Terrain Based Avalanche Path Prediction

Oliver Oxford
oli_ox@yahoo.com

Andrew Schrader
andrew@node.rip

Abstract

Avalanches pose a serious risk for hobbyists and infrastructure in mountainous regions. Accordingly, much research has been done on using machine learning to provide regional risk analysis. While this is highly valuable in mitigating danger, it provides no context for the specific paths or routes which avalanches could take. In this paper, we utilized a modified U-Net architecture to treat avalanche prediction as an image segmentation problem, predicting not only the general avalanche risk, but also their shape and slide paths, allowing more precise risk analysis, safety planning and a deeper understanding of the profiling of avalanches. Our model converged to a binary cross entropy training loss of around 0.38 and a test loss of 0.36, with a consistent dice coefficient over epochs. It was able to reconstruct ground truths masks on testing data to a good degree of precision and predicts realistic avalanche paths with minimal false positives.

1 Introduction

In mountainous regions, an avalanche is when large masses of snow slide down steep mountain slopes. A variety of factors play a role in contributing to avalanche formation. Weather conditions play a significant role - new snow which has not had the chance to properly settle is more prone to slipping, strong wind can cause heavier snow accumulations in certain regions, and specific temperature ranges can vary the mechanical properties of the snow and therefore the avalanche risk. Another vital contributor to avalanche risk is the terrain features. Surface aspect influences sun and wind blow, valleys shape the paths which avalanches take, and slopes with an angle of 35-40° have the highest chance of causing slides (Schweizer, Bruce Jamieson, and Schneebeli 2003).

Avalanches can be highly dangerous, causing more yearly deaths in the United States than earthquakes or landslides (Council et al. 1990). Because of this, avalanche prediction is a critical problem in mountainous regions, especially with increasing outdoor recreational activity and climate variability. Currently, avalanche forecasting services do a good job of predicting regional avalanche risk, but are not able to localize those predictions to anticipate specifically where those avalanches can occur. In this paper we aim to use image segmentation to create a highly localized specific predictor of the shape and area in which avalanche slips are likely. By basing our model on topological survey data and analyzing labeled shapes of past avalanche debris, we can predict and overlay the most likely path avalanches would take, contributing to a better understanding of avalanche behavior and allowing more precise risk analysis and emergency response planning for those entering avalanche prone terrain.

2 Related Work

Much existing work has been done on using machine learning to predict avalanche danger using a variety of different factors. Fromm and Schönberger (2022) and Pozdnoukhov, Purves, and Kanevski (2008) focus on meteorological data and snow pack analysis and use it to predict risk values over a mountain range. Topographical features are also used, such as in Yariyan et al. (2022) and Choubin et al. (2019), who use simpler learning models such as support vector machines and multi-layer

Course project for UBC CPSC 440/550, April 2025.

perceptrons. Deep learning models have also been explored for avalanche prediction, such as Chen et al. (2022) who use CNNs with a variety of topological metrics to predict avalanche risk.

Common to all these attempts, however, is that they seek to make regional predictions of risk, and while they are specific enough to account for local topography (see (Figure 1 for an example), they do not predict information about the specific path which the avalanche would take should it occur. The training targets these papers use is point data of where avalanches occurred, and accordingly they learn only the general risk for an area. Where our work differs is we use labeled masks of the debris fields of past avalanches, giving a specific boundary and path of the slide. We then frame the problem as a segmentation task, seeing to predict the exact path of the avalanche on a pixel by pixel basis based on high resolution topographical data. To the best of our knowledge the use of machine learning techniques to predict shape and area of avalanches is a novel contribution to the field of avalanche risk analysis. Other work relating to avalanche shape mapping was done by Hafner et al. (2022), who used deep learning with satellite imagery to automatically detect the visual debris and highlight areas where avalanches occurred, but this work is focused on retroactive identification rather than making generalized predictions about future avalanche paths.

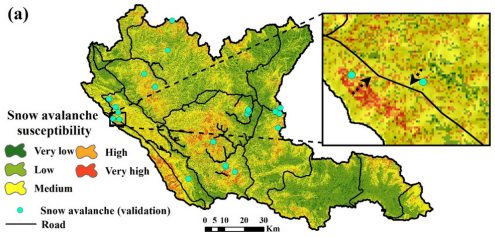


Figure 1: An example of the risk prediction output from Chen et al. (2022)

3 Methodology - Data

3.1 Acquisition

To get target data of avalanche shapes, we reached out to Hafner et al. (2022) and they were kind enough to provide us with the labeled dataset from their research. This contained a shapefile with the polygons shapes of past avalanches, drawn by an expert who analyzed satellite imagery for traces of slides. Crucially, this data also contained global position metadata, which enabled us to map the latitude and longitude of our labeled polygons precisely onto the topographic map of the mountain region. This is shown in Figure 2. One important note about this data is that it only includes the visible slides at a single point in time. Accordingly, we expect our model to predict some more positives than this data shows, since it is likely that there are places with potential for slides that had not been triggered at the point of image capture. Despite this, the data was gathered after a period of high risk conditions and has a high concentration of positive samples, so we believe it is still an effective source for our purposes.

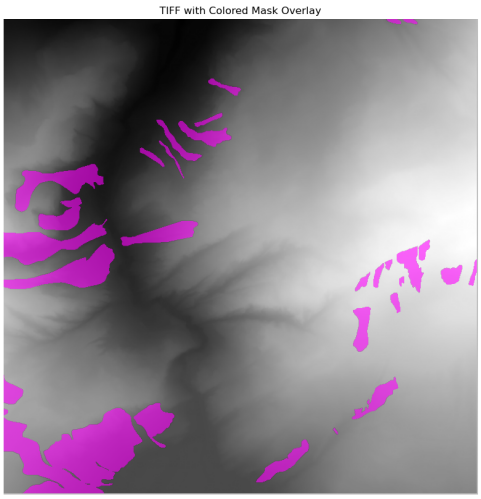


Figure 2: The avalanche mask overlaid on the mountain topography

To predict avalanche shapes, we also needed data of the mountain region to act as input for our model. Initially, we sought to use satellite imagery, however this ran into numerous issues. Finding up to date images with adequate precision in highly specific regions proved difficult. Instead, we elected to use digital elevation model (DEM) data. This provides the specific shape of the mountain, is not specific to any time period, and is available in extremely high precision. We acquired our DEM data from the swiss government (Federal Office of Topography swisstopo 2022). Obtained using LiDAR technology, this data came in a cloud optimized geotiff format and contained elevation values at a 2m resolution in a grid over a 1km x 1km region as well as metadata such as its latitude

and longitude and scale. Essentially, this data is analogous to a 500x500 image with elevation values at pixels instead of RGB colors. We downloaded all tiles for a region carefully bounded to stay within the region for which avalanches were labeled in our target data.

3.2 Loading and Preprocessing

Many of the identified avalanches spanned regions of a few hundred meters up to a kilometer in distance. Accordingly, it was not sufficient to load in each 1x1km tile of topological data, since we want to minimize the cases where only a portion of an avalanche is viewed, since the model needs the full context of the mountain region around the avalanche to detect the features which define its shape. Accordingly, for each sample, we load a 3x3 grid of adjacent tiles and stitch them together. We select center tiles with a step size of 3 to ensure that no tiles are duplicated in multiple samples. We also discount all edge tiles where we lack data to fully surround them. Then, we rasterize the shapefile of avalanche locations and use the global coordinate metadata to trim it to overlay exactly on each specific tile, creating a pixel-based mask of the same shape as the input image.

In total, we get 1079 tiles, each of which is a 1500x1500 pixel image representing altitude values in a 3x3km region. We also generate identically sized image masks for our targets. We then split these tiles using an 80% - 20% split for training and testing data respectively.

3.3 Feature construction

To speed up training and ensure the model is able to learn properly, we utilized domain knowledge to calculate additional features for the model to learn off.

- Slope: research shows that maximal avalanche risk occurs on slopes of roughly 35°. Accordingly, we calculate the gradient of the image at each pixel in both the x and y direction, then set the final slope value of the pixel to be $\sqrt{dzdx^2 + dzdy^2}$.
- Aspect: since our slope calculation is directionless, we also want to encode the direction of the slope for our model. Since $(-dzdy, dzdx)$ is the steepest direction of the slope, we calculate *np.arctan2*$(-dzdy, dzdx)$ to get the angle of this slope. Since we are in a mapping context, $-dzdy$ is used to make 0 point north.
- Curvature: finally, we calculate the Laplacian to get the second derivative of our image by summing the gradients of $dzdx$ and $dzdy$.

Since taking the derivative of an image exaggerates noise, we pass a gaussian blur filter over the image first. We also standardize the data using the mean and standard deviation calculated over the full dataset, since most elevation values are originally in the low 2000s and we want small usable numbers. By doing this we still retain information about the relative altitude which may be a meaningful property for the model to consider. Finally, we downsample the images by a factor of 0.5. This speeds up training with minimal accuracy loss, since the primary influencing factors of avalanche shape is the large scale terrain patterns, low level detail is not especially important to maintain.

3.4 Dataset Versions

While binding the avalanche shape data (Hafner et al. 2022) to the DEM topology data (Federal Office of Topography swisstopo 2022), each avalanche shape had to be assigned to the correct DEM tile. The avalanches were labeled for a specific region on the Swiss alps; however, the DEM data was only available by manually selecting an area on the website. In lieu of a programmatic approach, we had to be very careful about selecting the correct area bound for the DEM data. Any extra DEM tiles in the dataset would be avalanche-free zones, inevitably skewing the model towards predicting no avalanche.

Dataset V1 of DEM data was selected to quickly develop an mvp. The area bound of the DEM data was not very tight along the northern alps.

Dataset V2 of the DEM data was chosen to error on the side of throwing out avalanche shape data in an effort make sure all avalanche free 1x1km DEM tiles were actually avalanche free. This resulted in slightly less avalanche examples, but a much better dataset overall.

3.5 Dataset Limitations

One important limitation of our dataset is that it represents a distinct collection of avalanches from a particular time period, introducing bias toward the conditions and events of that specific era.

4 Methodology - Model Iterations

Over the course of this project's development, we tried multiple different models, and the iteration process taught us numerous things about our problem domain. While we tested many unique model configuration, the following are a few of the main model iterations as well as our learning.

4.1 Mini CNN

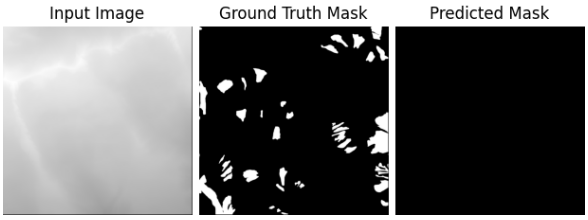
Our early models took a long time to train, so in an effort to speed things up, we tested a minimalist model with only two layers of convolutions. Interestingly, this still ran fairly slowly despite having very few parameters, which spurred us to investigate our data loading process and uncover an inefficiency in the way our masks were calculated. After fixing this, our training times increased rapidly. What was interesting though, is that the model predicted a 0.5 probability of avalanche at all pixels, essentially failing to make any meaningful prediction. This also used Dataset V1 and was biased towards no avalanche.

4.2 Feature Expansion

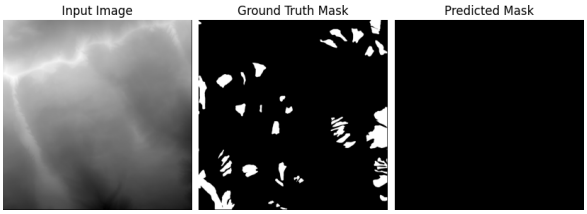
In this model we shifted to using Dataset V2 and addressed the issues with the mini CNN, which allowed us to get our first model which effectively made predictions. Firstly, since the mini CNN only had two convolutional layers, it was only able to sees a range of 5 pixels at a time, which corresponds to a 10x10m space in the real world. Mountain features like peaks, valleys and mountain faces occur on a much larger scale than this, so a model with a larger receptive was needed. The second problem in the model is that while technically a CNN should be able to calculate important features such as the slope, aspect and curvature with enough parameters, doing so is expensive and time consuming and relies on the model not getting stuck at local minima. Accordingly, this was an opportunity for us to utilize our domain knowledge and pre-compute these features to help the model out.

4.3 Larger Models

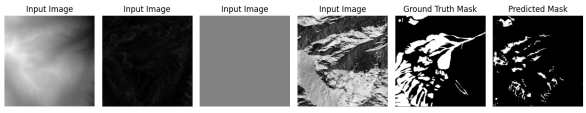
We then proceeded to test a variety of different permutations of our model. We modified the number of layers and parameters



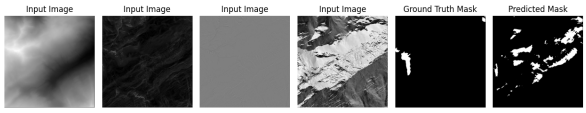
(a) Naive CNN Prediction 1



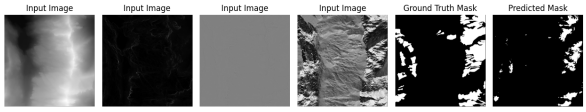
(b) Mini CNN Prediction 1



(c) Feature Expansion Prediction 1



(d) Feature Expansion Prediction 2



(e) Feature Expansion Prediction 3

Figure 3: Model predictions from different CNN architectures.

our model used, and increased the number of training epochs ran. In the end, we found that beyond a certain point, adding more parameters did not increase our performance significantly, likely since the task is simple enough that once the model can detect the relevant mountain features it does not need to do particularly complex analysis to make effective predictions. The final architecture of our model which we found to be most effective is detailed below.

5 Final Model Architecture

Our model is based on the U-Net architecture for image segmentation (Huang et al. 2020). U-Nets are characterized by a sequence of downsampling step which feature two 3x3 convolutions separated by nonnonlinearities and a 2D max pooling layer. Therefore after each step the dimensions of the image shrink by a factor of 2. At the bottom, the encoding is passed through a bottleneck layer, before it proceeds through a series of upsampling stages which invert the downsampling structure and return the model to its originally dimensionality. The defining feature of U-Nets is that in addition to the standard forward progression of the model, skip connections also connect the corresponding encoding and decoding stages, allowing the model to utilize both the latent representation built through the encoding process as well as the original data from that level during the decoding process. Since the features we are seeking to detect can take place over the range of 100s of pixels, we modified our U-Net construction to emphasize a large receptive field without too much parameter bloat. First this is accomplished by a fairly deep series of 5 downsampling steps. The bottleneck also features an atrous convolution with dilation 2, further growing the receptive field with the breadth of a 5x5 convolution without the corresponding parameter or training time bloat. In the up direction, the model follows standard U-Net construction, mirroring the downward path and receiving as input both the previous layer and output from its corresponding downsampling phase. In total, this results in a receptive field of 377 pixels. The growth of this is pictured here **??** Since our input is downsampled by a factor of 1/2, and the initial image resolution was 2m per pixel, this corresponds to a $377 * 4 = 1508$ meter scale at which our model can recognize features.

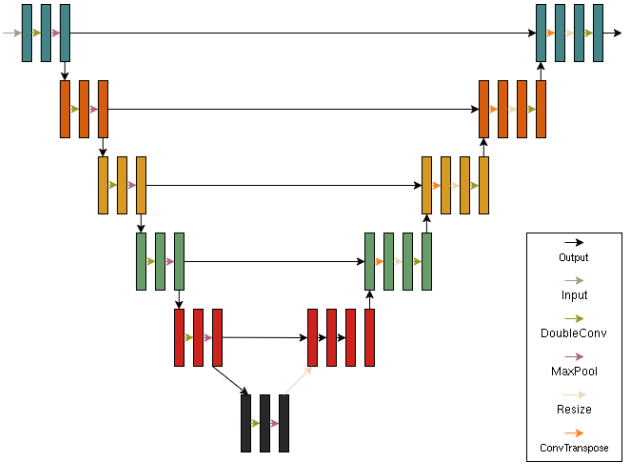


Figure 4: U-Net architecture

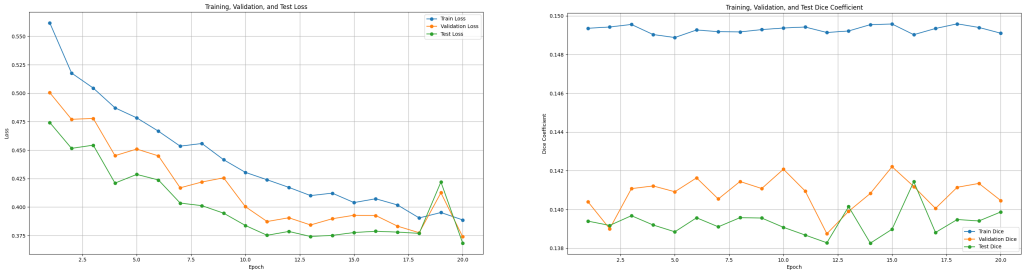
5.1 Training configuration

We used a learning rate of $3e-4$, batch size of 2 and ran 20 training epochs. Our loss function was binary cross entropy loss and we used the AdamW optimization algorithm.

6 Results

Our model converged to a binary cross entropy loss of 0.38 for training data, 0.37 for validation, and 0.36 for test. The Dice Coefficient stayed relatively the same for training, validation, and test. This suggests that the model is reliably segmenting the images across epochs (Bardis et al. 2020).

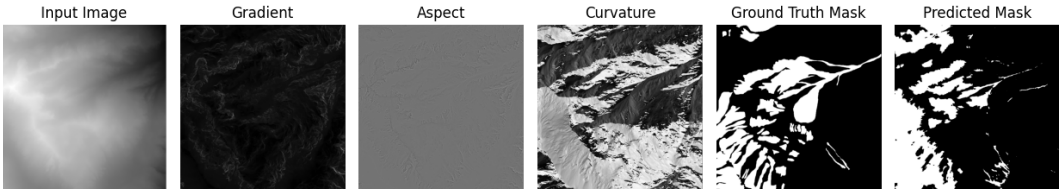
Figure 6 displays four test examples to illustrate the model's performance. Each input image represents the 3×3 km stitched DEM topography image for a specific test example. The next three



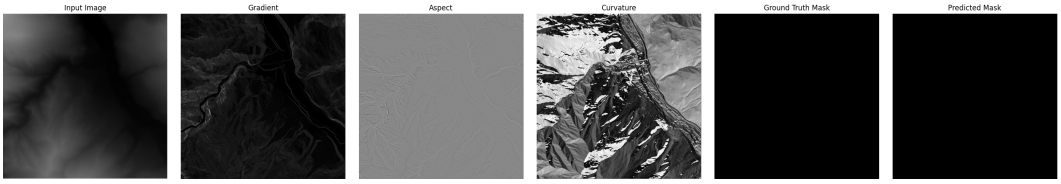
(a) Final Model Loss per Epoch

(b) Final Dice Coefficient

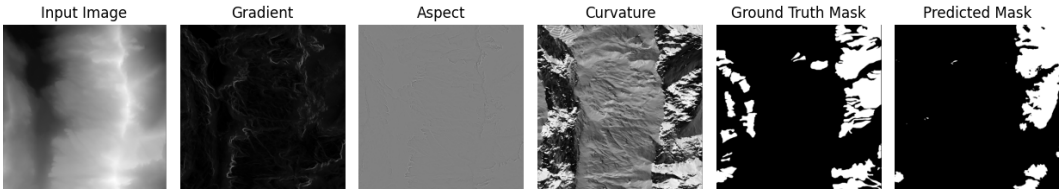
Figure 5: Comparison of Final Model Performance



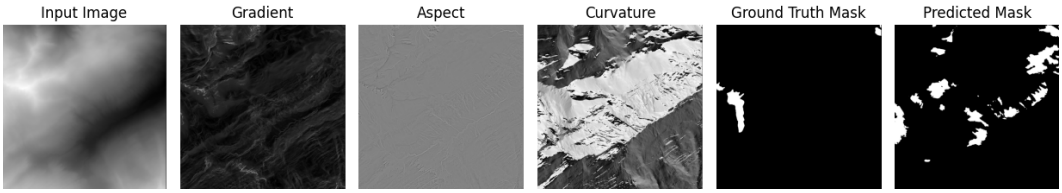
(a) Final Model Prediction 1



(b) Final Model Prediction 2



(c) Final Model Prediction 3



(d) Final Model Prediction 4

Figure 6: Visualizations of Final Model Performance

images show the gradient, aspect, and curvature derived from the DEM for that example. The final two images show the avalanche ground truth shape and the model's predicted shape, respectively.

The first three test examples correspond to those previously shown in the Feature Expansion model (Figure 3). Specifically, Figure 6a corresponds to Figure 3c, Figure 6b to Figure 3d, and Figure 6c to Figure 3e.

Compared to the Feature Expansion model, the final model predictions show that the yes-avalanche pixels are now clustered more closely around the ground truth avalanche areas. While the total predicted avalanche area appears smaller, the number of false positives has also noticeably decreased. This improvement is particularly clear when comparing the false positives in Figure 6b to Figure 3d. Semantically, the predicted avalanche paths represent realistic routes which commonly align with both the ground truth labels and our domain knowledge evaluation.

Figure 6a represents a success case of our model. Using only the mountain topography it captures all of the avalanche points and predicts fall paths which align well to the geometry of the true avalanche occurrences.

The final test example, Figure 6b, shows a 3×3 km area correctly predicted as no-avalanche, demonstrating the model's ability to avoid false positives in non-avalanche zones.

Figure 6c shows very accurate predictions of avalanche paths on the eastern slope of the ridge, but has false negative labels on the series of smaller avalanches to the west.

Figure 6d also highlights an interesting challenge. Although the false-positive rate improved, the curvature map shows a prominent ridge spanning the entire region — a feature often associated with avalanche-prone areas in the rest of the dataset. Based on our domain knowledge, we believe that the model is reasonable in its predictions here, and this could be a case where the ground truth labels are lacking as a product of the data collection method.

7 Discussion

Overall, our model was quite effective at predicting avalanche regions and made consistent and reasonable predictions based on both our domain knowledge and labeled ground truth examples.

One shortcoming of our process was that the avalanche labels were taken at a single point in time in a single region of the Swiss Alps. Accordingly, our model likely overfits to Swiss geography and the specific weather conditions of that time. The effects of topography on avalanches are universal, so we believe the model will still generalize well, but different weather conditions in particular can influence the way in which snow slides, so our model would likely overfit less if trained on more diverse data. To take it a step further, future work could include actually feeding in current weather conditions as input data, which would make the model significantly more precise in tailoring its prediction to specific user scenarios. The main obstacle to this is finding sufficient labeled data across varied time frames.

8 Code

Source code for our project can be found here: https://github.com/schr4der/avalanche/

Acknowledgments

Special thanks to Hafner et al. (2022) for granting us access to the dataset from their work.

References

Michelle Bardis, Roozbeh Houshyar, Chanon Chantaduly, Alexander Ushinsky, Justin Glavis-Bloom, Madeleine Shaver, Daniel Chow, Edward Uchio, and Peter Chang (2020). "Deep Learning with Limited Data: Organ Segmentation Performance by U-Net." *Electronics* 9.8. ISSN: 2079-9292. DOI: 10.3390/electronics9081199. URL: https://www.mdpi.com/2079-9292/9/8/1199.

Yunzhi Chen, Wei Chen, Omid Rahmati, Fatemeh Falah, Dominik Kulakowski, Saro Lee, Fatemeh Rezaie, Mahdi Panahi, Aref Bahmani, Hamid Darabi, Ali Torabi Haghighi, and Huiyuan Bian and (2022). "Toward the development of deep learning analyses for snow avalanche releases in mountain regions." *Geocarto International* 37.25, pp. 7855–7880. DOI: 10.1080/10106049.2021.1986578. eprint: https://doi.org/10.1080/10106049.2021.1986578. URL: https://doi.org/10.1080/10106049.2021.1986578.

Bahram Choubin, Moslem Borji, Amir Mosavi, Farzaneh Sajedi-Hosseini, Vijay P. Singh, and Shahaboddin Shamshirband (2019). "Snow avalanche hazard prediction using machine learning methods." *Journal of Hydrology* 577, p. 123929. ISSN: 0022-1694. DOI: https://doi.org/10.1016/j.jhydrol.2019.123929. URL: https://www.sciencedirect.com/science/article/pii/S0022169419306493.

National Research Council, Division on Engineering, Physical Sciences, Commission on Engineering, Technical Systems, and Committee on Ground Failure Hazards Mitigation Research (1990). *Snow avalanche hazards and mitigation in the United States*. National Academies Press.

Federal Office of Topography swisstopo (2022). *SwissALTI3D: High-precision digital elevation model of Switzerland*. https://www.swisstopo.admin.ch/en/height-model-swissalti3d. Accessed: 2025-04-25.

Reinhard Fromm and Christine Schönberger (2022). "Estimating the danger of snow avalanches with a machine learning approach using a comprehensive snow cover model." *Machine Learning with Applications* 10, p. 100405. ISSN: 2666-8270. DOI: https://doi.org/10.1016/j.mlwa.2022.100405. URL: https://www.sciencedirect.com/science/article/pii/S2666827022000809.

E. D. Hafner, P. Barton, R. C. Daudt, J. D. Wegner, K. Schindler, and Y. Bühler (2022). "Automated avalanche mapping from SPOT 6/7 satellite imagery with deep learning: results, evaluation, potential and limitations." *The Cryosphere* 16.9, pp. 3517–3530. DOI: 10.5194/tc-16-3517-2022. URL: https://tc.copernicus.org/articles/16/3517/2022/.

Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu (2020). "UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation." *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1055–1059. DOI: 10.1109/ICASSP40776.2020.9053405.

A. Pozdnoukhov, R.S. Purves, and M. Kanevski (2008). "Applying machine learning methods to avalanche forecasting." *Annals of Glaciology* 49, pp. 107–113. DOI: 10.3189/172756408787814870.

Jürg Schweizer, J. Bruce Jamieson, and Martin Schneebeli (2003). "Snow avalanche formation." *Reviews of Geophysics* 41.4. DOI: https://doi.org/10.1029/2002RG000123. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2002RG000123. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2002RG000123.

Peyman Yariyan, Ebrahim Omidvar, Foad Minaei, Rahim Ali Abbaspour, and John P. Tiefenbacher (Mar. 2022). "An optimization on machine learning algorithms for mapping snow avalanche susceptibility." *Natural Hazards* 111.1, pp. 79–114. ISSN: 1573-0840. DOI: 10.1007/s11069-021-05045-5. URL: https://doi.org/10.1007/s11069-021-05045-5.