International Journal of
Geo-Information

Article

Interactive Visualization and Representation Analysis Applied to Glacier Segmentation

Minxing Zheng, Xinran Miao and Kris Sankaran *

Department of Statistics, University of Wisconsin Madison, Madison, WI 53706, USA; mzheng54@wisc.edu (M.Z.); xmiao27@wisc.edu (X.M.)
* Correspondence: ksankaran@wisc.edu

Abstract: Interpretability has attracted increasing attention in earth observation problems. We apply interactive visualization and representation analysis to guide the interpretation of glacier segmentation models. We visualize the activations from a U-Net to understand and evaluate the model performance. We built an online interface using the Shiny R package to provide comprehensive error analysis of the predictions. Users can interact with the panels and discover model failure modes. We illustrate an example of how our interface could help guide decisions for improving model performance. Further, we discuss how visualization can provide sanity checks during data preprocessing and model training. By closely examining the problem of glacier segmentation, we are able to discuss how visualization strategies can support the modeling process and the interpretation of prediction results from geospatial deep learning.

Keywords: interactive visualization; representation analysis; glacier segmentation; Shiny app

Citation: Zheng, M.; Miao, X.; Sankaran, K. Interactive Visualization and Representation Analysis Applied to Glacier Segmentation. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 415. https://doi.org/10.3390/ijgi11080415

Academic Editors: Wolfgang Kainz and Florian Hruby

Received: 26 April 2022
Accepted: 20 July 2022
Published: 22 July 2022

1. Introduction

Advances in machine learning for remote sensing have enabled the automatic tracking of changes of glacier cover. Researchers have proposed to implement deep-learning models on satellite images to identify the positions and boundaries of glaciers. These approaches have demonstrated improved performance relative to non-deep-learning baselines. However, the literature on the interpretability of these black-box models is limited. This gap is problematic because interpretability can help researchers find systematic failures of modeling.

Motivated by this, the goal of this work is to provide visual interpretation techniques to facilitate understanding across the geospatial deep-learning workflow from data preparation to error analysis. We conduct representation analysis to visually interpret the modeling process of a U-Net architecture deep-learning model. We use interactive visualization to provide a comprehensive error analysis of the predictions. We build an online interactive interface using the R Shiny package where users can examine prediction results in context [2]. We show that, with the help of visual interpretations, it is possible to detect underlying errors in the raw data, which are difficult to detect otherwise.

The rest of the paper is organized as follows. In Section 2, we introduce several concepts used throughout. In Section 3, we discuss data preprocessing. Then, we introduce the details of our model and model training process. Both steps are guided and explained with data visualization. In Section 4, we present the interpretation and analysis of the trained model using representation analysis. In Section 5, we introduce a Shiny app for error analysis and discuss the findings with the help of our app. In Section 6, we conclude our work and discuss how to reproduce our work for further research.

2. Background

We first review the background on interactive visualization and representation analysis of deep-learning models.

2.1. Interactive Visualization

The large scale of remote sensing data—both in the scope of area covered and number of sensor features—creates challenges for data exploration. An increasing number of interactive visualization techniques have been proposed to support geospatial and remote sensing research. Focusing and linking are two techniques for high-dimensional data visualization [3]. Focusing supports visualization of only a part of the data in a single view.

Focusing visualization techniques include subset selection and dimension reduction. They are commonly applied by zooming, panning, slicing, projecting and data reduction. Focusing limits the amount of presented information. In contrast, to have a more comprehensive understanding of data, linking can be used to display multiple views of data together. Different views are synchronized to give a more extensive description of the data. Linked visualizations are often implemented by brushing, clicking and dragging.

For example, Anselin et al. [4] dynamically linked a cartographic representation of data on a map with summary statistical graphics, such as histograms, box plots and scatterplots. Their interface implements linking and brushing between maps and statistical graphics. Anselin [5] presented an interactive dynamic framework in which brushing across a variogram cloud plot highlights pairs of observations on the map, suggesting potential spatial outliers. Hibbard and Santek [6] implemented rotation, zooming and panning in three dimensions. This also allows users to select combinations of scalar variables, and users can interactively control the time animation of the data.

A data reduction method proposed by Tasnim and Mondal [7] reduces the data size by 75% while preserving the visual elements of images. Keim et al. [8,9] discussed the use of visualization techniques to explore large-scale geospatial datasets using more classical data mining methods. Janik et al. [10] combined interactive visualization with representation learning to characterize latent representations on a building segmentation dataset. Humer et al. [11] developed explanations of segmentation networks, allowing users to select predicted segmentations and visualize saliency maps summarizing pixel-level importance measures.

2.2. U-Net Model

The U-Net deep learning architecture has been widely used in image segmentation problems. It and its variants have been widely applied to medical image segmentation problems [12–14] and geospatial satellite image segmentation problems [15–18]. In the glacier segmentation problems, previous works have used U-Net to identify positions and shapes of glaciers [19–21]. The U-Net model contains two parts: an encoder and a decoder.

The encoder contains down-sampling layers, and the decoder contains up-sampling layers. Features are extracted by the down-sampling layers at the pixel level. The extracted information is then up-sampled to the input resolution using the decoder. Skip connections between down-sampling layers and the corresponding up-sampling layers provide an alternative way of learning features. Through them, features can bypass downsampling and be used directly for prediction. However, richer and semantic information is normally learned by deeper layers.

Activations are formed at each layer and are used in representation analysis. They are computed as nonlinear transformations of outputs from convolution layers. From the activations, we can discover how the input is changed and gain information about which features are learned in each U-Net layer.

2.3. Representation Analysis

Though they often achieve state-of-the-art performance on earth observation problems, deep neural networks are, in a way, black boxes since their decision rules can not be easily described. Understanding deep models can inspire improvements on data analysis problems as well as methodology [22]. To describe what a trained deep neural network model has learned, some researchers have studied the parameters of neurons at each layer [23,24]. For example, Luo et al. [25] visualized the weights of all units in a deep model for digit classification.

The learned weights appear as strokes with clearer borders as the model goes deeper. Others aimed to interpret the functions of neurons [26]. For example, one approach is to investigate what certain units are looking for by generating artificial inputs that maximizes an individual neuron's activation [27]. Alternatively, it is possible to study the activations of each neuron after passing certain data through the model, whose results can reflect on the input data and allow for further unsupervised investigation [28,29]. More broadly, a rich literature has emerged at the boundary between interactive visualization and interpretation of deep-learning models—we refer the interested reader to the survey papers [30,31].

3. Data Preprocessing and Modeling

In this section, we introduce our data preprocessing and modeling procedures. Further, we illustrate how visualizations can inform preprocessing and modeling choices. These visualizations also provide a sanity check for the preprocessed data.

3.1. Data Preprocessing

Geographically, we study the glaciers in the Hindu Kush Himalayas region, which contains one of the world's largest concentrations of snow and glaciers. A geographical map of this area is provided in Figure A1. Recent studies have documented that glaciers in this region are retreating due to rising temperatures [32]. The potential loss of the associated ecosystem services is reason for concern. Therefore, automatic tracking and identification of glacier position and size is needed.

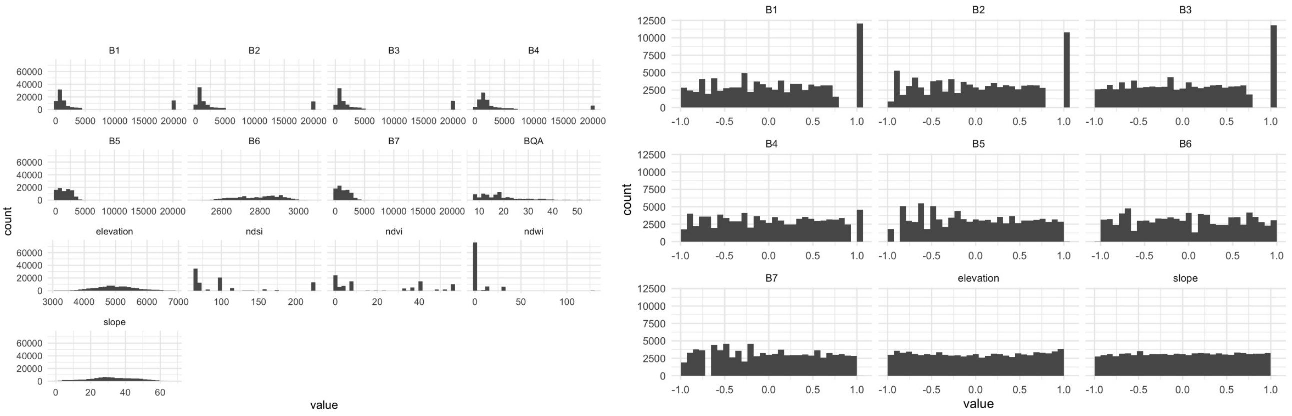
Our raw imagery data contains 13 bands from Landsat 7 (LE7) and the Shuttle Radar Topography Mission (SRTM). It contains the bands from B1 to B7, BQA of LE7, elevation from SRTM, derived Normalized Difference Snow Index (NDSI), Normalized Difference Vegetation Index (NDVI), Normalized Difference Water Index (NDWI) and slope. Bands B1 to B3 are RGB color bands, B4 to B7 are infrared related bands, and BQA is a quality assurance bitmask band. Figure 1 provides a histogram for each channel in the raw data. We found that different features have significantly different scales—this could lead to difficulty in model training and generalization.

Specifically, large weights will be needed for features with small scales; otherwise they may be overlooked during training. Hence, we equalize each feature to the −1 to 1 scale. It is noted that, in the distributions of original features, e.g., B1, B2 and B3, there is an outlying bar in each distribution indicating the value is truncated at a certain threshold. This is because the raw imagery released by [20] has been truncated. As a consequence, even after equalization, there is a bar that is away from the majority of the values. We also note that, after equalization, the color channel values are more condensed. Thus, the visualization of the processed patches will differ from those in typical RGB views. We select a subset of the original features to train the model more efficiently.

Working with a subset of channels allows the model to be trained with larger batch sizes, stabilizing model optimization. Specifically, we drop the BQA, NDSI, NDVI and NDWI bands and use the remaining features to train the model. Below, we show that the resulting model has satisfactory performance. The scaled features' histograms are given in Figure 1. We can see that all features have the same scales and almost have uniform distributions. However, for B1 to B3, a bar in each histogram stands out. This reflects the outlying bar in the raw data histogram. However, compared with the raw data, the deviating value in the scaled histogram is much closer to the bulk of non-outlier values. Though simple, these histograms of raw and transformed data give reassurance that the original data are appropriately processed for model training.

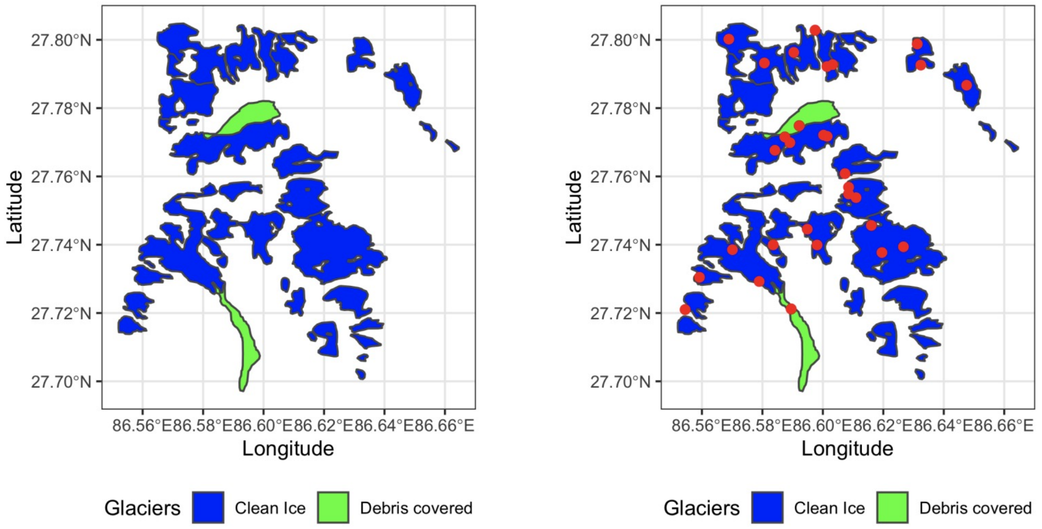
The label data that we used to train the model and validate predictions came from the International Center for Integrated Mountain Development (ICIMOD), an intergovernmental knowledge and learning center working on behalf of the people of the Hindu Kush Himalayas [1]. To understand the structure of the labels, we drew glacier boundaries in a partial area in Figure 2. The left panel of Figure 2 suggests a potential label imbalance issue. Compared to the non-glacier background, target glaciers make up only a small fraction of the whole area.

To obtain more informative training data and train the model more efficiently, we would prefer our target glaciers to make up a larger proportion in the training data. To achieve this, we resample the raw imagery data and create training patches in a way that ensures that more glaciers are present. Specifically, we randomly sample patch centers from within glaciers boundaries so that all the training patches are centered at a glacier. The resulting patches have higher coverage rates of glaciers. The right panel of Figure 2 presents an example of sampling results—the red points are the centers of sampled patches. We show an example of sampled patches in Figure 3. The preprocessed satellite image associated with a patch is given in Figure 3 along with its ground truth mask. In the mask, we see that two types of glaciers cover most areas in the patch, reflecting the sampling strategy.



(a) (b)

Figure 1. (**a**) Feature histogram before preprocessing. (**b**) Feature histogram after preprocessing. Each feature before preprocessed has significantly different ranges. Each feature is equalized into the same range from −1 to 1.



(a) (b)

Figure 2. (**a**) Raw glacier boundaries. (**b**) Glacier boundaries with sampled patches. In raw data, target glaciers only make up a limited fraction of area in the whole data. We sample patches from areas with glaciers, and the center of the sampled patches are marked as red points.

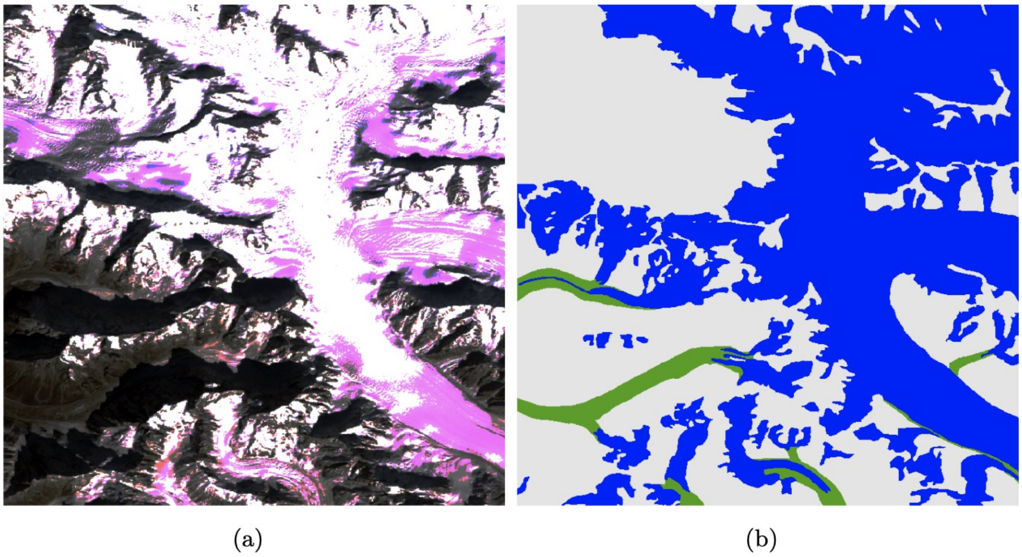


Figure 3. (**a**) An example of the preprocessed patch. (**b**) Corresponding mask patch. In the preprocessed patch, target glaciers now make up a large proportion. The pink areas are a consequence of feature equalization. They tend to represent the clean ice glaciers. As noted in the main text, equalization makes the color channel values more condensed, and these distortions lead to images that differ from the typical RGB views. We further discuss the relationship between feature equalization and image color in the appendix.

3.2. Modeling

We next introduce details about our model. We then discuss the modeling process and the ways that visualization supports it.

We implemented a U-Net architecture for the glacier segmentation task, following prior work [20,21]. We used a kernel size of 3 × 3 with 1 padding for the convolution layers in the down-sampling blocks, middle block and up-sampling blocks. We used a kernel size of 2 × 2 with stride of 2 for the up-sampling convolution layers in the up-sampling blocks. For the pooling layer, we used maxpool with kernel size 2 × 2. The input data had nine channels with size of 512 × 512 and three output channels with the same size.

The output channels correspond to clean-ice glaciers, debris-covered glaciers and background classes. We doubled the number of channels after each layer in the encoder and halved the number of channels after each layer in the decoder. The U-Net model had depth 4 with four down-sampling layers, four up-sampling layers and one bottleneck. During training, we used the Adam optimizer and set the learning rate to 0.0001. A dropout probability of 0.2 and ℓ_2-regularization of $\lambda = 0.0005$ were used to prevent overfitting. For the loss function, we used a combination of BCE and Dice loss. Figure 4 is a diagram for our model.

We draw the training and validation loss curves across epochs in Figure 5. In this figure, we see that, from epoch 1 through 30, both losses rapidly decrease, and from epoch 11 through epoch 50, the training loss slowly decreases while the validation loss stabilizes. This indicates that, after 30 epochs, the model's performance does not increase on the validation set, and thus we conclude that the model converged.

Figure 6 provides example model predictions. Comparing the predicted patch in the right panel with the label mask in the middle panel suggests that the model successfully recognizes most glacier pixels in this patch. However, the model fails to accurately recognize glacier borders or to detect connections between major glacier masses. Additionally, it seems that the debris-covered glaciers are not recognized well. These views provide clearer directions for improvement than an average performance metric alone.

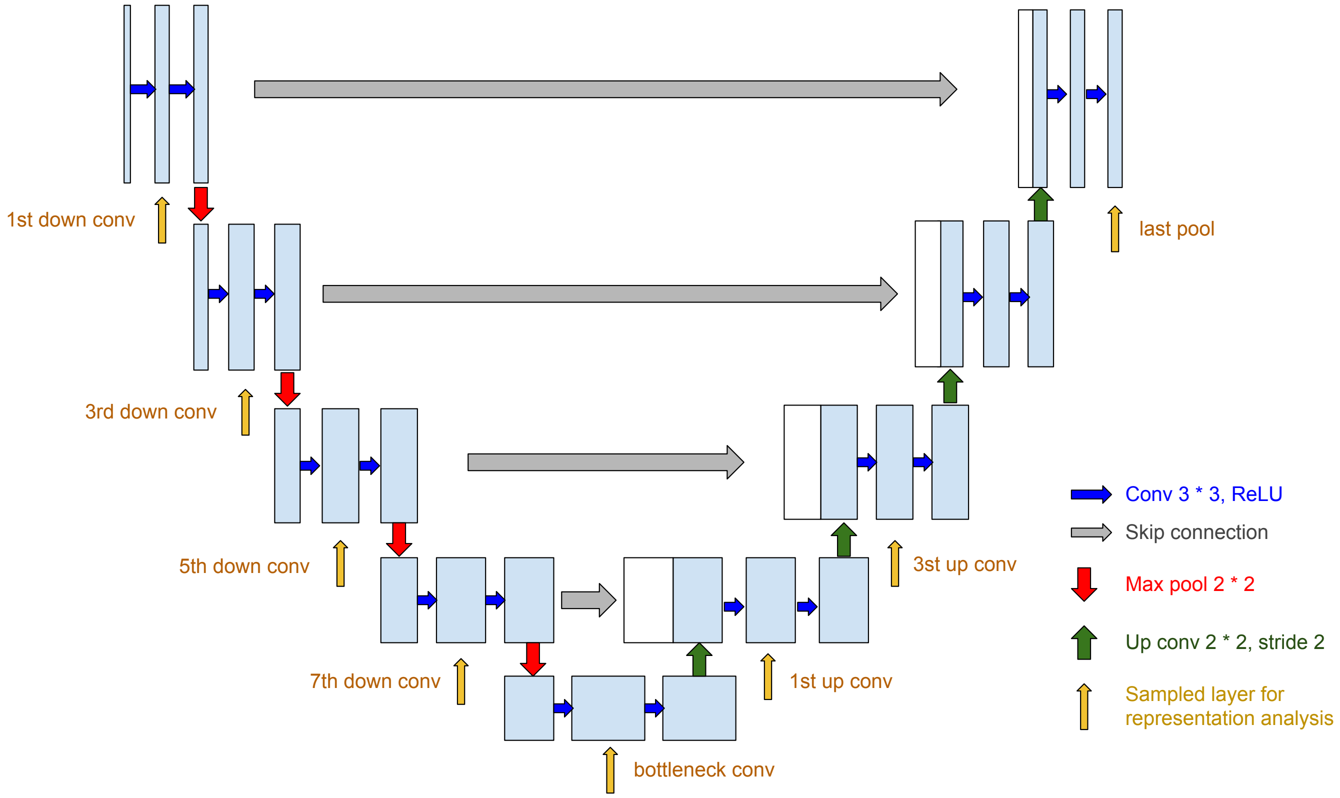


Figure 4. U-Net model diagram [12]. The arrows with different colors indicate different operations. The input is a nine-band image, and the output is a three-band image with the probability of each class for each pixel.

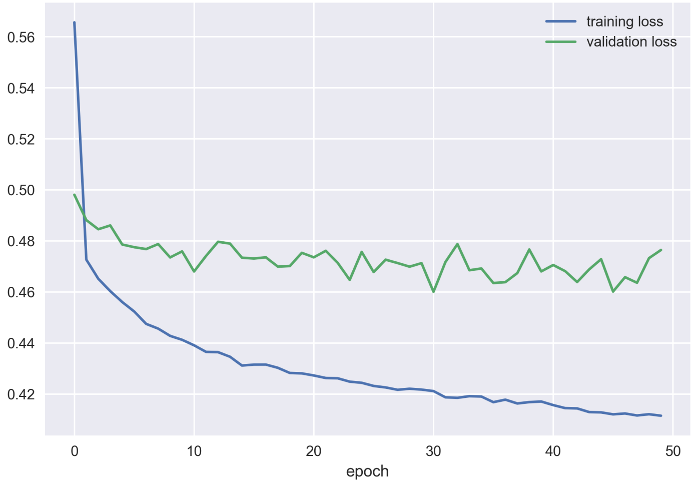


Figure 5. The training loss curve. The blue line corresponds to the training loss. It decreases over epochs indicating the model is learning features from data. The green line corresponds to the validation loss. It first decreases and then becomes relatively steady, meaning the model has converged.

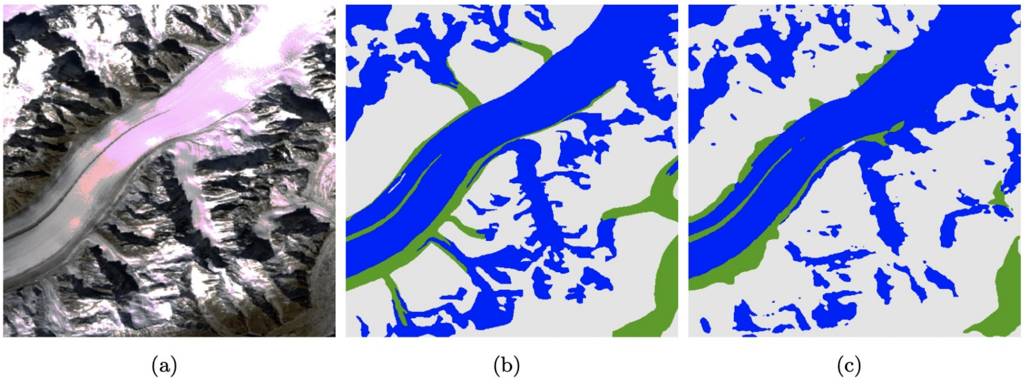


Figure 6. (**a**) Raw image patch. (**b**) Mask patch. (**c**) Predicted patch. The model can detect the position of the bulk of target glaciers but fails to detect connections between glaciers.

4. Representation Analysis

To better understand the modeling process, we attempt to understand how the model captures the original features and how they appear across different layers of the model. To achieve these goals, we used representation analysis to visually interpret the U-Net model layers. We investigated the functions of neurons on specific inputs by passing data through the model and visualizing the activations of each neuron.

The U-Net model starts with down-sampling layers that condense information for distinguishing labels, followed by up-sampling layers that recover spatial context for each class. We sampled activations from the first, third, fifth and seventh downsampling convolutional layers, the second middle convolutional layer, the first and third upsampling convolutional layers and the last pooling layer. We also annotate those sampled activations in Figure 4. We present the visualization of the sampled activations in Figure 7 where the rows correspond to the sampled activations.

This figure shows that the model-learned borders between labels become clearer and more accurate during the initial down-sampling, and these activations are recovered in the last few up-sampling layers. However, the U-Net deeper layers and bottleneck fail to learn any information since the corresponding rows are all black. This indicates that the bottleneck is bypassed, with most activations flowing through skip connections. Guided by this visualization, we may infer that removing these final encoding layers could result in comparable performance with less computation. Consistent with this observation, recent literature has suggested that U-Net does not seem to learn long-range spatial relationships. For example, Malkin et al. [33] observed that a segmentation model trained on a landcover segmentation misclassifies roads when they are interrupted by trees.

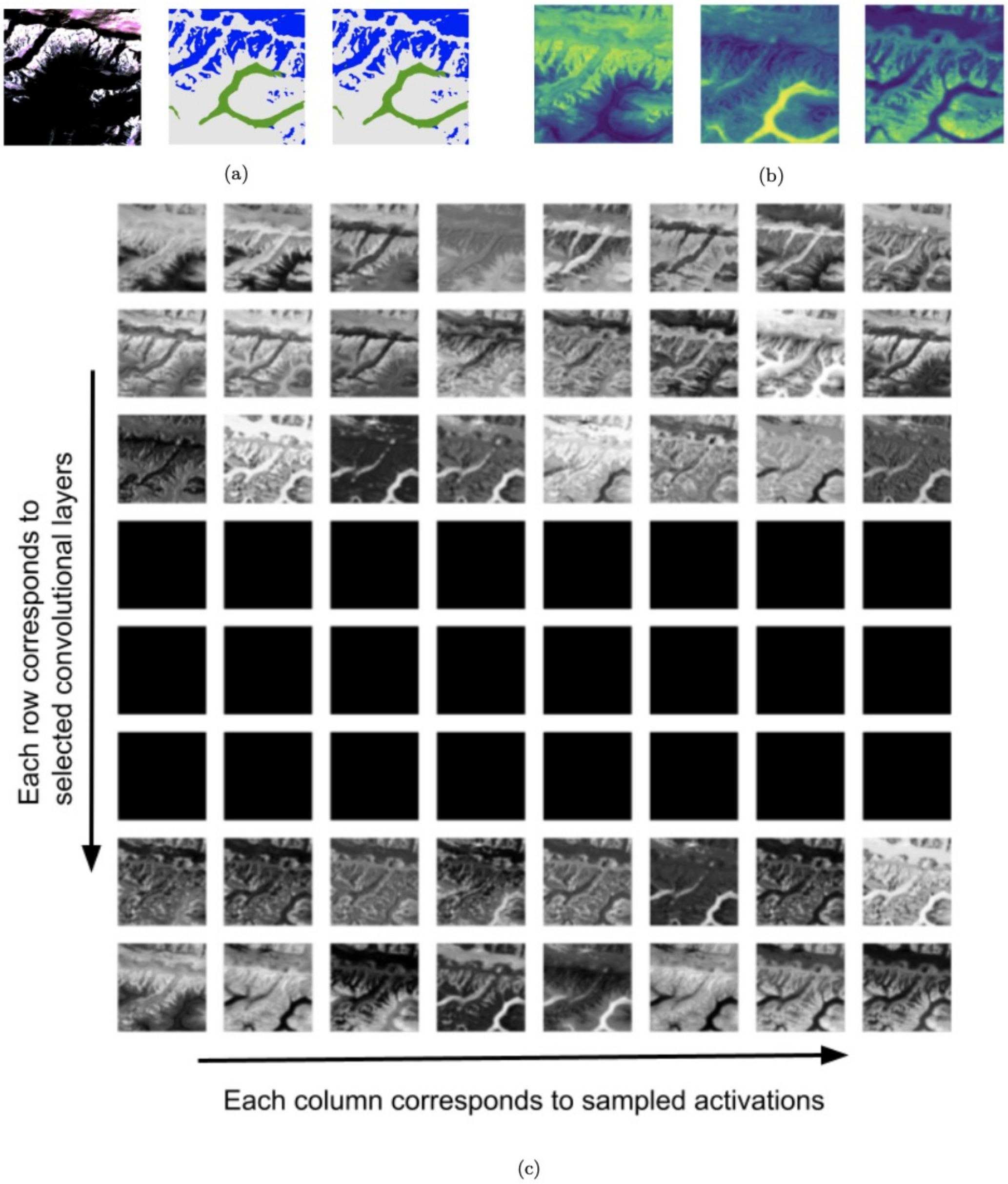

(a)
(b)
Each row corresponds to selected convolutional layers
Each column corresponds to sampled activations
(c)

Figure 7. (**a**) Original figure, true labels and the prediction. Blue, green and gray represent clean-ice glaciers, debris-covered glaciers and background, respectively. (**b**) Visualization of the segmentation layer. The three panels represent clean ice, debris and background, respectively. (**c**) Activations of one satellite image across eight convolutional layers of the U-Net model. For each layer, we randomly plot eight activations in grayscale. Each row corresponds to different convolutional layers, and each column corresponds to different sampled activations. From the top to the bottom rows, they correspond to the first, third, fifth and seventh downsampling convolutional layers, the second middle convolutional layer (bottleneck), the first and third upsampling convolutional layers and the last pooling layer. We also annotate these layers in the model architecture diagram in Figure 4. We observe that the activations capture basic features at the first layer and become more blurred in deeper downsampling. The model appears to skip the bottleneck layer on these example patches, since the associated activations do not pass the ReLU threshold.

5. Error Analysis of Prediction Results

In this section, we introduce a visualization interface built using the R Shiny package [2]. It provides a visual error analysis of model prediction results. Below, we first introduce the design and the functionality of this interface.

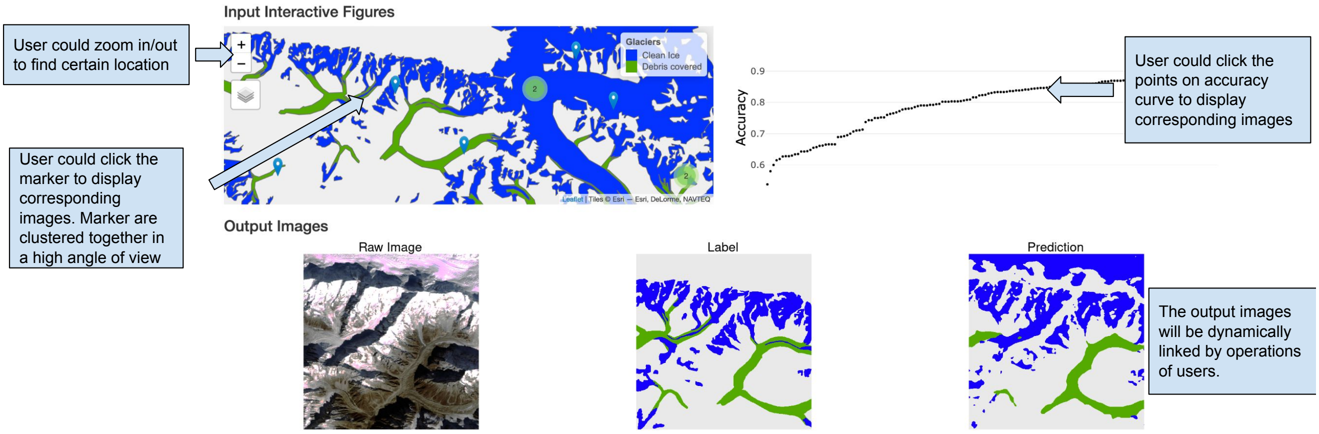
Then, we use our prediction results as an example and explore how, with the help of this interface, we are able to detect a problem in the raw data that is otherwise difficult to detect. This visual interpretation approach could be generalized to other geospatial deep-learning analysis and are helpful when interpreting model prediction results.

5.1. Interactive Interface Introduction

We build an online interactive interface with the Shiny package; it can be accessed at https://tinyurl.com/yc6bj59z accessed on 6 March 2022. It allows users to interact with components representing different aspects of the data. It also links users' interaction with the corresponding prediction results. Figure 8 presents an annotated screenshot of the interface. In the upper panels of the interface, we place a glacier map and an accuracy curve of the training patches. These two parts are both dynamically linked to the associated data patches in the lower panels of the interface.

These patches include a raw satellite image, a ground truth label mask and the prediction results made by the model. The purpose of this design is to enable users to zoom in and out of the glacier map so that they may select specific regions of interest. By clicking different markers on the map, users will be brought to the corresponding images, shown in the space below. The map's markers cluster together when the user zooms out, preventing labels from obfuscating one another.

Moreover, the interface allows users to click points on the accuracy curve calculated by the pixel-wise classification accuracy. In this way, users may compare prediction results with the label masks and raw images. This allows users to easily view training patches with varying performance, facilitating a more comprehensive understanding of model prediction properties. Via such a comparison, we may discover problems that cause low accuracy. For example, from patches with low accuracy, users could identify areas where the model fails and specific glacier patterns on which the model has poor performance.


User could zoom in/out to find certain location

User could click the marker to display corresponding images. Marker are clustered together in a high angle of view

User could click the points on accuracy curve to display corresponding images

The output images will be dynamically linked by operations of users.

Figure 8. Screenshot of the Shiny app with annotations. Users could interact with the app through the upper part of the app. Users could click the map and loss curve to switch the prediction results in the lower part of the app.

5.2. Error Analysis Discussion

We illustrate the use of interactive visualization to support error analysis of glacier segmentation results. Figure 9 presents an example of a test patch with relatively low accuracy. Surprisingly, based on the images in the lower panel, the model predicts the labels of the majority of the pixels in the raw satellite image correctly. The reason why it has low accuracy is that the label mask is not fully labeled—labels were only available in Nepal, not China. Compared with the map, we find that the upper right part of this area is within the Chinese border—it appears that the labeling campaign deliberately excluded glaciers within the Chinese border. Since this patch includes substantial areas within this excluded area, many predictions are mistakenly declared as false positives.

Guided by this observation, the machine learning scientist may decide to supplement the existing labels with more from within the Chinese border. Alternatively, the scientist may choose to limit evaluation to only those image patches that lie fully within Nepal. This structure is straightforward to discover with the help of this interactive panel, but it would require a more deliberate effort without it. This visual approach could be generalized to other real cases in geospatial machine-learning problems and gives researchers a hint about how to improve their prediction results.

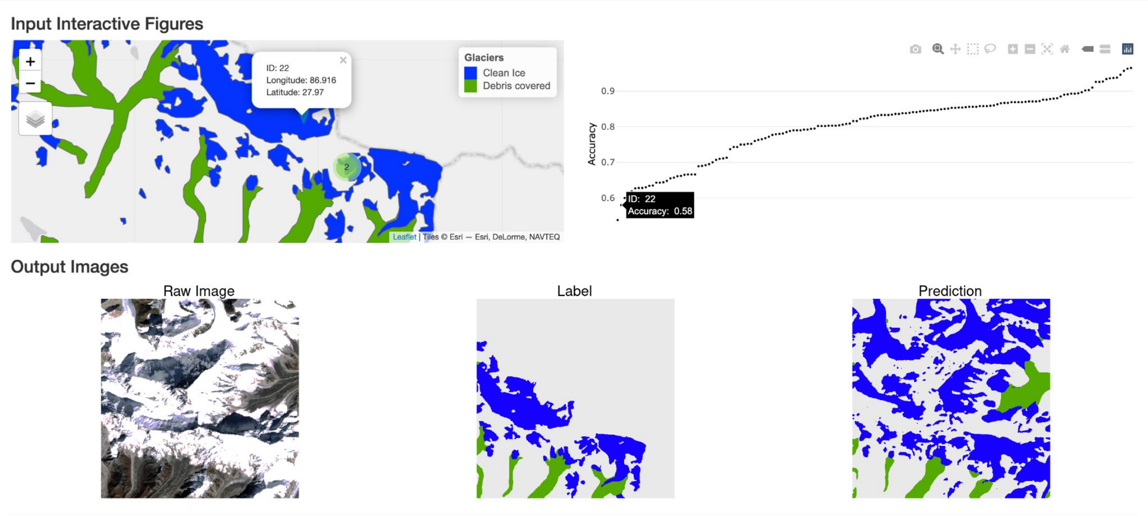


Figure 9. An example of prediction results with low accuracy. We notice that the prediction result actually matches the real label of the raw image well; however, the problem is that the label mask is incomplete, i.e., the upper right part is missing. Compared with the map, we find that the missing-label area is within the Chinese border. The glaciers within the Chinese border are not labeled.

6. Conclusions

We used visualization to provide guidance for machine-learning supported glacier segmentation. During preprocessing, we detected potential issues in the raw data, such as label imbalance and divergent feature scales via strategic static and interactive visualizations. We also provided sanity checks of the preprocessed data to demonstrate their appropriateness for modeling. In Section 3.2, based on the performance of training and validation losses, it appears that the model learned from the data over epochs and was trained properly after 30 epochs.

To better understand the internal training process, we visualized model activations. Based on the activations, we found that the deepest layers were bypassed with most activations flowing through skip connections. To support prediction error analysis, we built an online interactive visualization to display and critique the prediction results. Users can easily interact with the app to discover patterns across patch prediction results. We shared an example where the app revealed an issue with the source labels.

We release our code at github repository (https://github.com/krisrs1128/geo_mlvis, accessed on 6 March 2022). We provide the code for data preprocessing, model training and inference, representation analysis and Shiny app definition. We hope this code can be re-used by others seeking visualization or representation analysis of geospatial deep-learning models.

Author Contributions: Formal analysis, Minxing Zheng and Xinran Miao; Project administration, Kris Sankaran; Supervision, Kris Sankaran; Validation, Kris Sankaran; Visualization, Minxing Zheng and Xinran Miao; Writing—original draft, Minxing Zheng and Xinran Miao; Writing—review and editing, Minxing Zheng, Xinran Miao and Kris Sankaran. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to express our thankfulness to all the members in our Latent Structure Lab (https://krisrs1128.github.io/LSLab/, accessed on 6 March 2022) for their questions during preliminary presentations and feedback on drafts. We gained much help from them and extended our ideas in many aspects of this paper. Thanks to Keith Levin and Hyunseung Kang at UW-Madison who provided great suggestions for the Shiny app. We thank the Center for High Throughput Computing (CHTC) at UW-Madison, which provided our computational resources.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

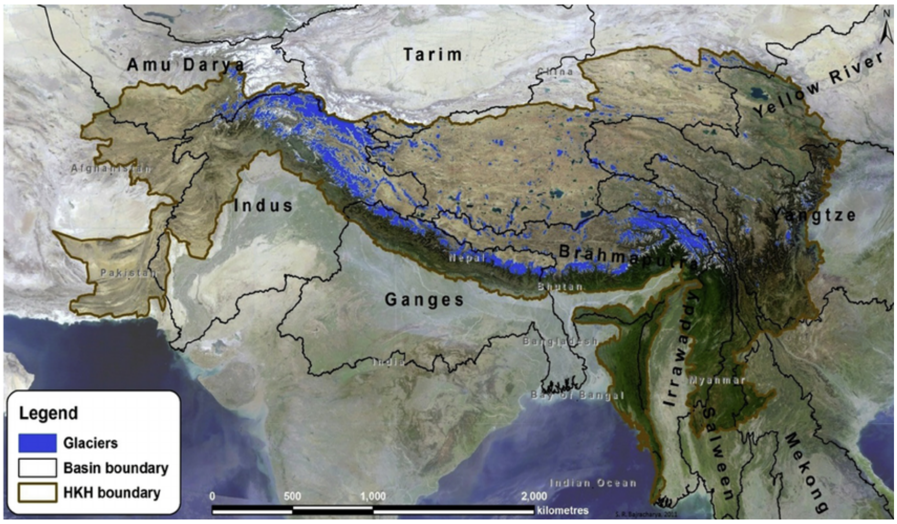


Figure A1. A geographical map of Hindu Kush Himalayas region, reprinted from [34,35].

This appendix discusses the color distortions that appear after preprocessing. In particular, we explore the reasons for the appearance of "pink" regions in the preprocessed patches in Figures 3a, 7a, 8 and 9. To this end, we take a closer look at these values in each color channel (red, green and blue), and we compare values before and after preprocessing in Figure A2. We find that the limit of majority of the values (excluding outliers) of the red channel is smaller than that of the blue and green channels. Therefore, when we equalize each band into the same range, the red channels are stretched into a relatively larger range (compared with blue and green). Consequently, the processed image looks redder as we present in Figure 3.

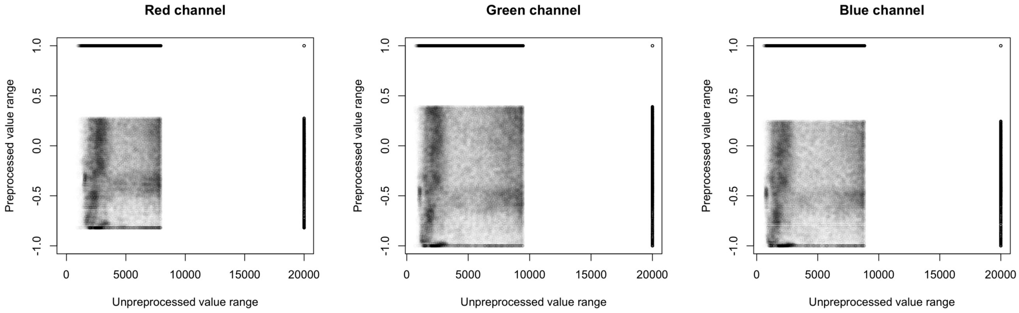


Figure A2. An example of comparison of "Unprocessed vs. Processed" values of each color channel. We found that, in the unprocessed data, the red channel tends to have a smaller range for the majority of values (excluding outliers), compared with the green and blue channels. Histogram equalization brings the ranges for all channels closer to one another, resulting in the color distortion visible in the visualization of the preprocessed data.

Appendix B

This appendix provides further examples and discussion of raw input images and labels. Figure A3 displays the input patches and output masks from raw data. The nine channels complement information with each other in the sense of distinguishable borders between labels as well as variations in resolution across channels.

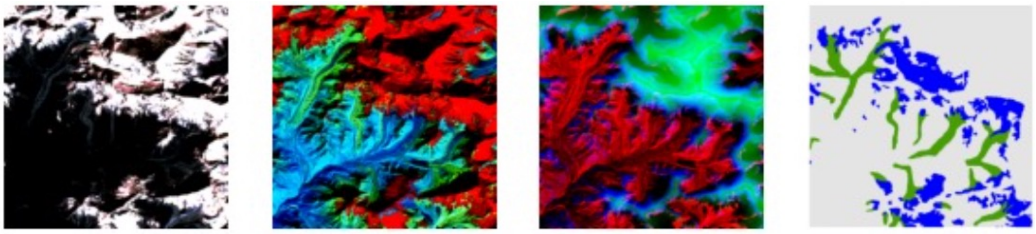


Figure A3. An example of input channels and output labels. We visualize the nine input channels (first three panels) in groups of three and the output labels (last column). In the last panel, blue, green and gray represent clean-ice glaciers, debris-covered glaciers and background, respectively.

References

1. *The Status of Glaciers in the Hindu Kush-Himalayan Region*; International Centre for Integrated Mountain Development (ICIMOD): Patan, Nepal, 2011.
2. Chang, W.; Cheng, J.; Allaire, J.; Sievert, C.; Schloerke, B.; Xie, Y.; Allen, J.; McPherson, J.; Dipert, A.; Borges, B. Shiny: Web Application Framework for R. R Package Version 1.7.1; 2021. Available online: https://rdrr.io/cran/shiny/ (accessed on 26 March 2022).
3. Buja, A.; McDonald, J.A.; Michalak, J.; Stuetzle, W. Interactive data visualization using focusing and linking. In Proceedings of the second Conference on Visualization'91, San Diego, CA, USA, 22–25 October 1991; pp. 156–163.
4. Anselin, L.; Syabri, I.; Smirnov, O. Visualizing multivariate spatial correlation with dynamically linked windows. In Proceedings of the CSISS Workshop on New Tools for Spatial Data Analysis, Santa Barbara, CA, USA, 20–23 March 2002.
5. Anselin, L. Interactive Techniques and Exploratory Spatial Data Analysis. 1996. Available online: https://researchrepository.wvu.edu/rri_pubs/200/ (accessed on 26 March 2022).
6. Hibbard, W.; Santek, D. Visualizing large data sets in the earth sciences. *Computer* **1989**, *22*, 53–57. [CrossRef]
7. Tasnim, J.; Mondal, D. Data Reduction and Deep-Learning Based Recovery for Geospatial Visualization and Satellite Imagery. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 5276–5285.
8. Keim, D.A.; Panse, C.; Schneidewind, J.; Sips, M.; Hao, M.C.; Dayal, U. Pushing the Limit in Visual Data Exploration: Techniques and Applications. In *KI 2003: Advances in Artificial Intelligence*; Günter, A., Kruse, R., Neumann, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 37–51.
9. Keim, D.; Panse, C.; Sips, M.; North, S. Visual data mining in large geospatial point sets. *IEEE Comput. Graph. Appl.* **2004**, *24*, 36–44. [CrossRef] [PubMed]
10. Janik, A.; Sankaran, K.; Ortiz, A. Interpreting Black-Box Semantic Segmentation Models in Remote Sensing Applications. 2019. Available online: https://diglib.eg.org/handle/10.2312/mlvis20191158 (accessed on 26 March 2022).

11. Humer, C.; Elharty, M.; Hinterreiter, A.; Streit, M. *Interactive Attribution-based Explanations for Image Segmentation*; Johannes Kepler University Linz: Linz, Austria, 2022.
12. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
13. Siddique, N.; Paheding, S.; Elkin, C.P.; Devabhaktuni, V. U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. *IEEE Access* **2021**, *9*, 82031–82057. [CrossRef]
14. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–11.
15. Pan, Z.; Xu, J.; Guo, Y.; Hu, Y.; Wang, G. Deep learning segmentation and classification for urban village using a worldview satellite image based on U-Net. *Remote Sens.* **2020**, *12*, 1574. [CrossRef]
16. Bai, Y.; Mas, E.; Koshimura, S. Towards operational satellite-based damage-mapping using u-net convolutional network: A case study of 2011 tohoku earthquake-tsunami. *Remote Sens.* **2018**, *10*, 1626. [CrossRef]
17. Freudenberg, M.; Nölke, N.; Agostini, A.; Urban, K.; Wörgötter, F.; Kleinn, C. Large scale palm tree detection in high resolution satellite images using U-Net. *Remote Sens.* **2019**, *11*, 312. [CrossRef]
18. Gonzalez, J.; Sankaran, K.; Ayma, V.; Beltran, C. Application of semantic segmentation with few labels in the detection of water bodies from perusat-1 satellite's images. In Proceedings of the 2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS), Santiago, Chile, 21–26 March 2020; pp. 483–487.
19. He, Q.; Zhang, Z.; Ma, G.; Wu, J. Glacier Identification from Landsat8 Oli Imagery Using Deep U-Net. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *5*, 381–386. [CrossRef]
20. Baraka, S.; Akera, B.; Aryal, B.; Sherpa, T.; Shresta, F.; Ortiz, A.; Sankaran, K.; Ferres, J.L.; Matin, M.; Bengio, Y. Machine Learning for Glacier Monitoring in the Hindu Kush Himalaya. *arXiv* **2020**, arXiv:2012.05013.
21. Holzmann, M.; Davari, A.; Seehaus, T.; Braun, M.; Maier, A.; Christlein, V. Glacier Calving Front Segmentation Using Attention U-Net. *arXiv* **2021**, arXiv:2101.03247.
22. Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; Lipson, H. Understanding neural networks through deep visualization. *arXiv* **2015**, arXiv:1506.06579.
23. Mahendran, A.; Vedaldi, A. Understanding deep image representations by inverting them. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5188–5196.
24. Erhan, D.; Courville, A.; Bengio, Y.; Vincent, P. Why does unsupervised pre-training help deep learning? In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Sardinia, Italy, 13–15 May 2010; pp. 201–208.
25. Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 4905–4913.
26. Olah, C.; Mordvintsev, A.; Schubert, L. Feature visualization. *Distill* **2017**, *2*, e7. [CrossRef]
27. Erhan, D.; Bengio, Y.; Courville, A.; Vincent, P. Visualizing Higher-Layer Features of a Deep Network. *Univ. Montr.* **2009**, *1341*, 1.
28. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv* **2013**, arXiv:1312.6034.
29. Raghu, M.; Gilmer, J.; Yosinski, J.; Sohl-Dickstein, J. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *arXiv* **2017**, arXiv:1706.05806.
30. Qin, Z.; Yu, F.; Liu, C.; Chen, X. How convolutional neural network see the world-A survey of convolutional neural network visualization methods. *arXiv* **2018**, arXiv:1804.11191.
31. Hohman, F.; Kahng, M.; Pienta, R.; Chau, D.H. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 2674–2693. [CrossRef] [PubMed]
32. Williams, M.W. The Status of Glaciers in the Hindu Kush–Himalayan Region. *Mt. Res. Dev.* **2013**, *33*, 114–115.. [CrossRef]
33. Malkin, N.; Ortiz, A.; Jojic, N. Mining self-similarity: Label super-resolution with epitomic representations. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 531–547.
34. Gurung, D.R.; Giriraj, A.; Aung, K.S.; Shrestha, B.R.; Kulkarni, A.V. *Snow-Cover Mapping and Monitoring in the Hindu Kush-Himalayas*; Technical Report; International Centre for Integrated Mountain Development (ICIMOD): Patan, Nepal, 2011.
35. Gertler, C.G.; Puppala, S.P.; Panday, A.; Stumm, D.; Shea, J. Black carbon and the Himalayan cryosphere: A review. *Atmos. Environ.* **2016**, *125*, 404–417. [CrossRef]