Glacier Segmentation From Remote Sensing Imagery Using Deep Learning

Bibek Aryal
University of Texas at El Paso

Follow this and additional works at: https://scholarworks.utep.edu/open_etd

Part of the Computer Sciences Commons, Environmental Sciences Commons, Geology Commons, and the Remote Sensing Commons

Recommended Citation
https://scholarworks.utep.edu/open_etd/3645

This is brought to you for free and open access by ScholarWorks@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of ScholarWorks@UTEP. For more information, please contact lweber@utep.edu.
GLACIER SEGMENTATION FROM REMOTE SENSING IMAGERY

USING DEEP LEARNING

BIBEK ARYAL

Doctoral Program in Computational Science

APPROVED:

______________________________
Olac Fuentes, Ph.D., Chair

______________________________
Craig Tweedie, Ph.D., Co-Chair

______________________________
José M. Hurtado, Jr., Ph.D.

______________________________
Marianne Karplus, Ph.D.

______________________________
Anthony Ortiz, Ph.D.

______________________________
Stephen L. Crites, Ph.D.
Dean of the Graduate School
GLACIER SEGMENTATION FROM REMOTE SENSING IMAGERY

USING DEEP LEARNING

by

BIBEK ARYAL, M.S.

DISSERTATION
Presented to the Faculty of the Graduate School of
The University of Texas at El Paso
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Computational Science Program
THE UNIVERSITY OF TEXAS AT EL PASO
December 2022
Acknowledgements

To say this dissertation is “by Bibek Aryal” overstates the case. Without the significant contributions made by other people, this dissertation would certainly not exist.

At the top of the list is my advisor, Dr. Olac Fuentes. I consider myself fortunate to have had the opportunity to learn from him and would like to express my deep gratitude for his continuous support of my research. I should also thank my co-advisor, Dr. Craig Tweedie for all his support during this journey. Thank you for giving me the freedom to nurture my ideas and for enthusiastic encouragement to implement them. I would also like to thank members of my committee, Dr. José M. Hurtado, Jr., Dr. Marianne Karplus, and Dr. Anthony Ortiz for their time and effort. Their feedback, support, and valuable guidance were fundamental to the completion of this work.

Throughout this academic journey, I got the opportunity to collaborate with some brilliant and wonderful people. I would like to use this space to thank Dr. Kris Sankaran for introducing me to this problem space, Dr. Sergio A. Vargas Zesati for his invaluable assistance and support, and Dr. Katie Miles for her guidance and insightful conversations. I would like to give special thanks to Sebastian de la Chica for his mentorship during my two internships with Microsoft and shape me to be the person I am today.

Additionally, I want to thank The University of Texas at El Paso professors and staff for all their hard work and dedication, providing me with the means to complete my degree. I would also like to extend my gratitude to my colleagues from the Vision and Learning Lab and Systems Ecology Lab who have been directly or indirectly involved in my learning process. I wish to acknowledge the support and love of my mother, Chandrakala Aryal, and my sister, Bijaya Aryal, who have always been there for me whenever I needed them. Finally, I am grateful to all my friends, especially Kamal Nyaupane, Keshav Bhandari, and Laxman Bokati for making me feel at home every day, despite being more than 8,200 miles away from my original home in Nepal.
Abstract

Large-scale study of glaciers improves our understanding of global glacier change and is imperative for monitoring the ecological environment, preventing disasters, and studying the effects of global climate change. In recent years, remote sensing imagery has been preferred over riskier and resource-intensive field visits for tracking landscape level changes like glaciers. However, periodic manual labeling of glaciers over a large area is not feasible due to the considerable amount of time it requires while automatic segmentation of glaciers has its own set of challenges. Our work aims to study the challenges associated with segmentation of glaciers from remote sensing imagery using machine learning, improve on the performance of existing methods using deep learning techniques, and interpret the working mechanism of these deep learning models.

In this dissertation, several machine learning and deep learning techniques were used to delineate glaciers from Landsat-7 imagery and we observed that the U-Net based model outperformed the other methods. While the methods used in this research are generalizable across all alpine glaciers, we evaluate our performance on the glaciers in the Hindu Kush Himalayas (HKH) as the HKH is one of the world’s most sensitive region for climate change. Alpine glaciers, as the ones seen in HKH, have clean ice/snow surface where they form. As these Clean Glacial Ice (CIG) move down the valleys, they sometimes gather a significant covering of dirt, rocks, and boulders on their surface and are known as Debris Glacial Ice (DGI). As expected, our experimental results verify that segmenting DGI is significantly harder than segmenting CIG. To improve the performance on DGI, we introduce a novel Self-Learning Boundary-Aware loss ($\mathcal{L}_{SLBA}$). $\mathcal{L}_{SLBA}$ combines masked dice loss and boundary loss to simultaneously learn multiple objectives during the training process for improved performance. Experimentally, we show that $\mathcal{L}_{SLBA}$ outperforms the commonly used dice loss for DGI mapping. We also propose feature-wise saliency scores to quantify the contributions of each channel in the input image towards the final label. This
can help identify which features are most important in the context of glacier mapping and has the potential to change the way people view glaciers and the features associated with them, leading to a better understanding in monitoring them.

A limitation of U-Net based model is the need for densely labeled training data where a label is assigned to every pixel within the image. Due to the time and effort associated with creating dense labels, having access to training data remains a limiting factor in many applications of landcover mapping. To solve this problem, we introduced a technique to train the U-Net based model using sparsely labelled data where only some pixels for a given image are labelled. Unlike dense labels, sparsely labelled samples can be collected in large numbers in a relatively fast and cheap manner and often without the need for an expert. We used the technique to segment water bodies in the Arctic National Wildlife Refuge (ANWR) using the 4 Band Orthorectified NOAA Airborne Imagery and sparse training labels. This also shows that the methods used in this research can be used to segment geomorphological features on the Earth’s surface other than glaciers and across different types of satellite and overhead imagery for mapping.

In recent years, large volumes of public freely-available large-scale satellite images have been made available. However, there exists a knowledge gap on accurate pixel level understanding of what is going on in these images. In this dissertation, we present multiple methods for segmentation of geomorphological landscape features from overhead imagery. These methods can have major implications in understanding global challenges such as climate change and anthropogenic impacts to ecosystems (i.e. deforestation, urbanization, land use change).
# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table of Contents</strong></td>
<td>vi</td>
</tr>
<tr>
<td><strong>List of Tables</strong></td>
<td>ix</td>
</tr>
<tr>
<td><strong>List of Figures</strong></td>
<td>xi</td>
</tr>
<tr>
<td><strong>Chapter</strong></td>
<td></td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Thesis Contributions</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Thesis Structure</td>
<td>5</td>
</tr>
<tr>
<td>2 Deep Learning for Glacier Segmentation</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Glaciers</td>
<td>7</td>
</tr>
<tr>
<td>2.1.1 Clean Glacial Ice</td>
<td>8</td>
</tr>
<tr>
<td>2.1.2 Debris Glacial Ice</td>
<td>9</td>
</tr>
<tr>
<td>2.1.3 Glacier Mapping in the HKH</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Remote Sensing</td>
<td>10</td>
</tr>
<tr>
<td>2.2.1 Landsat 7 Satellite</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Semantic Segmentation</td>
<td>13</td>
</tr>
<tr>
<td>2.3.1 Metrics to Evaluate Semantic Segmentation Model</td>
<td>14</td>
</tr>
<tr>
<td>2.4 Convolutional Neural Network (CNN)</td>
<td>15</td>
</tr>
<tr>
<td>2.4.1 Fully Convolutional Network (FCN)</td>
<td>17</td>
</tr>
<tr>
<td>2.4.2 U-Net</td>
<td>17</td>
</tr>
<tr>
<td>2.5 Machine Learning Models</td>
<td>19</td>
</tr>
<tr>
<td>2.6 Deep Learning for Glacier Segmentation and Current Challenges</td>
<td>22</td>
</tr>
<tr>
<td>3 Machine Learning for Glacier Monitoring in the Hindu Kush Himalaya</td>
<td>24</td>
</tr>
<tr>
<td>3.1 Data Description</td>
<td>25</td>
</tr>
<tr>
<td>3.1.1 Understanding the Label Data</td>
<td>25</td>
</tr>
</tbody>
</table>
5.3 Chapter Conclusions ............................................................... 66
6 Conclusion and Future Work ...................................................... 69
References .............................................................................. 72
7 Curriculum Vitae ..................................................................... 87
## List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Landsat 7 Band Descriptions</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Calculation of Mean IoU</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>The exact numbers used in Figure 3.7</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>A comparison of error rates on clean ice and debris-covered glaciers across three modeling approaches. The first row is a model trained to predict glacier or background, without distinguishing between debris-covered or ice glaciers. The second row is a multiclass model trained to simultaneously segment debris-covered and clean ice glacier. The final row gives the result of training two separate models to distinguish each type of glacier. Results are comparable across approaches, with a slight edge for the split training approach.</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Data Distribution for Pixel Based Segmentation</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>Machine Learning Models Evaluation Metrics</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>Comparison Between Different Models</td>
<td>36</td>
</tr>
<tr>
<td>3.6</td>
<td>Performance Comparison for Clean Glacial Ice (CIG) and Debris Covered Glacial Ice (DCG)</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>Threshold selection using exhaustive search DS.</td>
<td>52</td>
</tr>
<tr>
<td>4.2</td>
<td>Experimental results for binary segmentation masks generated using different methods. The IoU when using random forest is higher than when using all other methods for both land and water classes.</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>Experimental results for coastal subregions using different methods.</td>
<td>55</td>
</tr>
<tr>
<td>5.1</td>
<td>Landsat 7 bands description</td>
<td>60</td>
</tr>
</tbody>
</table>
5.2 Labels Distribution - Random Sampling ........................................... 61
5.3 Results showing performance for combined loss and self-learning boundary-aware loss ............................................................... 63
6.1 Intersection over Union (IoU) of existing glacier labels for HKH ............ 70
List of Figures

1.1 Where is Earth’s water? [97] ............................................................... 1

2.1 Clean Ice & Debris Glaciers in Satellite Images ................................. 7
2.2 Khumbu Glacier ............................................................................. 8
2.3 Fixing Scan Line Corrector Failure .................................................. 12
2.4 Applications of Computer Vision [42] ............................................. 13
2.5 Sample Prediction ........................................................................ 14
2.6 Intersection-Over-Union ............................................................... 15
2.8 U-Net Architecture [93] ............................................................... 18
2.9 Comparison of U-Net Architecture with Other Methods [76] ............ 20

3.1 Data in Geographical Context .......................................................... 25
3.2 Our methodological pipeline first converts LE7 tiles to preprocessed patches used for model training. The trained model is deployed as an interactive web tool ................................................................. 27
3.3 Feature importance scores using Random Forest. Slope and elevation are key variables .............................................................. 28
3.4 Experimental results for channel selection. The $x$-axis indicates which LE7 bands were used. Color describes whether elevation and slope were used. Runs using Normalized Difference Water Index (NDWI), Normalized Difference Snow Index (NDSI), and Normalized Difference Vegetation Index (NDVI) are labeled with triangles. Elevation and slope data significantly boost performance, and using all bands is better than using any subset. Results when using RF features are enclosed in a square. .............................. 29
3.5 Example imagery and labels (a - c), with model predictions (d - f). Blue and orange labels are for clean ice and debris-covered glaciers, respectively. (d) A model trained to recognize the union of clean ice or debris-covered glaciers fails to recognize major debris-covered glaciers. (e - f) Multiclass and combined binary models give comparable predictions.

3.6 Slope and elevation are critical for improving precision of glacier predictions. Light blue labels represent both clean ice and debris-covered glaciers. Light yellow and green are areas of high elevation (e) and slope (f). Clean ice glaciers tend to be found at high elevation, while debris-covered glaciers are found in valleys. Neither type of glacier is found on steep slopes.

3.7 Effect of debris percentage on IoU. The multiclass model performs well on areas with high density of debris-covered glaciers. The binary model trained to distinguish any type of glacier from background suffers in these regions. When making no distinguish between glacier types, the model only learns to recognize clean-ice glaciers.

3.8 The image on the left shows the polygonized prediction for an area of interest. The image to the right shows the tool’s functionality of allowing users to correct predictions.

4.1 Our proposed method learns from sparsely labelled data for land/water segmentation. (a) Sample Image; (b) Corresponding Dense Labels; (c) Corresponding Sparse Labels.

4.2 Overview map showing the ANWR region on the eastern North Slope of Alaska, the extent of the 2017 NOAA airborne image collections (in blue) and the individual image tiles (in orange) from this collection used for creation of dense testing labels. The top right corner displays an example high resolution airborne image of a section of coastline acquired by the NOAA RSD airborne imaging.
4.3 Natural color (RGB) image tiles selected from 2017 NOAA airborne imagery taken over the study area displaying the spectral variability of coastal and surface water bodies: (a) lagoonal waters bound by tundra and a sand spit; (b) dark, turbid coastal waters near the mouth of a river; (c) waters of a braided river carrying highly reflective glacial-derived sediments from the Brooks Range to the south of the study area; (d) coastal waters breaking near barrier islands; (e) dark and deep waters of a tundra pond; (f) shallow and turbid waters near a deltaic region; (g) blue waters of a large thermokarst lake; (h) sediment dispersion from coastal erosion in the nearshore; (i) shallow waters of a coalesced low-center polygonal pond.

4.4 Natural color (RGB) image tiles selected from 2017 NOAA airborne imagery taken over the study area displaying the spectral variability of land surfaces: (a) coastal foredune features; (b) alluvial deposits; (c) polygonized tundra; (d) drained thaw lake; (e) bare tundra; (f) wet and dry sands along a coastal spit.

4.5 Our methodological pipeline takes in airborne imagery and its corresponding sparsely labelled shapefile to prepare subregions for training and evaluation.

4.6 Output intensity on a sample image from the test set using different models. The intensity has been normalized to 0–1 range for NDWI and NDSWI for plotting. The intensity for U-Net, Random Forest, and XGBoost represents the probability of the corresponding pixel to be classified as water.

4.7 Human annotated dense label for the image in Figure 4.6. The intensity masks are converted to respective binary masks using thresholds as seen in Section 4.2.3 for evaluation.

4.8 Binary mask generation by thresholding.

5.1 a) Spatially non-overlapping regions using fishnet grid. b) A zoomed image of one of the cells showing clean ice and debris glacier labels.
5.2 (a) Sample sub-image, (b) Corresponding Clean Glacial Ice, Debris Glacial Ice, Background, and Masked labels

5.3 Input preprocessing

5.4 Our modified U-Net architecture has 32 feature maps in the first convolution layer. We also introduce Batch normalization and spatial dropout in the modified architecture.

5.5 Masked Dice Loss weights and Boundary Loss weights vs. epoch for DCGs

5.6 Average saliency scores for all sub-images in training set.

5.7 (a) Sample subimage from test set (b) Corresponding CIG and DCG ground truth labels (c) True positive (TP), False positive (FP), False negative (FN) for CIG (IoU 79.17%) (d) TP, TP, FN for DCG (IoU 59.19%)
Chapter 1

Introduction

While approximately 71% of Earth’s surface is covered by water, only about 2.5% of the Earth’s water is fresh and about 69% of it is stored in glaciers and ice caps (Figure 1.1). As a result, the importance of glaciers cannot be understated. Understanding the changes in glaciers is critical in order to develop infrastructures that improve the well-being and livelihoods of communities that rely on river run-offs from these glaciers, to understand the impact of global climate change on the glacier ecosystem, and to empower critical and urgent decision-making during regional humanitarian crises. However, understanding and tracking these landscape level changes can be costly, time-consuming, and unsafe as glaciers are usually situated in remote areas that can be difficult to navigate through and map using traditional approaches, such as field visits and vertical air photography.

Figure 1.1: Where is Earth’s water?
In recent years, petabyte-scale archives of remote sensing data have become freely available from multiple U.S. Government agencies as well as from the European Space Agency. These satellite remote sensing data can be helpful in this scenario for periodical analysis of Earth’s surface over large areas. But periodic manual labeling of remote sensing images over a large area is practically impossible as it requires a considerable time investment. The experts who are in charge of labeling have to look at petabytes of high-resolution satellite images and manually label features of interest within each scene to corresponding labels or classes. The process of classifying the object class for each pixel within an image is a widely studied problem in computer vision under the term “Semantic Segmentation”. With the advent of deep learning, there have been advancements in semantic segmentation that may allow researchers to automate glacier segmentation. Landcover mapping (i.e. detecting and labeling geological features in satellite imagery) is one example application of semantic segmentation. There are multiple methods that have been developed over the years for semantic segmentation and, in recent years, semantic segmentation techniques have improved due to rapid progress in the field of deep learning, particularly, Convolutional Neural Networks (CNNs) [68, 93]. However, unlike segmentation for general images, the results when it comes to glacier segmentation are not very good, particularly for DCG.

While the methods presented in this dissertation will generalize across all alpine glaciers, we evaluate the model’s performance on the glaciers in the Hindu Kush Himalayas (HKH). The HKH is one of the world’s most sensitive regions for climate change. Often referred to as the third pole [35], the HKH holds approximately 14.5% of the world’s total glaciers, the most outside the polar regions. These glaciers are a source of many of Asia’s major river systems and are of high economic and social importance to a population of around 2 billion people [4]. The river runoffs from melting snow and glaciers is used as a source of drinking water, for agriculture, and for power generation in densely populated regions in South and Central Asia [13, 22]. Climate change poses a risk to the glaciers and its impact in the Himalayas can already be seen [16, 29]. The changes in glaciers due to climate change may have a significant impact on the quantity and timing of water availability downstream [87].
and could also increase the risk of natural hazards, ultimately affecting human livelihood. Efficient and timely monitoring of the state of glaciers is key for water resource and hazard management in the region.

In this thesis we aim to develop new machine learning methods for semantic segmentation of glaciers from remote sensing imagery and address the challenges associated with it. In particular, we focus on tackling the following four challenges:

- **Performance issues**: Even with the recent advancements in the field of deep learning, automatic segmentation of glaciers is a difficult problem to solve. To tackle this problem, we developed a methodology to select, combine, and generate features from multiple satellites for efficient feature engineering. We show that additional features from other remote sensing instruments such as elevation, slope, and derived spectral indices, when appended to the training data, can improve the performance of the glacier segmentation model (see Chapter 3). Furthermore, we compared the performances of various existing machine learning techniques and found that the U-Net based model outperforms the traditional methods. We then proceed to improve the performance of the U-Net model by incorporating the information from glacier boundaries into the training process (see Chapter 5).

- **Cost associated with collecting labelled training data**: While a large amount of remote sensing data is available for free, having access to labelled data remains a limiting factor. Supervised learning methods, such as the ones used for this research, require labelled data to extract the underlying information. However, generating pixel level annotations for training can require a considerable amount of time, effort, and often skilled annotators with a good understanding of the region of interest/labelled classes. To address this issue, we modify the U-Net model to learn from sparsely labeled data where only some pixels for a given image are labelled (see Chapter 4). As sparsely labelled data can be collected for a large number of images in a relatively fast and cheap manner and often without the need for an expert, this can be used to
reduce the cost associated with collecting labelled training data for the deep learning methods.

- **Generalization and transferability:** We show that the methods used in this research can accept all or selected bands from multispectral imagery. We show that this method can also accept Very High-spatial Resolution (VHR) (<1 m) 4-band NOAA imagery. This shows that the method proposed in this research is generalizable across various kinds of remote sensing imagery regardless of the spectral information or spatial resolution. Furthermore, we use the same methods to segment water bodies and show that these methods can be used to track landcover features from remote sensing imagery other than glaciers as well (see Chapter 4).

- **Explainable deep learning model:** While deep learning models have been shown to perform better on various tasks involving computer vision, the interpretability of these models is limited. Deep Neural Networks are often considered black boxes since their decision rules can not be described easily. The development of transparent, understandable, and explainable models is imperative for the wide-scale adoption of deep learning models. We observe what features are important for segmenting glaciers by performing selective data imputation (see Chapter 3). We also propose a method to quantify the contribution of each channel in the input image towards the final label and use it to identify what features are important for segmenting both CIG and DCG separately (see Chapter 5).

### 1.1 Thesis Contributions

The contributions of this research are the following:

1. Our work was one of the first to adapt the state-of-the-art U-Net model for large-scale glacier segmentation in the Hindu Kush Himalayas [12].
2. We modify the U-Net model to accept low-cost sparse labels as input during training and show that the methods used for glacier segmentation are generalizable across other landcover mapping tasks [8].

3. We introduce a novel self-learning boundary-aware loss for the U-Net model to improve glacier segmentation from satellite imagery.

4. We propose feature-wise saliency scores to quantify the contributions of each channel in the input image towards the final label. We also combine remote sensing images from multiple sources and train with different band combinations to understand what features are most important for the segmentation of glaciers [12].

1.2 Thesis Structure

The organization of the remainder of this document is as follows. Chapter 2 provides background information related to this project along with related work in segmentation of geographical features from satellite imagery using machine learning approaches. We combine features from multiple remote sensing sources and compare the performances of various machine learning and deep learning methods for segmenting HKH glaciers in Chapter 3. Chapter 4 focuses on showing the generalization and transferability of remote sensing image-based models by applying the same techniques to segment water bodies from very high spatial resolution airborne imagery. We also present methods to train the deep learning model with low-cost sparsely labeled data in Chapter 4. In Chapter 5 we compare the model performance on different types of alpine glaciers and present methods to incorporate boundary information during the training process to improve glacier segmentation. Chapter 6 presents conclusions, discussion, and future research directions.
Chapter 2

Deep Learning for Glacier Segmentation

Glacier delineation using remote sensing imagery has seen a growing use of deep learning in recent years [38, 50, 12, 109, 102]. This can be attributed to factors such as the availability of large-scale remote sensing data from multiple sources, the development of state-of-the-art deep learning architectures for image analysis, and the growing interest due to the impacts of climate change on glaciers in recent decades.

The Himalayas continue to be interesting to glaciologists, which is not surprising when one considers the formidable water-resource problems in the glacier’s regional context and the rate at which the ice is disappearing [64, 65]. With approximately 14.5% of the world’s total glaciers, the HKH holds the highest concentration of snow and glaciers outside the polar region and is often referred to as the third pole [35]. Glaciers of the HKH feed into 10 major river basins that serve as drinking water, agriculture, and hydropower supply for more than 1.9 billion people in South and Central Asia [13, 22]. The Himalaya is one of the world’s most sensitive region to global climate change, with impacts manifesting at a particularly rapid rates [63, 56]. Alarmingly, at least one-third of the Himalayan glaciers are projected to melt by the end of the century according to a recent study by International Centre for Integrated Mountain Development (ICIMOD) [106]. As a result of this, mountain–river discharge will increase in the short term but reduce in the long term [90]. This can lead to an increased risk of natural hazards such as glacial lake outburst floods (GLOFs) and droughts and may present significant challenges to mountain ecosystems and their people. Unsurprisingly, much research has been focused on mapping glaciers in the HKH [12, 109].
However, studying these glaciers in remote locations through field investigations are very time consuming and can pose high safety hazards. Furthermore, several studies have reported the performance of CIG, and DCG mapping in the HKH; however, most research has been focused on specific glacier basins within the region and not across the region as a whole. The main focus of our research is to be able to automate the accurate and timely mapping of glaciers in the HKH in order to make their study easier.

### 2.1 Glaciers

Glaciers are large bodies of dense ice that are constantly moving under its own weight from a higher to lower elevation. Glaciers are formed when fallen snow remains in one location long enough to be compressed into thickened ice masses. Most glaciers tend to flow like rivers but much slower due to the force of gravity on its sheer ice mass. Those present in the Hindu Kush Himalayan (HKH) region exhibit ice movements from the upper part of the glaciers to the lower part, or snout. Near its formation, the glacial ice have snow or ice surface cover and are known as CIG. As the glaciers slowly move down the valleys under gravity, avalanches can deposit debris (rocks and sediment) on top of the glacier. Such glacial ice have a significant covering of dirt/rocks/boulders over their termini are known as DCG. CIG and DCG appear differently in remote sensing imagery as can be seen in Figure 2.1.

![Sample Landsat 7 scene and Glacier labels for the same region](image)

Figure 2.1: Clean Ice & Debris Glaciers in Satellite Images
According to the Randolph Glacier Inventory (RGI) [3], there are 198,000 glaciers in the world covering 726000 km$^2$. The HKH region has the highest concentration of snow and glaciers outside the polar region, and thus has been called the Third Pole [35]. Figure 2.2 shows Khumbu glacier located in the Khumbu region of northeastern Nepal between Mount Everest and the Lhotse-Nuptse ridge.

Figure 2.2: Khumbu Glacier

2.1.1 Clean Glacial Ice

In the Himalayan region, glaciers begin forming in places where more snow gathers each year and little to no melt occurs. The snow begins to compress under its own weight to become more dense and tightly packed. As snow keeps piling on top, it becomes a dense, grainy ice called firn. As years go by, layers of firn build on top of each other. When the ice grows thick enough—about 50 meters (160 feet)—the firn grains fuse into a huge mass of solid ice [33] and glaciers are formed. These glaciers have snow or ice surface cover and are known as clean glacial ice.
2.1.2 Debris Glacial Ice

The glacier is so heavy and exerts so much pressure that it starts to move under its own weight. Pulled by gravity, they move slowly down a valley. Avalanches and icefalls transfer glacial ice from glaciers in higher altitude to a larger glacier beneath them, or directly to the valley below. During the process, an accumulation of boulders, stones, or other debris in the region is carried out and deposited by a glacier. Such glaciers have a mixture of ice/snow and dirt/rocks/boulders on their surface and are known as debris glacial ice. While definitions of what constitutes a DCG vary widely, a glacier does not have to be fully debris-covered to be classified as DCG [74].

2.1.3 Glacier Mapping in the HKH

Before remote sensing imagery became available, mapping glaciers, glacial lakes, or other geomorphological features had been done through manual labeling or field investigations [64]. The mapping of glaciers through the use of field investigations can have safety considerations due to the rugged and inaccessible terrains of the Himalayan region. Due to the resource and time required for mapping using these methods, they are not feasible for timely mapping of the glaciers. In such a situation, using automated methods to extract glaciers from the satellite images can produce timely and accurate glacial mapping with no potential safety hazards. A more recent approach to map glaciers in HKH region include using various thresholds for features such as remote sensing indices, slope, elevation, etc. on satellite imageries [11]. The obtained polygons are then manually smoothed to delineate glacier boundaries.

When segmenting glaciers from satellite imagery, the challenge lies in differentiating CIG from temporary snow/ice cover and DCG from their moraines and the surrounding valley. Glacier delineation using remote sensing imagery has seen a growing use of deep learning in recent years [38, 50, 12, 109, 102]. This can be attributed to factors such as the availability of large-scale remote sensing data from multiple sources, the develop-
ment of state-of-the-art deep learning architectures for image analysis, and the growing interest due to the impacts of climate change on glaciers in recent decades. The earliest approach for DCG segmentation involving deep learning used multilayer perceptrons to estimate the supraglacial debris loads of Himalayan glaciers using pre-defined glacier outlines [20, 19]. More recent approaches for glacier segmentation use Convolutional Neural Networks (CNNs) due to their success in image-based applications [76, 12, 109, 113]. Recent advancements in image segmentation use the U-Net architecture [93]. Originally introduced for biomedical image segmentation, the U-Net has seen successes in numerous applications involving satellite image segmentation [92, 73, 111]. Moreover, different architectures based on the U-Net have also been used for glacier segmentation in recent years [12, 109]. However, unlike segmentation for general images, the results when it comes to glacier segmentation are not very good, particularly for DCG.

2.2 Remote Sensing

Remote sensing is the science of acquisition of information about an object, primarily the earth surface, from a distance using sensors on airplanes or satellites. These sensors collect data in the form of optical images and provide specialized capabilities for manipulating, analyzing, and visualizing those images. In current usage, the term “remote sensing” generally refers to the use of satellite or aircraft-based sensor technologies to detect and monitor the physical characteristics of an area by measuring its reflected and emitted radiation at a distance.

Currently, petabyte-scale archives of remote sensing data are freely available from multiple U.S. Government agencies including the NASA, the U.S. Geological Survey, and National Oceanic and Atmospheric Administration (NOAA) [108, 69, 77], as well as the European Space Agency [1]. The catalog of geospatial datasets is continuously updated at a rate of nearly 6000 scenes per day, with a typical latency of about 24 hours from scene acquisition time.
2.2.1 Landsat 7 Satellite

Landsat 7 is an Earth-observing satellite that has been operational since 1999. Landsat 7 collects data in accordance with the World Wide Reference System 2, which has catalogued the world’s land mass into 57,784 scenes, each 183 km wide by 170 km long. The satellite carries the Enhanced Thematic Mapper Plus (ETM+) sensor that measures different ranges of frequencies (each called a band) along the electromagnetic spectrum – a color, although not necessarily a color visible to the human eye. The landsat 7 platform has eight bands as highlighted in Table 2.1. With an exception of the panchromatic band which is of the nominal resolution of 15 meters, the multispectral scanner (MSS) of Landsat 7 is of nominal resolution of 30 meters. This means that each pixel in an image represents 30 meters by 30 meters square on the ground. Because of this, we can only pick out individual features larger than 30 meters, but it is ideal for analyzing glacier size, glacier characteristics, and for mapping glacier change as the area covered by a glacier range from as small as a football field to hundreds of kilometers. The temporal granularity of landsat 7 satellite is 16 days which means there is a difference of 16 days between two subsequent images of the same

<table>
<thead>
<tr>
<th>Name</th>
<th>Wavelength</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Band 1 Visible</td>
<td>0.45 - 0.52 µm</td>
<td>(blue) surface reflectance</td>
</tr>
<tr>
<td>Band 2 Visible</td>
<td>0.52 - 0.60 µm</td>
<td>(green) surface reflectance</td>
</tr>
<tr>
<td>Band 3 Visible</td>
<td>0.63 - 0.69 µm</td>
<td>(red) surface reflectance</td>
</tr>
<tr>
<td>Band 4 Near-Infrared</td>
<td>0.77 - 0.90 µm</td>
<td>(near infrared) surface reflectance</td>
</tr>
<tr>
<td>Band 5 Near-Infrared</td>
<td>1.55 - 1.75 µm</td>
<td>(shortwave infrared 1) surface reflectance</td>
</tr>
<tr>
<td>Band 6 Thermal</td>
<td>10.40 - 12.50 µm</td>
<td>brightness temperature</td>
</tr>
<tr>
<td>Band 7 Mid-Infrared</td>
<td>2.08 - 2.35 µm</td>
<td>(shortwave infrared 2) surface reflectance</td>
</tr>
<tr>
<td>Band 8 Panchromatic (PAN)</td>
<td>0.52 - 0.90 µm</td>
<td>(black and white band) surface reflectance</td>
</tr>
</tbody>
</table>
area. One of the issues with remote sensing data is its temporal availability. Presence of clouds, cirrus, snow, and ice can affect the quality of images ultimately limiting the available data. Fang Chen et al. were able to demonstrate the availability of Landsat 7 data for guaranteeing a high quality yearly map of glacial lake in Tibet Plateau [25]. On average, landsat 7 had an average of 23.6 observations in the study area in 2015.

**Scan Line Corrector failure**

The Scan Line Corrector (SLC) in the ETM+ instrument failed on May 31, 2003. Without the effects of the SLC, the instrument images the Earth in a “zig-zag” fashion. This results in some areas being imaged twice and others that are not imaged at all. The net effect is that approximately 22% of the data in a Landsat 7 scene is missing when acquired without a functional SLC. We use the multi-scene (same path/row) gap-filled products developed by the U.S. Geological Survey (USGS) Earth Resources Observation Systems (EROS) Data Center (EDC) to improve the usability of Enhanced Thematic Mapper Plus (ETM+) [26]. Figure 2.3 shows the image before and after SLC line corrector fix for one of the Landsat 7 satellite images.

![Image due to SLC failure](image1.png) ![Image after fill](image2.png)

(a) Image due to SLC failure  (b) Image after fill

**Figure 2.3: Fixing Scan Line Corrector Failure**
2.3 Semantic Segmentation

Semantic segmentation is the process of classifying the object class for each pixel within an image, meaning there is a label for each pixel. It is one of the oldest and most widely studied problems in computer vision [23, 84, 91, 78, 94]. It is a widely studied area in statistics as cluster analysis and involves understanding not only what happens to be in the scene, but also what regions of the image those things are located in and at a very fine-grained levels. Figure 2.4 shows image classification, object detection, semantic segmentation, and instance segmentation for a visual representation. Semantic segmentation has a wide range of application areas like biomedical image diagnosis, autonomous vehicles, and geo-sensing.

![Image Classification](image1)

**Figure 2.4: Applications of Computer Vision**

![Object Localization](image2)

![Semantic Segmentation](image3)

![Instance Segmentation](image4)
where pixel-level processing of images is required.

### 2.3.1 Metrics to Evaluate Semantic Segmentation Model

How do we know our segmentation model is performing well? We visualize the predicted segmentation masks for qualitative evaluation. For quantitative evaluation, we evaluate the performance of our segmentation model based on Intersection over Union.

![Sample slice](image1)

![Ground Truth](image2)

![Model Prediction](image3)

Figure 2.5: Sample Prediction

**Intersection over Union**

The Intersection over Union (IoU) is the ratio of area of overlap between predicted label and ground truth and the area of union between predicted segmentation and ground truth. IoU ranges from 0-1 (0-100%) with 0 signifying no overlap (bad prediction) and 1 signifying perfectly overlapping segmentation (good prediction). For multi-class segmentation, the mean IoU is calculated by taking IoU of each class and averaging them. For example, in figure 2.5, mean IoU can be calculated by calculating individual IoU in each class and then averaging them as in Table 2.2. Visually, we can see IoU in Figure 2.6.

Mean IoU can be represented as: \( \frac{1}{n_{cl}} \sum_{i} n_{ii}/(t_{i} + \sum_{j} n_{ji} - n_{ii}) \), where \( n_{ij} \) is the number of pixels of class \( i \) predicted to belong to class \( j \), \( n_{cl} \) is the number of different classes, and \( t_{i} = \sum_{j} n_{ij} \) is the total number of pixels of class \( i \).
Table 2.2: Calculation of Mean IoU

<table>
<thead>
<tr>
<th>Class</th>
<th>Intersecting Area</th>
<th>Union Area</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean Ice Glacier</td>
<td>30.10%</td>
<td>35.70%</td>
<td>84.31%</td>
</tr>
<tr>
<td>Debris Glacier</td>
<td>7.43%</td>
<td>7.99%</td>
<td>93.02%</td>
</tr>
</tbody>
</table>

\[
\text{Mean IoU} = \frac{84.31\% + 93.02\%}{2} = 88.67\%
\]

![Intersection-Over-Union](image)

Figure 2.6: Intersection-Over-Union

### 2.4 Convolutional Neural Network (CNN)

With the increase in the size of the dataset, the number of parameters in a neural network architecture needs to be increased for effective learning. While this problem may not be as prevalent when the size of the data is smaller, a dense neural network has a lot of parameters and is harder to train when the size of data gets large. Convolutional Neural Network (CNN), also known as ConvNet, is a special type of neural network architecture
that makes the explicit assumption that the inputs are images, which allows us to encode certain properties (namely, stationary of statistics and locality of pixel dependencies) into the architecture. Thus, compared to standard feed-forward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters. These then make the forward function more efficient to implement while the theoretically-best performance is likely to be only slightly worse. A comparison of a convolutional neural network with a similar feed forward neural network is shown in Figure 2.7.

![Figure 2.7: Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be number of channels.](image)

There are three main types of layers that are used to build ConvNet architectures: convolutional layer, pooling layer, and fully-connected layer. Convolutional layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. The primary purpose of convolutional layer is to extract features from the input image. Pooling layer performs a downsampling operation along the spatial dimensions (width, height). The main function of the pooling layer is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Neurons in the fully-connected layer are the same as that in a simple feed forward neural network. It is usually attached at the end of the network architecture to
output fixed sized labels for classification.

2.4.1 Fully Convolutional Network (FCN)

One of the first papers to introduce the concept of CNNs for semantic segmentation was Fully Convolutional Networks (FCNs) for Semantic Segmentation [68]. In classification, conventionally, an input image is downsized and goes through the convolution layers and fully connected (FC) layers, and output one predicted label for the input image. Imagine replacing FC layer with convolution layers. The output will not be a single label. Instead, the output has a size smaller than the input image (due to the max pooling). The authors introduced this concept of transforming fully connected layers into convolutional layers as “convolutionalizing”. If we upsample the output, then we can calculate the pixel-wise output (label map). Fully Convolutional Networks (FCNs) owe their name to their architecture, which is built only from locally connected layers, such as convolution, pooling, and upsampling.

Since no dense layer is used in this kind of architecture, this reduces the number of parameters and computation time. The network can work regardless of the original image size, without requiring any fixed number of units at any stage, given that all connections are local. The authors of that paper were able to demonstrate that the convolutional networks by themselves, trained end-to-end, pixels-to-pixels exceed the state-of-the-art in semantic segmentation. This was the first work to train FCNs end to end (1) for pixel-wise prediction and (2) from supervised training. The authors introduced two methods for upsampling: Shift-and-stitch, and deconvolution layers and found the latter to be more effective and efficient.

2.4.2 U-Net

Inspired by the improvement achieved by FCNs on the task of semantic segmentation, Olaf Ronnenberger et al. introduced an architecture for biomedical image segmentation in 2015
called U-Net \cite{93}. The architecture of U-Net is shown in Figure 2.8. The U-Net, being a variant of FCN, inherits the property of FCN like reduced computation time. However, due to the specific property of the architecture, U-Net can only work on image sizes that satisfy certain conditions. U-Net architecture has two different but unique parts. The contraction phase, also sometimes known as the encoder, is used to capture the context of an image. Basically, it increases “what” and reduces the “where” in the image. The expanding phase, also sometimes known as the decoder, is used to enable precise localization using transposed convolution. So basically, the expanding phase adds “where” to the imagery. The main difference from a FCN is that convolution layers are followed by another successive convolution layer during the contraction phase and pooling operators are replaced by upsampling operations in the expansion layer. In order to localize, high-resolution features from the contracting path are combined with the upsampled output. A successive
convolution layer can then learn to assemble a more precise output based on the combined
information.

In a recent study published in 2019 by Yara Nohajerani et.al., the authors were able
to modify the U-Net architecture for effectively detecting glacier calving front margins
in satellite images. They trained their neural network architecture with glaciers from
Jakobshavn, Sverdrup, and Kangerlussuaq in Greenland and test the results on images from
Helheim glacier to evaluate the performance of the approach. They were trained on images
obtained from Landsat 5 ("green" band) and Landsat 7 and 8 ("panchromatic" band).
They performed the training on a set of 123 preprocessed images utilizing the capacity of
U-Net to perform well with a small set of data. Ten percent of the image during training
was left aside for cross-validation of the model and to prevent overfitting. The optimizer
they used is Adam and performed the training on a batch of 10 images at a time. They
use the concept of variable learning rate in which the learning rate is reduced by half after
every 5 epochs without improvement in the accuracy. This is done to ensure the network is
converging as with higher learning rates the network seems to diverge. The neural network
is able to get an accuracy of 92.4% on the training set and 93.6% on the test set after 54
epochs. Upon testing, they observed a mean deviation error of 96.3m, equivalent to 1.97
pixels on average which are comparable to the mean error of 92.5m obtained from hand-
drawn results on the same resolution. As a comparison, the Sobel filter, a commonly used
analytical edge-detection method, results in a mean error of 836.3m on the same dataset.
Figure 2.9 shows the comparison among Sobel filter, U-Net, and manual labeling. Here NN
depicts the corresponding extracted calving fronts using U-Net.

2.5 Machine Learning Models

Traditional machine learning models in general require less computational power and train-
ing time when compared to Deep Learning based approaches. Machine learning based ap-
proaches have been shown to work well for land cover mapping problems such as mapping
debris glaciers [61], and coastal features [81]. We performed initial tests with algorithms like naive bayes and support vector machines (SVM). However, due to the characteristics of these algorithms like quadratic nature of SVM and assumption that features have multinomial distribution in case of naive bayes, we decided to not use these algorithms. What follows is a discussion of three ML approaches for our glacier mapping task.

1. Random Forest: To understand random forest, we must first look at decision trees. A decision tree, put simply, is a graphical depiction of a decision and every potential outcome or result of making that decision. By displaying a sequence of steps, a decision tree makes it easy to follow steps to reach one of possible outcomes. Having more of these sequences is adding depth to the decision tree which makes it more accurate. However, each depth adds $2^n$ decisions to make where $n$ is the current depth of the tree.

Random forest are a collection of decision trees with a simple but powerful fundamental concept—the wisdom of crowds. The reason that the random forest model works so well is based on a principle that a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent
models. It operates as an ensemble of decision trees where each individual tree in the random forest produces a class and the class with the most votes becomes the prediction of the model. Unlike increasing depth which increases the number of decisions by a factor of \(2^n\), adding a new tree only increases the number of decisions by a factor of \(n\). Random forest is one of the most popular predictive models in machine learning due to its outstanding performance even with little parameter tuning \([39]\).

2. Extreme Boosting (XGBoost): Extreme Boosting, also known as eXtreme Gradient Boosting or XGBoost \([27]\), like random forest, is an ensemble based technique for performing supervised machine learning task. XGBoost is an efficient and scalable implementation of gradient boosting framework \([40]\). Decision trees are the most common type of individual model used in XGBoost and XGBoost seems to be similar to random forest in many ways. The most distinct difference between random forest and XGBoost is that the individual models (or trees) in case of XGBoost are not built on completely random subsets of data and features like random forest. XGBoost instead builds individual models sequentially by putting more weights on instances with wrong prediction and high errors. XGBoost is one of the most powerful machine learning algorithm at the time of writing. In case of XGBoost, the gradient is used to minimize a loss function, similar to how Neural Nets utilize gradient descent to optimize weights. Lets say that XGBoost is an ensemble of multiple decision trees. In each round of training, a decision tree is built and its predictions are compared to the output we expect. We compute the gradient by calculating the difference in predictions from the model and ground truth and use it to find the direction in which to change the model parameters to reduce the error. Unlike neural nets, where gradients are used to minimize the loss function and learn weights, in XGBoost, the gradients are used to add the next tree to the ensemble. This in turn makes XGBoost a powerful machine learning algorithm winning many competitions.

3. Multi Layered Perceptron (MLP): The basic unit forming MLP architecture is called
a perceptron. Each perceptron in a neural network architecture has one or more input values \((x_1, x_2, x_3, \ldots, x_n)\) and produces a single output value. In case of multi layered perceptron, this output value serves as an input to the next layer. The operation on each neuron is \(\sum W_i x_i + b_i\) followed by an activation function where \(W_i\) and \(b_i\) are the weights and bias values for the neuron respectively. Training a neural network means selecting optimal values for \(W_i\) and \(b_i\) for which the training set and the final output of the network are close to each other. After each iteration, the difference in labels from the training set and final output of the network is calculated and the weights of the network are updated using a process known as “back propagation” to make the output closer to the labels.

### 2.6 Deep Learning for Glacier Segmentation and Current Challenges

The major challenge in segmentation of glaciers from remote sensing imagery lies in differentiating CIG from temporary snow/ice cover and DCG from their moraines and the surrounding valley. The spectral uniqueness of CIG compared to surrounding terrains makes them easier to identify and localize. However, the delineation of DCG poses challenges because of the non-unique spectral signatures. Furthermore, while image segmentation techniques in recent years have gotten better with the help of deep learning, unlike segmentation for general images, the results when it comes to glacier segmentation are not very good, particularly for DCG.

Despite the advances in techniques for land cover mapping in recent years, having access to labelled data remains a limiting factor. Deep neural networks usually require thousands of images as training samples where the desired features have already been determined. However, U-Net [93] has been shown to provide highly accurate semantic segmentation with small numbers of training data. Since manual labelling from satellite images requires
considerable time and effort to generate training labels, the U-Net model is a good alternative for this situation. While the U-Net model requires significantly fewer volumes of data compared to traditional deep-learning based models, one of the shortcomings of this method is the need for accurate pixel level annotations during training. These types of annotations, typically referred to as dense labels, require considerable amount of time, effort, and often skilled annotators with a good understanding of the region of interest/labelled classes. Sparsely labelled data, where only some pixels for a given image are labelled, can be collected in large amounts in a relatively fast and cheap manner and often without the need for an expert. Promising advances in computer vision research have shown that such methods that can learn from unlabeled or partially labeled data [62]. As such, there have been many research projects focused on training CNN model with sparse training labels [6, 7, 105].

While deep learning models have been shown to perform better on various tasks involving computer vision, the interpretability of these models is limited. Deep Neural Networks are often considered black boxes since their decision rules can not be described easily. Unlike coefficients and decision boundaries of simpler machine learning methods like linear regression and decision trees, weights of neurons in deep neural networks can not be understood as knowledge directly. The development of transparent, understandable, and explainable models is imperative for the wide-scale adoption of deep learning models.
Chapter 3

Machine Learning for Glacier Monitoring in the Hindu Kush Himalaya

Glaciers are a source of freshwater and are critical to the Hindu Kush Himalayan (HKH) region both ecologically and societally [10]. However, glaciers are continuing to shrink at an alarming rate and this will result in diminished freshwater flow. This is likely to cause adverse effects for the benefactors of freshwater flow from glaciers. Additionally, glacier shrinkage has been shown to be a significant factor in the current sea-level rise [36]. This calls for effective and efficient methods to map and delineate glaciers in order to monitor changes and plan integrated water resource management and glacial hazard and risk management.

In such areas, remote sensing offers complementary information that can be used to monitor glaciers [88, 83]. Remote sensing allows the estimation of parameters like snow cover, glacier elevation, and ice index over large geographical and temporal scales. Utilising this information, different automated methods of delineating glaciers have been developed. However, the efficacy and accuracy of these methods are affected by cloud cover, highly variable snow conditions, and the spectral similarity of supra-glacial debris with moraines and bedrock [18]. These errors are somewhat addressed through the application of semi-automated mapping methodologies, which combine outputs from automated methods with manual interventions. However, this is labor intensive and time-consuming. Machine learning techniques can play a significant and positive role in speeding the process up.
3.1 Data Description

The HKH region covers an area of about 4.2 million km\(^2\) from \(\sim 15^\circ\) to \(\sim 39^\circ\) N latitude and \(\sim 60^\circ\) to \(\sim 105^\circ\) E longitude extending across eight countries of Afghanistan, Bangladesh, Bhutan, China, India, Myanmar, Nepal and Pakistan [11]. The training data (features) are Landsat 7 satellite imagery queried using Google Earth Engine. The corresponding glacier mappings (labels) are provided by International Centre for Integrated Mountain Development (ICIMOD) [11]. Figure 3.1 shows region bounded by each feature image and the labels in geographical map.

![Figure 3.1: Data in Geographical Context](image)

3.1.1 Understanding the Label Data

The label data is a shapefile with unique geometric regions for each glaciers as a separate polygon. The glacier outlines were derived semi-automatically using object-based image classification (OBIC) method separately for clean ice and debris cover. The outlines were then further edited and validated by carefully draping over the high resolution images from google earth. The glaciers for the HKH region are labelled during the time period of 2005\(\pm 3\) years. The labels we use have been generated through a semi-automated pipeline based on hyperpixel segmentation. Historically, ICIMOD has used the eCognition software [44] to segment Landsat imagery into image objects defined by a contiguous set of pixels with
similar intensity value. Those hyperpixels that contain debris or ice glacier are then merged and downloaded for refinement. The manual refinement phase involves removing labeled regions that are not at plausible glacier elevations or which do not pass specified NDVI, NDSI or NDWI thresholds \[47, 48, 41\].

### 3.1.2 Understanding the Feature Data

The feature data is a collection of 36 scenes from Landsat 7 satellite imagery used for generating glacier labels \[11\]. These scenes cover the area defined by the label polygons. Each scene is of approximately 170 km north-south by 183 km east-west (106 mi by 114 mi) in size. In addition to the 10 channels of Landsat 7 image in GEE, we included Normalized Difference Snow Index (NDSI) \[48\], Normalized Difference Water Index (NDWI) \[41\], Normalized Difference Vegetation Index (NDVI) \[47\], slope, and elevation as additional channels or bands. We took the additional information about slope and elevation from the Shuttle Radar Topography Mission (SRTM) digital elevation dataset and computed the remote sensing indices using different bands from Landsat 7 images as shown in equation \[4.1\].

\[
\begin{align*}
\text{NDSI} &= \frac{(B5 - B2)}{(B5 + B2)}, \\
\text{NDWI} &= \frac{(B4 - B5)}{(B4 + B5)}, \\
\text{NDVI} &= \frac{(B4 - B3)}{(B4 + B3)}.
\end{align*}
\]

We release our data in the [1](http://lila.science/datasets/hkh-glacier-mapping). The input data come in two forms – the original 35 Landsat tiles and 14,190 extracted numpy patches. Labels are available as raw vector data in shapefile format and as multichannel numpy masks. Both the labels and the masks are cropped according to the borders of HKH. The numpy patches are all of size 512 × 512 and their geolocation information, time stamps, and source Landsat IDs are available in a geojson metadata file.
3.2 Model Architecture and Methodological Pipeline

The task of identifying and mapping glaciers in remote sensing images fits well within the framework of semantic segmentation. We adapted the U-Net architecture for this task [93]. The U-Net is a fully convolutional deep neural network architecture; it consists of two main parts, an encoder network and a decoder network. The encoder is a contracting path that extracts features of different levels through a sequence of downsampling layers making it possible to capture the context of each pixel while the decoder is an expanding sequence of upsampling layers that extracts the learned encoded features and upsamples them to the original input resolution. Skip connections are employed between the corresponding encoder and decoder layers of the network to enable efficient learning of features by the model without losing higher resolution spatial information because of low spatial resolution in the bottleneck between encoder and decoder.

The model was trained using gradient descent and the Dice loss [100] was used as the optimization criterion (see the Appendix). We adapt a human-in-the-loop approach to correct the segmentation errors made by the model. This is useful because glacier mapping often requires expert opinion and models make errors that need to be resolved by people.

Figure 3.2: Our methodological pipeline first converts LE7 tiles to preprocessed patches used for model training. The trained model is deployed as an interactive web tool.

Our approach is summarized in a multi-step pipeline presented in Figure 3.2. It first...
converts the raw tiles into patches and converts their vector data labels to masks. We filter, impute and normalize the resulting patch-mask pairs before splitting them into train, test and validation data sets. The code to replicate our process can be found in a GitHub repository[2]. The script to query Landsat 7 tiles using Google Earth engine is in another GitHub repository[3].

3.3 Experiments

In this section, we characterize the performance of existing methods on tasks related to glacier segmentation. We intend to provide practical heuristics and isolate issues in need of further study.

![Feature importance scores using Random Forest. Slope and elevation are key variables.](image)

Figure 3.3: Feature importance scores using Random Forest. Slope and elevation are key variables.

**Band Selection** Model performance tends to deteriorate in the many-bands limited-training-data regime [79]. This is often alleviated through band subset selection. Here, we study whether specific channels are more relevant for glacier mapping. We experimented

---

[3] https://github.com/Aryal007/GEE_landsat_7_query_tiles/
Figure 3.4: Experimental results for channel selection. The $x$-axis indicates which LE7 bands were used. Color describes whether elevation and slope were used. Runs using NDWI, NDSI, and NDVI are labeled with triangles. Elevation and slope data significantly boost performance, and using all bands is better than using any subset. Results when using RF features are enclosed in a square.

with the combination of bands B5 (Shortwave infrared), B4 (Near infrared), and B2 (Green) which is the false-color composite combination used to differentiate snow and ice from the surrounding terrain when manually delineating glaciers. We compare this with (1) the true color composite band combination, B1 (Blue), B2 (Green), B3 (Red) and (2) all Landsat 7 bands. We also consider (1) slope and elevation from the Shuttle Radar Topography Mission (SRTM) as additional channels and (2) spectral indices - snow index (NDSI), water index (NDWI), and vegetation index (NDVI) - as used in manual glacier delineation [11]. Lastly, we perform pixel-wise classification on all channels with a random forest (RF) and select channels with feature importance scores greater than 5% (Figure 3.3).

Figure 3.4 shows performance when varying input channels. The experiments are carried out on the 383 patches with at least 10% of pixels belonging to either clean ice or debris-covered glaciers. We evaluated the model over 55 patches using Intersection over Union (IoU). The RF classifier features did not achieve the maximum IoU, likely due to a lack of
Figure 3.5: Example imagery and labels (a - c), with model predictions (d - f). Blue and orange labels are for clean ice and debris-covered glaciers, respectively. (d) A model trained to recognize the union of clean ice or debris-covered glaciers fails to recognize major debris-covered glaciers. (e - f) Multiclass and combined binary models give comparable predictions.

spatial context. Adding elevation and slope channels provides an improvement of 10-14% IoU. This agrees with domain knowledge – elevation and slope maps are referred to in the current process. Figure 3.6 illustrates that the model learns that low elevation and steep areas typically do not contain glaciers. Using NDVI, NDSI, and NDWI improves results when input channels are different from those used to define the indices.

Debris covered versus clean ice glaciers There are two types of glaciers we care about: clean ice glaciers and debris-covered glaciers. Clean ice glaciers have an appearance
Figure 3.6: Slope and elevation are critical for improving precision of glacier predictions. Light blue labels represent both clean ice and debris-covered glaciers. Light yellow and green are areas of high elevation (e) and slope (f). Clean ice glaciers tend to be found at high elevation, while debris-covered glaciers are found in valleys. Neither type of glacier is found on steep slopes.

Similar to snow, debris-covered glaciers are covered in a layer of rock and flow through valley-like structures. For segmentation, clean ice glaciers are often confused with snow, resulting in false positives. Debris-covered glaciers are more similar to the background, often leading to false negatives. Debris-covered glaciers are also much rarer. We experimented with binary and multiclass approaches to segmentation.

We trained a 2-class model to segment glacier from background areas and compared it with 3-class model for clean ice vs. debris-covered vs. background. We also compared the
Figure 3.7: Effect of debris percentage on IoU. The multiclass model performs well on areas with high density of debris-covered glaciers. The binary model trained to distinguish any type of glacier from background suffers in these regions. When making no distinguish between glacier types, the model only learns to recognize clean-ice glaciers.

Table 3.1: The exact numbers used in Figure 3.7

<table>
<thead>
<tr>
<th>% of Debris</th>
<th>% of Data</th>
<th>Binary Class IoU</th>
<th>Multiclass IoU</th>
<th>IoU Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 0%</td>
<td>100%</td>
<td>0.476</td>
<td>0.473</td>
<td>-0.3%</td>
</tr>
<tr>
<td>&gt; 1%</td>
<td>77%</td>
<td>0.523</td>
<td>0.532</td>
<td>+0.9%</td>
</tr>
<tr>
<td>&gt; 2%</td>
<td>52%</td>
<td>0.524</td>
<td>0.544</td>
<td>+2%</td>
</tr>
<tr>
<td>&gt; 5%</td>
<td>18%</td>
<td>0.497</td>
<td>0.571</td>
<td>+7.4%</td>
</tr>
<tr>
<td>&gt; 10%</td>
<td>6%</td>
<td>0.46</td>
<td>0.603</td>
<td>+14.3%</td>
</tr>
</tbody>
</table>

3-class model with two binary models for each glacier type. We filtered to patches where both debris-covered and clean ice glaciers were present, resulting in 648 training patches and 93 validation patches. Since many patches contain few positive class pixels, we evaluate IoU over the whole validation set rather than the mean IoU per patch. Table 3.2 shows that the multiclass model and binary model deliver comparable overall performance. However, the approaches differ in regions with higher coverage from debris-covered glaciers. Table
Figure 3.8: The image on the left shows the polygonized prediction for an area of interest. The image to the right shows the tool’s functionality of allowing users to correct predictions.

3.1 and Figure 3.5 show an increase in the performance gap in favour of the multiclass model as the debris-covered glacier percentage increases.

Table 3.2: A comparison of error rates on clean ice and debris-covered glaciers across three modeling approaches. The first row is a model trained to predict glacier or background, without distinguishing between debris-covered or ice glaciers. The second row is a multiclass model trained to simultaneously segment debris-covered and clean ice glacier. The final row gives the result of training two separate models to distinguish each type of glacier. Results are comparable across approaches, with a slight edge for the split training approach.

<table>
<thead>
<tr>
<th>Model</th>
<th>Glacier IoU</th>
<th>CIG IoU</th>
<th>DCG IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Model</td>
<td>0.476</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Multiclass Model</td>
<td>0.473</td>
<td>0.456</td>
<td>0.291</td>
</tr>
<tr>
<td>Two Binary Models</td>
<td>0.48</td>
<td>0.476</td>
<td>0.31</td>
</tr>
</tbody>
</table>

3.4 Glacier Mapping Tool

To support the work of geospatial information specialists to delineate glaciers accurately we developed an interactive glacier mapping tool. The tool allows users to test our segmenta-
tion models on different sources of satellite imagery. Users can visualize predictions in the form of polygons and edit them to obtain a glacier map for the area of interest. This interactivity supports the process of validating models, identifying systematic sources of error, and refining predictions before release. Users can compare data sources, which can clarify ambiguities. As future work, we intend to incorporate model retraining functionality. A screenshot from the tool is visible in Figure 3.8.

3.5 Pixel Based Segmentation

In this approach, we isolated the pixels for each of the channels in from a sample of satellite images that we prepared for image based approach. We then classified this feature data to one of the labels “Clean Ice Glaciers”, “Debris Glaciers”, or “Background” using the mask we created. Afterwards, we train this prepared data on machine learning models mentioned in Section 2.5 using samples from the training set and evaluate on test set. The validation set was used to select the best hyperparameters. We then use the trained model to predict corresponding class labels for each pixels in our test image to create segmentation mask based on the predicted label and compute IoU with the ground truth.

3.5.1 Data Preparation

We randomly sample at most 60000 pixels (20000 pixels max for each class) from each slice in the corresponding directory disregarding any spatial information between the pixels. Each row of data consists of 15 features, one for each channel in the satellite image and one of the three labels, “Background”, “Clean Ice Glaciers”, or “Debris Glaciers”. Since we have less pixels for the “Debris Glacier” class, we undersample from background and clean ice class on training and validation set so that we do not have class imbalance problem. We leave the test set unchanged so that it may represent the actual distribution of pixels. The distribution of the pixels is as shown in Table 3.3.
Table 3.3: Data Distribution for Pixel Based Segmentation

<table>
<thead>
<tr>
<th>Label</th>
<th>Training</th>
<th>Validation</th>
<th>Testing</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean Ice Glaciers</td>
<td>3422713</td>
<td>905140</td>
<td>2200000</td>
<td>6527853</td>
</tr>
<tr>
<td>Debris Glaciers</td>
<td>3422713</td>
<td>905140</td>
<td>622072</td>
<td>4949925</td>
</tr>
<tr>
<td>Background</td>
<td>3422713</td>
<td>905140</td>
<td>2200000</td>
<td>6527853</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>10268139</td>
<td>2715420</td>
<td>5022072</td>
<td><strong>18005631</strong></td>
</tr>
</tbody>
</table>

3.5.2 Machine Learning Models

We train the model using Random Forest (RF), Gradient Boosted Decision Trees (XGBoost), and Multi Layered Perceptrons (MLP). We use the validation set to find the optimal combination for selected set of hyperparameters for each model.

3.5.3 Evaluation Metrics

We perform evaluation on the test set along the metrics described in Table 3.4 when searching for best hyperparameters. We calculate the metrics averaged across different classes weighted by the number of true instances for those classes using formula (3.2)

\[
\text{weighted}_m = \frac{\sum_{i=1}^{n} w_i \cdot m_i}{\sum_{i=1}^{n} w_i}
\]

Here, \( \text{weighted}_m \) represents weighted metrics score (accuracy, precision, recall, f score), \( w_i \) represents the true instances for class \( i \), and \( m_i \) represents the corresponding metrics score for class \( i \).

3.6 Comparison Between Different Models

In this section, we observe quantitative evaluation for the segmentation masks generated using different model architecture. The metrics computed for this section are averaged for
Table 3.4: Machine Learning Models Evaluation Metrics

<table>
<thead>
<tr>
<th>Evaluation Metrics</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (Acc)</td>
<td>(\frac{(TP+TN)}{N})</td>
</tr>
<tr>
<td>Precision</td>
<td>(\frac{TP}{(TP+FP)})</td>
</tr>
<tr>
<td>Recall</td>
<td>(\frac{TP}{(TP+FN)})</td>
</tr>
<tr>
<td>F score</td>
<td>(\frac{2 \times \text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})})</td>
</tr>
</tbody>
</table>

\(TP = \text{True Positive}, FP = \text{False Positive}, TN = \text{True Negative}, FN = \text{False Negative}, N = \text{Total Number of Samples in Test Set}\)

clean ice and debris glaciers class disregarding the background class. The average metrics for all of the images in test set are shown in table 3.5.

Table 3.5: Comparison Between Different Models

<table>
<thead>
<tr>
<th>Model</th>
<th>IoU</th>
<th>Precision</th>
<th>Recall</th>
<th>Training Time</th>
<th>Inference Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>39.15%</td>
<td>43.53%</td>
<td>86.09%</td>
<td>5.90 minutes</td>
<td>2.43 minutes</td>
</tr>
<tr>
<td>XGBoost</td>
<td>38.46%</td>
<td>44.58%</td>
<td>84.14%</td>
<td>2.54 hours</td>
<td>3.45 minutes</td>
</tr>
<tr>
<td>MLP</td>
<td>36.16%</td>
<td>41.45%</td>
<td>82.23%</td>
<td>15.04 hours</td>
<td>7.33 minutes</td>
</tr>
<tr>
<td>U-Net</td>
<td>47.67%</td>
<td>61.50%</td>
<td>68.89%</td>
<td>3.64 hours</td>
<td>6.91 seconds</td>
</tr>
</tbody>
</table>

Furthermore, we also observe the performance of models on clean ice glaciers and debris ice glaciers separately. This comparison is shown in table 3.6.
Table 3.6: Performance Comparison for CIG and DCG

<table>
<thead>
<tr>
<th>Model</th>
<th>CIG</th>
<th>DCG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IoU</td>
<td>Precision</td>
</tr>
<tr>
<td>Random Forest</td>
<td>58.07%</td>
<td>66.36%</td>
</tr>
<tr>
<td>XGBoost</td>
<td>56.63%</td>
<td>69.50%</td>
</tr>
<tr>
<td>MLP</td>
<td>54.52%</td>
<td>64.60%</td>
</tr>
<tr>
<td>U-Net</td>
<td>58.29%</td>
<td>74.24%</td>
</tr>
</tbody>
</table>

3.7 Chapter Conclusions

We have presented deep learning and remote sensing techniques that support semi-automated glacier mapping. We have experimentally explored the effects of channel selection and task definition on performance. Finally, we describe a web tool to provide feedback and correct errors made by the model. Based on quantitative analysis of the results, we observe that image based segmentation method using U-Net architecture outperforms conventional machine learning based pixel wise classification for mapping glaciers in satellite imagery in terms of mean IoU by a margin of at least 8.52%. The inference time for generating new labels using U-Net model was about 21.1 times faster than the fastest pixel based segmentation approach using random forest. This adds to the usability of the method to predict glaciers for new images. Furthermore, one of the major issue seen with all pixel based segmentation is fragmentation of the generated segmentation masks. However, U-Net does not seem to have this problem. The performance improvement when using the U-Net model is highlighted when especially when we observe the performance on debris glaciers.
Chapter 4

Training U-Net on Sparsely Labelled Data for Segmentation of Waterbodies from Airborne Imagery

Despite the advances in techniques for land cover mapping in recent years, having access to labelled data remains a limiting factor. Deep neural networks usually require thousands of images as training samples where the desired features have already been determined. However, U-Net has been shown to provide highly accurate semantic segmentation with small numbers of training data. Since manual labelling from satellite images requires considerable time and effort to generate training labels, the U-Net model is a good alternative for this situation. While the U-Net model requires significantly fewer volumes of data compared to traditional deep-learning based models, one of the shortcomings of the model is the need for accurate pixel level annotations during training. These types of annotations, typically referred to as dense labels, as seen in Figure 4.1b, require considerable amount of time, effort, and often skilled annotators with a good understanding of the region of interest/labelled classes. Sparsely labelled data, as seen in Figure 4.1c, where only some pixels for a given image are labelled, can be collected in large amounts in a relatively fast and cheap manner and often without the need for an expert.

Promising advances in computer vision research have shown that such methods that can learn from unlabeled or partially labeled data [62]. As such, there have been many research projects focused on training CNN model with sparse training labels [6, 7, 105]. In this chapter, we present supervised ML methods that can learn from sparsely labeled
Figure 4.1: Our proposed method learns from sparsely labelled data for land/water segmentation. (a) Sample Image; (b) Corresponding Dense Labels; (c) Corresponding Sparse Labels.

data. Furthermore, we also show that U-Net model can be generalized across high resolution airborne imagery for segmentation of land and water pixels to detect water and land boundaries with performance comparable to that of existing methods.

4.1 Study Area and Data Sources

This study focuses on the coastal margin of the “1002 area” of the Arctic National Wildlife Refuge (ANWR). ANWR is a ~ 78,000 km$^2$ coastal plain region on the eastern North Slope of Alaska and was established as a refuge in 1980 through the Alaska National Interest Lands Conservation Act by Congress, recognizing the large potential for oil and gas resources and its importance as a wildlife habitat [45]. The 1002 area consists of barrier islands, salt marshes, coastal lagoons, coastal bluffs and river deltas that provide habitats for over 42 fish species, 37 land mammals, eight marine mammals and ~200 residential and migratory bird species (Figure 4.3, 4.4) [45]. These coastlines can display narrow, low-lying beaches while backshore coastal morphology can consist of sand and gravel beaches, barrier islands,
wetlands, barrier spits, and low-lying permafrost coastal bluffs with a range of ice content that are typically 2–6 m above sea-level in some areas [43, 57, 45]. Surface features of the coastal plain within our study area consist of tapped and untapped thermokarst lakes, coalesced low-center polygons, and braided rivers rich with sediment from the interior Brooks Range [45].

We obtained National Oceanic and Atmospheric Administration (NOAA) RSD high-resolution airborne RGB and NIR imagery covering the roughly 170-kilometer coastline of the 1002 area of ANWR collected between 18-19 July 2017. For both RGB and NIR scenes, 265 orthomosaiced image tiles were downloaded through NOAA’s Data Access Viewer [1] (accessed on 23 January 2021). Each image tile measured 2.5 km × 2.5 km for a total imagery footprint of 1672.80 sq. km (Figure 4.2). Images tiles were download in either 3-band (RGB) or 1-band (NIR), 8-bit GeoTIFF format and later reprojected to a NAD83/Alaska Albers (EPSG: 3338) coordinate reference system. Finally, corresponding RGB and NIR image tiles were composited into 265 4-band images using ArcMap 10.6. NOAA RSD collected this imagery from a Beechcraft King Air 350CER manned aircraft flying at a nominal altitude of ~2286 m above ground level (AGL) with two Appanix Digital Sensor System (DSS) SN580 cameras (one each for RGN and NIR). Image capture and precision georeferencing were synchronized and completed with an on-board Applanix POS/AV410 Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU). RGB and NIR camera systems had focal lengths of 52 mm and CDD pixel sizes of 5.2 × 5.2 µm and 6.0 × 6.0 µm respectively. While the ground sampling distance (GSD) of posted RGB and NIR orthomosaic images tiles was 35 cm. Stated horizontal accuracy for posted orthomosaics was ± 1.5 m at 95% confidence interval.

Coastlines along the Beaufort Sea are geomorphologically variable but can generally be classified as either bays/inlets, deltas, exposed bluffs, lagoons or tapped basins [58], where coastal lagoons make up over 50% of the region [19]. There are numerous factors controlling erosion of these coastal features including duration of sea ice-free extent, wind

[1] https://www.coast.noaa.gov/dataviewer/
Figure 4.2: Overview map showing the ANWR region on the eastern North Slope of Alaska, the extent of the 2017 NOAA airborne image collections (in blue) and the individual image tiles (in orange) from this collection used for creation of dense testing labels. The top right corner displays an example high resolution airborne image of a section of coastline acquired by the NOAA RSD airborne imaging.

fetch length, nearshore bathymetry, land cover type, and ground ice-content \[86, 104\]. These coastal features contain substantial stores of soil organic carbon (SOC) \[86\] and the erosion and subsequent release of SOC is partially responsible for the organic matter input
to nearshore marine environments that supports productive food webs \[34\] while riverine sediment transport also plays a significant role \[51\]. This influx of terrestrial and organic materials along with the range of water depths and sediment compositions are what makes these nearshore waters (from a remote sensing perspective) optically complex \[53\, 51\]. Within these areas are shallow lagoons and embayments with depths typically no greater than 10 m \[43\]. Sea level is minimally impacted by tidal range along the Beaufort Sea Coast (<50 cm) but can be dramatically elevated by a couple of meters through wind driven action \[14\, 45\].

4.2 Methods

The task of identifying and mapping geomorphological features in remote sensing images fits well within the framework of semantic segmentation. Semantic Segmentation is one of the oldest and most widely studied problems in computer vision \[23\, 84\, 91\, 78\, 94\] and involves understanding not only what objects are in the scene, but also what regions of the image the objects are located in and at what spatial resolution. In recent years, land cover mapping using semantic segmentation of satellite/airborne images has seen great success in different application domains. These can partly be accredited to an increasingly large amount of fully annotated images. However, collecting large scale accurate pixel-level annotations is time consuming and sometimes requires substantial amounts of financial investment and skilled labor. We get around such challenges by introducing a new method to train the modified U-Net model with easy to generate sparse labels. Here, we apply two remote sensing indices—NDWI \[71\], and NDSWI \[46\]—and two classical ML techniques—random forest, eXtreme Gradient Boosting or xgboost \[27\], and modified U-Net (Section 4.2.2)—to automate methods using a high-performance cloud computing environment for mapping water bodies from airborne imagery. We then fine-tuned the threshold to generate binary labels using Decision Stump (DS) (Section 4.2.3). We utilized a modified version of the U-Net architecture \[93\] to perform semantic segmentation—pixel-wise classification in im-
Figure 4.3: Natural color (RGB) image tiles selected from 2017 NOAA airborne imagery taken over the study area displaying the spectral variability of coastal and surface water bodies: (a) lagoonal waters bound by tundra and a sand spit; (b) dark, turbid coastal waters near the mouth of a river; (c) waters of a braided river carrying highly reflective glacial-derived sediments from the Brooks Range to the south of the study area; (d) coastal waters breaking near barrier islands; (e) dark and deep waters of a tundra pond; (f) shallow and turbid waters near a deltaic region; (g) blue waters of a large thermokarst lake; (h) sediment dispersion from coastal erosion in the nearshore; (i) shallow waters of a coalesced low-center polygonal pond.
Figure 4.4: Natural color (RGB) image tiles selected from 2017 NOAA airborne imagery taken over the study area displaying the spectral variability of land surfaces: (a) coastal foredune features; (b) alluvial deposits; (c) polygonized tundra; (d) drained thaw lake; (e) bare tundra; (f) wet and dry sands along a coastal spit.

ages. Using the ANWR as an area of study, we leveraged freely available high-resolution orthomosaic imagery for training and evaluation. We generated sparse labels for training and dense labels for evaluation using the approach in Section 4.2.1. Using these resources, we developed an extensible pipeline, a dataset and baseline methods that can be utilized for generating land/water masks from high resolution airborne images. We also present qualitative and quantitative results describing properties of our models.

4.2.1 Label Creation Strategy

Composited 4-band airborne image tiles were used to manually delineate areas of both water and land pixels by creating polygon features in ESRI shapefile formats within ArcMap [89]. These shapefiles were then used as model training and testing datasets. From the 265 total
airborne scenes, 165 scenes were hand-annotated with sparse labels by two remote sensing and ecological scientists familiar with landforms in the study area. Annotations were used for training after filtering out scenes that contained only water or only land or had image artifacts due to orthomosaicing. A total of 20 different scenes were used to create dense labels for the test sets. The strategy used by the specialists to make the sparse labels consisted of creating circular polygon features of various sizes within some (but not all) land and water regions of each scene. The dense labels, however, were annotated more carefully, and therefore more time was needed in delineating every single water pixel from each scene when creating the dense labels. While coastal water annotation was visually more straightforward, with the exception of deltaic regions, some terrestrial features required additional scrutiny to determine the presence of surface water. Annotators used a combination of visual characteristics to classify a feature as containing surface water:

1. Dark color in the visual spectrum indicating sufficient light attenuation in standing water;
2. the presence of reflected light due to ripples or waves caused by wind; and
3. the presence of accumulated white water on the western shorelines of water bodies caused by prevailing easterly winds.

The corresponding land labels were created by inverting water polygons for each scene.

4.2.2 Model Selection

Spectral Water Indices:

This study focused on testing and utilizing the capacity of NDWI and NDSWI in masking water and land pixels along arctic coastal tundra shorelines. Due to the selection of available bands in the source imagery used in this study (near-infrared, Red, Green and Blue), we were limited to the number of indices useful for water detection (primarily those that utilize the near-infrared band as opposed to the shortwave-infrared). Moreover, NDSWI
was specifically developed using in situ hyperspectral data of tundra wetlands and with a goal to develop an index that could not be confounded by atmospheric moisture as other water spectral indices have shown to be sensitive to in these arctic coastal marine ecosystems [46]. The equations for the indices are as shown in Equation (4.1).

\[
\begin{align*}
NDWI & = \frac{NIR - \text{Green}}{NIR + \text{Green}}, \\
NDSWI & = \frac{\ln(NIR) - \ln(\text{Red})}{\ln(NIR) + \ln(\text{Red})}.
\end{align*}
\]

(4.1)

Machine Learning:

Fundamental to any form of geospatial remote sensing image processing is the need for reliable, repeatable, and accurate landscape feature (e.g., shoreline, bluff edge, and beach width) identification and delineation. Arctic Coastal Change Detection (ACCD) is challenged by the need to detect change in coastal features (waterline, bluff edge, beach etc.) over thousands of kilometers of coast at high spatial and temporal resolutions. This type of land cover mapping is an application area of a wider range of problem in the computer vision community, known as semantic segmentation, for which supervised ML based approaches have performed well. Furthermore, these techniques have improved rapidly in recent years due to progress in deep learning and semantic segmentation with Convolutional Neural Networks (CNNs) [68, 93]. Recently, high performance computing, ML, and deep learning approaches have provided solutions for efficient and accurate landscape feature mapping across difference ecosystems. In the Arctic, studies have delineated polygonal tundra geomorphologies [5, 112], arctic lake features [28], glacier extents [12, 25, 107], and coastal features [81, 32, 59, 15]. In this research, we propose an automated pipeline using traditional ML based methods—random forest, and xgboost, and a deep neural network based U-Net architecture for arctic coastal mapping and compare their performances. One of the advantages of using ML based approaches over spectral indices is that the ML based method for land cover mapping generalizes geological features such as impervious surfaces, wetlands, and Plant functional types (PFT) for which the spectral indices are not
4.2.3 Threshold Fine-Tuning

Given that NDWI and NDSWI values ranges from $-1$ to $1$; the outputs from random forest, xgboost, and U-Net being probabilistic in nature, ranges from $0$ to $1$; and we require binary labels as the final segmentation mask; we needed to effectively convert these intensities to binary mask. A simple and widely used approach to do so is to use a threshold of $0.5$ for probabilistic output intensities and select the appropriate threshold based on the literature for NDWI ($\geq 0.3$ for water [72]) and NDSWI. Other methods for threshold fine-tuning include analyzing ROC curve [95], Otsu’s method [80], and DS—a one-level decision tree—[55]. We implement DS with IoU as the single input feature using exhaustive search.

4.3 Architectural Overview

We used two different spectral water indices—NDWI and NDSWI—and three ML methods—random forest, xgboost, and modified variant of the U-Net—for this study. Since the U-Net expects training labels to be dense, we modified dice loss [100] by masking this with pixel locations on the sparse labels. Then, this masked dice loss with gradient descent as the optimization algorithm was used for training the modified U-Net architecture.

Our approach is summarized in the multi-step pipeline presented in Figure 4.5 using NDWI and NDSWI to generate intensity masks (see Section 4.2.2 below). These approaches for generating intensity masks do not require training labels. To train the ML models, we first converted the raw vector data sparse labels to corresponding image masks for each image. We then augmented the 4 Band orthorectified airborne imagery with NDWI and NDSWI as additional channels, divided each airborne image/corresponding mask into 255 subregions, filtered out any subregion for which there was not at least 10% of labeled pixels, and randomly split them into training, testing and validation data sets. The training and validation samples are normalized dynamically during training with the mean and standard
deviation of the test set. As a post-processing step, we then generated a binary mask by thresholding the intensity for “water” class using optimal thresholds calculated using the validation set. The final scores that we report use the densely labelled test set.

Figure 4.5: Our methodological pipeline takes in airborne imagery and its corresponding sparsely labelled shapefile to prepare subregions for training and evaluation.
4.3.1 Masked Dice Loss

With the introduction of Convolutional Neural Networks (CNNs), different application areas involving semantic segmentation has achieved good results \[93, 12, 68\]. One of the downside of these CNN architecture is the need for dense labeled segmentation data. Obtaining large amounts of dense labeled segmentation data can sometimes be time consuming. It is therefore, a promising direction in computer vision research to develop semantic segmentation methods that can learn without the need for dense labels. These types of labels are also commonly referred to as weak labels. Previous research has reported semantic segmentation networks trained with various types of weak labels such as image level annotations \[54, 62\] and sparse labels \[6, 105\]. We present masked dice loss and a method to train a deep neural architecture using sparse training labels and masked dice loss in this research. Dice loss is based on the Sorensen–Dice coefficient \[31, 101\] which is a statistic used to gauge the similarity of two samples. In the computer vision community, it was introduced for 3D medical image segmentation \[75\]. Dice Loss is given by the equation:

\[
D = \frac{2 \times \sum_{i \in I} p_{i,c} \times g_{i,c}}{\sum_{i \in I} p_{i,c}^2 + \sum_{i \in I} g_{i,c}^2},
\]

where \( C \) is the set of classes that are present in the image, \( I \) is the set of pixels in the image, \( p_{i,c} \) denotes the probabilistic output from the model for class \( c \) at position \( i \), and \( g_{i,c} \) denotes the ground truth value for class \( c \) at position \( i \).

For our purpose, we need to train the image semantic segmentation network using sparse labels. For this purpose, we introduce masked dice loss in the equation:

\[
M(i) = \begin{cases} 
TRUE, & \text{if } \sum_{c \in C} g_{i,c} \neq 0, \\
FALSE, & \text{otherwise.}
\end{cases}
\]

\[
MaskedD = \sum_{c \in C, M(i) = TRUE} \frac{2 \times \sum_{i \in I} p_{i,c} \times g_{i,c}}{\sum_{i \in I} p_{i,c}^2 + \sum_{i \in I} g_{i,c}^2}.
\]

Implementing the masked dice loss, gradients are computed and back-propagated based only on the output for the pixels present in the ground truth sparse labels.
Reproducibility: Our approach implementation is based on scikit-learn \[85\] and pytorch \[82\]. All networks were trained on Azure NC6 Virtual Machine powered by NVIDIA Tesla K80 GPU. The code to replicate our process is available at \[2\] (accessed on 30 May 2021).

### 4.4 Results

Figure 4.6 shows examples of output land/water segmentation masks using different spectral indices and ML based approaches. From the figure, we can observe that the U-Net model has an intensity close to the end values for land/water classification compared to other methods. For quantitative evaluation, the intensity masks need to be converted to binary mask with unique values for each classes (in this case land and water) as can be seen on Figure 4.7.

#### 4.4.1 Threshold Fine-Tuning

While visualizing the results, instead of just finding a threshold that works the best for each of the methods, we plot a curve showing the performance at each possible threshold interval with a step size of 0.01. We use the validation set to determine the appropriate threshold that produces the highest IoU for class water using each method. The highest IoU and the threshold that yielded highest IoU on the validation is summarized in Table 4.1.

Figure 4.8 shows the histogram for pixels corresponding to land and water classes in the validation set, the IoU for land/water classes with an interval of 0.01 between thresholds, the threshold that yielded the maximum IoU, and maximum possible IoU. Based on the distribution of the histogram for land and water classes, we expect the results seen using remote sensing indices features to be more subjective to the thresholding value. This is exactly what we see in the line graph showing land and water IoU at each threshold interval. It is also important to note that the DS computed thresholds for NDWI (0.78),

\[2\] https://github.com/Aryal007/coastal_mapping
Figure 4.6: Output intensity on a sample image from the test set using different models. The intensity has been normalized to 0–1 range for NDWI and NDSWI for plotting. The intensity for U-Net, Random Forest, and XGBoost represents the probability of the corresponding pixel to be classified as water.

Random forest (0.4), and xgboost (0.38) are different from the commonly used thresholds (0.3, 0.5 and 0.5, respectively) while the DS computed threshold for U-Net model is close to the commonly used value of 0.5. At the time of writing, we could not find a published recommended threshold value for NDSWI. Based off of our findings, we propose using a
Figure 4.7: Human annotated dense label for the image in Figure 4.6. The intensity masks are converted to respective binary masks using thresholds as seen in Section 4.2.3 for evaluation.

Table 4.1: Threshold selection using exhaustive search DS.

<table>
<thead>
<tr>
<th>Method</th>
<th>Threshold</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS/NDWI</td>
<td>0.78</td>
<td>97.11</td>
</tr>
<tr>
<td>DS/NDSWI</td>
<td>0.48</td>
<td>96.42</td>
</tr>
<tr>
<td>DS/Random Forest</td>
<td>0.4</td>
<td>98.21</td>
</tr>
<tr>
<td>DS/XGBoost</td>
<td>0.38</td>
<td>98.26</td>
</tr>
<tr>
<td>DS/U-Net</td>
<td>0.53</td>
<td>97.43</td>
</tr>
</tbody>
</table>
value of 0.48 as the optimal threshold for land/water segmentation in environment similar
to the arctic coastal plain of Alaska. This means that the performance using the U-Net
model is less dependent on finding the optimal thresholding value over other methods.

4.4.2 Evaluations

We evaluate the performances of spectral water indices, ML models, and U-Net on the
densely labeled test set and used IoU, Precision, and Recall as our evaluation metrics.
The comparative performance between different methods can be seen in Table 4.2.

Table 4.2: Experimental results for binary segmentation masks generated using different
methods. The IoU when using random forest is higher than when using all other methods
for both land and water classes.

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>IoU</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>water</td>
<td>NDWI</td>
<td>94.90</td>
<td>96.99</td>
<td>97.78</td>
</tr>
<tr>
<td></td>
<td>NDSWI</td>
<td>93.83</td>
<td>95.91</td>
<td>97.75</td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>95.05</td>
<td>96.33</td>
<td>98.62</td>
</tr>
<tr>
<td></td>
<td>XGBoost</td>
<td>94.94</td>
<td>96.22</td>
<td>98.62</td>
</tr>
<tr>
<td></td>
<td>U-Net</td>
<td>94.86</td>
<td>96.56</td>
<td>98.19</td>
</tr>
<tr>
<td>land</td>
<td>NDWI</td>
<td>80.31</td>
<td>90.61</td>
<td>87.60</td>
</tr>
<tr>
<td></td>
<td>NDSWI</td>
<td>75.95</td>
<td>90.01</td>
<td>82.94</td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>80.51</td>
<td>92.41</td>
<td>86.22</td>
</tr>
<tr>
<td></td>
<td>XGBoost</td>
<td>79.65</td>
<td>93.71</td>
<td>84.16</td>
</tr>
<tr>
<td></td>
<td>U-Net</td>
<td>79.77</td>
<td>92.04</td>
<td>85.68</td>
</tr>
</tbody>
</table>
Figure 4.8: Binary mask generation by thresholding.
Region Based Evaluations

Each scene was divided into 225 512 \times 512-pixel sub regions with no overlap. The sub-regions were classified as coastal if they contained portions of the land/water interface adjacent to mainland backshore environments. This classification excluded barrier islands and deltaic regions where narrow beaches and/or permafrost bluffs do not directly interface with the waterline. A total of 162 sub regions were classified as coastal. We see a similar performance trend across all the models with random forest performing the best in terms of IoU. The comparative performance for region based evaluations between different methods can be seen in Table 4.3.

Table 4.3: Experimental results for coastal subregions using different methods. The overall performance of all the models are better for coastal subregions.

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>IoU</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>water</td>
<td>NDWI</td>
<td>96.55</td>
<td>97.81</td>
<td>98.69</td>
</tr>
<tr>
<td></td>
<td>NDSWI</td>
<td>95.96</td>
<td>97.41</td>
<td>98.46</td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>96.73</td>
<td>97.50</td>
<td>99.19</td>
</tr>
<tr>
<td></td>
<td>XGBoost</td>
<td>96.66</td>
<td>97.46</td>
<td>99.16</td>
</tr>
<tr>
<td></td>
<td>U-Net</td>
<td>96.64</td>
<td>97.6</td>
<td>98.99</td>
</tr>
<tr>
<td>land</td>
<td>NDWI</td>
<td>88.03</td>
<td>95.18</td>
<td>92.14</td>
</tr>
<tr>
<td></td>
<td>NDSWI</td>
<td>86.03</td>
<td>94.33</td>
<td>90.72</td>
</tr>
<tr>
<td></td>
<td>Random Forest</td>
<td>88.42</td>
<td>96.93</td>
<td>90.96</td>
</tr>
<tr>
<td></td>
<td>XGBoost</td>
<td>88.21</td>
<td>96.82</td>
<td>90.84</td>
</tr>
<tr>
<td></td>
<td>U-Net</td>
<td>88.21</td>
<td>96.23</td>
<td>91.37</td>
</tr>
</tbody>
</table>
4.5 Chapter Conclusions

We have addressed the problem of training DL based U-Net model using sparse labels for training and have shown that the sparsely labeled data allow it to learn good distribution comparable to other ML based methods. The results are very competitive but the random forest model provides slightly better results over the U-Net model trained using sparse labels for our task of land/water classification. Additionally, and from an operational perspective, our findings suggest that efficient and accurate surface water mapping can be achieved with less labeling effort and a lower barrier to entry in terms of computer science expertise. Although, since the performance of remote sensing indices are highly dependent on finding the optimal threshold, an exhaustive search for the threshold is needed to observe the best results. A relatively similar performance to that of ML based approaches is seen when using the remote sensing indices (NDWI and NDSWI), which could be attributed to the simplistic nature of the task itself (only two output classes), or to limited multispectral properties being incorporated in the input indices.
Chapter 5

Boundary Aware U-Net for Glacier Segmentation

It has been shown that the performance of deep learning models can be improved by learning multiple objectives from a shared representation [24]. We propose that learning the boundary objective in combination with traditionally used dice coefficient during the training process can improve glacier segmentation from Landsat-7 imagery. Early approaches to learn multiple tasks use weighted sum of losses where the loss weights are uniform, or manually tuned [37, 67, 96]. We propose a method to combine two different loss functions - masked dice loss [8] and boundary loss [21] - to simultaneously learn multiple objectives automatically during the training process for improved performance.

Additionally, one of the issues with deep learning models is their interpretability. While deep learning models have been shown to perform better on various tasks involving computer vision, the interpretability of these models is limited. Deep Neural Networks are often considered black boxes since their decision rules can not be described easily. Unlike coefficients and decision boundaries of simpler machine learning methods like linear regression and decision trees, weights of neurons in deep neural networks can not be understood as knowledge directly. The development of transparent, understandable, and explainable models is imperative for the wide-scale adoption of deep learning models. Over the years, many have proposed different approaches to describe deep learning models [70, 110, 113]. One of the most widely used methods to envision which pixels in the input image affect the outputs the most is by visualizing saliency maps [98]. A saliency map can be visualized by calculating the gradient of the given output class with respect to the input image by
letting gradients backpropagate to the input. In the case of multispectral or hyperspectral images, spectral saliency [66] is used to visualize salient pixels of an image. Image saliency maps, computed independently for all channels on a multispectral image, can be used to visualize the contribution of each pixel in each channel toward the final output. We propose a method to quantify each channel’s contributions towards the final label in the context of glacier segmentation using Landsat 7 imagery.

5.1 Dataset and Methodology

The HKH region covers an area of about 4.2 million km$^2$ from about 15° to 39° N latitude and about 60° to 105° E longitude extending across eight countries consisting of Afghanistan, Bangladesh, Bhutan, China, India, Myanmar, Nepal and Pakistan [11]. The geographic extent of the glaciers within the HKH however range from about 27° to 38° N and about 67° to 98° E (Figure 5.1).

We downloaded the Landsat 7 images used for label creation using Google Earth Engine. Landsat 7 contains the Enhanced Thematic Mapper Plus (ETM+) sensor which captures multiple spectral bands as shown in Table 5.1. The thermal infrared bands are upsampled from 60 meters to 30 meters resulting in all bands having a spatial resolution of 30 meters. The glacier outlines (labels) [9] were downloaded from ICIMOD Regional Database System. The glacier outlines contain information on clean-ice and debris-covered glaciers in the HKH for regions within Afghanistan, Bhutan, India, Nepal, and Pakistan.

The Landsat 7 images that were used for delineating glacier labels [11] overlap spatially. To avoid spatial overlap between train and test regions, we created polygon features representing a fishnet of rectangular cells for the entire geographical region. We then created a mosaic of all Landsat 7 images used for labeling into a single raster and clipped the raster mosaic to country boundaries for glacier labels (Figure 5.1) to avoid false negative glacier labels in the dataset. Finally, we discarded the rasters within the polygon cells that do not

1http://rds.icimod.org/Home/DataDetail?metadataId=31029
Figure 5.1: a) Spatially non-overlapping regions using fishnet grid. b) A zoomed image of one of the cells showing clean ice and debris glacier labels.

contain any glacier labels and downloaded clipped regions within selected cells. The raster pixels contained within selected cells were then randomly sampled into train, validation, and test sets with no geospatial overlap. Each cell was then cropped into multiple sub-images of $512 \times 512$ pixels and the sub-images with less than 10% of pixels as glacier labels were discarded. Every pixel within each sub-image can have one of four different classes as can be seen in Figure 5.2.

The step-by-step processing we followed to prepare input features for the model is shown in Figure 5.3. The distribution of pixels for train, validation, and test set across different
Table 5.1: Landsat 7 bands description

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>Blue</td>
</tr>
<tr>
<td>B2</td>
<td>Green</td>
</tr>
<tr>
<td>B3</td>
<td>Red</td>
</tr>
<tr>
<td>B4</td>
<td>Near Infrared</td>
</tr>
<tr>
<td>B5</td>
<td>Shortwave Infrared 1</td>
</tr>
<tr>
<td>B6_VCID_1</td>
<td>Low-gain Thermal Infrared</td>
</tr>
<tr>
<td>B6_VCID_2</td>
<td>High-gain Thermal Infrared</td>
</tr>
<tr>
<td>B7</td>
<td>Shortwave Infrared 2</td>
</tr>
</tbody>
</table>

Figure 5.2: (a) Sample sub-image, (b) Corresponding Clean Glacial Ice, Debris Glacial Ice, Background, and Masked labels
classes is shown in Table 5.2 and highlights that the distribution of pixels across different sets is similar and labels are heavily imbalanced across classes.

Table 5.2: Labels Distribution - Random Sampling

<table>
<thead>
<tr>
<th>split</th>
<th>background</th>
<th>CIG</th>
<th>DCG</th>
<th>masked</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>72.44%</td>
<td>21.77%</td>
<td>2.44%</td>
<td>3.35%</td>
</tr>
<tr>
<td>val</td>
<td>68.69%</td>
<td>23.22%</td>
<td>3.24%</td>
<td>4.85%</td>
</tr>
<tr>
<td>test</td>
<td>70.16%</td>
<td>22.97%</td>
<td>2.65%</td>
<td>4.21%</td>
</tr>
</tbody>
</table>

Figure 5.3: Input preprocessing

We used a modified version of the U-Net architecture [93] as shown in Figure 5.4. Each input sub-image is 512 × 512 pixels in size. Zero padding was added during each convolution operation to make the output labels the same size as input sub-images. We replaced the Rectified Linear Unit (ReLU) in the original U-Net architecture with Gaussian Error Linear Units (GELU) [52]. We applied batch normalization after each convolution operation and spatial dropout [103] of 0.1 at the end of each down sampling and up sampling block to control overfitting. We also randomly augment of 15% the training samples by
either rotating (90°, 180°, 270°) or flipping (horizontal/vertical) the input sub-images to the model. We trained the modified U-Net architecture for 250 epochs using adam optimizer and evaluated the performance based on precision, recall, and IoU.

Figure 5.4: Our modified U-Net architecture has 32 feature maps in the first convolution layer. We also introduce Batch normalization and spatial dropout in the modified architecture.

We trained two separate models for segmenting CIG and DCG and combined the outputs to get the final segmentation map. Definitions of what constitutes a DCG vary widely; however, a glacier does not have to be fully debris-covered to be classified as DCG [74]. Therefore, for the pixels where DCG labels overlapped with CIG labels on the final seg-
5.2 Experiments

5.2.1 Self-Learning Boundary-aware Loss

The subject with this section of our work intersects with two branches of research, which are penalizing misalignment of label boundaries by using boundary loss and learning multi-task weights during the training process. Here we propose combined loss ($\mathcal{L}_{\text{Combined}}$) which is a weighted sum of masked dice loss ($\mathcal{L}_{\text{MDice}}$) and boundary loss ($\mathcal{L}_{\text{Boundary}}$).

$$\mathcal{L}_{\text{Combined}} = \alpha \times \mathcal{L}_{\text{MDice}} + (1 - \alpha) \times \mathcal{L}_{\text{Boundary}}$$ (5.1)

Table 5.3: Results showing performance for combined loss and self-learning boundary-aware loss

<table>
<thead>
<tr>
<th>Loss ($\mathcal{L}$)</th>
<th>$\mathcal{L}_{\text{weight}(s)}$</th>
<th>CIG</th>
<th>DCG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{Combined}}$</td>
<td>0</td>
<td>68.40%</td>
<td>0.25%</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{Combined}}$</td>
<td>0.1</td>
<td>79.82%</td>
<td>79.66%</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{Combined}}$</td>
<td>0.5</td>
<td>81.60%</td>
<td>80.77%</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{Combined}}$</td>
<td>0.9</td>
<td>81.60%</td>
<td>80.77%</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{Combined}}$</td>
<td>1</td>
<td>80.31%</td>
<td>80.65%</td>
</tr>
<tr>
<td>$\mathcal{L}_{\text{SLBA}}$</td>
<td>Dynamic</td>
<td>81.59%</td>
<td>80.55%</td>
</tr>
</tbody>
</table>

$\alpha$ is a hyperparameter to the model whose value can be set manually between 0 and 1. An $\alpha$ of 0 is the same as training the model exclusively using masked dice loss and an $\alpha$ of 1 is the same as training the model exclusively using boundary loss. However, tuning the value of $\alpha$ manually for the best results is an expensive process. In order to learn the weights for $\mathcal{L}_{\text{Boundary}}$ and $\mathcal{L}_{\text{MDice}}$ through backpropagation, we initially assign $\alpha$ as 0.5 and
let the model find the best value of $\alpha$. However, we observe that without any constraints on the value of $\alpha$, the network updates $\alpha$ such that $L_{\text{Combined}}$ is minimized without necessarily having to minimize $L_{\text{MDice}}$ or $L_{\text{Boundary}}$. This results in poor performance. Inspired by [60] for weighing two different loss functions, we propose Self-Learning Boundary-Aware loss ($L_{\text{SLBA}}$) that is a combination of $L_{\text{MDice}}$ and $L_{\text{Boundary}}$.

$$L_{\text{SLBA}} = \frac{1}{2\alpha_1} \times L_{\text{MDice}} + \frac{1}{2\alpha_2} \times L_{\text{Boundary}} + |\ln(\alpha_1 \times \alpha_2)|$$  \hspace{1cm} (5.2)

In the case of $L_{\text{SLBA}}$, $\alpha_1$ and $\alpha_2$ both are initially set to 1 and we let the model find the best value for $\alpha_1$ and $\alpha_2$ through backpropagation. In Table 5.3, we show performance for different values of $\alpha$ in the case of $L_{\text{Combined}}$ and performance of $L_{\text{SLBA}}$. One advantage of using $L_{\text{SLBA}}$ over $L_{\text{Combined}}$ is that there is no extra hyperparameter that requires fine-tuning. All experiments in Table 5.3 use eight features from Landsat 7 imagery as inputs.

Figure 5.5: Masked Dice Loss weights and Boundary Loss weights vs. epoch for DCGs

From Table 5.3, we see that $L_{\text{SLBA}}$ performs the best for DCG segmentation and eliminates the need to fine-tune loss weights. We can also see that the model fails to converge while training solely on boundary loss ($\alpha = 0$) and training on glacier boundaries by incorporating boundary loss along with masked dice loss helps in an overall improvement in
performance for DCG regardless of the weighting factor. Figure 5.5 shows the weights for masked dice loss \((\frac{1}{2\alpha_1})\) and the weights for boundary loss \((\frac{1}{2\alpha_2})\) vs. epoch during training for DCGs. The optimal values for \(\alpha_1\) and \(\alpha_2\) are calculated to be 0.9569 and 1.045 for CIG segmentation and 0.952 and 1.05 for DCG segmentation for \(\mathcal{L}_{SLBA}\).

### 5.2.2 Representation Analysis

To understand the contribution of each feature in the multispectral image toward the final label, we computed Saliency Score (SS) for each feature by summing all pixels in the Saliency Map (SM) for that feature.

\[
SS_{\text{feature}} = \sum_{i=0}^{i=c} \sum_{j=0}^{j=r} SM_{\text{feature}}(i,j) \text{∀ feature ∈ Input} \tag{5.3}
\]

where:

\(r, c = \text{number of rows, columns in saliency map}\)

Average feature saliency scores across all the images in the training samples are shown in Figure 5.6. The channel-wise contribution towards DCG segmentation in decreasing order is: red, shortwave infrared 1, near infrared, green, high-gain thermal infrared, shortwave infrared 2, low-gain thermal infrared, and blue. Similarly, for CIG segmentation, the channel-wise contribution in decreasing order is: shortwave infrared 2, blue, shortwave infrared 1, high-gain thermal infrared, red, low-gain thermal infrared, near infrared, and green. As seen from Figure 5.6, the segmentation models have different high contributing channels for CIG and DCG segmentation.
5.3 Chapter Conclusions

In this chapter, we used a modified version of the U-Net for large-scale glacier mapping using the glaciers in the HKH to evaluate performance and concluded that DCG (IoU: 35.94%) are significantly harder to delineate compared to CIG (IoU: 68.17%) (Table 5.3). We have also shown that the performance of DCG can be improved by encouraging the deep learning model to focus on label boundaries. The performance can be improved further by correctly weighing loss terms and the relative weights can be learned automatically from the data during the training process. Figure 5.7 shows the performance of the models trained.
Figure 5.7: (a) Sample subimage from test set (b) Corresponding CIG and DCG ground truth labels (c) True positive (TP), False positive (FP), False negative (FN) for CIG (IoU 79.17%) (d) TP, TP, FN for DCG (IoU 59.19%)

Using $L_{SLBA}$ on a sample image from the test set. Furthermore, from Figure 5.5, we can see how the weights change for $L_{SLBA}$ while training. A higher weight is assigned to masked dice loss at the beginning and the weights for boundary loss are gradually increased during the training. The reason behind this could be that for an untrained model, it may be easier to learn glacier instances over trying to learn the boundaries. However once the network learns to label instances, it is easier to learn the glacier boundaries. This also explains why the model fails to converge when training solely on $L_{Boundary}$ from scratch as can be seen from the results in Table 5.3. We also introduced the concept of feature saliency scores to quantify the contribution of each feature (channel) in the input image toward the final
label and concluded that the red, shortwave infrared, and near infrared bands contribute the most towards the final label for DCG segmentation model while shortwave infrared 2, blue, shortwave infrared 1 bands contributed the most towards the final label for CIG segmentation model.
Chapter 6

Conclusion and Future Work

This dissertation presented solutions to what we see as four of the central issues in the application of deep learning for glacier segmentation from remote sensing imagery - performance issues, labeled data, model generalization, and explainability of the deep learning models.

In Chapter 3, we compared between various machine learning approaches for large-scale glacier segmentation in the HKH and showed that the U-Net model outperforms traditional machine learning approaches for the segmentation of glaciers from remote sensing imagery. We also observed that incorporating slope and elevation features to multispectral features from Landsat-7 satellite image helps in overall glacier segmentation performance. We have also experimentally explored the effects of channel selection and task definition on performance.

In Chapter 4, we used the methods presented in Chapter 3 to segment waterbodies from very high spatial resolution 4-band NOAA imagery. This shows the generalization capability of our methods across imageries with different spectral and spatial resolutions and for landcover mapping tasks other than glacier segmentation. We also introduced masked dice loss to train the U-Net model with sparse labels as input eliminating the need for high-cost dense label annotations for training.

Lastly, in Chapter 5, we used a modified version of the U-Net for large-scale glacier segmentation using the glaciers in the HKH to evaluate performance and concluded that DCG are significantly harder to delineate compared to CIG. We have also shown that the performance of DCG can be improved by encouraging the deep learning model to focus on label boundaries. The performance can be improved further by correctly weighing loss
terms and the relative weights can be learned automatically from the data during the training process. We introduce this loss as self-learning boundary-aware loss ($\mathcal{L}_{SLBA}$).

Table 6.1: IoU of existing glacier labels for HKH

<table>
<thead>
<tr>
<th>Source Data</th>
<th>Target Data</th>
<th>IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICIMOD glacier labels</td>
<td>RGI glacier labels</td>
<td>48.06%</td>
</tr>
<tr>
<td>ICIMOD glacier labels</td>
<td>GAMDAM glacier labels</td>
<td>42.77%</td>
</tr>
<tr>
<td>RGI glacier labels</td>
<td>GAMDAM glacier labels</td>
<td>56.69%</td>
</tr>
<tr>
<td>ICIMOD (clean ice only)</td>
<td>U-Net (clean ice only)</td>
<td>68.17%</td>
</tr>
<tr>
<td>ICIMOD (debris only)</td>
<td>U-Net (debris only)</td>
<td>35.94%</td>
</tr>
</tbody>
</table>

Glacier segmentation is a challenging task to solve, even for expert glaciologists. We compared the IoU between three publicly available glacier labels for the HKH region and found that the highest IoU scores between the labels was 56.69% (Table 6.1). However, some error may have occurred due to the temporal differences during label creation. In contrast, the IoU of the U-Net based model for CIG is 68.17%. While this may show an improved performance in terms of CIG, segmentation of DCG in particular is still far from being solved. Now we discuss different research directions for glacier segmentation from satellite imagery that we consider are worth studying. We think advancements in this research can be made along two different fronts:

- **Remote Sensing**: In this research, we have used a combination of Landsat-7 multispectral imagery along with digital elevation models and derived spectral indices for large scale glacier mapping in the HKH. The Operational Land Imager 2 (OLI-2) and the Thermal Infrared Sensor 2 (TIRS-2) sensors on the recently launched Landsat-9 provide data that is radiometrically and geometrically superior than instruments on previous generation Landsat satellites. The higher radiometric resolution
for Landsat-9 allows sensors to detect more subtle differences. With the higher radiometric resolution, Landsat 9 can differentiate 16,384 shades of a given wavelength compared to only 256 shades in Landsat-7. In addition, the TIRS-2 in Landsat-9 enables improved atmospheric correction and more accurate surface temperature measurements over Landsat-7. As DCG look similar to their surrounding regions in satellite imagery but may have significantly lower surface temperature, we believe the improved thermal sensor will help boost the performance on DCG segmentation. Furthermore, higher spatial resolution imagery such as the one from Sentinel-2 can also be explored for glacier segmentation. Finally, a promising research area can be to fuse higher spectral resolution/lower spatial resolution sensor data with higher spatial resolution/lower spectral resolution sensor data to generate synthetic higher spatial resolution/higher spectral resolution to aid in the task of glacier segmentation.

- **Deep Learning:** Deep learning research, particularly in recent years, is advancing in an accelerated rate. While the U-Net model was the state-of-the-art during the start of this research, recently vision based transformer models have been outperforming previous methods in most computer vision related tasks. Another very exciting area of future research involves using physics informed neural networks for incorporating information such as terrains and gravity to improve the performance.

    Furthermore, during the literature review, we found many research works that focus either on the ecological aspects associated with tracking changes in glaciers or on improving deep learning techniques used for image segmentation. However, only a few research works speak to the narrow intersection between these two different domains, and even fewer the large-scale implementation of these methods. We believe interdisciplinary research such as the one presented in this dissertation will have a huge role to play in future works to apply state-of-the-art methods developed in one discipline to tackle challenges and problems in another discipline of science.
References


[9] **Bajracharya, S., Maharjan, S., and Shrestha, F.** Clean Ice and Debris covered glaciers of HKH Region.


Chapter 7

Curriculum Vitae

Bibek Aryal was born on December 21\textsuperscript{st}, 1994. Son of Rajan Aryal and Chandrakala Aryal, he got his bachelors degree in Computer Science and Information Technology from Tribhuwan University, Kathmandu, Nepal on 2015. Between May of 2015 and August of 2018, Bibek worked as a web developer and later as a research programmer.

In the fall of 2018, he entered the graduate school of The University of Texas at El Paso as a Ph.D. scholar in the computational science program with professor Olac Fuentes of Department of Computer Science at UTEP as his research advisor. Bibek was a teaching assistant in the Mathematical Science Department from fall 2018 to spring 2020. Since the spring of 2020, he has been working as a Ph.D. research associate in Systems Ecology Lab, UTEP under the supervision of Dr. Craig Tweedie and Dr. Sergio A. Vargas Zesati. His research interest lies in the intersection between machine learning, computer vision, and remote sensing. He is particularly interested in the application of deep learning towards multispectral remote sensing imagery. He spent two summers in 2020 and 2021 with Microsoft Search, Assistant and Intelligence (MSAI) team at Microsoft as a data science intern under the guidance of Sebastian de la Chica. He also worked as a visiting scholar at Texas A&M AgriLife Research under the supervision of Dr. Saurav Kumar and Dr. Rocky Talchabhadel. He also received Microsoft AI for Earth Grant on December 2020. His work has been published in top AI and remote sensing venues including NeurIPS and MDPI Remote Sensing.

Email address: bibekaryal7@gmail.com