

2020-01-01

## Glacier Segmentation In Satellite Images For Hindu Kush Himalaya Region

Bibek Aryal  
*University of Texas at El Paso*

Follow this and additional works at: [https://scholarworks.utep.edu/open\\_etd](https://scholarworks.utep.edu/open_etd)



Part of the [Computer Sciences Commons](#), and the [Remote Sensing Commons](#)

---

### Recommended Citation

Aryal, Bibek, "Glacier Segmentation In Satellite Images For Hindu Kush Himalaya Region" (2020). *Open Access Theses & Dissertations*. 3140.  
[https://scholarworks.utep.edu/open\\_etd/3140](https://scholarworks.utep.edu/open_etd/3140)

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](mailto:lweber@utep.edu).

GLACIER SEGMENTATION IN SATELLITE IMAGES  
FOR HINDU KUSH HIMALAYA REGION

BIBEK ARYAL

Master's Program in Computational Science

APPROVED:

---

Olac Fuentes, Ph.D., 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

©Copyright

by

Bibek Aryal

2020

*To*

*MY MOTHER*

*for her unconditional love and support*

*&*

*MY LATE FATHER*

*the reason of what I become today*



GLACIER SEGMENTATION IN SATELLITE IMAGES  
FOR HINDU KUSH HIMALAYA REGION

by

BIBEK ARYAL

THESIS

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

MASTER OF SCIENCE

Master's Program in Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

December 2020

# Acknowledgements

First, I would like to express my deep gratitude to my advisor, Dr. Olac Fuentes, Professor of the Department of Computer Science at The University of Texas at El Paso, for his continuous support of my research, his advice, enthusiastic encouragement, and immense knowledge. His patient guidance helped me in all the time of research and writing of this thesis without which this work would not have been successfully completed.

I would also like to thank the other members of my committee, Dr. José M. Hurtado, Jr. and Dr. Marianne Karplus, both of the Department of Geology at the University of Texas at El Paso, and Dr. Anthony Ortiz for their time and effort. As a special note, Dr. Ortiz has been extremely helpful in providing the additional guidance and expertise I needed in order to complete this work.

I must also thank Microsoft AI for Earth initiative for supporting this work by providing computational resources, and the U.S. Geological Survey for making available the satellite images used in this project.

Additionally, I want to thank The University of Texas at El Paso Computational Science program professors and staff for all their hard work and dedication, providing me the means to complete my degree. I would like to recognize the invaluable assistance provided by Dr. Kris Sankaran, University of Wisconsin-Madison. I would also like to extend my gratitude to my colleagues from the Vision and Learning Lab at the University of Texas at El Paso who have been directly or indirectly involved in my learning process.

Finally, I wish to acknowledge the support and great love of my mother, Chandrakala Aryal, and my sister, Bijaya Aryal, who have always been there for me whenever I needed them. I also wish to thank all my friends in El Paso, who are like a second family to me.

NOTE: This thesis was submitted to my supervising committee on November 9<sup>th</sup>, 2020.

# Abstract

Climate change poses a risk to individuals whose livelihoods depend on the health of glacier ecosystems. Monitoring glaciers in the Himalayan Hindu Kush (HKH) region is of high importance especially when we consider the impact of recent climate change on them. Our work aims to provide an automated method to outline glaciers using machine learning techniques and publicly available remote sensing imagery.

In this work, we present ways to delineate glaciers from Landsat-7 imagery using various machine learning and computer vision techniques. The multi-step methodology that we present in this work is generalizable across different types of satellite and overhead imagery, lending itself to map other geomorphological features on the Earth’s surface. Furthermore, we compare quantitatively and qualitatively the performance of pixel-wise classification using conventional machine learning to a more recent deep learning based architecture, U-Net. Our proposed works consist of integrating conventional computer vision methods with deep learning based approaches to improve the segmentation performance and later generalize across other landcover mapping applications beyond glacier mapping.

Despite being faster to train, pixel-wise classification approaches generate segmentation masks that are fragmented. On the other hand, the problem of fragmented prediction masks is visually less apparent when using a U-Net architecture. This could be attributed to the properties of Convolutional Neural Networks which are able to take spatial information into consideration. Specifically, pixels in the predicted segmentation mask using U-Net architecture are computed by taking a neighborhood of pixels in the input image, as opposed to one pixel at a time, resulting in less fragmentation.

We also analyze the features of satellite images that are most helpful in classification of glaciers in the HKH region. Based on the domain knowledge, we calculate and add slope, elevation, and spectral indices (i.e. Normalized Difference Snow Index (NDSI), Normalized Difference Vegetation Index (NDVI), and Normalized Difference Water Index (NDWI)) as

additional bands on top of the bands from Landsat-7 satellite to help with the segmentation task. We observed that slope, elevation, and NDSI contribute the most towards the final segmentation mask. However, these three channels are not present in Landsat-7 satellite scenes and need to be calculated separately. These findings can change the way people view glaciers and the features associated with it, leading to a better understanding in monitoring them.

Overall, we present multiple methods for mapping 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), particularly due to the large volumes of public freely-available large-scale satellite images made available in recent years.

We expect to present a novel method that is optimized for the task of glacier mapping using a combination of deep learning and conventional computer vision methods. We also expect to present an optimal architecture for glacier delineation and deploy it in the form of a tool that will automate the process and also be able to facilitate the delineation of glaciers on satellite images acquired from other sensors.

# Table of Contents

	<b>Page</b>
Table of Contents . . . . .	viii
List of Tables . . . . .	xi
List of Figures . . . . .	xii
<b>Chapter</b>	
1 Introduction . . . . .	1
1.1 Motivation . . . . .	1
1.2 Thesis Statement . . . . .	3
1.3 Research Questions . . . . .	3
1.4 Expected Contributions . . . . .	3
1.5 Outline . . . . .	4
2 Mapping Glaciers in Satellite Imagery . . . . .	6
2.1 Glaciers . . . . .	6
2.1.1 Clean Ice Glaciers . . . . .	7
2.1.2 Debris Glaciers . . . . .	7
2.1.3 Glacier Mapping . . . . .	8
2.2 Remote Sensing . . . . .	9
2.2.1 Landsat 7 Satellite . . . . .	9
2.2.2 Google Earth Engine . . . . .	11
2.3 Semantic Segmentation . . . . .	12
2.3.1 Metrics to Evaluate Semantic Segmentation Model . . . . .	12
2.4 Convolutional Neural Network (CNN) . . . . .	15
2.4.1 Fully Convolutional Network (FCN) . . . . .	16
2.4.2 U-Net . . . . .	17
2.5 Active Contour Models (ACMs) . . . . .	19

2.6	Machine Learning Models . . . . .	21
2.7	Summary . . . . .	23
3	Proposed Research . . . . .	25
3.1	Data Description . . . . .	25
3.1.1	Understanding the label data . . . . .	25
3.1.2	Understanding the feature data . . . . .	26
3.2	Image Based Segmentation . . . . .	27
3.2.1	Data Preparation . . . . .	28
3.2.2	Model Architecture . . . . .	29
3.3	Pixel Based Segmentation . . . . .	29
3.3.1	Data Preparation . . . . .	30
3.3.2	Machine Learning Models . . . . .	31
3.3.3	Evaluation Metrics . . . . .	31
3.4	Conclusion . . . . .	32
4	Preliminary Results . . . . .	33
4.1	Results: Pixel Based Segmentation . . . . .	33
4.1.1	Random Forest . . . . .	33
4.1.2	Extreme Boosting (XGBoost) . . . . .	36
4.1.3	Multi Layered Perceptrons (MLP) . . . . .	39
4.2	Results: Image Based Segmentation . . . . .	40
4.3	Combined Results . . . . .	42
4.4	Generated Segmentation Masks . . . . .	43
4.5	Significance of the Result . . . . .	44
5	Research Plan and Timeline . . . . .	47
5.1	Proposed Work . . . . .	47
5.2	Timeline . . . . .	49
	References . . . . .	50
6	Appendix: Segmentation masks . . . . .	57

7 Curriculum Vitae . . . . .	63
------------------------------	----

# List of Tables

2.1	Landsat 7 Band Descriptions . . . . .	10
2.2	Calculation of Mean IoU . . . . .	14
3.1	Data Distribution for Pixel Based Segmentation . . . . .	30
3.2	Machine Learning Models Evaluation Metrics . . . . .	31
4.1	Random Forest Number of Trees vs Nodes (Maximum Depth 15) . . . . .	34
4.2	Random Forest Maximum Depth vs Nodes (Number of Trees 15) . . . . .	34
4.3	Experimental Result for Random Forest . . . . .	35
4.4	Confusion Matrix for Random Forest . . . . .	35
4.5	Extreme Boosting Parameters Tuning . . . . .	38
4.6	Experimental Result for Extreme Boosting . . . . .	38
4.7	Confusion Matrix for Extreme Boosting . . . . .	38
4.8	Experimental Result for Multi Layered Perceptrons . . . . .	40
4.9	Confusion Matrix for Multi Layered Perceptron . . . . .	40
4.10	Comparison Between Different Models . . . . .	43
4.11	Performance comparison for clean ice and debris glaciers . . . . .	43



# List of Figures

2.1	Khumbu Glacier . . . . .	7
2.2	Clean Ice & Debris Glaciers in Satellite Images . . . . .	8
2.3	The Earth Engine Interactive Development Environment . . . . .	11
2.4	Applications of Computer Vision [19] . . . . .	13
2.5	Sample Prediction . . . . .	13
2.6	Intersection-Over-Union . . . . .	14
2.7	Left: A regular 3-layer Neural Network. Right: A ConvNet [2]. . . . .	15
2.8	U-Net Architecture [45] . . . . .	17
2.9	Comparison of U-Net Architecture with Other Methods [35] . . . . .	19
2.10	Example of an Image Block for Glacial Lake Extraction [10] . . . . .	21
3.1	Data in Geographical Context . . . . .	26
3.2	Fixing Scan Line Corrector Failure . . . . .	27
3.3	Sample Sliced Landsat 7 Image and Corresponding Label Mask . . . . .	28
4.1	Tuning Parameters for Random Forest . . . . .	34
4.2	Feature Importance, Random Forest . . . . .	36
4.3	Sample Prediction using Random Forest . . . . .	36
4.4	Tuning Parameters for Extreme Boosting . . . . .	37
4.5	Feature Importance, Extreme Boosting . . . . .	39
4.6	Sample Prediction using XGBoost . . . . .	39
4.7	Sample Prediction using Multi Layered Perceptron . . . . .	41
4.8	Sample Prediction using U-Net Model . . . . .	42
4.9	Example Image Slices . . . . .	44
4.10	Corresponding Labels . . . . .	44

4.11 Prediction Mask using Random Forest . . . . .	44
4.12 Prediction Mask using Extreme Boosting . . . . .	45
4.13 Prediction Mask using Multi Layer Perceptron . . . . .	45
4.14 Prediction Mask using U-Net . . . . .	45
5.1 Timeline . . . . .	49

# Chapter 1

## Introduction

### 1.1 Motivation

The glaciers in the Himalayan Hindu Kush (HKH) region possess one of the largest resources of snow and ice on Earth, constituting a huge freshwater reservoir. Changes in the glacial extent and their influence on river run-off are important to map and plan ahead for future strategies of power generation as numerous power projects are under operation and construction in these river basins [32]. Moreover, monitoring changes such as melting of glaciers over time which increases the risk of glacial lake outburst floods (GLOFs) becomes critical. GLOFs are among the most serious natural hazards in high mountainous regions and can cause catastrophic damages and fatalities in downstream communities and ecosystems [51, 54]. Understanding the changes in these glaciers is critical in order to develop infrastructure that improves the well-being and livelihoods of high-altitude communities, to promote and accelerate learning on the challenges facing mountain ecosystems and their people, and to empower critical and urgent decision making during regional humanitarian crises.

However, understanding and tracking these landscape level changes can be difficult. For example, glaciers and particularly those located in the HKH region, are situated in remote areas that can be difficult and unsafe to navigate through and map using traditional approaches, such as field visits and vertical air photography. In addition to having safety considerations, the use of these methods is also costly and time-consuming. Satellite remote sensing data can be helpful in this scenario as it is a tool for periodical observations of Earth's surface over large areas. Petabyte-scale archives of remote sensing data

are freely available from multiple U.S. Government agencies as well as from the European Space Agency. Additionally, Google Earth Engine (GEE) provides an application programming interface (API) and an associated web-based Interactive Development Environment (IDE) for processing very large geospatial datasets collected from multiple satellites such as Landsat, Sentinel, and MODIS [20].

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. Fortunately, there have been advancements in areas of computer vision that allow researchers to automate this task. The specific task that deals with assigning pixels to corresponding labels of interest is called semantic segmentation and one example application is landcover mapping (i.e. detecting and labeling geological features in satellite imagery). There are multiple methods that have been developed over the years for semantic segmentation such as threshold segmentation, Sobel Filters, and Neural Networks [53]. Semantic segmentation techniques have improved rapidly in recent years due to the progress in Convolutional Neural Networks (CNNs) [30, 45].

Despite the advances on semantic segmentation, having access to label data remains a limiting factor. Deep neural networks usually require thousands of images as training samples where the desired features (in this case glaciers) have already been determined. However, a recently developed model, U-Net, [45] has been shown to provide a highly accurate semantic segmentation with small numbers of training data. Since manual labelling of glaciers from satellite images requires considerable time and effort, the U-Net model is a good alternative for our situation.

The HKH is often referred to as the third pole [15] and holds approximately 14.5% of the world's total glaciers. 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 water 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 [6, 8]. Climate change poses a risk to the glaciers and its impact in the Himalayas can already be seen [7, 13]. The changes in glaciers due to climate change may have a significant impact on the quantity and timing of water availability downstream [43] and could also increase the risk of natural hazards such as GLOFs ultimately affecting human livelihood. Efficient and timely monitoring of the state of glaciers is key for water resource and hazard management in the region.

## 1.2 Thesis Statement

*The information contained in multi-band remote sensing scenes can be used to accurately delineate glacier boundaries through the use of a combination of deep learning and conventional machine learning and computer vision techniques.*

## 1.3 Research Questions

Our work aims to answer the following questions:

1. Is it possible to delineate glacier boundaries automatically with an accuracy that is similar to that of human experts using publicly available remote sensing satellite scenes?
2. Can methods for glacier mapping be extended to other land cover mapping applications?

## 1.4 Expected Contributions

As a result of the proposed research, we expect to achieve the following:

1. **Multisource feature analyses and fusion.** We will develop a methodology for selecting, combining, and generating features from different satellites such that these

features correspond to the same geographical location on the Earth’s surface. These modified satellite images will then be used as inputs to the models to identify which features are important for mapping glaciers. In addition to improving results for automated models, these analyses may be helpful to the experts in the domain for efficient mapping of glaciers.

2. **A novel framework for mapping glaciers that combines deep learning and a conventional computer vision method.** We will develop a novel framework that unifies deep learning with active contour models that is optimized for glacier segmentation mask generation. The framework will also be generalizable across other computer vision applications.
3. **An architecture optimized for the task of glacier mapping.** We will experiment with different modifications of the U-Net architecture and tune the hyperparameters to find an optimal architecture for glacier mapping task.
4. **Tool for glacier mapping from satellite imagery.** We will develop an automated system for end-to-end mapping of glaciers in the HKH region. The automated tool will be able to read the satellite images and corresponding glacier labels, preprocess them, and train a classifier optimized for the task of glacier mapping. The trained classifier can then be used to segment glaciers from the satellite images. The tool will be easy to tune and reconfigure in response to user needs.
5. **Extension of the tool across different images for other landcover mapping tasks.** We will extend the tool for other applications of landcover mapping in addition to glacier mapping.

## 1.5 Outline

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 mapping geographical

features from satellite imagery using machine learning approaches. Chapter 3 describes the dataset and presents research approaches to achieve some of the expectations of this work. Preliminary results are discussed in chapter 4. Finally, in chapter 5, we will propose our work plan and timeline to complete it.

# Chapter 2

## Mapping Glaciers in Satellite Imagery

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 [28, 29]. However, studying these glaciers in remote locations through field investigations are very time consuming and can pose high safety hazards. The main focus of our research is to be able to automate the accurate and timely mapping of these glaciers 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 When fallen snow remains in one location long enough to be compressed into thickened ice masses and most 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. According to the Randolph Glacier Inventory (RGI) [3], there are 198,000 glaciers in the world covering 726000 km<sup>2</sup>. The HKH region has the highest concentration of snow and glaciers outside the polar region, and thus has been called the Third Pole [15]. Figure 2.1 shows Khumbu Glacier located in the Khumbu region of northeastern Nepal between Mount Everest and the Lhotse-Nuptse ridge.

Since our work aims to distinguish glaciers from non-glacier background through surface





Figure 2.1: Khumbu Glacier

reflectance on Landsat 7 satellite images, the major challenges occur in distinguishing clean ice glaciers from temporary snow/ice covers and debris glaciers from rocks and other geomorphological features. In this case, we want to have three different output labels - Clean Ice glaciers, Debris Glaciers, and Non Glaciers/Background.

### 2.1.1 Clean Ice Glaciers

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 [14] and glaciers are formed. These glaciers have snow or ice surface cover and are known as clean ice glaciers.

### 2.1.2 Debris Glaciers

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 glaciers.

## Glaciers (Continued)

Visibly, this difference in clean ice and debris glaciers is quite obvious in satellite images as we can see in Figure 2.2.

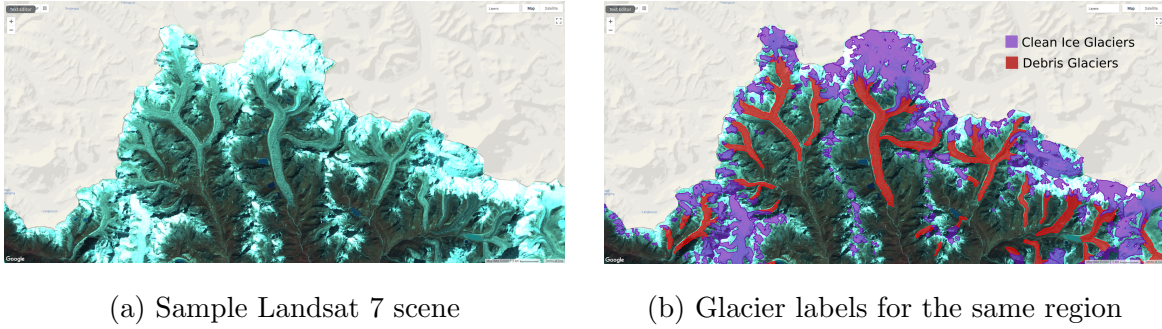


Figure 2.2: Clean Ice & Debris Glaciers in Satellite Images

Here the red labels denote debris glaciers and purple labels denote clean ice glaciers. The labels correspond to the digital polygon data of status of glaciers in Hindu Kush Himalayan (HKH) region [5].

### 2.1.3 Glacier Mapping

Before remote sensing imagery became available, mapping glaciers, glacial lakes, or other geomorphological features had been done through manual labeling or field investigations [27]. 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 [5]. The obtained polygons are then manually smoothed to delineate glacier boundaries.

## 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) [52, 31, 37], 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

Table 2.1: Landsat 7 Band Descriptions

Name	Wavelength	Description
Band 1 Visible	0.45 - 0.52 $\mu\text{m}$	(blue) surface reflectance
Band 2 Visible	0.52 - 0.60 $\mu\text{m}$	(green) surface reflectance
Band 3 Visible	0.63 - 0.69 $\mu\text{m}$	(red) surface reflectance
Band 4 Near-Infrared	0.77 - 0.90 $\mu\text{m}$	(near infrared) surface reflectance
Band 5 Near-Infrared	1.55 - 1.75 $\mu\text{m}$	(shortwave infrared 1) surface reflectance
Band 6 Thermal	10.40 - 12.50 $\mu\text{m}$	brightness temperature
Band 7 Mid-Infrared	2.08 - 2.35 $\mu\text{m}$	(shortwave infrared 2) surface reflectance
Band 8 Panchromatic (PAN)	0.52 - 0.90 $\mu\text{m}$	(black and white band) surface reflectance

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 area. One of the issues with remote sensing data is its temporal availability. Presence of clouds, snow/ice and cirrus 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[10]. On average, landsat 7 had an average of 23.6 observations in the study area in 2015.

## 2.2.2 Google Earth Engine

The easiest way to access these remote sensing data is through the use of the google earth engine. The google earth engine brings together the world's satellite imagery that have been collected through the use of various satellites such as landsat, modis, and sentinel over the years. Google earth engine is a cloud-based platform that makes it easy to access high-performance computing resources for processing very large geospatial datasets, without having to suffer the IT pains currently surrounding either [20]. It is accessed and controlled through an Internet-accessible application programming interface (API) and enables rapid prototyping and visualization of results with the help of associated web-based interactive development environment (IDE). Users can access the API either through a thin client library or through a web-based interactive development environment built on top of that client library. Figure 2.3 shows the web-based interactive development environment provided by google earth engine.

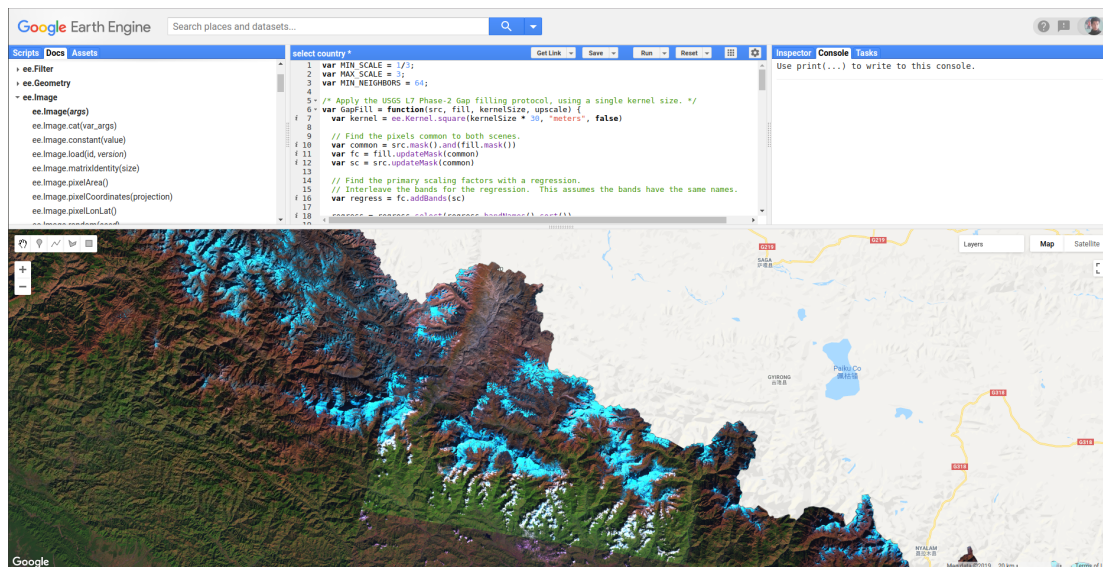


Figure 2.3: The Earth Engine Interactive Development Environment

Users can sign up for access at <https://earthengine.google.com>, the Earth Engine home-page, and access the user interface, as well as a user's guide, tutorials, and examples. Ac-

counts come with a quota for uploading personal data and saving intermediate products, and any inputs or results can be downloaded for offline data.

## 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 [9, 41, 44, 38, 46]. 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 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.

#### 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.

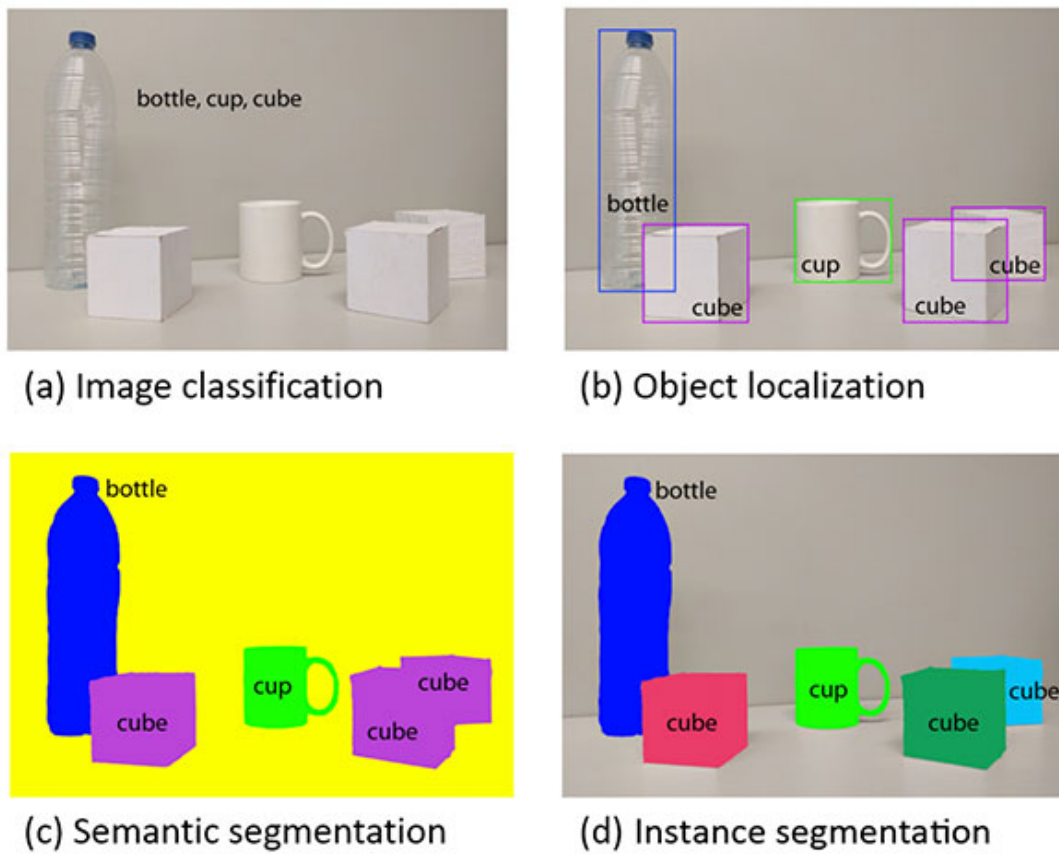


Figure 2.4: Applications of Computer Vision [19]

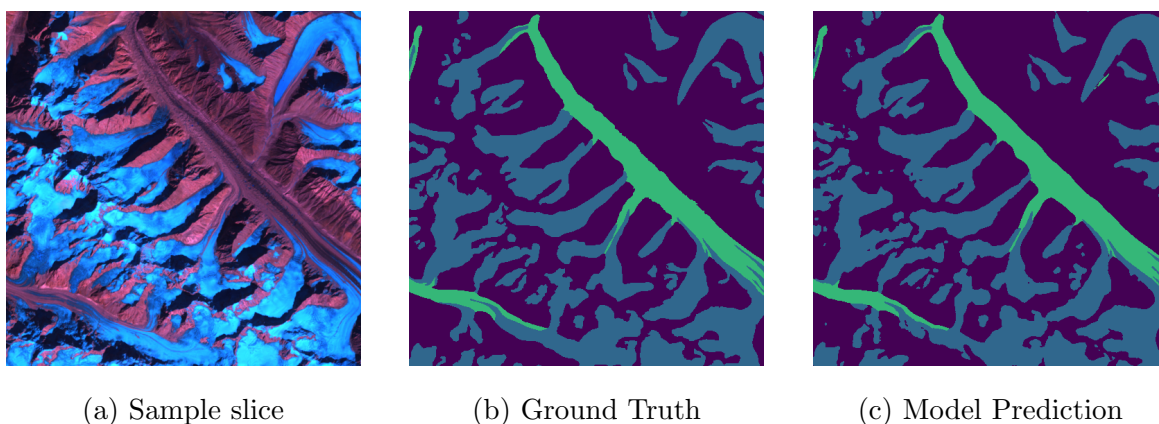


Figure 2.5: Sample Prediction

Table 2.2: Calculation of Mean IoU

Class	Intersecting Area	Union Area	IoU
Clean Ice Glacier	30.10%	35.70%	84.31%
Debris Glacier	7.43%	7.99%	93.02%

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

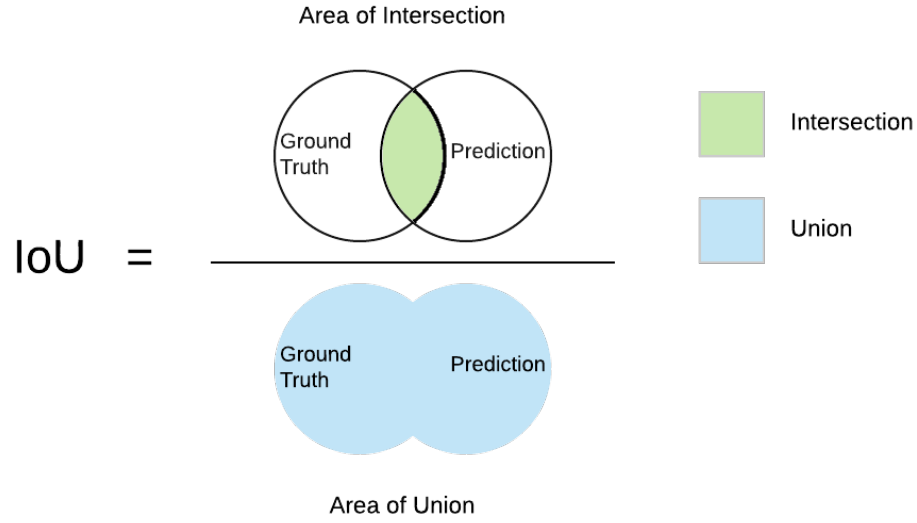


Figure 2.6: Intersection-Over-Union

Mean IoU can be represented as:  $(1/n_{cl}) \sum_i n_{ii} / (t_i + \sum_j n_{ji} - n_{ii})$ , 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  $i$ .



## 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, stationarity 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.

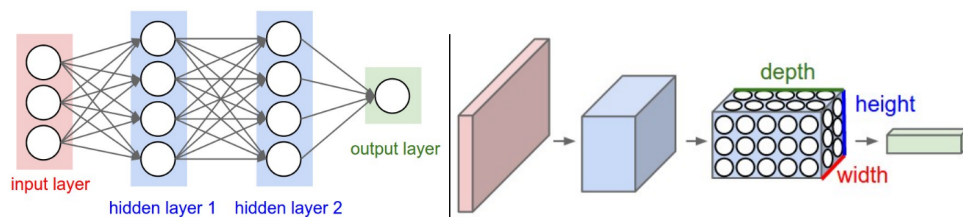


Figure 2.7: Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be number of channels [2].

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 [30]. 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 [45]. The architecture of U-Net is shown in Figure 2.8. The U-Net, being

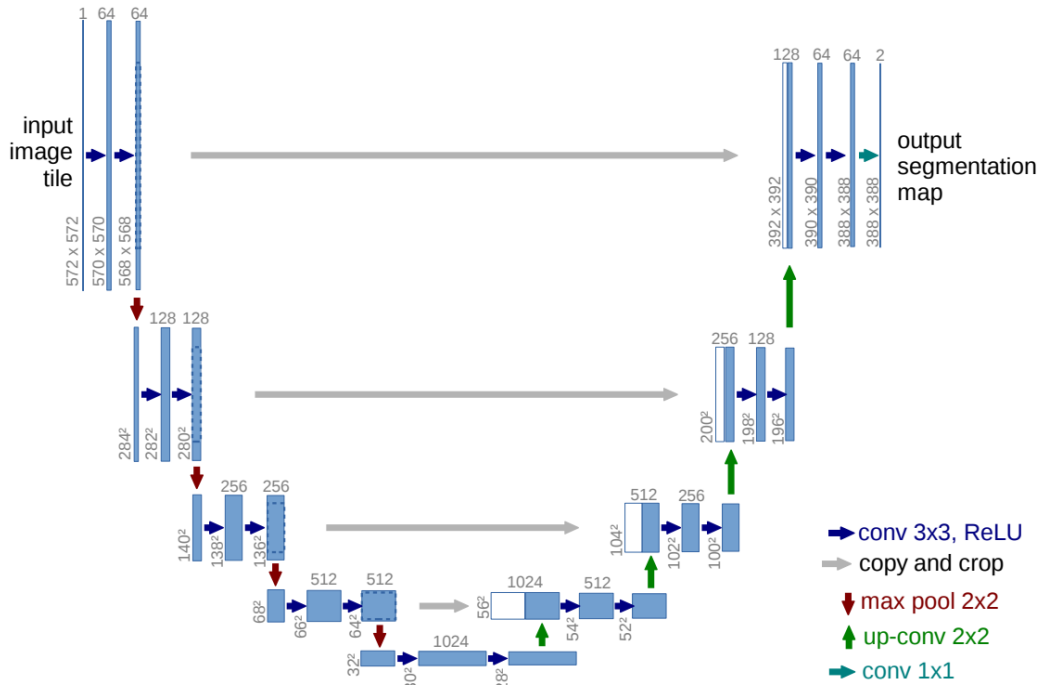


Figure 2.8: U-Net Architecture [45]

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 [35]. 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.

**d) Processed Comparison**

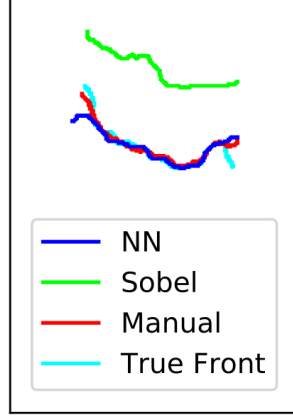


Figure 2.9: Comparison of U-Net Architecture with Other Methods [35]

## 2.5 Active Contour Models (ACMs)

Active Contour Models (ACMs) is a broader term that includes algorithms such as snakes [24], intelligent scissors [36], and level set methods [39]. Among these, snakes has been effectively used to map glacial lakes in Tibetan Plateau [55, 10]. Snakes is an energy-minimizing, two-dimensional spline curve that moves towards features such as edges in images. The internal spline energy function of the snakes algorithm is shown in equation 2.1.

$$\varepsilon_{st} = \int \alpha(s) \|f_s(s)\|^2 + \beta(s) \|f_{ss}(s)\|^2 ds, \quad (2.1)$$

where  $s$  is the arc-length along the curve  $f(s)$ , and  $\alpha(s)$  and  $\beta(s)$  are the first-order and second-order continuity weighting functions that penalize length and curvature of the spline respectively.

In addition to this internal spline energy, snakes can be implemented to simultaneously minimize external image-based potentials [49]. The image-based potentials are sum of several terms given by the equation 2.2.

$$\varepsilon_{image} = \omega_{line}\varepsilon_{line} + \omega_{edge}\varepsilon_{edge} + \omega_{term}\varepsilon_{term} \quad (2.2)$$

where *line* term attracts the snakes to dark ridges, *edge* term attracts the snakes to strong gradients (edges), and *term* term attracts it to line terminations. However, in practice, most systems only use the edge term.

Using a variation of an active contour model, Fang Chen et al. were able to automate the process to generate a glacial lake map in Tibetan Plateau (TP) region using Landsat 8 imagery [10] effectively. The process was carried out for satellite images collected for the year of 2015. The initial data collected through Landsat 8 was divided into 4 test sites. A validation set was created by sampling images of glacial lakes from those four typical regions as samples. This is to ensure that the validation samples were characterized by diverse climatic conditions, physical properties, and surrounding environments. The methodology used by the authors can be divided into three parts.

1. Image preprocessing: During the preprocessing part, the authors first selected the least cloudy image and clipped by glacier buffer. Then, the authors excluded glacial lakes farther than 10km from a glacier terminus due to the assumed weak interaction with glaciers. After that, they masked the topographic shadows and generated initial lake boundaries by calculating Modified Normalized Difference Water Indices (MNDWI). A buffer zone was added around the initial lake boundaries and the image blocks were exported.
2. Regional Implementation of NLAC: The authors implemented NLAC algorithm to generate glacial lake boundaries for the exported image blocks.
3. Data Post-Processing and Validation: The segmentation results were assessed along the metrics kappa coefficients, commission and omission error, and overall accuracy for the four test sites. The vector shapes generated after implementing the NLAC algorithms were then added back to the thematic map.

Figure 2.10 shows an example of an image block for glacial lake extraction using the proposed method. The red contour, the black contour, and the yellow contour indicate the

initial outline obtained by global thresholding, the buffer zone of initial outline, and the final segmentation result after regional implementation of NLAC respectively.

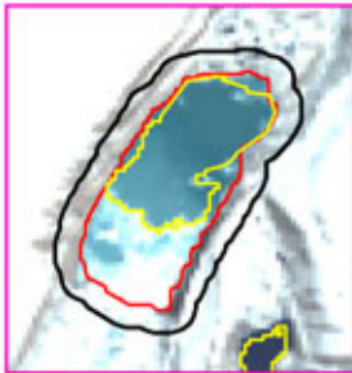


Figure 2.10: Example of an Image Block for Glacial Lake Extraction [10]

The overall accuracy presented in the paper is higher than 91% for all the test sites with Central Himalayas having the lowest overall accuracy of 91.57% and Southeast Tibet having the highest overall accuracy of 97.65%.

## 2.6 Machine Learning Models

Traditional machine learning models in general require less computational power and training time when compared to Deep Learning based approaches. Machine learning based approaches have been shown to work well for land cover mapping problems such as mapping debris glaciers [25], and coastal features [40]. 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 [16].

2. Extreme Boosting(XGBoost): Extreme Boosting, also known as eXtreme Gradient Boosting or XGBoost [12], 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 [17]. 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  $(x_1, x_2, x_3, \dots x_n)$  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.

## 2.7 Summary

Multiple methods have been developed over the years for semantic segmentation such as threshold segmentation, Sobel Filters, and Neural Networks [53]. Semantic segmentation techniques have improved rapidly in recent years due to the progress in deep learning and semantic segmentation with Convolutional Neural Networks (CNNs) [30, 45]. Traditionally, deep neural networks usually require thousands of images as training images where the desired features (in this case glaciers) have already been determined. However, a recent

model, U-Net, [45] has been shown to provide a highly accurate semantic segmentation with as little as 30 samples in the training data.

Various independent researches have shown high accuracy for such automatic methods to map geomorphological features such as glaciers, glacier lakes, and glacier calvings. Traditional machine learning algorithms have shown to be able to delineate glaciers from satellite imagery to some extent [25]. Furthermore, algorithms such as Non Local Active Contour and U-Net, that were primarily developed for biomedical image segmentation, have been shown to scale well for segmentation of remote sensing satellite imagery [10, 45]. We aim to implement these methods to map glaciers in the HKH region and compare among their performances. Additionally, we propose to combine multiple approaches to develop a novel framework optimized for glacier segmentation.

# Chapter 3

## Proposed Research

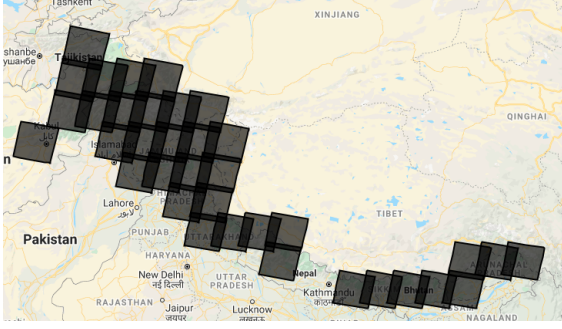
In this chapter, we first describe the dataset and its properties. We then present different models that we used to train on the given dataset. Our proposed research divides the segmentation task into two categories: *Image based Segmentation*, and *Pixel based Segmentation*. We then describe the preprocessing steps on the input data that we implemented in order to make our data compatible with the model.

### 3.1 Data Description

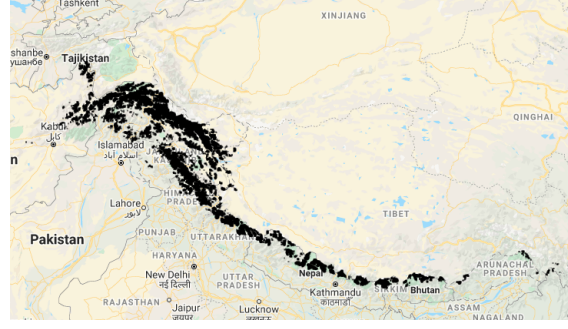
The HKH region covers an area of about 4.2 million km<sup>2</sup> from 15.95° to 39.31° N latitude and 60.85° to 105.04° E longitude extending across eight countries of Afghanistan, Bangladesh, Bhutan, China, India, Myanmar, Nepal and Pakistan [5]. The training data (features) are Landsat 7 satellite imagery queried using Google Earth Engine. The corresponding glacier mappings (labels) are provided by International Centre for Integrated Mountain Development (ICIMOD) [5]. Figure 3.1 shows region bounded by each feature image and the labels in geographical map.

#### 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±3



(a) Landsat 7 image regions



(b) Corresponding Glacier Labels

Figure 3.1: Data in Geographical Context

years. The geographic extent of these glaciers range from  $27.492^\circ$  to  $38.346^\circ$  N latitude and  $67.631^\circ$  to  $98.492^\circ$  E longitude.

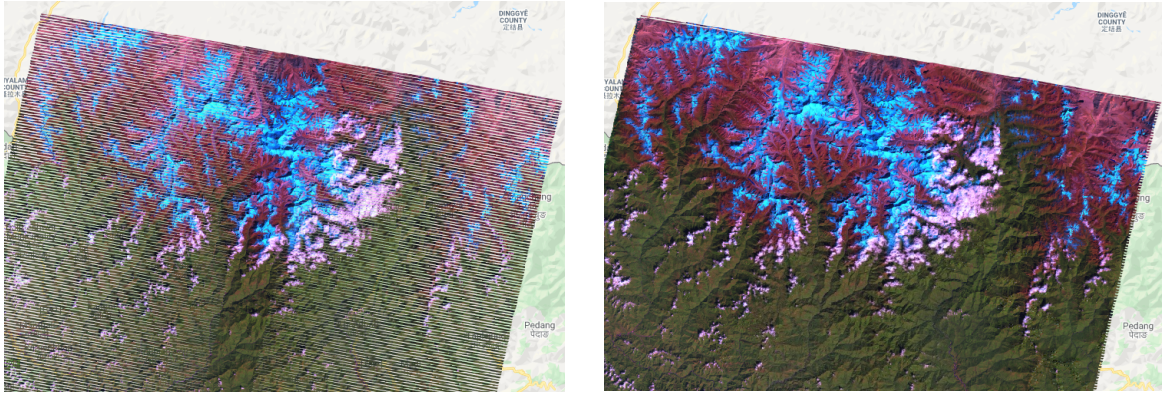
### 3.1.2 Understanding the feature data

The feature data is a collection of 36 scenes from Landsat 7 satellite imagery used for generating glacier labels [5]. 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) [22], Normalized Difference Water Index (NDWI) [18], Normalized Difference Vegetation Index (NDVI) [21], 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 3.1.

$$\begin{aligned}
 NDSI &= \frac{(B5 - B2)}{(B5 + B2)}, \\
 NDWI &= \frac{(B4 - B5)}{(B4 + B5)}, \\
 NDVI &= \frac{(B4 - B3)}{(B4 + B3)},
 \end{aligned} \tag{3.1}$$

### 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+) [11]. Figure 3.2 shows the image before and after SLC line corrector fix for one of the Landsat 7 satellite images.



(a) Image due to SLC failure

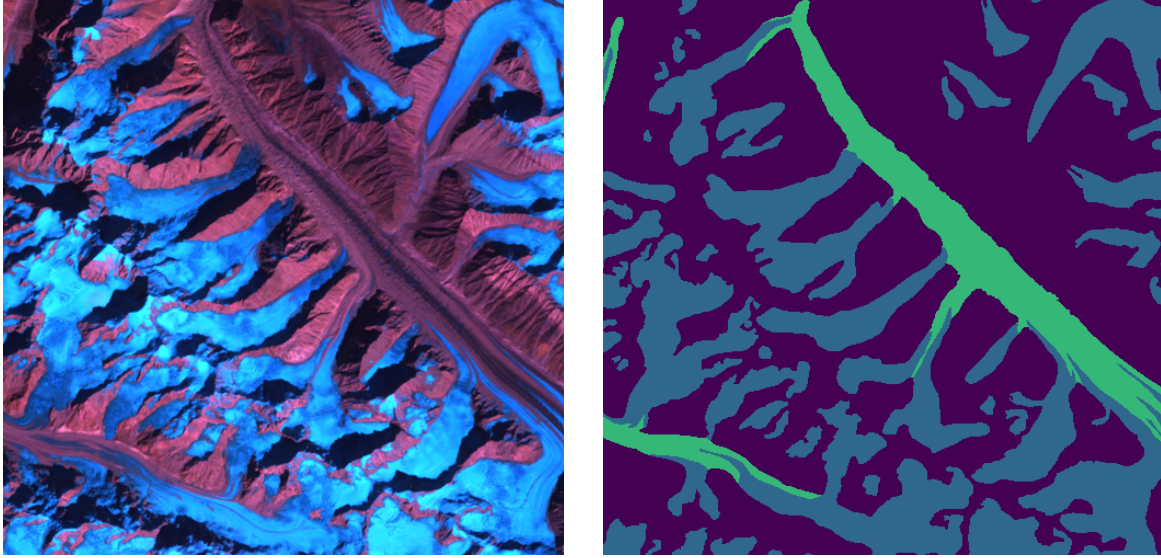
(b) Image after fill

Figure 3.2: Fixing Scan Line Corrector Failure

## 3.2 Image Based Segmentation

In this approach, we sliced the  $\approx 7700 \times 7100$  pixels Landsat 7 scene into 224 images each of size  $512 \times 512$  pixels in order to fit the data into memory for training. A corresponding mask of the given  $512 \times 512$  slice was generated using the given label data. A sliced image and its corresponding mask are shown in Figure 3.3. Due to the property of the U-Net

architecture to accept input data and output labels as images, such prepared data can be directly fed to the model. This allows the model to retain spatial information from the satellite image.



(a) Example image slice

(b) Corresponding mask

Figure 3.3: Sample Sliced Landsat 7 Image and Corresponding Label Mask

Here, pixels with green color denotes debris glaciers, blue color denotes clean ice glaciers, and purple color denotes background.

### 3.2.1 Data Preparation

After slicing the satellite image to the size of  $512 \times 512$ , we performed the following operations on each slice to make it ready to be fed into the model.

1. Filtering: Many of the patches did not contain any glacier labels. In this step, we kept the patches if they contained at least 10% of the pixels as glaciers and discard them otherwise. This is necessary to ensure that the trained model is not skewed towards non-glacier class due to the presence of higher number of samples in the class.
2. Reshuffle: The remaining images and masks were reshuffled.

3. Random Split: The remaining images and their corresponding masks were randomly split into training, testing, and validation groups of 70%-10%-20% respectively.
4. Generate Stats: Next, the mean and standard deviation of the images across each channel/band were calculated and stored. This valuable information is later used to normalize the input image for training purposes.
5. Normalization: The mean and standard deviation that we calculated in the previous step were then used to normalize the final dataset.
6. Imputation: For all the input images, we checked for missing values (NaNs) and replaced them with 0.

A total of 388 unique slices were achieved in the train directory, 110 unique slices in the test directory, and 55 unique slices in the validation directory at the end of this preprocessing step.

### 3.2.2 Model Architecture

For training such prepared data, we used a modified variant of U-Net architecture discussed in Section 2.4.2. We performed experiments with different variations of the U-Net architecture by changing the number of channels in each layer, by changing the depth of the network (total number of downsampling and its corresponding upsampling layers), and by introducing dropouts in between the convolution layers. We then selected the best performing model architecture.

## 3.3 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 2.6 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.

### 3.3.1 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 3.1.

Table 3.1: Data Distribution for Pixel Based Segmentation

<b>Label</b>	<b>Training</b>	<b>Validation</b>	<b>Testing</b>	<b>Total</b>
Clean Ice Glaciers	3422713	905140	2200000	6527853
Debris Glaciers	3422713	905140	622072	4949925
Background	3422713	905140	2200000	6527853
<b>Total</b>	10268139	2715420	5022072	<b>18005631</b>



### 3.3.2 Machine Learning Models

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

### 3.3.3 Evaluation Metrics

We perform evaluation on the test set along the metrics [47] described in Table 3.2 when searching for best hyperparameters. We calculate the metrics averaged across different

Table 3.2: Machine Learning Models Evaluation Metrics

Evaluation Metrics	Formula
Accuracy (Acc)	$(TP+TN)/N$
Precision	$TP/(TP+FP)$
Recall	$TP/(TP+FN)$
F score	$\frac{2*Precision*Recall}{(Precision+Recall)}$

*\*TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative, N = Total Number of Samples in Test Set*

classes weighted by the number of true instances for those classes using formula 3.2.

$$weighted_m = \frac{\sum_{i=1}^n w_i * m_i}{\sum_{i=1}^n w_i} \quad (3.2)$$

Here,  $weighted_m$  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  $i$ .

## 3.4 Conclusion

The main objective of this chapter is to design experiments that will be adequate for answering our research question; *“weather it is possible to accurately delineate glacier boundary in an automated way using conventional machine learning and deep learning methods”*. We aim to answer the question by

1. presenting an automated, end-to-end, reproducible pipeline that is designed to delineate glacier boundaries, and
2. analyzing results to compare conventional machine learning models and deep learning architectures on the same test set.

# Chapter 4

## Preliminary Results

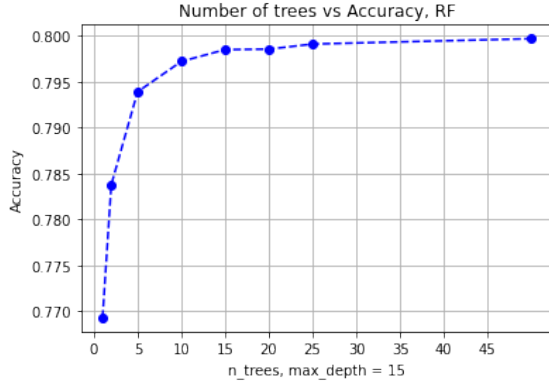
In this chapter we present some experiments for training models to generate segmentation masks for glaciers using the approaches mentioned in section 3.2 and section 3.3. We then generate and present some segmentation masks for image slices in the test set using these trained models. Lastly, we present quantitative evaluations of these masks generated using different models and compare them.

### 4.1 Results: Pixel Based Segmentation

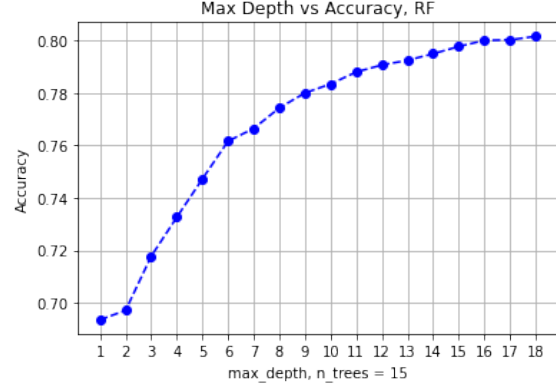
In this approach, we considered each band of the satellite imagery as an individual feature each belonging to one of the labels “Debris Glaciers”, “Clean Ice Glaciers”, or “Background” on machine learning models mentioned in section 2.6. We then found the optimal values of hyperparameters for each model through evaluation on validation set. Finally, we computed our metrics on the test set and report the performance of the model.

#### 4.1.1 Random Forest

Using a tree with high number of features generally seemed to perform better for our task of classification. However due to the nature of random forest, the improvement in model performance is relatively small after a certain size of the tree. In order to maximize efficiency of the model with respect to its size, we visualized the accuracy of the model with incremental increase in number of trees and depth of the model as shown in Figure 4.1. Here, for Figure 4.1a, we gradually increased the number of trees for the model keeping maximum depth constant at 15. We observed little improvement on accuracy when increasing the



(a) Tuning Number of Trees for Random Forest



(b) Tuning Depth for Random Forest

Figure 4.1: Tuning Parameters for Random Forest

number of trees beyond 15. The number of parameters increased linearly with increase in number of trees for random forest as seen from Table 4.1. Similarly for Figure 4.1b, we gradually increased the maximum depth of the model keeping the number of trees constant at 15. Once again, we observed a point of diminishing return after maximum depth of 15. Furthermore, the number of parameter increases by a factor of  $2^n$  with increase in maximum depth for the random forest model as seen from Table 4.2. We then trained a

Table 4.1: Random Forest Number of Trees vs Nodes (Maximum Depth 15)

Number of trees	1	2	5	10	15	20	25	50
Number of nodes	19123	38366	97769	198170	293379	387372	486447	981752

Table 4.2: Random Forest Maximum Depth vs Nodes (Number of Trees 15)

Max depth	1	5	10	12	13	14	15	16	17	18
Nodes	45	945	26971	82095	129715	199547	293379	403069	542655	691577

random forest model with 15 estimators and depth of 15 using data in the training sample.

The performance of random forest model on test data is shown in Table 4.3.

Table 4.3: Experimental Result for Random Forest

Accuracy	Precision	Recall	F-Score	Train Acc.	Train Time
80.19%	81.61%	80.19%	80.47%	83.46%	5.90 minutes

We observed an accuracy of 80.19% on the training set and 81.61% on the test set. Furthermore, we observed an accuracy of 82.22% on the test set using model with number of trees 50 and maximum depth of 50. However, there is a cost associated with this increase in accuracy. This larger model takes 2.65 times longer to train (5.90 minutes vs 15.6 minutes) and is about 77.8 times larger to store (25 MB vs 1.9 GB). The confusion matrix for the Random Forest model trained on maximum depth of 15 and 15 trees is shown in Table 4.4. The feature importance scores on the random forest model with top

Table 4.4: Confusion Matrix for Random Forest

n=5022072		Predicted			
		Clean Ice	Debris	Background	Total
True	Clean Ice	1740053	188090	271857	2200000
	Debris	29750	554689	37633	622072
	Background	292945	174776	1732279	2200000
	Total	2062748	917555	2041769	

10 most important features are shown in Figure 4.2. The top 3 features with highest scores are “slope”, “elevation”, and “NDSI”.

Prediction on example image slice using the trained random forest model is shown in Figure 4.3.

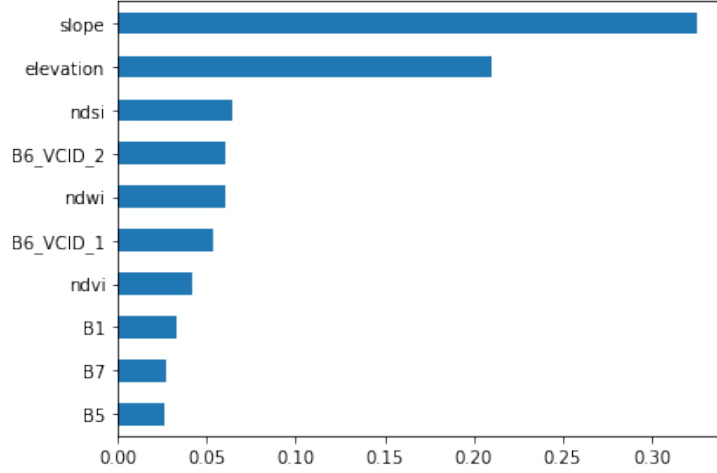
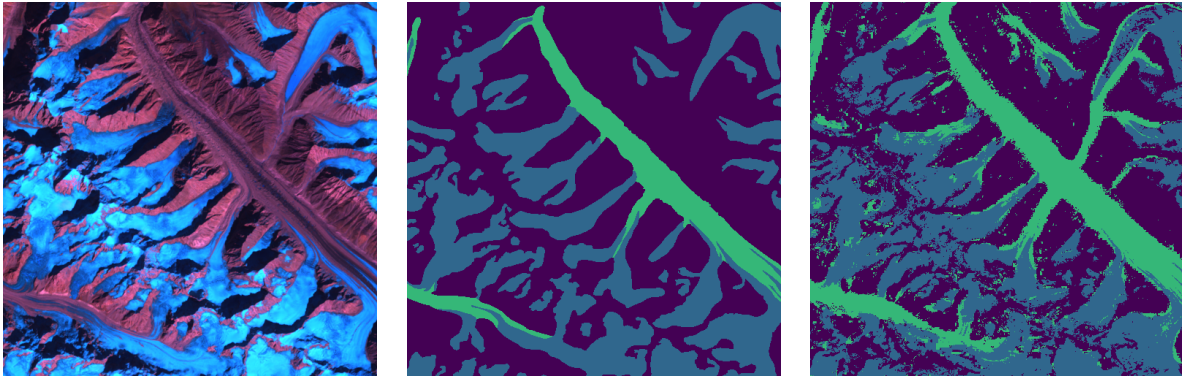


Figure 4.2: Feature Importance, Random Forest



(a) Example Image Slice

(b) Ground Truth

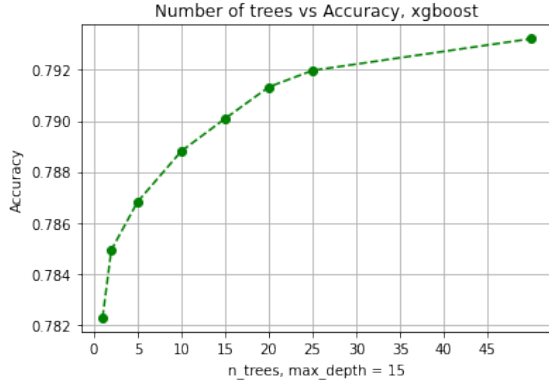
(c) Predicted Mask

Figure 4.3: Sample Prediction using Random Forest

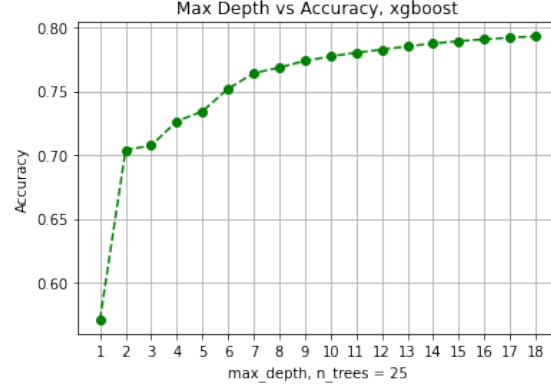
### 4.1.2 Extreme Boosting (XGBoost)

Similar to random forest algorithm, the improvement in model performance for XGBoost is also small beyond a certain size. In order to maximize efficiency of the model with respect to its size, we visualized the accuracy of the model with incremental increase in number of trees and depth of the model as shown in Figure 4.1.

Here, for Figure 4.4a, we gradually increased the number of trees for the model keeping



(a) Tuning Number of Trees for XGBoost



(b) Tuning Depth for XGBoost

Figure 4.4: Tuning Parameters for Extreme Boosting

maximum depth constant at 15. Similarly for Figure 4.4b, we gradually increased the maximum depth of the model keeping the number of trees constant at 25. The learning rate is kept constant at 0.01 for all of the experiments.

Based on the graphs, we selected a threshold of 14 for maximum depth and 25 for maximum number of trees for XGBoost. Like random forest, the number of parameters increased linearly with increase in number of trees and increased by a factor of  $2^n$  with increase in maximum depth for XGBoost. Our observation for various models with different number of parameters to see the performance with respect to the size and training time is summarized in Table 4.5.

We observed that the size of the model increases significantly with the increase in maximum depth and the training time increases when adding number of trees. The detailed performance of XGBoost model on test data using 25 trees with depth 14 is shown in Table 4.6.

We also observed an accuracy of 84.76% on the training set and 80.60% on the test set using XGBoost. The confusion matrix for the XGBoost model is shown in Table 4.7.

The feature importance scores on the XGBoost model are shown in Figure 4.5. We observed that 8 out of the top 10 most important features are same for both XGBoost and

Table 4.5: Extreme Boosting Parameters Tuning

Max. Depth	Num. Trees	Accuracy	F-Score	Train Time	Model Size
10	50	78.00%	78.21%	1.71 hours	9.7 MB
10	100	78.27%	78.47%	3.46 hours	20 MB
16	10	79.18%	79.36%	34 minutes	44 MB
18	20	79.89%	80.07%	1.29 hours	181 MB
50	50	82.27%	82.40%	7.12 hours	2.2 GB

Table 4.6: Experimental Result for Extreme Boosting

Accuracy	Precision	Recall	F-Score	Train Acc.	Train Time
80.60%	82.06%	80.60%	80.87%	84.76%	2.54 hours

Table 4.7: Confusion Matrix for Extreme Boosting

n=5022072		Predicted			
		Clean Ice	Debris	Background	Total
<b>True</b>	Clean Ice	1736951	189282	273767	2200000
	Debris	27761	563121	31190	622072
	Background	279626	172845	1747529	2200000
	Total	2044338	925248	2052486	

random forest model. Furthermore, the top 3 features “NDSI”, “slope”, and “elevation” are same for both models albeit the scores being different.

Prediction on example image slice using the trained XGBoost model is shown in Figure 4.6.



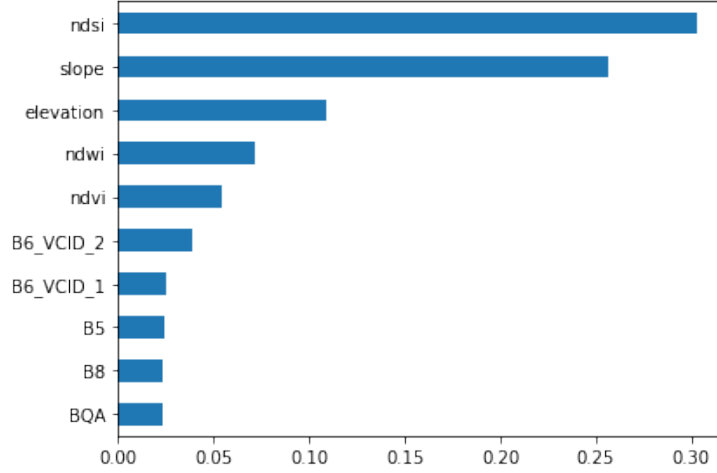
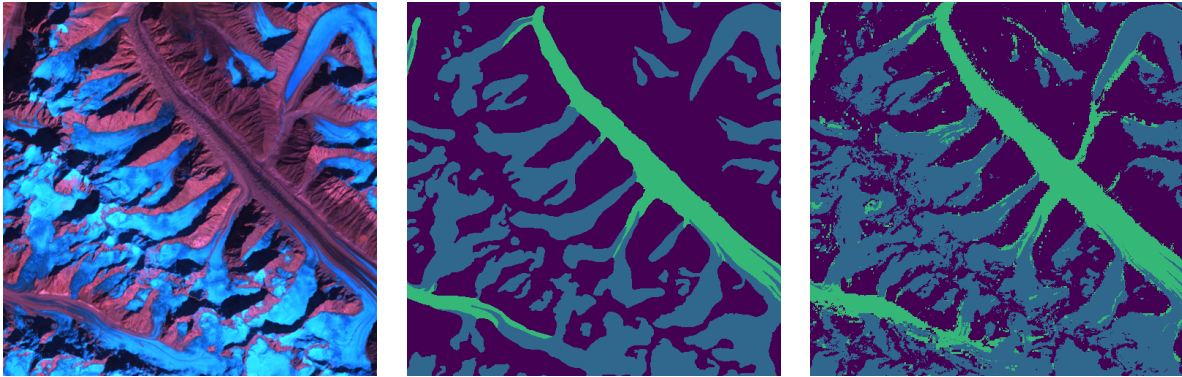


Figure 4.5: Feature Importance, Extreme Boosting



(a) Example Image Slice

(b) Ground Truth

(c) Predicted Mask

Figure 4.6: Sample Prediction using XGBoost

### 4.1.3 Multi Layered Perceptrons (MLP)

We first determined the optimal parameters for the classifier using grid search. The optimal parameters were found to be: optimizer, adam; learning rate,  $e^{-5}$ ; three hidden layers with the number of neurons on each layer 128, 256, and 128 respectively; number of iteration, 250; and L2 regularization parameter, 1. We then trained the MLP with optimal parameters. The performance of MLP model on test data is shown in Table 4.8.

Table 4.8: Experimental Result for Multi Layered Perceptrons

Accuracy	Precision	Recall	F-Score	Train Acc.	Train Time
76.99%	78.63%	76.99%	77.32%	77.85%	15.04 hours

Table 4.9: Confusion Matrix for Multi Layered Perceptron

n=5022072		Predicted			
		Clean Ice	Debris	Background	Total
<b>True</b>	Clean Ice	1695816	219228	284956	2200000
	Debris	44372	532768	44932	622072
	Background	372218	190093	1637689	2200000
	Total	2112406	942089	1967577	

We observed that the accuracy of the MLP model is lower than random forest and XGBoost. However, the final size of the trained model is the smallest for MLP among the ones we observed. The trained model using MLP is about 2.1 MB in size while that of XGBoost is about 2.2 GB. We also observed that the training time of MLP was 1.5 times lower than that of XGBoost. The confusion matrix for MLP model is shown in Table 4.9.

Prediction on example image slice using the trained multi layered perceptron model is shown in Figure 4.7.

## 4.2 Results: Image Based Segmentation

After preparing the data as mentioned in section 3.2.1, we trained with a modified variant of the U-Net architecture to evaluate the performance of the model.

We trained the U-Net model with Adam as the learning algorithm and weighted dice loss as our loss function. We used an initial learning rate of  $e^{-4}$  and decreased the learning rate by 0.1 for no change in validation loss for 20 epochs until the minimum learning rate

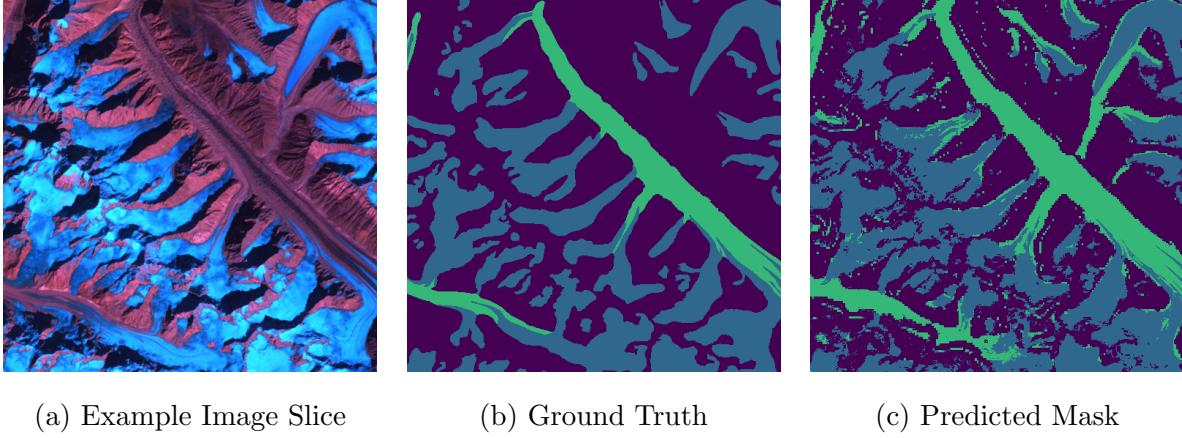


Figure 4.7: Sample Prediction using Multi Layered Perceptron

of  $e^{-6}$  is reached. We trained the model for 250 epochs with a batch size of 16. Spatial dropout with dropout probability 0.3 is introduced after the second convolution layer in each level to prevent the model from overfitting.

Dice loss originates from a statistics developed in 1940s, called Sørensen–Dice coefficient, to gauge the similarity between two sample. Milletari et al. later brought it to computer vision community for 3D medical image segmentation [34]. As there are fewer pixels corresponding to the debris glaciers labels, the model, during training, tends to overlook debris glaciers. As a solution to this, we use weighted dice loss and assigned the debris glaciers class with higher weight. The weighted dice loss is shown in the equation 4.1.

$$Loss(labels, preds) = 1 - \sum_{i=1}^c \frac{w_i * 2 * \sum_{j=1}^N labels_{ij} * preds_{ij}}{\sum_{j=1}^N labels_{ij}^2 + \sum_{j=1}^N preds_{ij}^2} \quad (4.1)$$

Here,  $c$  is the number of classes in output labels and  $w_i$  is the weight for class  $i$ . We weigh classes clean ice, debris glaciers, and background by weights 0.9, 0.6, and 0.2 respectively.

Dropout [48] is a commonly used technique to prevent the model from overfitting. Overfitting in deep learning is a term used to denote when the model performs exceptionally well on training data but poorly in the validation data. This means, instead of learning features of the training data, the model is memorizing the training data itself. Dropout is used to prevent overfitting by dropping out neurons from layers randomly during the

training phase. Spatial Dropout, [50] unlike regular dropouts, randomly sets entire feature maps (channels) to 0, rather than individual ‘pixels’. This seems to work better for images as adjacent pixels are highly correlated. This in-turn prevents the model from memorizing the training data.

Prediction on example image slice using the trained U-Net model is shown in Figure 4.8.

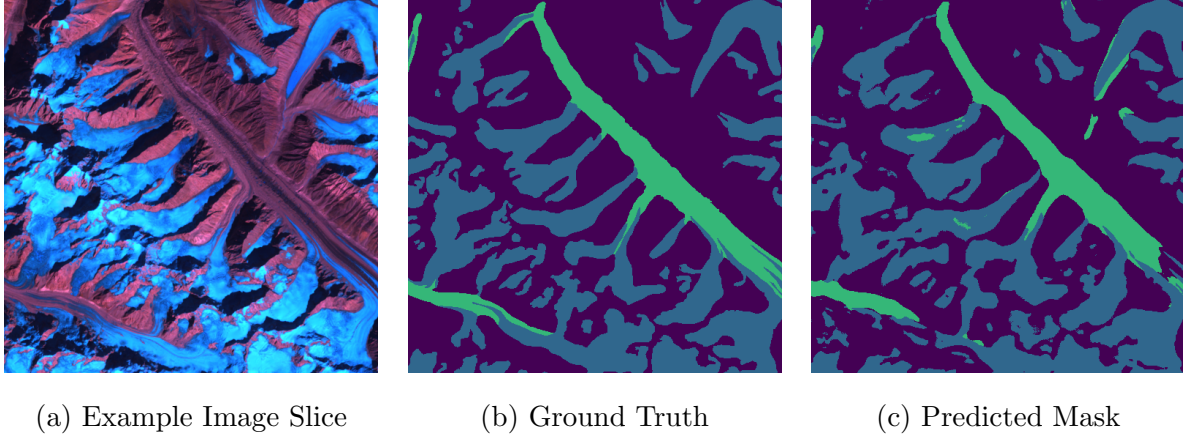


Figure 4.8: Sample Prediction using U-Net Model

We see that the segmentation mask generated by the U-Net model is not fragmented as in pixel based classifiers. This can be attributed to the property of CNNs to consider neighboring pixels.

### 4.3 Combined Results

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 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 4.10.

Furthermore, we also observe the performance of models on clean ice glaciers and debris ice glaciers separately. This comparison is shown in table 4.11.

Table 4.10: Comparison Between Different Models

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

Table 4.11: Performance comparison for clean ice and debris glaciers

Model	Clean Ice Glaciers			Debris Glaciers		
	IoU	Precision	Recall	IoU	Precision	Recall
Random Forest	58.07%	66.36%	82.29%	20.24%	20.71%	89.90%
XGBoost	56.63%	69.50%	77.15%	19.30%	19.67%	91.13%
MLP	54.52%	64.60%	77.74%	17.81%	18.31%	86.73%
U-Net	58.29%	74.24%	73.07%	37.07%	48.77%	60.71%

## 4.4 Generated Segmentation Masks

In this section, we present some segmentation masks that were generated using the trained model for the test set for qualitative evaluation of the segmentation masks. Figure 4.9 shows some of the example slices in the test data and Figure 4.10 shows the corresponding labels for these images. Examples of segmentation masks generated using random forest, XGBoost, MLP, and U-Net on these slices are shown in Figure 4.11, Figure 4.12, Figure 4.13, and Figure 4.14 respectively.

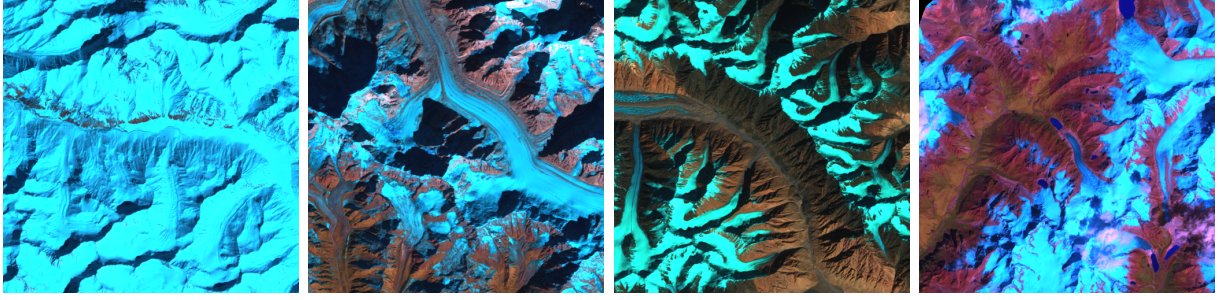


Figure 4.9: Example Image Slices

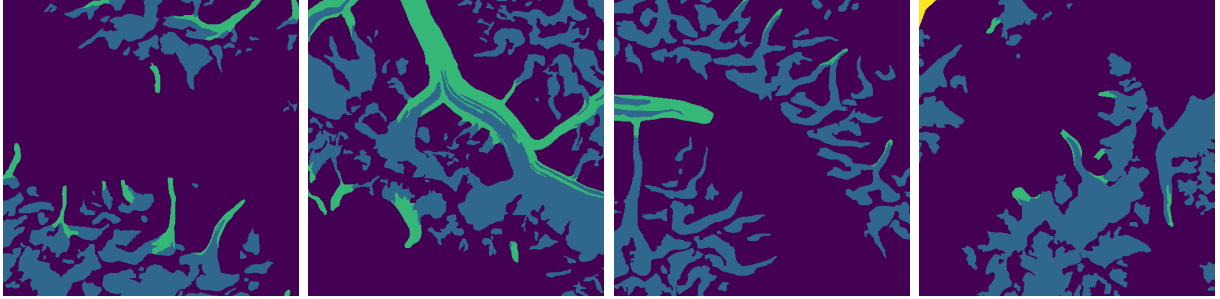


Figure 4.10: Corresponding Labels

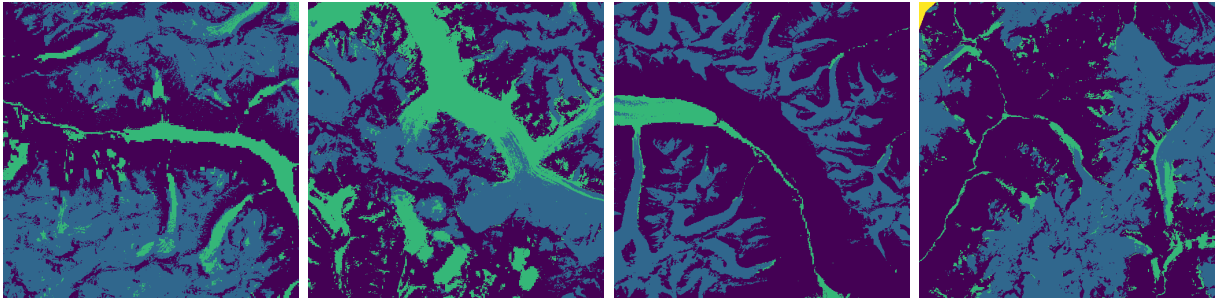


Figure 4.11: Prediction Mask using Random Forest

## 4.5 Significance of the Result

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



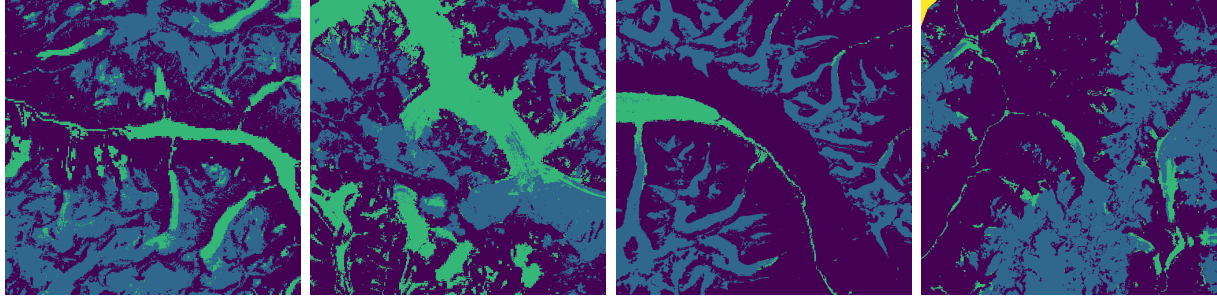


Figure 4.12: Prediction Mask using Extreme Boosting

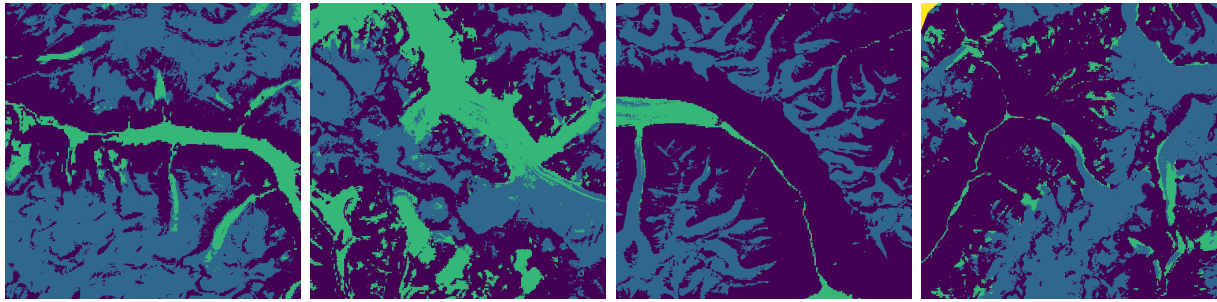


Figure 4.13: Prediction Mask using Multi Layer Perceptron

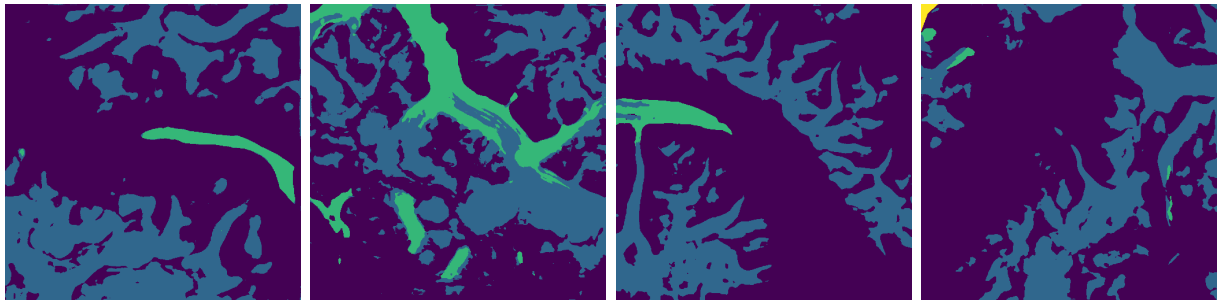


Figure 4.14: Prediction Mask using U-Net

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. Qualitatively, the segmentation masks generated using U-Net seems to be better than the ones generated using pixel based methods.

We must also add that we have not performed exhaustive search of optimal hyperparameters in the case of U-Net model like we did for random forest and extreme boosting. Further improvements in results may be observed through hyperparameter tuning.



# Chapter 5

## Research Plan and Timeline

In this chapter, we present the proposed objectives of this work and the approximate timeline to complete them.

### 5.1 Proposed Work

Now that we have shown that it is possible to generate accurate segmentation masks for glaciers using satellite images, this opens up a number of research opportunities. Exploring these opportunities, throughout the duration of this project, we will be working on the following activities.

#### Experiments

1. The segmentation mask can be made better by using post processing techniques such as erosion, dilation, connected components, and conditional random fields [26]. We will be experimenting with these postprocessing techniques to see if the quality of the segmentation mask can be improved.
2. We will experiment with NOAA airborne imagery on the region of Eastern Beaufort Sea Coast, Alaska for coastal feature segmentation using the same methodology to test the generalization capability across different images and for different labels.
3. Active Contour Models have been effectively used by themselves [10, 54] or by integrating them with deep learning based approaches [23, 33] for land-cover mapping in satellite imagery. Combination of deep learning and contour models have also

been shown to perform very well for the general segmentation tasks [42]. We will be working on applying the active contour model for generating segmentation mask and comparing the results with U-Net model. We will also be exploring at the ways to combine the expressiveness of deep Neural Networks with the versatility of ACMs in a unified framework.

## Literature Review

1. Throughout the development of this project, we will be constantly reviewing the literature in order to discover possible techniques or approaches that might be integrated to this project.

## Publications

1. Machine Learning for Glacier Monitoring in the Hindu Kush Himalaya (NeurIPS 2020 Workshop Tackling Climate Change with Machine Learning, Accepted): In this work, we present a method based on U-Net architecture to support ecological monitoring with a focus on glaciers. We present an automated method to outline both clean ice and debris glaciers from readily available Landsat 7 satellite images. We also release data and develop a web tool that allows experts to visualize and correct model predictions, with the ultimate aim of accelerating the glacier mapping process.
2. Coastal Change detection using Landcover Mapping on airborne images (Progress 10%): In this work, we aim to extract coastal features from NOAA airborne imagery on the region of Eastern Beaufort Sea Coast, Alaska using machine learning approaches. We aim to use the trained model to track coastal erosion in the region.
3. Generative adversarial network for super-resolution of satellite images (In Progress).

## 5.2 Timeline

My research plan for this project summarizing all the related activities is shown in the Figure 5.1.

Major Activities	2018 Fall	2019 Spring	2019 Summer	2019 Fall	2020 Spring	2020 Summer	2020 Fall	2021 Spring	2021 Summer	2021 Fall	2022 Spring
Course work and degree requirements											
Graduate school acceptance	X										
Course work	X	X		X			X	X		X	
Qualifying Exam			X								
Paper submissions and internships											
Internship						X			X		
Literature Review					X	X	X	X	X	X	X
Paper submission							X	X		X	
Ph.D. Dissertation											
Proposal Defense							X				
Run Experiments					X		X	X		X	X
Analyze results							X	X		X	X
Dissertation writeup								X		X	X
Dissertation defense											X

Figure 5.1: Timeline

# References

- [1] Access to Data — Copernicus. <https://www.copernicus.eu/en/access-data>. (Accessed on 12/10/2019).
- [2] CS231n Convolutional Neural Networks for Visual Recognition. <http://cs231n.github.io/convolutional-networks/>. (Accessed on 12/10/2019).
- [3] Randolph Glacier Inventory – A Dataset of Global Glacier Outlines: Version 6.0: Technical Report.
- [4] The World’s Third Pole Is Melting – The Diplomat. <https://thediplomat.com/2019/03/the-worlds-third-pole-is-melting/>:text=The (Accessed on 11/07/2020).
- [5] BAJRACHARYA, S. R., AND SHRESTHA, B. R. The status of glaciers in the Hindu Kush-Himalayan region. Tech. rep., International Centre for Integrated Mountain Development (ICIMOD), 2011.
- [6] BARNETT, T. P., ADAM, J. C., AND LETTENMAIER, D. P. Potential impacts of a warming climate on water availability in snow-dominated regions. *Nature* *438*, 7066 (2005), 303–309.
- [7] BENISTON, M. Climatic change in mountain regions: a review of possible impacts. In *Climate variability and change in high elevation regions: Past, present & future*. Springer, 2003, pp. 5–31.
- [8] BOOKHAGEN, B., AND BURBANK, D. W. Toward a complete Himalayan hydrological budget: Spatiotemporal distribution of snowmelt and rainfall and their impact on river discharge. *Journal of Geophysical Research: Earth Surface* *115*, F3 (2010).

- [9] BRICE, C. R., AND FENNEMA, C. L. Scene analysis using regions. *Artificial intelligence* 1, 3-4 (1970), 205–226.
- [10] CHEN, F., ZHANG, M., TIAN, B., AND LI, Z. Extraction of Glacial Lake Outlines in Tibet Plateau Using Landsat 8 Imagery and Google Earth Engine. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10, 9 (2017), 4002–4009.
- [11] CHEN, J., ZHU, X., VOGELMANN, J. E., GAO, F., AND JIN, S. A simple and effective method for filling gaps in landsat ETM+ SLC-off images. *Remote Sensing of Environment* 115, 4 (2011), 1053–1064.
- [12] CHEN, T., HE, T., BENESTY, M., KHOTILOVICH, V., AND TANG, Y. Xgboost: extreme gradient boosting. *R package version 0.4-2* (2015), 1–4.
- [13] CRUZ, R., HARASAWA, H., LAL, M., WU, S., ANOKHIN, Y., PUNSALMAA, B., HONDA, Y., JAFARI, M., LI, C., AND NINH, N. Asia. Climate change 2007: Impacts, adaptation and vulnerability. *Contribution of Working Group II to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change* (08 2007), 469–506.
- [14] DUNN, M. *Exploring your world: The adventure of geography*. National Geographic Society, 1989.
- [15] DYHRENFURTH, G. O. *To the Third Pole-The History of the High Himalaya*. Nielsen Press, 2011.
- [16] FRIEDMAN, J., HASTIE, T., AND TIBSHIRANI, R. *The elements of statistical learning*, vol. 1. Springer series in statistics New York, 2001.
- [17] FRIEDMAN, J., HASTIE, T., TIBSHIRANI, R., ET AL. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics* 28, 2 (2000), 337–407.

- [18] GAO, B.-C. NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. *Remote sensing of environment* 58, 3 (1996), 257–266.
- [19] GARCIA-GARCIA, A., ORTS-ESCOLANO, S., OPREA, S., VILLENA-MARTINEZ, V., AND GARCIA-RODRIGUEZ, J. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857* (2017).
- [20] GORELICK, N., HANCHER, M., DIXON, M., ILYUSHCHENKO, S., THAU, D., AND MOORE, R. Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment* (2017).
- [21] GOWARD, S. N., MARKHAM, B., DYE, D. G., DULANEY, W., AND YANG, J. Normalized difference vegetation index measurements from the Advanced Very High Resolution Radiometer. *Remote sensing of environment* 35, 2-3 (1991), 257–277.
- [22] HALL, D. K., AND RIGGS, G. A. Normalized-difference snow index (NDSI).
- [23] HATAMIZADEH, A., SENGUPTA, D., AND TERZOPOULOS, D. End-to-end deep convolutional active contours for image segmentation. *arXiv preprint arXiv:1909.13359* (2019).
- [24] KASS, M., WITKIN, A., AND TERZOPOULOS, D. Snakes: Active contour models. *International journal of computer vision* 1, 4 (1988), 321–331.
- [25] KHAN, A. A., JAMIL, A., HUSSAIN, D., TAJ, M., JABEEN, G., AND MALIK, M. K. Machine-learning algorithms for mapping debris-covered glaciers: the Hunza Basin case study. *IEEE Access* 8 (2020), 12725–12734.
- [26] KRÄHENBÜHL, P., AND KOLTUN, V. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems* (2011), pp. 109–117.

- [27] KULKARNI, A. Glacial retreat in Himalaya using Indian remote sensing satellite data. *Proceedings of SPIE - The International Society for Optical Engineering* (12 2006).
- [28] KULKARNI, A. V., BAHUGUNA, I., RATHORE, B., SINGH, S., RANDHAWA, S., SOOD, R., AND DHAR, S. Glacial retreat in Himalaya using Indian remote sensing satellite data. *Current science* (2007), 69–74.
- [29] KULKARNI, A. V., AND KARYAKARTE, Y. Observed changes in Himalayan glaciers. *Current Science* (2014), 237–244.
- [30] LONG, J., SHELHAMER, E., AND DARRELL, T. Fully Convolutional Networks for Semantic Segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015).
- [31] LOVELAND, T. R., AND DWYER, J. L. Landsat: Building a strong future. *Remote Sensing of Environment* 122 (2012), 22–29.
- [32] MADAN, M. River valley development: the hydroelectric power option in Parbati basin-technological, environmental and economical aspects of Parbati hydroelectric project, stage ii. *Indian Journal of Power and River Valley Development* 55, 3/4 (2005), 48.
- [33] MARCOS, D., TUIA, D., KELLENBERGER, B., ZHANG, L., BAI, M., LIAO, R., AND URTASUN, R. Learning deep structured active contours end-to-end. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 8877–8885.
- [34] MILLETARI, F., NAVAB, N., AND AHMADI, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)* (2016), IEEE, pp. 565–571.

- [35] MOHAJERANI, Y., WOOD, M., VELICOGNA, I., AND RIGNOT, E. Detection of Glacier Calving Margins with Convolutional Neural Networks: A Case Study. *Remote Sensing* 11, 1 (2019), 74.
- [36] MORTENSEN, E. N., AND BARRETT, W. A. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), pp. 191–198.
- [37] NEMANI, R., VOTAVA, P., MICHAELIS, A., MELTON, F., AND MILESI, C. Collaborative supercomputing for global change science. *Eos, Transactions American Geophysical Union* 92, 13 (2011), 109–110.
- [38] OHLANDER, R., PRICE, K., AND REDDY, D. R. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing* 8, 3 (1978), 313–333.
- [39] OSHER, S., AND PARAGIOS, N. *Geometric level set methods in imaging, vision, and graphics*. Springer Science & Business Media, 2003.
- [40] PARK, S. J., ACHMAD, A. R., SYIFA, M., AND LEE, C.-W. Machine learning application for coastal area change detection in gangwon province, South Korea using high-resolution satellite imagery. *Journal of Coastal Research* 90, SI (2019), 228–235.
- [41] PAVLIDIS, T., ET AL. Polygonal approximations by Newton’s method. *IEEE Trans. Computers* 26, 8 (1977), 800–807.
- [42] PENG, S., JIANG, W., PI, H., LI, X., BAO, H., AND ZHOU, X. Deep Snake for Real-Time Instance Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8533–8542.
- [43] RABATEL, A., FRANCOU, B., SORUCO, Á., GOMEZ, J., CÁCERES, B., CEBALLOS, J., BASANTES, R., VUILLE, M., SICART, J.-E., HUGGEL, C., ET AL. Current state



of glaciers in the tropical Andes: a multi-century perspective on glacier evolution and climate change. *The Cryosphere* 7, 1 (2013), 81–102.

- [44] RISEMAN, E. M., AND ARBIB, M. A. Computational techniques in the visual segmentation of static scenes. *Computer Graphics and Image Processing* 6, 3 (1977), 221–276.
- [45] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (Cham, 2015), N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Springer International Publishing, pp. 234–241.
- [46] ROSENFELD, A., AND DAVIS, L. Image segmentation and image model. *Proceedings of IEEE* 67, 5 (1979), 764–772.
- [47] SOKOLOVA, M., AND LAPALME, G. A systematic analysis of performance measures for classification tasks. *Information Processing Management* 45, 4 (2009), 427 – 437.
- [48] SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., AND SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.
- [49] SZELISKI, R. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [50] TOMPSON, J., GOROSHIN, R., JAIN, A., LECUN, Y., AND BREGLER, C. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 648–656.
- [51] WANG, W., XIANG, Y., GAO, Y., LU, A., AND YAO, T. Rapid expansion of glacial lakes caused by climate and glacier retreat in the Central Himalayas. *Hydrological Processes* 29, 6 (2015), 859–874.

- [52] WOODCOCK, C. E., ALLEN, R., ANDERSON, M., BELWARD, A., BINDSCHADLER, R., COHEN, W., GAO, F., GOWARD, S. N., HELDER, D., HELMER, E., ET AL. Free access to Landsat imagery. *Science* 320, 5879 (2008), 1011–1011.
- [53] ZAITOUN, N. M., AND AQEL, M. J. Survey on Image Segmentation Techniques. *Procedia Computer Science* 65 (2015), 797 – 806. International Conference on Communications, management, and Information technology (ICCMIT’2015).
- [54] ZHANG, G., YAO, T., XIE, H., WANG, W., AND YANG, W. An inventory of glacial lakes in the Third Pole region and their changes in response to global warming. *Global and Planetary Change* 131 (2015), 148–157.
- [55] ZHANG, M.-M., CHEN, F., AND TIAN, B.-S. An automated method for glacial lake mapping in High Mountain Asia using Landsat 8 imagery. *Journal of Mountain Science* 15, 1 (2018), 13–24.

# Chapter 6

## Appendix: Segmentation masks

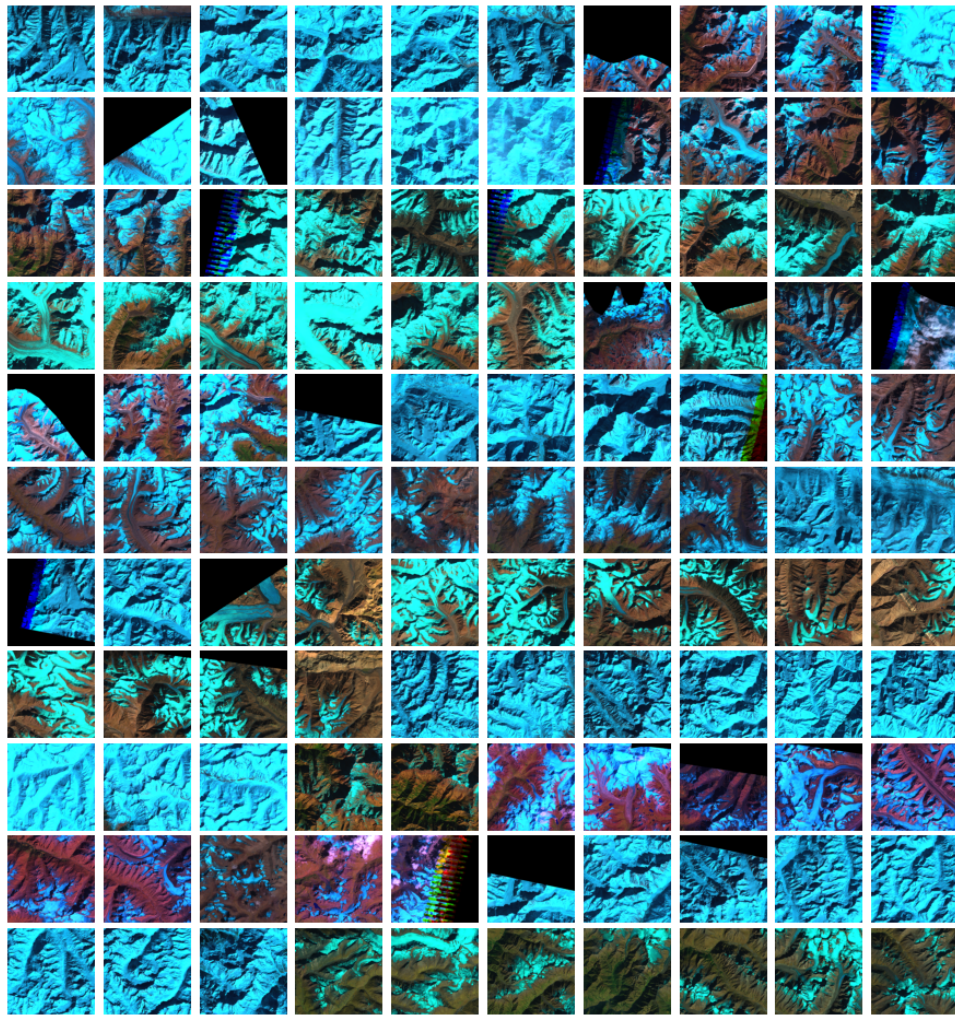


Figure 6.0.1: Image Slices in Test Set

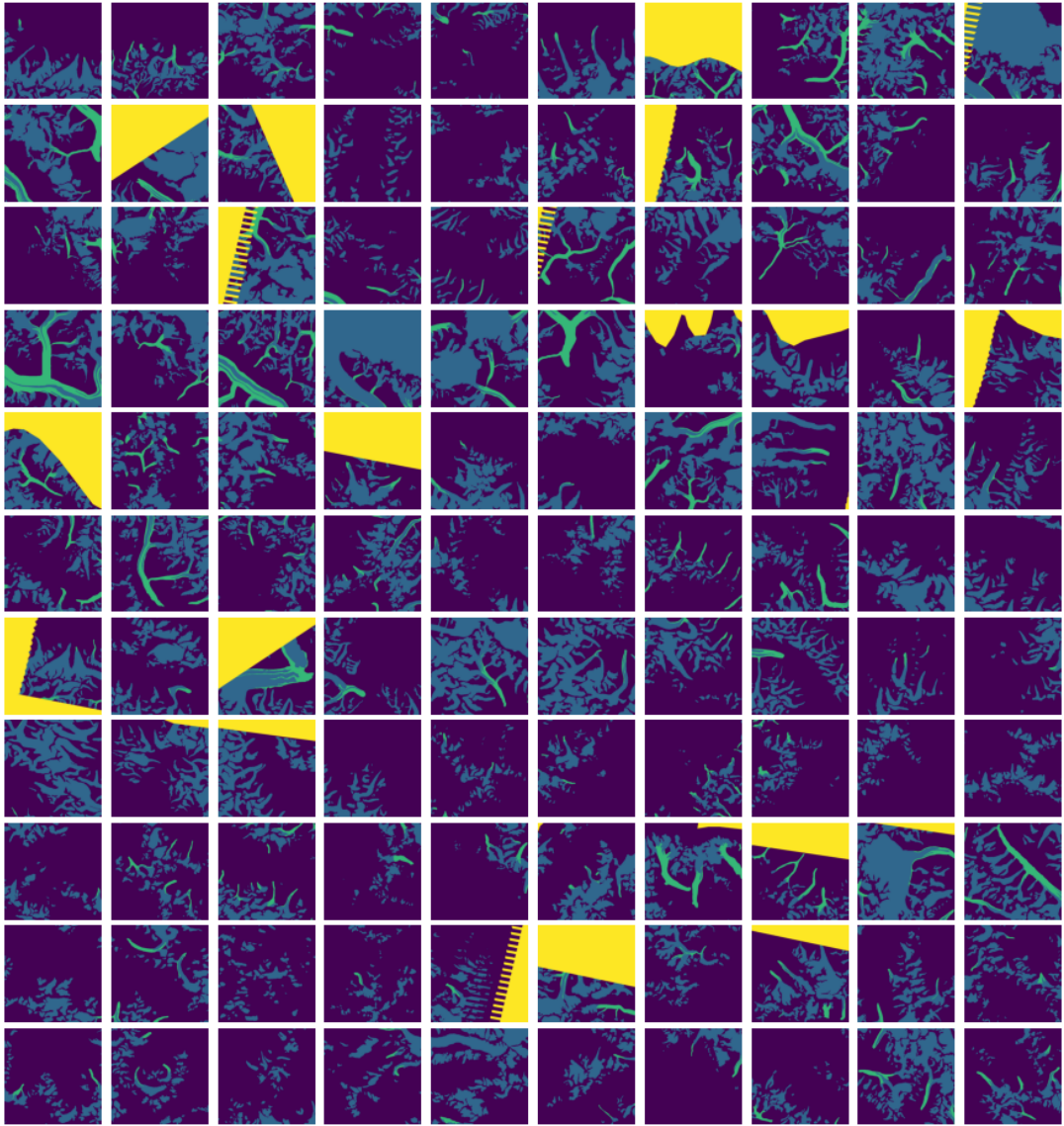


Figure 6.0.2: Corresponding Masks



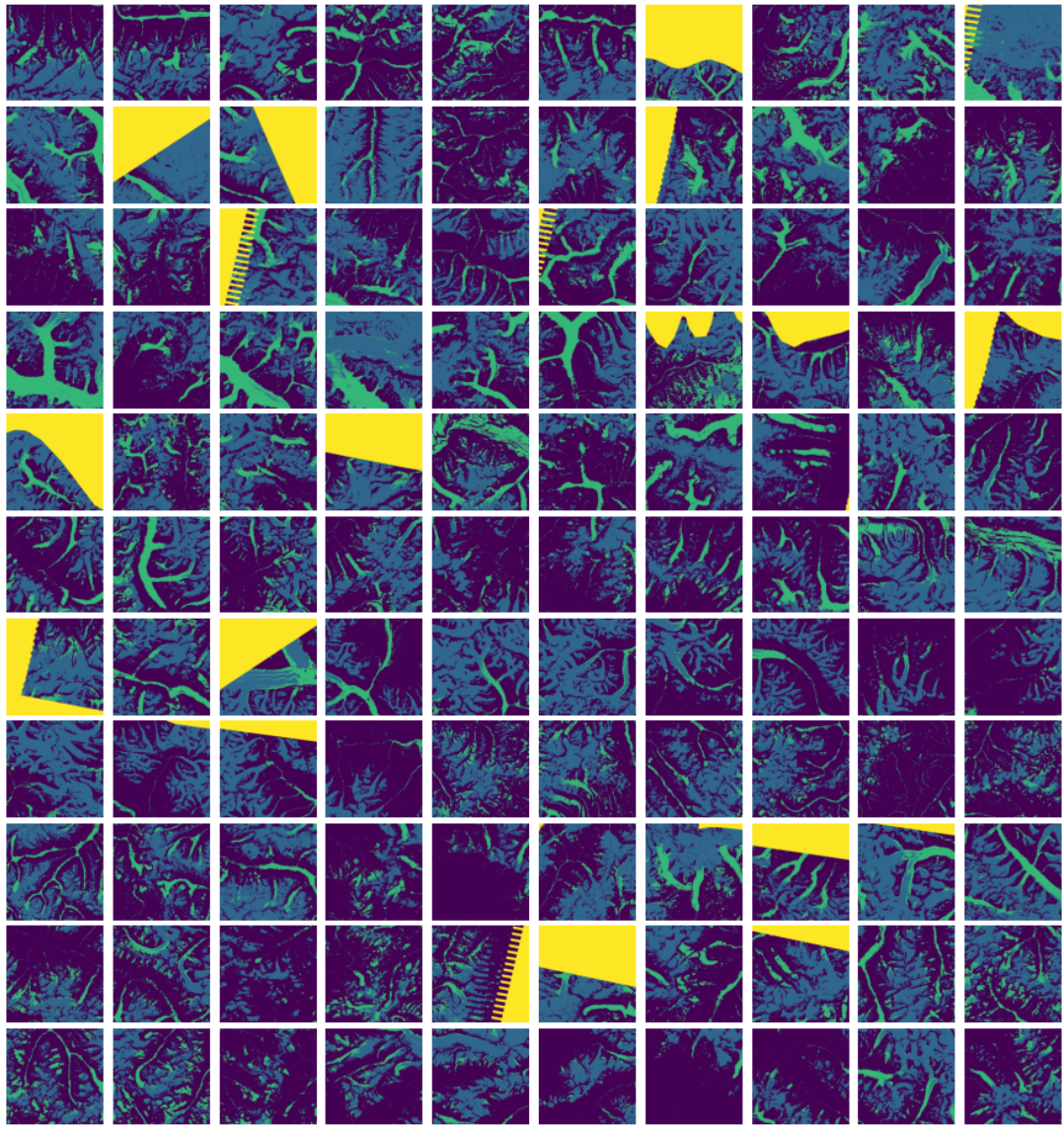


Figure 6.0.3: Corresponding Predictions using Random Forest

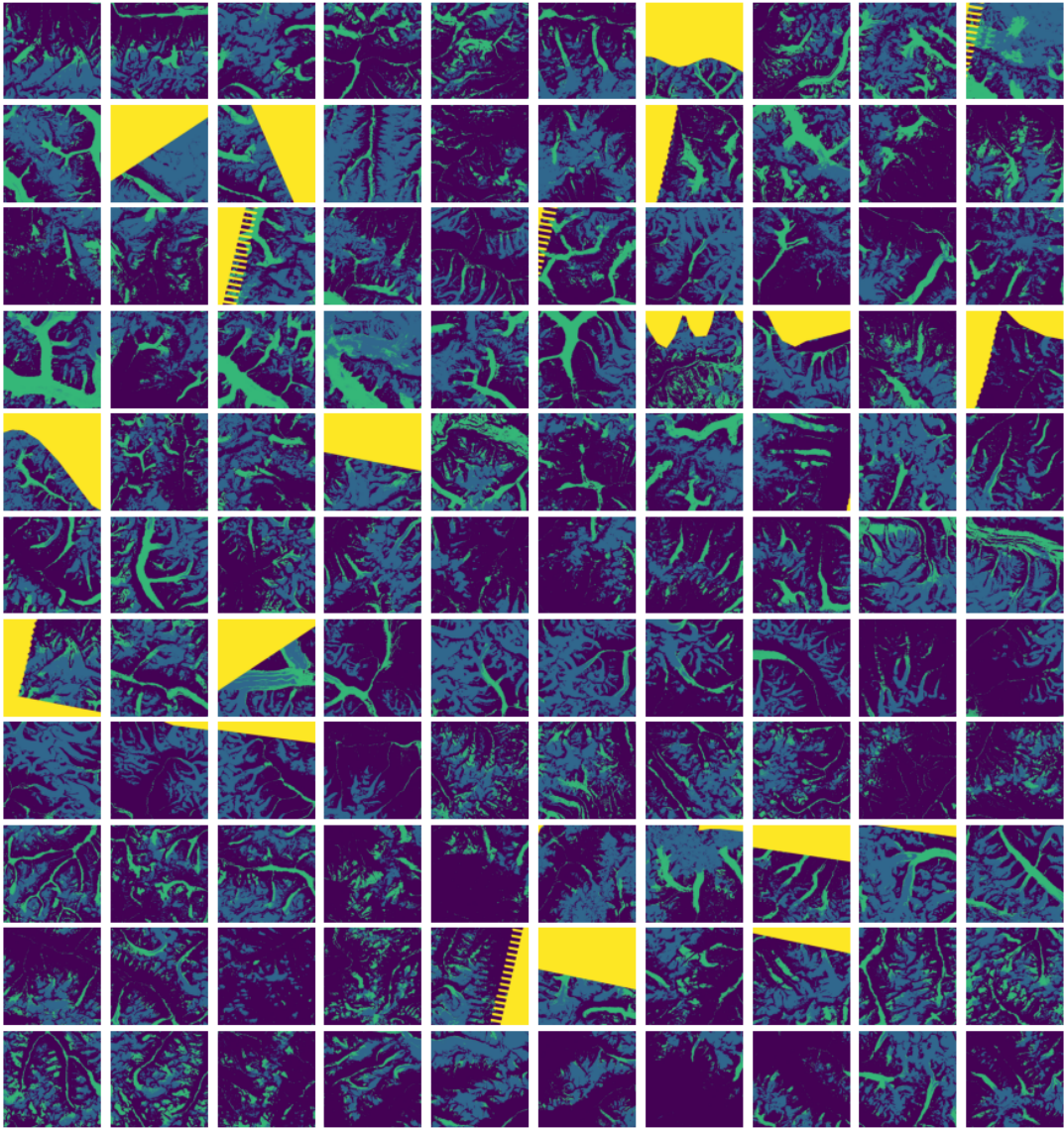


Figure 6.0.4: Corresponding Predictions using XGBoost



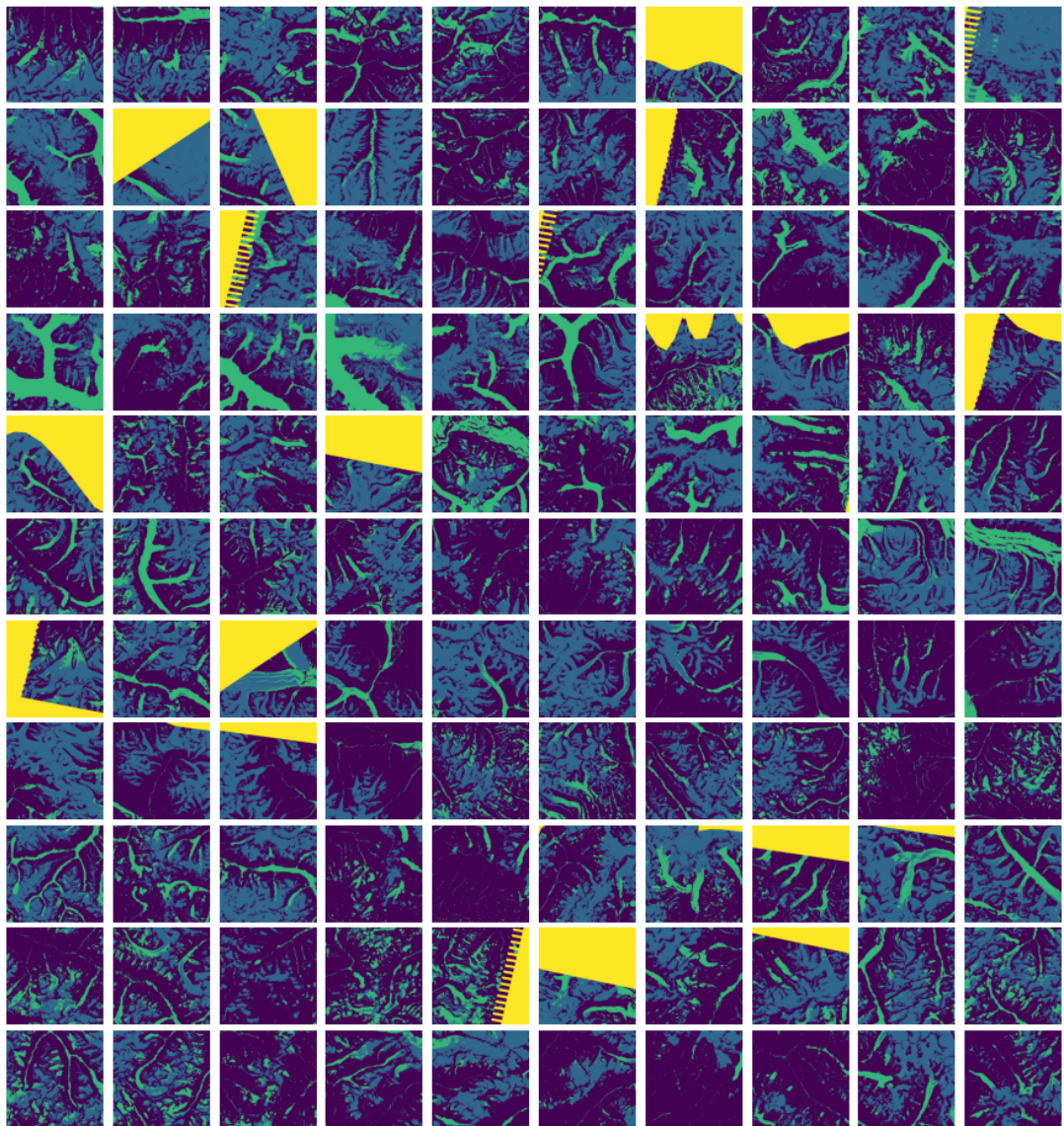


Figure 6.0.5: Corresponding Predictions using MLP

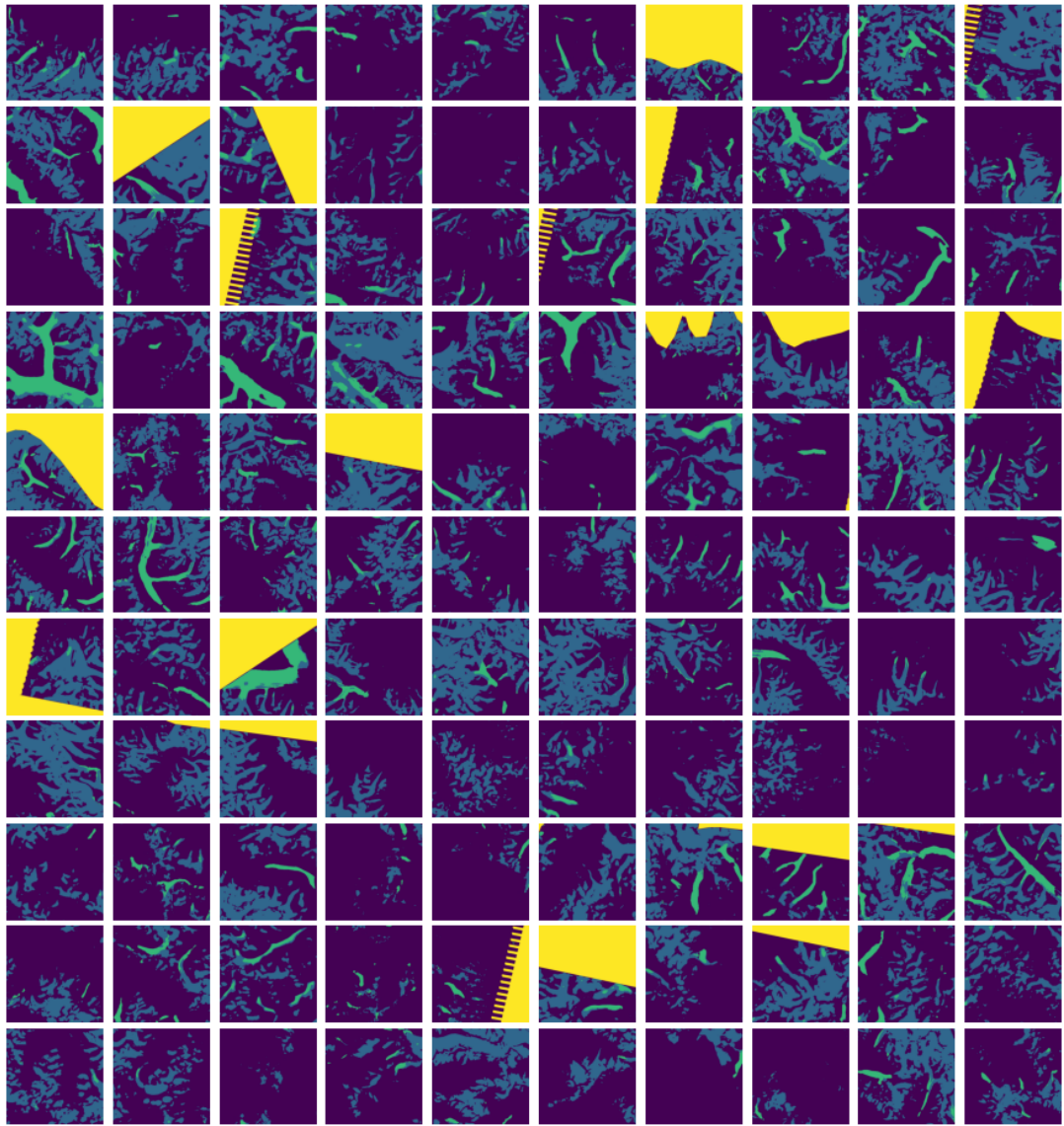


Figure 6.0.6: Corresponding Predictions using U-Net



# Chapter 7

## Curriculum Vitae

Bibek Aryal was born on December 21, 1994. Son of Rajan Aryal and Chandrakala Aryal, he got his bachelors degree from Tribhuwan University, Kathmandu, Nepal on 2015. Between 2015 and the Fall of 2018, Bibek worked as a web developer and later as a Research Programmer in Kantipur Digital Corp, Kathmandu.

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 computaional science program. Bibek was a Teaching Assistant in the Mathematical Science Department from Fall 2018 to Spring 2020. He is currently working as a Ph.D. Research Associate in Systems Ecology Lab, UTEP and has been a member of Vision and Learning Lab, UTEP since Fall 2019. Bibek is pursuing his research in the field of computer vision with professor Olac Fuentes of Department of Computer Science at UTEP as his research advisor. On Summer 2020, he worked as a Data Scientist Intern at Microsoft.

Email address: baryal@miners.utep.edu