Interdisciplinary Project in Data Science

Water Body Classification
Project Report

Author:
Adam Böröndy

Matriculation Number:
01610133

Supervisors:
Ao.Univ.Prof. Dr. Andreas Rauber
Univ.Ass. Felix David Reuß, MSc

Semester:
Winter Semester 2020

Course Number:
194.047

September 30, 2021

Abstract

This report presents a machine learning workflow for the classification of water bodies in satellite images. Using the U-Net deep learning architecture[5], a pixel-by-pixel classification of the image content is performed. The training of the network is based on a set of Sentinel-1 acquisitions of the same geographical area at different time points and a hand annotated ground truth classification of the area.

Contents

List of Figures

List of Tables

1 Data

1.1 Data Selection

For the project, a large set of Sentinel-1 synthetic aperture radar images was made available. These satellites use specific wavelengths and measure the intensity of backscatter, thus making acquisitions about the Earth's surface. All images in the collection describe the same area in the Netherlands at different times in 2018. In order to match the available data to the task and to optimize the training time of the model, a pre-selection of the images was conducted. The main criteria for the selection were:

- The relative orbit number: While there is a wide variety of available orbit numbers, only certain ones cover larger portions of the area in question. Since the main criterion for temporal coverage in waterbody classification is to include imagery with varying (weather and seasonal) conditions, the training imagery was limited to a single best-fitting orbit number (#88). This setting provided an excellent overview of the entire area with a temporal coverage of 12 days, meeting both the spatial and temporal requirements for the task. A list of the selected images can be found in Appendix A.
- The measured return waves: Depending on the setting, the observations can measure the vertically or horizontally polarized return waves. As highlighted in [3], the vertical setting is preferred for water body classification. Therefore, the selected images were limited to this setting.

Overall, the comparable technical settings for all training images provided a reliable training and testing environment for subsequent model training - and clear guidelines for the theoretical use of the model in production.

1.2 Pre-processing and Patch Generation

As part of the workflow, the selected **satellite images** were further pre-processed before training the deep neural network.

1. In a first step, the values described by the observations were normalized so that their range spans from 0 to 1. For this, both image-wise and cross-image normalization were evaluated. The reason for image-wise normalization is typically a circumstance that causes systematic differences between images (e.g., day and night in common image classification tasks). In the case of this project, a similar systematic difference was expected due to the changing weather conditions between different time steps. However, normalizing each image individually did not result in a significant difference, so I kept the general approach of normalizing the images based on the information from the entire training set.
2. Next, a patch generation step was applied to decrease the size of the input images for the network. As an additional requirement, an overlap between

patches was introduced to ensure that the spatial context of the image features is taken into account. Given the size of the satellite imagery and the input size preferences for the U-Net, a size of 256*256 pixels was chosen for the patches. With an overlap of 24px between the edges of the patches, 43*43 = 1849 patches were available in each image. Figure 1 illustrates the approach by showing the boundaries of some sample patches.

3. Finally, the patches were divided into training, validation, and testing sets. To cover larger contiguous areas, so that testing results would provide visible results for inference, adjacent patches were grouped into larger rectangular areas and used in later steps for validation and testing purposes. This step also addressed one of the major limitations of the data, namely that the non-water class outweighed the water area by a factor of ∼40. Therefore, randomly sampling 10-20% of the patches would have yielded highly volatile validation/testing sets in terms of their class composition. As a solution, the larger validation and test sites were assembled and placed in a way, which ensured a comparable composition (the idea was based on the stratified sampling approach of unbalanced datasets). This was particularly important for the validation set, as it was used not only to monitor training process but also to adjust learning rates in later steps.

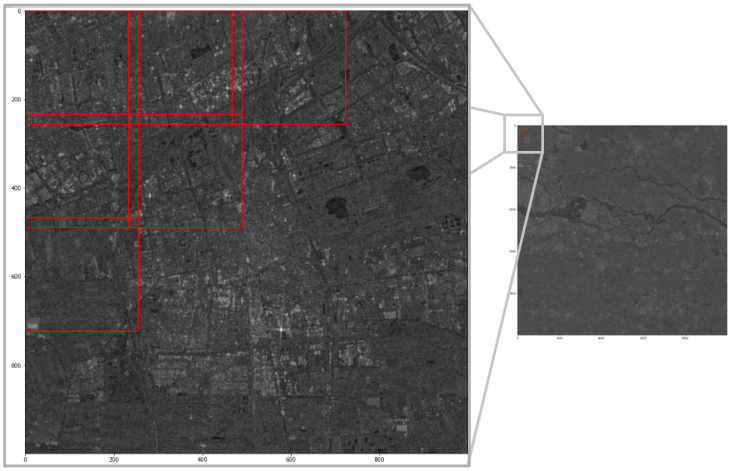

Figure 1: Patch Generation with Overlap

All these steps were integrated into a data loading function. To ensure the efficiency of the approach, single acquisitions were loaded into memory and normalized only once. This way the retrieval of patches (and batches of patches) could happen fluently by extracting the relevant subset from the image. On a side note, this approach came with the limitation of making a shuffled retrieval of the patches, from different acquisitions, inefficient and undesirable.

The **target image** containing the ground truth annotation for the region of interest was also adapted for the use case. There were two different classes

of water body: permanent waters and open ocean. However, in this case, the region of interest appeared to be the intersection of these two classes - with a clearly visible boundary line within the body of water. Therefore, they were not necessarily meaningful to distinguish and were treated as one combined water body class. This was the target class (1), while any other area was simply considered as another class of no interest (0). Figure 2 shows the resulting target image, with the water bodies highlighted by white color. In addition, the blue rectangles show the areas selected for validation and the red ones show the test areas.

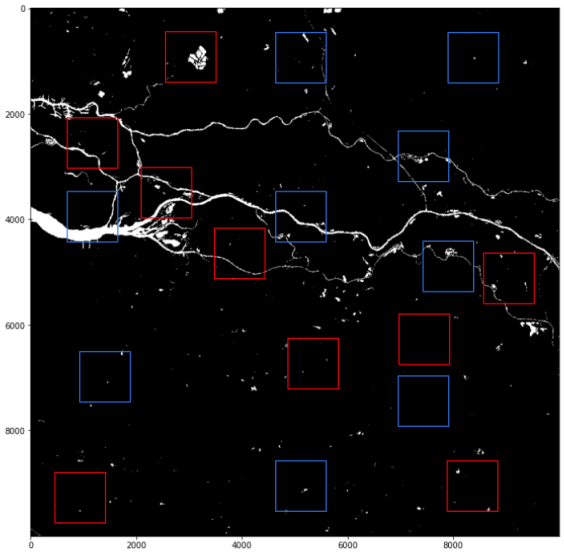

Figure 2: Adjusted Ground Truth Annotation

2 Modeling

For the modeling part, a U-Net architecture was implemented. In this approach, a pixel-by-pixel classification is performed for an (preferably high-resolution) input image. The peculiarity of a U-Net is that it operates symmetrically. A more or less general convolution process on the contracting or encoder side of the net is followed by an expansive or decoder side, which essentially serves as an upsampling, so that we get a prediction or the same dimension as the input. In addition, the two sides are connected at each level by skip connections that ensure that the fine-grained information extracted on the encoder side is passed to the decoder side and used for more accurate segmentation. Figure 3 shows a visual depiction of the U-Net used as guideline for this project. The only modification applied was essentially decreasing the size of the network by excluding the first two levels and beginning with input patches of size 256*256.

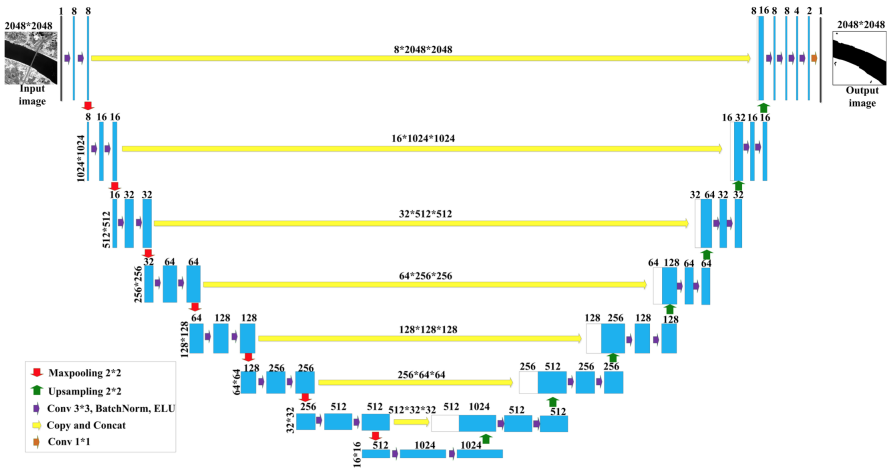

Figure 3: U-Net Structure as Presented in [4]

Apart from the network itself, the two most crucial elements of the modeling were the loss function and the optimizer. For the loss function, the binary cross entropy was chosen, with the small extension of using a version that can be applied directly to the logits provided by the last convolution of the model. This also meant that in order to interpret the predictions as percentages in later steps, first a sigmoid activation had to be applied on them - since it was not part of the network itself.

Furthermore, as mentioned earlier, the ratio of water and non-water pixels presented challenges on several occasions. In order for the loss to provide reasonable results that could be used as an indication of the training progress, the positive class was weighted according to its frequency in the patches of the training set. Instead of taking this average across all training patches, I also tried making the weight depend only on the particular batch. However, in this case the weights changed very quickly, making the loss metric extremely volatile. The reason for this was, that not only are water bodies rare, but when they do show up, they are obviously clustered.

For the optimizer, the Adam optimizer was chosen. Considering the time and computational cost of training the model, I opted for a progressively decreasing learning rate to optimize the model, instead of a classical fine-tuning by iteratively evaluating different settings. The approach was to track changes in the validation performance and decrease the learning rate if it increased for two consecutive epochs compared to the best model. In this case, the learning rate was decreased by a factor of 10 and the weights of the best model were reapplied.

3 Evaluation

Compared to similar research projects, this project took the seemingly rarer evaluation approach of applying and evaluating the model on a completely unseen area, rather than applying it to the same spatial area from which the training data originated, but at new time steps. Therefore, the following performance measures must be interpreted according to the purpose of the model, i.e., the application to new spatial areas, preferably in the vicinity of the training area. The following table shows a summary of the predictive performance achieved by the model on the testing set:

Performance Measure	Minimum	Mean	Median	Maximum
Precision	0.622	0.792	0.793	0.88
Recall	0.04	0.646	0.728	0.884
F1	0.076	0.677	0.765	0.821

Table 1: Summary of Performance Metrics

Apart from these general summary metrics, the performance differences between the images as well as the mispredictions within the images are also of interest. In terms of performance between images, we can see that there are some negative outliers, but no exceptional positive outliers. This becomes more visible, when visualizing the image-wise performance in a boxplot:

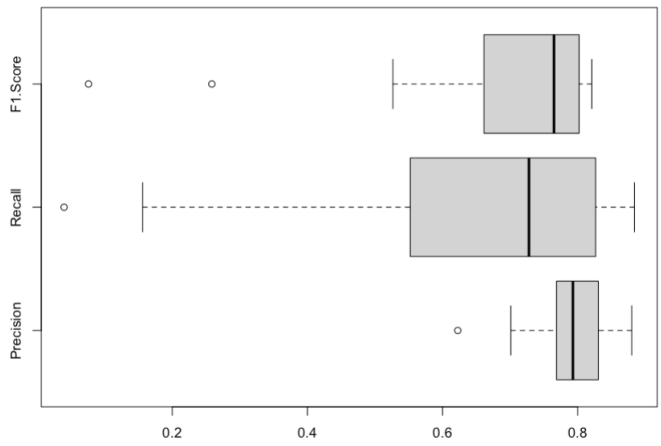

Figure 4: Distribution of Performance Across Acquisitions

In the case of the surprisingly poor predictions, we can see upon inspecting the original input acquisitions that they were made clearly under more challenging conditions (e.g., rain or wind easily affecting the measurements of VV polarization[2]). As an example, Figure 7 compares sample testing patches of the target annotation with the worst and best performing acquisitions:

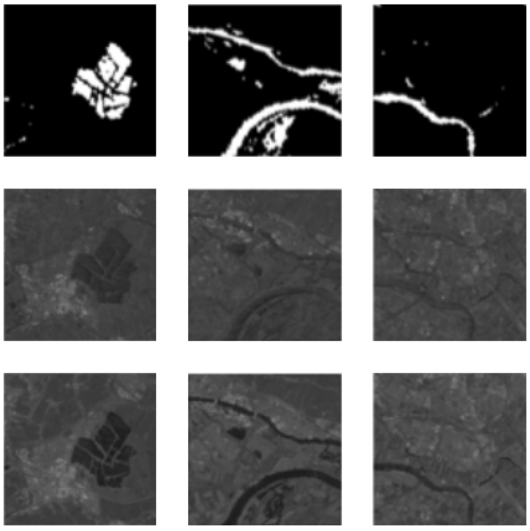

Figure 5: Comparison of Acquisition Quality (Target, Worst and Best Perf.)

On closer inspection, we can see that the worst performing outliers have a high imbalance between accuracy and recall. While their accuracy is high, they miss many of the relevant areas. Therefore, a more sensitive cut-off threshold for these conditions might be beneficial and extract further relevant features. In the specific case of the worst-performing test acquisition I have used a Precision-Recall curve to find a more balanced cut-off threshold. With a threshold of 0.032 the predictions reach an F1-Score of 0.52, which a significant improvement but still far from the mean or median performance of the model. Overall, although on average the threshold of 0.5 performs relatively stable, there are conditions in which a more sensitive setting might provide more useful predictions.

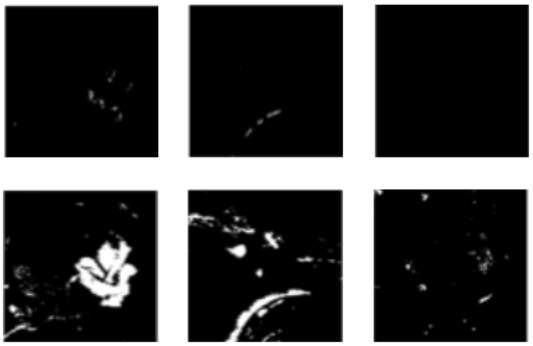

Figure 6: Segmentation Improvement with Adjusted Cut-Off Threshold

As for the misclassification within the patches, Figure 7 visualizes the false positive pixel classifications within a sample testing area. As expected[1], the most critical area was around the border of the water bodies. There are several explanations that could play a role in this pattern. Most importantly, the majority of the edges has a very fine grained shape, which is challenging to predict with

pixel accuracy. In addition, the boundaries are poorly captured by the relatively lower resolution of the ground truth annotation. Furthermore, a change in the water bodies - especially around their borders, where deviations are more likely to happen during the year - is not captured by the single ground truth annotation.

Figure 7: Resolution Comparison (Target, Input, Misclassification)

4 Limitations and Improvement

The amount of data available was more than adequate for the task, and even with the limitations mentioned in the Section 1.1, the training was very time and resource consuming. According to the estimations of the *torch-summary* package, a single iteration of the network required about 1GB of memory. Initially, I relied on Google Colab[1]. However after quickly reaching resource limitations, I have switched to Saturn Cloud[2]. In this case, a training epoch lasted about 20 minutes and used the majority of available resources. Although the inclusion of more training data, especially more samples with lower visibility to waters, may lead to improvements, a reasonable boundary must be drawn between the tradeoff of resource requirements and performance.
Nonetheless, further improvements might be possible at other parts of the workflow. One could experiment with modifications within the modeling process, e.g. by evaluating different tuning approaches. In addition, improvements could be made by post-processing the predictions. For example, it would be possible to smoothen the boundaries of the predicted water bodies and address minor speckle-like mispredictions with noise filtering techniques.
Finally, as already mentioned, the training and the evaluation of the entire approach was based on a single ground truth annotation. Depending on the purpose of the model, one might improve the workflow by constructing and relying on multiple annotations, perhaps also ones with more fine grained resolution. In the case of this project, it was assumed that significant changes in water bodies would be rare, while smaller changes might go unnoticed anyway, given the 20m resolution of the acquisitions.

[1] https://colab.research.google.com
[2] https://saturncloud.io

References

[1] Filsa Bioresita, Anne Puissant, André Stumpf, and Jean-Philippe Malet. A method for automatic and rapid mapping of water surfaces from sentinel-1 imagery. *Remote Sensing*, 10(2):217, 2018.

[2] Han Cao, Hong Zhang, Chao Wang, and Bo Zhang. Operational flood detection using sentinel-1 sar data over large areas. *Water*, 11(4):786, 2019.

[3] Miles A Clement, CG Kilsby, and P Moore. Multi-temporal synthetic aperture radar flood mapping using change detection. *Journal of Flood Risk Management*, 11(2):152–168, 2018.

[4] Wenqing Feng, Haigang Sui, Weiming Huang, Chuan Xu, and Kaiqiang An. Water body extraction from very high-resolution remote sensing imagery using deep u-net and a superpixel-based conditional random field model. *IEEE Geoscience and Remote Sensing Letters*, 16(4):618–622, 2018.

[5] 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.

Appendices

A Acquisitions used

Date	Time	Variable	Mode	Orbit	Software	Grid ID	Tile ID
M20180117	172405	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180210	172405	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180222	172404	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180306	172405	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180318	172405	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180330	172405	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180411	172406	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180423	172406	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180505	172407	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180529	172408	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180716	172413	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180809	172412	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180821	172413	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180902	172413	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180914	172414	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20180926	172414	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20181020	172415	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20181101	172415	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20181113	172414	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20181207	172414	SIG0	VVA	088	A0104	EU010M	E045N021T1
M20181219	172413	SIG0	VVA	088	A0104	EU010M	E045N021T1