ENV-540 : Image processing for earth observation

Deep learning project : Mapping glaciers with Sentinel-2 imagery and GLIMS database

GitHub Code

Authors:
Malo Olszewski
Ben Kriesel
Romane Maroye
Edison Von Matt

Supervised by:
Gaston Lenczner
Professor:
Devis Tuia

Contents

1 Introduction

In recent decades, the impact of anthropogenic greenhouse gas emissions has led to climate change. In the context of this climate change, Arctic warming has been two to three times greater than the global average near-surface air temperature trend. The concern is that recent warming of the Arctic causes physical environmental changes in Greenland, affecting vegetation and land ice, with the retreat of Arctic sea ice. These alterations will inevitably have repercussions on local ecosystems, socioeconomic, and cultural aspects [1].

In the short and medium term, variations in surface temperatures and precipitation in Greenland are mainly influenced by atmospheric heat advection associated with the North Atlantic Oscillation (NAO). Winter variability of the NAO has three times the impact of summer variability in southern Greenland. Since 1993, there has been a significant increase in surface temperatures in Greenland, with the period 2001-2010 being the warmest since measurements began in the 1780s. The year 2010 was particularly remarkable, with coastal temperatures three times higher than the climatological average from 1960 to 1990 and record melting of the ice cap [1].

This underscores the significance of monitoring the changes in glacier extent and volume in order to understand the overall impact of climate change over time.

This project employs satellite data of two different glaciers in Greenland near Narsarsuaq, where one is used for training and the other one for evaluation, to build a deep learning model performing glacier segmentation. The dataset is composed of images from the Sentinel-2 and GLIMS database, both tiled into 128x128 patches. Sentinel-2 is part of European Earth observation satellite mission which provides high-resolution multispectral data with a global revisit time of 5 days. The GLIMS database will be used as ground-truth reference, GLIMS is a resource that provides coverage of glaciers and ice-covered regions from all around the world. The images cover Greenland from July and August 2023 with a resolution of 10m RGB+NIR.

The project aims to spatially map glaciers and conduct a temporal comparison by employing two different recording periods of the glaciers. First, an optimal model is selected through an evaluative analysis of nine different deep learning models trained on both dates on the train glacier. After finding and selecting the optimal model architecture for the task at hand, hyperparameters and data pre-processing techniques are used to train a final model for each distinct date (July, and August), while also training a model on both dates, to finally compare model performances and deduce temporal changes in the glaciers.

2 Methods

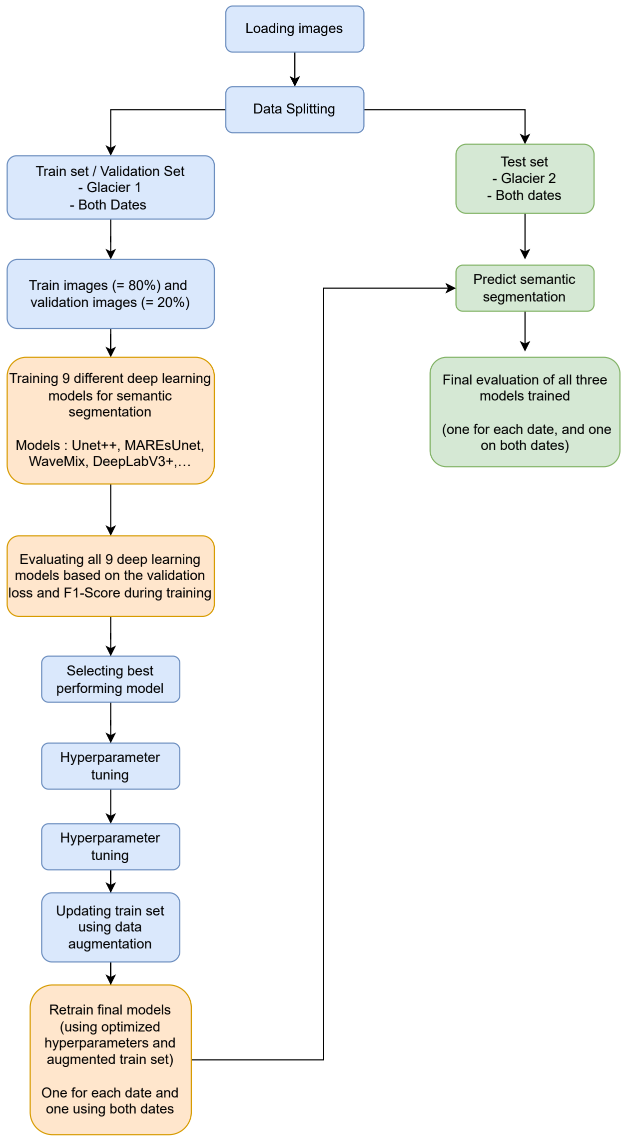

Figure 1: Processing flowchart for semantic segmentation

Figure 1 gives an overview over the methods used in this report.

2.1 Model selection

We trained nine different models on satellite images of the glaciers near Narsarsuaq, Greenland and compared their performance in segmenting the satellite images according to the glaciers. Each of the nine different neural networks is incorporates a distinct network architecture. First, our goal was to find the best performing model and then improve its performance with hyperparameter tuning and data augmentation. The following sections will give you a short overview over all evaluated models and the hyperparameter optimization and data augmentation techniques that we used.

For each model the losses and F1-scores for the train and validation sets were recorded during training. We used the F1 score as a more robust measure to account for the class imbalance between pixels that belong to the glacier and pixels that do not. The model with the highest F1-score on the validation set was then chosen for further analysis.

Below is an overview over all models that were tested:

2.1.1 MAResUNet

The model, multi-stage attention residual UNet, is based on the standard UNet architecture with an additional residual network in the encoding part, allowing the model to avoid the problem of vanishing gradients. Moreover, it employs an attention mechanism [2] which additionally helps the model to focus on specific attributes of the image mocking the performances of a vision transformer (ViT) [3].

2.1.2 UNet ++

UNet++ consists of an encoder and decoder that are connected through a series of nested dense convolutional blocks. It increases the overall UNet performance by capturing the fine-grained details (glacier borders) on high resolution images more effectively, and could thus be well suited for the satellite images at hand [4].

2.1.3 MANet

The Multi-Attention-Network (MANet) uses multiple efficient attention modules to extract contextual dependencies. A new attention mechanism called kernel attention, with linear complexity, is proposed to reduce the computational demand. Through kernel attention and channel attention, local feature maps from ResNeXt-101 are combined with their global dependencies, adaptively reweighting interdependent channel maps and addressing limitations exist in standard convolutional models [5].

2.1.4 WaveMix

The WaveMix model achieves comparable or superior accuracy to state-of-the-art convolutional neural networks, vision transformers, and token mixers across various tasks, specifically setting new benchmarks in segmentation of city scapes. Notably, WaveMix requires fewer parameters compared to the previous state-of-the-art, resulting in time, cost, and energy savings. The key innovation is the use of multi-level 2D discrete wavelet transform in WaveMix blocks, leveraging scale-invariance, shift-invariance, and sparseness of edges to reorganize spatial information without adding parameters, reduce spatial sizes of feature maps, and expand the receptive field more efficiently than convolutions.

2.1.5 LinkNet

The LinkNet model allows bypassing spatial information from the encoder to the corresponding decoder to improve segmentation accuracy. Additionally, since the decoder shares knowledge learned by the encoder at every layer, it can use fewer parameters, resulting in more efficient processing [6].

2.1.6 FPN

The Feature Pyramid Networks (FPN) is particularly efficient for detecting objects at different scales. This ability can be advantageous for accurately detecting parts of the glaciers, involving the extraction of features across multiple spatial scales. This network addresses the challenge of loss in spatial information in deep convolutional networks by utilizing a feature pyramid structure that integrates both bottom-up and top-down pathways, along with lateral connections [7].

2.1.7 PSPNet

The Pyramid Scene Parsing Network (PSPNet) works well with open vocabulary parsing scene, meaning it is able to segment objects without knowing its class during training. Based on the same pyramid pooling architecture, it has several parallel convolution layers that work as an attention mechanism to detect various objects. Moreover, PSPNet offers great results in more detailed image scenes [8].

2.1.8 PAN

The PanNet architecture incorporates domain-specific knowledge, targeting two key aspects of the pan-sharpening problem: spectral and spatial preservation. For spectral preservation, it includes up-sampled multispectral images in the network output, ensuring the direct propagation of spectral information to the reconstructed image. To maintain spatial structure, it trains its network parameters in the high-pass filtering domain instead of the image domain. The network demonstrates strong generalization to images from various satellites without requiring retraining. Experimental results indicate substantial improvements over state-of-the-art methods, both visually and in terms of standard quality metrics [9].

2.1.9 DeepLabV3+

Previous deep learning networks could grasp broad context by using filters or pooling at various rates, while others could define object boundaries more precisely by regaining spatial details. The DeepLabv3+ network merges the strengths of both. It builds on DeepLabv3 by introducing a straightforward but impactful decoder module to enhance segmentation results, particularly around object edges. It also incorporates the Xception model and use depthwise separable convolution in both the Atrous Spatial Pyramid Pooling and decoder modules, creating a faster and more powerful encoder-decoder network [10]. The authors claim that this network architecture also helps the model to understand the context of the different parts of the images better thus being well suited for the glacier task at hand.

2.2 Hyperparameter Tuning

After the initial evaluation of the different network architectures, we wanted to tune the hyperparameters of the model achieving the best performance using grid search. The combination of hyperparameters over which we optimized are listed in the table below:

Dropouts	Activations	Global Poolings
[0.1, 0.2, 0.3, 0.4, 0.5]	[Sigmoid, Softmax]	[Max, Mean]

Table 1: All hyperparameters configurations

2.3 Data augmentation

Furthermore, to make our final model more robust, we decided to perform data augmentation by modifying the images of the dataset and concatenating these with the original dataset to gain a larger train set. For this we used the 'torchvision.transforms' framework. The two processes used to transform the images are:

- a random flip with a 50% chance along the horizontal axis
- color jittering (brightness = 0.25, contrast=0.25, saturation=0.25, hue=0.25).

All process steps were done in the RGB format. The flipping of the images was chosen to introduce variations in object positions and orientations. The color jittering should help the model become more robust to different lighting conditions and color variations that may be present in different glaciers.

3 Results

3.1 Model selection

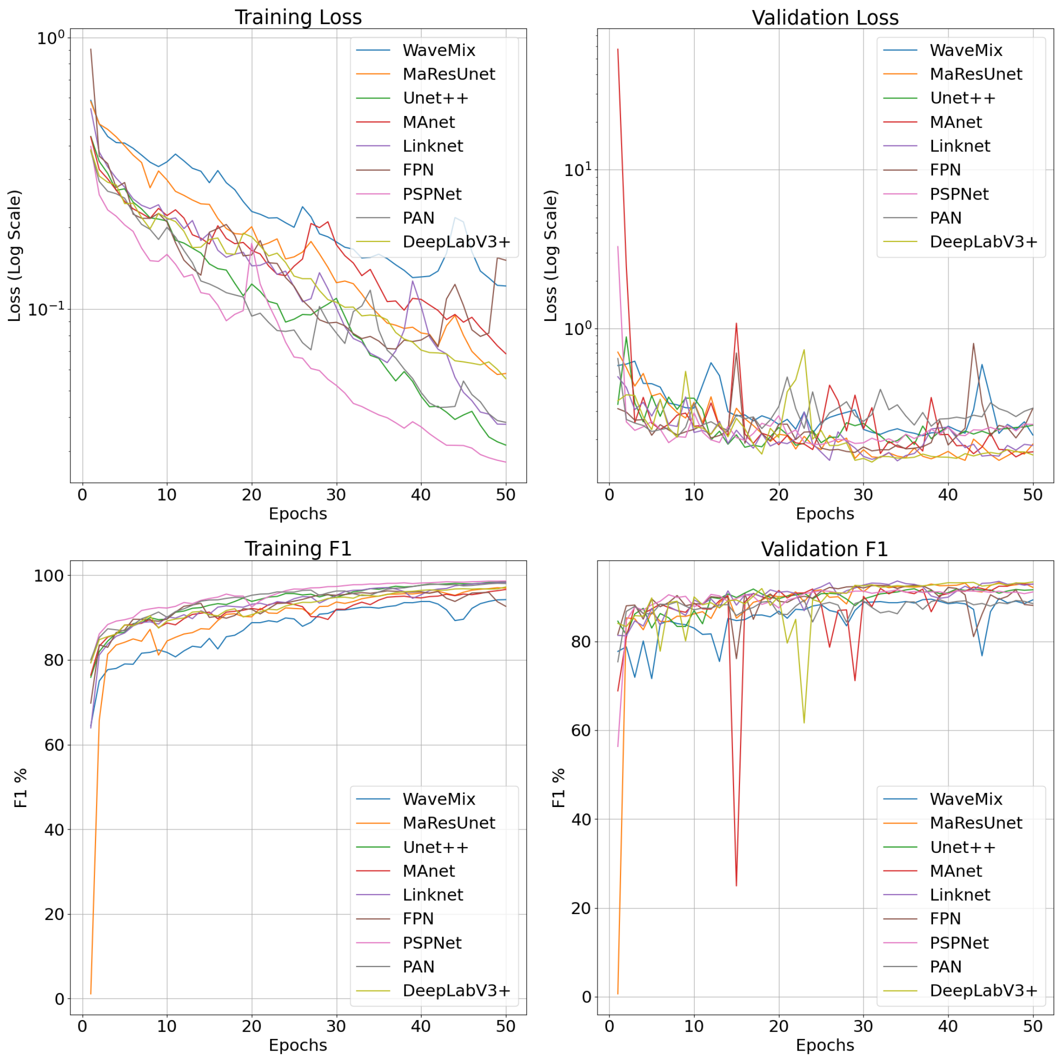

Figure 2: Model comparisons

All models were trained with their default parameters as defined in the segmentation-models-pytorch library. Additionally, we used a uniform dropout rate of 0.4, mean pooling and the softmax activation function, as we found out that these parameters generally provided more stable results during a prior analysis.

As visible in figure 2 all models behave similarly when comparing training and validation loss. This means that all models are comparable in the time they take to converge. When focusing on the validation loss we see that some of the models start to diverge in groups after around 30 epochs. Generally, the MAnet, MaResUnet and DeepLabV3+ achieve lower validation losses compared to the other models.

Some models, such as WaveMix and FPN are much more variable in their F1-scores than other models after 25 epochs such as DeepLabV3+ which shows a more smooth performance in the subsequent epochs. The F1-Score plot shows that the DeepLabV3+ was the best performing model for the last 5 epochs, which is the reason why it was chosen for further analysis.

3.2 Hyperparameter Tuning

During the hyperparameter tuning, we employed grid search and trained the DeepLabv3+ model on every combination over 50 epochs, and selected the hyperparameters that yielded the best F1-Scores on the validation set during the last 5 epochs. The resulting hyperparameters used are visible in Table 2.

Dropout	Activation	Global Pooling
0.5	Sigmoid	Mean

Table 2: Resulting Hyperparameters

3.3 Data augmentation

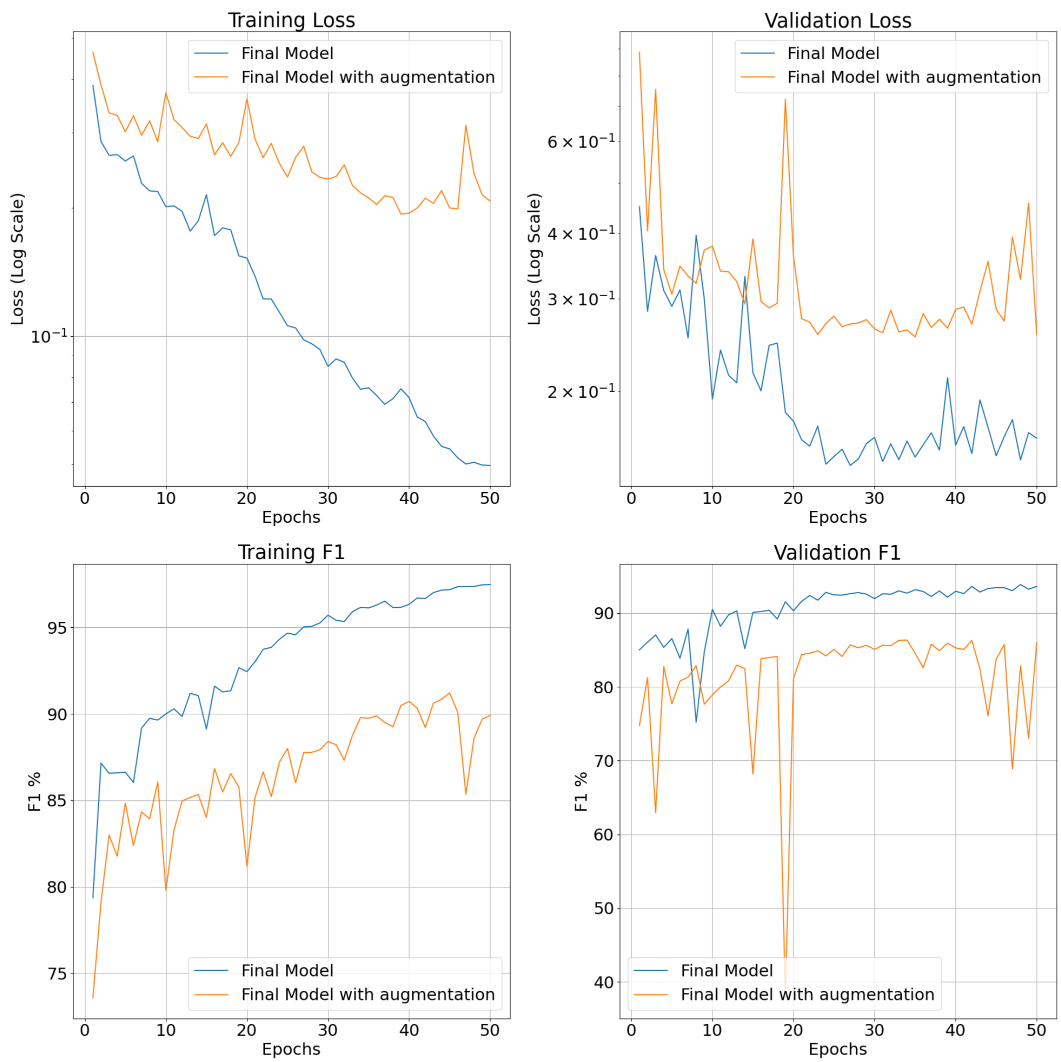

Figure 3: Results of DeepLabV3+ with and without data augmentation

Figure 3 shows the results of data augmentation on the performance of the DeepLabV3+ model. Overall we saw that augmentation did not improve the model's performance but drastically worsened it. It is important to say, that we tried different data augmentation techniques and parameters but the results were similar as shown in the plot above. This was unexpected, as we thought data augmentation would enable our models to be more robust. Thus, in the following, we discarded data augmentation and used the original dataset for the final segmentation prediction.

3.4 Visualization of results

3.4.1 Final models

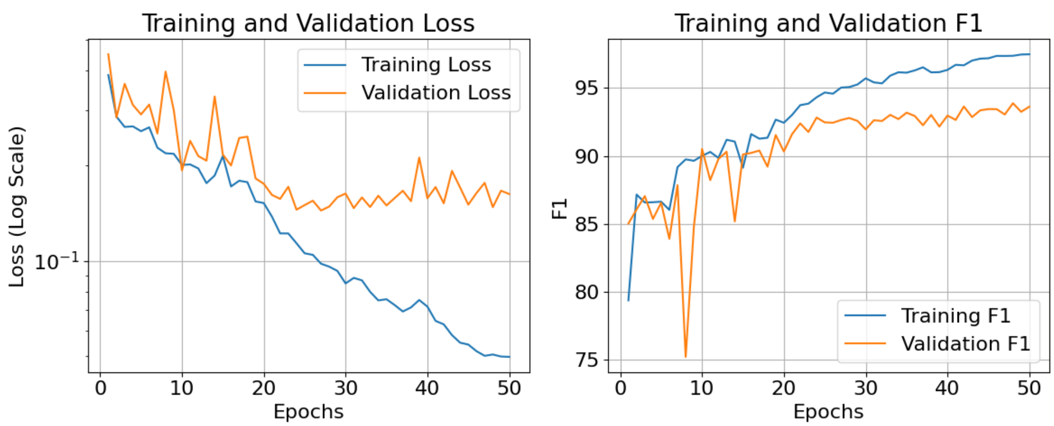

Figure 4: Train history of final model on both dates

The figure above shows how the final model trained on both dates performed during learning. Overall, we see that the validation loss and F1 score does not decrease greatly after 20 epochs and stays constant until the end of the training.

Trained on Date	F1-score on August test set	F1-score on July test set
August	87.42%	79.15%
July	86.82%	81.35%
August and July	88.26%	80.41%

Table 3: Performance on test set

We can see in Table 3 that the final model trained on both dates of the train glacier is able to perform best compared to the ones trained on singular dates. However, the model just trained on the July date was able to perform better on the July test set, compared to the model trained on both dates.

However, as the model trained on both test set is able to perform better taking both dates into account, we will focus on the combined model to show the segmentation results in greater detail below.

3.4.2 Segmentation Results

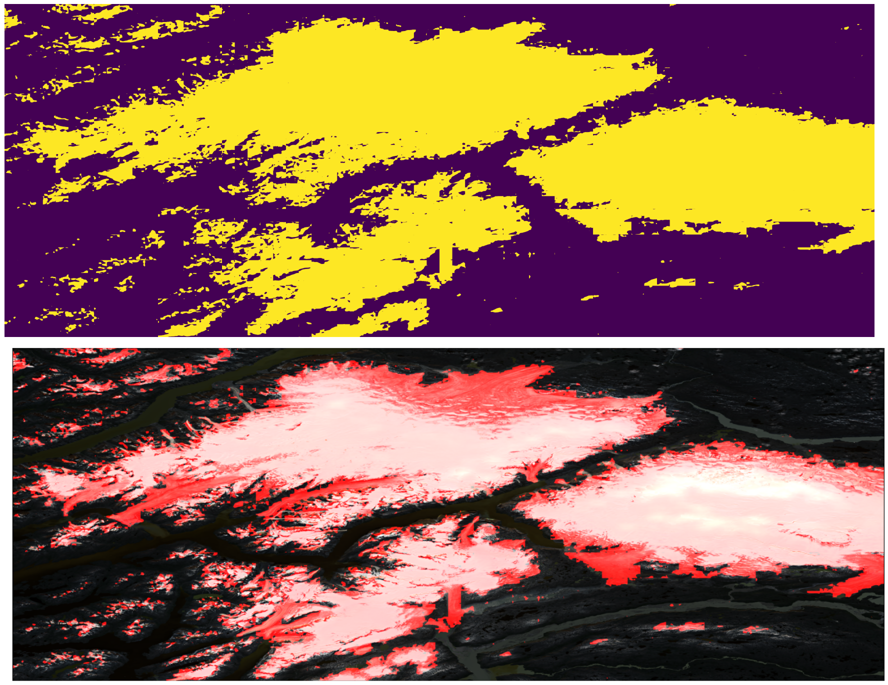

Figure 5: Train glacier prediction labels and overlay - August

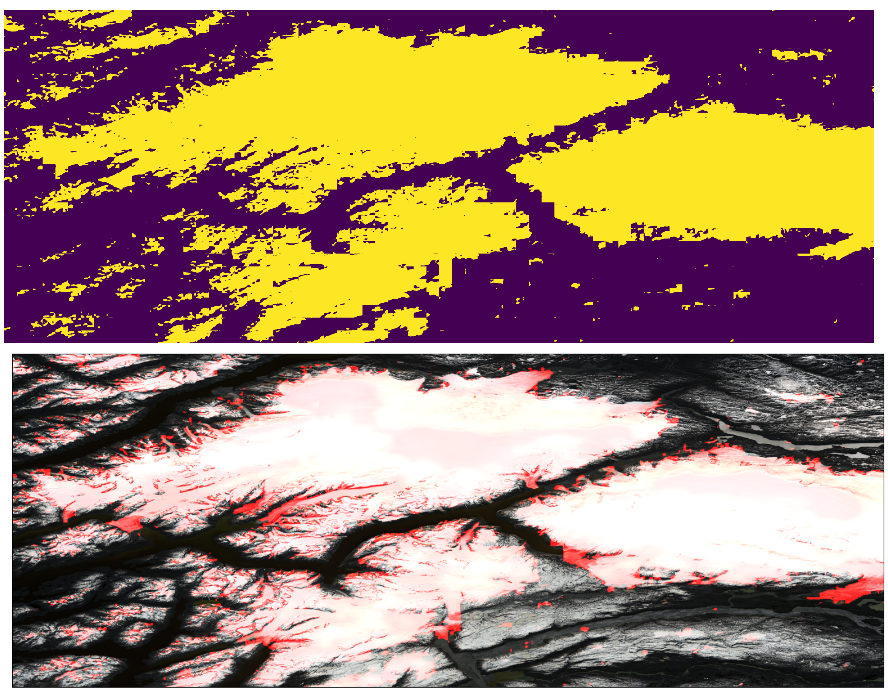

Figure 6: Train glacier prediction labels and overlay - July

The figures above show how the final model predicts the glacier segmentation values, based on the image input. The upper image shows the raw glacier predictions, and the lower one shows the overlay of the prediction in red onto the original satellite images. For the August glacier, we can see that the final model is able to segment the glacier content pretty accurately. However, for the July date, the performance is weak in the upper right and upper left part, where the glacier shows a darker grey tone, which model is unable to predict correctly.

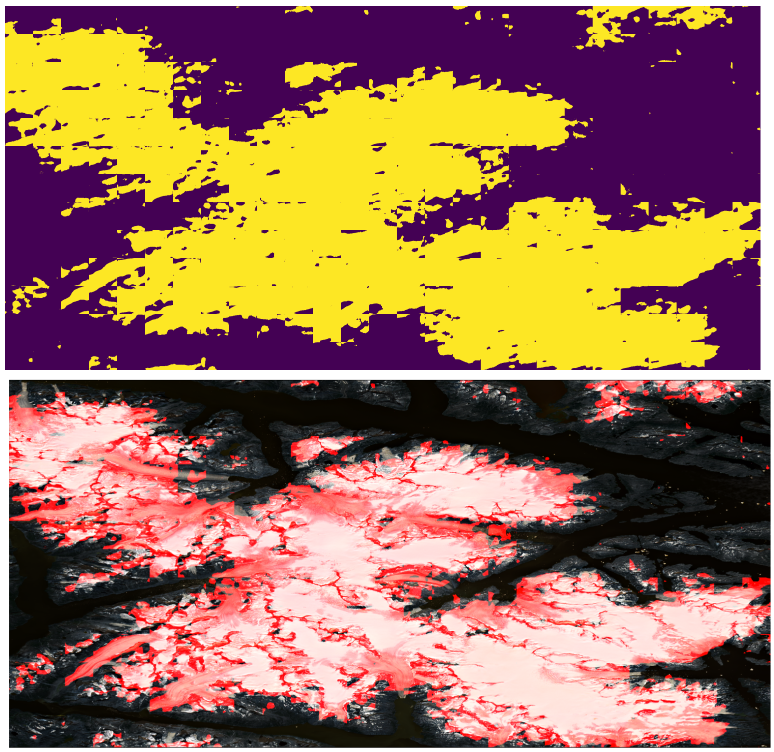

Figure 7: Test glacier prediction labels and overlay - August

Here, we can see the performance on the second test glacier in August, where the model showed a reasonably high F1-Score, which is also reflected in the image above, as the red glacier prediction from the model fits the true glacier in the original satellite image quite well.

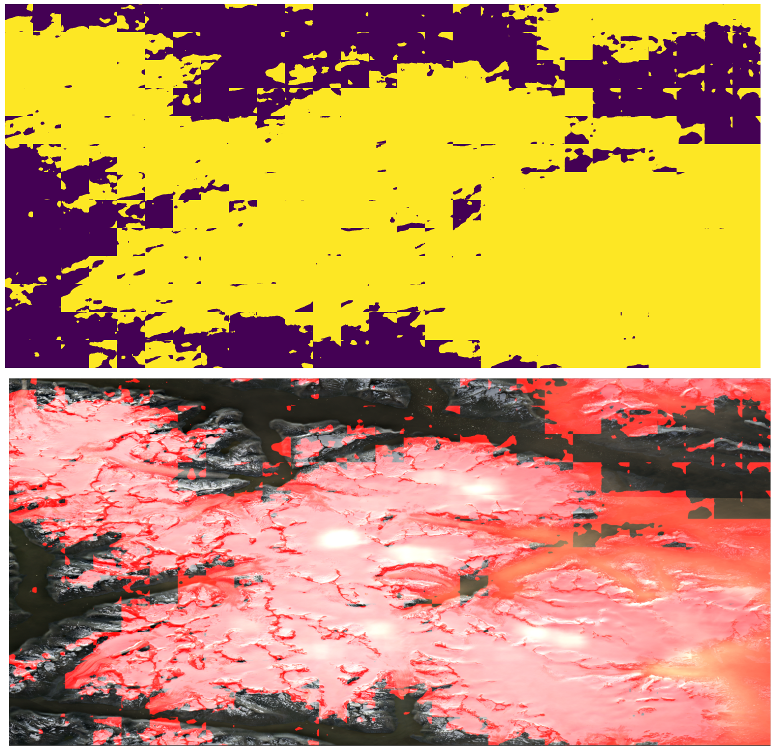

Figure 8: Test glacier prediction labels and overlay - July

Lastly, we can see the final model predictions on the test glacier in July, where the model seems to predict parts of the water in the right side of the image as parts of the glacier. This is also reflected by a lower F1-Score noted above. To summarize, the model seems to have more trouble segmenting the glaciers in July than in August.

4 Discussion

This report tested multiple models on glacier images, and found out that the DeepLabV3+ architecture, performed best given the tested parameters. However, here it is to say that we experienced very fluctuating results among the models, depending on different random seeds runs, which indicates that a more fine grained analysis is necessary to really filter out the best suited models for the task at hand.

We optimized the hyperparameters using grid search and tried to make our model more robust by augmenting the dataset. Unfortunately, due to computational constraints, we were not able to test more than 20 combinations of hyperparameters. A more fine-grained analysis of hyperparameters could have filtered out more suitable model parameters for the glacier segmentation.

Furthermore, the data augmentation led to worse segmentation results for our selected final model. Here, it is to say that we tested multiple data augmentation techniques, like color jittering, random rotations, perspective changes, but every processing technique led to worse results compared to the original dataset. We suspect, that only having training data of one glacier from two different dates, is not enough as a base to perform augmentation on. More satellite images of glaciers would have been necessary for the augmentation to be fruitful.

Finally, our model with optimized hyperparameters trained on both dates of the original dataset performed quite well on the second glacier that makes up the test set. We can see that our model generally performed better on glaciers in August than in July. A reason for this could be that the difference between the glaciers and the surrounding objects were more clear near the end of the summer. This becomes evident, as both glaciers in July are more wide-ranging, with bigger parts in dark-greyish that vanished until August, as these parts most likely melted during the extented summer. The segmentation result plot on the test glacier in July also shows that our model predicted already faded parts of the glacier on the right part of the image to belong to the core of the glacier. The model shows a more accurate and fine-grained prediction in August, as these parts got much darker and nearly faded completely, making it easier to distinct between the core glacier and its fading fragments.

We expected that color jittering, specifically with brightness, would have enhanced our model addressing these pitfalls, but this was unfortunately not the case. Further data augmentation experimentation would be necessary to find the best pre-processing methods enhancing the model further.

5 Conclusion

This project aimed to see if machine learning can be used to track glaciers in the context of climate change, specifically focusing on the glaciers near Narsarsuaq, Greenland. Through the utilization of satellite data from the Sentinel-2 and GLIMS databases, we employed a deep learning approach for semantic segmentation. The evaluation involved training and testing multiple neural network architectures, including MAResUNet, UNet++, MANet, WaveMix, LinkNet, FPN, PSPNet, PAN, and DeepLabV3+. After careful model selection and hyperparameter tuning, the DeepLabV3+ architecture emerged as the optimal choice for our specific task.

We think that the spatial mapping of glaciers and the temporal comparison enabled by deep learning models can provide valuable information for scientific research and environmental monitoring but is not without risks, as the model tested in this report has varying quality of output given different days. Our model could thus allow researchers to scale and lessen the manual work of classifying glaciers, but the output of the model still has to be scrutinized by the researcher themselves.

Future work could explore more sophisticated architectures, add additional environmental variables, and address challenges related to data variability and generalization. More specifically, the use of visual transformers (ViTs) and the use of 3D mappings with altitude measurements could greatly enhance predictions.

References

[1] Valérie Masson-Delmotte, Didier Swingedouw, Amaelle Landais, Marit-Solveig Seidenkrantz, Emilie Gauthier, Vincent Bichet, Charly Massa, Bianca Perren, Vincent Jomelli, Gudfinna Adalgeirsdottir, et al. Greenland climate change: from the past to the future. *Wiley Interdisciplinary Reviews: Climate Change*, 3(5):427–449, 2012.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[3] Rui Li, Jianlin Su, Chenxi Duan, and Shunyi Zheng. Multistage attention resu-net for semantic segmentation of fine-resolution remote sensing images. *CoRR*, abs/2011.14302, 2020.

[4] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. *CoRR*, abs/1807.10165, 2018.

[5] Rui Li, Shunyi Zheng, Ce Zhang, Chenxi Duan, Jianlin Su, Libo Wang, and Peter M. Atkinson. Multiattention network for semantic segmentation of fine-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

[6] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE visual communications and image processing (VCIP)*, pages 1–4. IEEE, 2017.

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[8] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016.

[9] Junfeng Yang, Xueyang Fu, Yuwen Hu, Yue Huang, Xinghao Ding, and John Paisley. Pannet: A deep network architecture for pan-sharpening. In *Proceedings of the IEEE international conference on computer vision*, pages 5449–5457, 2017.

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.