Semantic segmentation of historical photographs of the Antarctica peninsula

Dahle, Felix; Tanke, Julian; Wouters, Bert; Lindenbergh, Roderik

DOI
10.5194/isprs-annals-V-2-2022-237-2022

Publication date
2022

Document Version
Final published version

Published in
ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences

SEMANTIC SEGMENTATION OF HISTORICAL PHOTOGRAPHS OF THE ANTARCTICA PENINSULA

Felix Dahle* [1], Julian Tanke[2], Bert Wouters[1,3], and Roderik Lindenbergh[1]

[1] Dept. of Geoscience & Remote Sensing, Delft University of Technology, the Netherlands - F.Dahle@tudelft.nl
[2] Dept. of Information Systems and Artificial Intelligence, University of Bonn - tanke@iai.uni-bonn.de
[3] Institute for Marine and Atmospheric Research Utrecht, Department of Physics, Utrecht University, Utrecht, The Netherlands

Commission II, WG II/6

KEY WORDS: Semantic Segmentation, Historic images, Cryospheric Images, U-Net, Machine Learning, Antarctica

ABSTRACT:

A huge archive of historical images of the Antarctica taken by the US Navy between 1940 and 2000 is publicly available. These images have not yet been used for large-scale computer-driven analysis as they were captured with analog cameras. They were only later digitized and contain no semantic information. Most modern deep-learning based semantic segmentation algorithms are trained on modern images and fail on these scanned historical images, due to varying image quality, lack of color information, and most crucially, due to artifacts in both imaging as well as scanning (e.g. Newton's rings). The analysis of such historic data can give a view on Antarctica's glaciers predating modern satellite imagery and provide a unique insight into the long-term impact of changing climate conditions with essential validation data for climate modelling. An important first step for analysis of such data is the extraction and localization of semantic information, e.g. where in the image is water, rocks, or snow. In this work we present the first deep-learning based method to perform semantic segmentation on historical imagery archives of the Antarctic Peninsula. Our results show that our method can handle very challenging images even after being trained with only a low number of training data and catch the general semantic meaning of a scene. For eight test images we achieve an accuracy of 74%, where the majority of errors can be explained by the classification of ice as snow.

1. INTRODUCTION

In the 1940s, the U.S. Navy began flight campaigns over Antarctica to take images from airplanes. Until 2000, over 330.000 images were taken and stored in the so called TMA (trimetrogon aerial)-archive, with the majority of the images from the 1960s. These images give valuable insights into the state of Antarctica before the satellite era. However, these images are only available as scanned image files and do not contain any semantic information, which makes it difficult to use them for research purposes.

To support the use of this image archive, semantic information can be added to the images by semantic segmentation. Instead of looking at all pictures file by file, researchers could then filter for images with certain classes (for example searching only for images containing water or excluding all images with cloud coverage over a certain percentage). Even though there are many algorithms available for semantic segmentation of imagery, these algorithms fail for the available imagery due to the following reasons:

1. Poor image quality: Many of these images have flaws, for example low contrast, over/under exposure or Newton's rings from scanning. See Lu et al. (2013) for an explanation of this phenomenon.

2. Limited spectral information: The images are only available in grayscale with a limited number of information (256 possible pixel values).

3. Different image angles: The images are not only shot perpendicular to the ground, but are also oblique. Furthermore, some images are upside down with the sky at the bottom.

4. Difficult semantic classes: The combination of classes (snow/clouds: white pixels or rocks/water: black pixels) is difficult in a spectral sense.

Even though each of these individual segmentation problems have already been tackled and solved by researchers, the interaction of all four problems at the same time makes this very challenging. In this paper we present a workflow for semantic segmentation to tackle all four problems simultaneously. Our contributions are therefore three-fold: First, we create labelled training data for the segmentation of historical cryospheric imagery. Second, we develop an easy-to-use workflow for semantic segmentation of historical imagery. Last, we provide and train a U-net which successfully segments the images under challenging conditions. Our implementation and labeled data will me made available after sufficient testing[1].

2. RELATED RESEARCH

There are numerous examples of successful image segmentation algorithms and tools, starting with simple mathematical operations to the latest trends of using deep learning based U-nets, compare Hao et al. (2020) or Minaee et al. (2021). This section will put focus on semantic segmentation especially of cryospheric imagery as well as for historical imagery.

* Corresponding Author

[1] Please see `https://github.com/fdahle/antarctic_segmentation`

As the cryospheric regions (mainly the Arctic, Antarctica and Greenland) are located at remote areas that are difficult to reach, semantic segmentation is usually applied on satellite imagery and less on aerial imagery. Examples can be found at Hartmann et al. (2021), Holzmann et al. (2021), Dirscherl et al. (2021) or Mohajerani et al. (2019). All of them are using machine learning based approaches for image segmentation, however always with the goal of finding a certain class (and therefore only applying binary segmentation with two classes).

An good example for semantic segmentation of the cryosphere with multiple classes can be found at Marochov et al. (2021), in which CNNs are used for automated classification of Sentinel-2 satellite imagery in different classes. Due to the combination of RGB and near infrared data they can achieve an accuracy of over 90%. This approach can also be used for non satellite imagery, as described by Dowden et al. (2021). RGB imagery, captured with a Go-Pro attached to a ship, is used to classify the sea-ice in front of the ship. Using a U-Net architecture with transfer-learning an accuracy of 97.7% was achieved.

In comparison to cryospheric imagery, semantic segmentation for historical imagery is more rare for several reasons: It is more difficult to collect enough labelled data for a successful training of a segmentation model. Furthermore, historical imagery is often only available in grayscale and with less contrast and contains therefore less information that can be used for the segmentation. However, some examples for semantic segmentation of historical imagery can be found at Mboga et al. (2020) and Dias et al. (2020).

In the first paper, a U-Net was used for semantic segmentation as well as for classification of historical aerial photographs in central Africa with a high accuracy of over 90%. In the second paper an approach was described using a W-Net to simultaneously segment and colorize grayscale aerial imagery of Potsdam with a high accuracy of around 90% as well. Both papers however worked with images that depicted a very diverse environment with very distinct classes.

3. INPUT DATA

As input data we utilize aerial images from the TMA archive by the USGS (2018), a vast archive of historical imagery of Antarctica collected by the U.S. Navy between 1946 and 2000. TMA photography is a camera system that collects a left-oblique, on-nadir (straight down), and right-oblique image, as shown by Figure 1. These photographs were used for early topographic mapping and provide an historical snapshot of many parts of Antarctica. This archive was digitally scanned by the USGS with a resolution of 25 micron/1000dpi and is publicly available at their website[2]. Note that these images are not georeferenced. This archive has not yet been used for semantic segmentation but only for small scale point cloud reconstructions, as done by Child et al. (2020) or Kunz et al. (2012).

For semantic segmentation, the images should be segmented in one of the classes described in table 1 (A legend of the classes can be found in figure 6). Random images from the archive are used as training data. As can be seen in the second column of the table, the class distribution of this data (but also of the whole archive) is very imbalanced. The last class unknown is only

[2] see `https://www.pgc.umn.edu/data/aerial/` for a description and `https://data.pgc.umn.edu/aerial/usgs/tma/photos/` for the images

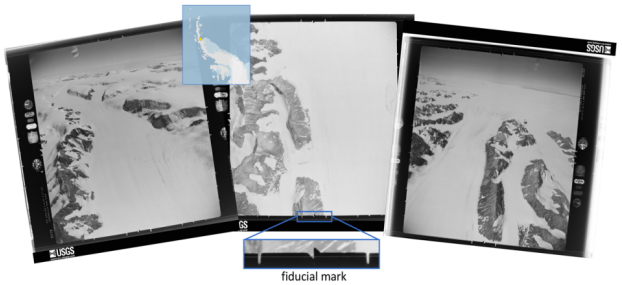

Figure 1. Example image triple from Antarctic TMA, consisting of a left-oblique, an on-nadir and a right-oblique image.

prevalent in the training data and is not a valid output class. It is used as a class, if the pixels cannot be assigned with a certain confidence to another class.

Class	Percentage
Ice	5
Snow	69
Rocks	2
Water	7
Clouds	8
Sky	6
Unknown	2

Table 1. Class distribution of all training images used to train the model

These images are only available in grayscale with 8-bit (=256 different shades of gray) resolution. Often, the images cannot provide a good contextual meaning, e.g. it is difficult to distinguish between different classes: snow/clouds (both white) or rocks/water (both black).

4. METHODOLOGY

Figure 2 depicts the workflow of this method in a flow chart. As a first step (4.1) initial training data must be generated. Afterwards, the actual training of the model (4.3) can begin with a U-net based model (4.2). It is followed by a post-processing step to improve the quality of prediction, (4.4). In the last step, the model is evaluated (4.5).

All steps in this workflow are applied in an iterative process, in which training data is gathered and a model is trained with this data. This model is evaluated with some unseen images and if necessary trained again (because the classification failed) with more training images.

4.1 Pre-processing

As a first pre-processing step, the prevalent borders in the images (as can be seen in figure 1) must be removed. These borders do not contain any semantic information for the scenes and will only limit the efficiency of the model. All images contain fiducial marks that describe the limits of the borders. Using the free library of dlib described in King (2009), these fid-points can automatically be recognized and used to separate the inner part of the images from the borders.

Having the raw images without borders, the next step is to create the training data manually. To support this creation of training data, an unsupervised neural network for image segmentation was applied on the raw images. This will create initial segments of the images, based on matching parts of the images.

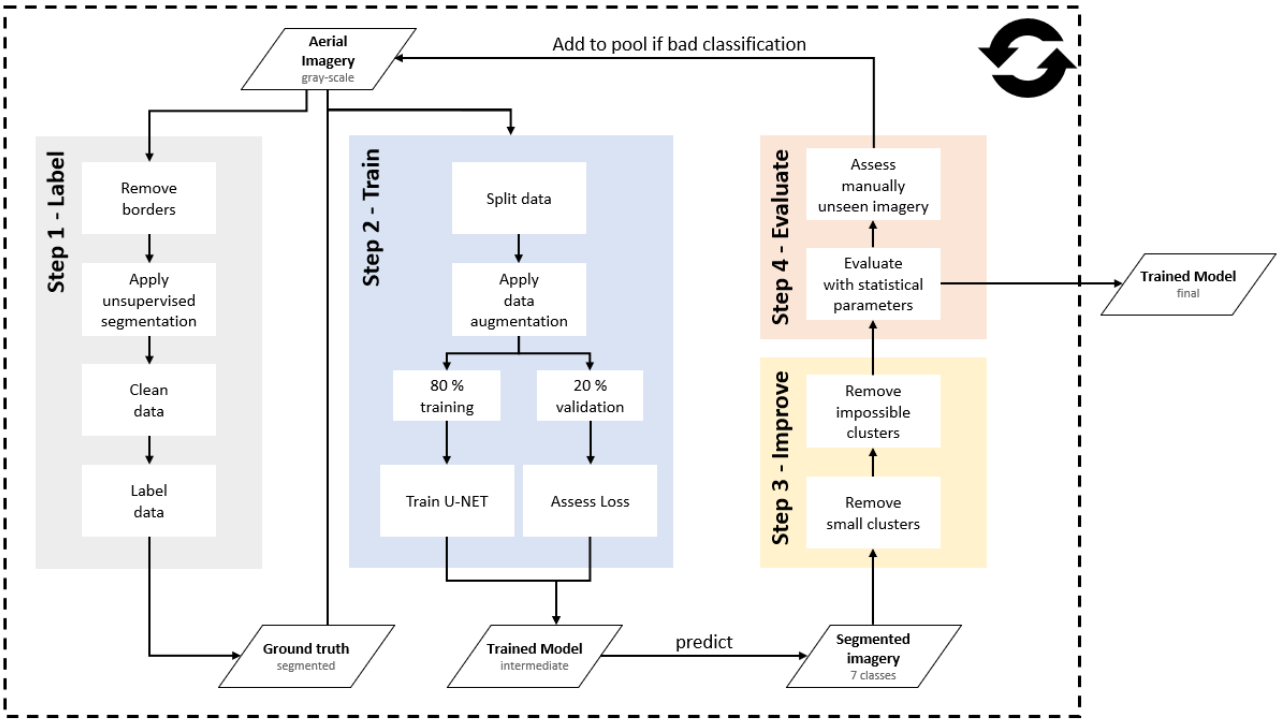

Figure 2. Flow chart of the processing chain from aerial imagery to trained model. All steps inside the dashed rectangle are applied in an iterative process.

In this paper the end-to-end network of unsupervised image segmentation from Kanezaki (2018) was used. This method works with the following criteria:

1. Pixels of similar features should be assigned the same label
2. Spatially continuous pixels should be assigned the same label
3. The number of unique labels should be small

In order to use the unsupervised segmented images for training, they must be further processed. The created segments do have errors and do therefore not match the images perfectly. Furthermore, these segments do only have consecutive numbers as labels and contain no semantic information. The following steps were applied:

1. Renumbering segments: The segments created by the unsupervised segmentation could consist of multiple, non connected parts. These parts will be assigned with a new number, so that every unique segment has its own number as label.
2. Removing small segments: Small segments under a certain threshold size of 20 pixels will be removed.
3. Filling the removed segments: The removed segments from the previous step will be filled with their surrounding pixels (by using the watershed algorithm).
4. Separating segments: Sometimes only one segment is created, where in reality two different classes are present (See the left images in figure 3: The classes sky and snow at the top are wrongfully merged in one big segment). These segments will be separated manually.

To make the labelling as easy and quick as possible, a tool was written in Python that allows to segment images with the previously described steps[3]. Figure 3 shows some examples of self-labelled training data.

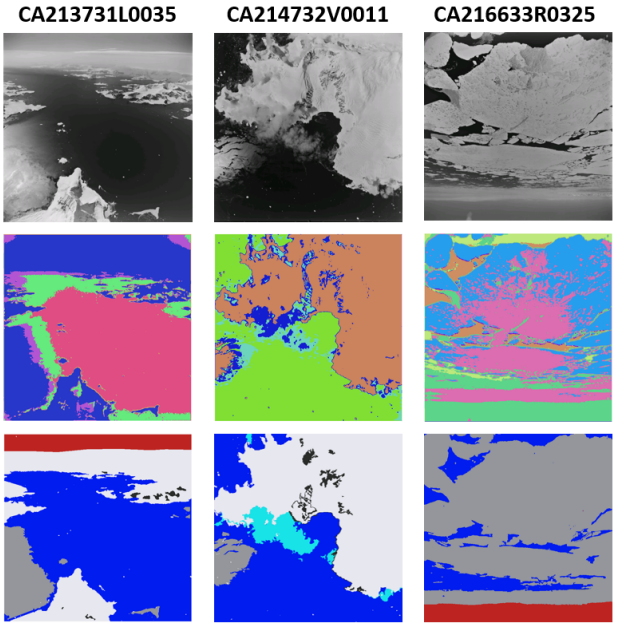

Figure 3. Examples of self-labelled images with raw images at the top, the unsupervised segmentation in the middle (colours are assigned randomly) and the final ground truth at the bottom.

[3] This tool enables for example relabelling of already labelled images together with adapting segments (e.g. separating)

4.2 Architecture of the U-net

In order to apply image segmentation on these images, a classifier was developed that uses both pixel values and spatial context for classification. The basic structure of this algorithm is based on the U-net from Ronneberger et al. (2015), originally developed for bio-medical purposes. This neural network is based on fully convolutional networks and works especially good for precise segmentation with few training images like in Xu et al. (2020), as equivalent for this scenario. It consists of a contracting path followed by an expansive path, resulting in a U-shaped network structure.

Figure 4 visualizes the U-net used in this paper. It is based on Kattenborn et al. (2019) and contains four encoders (contracting path) and four decoders (expansive path). Small modifications were applied to their approach: (1) An additional encoding/decoding layer was added (2) Different number of input layers and output layers were used (3) A bigger input-size was used for the images.

Each encoder/decoder in the network is built with the same components sharing the same attributes:

- Conv2D: A convolutional layer with a kernel-size of 3, that convolutes to additional feature maps. After convolution a stride of 1 is added to maintain image size.
- BatchNorm: The input batch is normalized by re-centering and -scaling to make the network more stable and converge faster, as described by Ioffe and Szegedy (2015).
- ReLU: The activation function used in this network. Short for rectified linear units.
- MaxPool2D: Downsizing of the image in order to reduce the computational cost. The kernel size is 3 with a stride of 2 which results in halving the image.
- ConvTranspose2D: A transposed convolutional layer. The kernel size is 3 with a stride of 2 so that the output image size is doubled in comparison to the input image size.

With each encoding block, the image size is halved, whereas the number of feature maps is doubled (except for the first encoder/decoder). This allows to reduce the image size and therefore reduces the computational cost. With each decoding block, the image size is doubled, but the number of feature maps is halved. Note that for decoder 1-3, the number of input feature maps is doubled as the output of the previous decoder is concatenated with the output of the respective encoder.

After all encoders/decoders are applied, the output image size equals the input image size, and consists of 6 channels, one per class, each containing each containing a value describing how likely it is for each pixel to belong to that particular class. To create the final segmented image, sigmoid is applied on the data and for each pixel the class with the highest probability is selected.

Instead of using dropout layers against overfitting, as done in Baumhoer et al. (2019), batch normalization is used in every encoder/decoder. As described in Garbin et al. (2020), this improves the training speed while also resulting in a better generalization.

In this work we resize the images from different sizes around 10000×10000 to 1200×1200. We found this to be the best compromise between speed and quality of the model. In total, this network contains around two million trainable parameters. With this structure, the receptive field for one pixel is 155×155.

4.3 Training

To train the model, the images were resized to 1200×1200 pixels and fed to the network with one input channel. The resizing of the images was necessary, as original-sized images did not fit the GPU memory.

In total, a number of 67 images were used, split in training and validation data with respective 80% and 20% of the images. As the number of classes in the images is very imbalanced, random split could not be used, instead stratified sampling was applied, inspired by Yuan et al. (2018): This ensures a equal distribution of the classes for all different sets.

During training, augmentation is applied to artificially increase the number of images and make the model generalize better on unseen data. We randomly augment our data with the following operations:

- Flipping: The images are randomly (p=0.5) flipped vertical and/or horizontal
- Rotation: The images are randomly (p=0.5) rotated around a random degree
- Brightness: The pixels in the images are randomly (p=0.5) increased or decreased by a random number (from 1 to 10)
- Noise: Gaussian noise is added randomly (p=0.5) to the images

For the final iteration, 67 images are used, split in 52 training images and 15 validation image, Each model is trained for a maximum of 10000 epochs. For the final iteration, after around 6000 epochs and 78h training time the model converges and training was stopped in order to prevent overfitting.

We utilize the Adam optimizer with an initial learning rate of 0.001. The models were trained on a NVIDIA Tesla P100 with 16GB of video memory. As this image segmentation is a multi-class classification problem with imbalanced classes, the average weighted cross-entropy as described for example in Phan and Yamamoto (2020) is used as loss[4]:

After a training round, the model was applied on unknown imagery and evaluated visually. If big discrepancies could be seen between the raw image and the predicted outcome, this respective image was manually labeled and therefore added to the pool of training images. Afterwards, the model was trained again, using the enlarged pool of training/validation images. This approach allowed to start with a lower number of training images, train with more challenging images and quickly increase the overall quality of the models. For the first iteration, the data set started with 14 segmented images and grew to a final size of 67 images after seven iterations.

[4] Note that the weight for the class unknown is always zero, as learning this class should be prevented

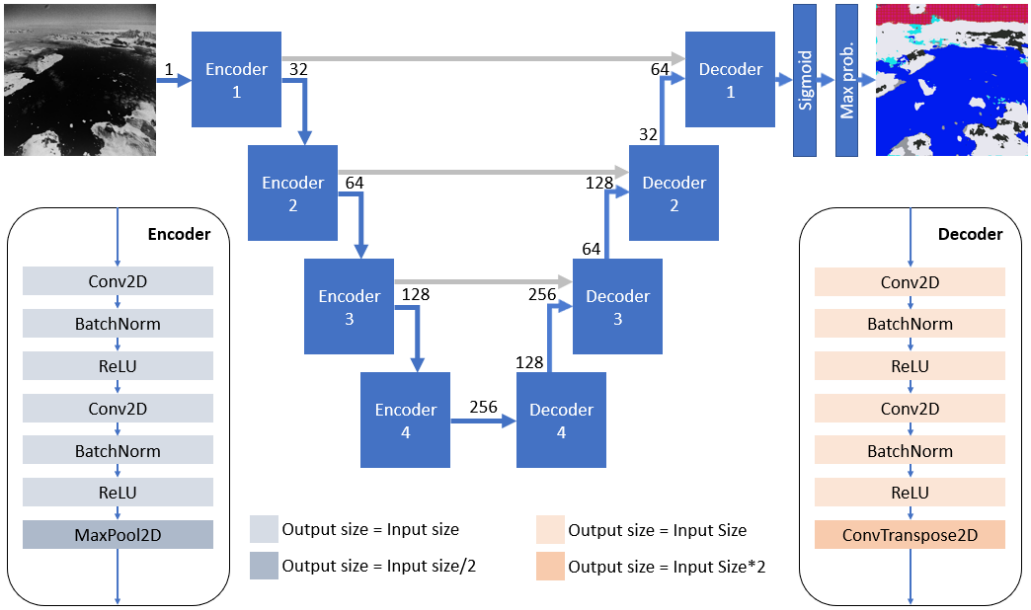

Figure 4. Structure of the U-Net

4.4 Post-Processing

With post-processing the quality of the image segmentation can be further improved.

- Some images are made with cameras facing down vertically and are just depicting the ground. For these images it is impossible to have the class sky. If this class is appearing in these images, it will be replaced with the class with the second-highest probability.
- Some combinations of classes are physically not possible and can therefore easily be recognized and removed. Examples would be small patches of the class sky that are located far away from the real sky or small patches of the class snow inside the sky. These patches will be removed and filled with the value of the surrounding pixels via a watershed-algorithm.
- The probability scores for the different classes are very close together, often in the range of 0.001% and both classes are therefore very likely. For these patches it was checked if this patch is under a certain size-limit and if there were neighbouring patches prevalent that contain the same class. If this was not the case, this patch was considered as an outlier and the class was changed to the most prevalent surrounding class. This was often the case for small patches of the classes clouds and water that were converted to the class snow.

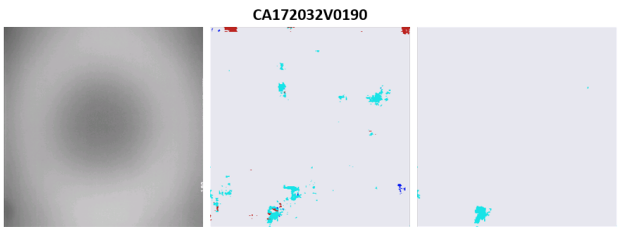

Figure 5. Example for post-processing with raw image (left) and prediction before (center) and after (right) post-processing

4.5 Evaluation

The evaluation of this model poses some challenges:

- As the labelling of the images is very time-consuming, only a limited number of labelled images are available, from which most are required for training and validation of the model.
- The results of the model are very dependent on the images themselves, as the images are not equally easy to segment. Some images have easy recognizable segments and are classified with almost no error, whereas other images (especially the underexposed) are more challenging
- The ground truth data was also created manually and the quality of this data is therefore dependent on the person creating it. Especially small structures are often not classified in the ground truth data, but recognized by the model. Even though the model is in this case correct, it will be classified as a bad result.

The evaluation of this model is therefore applied with a small selection of imagery from the same TMA archive but far away from the locations of the training/validation images. The images were selected randomly with two constraints: (1) All classes should be available in order to check the performance of the model in a complete manner; (2) the images should be difficult to segment in order to challenge the model (a picture of good image quality just containing the class snow for example is not a challenge).

The evaluation metrics for this model are based on the standard evaluation parameters for classification: accuracy, precision, recall and f1-score as for example described in (Diego Oliva, 2021, p. 450).

5. RESULTS

Figure 6 shows the segmentation of the model for these different images. It can be seen that for every picture the general

semantic meaning is captured; it is therefore clear if the picture is shot perpendicular / oblique or if it contains parts of the ocean. Even for difficult parts the segmentation is working: The shadowy part (second column) is correctly identified as snow and the newton's rings (third column) have no influence either. The biggest difference between ground truth and result can be seen in the first and the fifth column: Ice is almost exclusively switched for snow, only in the first image some parts of the ice are classified correctly.

Table 2 gives the performance of the model on selected test images. None of these images was used either for training or validation. Precision and recall are averaged over all the classes.

Image	Accuracy	Recall	Precision	F1-Score
CA035131L0077	0.55	0.92	0.55	0.55
CA139132V0154	0.96	0.96	0.96	0.96
CA172032V0190	0.99	0.98	0.99	0.98
CA172733R0183	0.93	0.95	0.93	0.94
CA179231L0038	0.25	0.25	0.62	0.16
CA180031L0060	0.96	0.95	0.96	0.96
CA135432V0377	0.56	0.87	0.56	0.68
average	0.74	0.84	0.80	0.75

Table 2. Results of model per image

Table 3 shows the performance of the model per class.

Class	Precision	Recall	F1-Score
Ice	0.02	0.74	0.04
Snow	0.98	0.71	0.82
Rocks	0.43	0.65	0.52
Water	0.80	0.94	0.86
Clouds	0.45	0.80	0.58
Sky	0.92	0.81	0.86

Table 3. Results for model per class

Table 4 shows a confusion matrix for the selected test-images. Note that the class unknown is not a valid class for output, so that there is no column for this class.

	Ice	Snow	Rocks	Water	Clouds	Sky	Total
Ice	**35919**	1560885	3669	19565	273	23277	1643588
Snow	9354	**5482627**	15205	10344	7537	80337	5605404
Rocks	895	62440	**51683**	4123	0	0	119141
Water	2067	179428	7757	**1027328**	69619	0	1286199
Clouds	0	364437	136	31470	**325789**	0	721832
Sky	0	42314	0	370	0	**553655**	596339
Unknown	521	68660	1475	5254	2579	29008	107497
Total	48756	7760791	79925	1098454	405797	686277	10080000

Table 4. Confusion matrix for the test-images with real classes on the left and predicted classes on the top

6. DISCUSSION

The average accuracy of this model on the test set is 74% over 6 classes. Most errors can be attributed to the model misclassified ice as snow, as can be seen in table 4. However, when looking at the matrix for the other classes, the majority of classifications is correct.

The model is able to distinguish between different classes and catches the general semantic meaning of a scene. Following annotations can be noted:

1. High percentage of snow for the pixels
   For almost any pixels of the images, regardless of the ground truth, there is a high probability for the snow-class. This is caused by the labelling of the data, as the labels can only be created with confidence for the bigger structures. Labelling correctly every pixel would take an unjustified amount of time, therefore small structures will rise up in the bigger segments, which is usually the class snow.

2. Ice gets confused with snow
   It is difficult for the model to distinguish the class ice from the class snow. Often these classes are close together and have a very similar semantic structure. This effect is reinforced by snow often covering the ice fields. For these images, only visual information is not enough for a correct segmentation and additional information must be included for a successful segmentation. However, for most applications this is not a major issue.

3. Segmentation of clouds failing for some cases
   As can be seen in the training results, a complete segmentation of the class clouds is difficult. Dependent on the thickness of the clouds, the land beneath the cloud can be completely obscured or only partly covered, which can lead the model to wrong assumptions. Furthermore, due to its similarity in the gray colour space (both are white), clouds often get confused with snow (the training data for snow is substantially larger, which makes for the model generally snow more likely than clouds).

7. CONCLUSION

In this paper we successfully created an initial workflow for semantic segmentation of historical cryospheric images with a U-net based model that could extract semantic information for the images of the TMA-archive. The created model learns to distinguish between most of the different classes with a certain confidence and does not get disturbed by flaws in the quality of the images. As can be seen for example in figure 7 (the example images from figure 1), it is able to catch the general semantic meaning of an image, especially after the post-processing. However, for some class combinations and some finer details (for example small rock structures) the model must yet be improved.

As a next step, more training data will be created and added to the model, which is expected to give the biggest improvement. Having more training data - especially from classes beside snow - will increase the variability of each class and prepare the model for more challenging scenes. This new training data will include both labelled images from the TMA archive and from other historical archives in order to increase the variability of classes even more. It could furthermore be useful to refine the currently very generic classes, for example to distinguish between sea-ice and land-ice from glaciers. As the amount of GPU-memory is limited, a resizing of the images is currently necessary. However, this reduces the amount of information in the images and removes finer texture details from the images. It will be checked if other methods (like training for example with crops) can be used instead and improve the differentiation between snow and ice.

In its current state, this model works with the grayscale images as the only source of information. It should be checked if the quality of the model can be further improved by including metadata of the images as additional data sources. This could be either the information if an image is taken with a oblique or nadir camera, the flight height or even the date when the picture was taken (to account for seasonal effects). However, these additional sources should be selected carefully, as this would limit to use this model for other images where this additional metadata is not available.

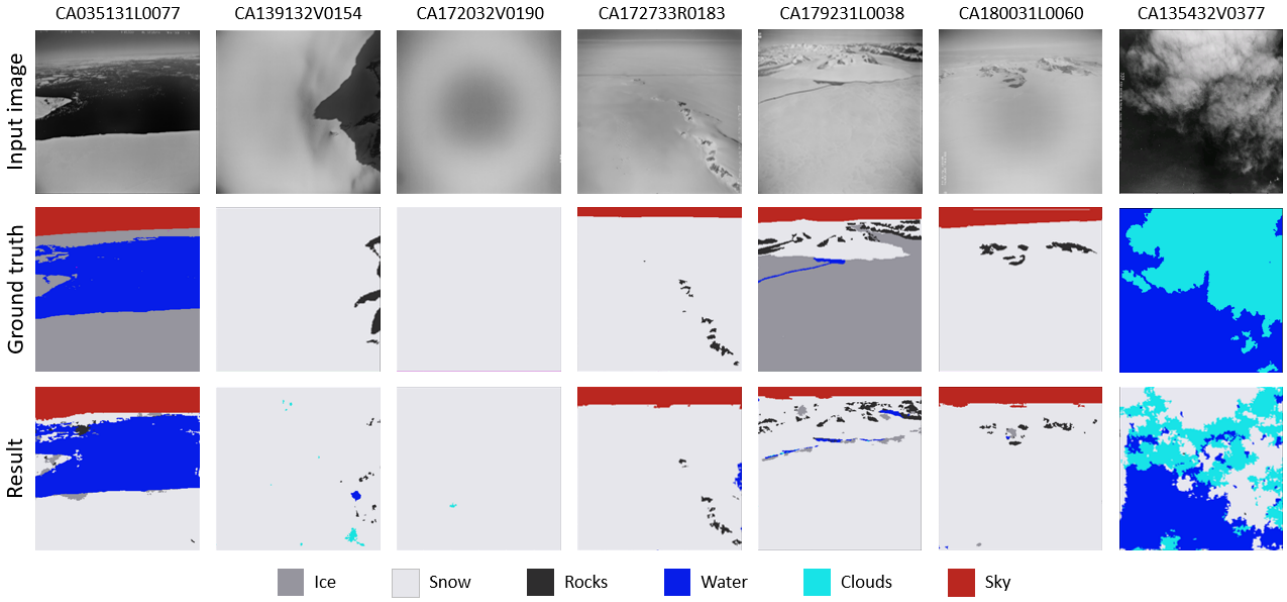

Figure 6. Result with input image (top), ground truth (middle) and prediction after post-processing (bottom)

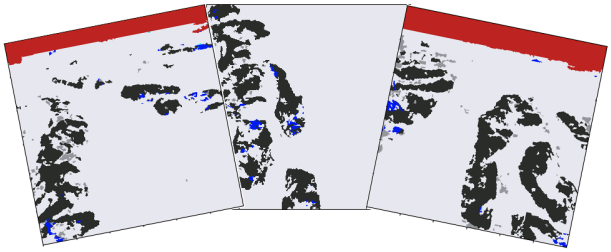

Figure 7. Semantic segmentation of the triplet from figure 1

ACKNOWLEDGMENTS

This work was funded by NWO-grant ALWGO.2019.044.

REFERENCES

Baumhoer, C. A., Dietz, A. J., Kneisel, C., Kuenzer, C., 2019. Automated Extraction of Antarctic Glacier and Ice Shelf Fronts from Sentinel-1 Imagery Using Deep Learning. *Remote Sensing*, 11(21), 2529.

Child, S. F., Stearns, L. A., Girod, L., Brecher, H. H., 2020. Structure-From-Motion Photogrammetry of Antarctic Historical Aerial Photographs in Conjunction with Ground Control Derived from Satellite Data. *Remote Sensing*, 13(1), 21.

Dias, M., Monteiro, J., Estima, J., Silva, J., Martins, B., 2020. Semantic Segmentation and Colorization of Grayscale Aerial Imagery with W-Net Models. *Expert Systems*, 37(6).

Diego Oliva, Essam H. Houssein, S. H. e., 2021. *Metaheuristics in Machine Learning: Theory and Applications*. Studies in Computational Intelligence, 1st ed. edn, Springer.

Dirscherl, M., Dietz, A. J., Kneisel, C., Kuenzer, C., 2021. A Novel Method for Automated Supraglacial Lake Mapping in Antarctica Using Sentinel-1 SAR Imagery and Deep Learning. *Remote Sensing*, 13(2), 197.

Dowden, B., De Silva, O., Huang, W., Oldford, D., 2021. Sea Ice Classification via Deep Neural Network Semantic Segmentation. *IEEE Sensors Journal*, 21(10), 11879–11888.

Garbin, C., Zhu, X., Marques, O., 2020. Dropout vs. Batch Normalization: An Empirical Study of Their Impact to Deep Learning. *Multimedia Tools and Applications*, 79(19-20), 12777–12815.

Hao, S., Zhou, Y., Guo, Y., 2020. A Brief Survey on Semantic Segmentation with Deep Learning. *Neurocomputing*, 406, 302–321.

Hartmann, A., Davari, A., Seehaus, T., Braun, M., Maier, A., Christlein, V., 2021. Bayesian U-Net for Segmenting Glaciers in Sar Imagery. *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, IEEE, Brussels, Belgium, 3479–3482.

Holzmann, M., Davari, A., Seehaus, T., Braun, M., Maier, A., Christlein, V., 2021. Glacier Calving Front Segmentation Using Attention U-Net. *arXiv:2101.03247 [cs]*.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, JMLR.org, 448–456.

Kanezaki, A., 2018. Unsupervised Image Segmentation by Backpropagation. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Calgary, AB, 1543–1547.

Kattenborn, T., Eichel, J., Fassnacht, F. E., 2019. Convolutional Neural Networks Enable Efficient, Accurate and Fine-Grained Segmentation of Plant Species and Communities from High-Resolution UAV Imagery. *Scientific Reports*, 9(1), 17656.

King, D. E., 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research*, 10, 1755-1758.

Kunz, M., King, M. A., Mills, J. P., Miller, P. E., Fox, A. J., Vaughan, D. G., Marsh, S. H., 2012. Multi-Decadal Glacier

Surface Lowering in the Antarctic Peninsula: Peninsula Glacier Surface Lowering. *Geophysical Research Letters*, 39(19).

Lu, M., Ni, G., Wang, T., Zhang, F., Tao, R., Yuan, J., 2013. Method for reducing Newton's rings pattern in the scanned image reproduced with film scanners. X. Lin, J. Zheng (eds), *2013 International Conference on Optical Instruments and Technology: Optoelectronic Imaging and Processing Technology*, 9045, International Society for Optics and Photonics, SPIE, 32 – 41.

Marochov, M., Stokes, C. R., Carbonneau, P. E., 2021. Image classification of marine-terminating outlet glaciers in Greenland using deep learning methods. *The Cryosphere*, 15(11), 5041–5059. https://tc.copernicus.org/articles/15/5041/2021/.

Mboga, N., Grippa, T., Georganos, S., Vanhuysse, S., Smets, B., Dewitte, O., Wolff, E., Lennert, M., 2020. Fully Convolutional Networks for Land Cover Classification from Historical Panchromatic Aerial Photographs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 167, 385–395.

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., Terzopoulos, D., 2021. Image Segmentation Using Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1.

Mohajerani, Y., Wood, M., Velicogna, I., Rignot, E., 2019. Detection of Glacier Calving Margins with Convolutional Neural Networks: A Case Study. *Remote Sensing*, 11(1), 74.

Phan, T. H., Yamamoto, K., 2020. Resolving Class Imbalance in Object Detection with Weighted Cross Entropy Losses.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (eds), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 9351, Springer International Publishing, Cham, 234–241.

USGS, 2018. USGS EROS Archive - Aerial Photography - Antarctic Single Frame Records. `https://www.usgs.gov/centers/eros/science/usgs-eros-archive-aerial-photography-antarctic-single-frame-records`, Last accessed on 2022-01-07.

Xu, W., Deng, X., Guo, S., Chen, J., Sun, L., Zheng, X., Xiong, Y., Shen, Y., Wang, X., 2020. High-Resolution U-Net: Preserving Image Details for Cultivated Land Extraction. *Sensors*, 20(15), 4064.

Yuan, X., Xie, L., Abouelenien, M., 2018. A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data. *Pattern Recognition*, 77, 160-172.