
Machine Learning for Glacier Monitoring in the Hindu Kush Himalaya

Shimaa Baraka^{1,*}, Benjamin Akera^{1,*}, Bibek Aryal^{2,*}, Tenzing Sherpa³, Finu Shresta³, Anthony Ortiz⁴, Kris Sankaran⁵, Juan Lavista Ferres⁴, Mir Matin³, and Yoshua Bengio¹

¹Mila

²University of Texas at El Paso

³Integrated Center for Mountain Development

⁴Microsoft Research

⁵University of Wisconsin - Madison

*Equal Contribution

Abstract

Glacier mapping is key to ecological monitoring in the [Hindu Kush Himalayan \(HKH\)](#) region. Climate change poses a risk to individuals whose livelihoods depend on the health of glacier ecosystems. In this work, we present a machine learning based approach to support ecological monitoring, with a focus on glaciers. Our approach is based on semi-automated mapping from satellite images. We utilize readily available remote sensing data to create a model to identify and outline both clean ice and debris-covered glaciers from satellite imagery. 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.

1 Introduction

Glaciers are a source of freshwater and are critical to the Hindu Kush Himalayan (HKH) region both ecologically and societally [1]. However, glaciers are continuing to shrink at an alarming rate and this will result in diminished freshwater flow. This is likely to cause adverse effects for the benefactors of freshwater flow from glaciers. Additionally, glacier shrinkage has been shown to be a significant factor in the current sea-level rise [4]. This calls for effective and efficient methods to map and delineate glaciers in order to monitor changes and plan integrated water resource management and glacial hazard and risk management.

In such areas, remote sensing offers complementary information that can be used to monitor glaciers [13, 12]. Remote sensing allows the estimation of parameters like snow cover, glacier elevation, and ice index over large geographical and temporal scales. Utilising this information, different automated methods of delineating glaciers have been developed. However, the efficacy and accuracy of these methods are affected by cloud cover, highly variable snow conditions, and the spectral similarity of supra-glacial debris with moraines and bedrock [3]. These errors are somewhat addressed through the application of semi-automated mapping methodologies, which combine outputs from automated methods with manual interventions. However, this is labor intensive and time-consuming. Machine learning techniques can play a significant and positive role in speeding the process up.

We apply machine learning techniques to automate methods for glacier mapping from satellite imagery. We utilize semantic segmentation - a deep learning approach that performs pixel-wise classification in images. Using the [HKH](#) glacier region as an area of study, we use available satellite imagery from Landsat and glacier labels provided by the [International Centre for Integrated Mountain Development \(ICIMOD\)](#) - a regional intergovernmental learning and knowledge sharing center

serving the eight regional member countries of the [HKH](#) [2]. Using these resources, we develop an extensible pipeline, a dataset, and baseline methods that can be utilized for automated glacier mapping from satellite images. We also present qualitative and quantitative results describing properties of our models. Additionally, we deploy our models as a web-based tool to demonstrate how machine learning can complement, rather than supplant, existing workflows.

2 Study Area and Data Sources

Our experiments are carried out on the [HKH](#) region. The [HKH](#) is also known as the Third Pole of the world as it consists of one of the largest concentrations of snow and ice besides the two poles. It constitutes more than 4 million square kilometers of hills and mountains in the eight countries of Afghanistan, Bangladesh, Bhutan, China, India, Myanmar, Nepal and Pakistan. Glaciers have been periodically identified and classified by experts at the [ICIMOD](#) [2]. Glaciers are categorized as either *clean ice* or *debris-covered* subtypes.

The labels we use have been generated through a semi-automated pipeline based on hyperpixel segmentation. Historically, [ICIMOD](#) has used the eCognition software [7] to segment Landsat imagery into image objects defined by a contiguous set of pixels with similar intensity value. Those hyperpixels that contain debris or ice glacier are then merged and downloaded for refinement. The manual refinement phase involves removing labeled regions that are not at plausible glacier elevations or which do not pass specified [Normalised Difference Vegetation Index \(NDVI\)](#), [Normalised Difference Snow Index \(NDSI\)](#) or [Normalised Difference Water Index \(NDWI\)](#) thresholds [8, 9, 6].

We release our data in the [LILA BC repository](#). The input data come in two forms – the original 35 Landsat tiles and 14,190 extracted numpy patches. Labels are available as raw vector data in shapefile format and as multichannel numpy masks. Both the labels and the masks are cropped according to the borders of [HKH](#). The numpy patches are all of size 512×512 and their geolocation information, time stamps, and source Landsat IDs are available in a geojson metadata file.

3 Model Architecture and Methodological Pipeline

The task of identifying and mapping glaciers in remote sensing images fits well within the framework of semantic segmentation. We adapted the U-Net architecture for this task [14]. The U-Net is a fully convolutional deep neural network architecture; it consists of two main parts, an encoder network and a decoder network.

The model was trained using gradient descent and the Dice loss [15] was used as the optimization criterion (see the Appendix). We adapt a human-in-the-loop approach to correct the segmentation errors made by the model. This is useful because glacier mapping often requires expert opinion and models make errors that need to be resolved by people.

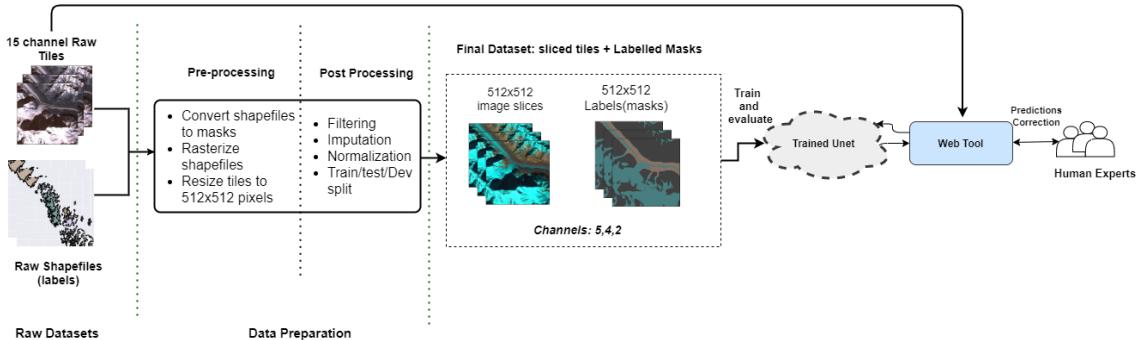


Figure 1: Our methodological pipeline first converts LE7 tiles to preprocessed patches used for model training. The trained model is deployed as an interactive web tool.

Our approach is summarized in a multi-step pipeline presented in Figure 1. It first converts the raw tiles into patches and converts their vector data labels to masks. We filter, impute and normalize the resulting patch-mask pairs before splitting them into train, test and validation data sets. The code to

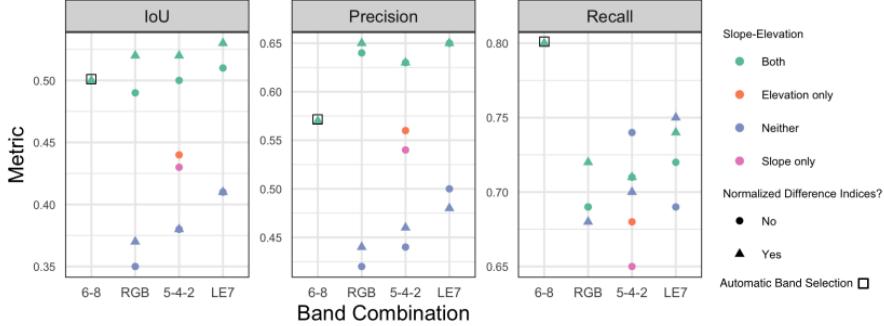


Figure 2: Experimental results for channel selection. The x -axis indicates which LE7 bands were used. Color describes whether elevation and slope were used. Runs using **NDWI**, **NDSI**, and **NDVI** are labeled with triangles. Elevation and slope data significantly boost performance, and using all bands is better than using any subset. Results when using RF features are enclosed in a square.

replicate our process can be found in a GitHub repository¹. The script to query Landsat 7 tiles using Google Earth engine is in another GitHub repository².

4 Experiments

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

Band Selection Model performance tends to deteriorate in the many-bands limited-training-data regime [11]. This is often alleviated through band subset selection. Here, we study whether specific channels are more relevant for glacier mapping. We experimented with the combination of bands B5 (Shortwave infrared), B4 (Near infrared), and B2 (Green) which is the false-color composite combination used to differentiate snow and ice from the surrounding terrain when manually delineating glaciers. We compare this with (1) the true color composite band combination, B1 (Blue), B2 (Green), B3 (Red) and (2) all Landsat 7 bands. We also consider (1) slope and elevation from the Shuttle Radar Topography Mission (SRTM) as additional channels and (2) spectral indices - snow index (**NDSI**), water index (**NDWI**), and vegetation index (**NDVI**) - as used in manual glacier delineation [2]. Lastly, we perform pixel-wise classification on all channels with a random forest (RF) and select channels with feature importance scores greater than 5%, see appendix Figure 5.

Figure 2 shows performance when varying input channels. The experiments are carried out on the 383 patches with at least 10% of pixels belonging to either clean ice or debris-covered glaciers. We evaluated the model over 55 patches using Intersection over Union (IoU). The RF classifier features did not achieve the maximum IoU, likely due to a lack of spatial context. Adding elevation and slope channels provides an improvement of 10-14% IoU. This agrees with domain knowledge – elevation and slope maps are referred to in the current process. Appendix Figure 6 illustrates that the model learns that low elevation and steep areas typically do not contain glaciers. Using **NDVI**, **NDSI**, and **NDWI** improves results when input channels are different from those used to define the indices.

Debris covered versus clean ice glaciers There are two types of glaciers we care about: clean ice glaciers and debris-covered glaciers. Clean ice glaciers have an appearance similar to snow. Debris-covered glaciers are covered in a layer of rock and flow through valley-like structures. For segmentation, clean ice glaciers are often confused with snow, resulting in false positives. Debris-covered glaciers are more similar to the background, often leading to false negatives. Debris-covered glaciers are also much rarer. We experimented with binary and multiclass approaches to segmentation.

We trained a 2-class model to segment glacier from background areas and compared it with 3-class model for clean ice vs. debris-covered vs. background. We also compared the 3-class model with two

¹https://github.com/krisrs1128/glacier_mapping/

²https://github.com/Aryal007/GEE_landsat_7_query_tiles/

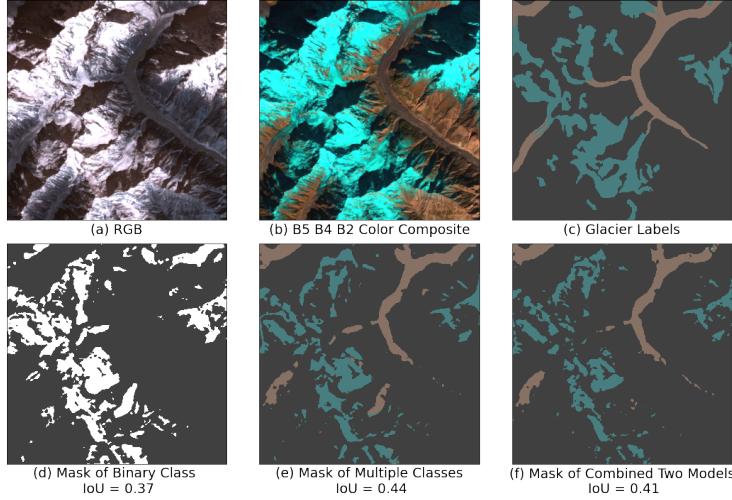


Figure 3: Example imagery and labels (a - c), with model predictions (d - f). Blue and orange labels are for clean ice and debris-covered glaciers, respectively. (d) A model trained to recognize the union of clean ice or debris-covered glaciers fails to recognize major debris-covered glaciers. (e - f) Multiclass and combined binary models give comparable predictions.



Figure 4: The image on the left shows the polygonized prediction for an area of interest. The image to the right shows the tool’s functionality of allowing users to correct predictions.

binary models for each glacier type. We filtered to patches where both debris-covered and clean ice glaciers were present, resulting in 648 training patches and 93 validation patches. Since many patches contain few positive class pixels, we evaluate IoU over the whole validation set rather than the mean IoU per patch. Table 2 shows that the multiclass model and binary model deliver comparable overall performance. However, the approaches differ in regions with higher coverage from debris-covered glaciers. Table 3 and figure 3 show an increase in the performance gap in favour of the multiclass model as the debris-covered glacier percentage increases.

5 Glacier Mapping Tool

To support the work of geospatial information specialists to delineate glaciers accurately we developed an interactive glacier mapping tool. The tool allows users to test our segmentation models on different sources of satellite imagery. Users can visualize predictions in the form of polygons and edit them to obtain a glacier map for the area of interest. This interactivity supports the process of validating models, identifying systematic sources of error, and refining predictions before release. Users can compare data sources, which can clarify ambiguities. As future work, we intend to incorporate model retraining functionality. A screenshot from the tool is visible in Figure 4.

6 Conclusion and Future Work

We have presented deep learning and remote sensing techniques that support semi-automated glacier mapping. We have experimentally explored the effects of channel selection and task definition on performance. Finally, we describe a web tool to provide feedback and correct errors made by the model. More work needs to be done to (1) incorporate the human feedback into the trained model through some form of active learning, (2) develop network architectures and criteria that better use domain knowledge, and (3) understand the generalizability of these methods to regions outside of the Hindu Kush Himalaya.

7 Acknowledgements

We acknowledge **ICIMOD** for providing a rich dataset which this work has been built on. We also appreciate Microsoft for funding this project under the AI for Earth program. This research was enabled in part by support provided by Calcul Quebec and Compute Canada. We would like to thank Dan Morris from Microsoft AI for Earth for making this collaboration between ICIMOD and academia possible

References

- [1] Samjwal R Bajracharya, Pradeep K Mool, and Basanta R Shrestha. The impact of global warming on the glaciers of the himalaya. In *Proceedings of the International Symposium on Geodisasters, Infrastructure Management and Protection of World Heritage Sites*, pages 231–242, 2006.
- [2] Samjwal Ratna Bajracharya and Basanta Raj Shrestha. The status of glaciers in the hindu kush-himalayan region. Technical report, International Centre for Integrated Mountain Development (ICIMOD), 2011.
- [3] Donald J Biddle. Mapping debris-covered glaciers in the cordillera blanca, peru: an object-based image analysis approach. 2015.
- [4] Mark B Dyurgerov. Mountain glaciers are at risk of extinction. In *Global Change and Mountain Regions*, pages 177–184. Springer, 2005.
- [5] Tom G. Farr, Paul A. Rosen, Edward Caro, Robert Crippen, Riley Duren, Scott Hensley, Michael Kobrick, Mimi Paller, Ernesto Rodriguez, Ladislav Roth, David Seal, Scott Shaffer, Joanne Shimada, Jeffrey Umland, Marian Werner, Michael Oskin, Douglas Burbank, and Douglas Alsdorf. The shuttle radar topography mission. *Reviews of Geophysics*, 45(2), 2007.
- [6] Bo-Cai Gao. NDWI—a normalized difference water index for remote sensing of vegetation liquid water from space. *Remote sensing of environment*, 58(3):257–266, 1996.
- [7] Trimble GeoSpatial. eCognition software by Trimble GeoSpatial: Munich, Germany. <http://www.ecognition.com/>, 2020 (accessed October 3, 2020).
- [8] Samuel N Goward, Brian Markham, Dennis G Dye, Wayne Dulaney, and Jingli Yang. Normalized difference vegetation index measurements from the advanced very high resolution radiometer. *Remote sensing of environment*, 35(2-3):257–277, 1991.
- [9] Dorothy K Hall and George A Riggs. Normalized-difference snow index (NDSI). 2010.
- [10] ICIMOD. Clean ice and debris covered glaciers of hkh region, 2011.
- [11] Anthony Ortiz, Alonso Granados, Olac Fuentes, Christopher Kiekintveld, Dalton Rosario, and Zachary Bell. Integrated learning and feature selection for deep neural networks in multispectral images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1196–1205, 2018.
- [12] Frank Paul, Nicholas E Barrand, S Baumann, Etienne Berthier, Tobias Bolch, K Casey, Holger Frey, SP Joshi, V Konovalov, Raymond Le Bris, et al. On the accuracy of glacier outlines derived from remote-sensing data. *Annals of Glaciology*, 54(63):171–182, 2013.

- [13] Adina E Racoviteanu, Frank Paul, Bruce Raup, Siri Jodha Singh Khalsa, and Richard Armstrong. Challenges and recommendations in mapping of glacier parameters from space: results of the 2008 global land ice measurements from space (glims) workshop, boulder, colorado, usa. *Annals of Glaciology*, 50(53):53–69, 2009.
- [14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [15] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017.
- [16] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656, 2015.

A Implementation Details

We query Landsat 7 raw images used for creating labels [2] using Google Earth Engine. In addition to the raw Landsat 7 tiles, we compute Normalized-Difference Snow Index (NDSI), Normalized-Difference Water Index (NDWI), and Normalized-Difference Vegetation Index (NDVI) and add them as additional bands to the tiles. Finally, we query slope and elevation from the Shuttle Radar Topography Mission [5] and add them as additional bands to give us final tiles with 15 bands. The vector data corresponding to glacier labels [10] is downloaded from [ICIMOD Regional Database System](#) (RDS). We then follow pre-processing and post-processing as shown in Figure 1 to prepare the data. The pre-processing steps include conversion of vector data to image masks, cropping the input image and vector data to HKH borders, and slicing the mask and tiles to patches of size 512×512 pixels. We then filter patches with low glacier density (thresholds vary by experiment), impute nan values with 0, normalize across channel for each patch, and randomly split the data into train (70%) / dev (10%) / test (10%).

We make use of a U-Net architecture [14] for the segmentation of glacier labels. We use a kernel size of 3×3 for convolution layers in the downsampling operations and kernel size of 2×2 for convolution layers and transpose convolution layers in the upsampling layers. For the pooling operation, we use maxpool with kernel size 2×2 . The output of the first convolution operation has 16 channels and we double the channels after each convolutional layer in during downsampling and in the bottleneck layer. We halve the output channels in each convolutional layer during upsampling. We use a depth of 5 meaning there are 5 downsampling layers followed by 5 upsampling layers with a bottleneck layer in between. We use Adam as our optimizer with a learning rate of $1e^{-4}$. We use spatial dropout [16] of 0.3 and ℓ^1 regularization with $\lambda = 5e^{-4}$ to prevent the model from overfitting on training data.

B Supplemental Tables and Figures

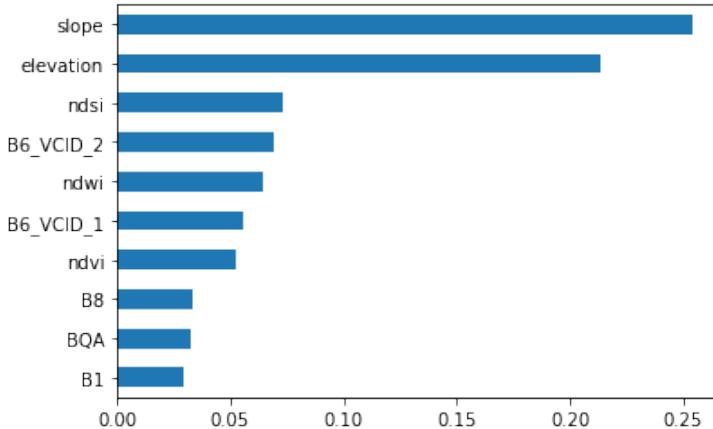


Figure 5: Feature importance scores using Random Forest. Slope and elevation are key variables.

Table 1: A comparison of IoU for the U-Net based model architecture with other traditional machine learning approaches. Pixels from train slices are sampled to train machine learning classifiers to predict labels for each pixel. There are a total of 15 features for each pixels, one for value in each band, and one of the three output labels. The output segmentation mask is generated by predicting the class for each pixel in the test slices using trained classifiers. The U-Net based classifier outperforms conventional machine learning classifiers especially in case of debris glaciers.

Model	IoU of Clean Ice Glaciers	IoU of Debris Glaciers
Random Forest	0.5807	0.2024
Gradient Boosting	0.5663	0.1930
Multi-Layered perceptrons	0.5452	0.1781
U-Net	0.5829	0.3707

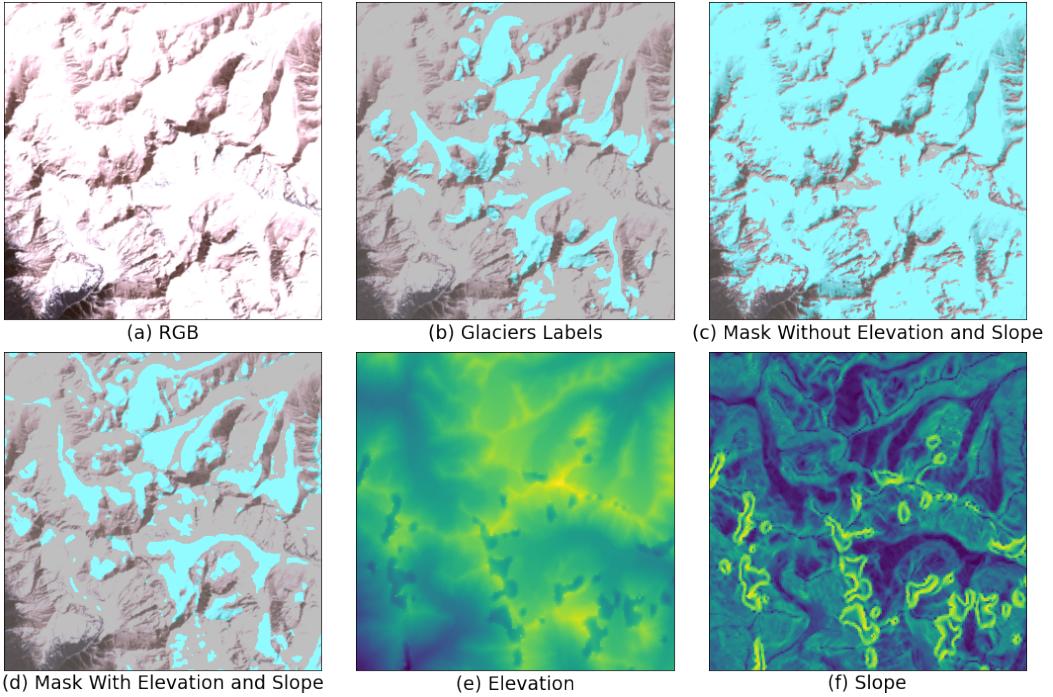


Figure 6: Slope and elevation are critical for improving precision of glacier predictions. Light blue labels represent both clean ice and debris-covered glaciers. Light yellow and green are areas of high elevation (e) and slope (f). Clean ice glaciers tend to be found at high elevation, while debris-covered glaciers are found in valleys. Neither type of glacier is found on steep slopes.

Table 2: A comparison of error rates on clean ice and debris-covered glaciers across three modeling approaches. The first row is a model trained to predict glacier or background, without distinguishing between debris-covered or ice glaciers. The second row is a multiclass model trained to simultaneously segment debris-covered and clean ice glacier. The final row gives the result of training two separate models to distinguish each type of glacier. Results are comparable across approaches, with a slight edge for the split training approach.

Model	IoU of Glaciers	IoU of Clean Ice Glaciers	IoU of Debris Glaciers
Binary Model	0.476	-	-
Multiclass Model	0.473	0.456	0.291
Two Binary Models	0.48	0.476	0.31

Table 3: The exact numbers used in Figure 7

% of Debris	% of Data	Binary Class IoU	Multiclass IoU	IoU Difference
> 0%	100%	0.476	0.473	-0.3%
> 1%	77%	0.523	0.532	+0.9 %
> 2%	52%	0.524	0.544	+2%
> 5%	18%	0.497	0.571	+7.4%
> 10%	6%	0.46	0.603	+14.3%

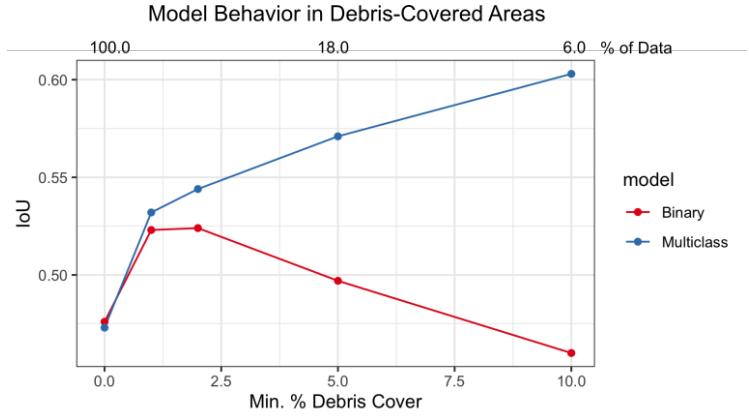


Figure 7: Effect of debris percentage on IoU. The multiclass model performs well on areas with high density of debris-covered glaciers. The binary model trained to distinguish any type of glacier from background suffers in these regions. When making no distinction between glacier types, the model only learns to recognize clean-ice glaciers.

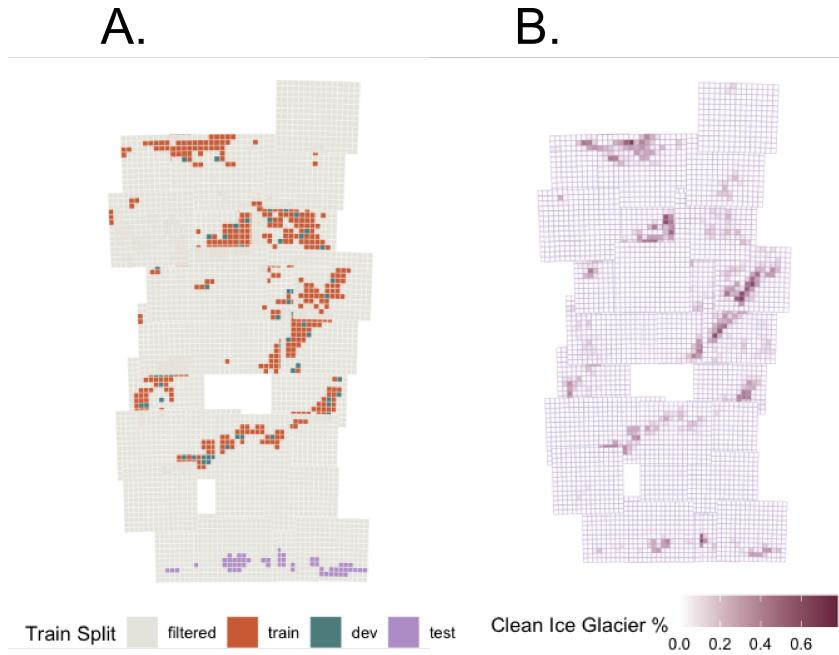


Figure 8: (a) One of three geographically disjoint train and test sets used in a supplemental experiment on geographic generalization (results in Figure 9). Each large square is a LE7 tile, each shaded element is a 512×512 patch. Patches without any glaciers are filtered as in prior experiments. The validation set is chosen at random from the same region as training. (b) The proportion of pixels covered by clean ice glacier within each patch. Note that the test region is less densely covered by glaciers.

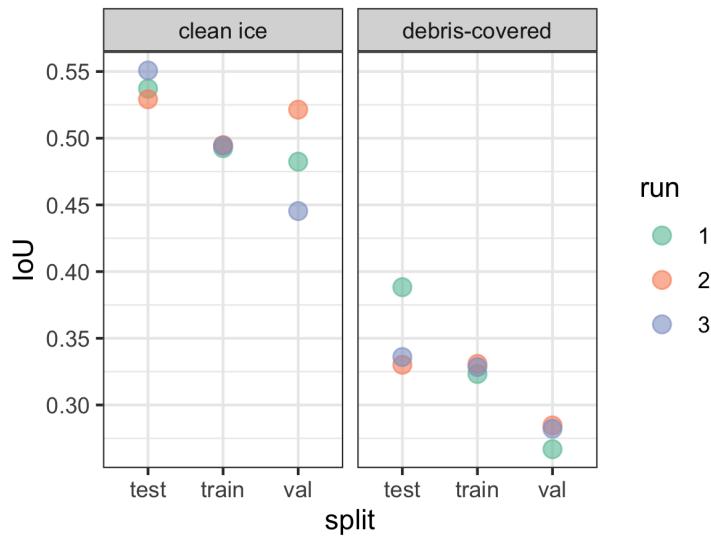


Figure 9: Results of a supplemental geographic generalization experiment using the multiclass model of Section 4. Train and test sets are chosen at random and required to be geographically disjoint, as in Figure 8a. Surprisingly, performance slightly increases in the test set. Evidently, glacier appearance is relatively homogeneous across the area of study. The increase in performance can be explained by the fact that, in each random geographic split, the test region had a lower glacier density, see Figure 8b.