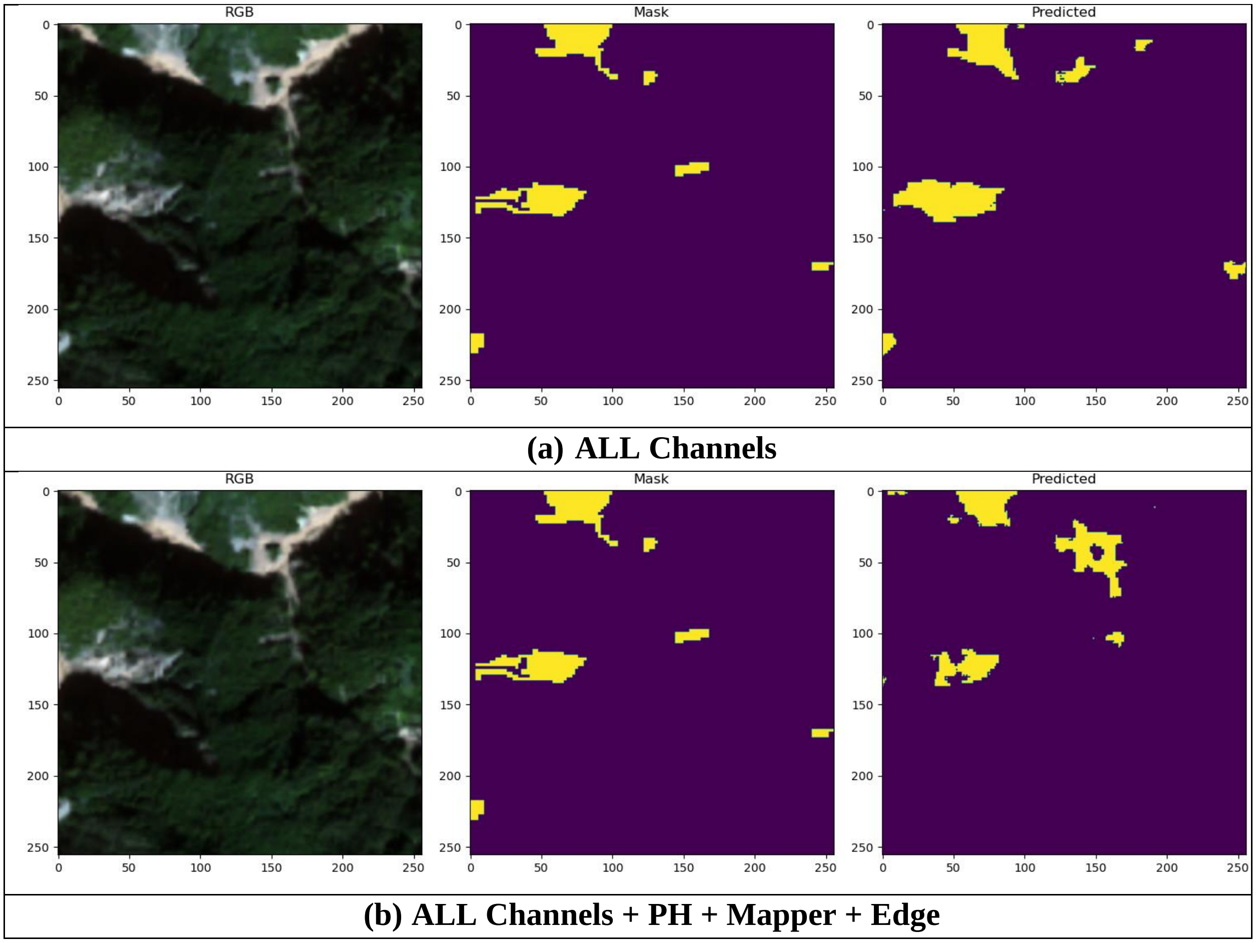
(p) ALL Channels + PH + Mapper + Edge

Figure 7.7 RGB image and predicted results of the model after 300 epochs of training

In addition to the previously discussed trends and findings, it is crucial to acknowledge the characteristics and limitations of the dataset used, which was derived from Sentinel-2 imagery. A notable constraint of this dataset is the absence of specific geographical coordinates. This limitation poses challenges in unequivocally determining whether the geospatial features identified by the ML models are indeed landslides.

During the experiments, detailed in Appendix B, it was observed that the trained models sometimes incorrectly classified bare earth areas as landslides. This misclassification can be attributed to the nature of the training labels, where landslides are consistently depicted as bare earth regions. Consequently, the models learned to associate this specific characteristic with landslides. When topological layers were incorporated into the models, a distinct pattern emerged. The models tended to focus on identifying and delineating the boundaries of circular features within the terrain, based on the premise that certain landslides exhibit circular patterns, albeit their specific shapes often generalized in the training labels. As a result, the labels may not accurately represent the intricate circular details within the actual landslide boundaries (refer to Figure 7.7 for illustration).

This methodology presents an additional challenge. There exist bare earth areas that inherently contain circular features but are not classified as landslides (as indicated in the labels). The ML models trained with topological information layers tend to identify these areas as potential landslides due to their circular content and bare earth texture, leading to false positives (as exemplified in Figure 7.8). These observations underscore the complexity of applying ML models to detect geospatial objects, particularly when dealing with datasets that lack detailed geographic coordinates and when using labels that may oversimplify or misrepresent the terrain's true characteristics. The findings also highlight the potential of topological information in improving detection accuracy, while simultaneously revealing the challenges in differentiating between true landslides and similar-looking geographical features.



(a) ALL Channels

(b) ALL Channels + PH + Mapper + Edge

Figure 7.8 RGB image and predicted results of the model after 300 epochs of training

In this chapter, we investigated the potential enhancement of geospatial object detection accuracy through the utilization of multichannel images in ML models, which include topological information and its location. The central hypothesis posited that integrating topological information into ML-based models could improve object detection accuracy, giving rise to specific research questions focused on the output, sensitivity, and representation of topological information.

8.0 Conclusions, Limitations, and Future Research

8.1 Conclusions

This thesis addresses the pressing challenges in geospatial object detection, particularly the escalating volume of geospatial data and the need for more effective processing. The main hypothesis in this thesis is that incorporating topological information in knowledge-based and ML-based methods will detect geospatial objects more accurately than when such information is not used. Three research questions guide the exploration in this thesis and PH-based, Topological KB, and ML-based geospatial object detection methods were developed and evaluated. Three sets of experiments, each addressing one of the three research questions, were conducted. The results from the experiments are:

- First, (in Chapter 4) we conducted a series of experiments where PH-based geospatial object detection method is used to detect landslides. We derived candidate polygons using PH and implemented filters based on topological information. In this experiment, we got better results than existing methods, though we did encounter FP. This observation led us to introduce additional filters.
- Second (in Chapter 6) we conducted a set of experiments where Topological KB geospatial object detection method is used to detect landslides. In this method, topological, geometrical, and contextual information was used for selecting candidate polygons. In this experiment, we observed that the integration of geometrical and contextual information into candidate polygons, that were derived based on their topological features using PH,

led to enhancements in accuracy but the accuracy was not improved by incorporating topological information.

- Third (in Chapter 7) we conducted a set of experiments by integrating topological information as an additional channel into multichannel image. Then we trained and tested various ML models under diverse parameters and differing numbers of input channels. In this experiment, we observed that in Topological ML, integration of topological information did not improve accuracy, but it has the potential to highlight object boundaries, adding a layer of precision to the detected objects, a fact that becomes particularly pronounced when the object possesses unique topological characteristics.

In summary, the test of the hypothesis indicates that incorporating new topological information, beyond the topological information embedded in the process of PH, does not improve detection accuracy of geospatial objects. This was evidence from the results of all experiments conducted by using PH, Topological KB, and Topological ML for detection of landslides.

8.2 Contributions

This thesis makes several contributions to the field of geospatial object detection:

1. Demonstrated the effectiveness of PH-based methods in detecting landslides, with an emphasis on integrating topological filters for improved accuracy.
2. Developed a novel LTFs extraction algorithm that is based on human perception of terrain.
3. Explored a Topological KB geospatial object detection method that uses combined topological, geometrical, and contextual information.

4. Explored the integration of topological information as an additional channel in ML models, highlighting its potential in improving object boundary precision.
5. Developed algorithms for transforming topological data into a multichannel image format suitable for ML applications.
6. Provided a thorough evaluation of the developed methods, offering insights into the role and impact of topological information in geospatial object detection.

8.3 Limitations

The thesis's limitations are:

1. The benchmarking approaches utilized to evaluate our proposed methods were suboptimal. In the Topological KB geospatial object detection method, we primarily relied on historical landslide locations furnished by official sources. Several of these locations were sourced from digitized historical landslide maps. Such a process introduces potential errors at various stages, such as during the original map creation and subsequent digitization.
2. The masks provided for the LandSlide4Sense competition posed challenges in evaluating the true capabilities of topological data. For instance, as illustrated in Figure 7.7, while our topological approach was adept at identifying distinct landslide boundaries, the masks used for assessing accuracy were rather generalized. This became evident, especially when topological information could detect unique features like holes within landslides which were not acknowledged by the generalized competition masks.
3. Our experiments were constrained using geometrical- and contextual-based rules that were tailored for specific study areas. These rules, derived from a literature review, represent just a

fraction of the potential detection methodologies. Numerous alternative rules have been cited across various papers, but we were unable to incorporate them owing to a lack of requisite data.

8.4 Future Research

While this thesis explored the use of topological information in geospatial object detection methods, the scope of exploration and experimentation can be further expanded. Highlighted below are some prospective research directions based on the work and findings of this thesis:

1. A pivotal direction for future research is revisiting the experiment presented in Chapter 6. By securing a more optimal ground truth dataset, we can better understand the efficacy of our methodologies and validate our findings further.
2. Current LTFs extraction algorithms have shown potential but also revealed areas for refinement. Leveraging ML models could offer a more effective algorithm for deriving LTFs. Additionally, such models can be tailored to potentially minimize noise and reduce FP rates.
3. Diverse rules, methodologies adapted from varied sources, and experimentation with new detection parameters to achieve improved geospatial object detection results will need to be explored.
4. In the realm of Topological ML, a promising avenue is the design of ML models that can inherently derive topological information from a provided image. Instead of relying on pre-existing topological information in a multichannel image, these models would generate topological layers automatically.

Bibliography

Access, P. S. D. (n.d.). Data Summary http://www. pasda. psu. edu/uci. *DataSummary. Aspx*.

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., & Ziegelmeier, L. (2017). Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, *18*.

Althuwaynee, O. F., Pradhan, B., Park, H.-J., & Lee, J. H. (2014). A novel ensemble bivariate statistical evidential belief function with knowledge-based analytical hierarchy process and multivariate statistical logistic regression for landslide susceptibility mapping. *Catena*, *114*, 21–36.

Aurenhammer, F. (1991). Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, *23*(3), 345–405.

Babic, Z. V., & Mandic, D. P. (2003). An efficient noise removal and edge preserving convolution filter. *6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service, TELSIKS 2003 - Proceedings*. https://doi.org/10.1109/TELSKS.2003.1246284

Bacha, A. S., Van Der Werff, H., Shafique, M., & Khan, H. (2020). Transferability of object-based image analysis approaches for landslide detection in the Himalaya Mountains of northern Pakistan. *International Journal of Remote Sensing*, *41*(9), 3390–3410.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Blaschke, T., Feizizadeh, B., & Hölbling, D. (2014). Object-based image analysis and digital terrain analysis for locating landslides in the Urmia Lake Basin, Iran. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(12), 4806–4817.

Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, *16*(1), 77–102.

Bunn, M. D., Leshchinsky, B. A., Olsen, M. J., & Booth, A. (2019). A simplified, object-based framework for efficient landslide inventorying using LIDAR digital elevation model derivatives. *Remote Sensing*. https://doi.org/10.3390/rs11030303

Burns, W. J., Maidin, I. P., & Ma, L. (2008). Statewide Landslide Information Database for Oregon (SLIDO), Release 1. *2008 Joint Meeting of The Geological Society of America, Soil Science Society of America, American Society of Agronomy, Crop Science Society of America, Gulf Coast Association of Geological Societies with the Gulf Coast Section of SEPM*.

Carlsson, G. (2009). Topology and data. *Bulletin of the American Mathematical Society*. https://doi.org/10.1090/S0273-0979-09-01249-X

Cesar, R. M., & da Fontoura Costa, L. (1995). A Pragmatic Introduction to Machine Vision, by R. Jain, R. Kasturi and B. G. Schunck. *Real-Time Imaging*. https://doi.org/10.1006/rtim.1995.1045

Chazal, F., & Michel, B. (2021). An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, *4*.

Cheng, G., & Han, J. (2016). A survey on object detection in optical remote sensing images. In *ISPRS Journal of Photogrammetry and Remote Sensing*. https://doi.org/10.1016/j.isprsjprs.2016.03.014

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*. https://doi.org/10.1177/001316446002000104

Edelsbrunner, H., & Harer, J. L. (2022). *Computational topology: an introduction*. American Mathematical Society.

Fanos, A. M., Pradhan, B., Mansor, S., Yusoff, Z. M., & Abdullah, A. F. bin. (2018). A hybrid model using machine learning methods and GIS for potential rockfall source identification from airborne laser scanning data. *Landslides*. https://doi.org/10.1007/s10346-018-0990-4

Guzzetti, F., Mondini, A. C., Cardinali, M., Fiorucci, F., Santangelo, M., & Chang, K. T. (2012a). Landslide inventory maps: New tools for an old problem. In *Earth-Science Reviews*. https://doi.org/10.1016/j.earscirev.2012.02.001

Guzzetti, F., Mondini, A. C., Cardinali, M., Fiorucci, F., Santangelo, M., & Chang, K. T. (2012b). Landslide inventory maps: New tools for an old problem. In *Earth-Science Reviews*. https://doi.org/10.1016/j.earscirev.2012.02.001

Hafiz, A. M., & Bhat, G. M. (2020). A survey on instance segmentation: state of the art. *International Journal of Multimedia Information Retrieval*, *9*(3), 171–189.

Hatcher, A. (2005). *Algebraic topology*. 清华大学出版社有限公司.

Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(7), 2217–2226.

Hensel, F., Moor, M., & Rieck, B. (2021). A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, *4*, 52.

Hoeser, T., Bachofer, F., & Kuenzer, C. (2020). Object detection and image segmentation with deep learning on Earth observation data: A review—Part II: Applications. *Remote Sensing*, *12*(18), 3053.

Hoeser, T., & Kuenzer, C. (2020). Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends. *Remote Sensing*, *12*(10), 1667.

Hölbling, D., Füreder, P., Antolini, F., Cigna, F., Casagli, N., & Lang, S. (2012). A semi-automated object-based approach for landslide detection validated by persistent scatterer interferometry measures and landslide inventories. *Remote Sensing*. https://doi.org/10.3390/rs4051310

Hong, H., Pradhan, B., Xu, C., & Bui, D. T. (2015). Spatial prediction of landslide hazard at the Yihuang area (China) using two-class kernel logistic regression, alternating decision tree and support vector machines. *Catena*, *133*, 266–281.

Horn, B. K. P. (1981). Hill Shading and the Reflectance Map. *Proceedings of the IEEE*. https://doi.org/10.1109/PROC.1981.11918

Jasiewicz, J., & Stepinski, T. F. (2013). Geomorphons-a pattern recognition approach to classification and mapping of landforms. *Geomorphology*, *182*, 147–156. https://doi.org/10.1016/j.geomorph.2012.11.005

Jensen, J. R. (1996). Introductory digital image processing: a remote sensing perspective. Second edition. In *Introductory digital image processing: a remote sensing perspective. Second edition*.

Jenson, S. K., & Domingue, J. O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*.

Karimi, B., Yanchuck, M., & Foust, J. (2019). A new landslide inventory and improved susceptibility model for northeastern Pennsylvania. *Environmental Geosciences*. https://doi.org/10.1306/eg.09191919008

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*. https://doi.org/10.2307/2529310

LandSlide4Sense. (n.d.). Retrieved April 12, 2022, from https://www.iarai.ac.at/landslide4sense/challenge/

Lee, J.-G., & Kang, M. (2015). Geospatial big data: challenges and opportunities. *Big Data Research*, *2*(2), 74–81.

Li, K., Wan, G., Cheng, G., Meng, L., & Han, J. (2020). Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, *159*, 296–307.

Liao, W. H. (2010). Region description using extended local ternary patterns. *Proceedings - International Conference on Pattern Recognition*. https://doi.org/10.1109/ICPR.2010.251

Liu, J.-K., Hsiao, K.-H., & Shih, P. T.-Y. (2012). A geomorphological model for landslide detection using airborne LIDAR data. *Journal of Marine Science and Technology*, *20*(6), 4.

Long, Y., Gong, Y., Xiao, Z., & Liu, Q. (2017). Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(5), 2486–2498.

Lü, G., Batty, M., Strobl, J., Lin, H., Zhu, A.-X., & Chen, M. (2019). Reflections and speculations on the progress in Geographic Information Systems (GIS): a geographic perspective. *International Journal of Geographical Information Science*, *33*(2), 346–367.

Martha, T. R., Kerle, N., Jetten, V., van Westen, C. J., & Kumar, K. V. (2010). Characterising spectral, spatial and morphometric properties of landslides for semi-automatic detection using object-oriented methods. *Geomorphology*, *116*(1–2), 24–36.

Martha, T. R., Kerle, N., Van Westen, C. J., Jetten, V., & Kumar, K. V. (2011). Segment optimization and data-driven thresholding for knowledge-based landslide detection by object-based image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, *49*(12), 4928–4943.

Matejka, J., & Fitzmaurice, G. (2017). Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 1290–1294.

Mezaal, M. R., Pradhan, B., Sameen, M. I., Shafri, H. Z. M., & Yusoff, Z. M. (2017). Optimized neural architecture for automatic landslide detection from high-resolution airborne laser scanning data. *Applied Sciences (Switzerland)*. https://doi.org/10.3390/app7070730

Oksuz, K., Cam, B. C., Kalkan, S., & Akbas, E. (2020). Imbalance problems in object detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(10), 3388–3415.

Otter, N., Porter, M. A., Tillmann, U., Grindrod, P., & Harrington, H. A. (2017). A roadmap for the computation of persistent homology. *EPJ Data Science*. https://doi.org/10.1140/epjds/s13688-017-0109-5

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., & Antiga, L. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32*.

Pereira, C. M. M., & De Mello, R. F. (2015). Persistent homology for time series and spatial data clustering. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2015.04.010

Pham, M.-T., Courtrai, L., Friguet, C., Lefèvre, S., & Baussard, A. (2020). YOLO-Fine: One-stage detector of small objects under various backgrounds in remote sensing images. *Remote Sensing*, *12*(15), 2501.

Pirotti, F., & Tarolli, P. (2010). Suitability of LiDAR point density and derived landform curvature maps for channel network extraction. *Hydrological Processes*. https://doi.org/10.1002/hyp.7582

Praveen, P., Babu, C. J., & Rama, B. (2016). Big data environment for geospatial data analysis. *2016 International Conference on Communication and Electronics Systems (ICCES)*, 1–6.

Preparata, F. P. (1984). Computational Geometry—A Survey. *IEEE Transactions on Computers*. https://doi.org/10.1109/TC.1984.1676388

Pun, C. S., Xia, K., & Lee, S. X. (2018). Persistent-Homology-Based Machine Learning and Its Applications -- A Survey. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3275996

Qiu, S., Zhu, Z., & Woodcock, C. E. (2020). Cirrus clouds that adversely affect Landsat 8 images: what are they and how to detect them? *Remote Sensing of Environment*, *246*, 111884.

Rana, S. (2006). Use of plan curvature variations for the identification of ridges and channels on DEM. *Progress in Spatial Data Handling - 12th International Symposium on Spatial Data Handling, SDH 2006*. https://doi.org/10.1007/3-540-35589-8_49

Rau, J.-Y., Jhan, J.-P., & Rau, R.-J. (2013). Semiautomatic object-oriented landslide recognition scheme from multisensor optical imagery and DEM. *IEEE Transactions on Geoscience and Remote Sensing*, *52*(2), 1336–1349.

Ristovska, D., & Sekuloski, P. (2019). MAPPER ALGORITHM AND IT'S APPLICATIONS. *Mathematical Modeling*, *3*(3), 79–82.

Salnikov, V., Cassese, D., & Lambiotte, R. (2018). Simplicial complexes and complex systems. *European Journal of Physics*, *40*(1), 14001.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, *42*(3), 1–21.

Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*. https://doi.org/10.2200/S00429ED1V01Y201207AIM018

Singh, G., Mémoli, F., & Carlsson, G. E. (2007). Topological methods for the analysis of high dimensional data sets and 3d object recognition. *PBG@ Eurographics*, *2*.

Syzdykbayev, M., Karimi, B., & Karimi, H. A. (2020a). A Method for Extracting Some Key Terrain Features from Shaded Relief of Digital Terrain Models. *Remote Sensing*, *12*(17), 2809.

Syzdykbayev, M., Karimi, B., & Karimi, H. A. (2020b). Persistent homology on LiDAR data to detect landslides. *Remote Sensing of Environment*. https://doi.org/10.1016/j.rse.2020.111816

Syzdykbayev, M., & Karimi, H. A. H. A. (2020). Persistent Homology for Detection of Objects from Mobile LiDAR Point Cloud Data in Autonomous Vehicles. *Advances in Intelligent Systems and Computing, 944*. https://doi.org/10.1007/978-3-030-17798-0_37

Szeliski, R. (2010a). *Computer vision: algorithms and applications*. Springer Science & Business Media.

Szeliski, R. (2010b). *Computer vision: algorithms and applications*. Springer Science & Business Media.

Tchinda, B. S., Tchiotsop, D., Noubom, M., Louis-Dorr, V., & Wolf, D. (2021). Retinal blood vessels segmentation using classical edge detection filters and the neural network. *Informatics in Medicine Unlocked*, *23*, 100521.

Tsangaratos, P., & Ilia, I. (2016). Landslide susceptibility mapping using a modified decision tree classifier in the Xanthi Perfection, Greece. *Landslides*. https://doi.org/10.1007/s10346-015-0565-6

Tymochko, S., Munch, E., Dunion, J., Corbosiero, K., & Torn, R. (2020). Using persistent homology to quantify a diurnal cycle in hurricanes. *Pattern Recognition Letters, 133*, 137–143.

Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res, 10*(66–71), 13.

Van Etten, A., Lindenbaum, D., & Bacastow, T. M. (2018). Spacenet: A remote sensing dataset and challenge series. *ArXiv Preprint ArXiv:1807.01232*.

Wang, Y., Bashir, S. M. A., Khan, M., Ullah, Q., Wang, R., Song, Y., Guo, Z., & Niu, Y. (2022). Remote sensing image super-resolution and object detection: Benchmark and state of the art. *Expert Systems with Applications*, 116793.

Weber, P., & Haklay, M. (2008). OpenStreetMap: user-generated street maps. *IEEE Pervasive Computing*.

Wieland, M., Martinis, S., Kiefl, R., & Gstaiger, V. (2023). Semantic segmentation of water bodies in very high-resolution satellite and aerial images. *Remote Sensing of Environment*, *287*, 113452.

World Machine. (n.d.). Retrieved August 1, 2020, from https://www.world-machine.com/

Xiao, Z., Song, K.-Y., & Gupta, M. M. (2021). Development of a CNN edge detection model of noised X-ray images for enhanced performance of non-destructive testing. *Measurement*, *174*, 109012.

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017a). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350–361.

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017b). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350–361.

Zhou, X., Li, W., & Arundel, S. T. (2019). A spatio-contextual probabilistic model for extracting linear features in hilly terrains from high-resolution DEM data. *International Journal of Geographical Information Science*. https://doi.org/10.1080/13658816.2018.1554814

Zomorodian, A. (2010). Fast construction of the Vietoris-Rips complex. *Computers & Graphics*, *34*(3), 263–271.

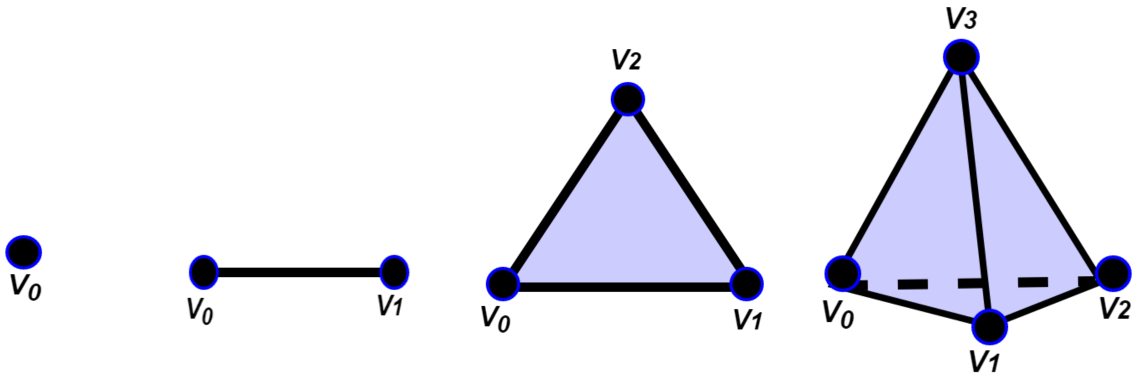
Appendix A Computation of PH and Mapper

Appendix A.1 Simplicial Complexes

Simplicial complex and its properties are utilized to derive topological information using the PH. The construction of the simplicial complex from geospatial data is discussed in the next section. Simplicial complexes are used to approximate complicated shapes. They highlight the structure of the data and the underlying topological information at different scales (Salnikov, Cassese, & Lambiotte, 2018). In other words, to get information about the shape of the object, instead of using every element of the object, the object can be sampled by using a set of points which then they are used to construct simplicial complexes. Formally, a simplicial complex K is a topological object that is constructed as a union of points, line segments, triangles, tetrahedra, and higher-dimensional geometrical objects. The building block of a simplicial complex is a simplex (simplices if plural). A simplex is an analog of a point, line segment, and triangle. The dimension d of a simplex (d-simplex) is identified by the number of elements in the set (Otter et al., 2017). For example, if a simplex consists of two elements, its dimension d is equal to 1, if a simplex consists of three elements its dimension d is equal to 2, and so on. More information about simplicial complexes can be found in (Edelsbrunner & Harer, 2022; Hatcher, 2005; Otter et al., 2017; Salnikov et al., 2018).

In general, a d-simplex, called σ_d, is the convex hull of $d + 1$ linearly independent vertices which exist in a d-dimensional Euclidean space. A 0-simplex, σ_0, is a point consisting of one vector $\{v_0\}$; a 1-simplex, σ_1, consisting of two vectors $\{v_0, v_1\}$ connected by a line segment; a 2-simplex, σ_2, consisting of three vectors $\{v_0, v_1, v_2\}$ connected by a three-line segment or a triangle;

a 3 -simplex, σ_3, consisting of four vectors $\{v_0, v_1, v_2, v_3\}$ connected by four triangles or a tetrahedron (Appendix Appendix Figure), and so on.



0-simplex (point)

1-simplex (line)

2-simplex (triangle)

3-simplex (tetrahedron)

Appendix Figure 1 Example of simplices

The simplex has boundaries called the faces of a simplex. A face of d- simplex σ_d, is a sub-simplex σ_f, where $f < d$, is the simplex generated by a subset of the vertices of σ_d. For that reason, if one of the vertices v_i from d-simplex is deleted, the face of the d-simplex will be created.

$$d - \text{simplex } = \{v_0, \dots., v_\text{d}\}$$
$$(d - 1) - \text{simplex} = \{v_0, \dots \widehat{v_i} \dots, v_\text{d}\}$$

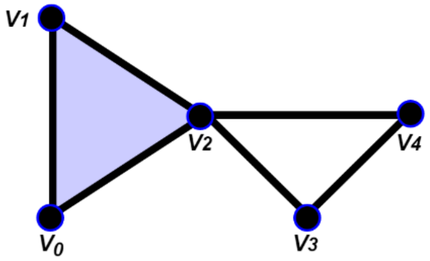
For example, the face of a triangle (2-simplex or σ_2) that consists of vertices $\{v_0, v_1, v_2\}$ can be a line 1-simplex (σ_1) $\{v_0, \widehat{v_1}, v_2\ \}$ and determined by the remaining vertices $\{v_0, v_2\}$. The symbol $\widehat{v_1}$ means that v_1 was removed from the set.

A simplicial complex K is a collection of the above-mentioned simplices, but these simplices need to be combined in a certain way that is based on specific conditions (Salnikov et al., 2018):

- If a simplicial complex K contains simplex σ, then every face of a simplex σ also belongs to K
- If any two simplices σ_1 and σ_2 in K intersect $\sigma_1 \cap \sigma_2$ and $\sigma_1 \cap \sigma_2 = \emptyset$, then their intersection is a common face of both σ_1 and σ_2

In other words, there are no missing building blocks in simplicial complex K. For example, simplicial complex K shown in Appendix Figure 2 should contain the following vertices:

$K = \{\langle v_0, v_1, v_2 \rangle, \langle v_0, v_1\rangle, \langle v_0, v_2\rangle, \langle v_1, v_2\rangle, \langle v_2, v_3\rangle, \langle v_2, v_4\rangle, \langle v_3, v_4\rangle, \langle v_0\rangle, \langle v_1\rangle, \langle v_2\rangle, \langle v_3\rangle, \langle v_4\rangle\}$



Appendix Figure 2 Simplicial complexes K

Simplicial complexes have specific properties that will allow us to derive topological information. These properties are the orientation of simplicial complexes, breaking into chain groups, and boundary operations.

Orientation of simplicial complexes. Each simplex in the simplicial complex has a property, orientation, which is defined by the order of its vertices. If a d-simplex has a fixed orientation, d-simplex will be denoted as oriented simplex $\vec{\sigma}$. The oriented simplex has a specific property. The property is that switching two vertices inside the d-simplex introduces a minus sign, for example,

$$[v_0, \boldsymbol{v_1}, \boldsymbol{v_2}] = -[v_0, \boldsymbol{v_2}, \boldsymbol{v_1}]$$

and simplicial complexes with different orders are not equal. Using this property, the d-simplex with integer coefficients can be added (Salnikov et al., 2018). In a way, one can think of an oriented simplex as representing an action. For example, $\overrightarrow{\sigma_1}\,[v_0, v_1]$ represents moving from v_0 to v_1, so moving from v_0 to v_1 and then from v_1 to v_0, is the same as adding the two movements, which is equal to staying in one place:

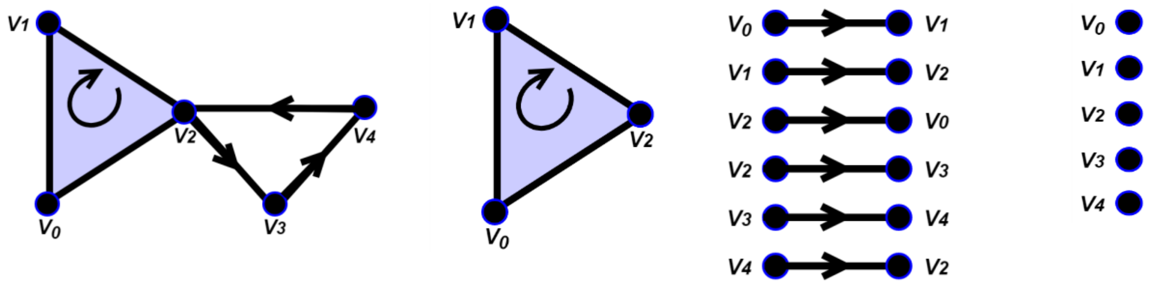
$$[v_0, v_1] + [v_1, v_0] = [v_0, v_1] - [v_0, v_1] = 0$$

From the above information, an oriented simplicial complex is a simplicial complex K with oriented simplices and denoted by $\vec{K}$. For example, oriented simplicial complex $\vec{K}$ (Appendix Figure a) contains the following vertices:

$\overrightarrow{K}$= {$[v_0, v_1, v_2]$, $[v_0, v_1]$, $[v_0, v_2]$, $[v_1, v_2]$, $[v_2, v_3]$, $[v_2, v_4]$, $[v_3, v_4]$, $[v_4, v_5]$, $[v_0]$, $[v_1]$, $[v_2]$, $[v_3]$, $[v_4]$}

Breaking into chain groups. A simplicial complex can be divided into groups of simplices with similar dimensions. In other words, each group is composed of a set of all d-simplices (Salnikov et al., 2018). For example, a simplicial complex $\vec{K}$can be divided into several chain groups (Appendix Figure 3) and for any group, the d-th chain group is the sum of simplexes inside each chain group.

$C_2(\vec{K}) = [v_0, v_1, v_2]$ (Appendix Figure 3b)
$C_1(\vec{\mathrm{K}}) = [v_0, v_1], [v_1, v_2], [v_2, v_0], [v_2, v_3], [v_3, v_4], [v_4, v_2]$ (Appendix Figure 3c)
$C_0(\vec{\mathrm{K}}) = [v_0], [v_1], [v_2], [v_3], [v_4]$ (Appendix Figure 3d)



a: $\overrightarrow{K}$ b: $C_2(\overrightarrow{K})$ c: $C_1(\overrightarrow{K})$ d: $C_0(\overrightarrow{K})$

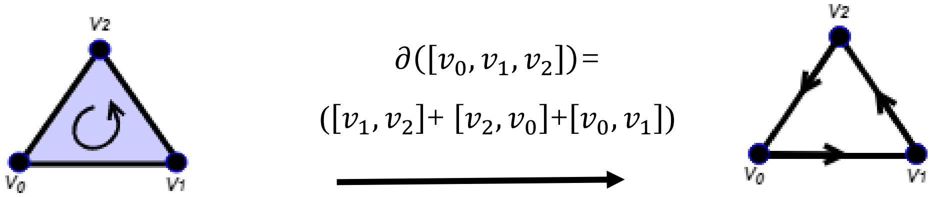
Appendix Figure 3 ((a) an oriented simplicial complex $\overrightarrow{K}$; (b) Chain group C_2 a set of 2-simplices; (c) C_1 a set of 1-simplices; (d) C_0 a set of 0-simplices

Boundary operations. There are several properties of simplicial complexes that are related to the sign of every other face. The equation to calculate the boundary of a simplex can be written as follows:

$$\partial(\overrightarrow{\sigma}) = \sum_{i=0}^{k} (-1)^i \, (v_0 \,, \ldots \widehat{v_i}, \ldots v_k \,) \qquad \text{(A.1)}$$

For example, using the equation presented above, the boundary of the 2-simplex, a triangle, can be calculated as follows (Appendix Figure 4):

$$\begin{aligned}\partial([v_0, v_1, v_2]) &= (-1)^0[\widehat{v_0} \,, v_1, v_2] + \, (-1)^1[v_0 \,, \widehat{v_1} \,, v_2 \,] + \, (-1)^2[v_0 \,, v_1, \widehat{v_2}] \\ &= \, [v_1, v_2] - [v_0, v_2] + [v_0, v_1] \\ &= \, [v_1, v_2] + [v_2, v_0] + [v_0, v_1]\end{aligned}$$



$\partial([v_0, v_1, v_2])=$

$([v_1, v_2]+ [v_2, v_0]+[v_0, v_1])$

Appendix Figure 4 Boundary operation on 2-simplex

The second property is that in the chain group the boundary of the boundary is equal to 0 (Equation A.2) and the boundary of the 0-simplex σ_0 is equal to 0:

$$\partial^2 \ = 0 \ and \ \partial_0 = 0 \qquad (A.2)$$

For example, the boundary of the boundary of the 2-simplex, a triangle (Appendix Figure 4), can be calculated as follows:

$$\partial^2([v_0, v_1, v_2])=\partial([v_1, v_2]+[v_2, v_0]+[v_0, v_1])=(+v_1-v_2+v_2-v_0+v_0-v_1)= 0$$

The third property is that the boundary of the chain group C_d is included inside chain group C_{d-1}. Using these properties, the simplicial complex chain can be written as the following:

$$C_{d+1} \rightarrow \ C_d \rightarrow C_{d-1} \dots \dots \rightarrow C_0 \rightarrow 0 \qquad (A.3)$$

For example, if d = 2, the boundaries of triangle $C_2(\vec{K})$ (Figure 2.6b) are included inside $C_1(\vec{K})$ that are lines (Figure 2.6c), and the boundary of the point $C_0(\vec{K})$ is equal to 0.

Appendix A.2 Construction of Simplicial Complexes from Geospatial Datasets

To construct simplicial complexes from a geospatial dataset, two tasks must be performed: (1) convert the geospatial dataset into a set of points with corresponding distance information and (2) select a simplicial complex type. There are different types of simplicial complex, each with different properties. The common simplicial complexes are Čech complex, Vietoris-Rips (VR) complex, alpha complex, and witness complex (Pun et al., 2018). We will discuss two more common simplicial complexes that are used in the computation of PH (Otter et al., 2017). These two complexes are VR complex and alpha complex. We discuss these two complexes because the

VR complex is simple and practical to explain, and the alpha complex is computationally efficient and is used in experimentation (see Chapter 4).

VR complex is a complex that is practical from a computational standpoint and explanation (Zomorodian, 2010). To construct VR complex from a set of points P, first, each point is initialized as 0-complex. Then, a parameter r as the radius of the circle that is created around each point should be defined (Appendix Figure 5a). Next, for all subsets in the set of points, if the distance between two points is less than $2r$, these points are connected and create a line segment. If the pairwise distance between three points is less than $2r$, these three points create a triangle, and so on. VR complex is calculated by using the following equation (Zomorodian, 2010):

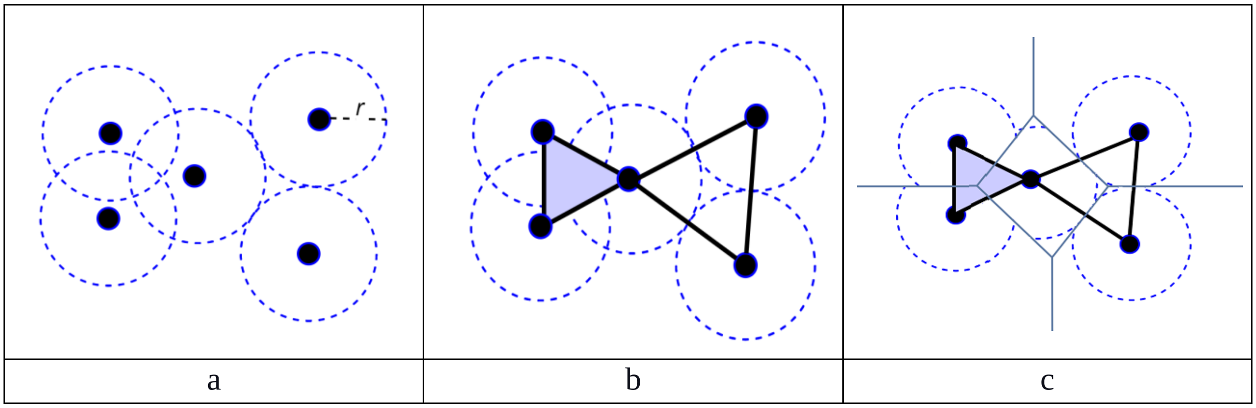
$$VR_r(P) = \{\sigma \subseteq P \mid distance(v_i, v_j) \leq 2r, v_i \neq v_j \in \sigma \} \tag{A.4}$$

In other words, first circles with a radius of r around each point are created. Then, if these circles intersect, simplices with these points will be created (Appendix Figure 5b).

Alpha complex relies on a distance parameter r and is restricted by a Voronoi diagram V_p (Pun et al., 2018). A Voronoi diagram divides an area with a set of points P into polygons where each polygon contains exactly one point P_i and every point inside each polygon is closer to the point P_i. Alpha complex is calculated by using the following equation (Aurenhammer, 1991):

$$A_r(P) = \{\sigma \subseteq P \mid distance(v_i, v_j) \leq 2r \cap V_p \} \tag{A.5}$$

In other words, the Voronoi diagram allows for the restriction of the dimension d-simplices based on the dimension of the input geospatial set of points (Appendix Figure 5c).



Appendix Figure 5 Simplicial complex construction: (a) set of points with a circle around each point with a radius of r; (b) a VR complex created from the set of points; (c) an Alpha complex created from the set of points

Appendix A.3 Persistent Homology

PH is applied to obtain topological information from data by studying the corresponding data connections and gaps (Otter et al., 2017). PH utilizes all three above discussed properties of simplicial complexes to derive topological information at different dimensions. To derive topological information from a given dataset, Chazal & Michel, (2021) presented the basic pipeline with four steps as follows (Appendix Figure 6).

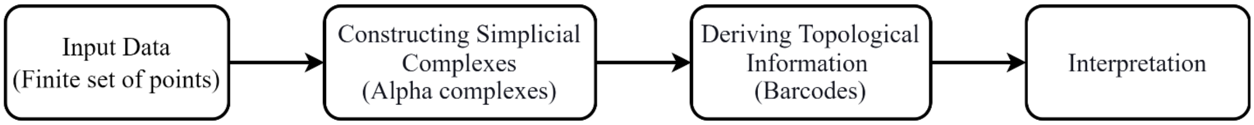
1. Input to the PH a finite set of points with corresponding distance information (Carlsson, 2009). The distance metric depends on the application, and the choice of the correct metric is essential. For example, protein data metrics can be measured in nanometers, and metrics for satellite image analysis can be measured in meters.
2. Construct a nested sequence of simplicial complexes from the set of points using different values of r (Equation A.10).

3. Derive topological information from the nested sequence of simplicial complexes. This step consists of two functions:

   C. *homology group* returns topological information given the simplicial complex that was constructed using the r.

   D. *persistent homology* utilizes *homology group* with a different value of r, and records each change. In other words, if the parameter r is changed, the topological information associated with the newly created simplicial complexes will change as well, and this second function records these changes.

4. Use the extracted topological information as a feature or descriptor for the dataset to assist in better understanding the dataset. This topological information can be visualized or can be a feature used in ML models.



Appendix Figure 6 PH pipeline (Chazal & Michel, 2021)

Homology group. In general, computing homology groups involve identifying holes in a specific dimension d in each oriented simplicial complex $\vec{K}$, where holes in dimension 0 are connected components (Otter et al., 2017). To compute homology groups, two operations need to be performed. The first operation is identifying all cycles in the given chain group C_d. These cycles are identified by selecting simplices whose boundary is equal to 0 and defined as:

$$Z_d = ker(\partial_d) = \{\sigma \in C_d(\vec{K}) | \partial(\sigma) = 0\} \quad \text{(A.6)}$$

In other words, all circles that exist in chain group $C_d(\vec{K})$ should be identified.

The second operation is identifying boundaries of $C_{d+1}(\vec{K})$, defined as:

$$B_d = im(\partial_{d+1}) = \{\partial(\sigma| \ \sigma \in C_{d+1}(\vec{K})\} \qquad \text{(A.7)}$$

From the relation between Z_d and B_d, every element of B_d is an element of Z_d. B_d and Z_d contain all the necessary information to compute the holes in $\vec{K}$. Intuitively, the cycles that exist in $C_d(\vec{K})$ but are not the boundaries of $C_{d+1}(\vec{K})$ are holes in the d dimension of $\vec{K}$ can be calculated using the following equation:

$$H_d = \frac{Z_d}{B_d} \qquad \text{(A.8)}$$

Finally, to use outputs of the homology group, Betti numbers β that carry topological information need to be derived.

$$\beta_d(\vec{K}) = dim(H_d) \qquad \text{(A.9)}$$

Betti number is the number of holes that exist in a specific chain group C_d. For example, the Betti numbers from small dimensions have geometric interpretations:

- $\beta_0(\vec{K})$: connected components
- $\beta_1(\vec{K})$: circles in two-dimensional space
- $\beta_2(\vec{K})$: voids in three-dimensional space

Persistent homology. The construction of simplicial complexes is dependent on the value of parameter r. If parameter r is equal to 0, every point is isolated (Appendix Figure 7a), and $\beta_0(\vec{K})$ will be equal to the number of points and $\beta_1(\vec{K})$ and $\beta_2(\vec{K})$ will be 0. With a large value of the parameter r, $\beta_0(\vec{K})$ will be 1 and every point will be connected (Appendix Figure 7d). Identifying the correct value of parameter r is vital to extract the topological information of the data. For example, in the set of points shown in Appendix Figure 7 (a), if parameter r is set to 1, two small circles are created (Appendix Figure 7b). However, if parameter r is set to 2, two small circles disappear, and one large circle is created. PH uses all possible values of parameter r and captures how the homology of the complexes $\vec{K}$ changes as the value of parameter r increases, and it detects the features that '*persist'* across changes in the values of parameter r.

To track these changes, a nested sequence of simplicial complexes $\vec{K}$, called filtered simplicial complexes, is required, and can be defined as:

$$\vec{K}_0 \subset \vec{K}_1 \subset \cdot \ \vec{K}_i \subset \vec{K}_j \subset \cdot \cdot \cdot \subset \vec{K}_l \ = \vec{K} \qquad \text{(A.10)}$$

A filtration complex is a sequence of simplicial complexes generated by continuously increasing the parameter r (Appendix Figure 7). In other words, a series of VR or Alpha complexes $\vec{K}_i$ with different values of the parameter r can be constructed from a set of points and can be defined as:

$$0 \leq i \leq j \leq l$$

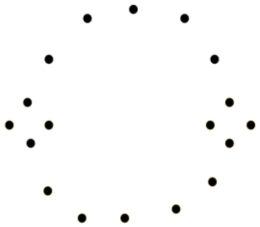
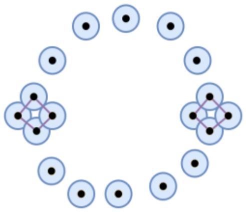
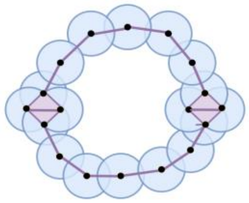
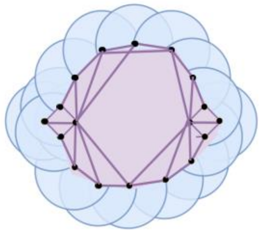
where l is the largest value of parameter r. Then, homology can be applied to record changes in Betti numbers β_d with function $f_{i,j}$,while the value of parameter r changes from i to j. The function can be written by using the following equation:

$$f_{i,j} : H(\vec{K}_i) \rightarrow H(\vec{K}_j) \qquad \text{(A.11)}$$

and this function records the following features:

- Homology groups that are born at i
- Homology groups that persist from $i \rightarrow j$
- Homology groups that die at j

Each homology class can be identified with a birth time and a death time. Features that are born and die soon after, are often considered to be topological noise, whereas features that persist for an extended period are considered to be true features of the underlying structure (Appendix Figure 7*c*).

			
a • $\vec{K}_1, r = 0$ • $\beta_0(\vec{K}) = 18$ • $\beta_1(\vec{K}) = 0$	b • $\vec{K}_2, \mathrm{r} \ = 1$ • $\beta_0(\vec{K}) = 12$ • $\beta_1(\vec{K}) = 2$	c • $\vec{K}_3, \mathrm{r} = 2$ • $\beta_0(\vec{K}) = 1$ • $\beta_1(\vec{K}) = 1$	d • $\vec{K}_4, r = \infty$ • $\beta_0(\vec{K}) = 1$ • $\beta_1(\vec{K}) = 0$

Appendix Figure 7 A nested sequence of simplicial complex $\vec{K}$: (a) simplicial complex with parameter $r = 0$; (b) simplicial complex with parameter $r = 1$; (c) simplicial complex with parameter $r = 2$; (d) simplicial complex with parameter $r = \infty$

A nested sequence of simplicial complex $\vec{K}$: (a) simplicial complex with parameter $r = 0$; (b) simplicial complex with parameter $r = 1$; (c) simplicial complex with parameter $r = 2$; (d) simplicial complex with parameter $r = \infty$

Appendix A.4 Mapper

Mapper is based on topological ideas where a notion of closeness is preserved but large distances can be discarded. In comparison with the PH, where a series of nested subcomplexes were created and analyzed to derive topological information, Mapper is designed to produce a single low-dimensional simplicial complex, in a graph, from which information about the underlying data can be extracted (Singh et al., 2007). The input to the Mapper, similar to the PH, is a set of points P. More information about Mapper can be found in (Ristovska & Sekuloski, 2019; Singh et al., 2007). The computation of the Mapper consists of three steps (Singh et al., 2007):

1. Mapping an input set of points into a lower-dimensional space by implementing a function f, called '*filter*', that maps a set of points into a set of real numbers:

   $$f(P) = R \tag{A.12}$$

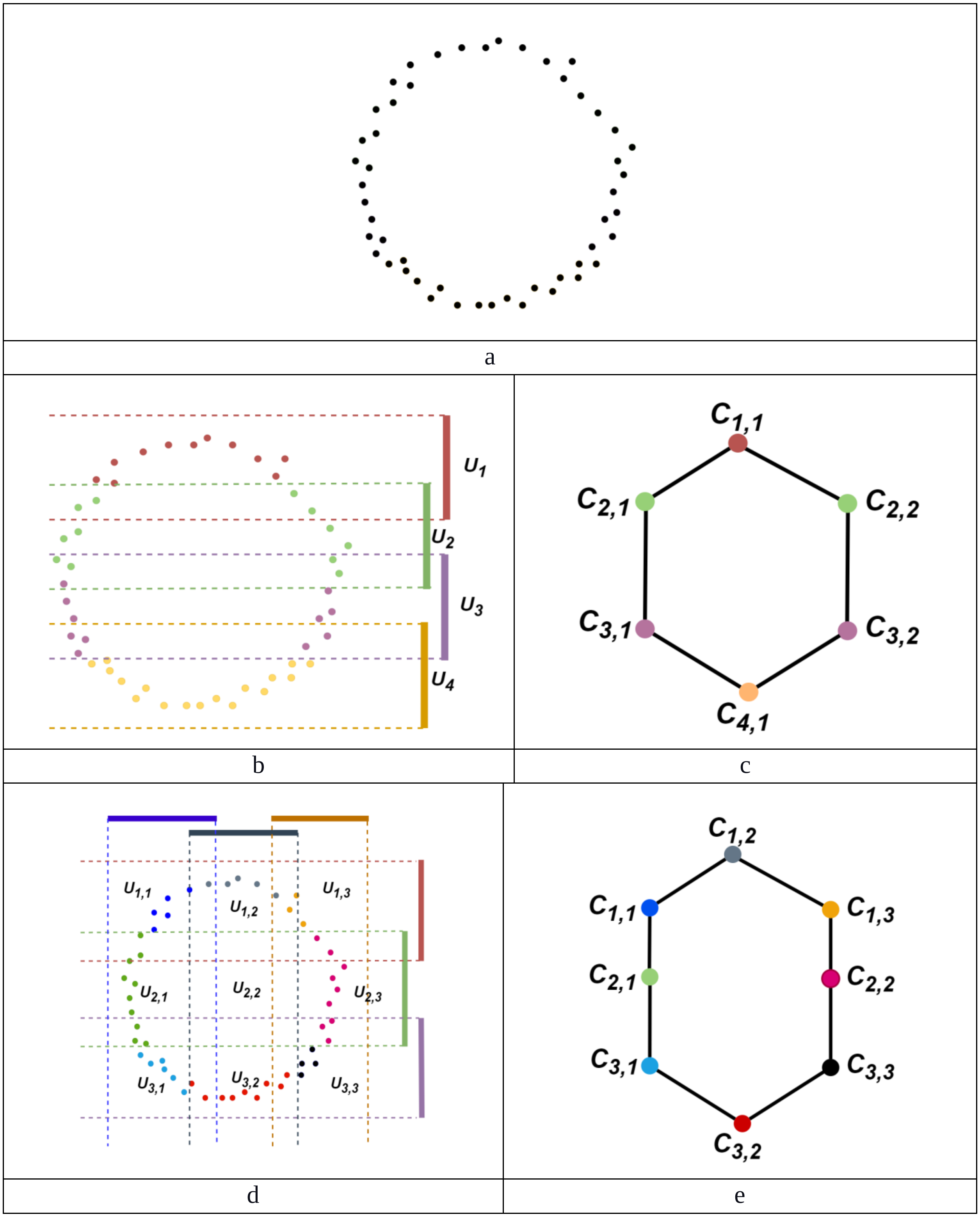
   If the higher dimensional dataset is given, the dimension needs to be reduced using dimensionality reduction algorithms such as Isomap, PCA, or t-SNE (Van Der Maaten, Postma, & Van den Herik, 2009).

2. Constructing a cover U of the space, in the form of a set of overlapping intervals that have same length. A cover U of $f(P)$ is an indexed set of open sets endowed with the standard topological rule such that:

   $$f(P) = \bigcup_{i \in I} U_i \tag{A.13}$$

   where I is the number of covers and the topological rule states that the union of all covers should contain all the points. In other words, after mapping a set of points into real numbers, the whole area needs to be covered with polygons where the union of all polygons should include all points (Appendix Figure 8). This requires the selection of two parameters: the dimension of the cover and the overlapping interval. For example, the set of points in Appendix Figure 8 (a) can be divided by using the overlapped blocks in Appendix Figure 8(b) or by using the overlapped squares in Appendix Figure 8 (d).

3. Selecting parameters and implementing a clustering method to cluster points inside each cover patch U_i. For clustering, the density-based spatial clustering of applications with noise (DBSCAN) algorithm can be used since the number of clusters inside each cover patch U_i is unknown (Schubert, Sander, Ester, Kriegel, & Xu, 2017). Once the points are clustered, each cluster is converted into nodes and an edge between each pair of nodes with overlapped points is created (Appendix Figure 8c, e).



Appendix Figure 8 Step-by-step computation of Mapper: (a) set of points P; (b) set of points in divided using overlapped blocks; (c) a graph, the output of the Mapper created by using overlapped blocks; (d) set of points in divided by using overlapped grids; (e) a graph,

Appendix B Additional Experiments

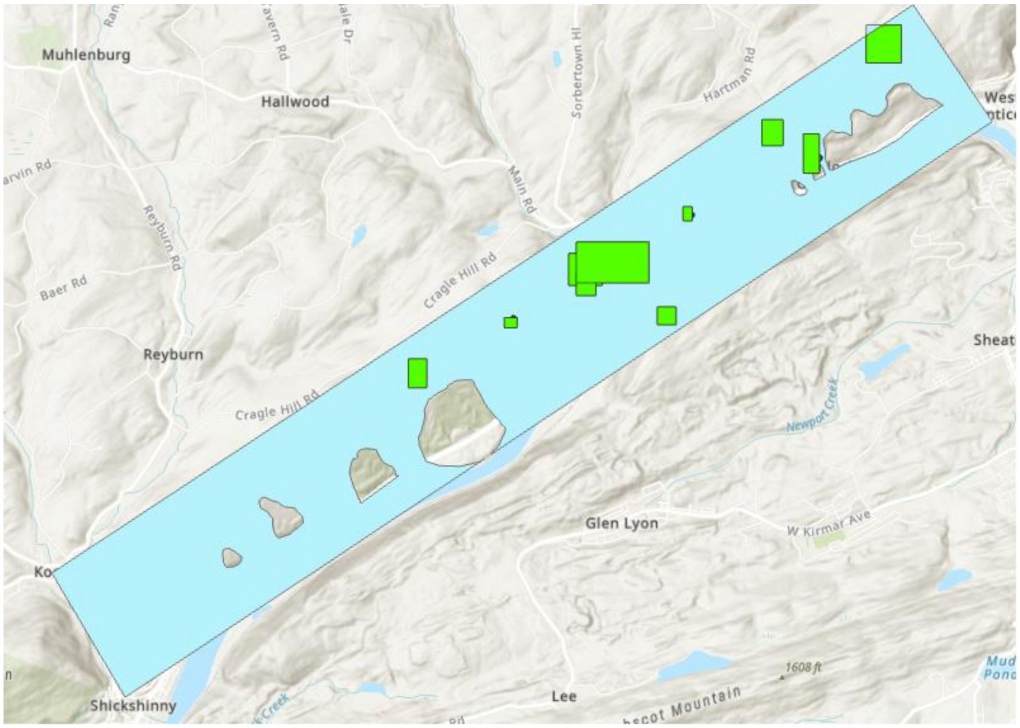
Appendix B.1 Topological KB Method: Candidate Polygon Selection via ML Models

In Chapter 6, we highlighted the feasibility of generating candidate polygons using the PH. This method aids in identifying geospatial objects from these polygons, leveraging embedded topological, geometrical, and contextual information. However, implementing this method necessitates a comprehensive set of rules and domain-specific knowledge about the objects being detected. In the context of this thesis, we focused on landslides as our target geospatial objects. Through an extensive literature review, we derived a specific set of rules. During our evaluation, the identified landslides were cross-referenced with a ground truth, which pinpointed the locations of existing landslides.

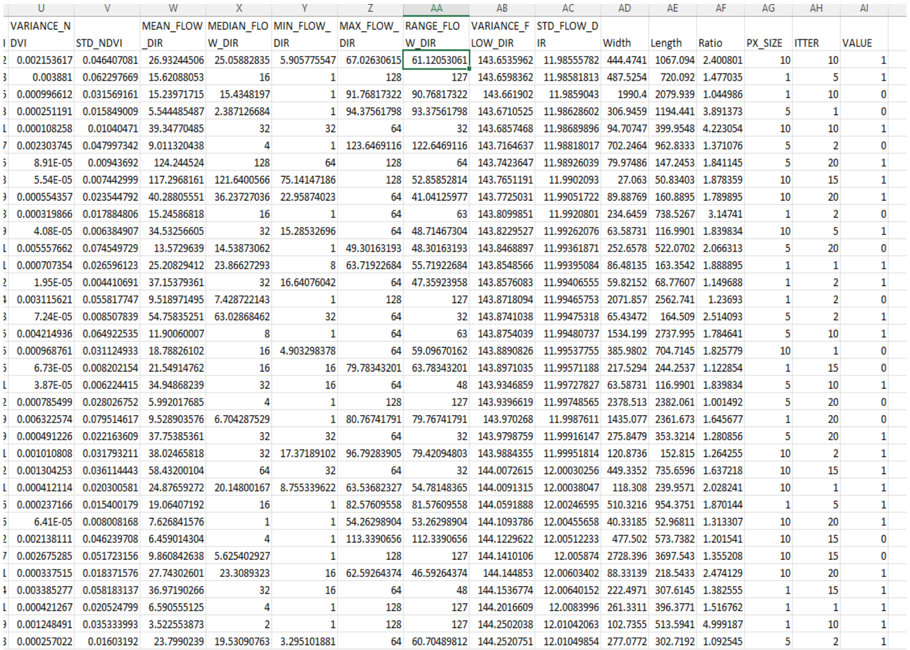
From our exploration in Chapter 6, we delved into the process of generating candidate polygons using PH and embedding geometrical and contextual information into each candidate polygon. This initiated further experimentation: we hypothesized the deployment of ML models in formulating detection rules, when applied to a set of candidate polygons enriched with embedded geometrical and contextual information and benchmarked against a recognized ground truth, will significantly improve the accuracy of geospatial object detection.

In other words, if we have candidate polygons derived from PH with embedded geometrical and contextual information and ground truth, can we convert these problems into ML classification problem and detect geospatial objects. We conducted experiments with the following steps:

1. Using the boundaries of the study area and ground truth polygons, we generated random polygons of varying sizes in regions void of landslides (Appendix Figure 9).
2. Geometrical and contextual information were embedded into these non-landslide polygons for each study area.
3. Similarly, we embedded geometrical and contextual information into the landslide polygons, using them as our ground truth.
4. Both sets of polygons were combined, and individual labels were assigned: '0' for non-landslide polygons and '1' for landslide polygons (Appendix Figure 10).
5. With the geometric and contextual data paired with their labels, we trained our ML model.
6. We employed the trained ML model to classify the candidate polygons, sourced from the PH and augmented with embedded geometrical and contextual information.
7. The experiment's outcomes were evaluated against the ground truth. We leveraged the same evaluation metrics that were introduced in Chapter 6 for consistency and precision.



Appendix Figure 9 Study Area 1 with Generation for Non-Landslide polygons



U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
VARIANCE_NDVI	STD_NDVI	MEAN_FLOW_DIR	MEDIAN_FLOW_DIR	MIN_FLOW_DIR	MAX_FLOW_DIR	RANGE_FLOW_DIR	VARIANCE_FLOW_DIR	STD_FLOW_DIR	Width	Length	Ratio	PX_SIZE	ITTER	VALUE
0.002153617	0.046407081	26.93244506	25.05882835	5.905775547	67.02630615	61.12053061	143.6535962	11.98555782	444.4741	1067.094	2.400801	10	10	1
0.003881	0.062297669	15.62088053	16	1	128	127	143.6598362	11.98581813	487.5254	720.092	1.477035	1	5	1
0.000996612	0.031569161	15.23971715	15.4348197	1	91.76817322	90.76817322	143.661902	11.9859043	1990.4	2079.939	1.044986	1	10	0
0.000251191	0.015849009	5.544485487	2.387126684	1	94.37561798	93.37561798	143.6710525	11.98628602	306.9459	1194.441	3.891373	5	1	0
0.000108258	0.01040471	39.34770485	32	32	64	32	143.6857468	11.98689896	94.70747	399.9548	4.223054	10	10	1
0.002303745	0.047997342	9.011320438	4	1	123.6469116	122.6469116	143.7164637	11.98818017	702.2464	962.8333	1.371076	5	2	0
8.91E-05	0.00943692	124.244524	128	64	128	64	143.7423647	11.98926039	79.97486	147.2453	1.841145	5	20	1
5.54E-05	0.007442999	117.2968161	121.6400566	75.14147186	128	52.85852814	143.7651191	11.9902093	27.063	50.83403	1.878359	10	15	1
0.000554357	0.023544792	40.28805551	36.23727036	22.95874023	64	41.04125977	143.7725031	11.99051722	89.88769	160.8895	1.789895	10	20	1
0.000319866	0.017884806	15.24586818	16	1	64	63	143.8099851	11.9920801	234.6459	738.5267	3.14741	1	2	0
4.08E-05	0.006384907	34.53256605	32	15.28532696	64	48.71467304	143.8229527	11.99262076	63.58731	116.9901	1.839834	10	5	1
0.005557662	0.074549729	13.5729639	14.53873062	1	49.30163193	48.30163193	143.8468897	11.99361871	252.6578	522.0702	2.066313	5	20	0
0.000707354	0.026596123	25.20829412	23.86627293	8	63.71922684	55.71922684	143.8548566	11.99395084	86.48135	163.3542	1.888895	1	1	1
1.95E-05	0.004410691	37.15379361	32	16.64076042	64	47.35923958	143.8576083	11.99406555	59.82152	68.77607	1.149688	1	2	1
0.003115621	0.055817747	9.518971495	7.428722143	1	128	127	143.8718094	11.99465753	2071.857	2562.741	1.23693	1	2	0
7.24E-05	0.008507839	54.75835251	63.02868462	32	64	32	143.8741038	11.99475318	65.43472	164.509	2.514093	5	2	1
0.004214936	0.064922535	11.90060007	8	1	64	63	143.8754039	11.99480737	1534.199	2737.995	1.784641	5	10	1
0.000968761	0.031124933	18.78826102	16	4.903298378	64	59.09670162	143.8890826	11.99537755	385.9802	704.7145	1.825779	10	1	0
6.73E-05	0.008202154	21.54914762	16	16	79.78343201	63.78343201	143.8971035	11.99571188	217.5294	244.2537	1.122854	1	15	0
3.87E-05	0.006224415	34.94868239	32	16	64	48	143.9346859	11.99727827	63.58731	116.9901	1.839834	5	10	1
0.000785499	0.028026752	5.992017685	4	1	128	127	143.9396619	11.99748565	2378.513	2382.061	1.001492	5	20	0
0.006322574	0.079514617	9.528903576	6.704287529	1	80.76741791	79.76741791	143.970268	11.9987611	1435.077	2361.673	1.645677	1	20	0
0.000491226	0.022163609	37.75385361	32	32	64	32	143.9798759	11.99916147	275.8479	353.3214	1.280856	5	20	1
0.001010808	0.031793211	38.02465818	32	17.37189102	96.79283905	79.42094803	143.9884355	11.99951814	120.8736	152.815	1.264255	10	2	1
0.001304253	0.036114443	58.43200104	64	32	64	32	144.0072615	12.00030256	449.3352	735.6596	1.637218	10	15	1
0.000412114	0.020300581	24.87659272	20.14800167	8.755339622	63.53682327	54.78148365	144.0091315	12.00038047	118.308	239.9571	2.028241	10	1	1
0.000237166	0.015400179	19.06407192	16	1	82.57609558	81.57609558	144.0591888	12.00246595	510.3216	954.3751	1.870144	1	5	1
6.41E-05	0.008008168	7.626841576	1	1	54.26298904	53.26298904	144.1093786	12.00455658	40.33185	52.96811	1.313307	10	20	1
0.002138111	0.046239708	6.459014304	4	1	113.3390656	112.3390656	144.1229622	12.00512233	477.502	573.7382	1.201541	10	15	0
0.002675285	0.051723156	9.860842638	5.625402927	1	128	127	144.1410106	12.005874	2728.396	3697.543	1.355208	10	15	0
0.000337515	0.018371576	27.74302601	23.3089323	16	62.59264374	46.59264374	144.144853	12.00603402	88.33139	218.5433	2.474129	10	20	1
0.003385277	0.058183137	36.97190266	32	16	64	48	144.1536774	12.00640152	222.4971	307.6145	1.382555	1	15	1
0.000421267	0.020524799	6.590555125	4	1	128	127	144.2016609	12.0083996	261.3311	396.3771	1.516762	1	1	1
0.001248491	0.035333993	3.522553873	2	1	128	127	144.2502038	12.01042063	102.7355	513.5941	4.999187	1	10	1
0.000257022	0.01603192	23.7990239	19.53090763	3.295101881	64	60.70489812	144.2520751	12.01049854	277.0772	302.7192	1.092545	5	2	1

Appendix Figure 10 Table with combined polygons and with assigned labels: '0' for non-landslide polygons and '1' for landslide polygons

Appendix Table 1 Results of the: (Syzdykbayev et al., 2020b) and Topological KB geospatial object detection method and ML models for each studied area

	Study Area 1	Study Area 2	Study Area 3	Study Area 4	Study Area 5
Title	Geospatial Object Detection: Topological KB Method (Chapter 6)				
F1 score	0.52	0.48	0.43	0.64	0.45
Title	Geospatial Object Detection: PH-Based Method (Syzdykbayev et al., 2020b) or (Chapter 4)				
F1 score	0.334	0.466	0.382	0.65	0.337
Title	Candidate Polygon Selection via ML Models (Appendix B)				
F1 score	0.44	0.41	0.36	0.59	0.42

The utilization of ML models in geospatial object detection demonstrated notable potential, although certain limitations were observed during the experimentation phase. The assessment of accuracy, specifically measured using the F1 score, highlighted both the strengths and limitations of the ML models (Table B.1). It is important to emphasize that the goal of these experiments was to explore the feasibility of deriving detection rules through ML models, leading to valuable insights for future research endeavours. In evaluating the accuracy outcomes, a comparison was drawn with two distinct geospatial object detection methods: a knowledge-based method utilizing only topological information, and a novel method combining topological, geometrical, and contextual information. This analysis aimed to understand the relative performance of the ML method across different study areas and detection methodologies.

Interestingly, the ML-based method exhibited favourable outcomes in certain contexts. Specifically, for Study Area 1 and Study Area 5, the ML-based method yielded superior results in terms of accuracy, as evidenced by the F1 score (Table B.1). These results align with the hypothesis that ML-based method can detect geospatial objects more accurately compared to PH-based methods that rely solely on topological information. Conversely, a different trend emerged when assessing the performance of the ML-based method across the other study areas (Table B.1). In these cases, the PH-based method utilizing topological information surpassed the ML-based method in terms of accuracy. This observation raises questions about the adaptability and generalizability of ML-based method to different geographical contexts. Furthermore, when compared to the Topological KB geospatial object detection method, which integrates both topological and additional information (geometrical and contextual), the ML-based method fell short in all instances (Table B.1). This suggests that while ML-based method offer promise, there is still room for refinement and innovation in terms of incorporating various types of data to achieve optimal geospatial object detection accuracy.

However, it is crucial to acknowledge the limitations that were encountered throughout the experiments. First, the labelling of landslide and non-landslide polygons presented challenges. The generation of non-landslide polygons in areas without confirmed ground truth might have introduced uncertainties (Appendix Figure 9). Additionally, the balanced number of non-landslide polygons generated for training, in contrast to the limited number of actual landslide polygons, could have influenced the model's ability to generalize effectively. Three distinct ML models were employed in the experiments: Logistic Regression, Decision Tree, and Random Forest. It is worth noting that Random Forest yielded the most promising results among these models.

In conclusion, this series of experiments provided valuable insights into the feasibility of utilizing ML-based method for deriving detection rules in geospatial object detection. While the ML-based method highlighted its potential in certain cases, the results underscore the importance of refining methodologies, optimizing model parameters, and adapting methods to diverse geographical settings. The limitations identified pave the way for future research to address these challenges, enabling the development of more robust and accurate geospatial object detection methods.

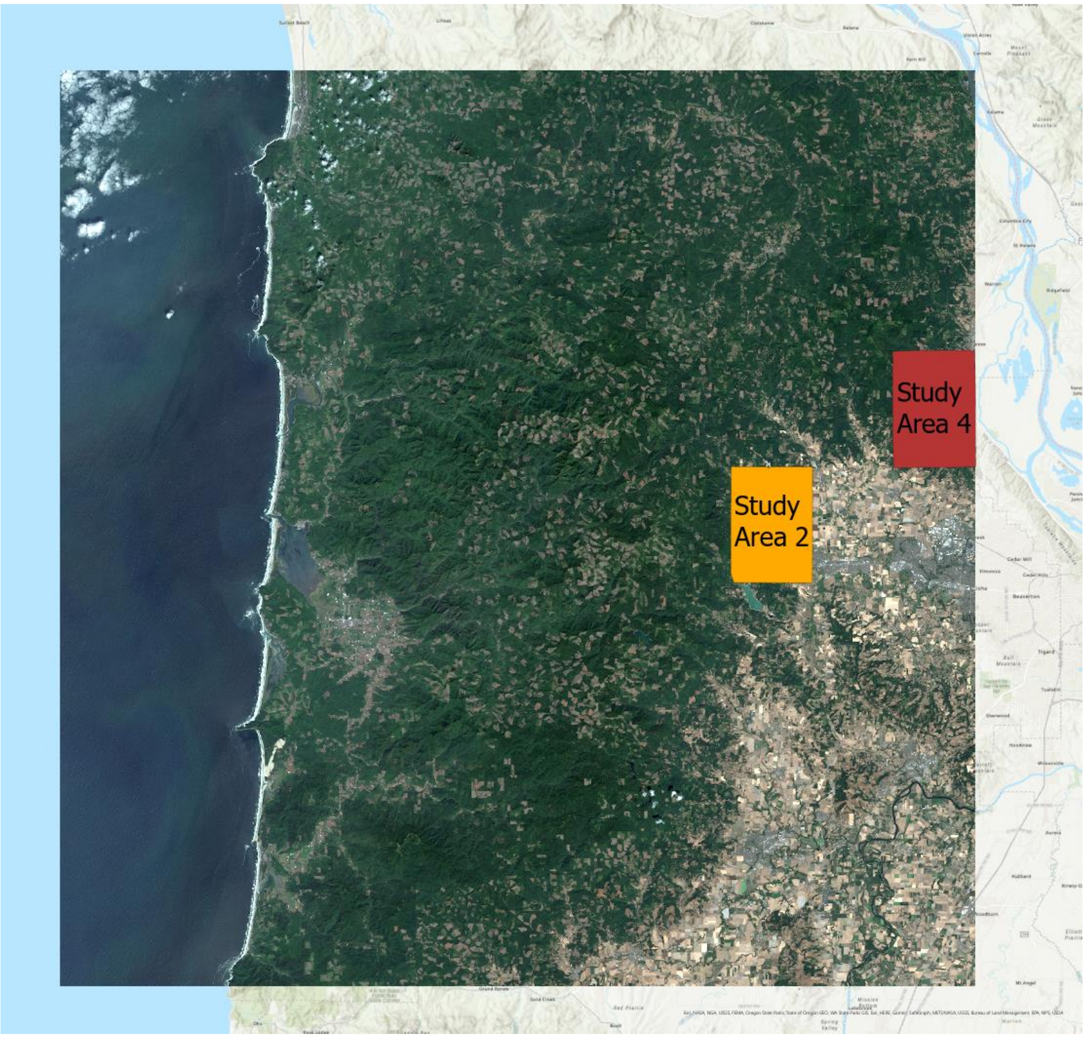
Appendix B.2 Detection of Geospatial Objects in 5 Study Areas Using Topological ML-Based Methods

In Chapter 7, our focus shifted towards exploring the potential improvement in the accuracy of geospatial object detection. This enhancement was pursued through the integration of multichannel images, effectively incorporating topological information along with the precise location of a subset of points that constitute this information. Our experiment involved training ML models using a dataset sourced from the LandSlide4Sense competition (LandSlide4Sense, n.d.). This dataset encompasses a diverse collection of images and associated labels derived from landslide-affected regions across the globe spanning from 2015 to 2021. The input images employed in this study consist of multichannel satellite imagery, comprising 14 channels.

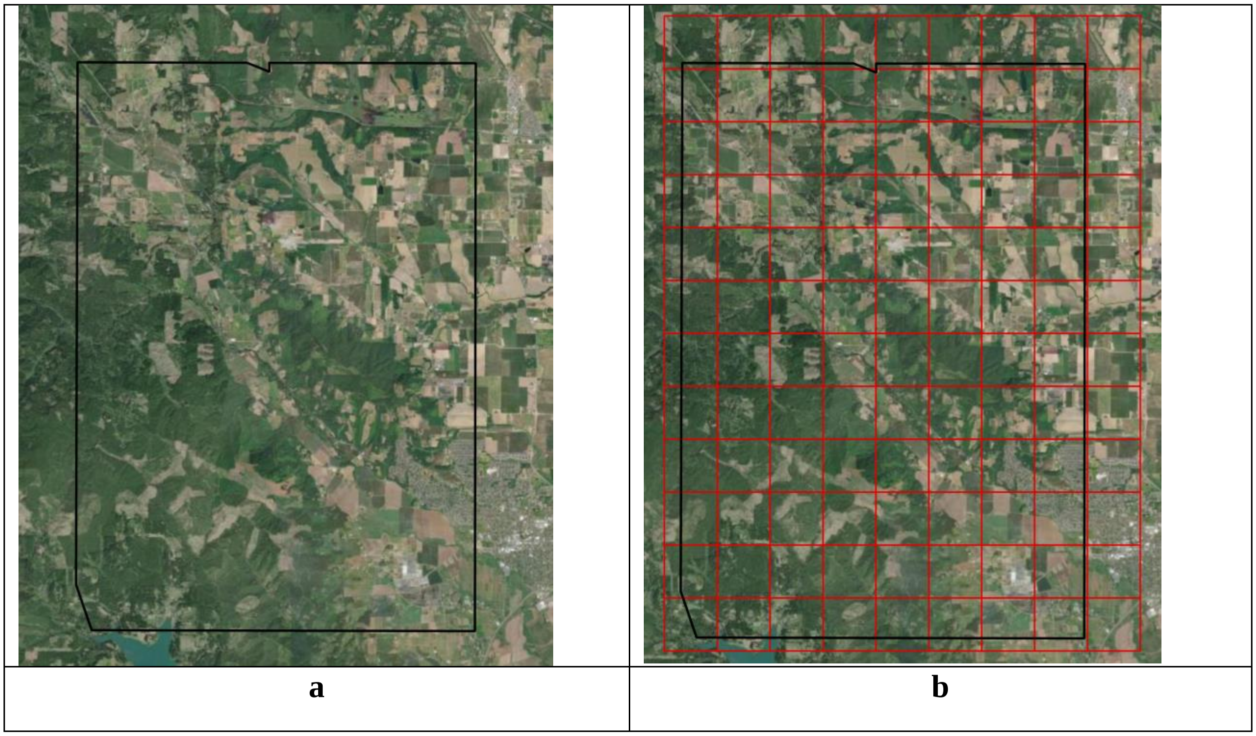
Building upon our existing work, we posed a new experiment idea to guide our investigation. With our focus on the dataset already preprocessed and models pretrained in Chapter 7 using the LandSlide4Sense dataset, we aimed to evaluate the accuracy of our Topological ML-based object detection approach. This evaluation involved a direct comparison between Topological KB geospatial object detection results, which were previously explored in Chapter 4 and Chapter 6. Hence, we hypothesized that ML models specifically trained on the LandSlide4Sense dataset can proficiently identify landslides within the identical five study regions previously scrutinized in Chapter 4 and Chapter 6, using Sentinel-2 data from the same origin.

Our objective revolved around determining the efficacy of the Topological ML-based approach in detecting landslides within study areas where geospatial objects had been detected using Topological KB methods. This comparison aimed to discern the relative strengths and limitations of the two distinct detection methodologies: the Topological ML-based method versus the Topological KB method. Our overarching goal was to gain a comprehensive understanding of how each method performs in varied scenarios. We conducted experiments with the following steps:

- Multispectral Sentinel-2 data for each of the study areas were obtained from the 'Copernicus Open Access Hub' (Appendix Figure 11).
- Slope and DEM data was derived from LiDAR point clouds that we already used in chapters 3 and 5.
- The pixel size was set to 10 meters per pixel, and all datasets were merged into a comprehensive 14-channel dataset (combining 12 Sentinel-2 channels, slope, and DEM data).
- The entire image was divided into smaller tiles, each comprising 128 x 128 pixels (Appendix Figure 12), to use the same pretrained models from Chapter 7.
- Building on the methodology from Chapter 7, topological information was derived and incorporated as an additional channel for each tile.
- Pretrained models from Chapter 7were utilized to predict landslide locations for each tile.
- The results were evaluated using the ground truth data applied in Chapter 4 and Chapter 6 (Appendix Figure 13).



Appendix Figure 11 Location of the Study Area 2 and Study Area 4 overlaid on Sentinel 2 RGB dataset



Appendix Figure 12 a) boundaries of Study Area 4; b) boundaries of Study Area 4 splatted into 128*128-pixel tiles

In general, the utilization of pretrained models across the five study areas from previous chapters yielded suboptimal results when compared to both the Topological KB methods presented in Chapters 4 and 6, as well as the ML-based method discussed in Chapter 7. The primary factors contributing to this diminished accuracy can be summarized as follows:

- In the PH- based and Topological KB methods described in Chapters 3 and 5, a range of pixel sizes, including 1m, 5m, and 10m, were employed as input data. It was observed that, in most cases, the best results were achieved when utilizing a 1m pixel size.
- The PH- based and Topological KB methods in Chapters 4 and 6 utilized DTM as input, which involved the removal of vegetation cover from the data. In contrast, Chapter 6 employed DEM for training ML models.

- All five study areas encompassed various man-made objects such as farmlands, residential properties, and roads. However, the input data used in the ML models did not consistently account for these objects, potentially impacting detection accuracy.
- The ML models in Chapter 7 were primarily designed to detect new landslides. In contrast, the PH- based and Topological KB methods in Chapters 4 and 6, as well as the ground truth data associated with them, included both new and historical landslide instances.

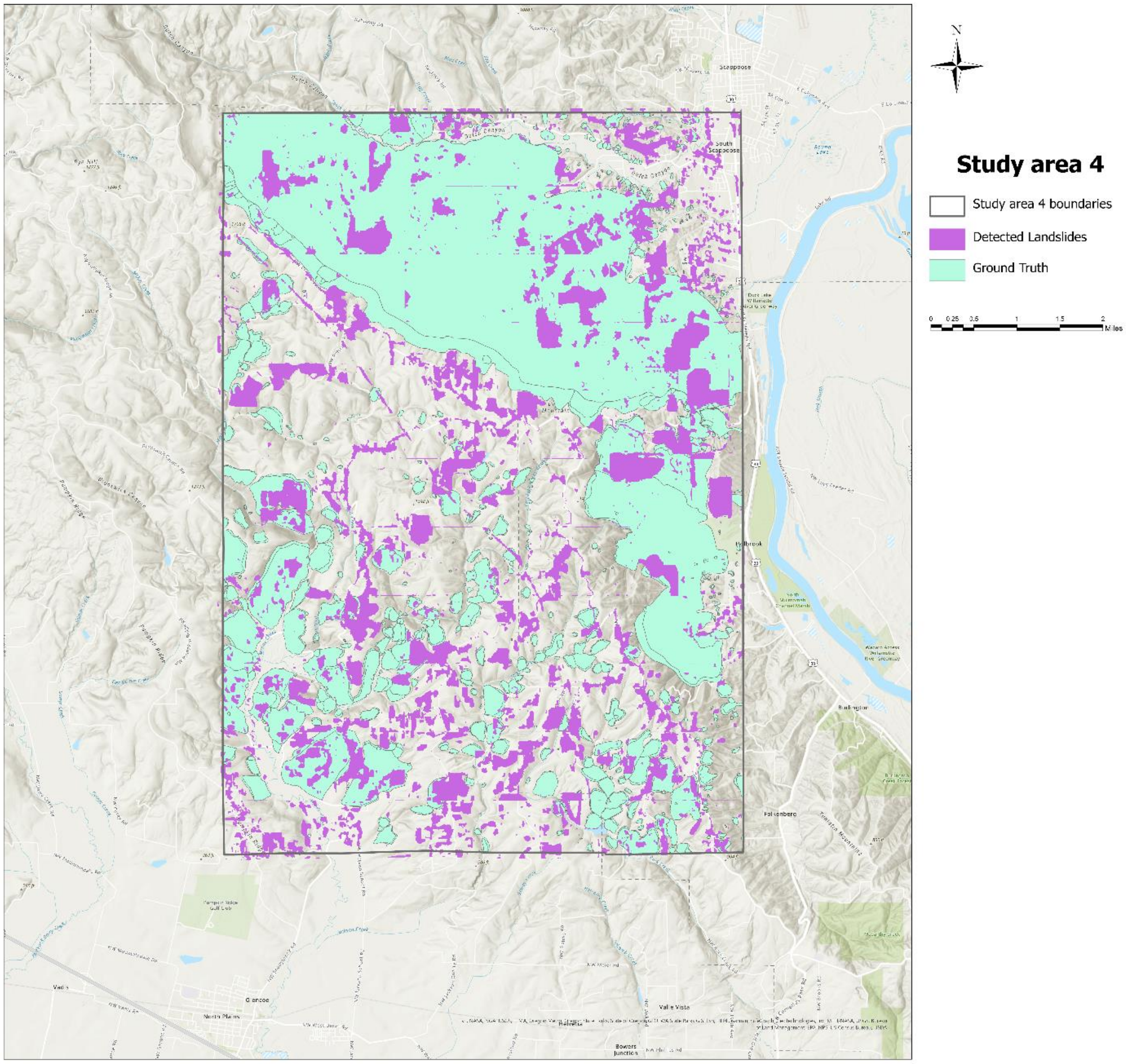
The analysis of our experiment results reveals some interesting findings. Specifically, the best performance among the pretrained models was achieved by the model that incorporated all channels, including topological layers, as evident in Table B.1 and Appendix Figure 13. However, upon closer examination of the detected images, we observed a peculiar behaviour in which the ML model classified areas of bare ground located on slopes as landslides, as illustrated in Appendix Figure 14. It's noteworthy that Study Area 4 contained numerous farmlands, which often consist of bare ground (as shown in Appendix Figure 12). This characteristic of the landscape may have influenced the model's predictions.

Furthermore, we encountered an unexpected behaviour when using a model that included a Laplace edge layer, generated using Laplacian filters (Table B.2). Surprisingly, the results were close to 0, implying that the pretrained models did not detect any landslides. To ensure the integrity of our analysis, we conducted a thorough examination of the input data, which was found to be identical to the data used in Chapter 7.

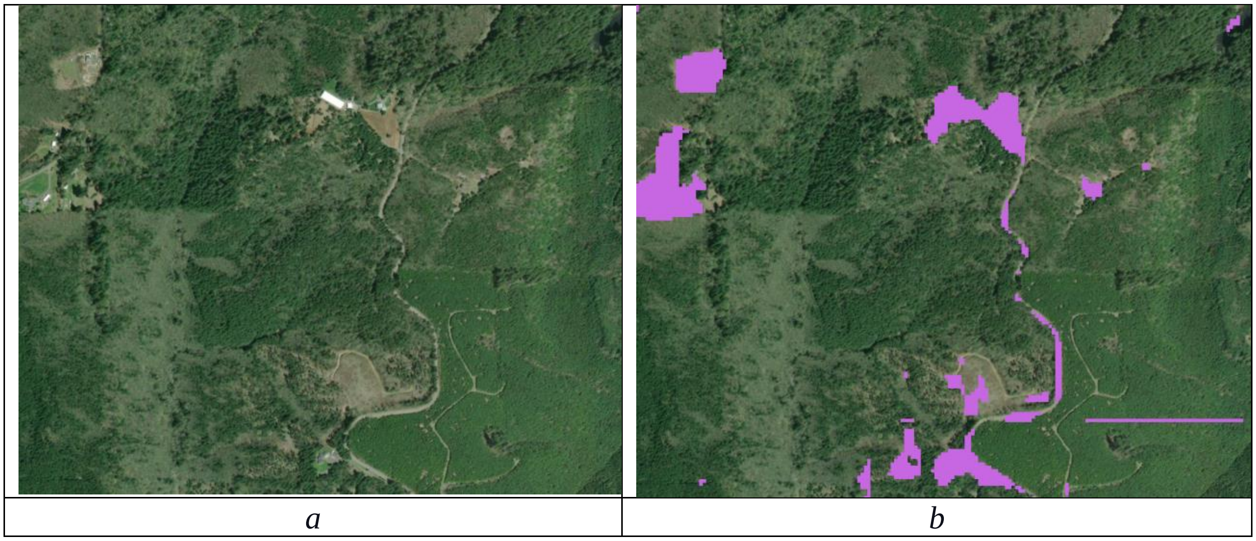
These observations highlight the complexity of geospatial object detection tasks, particularly when dealing with diverse and dynamic landscapes. The misclassification of ground without trees on slopes as landslides underscores the importance of refining model training and considering landscape-specific features. Additionally, the unexpected performance of the model with the Laplace edge layer suggests the need for further investigation into the impact of specific layers or filters on detection accuracy.

Appendix Table 2 Results: F1 score predicted landslides for each study area and for each input source combination

Channel Name	Number of Channels	F1 score: Study Area 1	F1 score: Study Area 2	F1 score: Study Area 3	F1 score: Study Area 4	F1 score: Study Area 5
RGB	3	0	0.01374	0.004276	0.002903	0.015358
RGB, Slope, DEM	5	0	0.042692	0.040054	0.06964	0.00235
ALL Channels	14	0.007004	0.045756	0.190421	0.248038	0.044559
RGB, + Edge	3	0	0	0	0	0
RGB, Slope, DEM+ Edge	6	0	0	0	0	0
ALL Channels +Edge	15	0	0	0	0	0
RGB + PH	7	0	0.032708	0.049151	0.102132	0.01797
RGB, Slope, DEM + PH	9	0.001578	0.037394	0.161145	0.115284	0.027007
ALL Channels+ PH	18	0.007303	0.026593	0.086071	0.18366	0.030836
RGB + Mapper	7	0	0.025089	0.007133	0.007617	0.016659
RGB, Slope, DEM + Mapper	9	0	0	0	0	0
ALL Channels + Mapper	18	0.023706	0.058016	0.164046	0.210576	0.047265
RGB + PH + Mapper	11	0	0.025775	0.010013	0.007808	0.009319
RGB, Slope, DEM+ PH + Mapper	13	0.022914	0.04247	0.157138	0.134353	0.029659
ALL Channels + PH + Mapper	22	0.025275	0.049643	0.137493	0.221471	0.047054
ALL Channels + PH + Mapper + Edge	23	0.01104	0.060779	0.163014	**0.257606**	0.047733



Appendix Figure 13 Study Area 4: Landslides Detected Using ML Models Trained on Datasets including ALL Channels, PH, Mapper, and Edge, Overlayed with Ground Truth



Appendix Figure 14 a) satellite image of Study Area 4; b) satellite image of Study Area 4 overlaid Landslides Detected Using ML Models Trained on Datasets including ALL Channels, PH, Mapper, and Edge

Appendix C Additional Validation Metrics for Chapter 7 Experiments

Appendix Table 3 Experimental configurations and Accuracy outcomes for encoder-decoder models after 300 Epochs

Encoder Name	Decoder Name	Loss function	Channel Name	Number of Channels	F1 score of Training set after 300 epochs	F1 score of Test set after 300 epochs
UNET	VGG11	Jaccard Loss	RGB	3	0.991	0.983
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM	5	0.990	0.986
UNET	VGG11	Jaccard Loss	ALL Channels	14	0.989	0.986
UNET	VGG11	Jaccard Loss	RGB, + Edge	3	0.991	0.983
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM+ Edge	6	0.991	0.985
UNET	VGG11	Jaccard Loss	ALL Channels +Edge	15	0.991	0.987
UNET	VGG11	Jaccard Loss	RGB + PH	7	0.990	0.982
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM + PH	9	0.991	0.985
UNET	VGG11	Jaccard Loss	ALL Channels+ PH	18	0.989	0.987
UNET	VGG11	Jaccard Loss	RGB + Mapper	7	0.992	0.982
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM + Mapper	9	0.990	0.985
UNET	VGG11	Jaccard Loss	ALL Channels + Mapper	18	0.991	0.987
UNET	VGG11	Jaccard Loss	RGB + PH + Mapper	11	0.992	0.982

UNET	VGG11	Jaccard Loss	RGB, Slope, DEM+ PH + Mapper	13	0.991	0.984
UNET	VGG11	Jaccard Loss	ALL Channels + PH + Mapper	22	0.991	0.987
UNET	VGG11	Jaccard Loss	ALL Channels + PH + Mapper + Edge	23	0.990	0.987

Appendix Table 4 Experimental configurations and precision outcomes for encoder-decoder models after 300 Epochs

Encoder Name	Decoder Name	Loss function	Channel Name	Number of Channels	F1 score of Training set after 300 epochs	F1 score of Test set after 300 epochs
UNET	VGG11	Jaccard Loss	RGB	3	0.819	0.651
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM	5	0.781	0.694
UNET	VGG11	Jaccard Loss	ALL Channels	14	0.749	0.708
UNET	VGG11	Jaccard Loss	RGB, + Edge	3	0.804	0.654
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM+ Edge	6	0.805	0.686
UNET	VGG11	Jaccard Loss	ALL Channels +Edge	15	0.796	0.712
UNET	VGG11	Jaccard Loss	RGB + PH	7	0.792	0.637
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM + PH	9	0.822	0.692
UNET	VGG11	Jaccard Loss	ALL Channels+ PH	18	0.768	0.719
UNET	VGG11	Jaccard Loss	RGB + Mapper	7	0.833	0.638
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM + Mapper	9	0.788	0.704

UNET	VGG11	Jaccard Loss	ALL Channels + Mapper	18	0.797	0.717
UNET	VGG11	Jaccard Loss	RGB + PH + Mapper	11	0.829	0.631
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM+ PH + Mapper	13	0.824	0.686
UNET	VGG11	Jaccard Loss	ALL Channels + PH + Mapper	22	0.799	0.733
UNET	VGG11	Jaccard Loss	ALL Channels + PH + Mapper + Edge	23	0.780	0.721

Appendix Table 5 Experimental configurations and recall outcomes for encoder-decoder models after 300 Epochs

Encoder Name	**Decoder Name**	**Loss function**	**Channel Name**	**Number of Channels**	**F1 score of Training set after 300 epochs**	**F1 score of Test set after 300 epochs**
UNET	VGG11	Jaccard Loss	RGB	3	0.792	0.608
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM	5	0.762	0.694
UNET	VGG11	Jaccard Loss	ALL Channels	14	0.752	0.728
UNET	VGG11	Jaccard Loss	RGB, + Edge	3	0.773	0.606
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM+ Edge	6	0.780	0.674
UNET	VGG11	Jaccard Loss	ALL Channels +Edge	15	0.790	0.723
UNET	VGG11	Jaccard Loss	RGB + PH	7	0.768	0.611
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM + PH	9	0.788	0.645
UNET	VGG11	Jaccard Loss	ALL Channels+ PH	18	0.766	0.717

UNET	VGG11	Jaccard Loss	RGB + Mapper	7	0.794	0.571
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM + Mapper	9	0.762	0.667
UNET	VGG11	Jaccard Loss	ALL Channels + Mapper	18	0.784	0.719
UNET	VGG11	Jaccard Loss	RGB + PH + Mapper	11	0.796	0.581
UNET	VGG11	Jaccard Loss	RGB, Slope, DEM+ PH + Mapper	13	0.791	0.643
UNET	VGG11	Jaccard Loss	ALL Channels + PH + Mapper	22	0.789	0.711
UNET	VGG11	Jaccard Loss	ALL Channels + PH + Mapper + Edge	23	0.771	0.713