Revisiting the Past

A comparative study for semantic segmentation of historical images of Adelaide Island using U-nets

Dahle, Felix; Lindenbergh, Roderik; Wouters, Bert

DOI
10.1016/j.ophoto.2023.100056

Publication date
2024

Document Version
Final published version

Published in
ISPRS Open Journal of Photogrammetry and Remote Sensing

ISPRS Open Journal of Photogrammetry and Remote Sensing 11 (2024) 100056

Contents lists available at ScienceDirect

ISPRS Open Journal of Photogrammetry and Remote Sensing

journal homepage: www.journals.elsevier.com/isprs-open-journal-of-photogrammetry-and-remote-sensing

Revisiting the Past: A comparative study for semantic segmentation of historical images of Adelaide Island using U-nets

Felix Dahle [*], Roderik Lindenbergh, Bert Wouters

Department of Geoscience & Remote Sensing, Delft University of Technology, the Netherlands

ABSTRACT

The TriMetrogon Aerial (TMA) archive is an archive of historical images of Antarctica taken by the US Navy between 1940 and 2000 with analogue cameras. The analysis of such historic data can give a view of Antarctica's glaciers predating modern satellite imagery and provide unique insights into the long-term impact of changing climate conditions with essential validation data for climate modelling. However, the lack of semantic information for these images presents a challenge for large-scale computer-driven analysis.

Such information can be added to the data using semantic segmentation, but traditional algorithms fail on these scanned historical grayscale images, due to varying image quality, lack of colour information and artefacts in the images. To address this, we present a deep-learning-based U-net workflow. Our approach includes creating training data by pre-processing and labelling the raw images. Furthermore, different versions of the U-net are trained to optimize its hyper-parameters and augmentation methods. With the optimal hyper-parameters and augmentation methods, a final model has been trained for a use-case to segment 118 images covering Adelaide Island.

We tested our approach by segmenting challenging historical images using a U-net model with just 80 training images, achieving an accuracy of 73% for 20 validation images. While no test data is available for our use case, a visual examination of the segmented images shows that our method performs effectively.

The comparison of the hyper-parameters and augmentation methods provides directions for training other U-net-based models so that the presented workflow can be used to segment other archives with historical imagery. Additionally, the labelled training data and the segmented images of the test are publicly available at https://github.com/fdahle/antarctic_segmentation.

1. Introduction

Historical imagery archives provide valuable information about various parts of the world from the pre-satellite era. In recent years, there has been a growing trend of digitizing such archives and using these as a data source in geo-sciences (Cowley and Stichelbaut, 2012; Heisig and Simmen, 2021). However, despite their potential, they remain under-exploited as most images are only available as scans without any metadata. This lack of metadata makes extracting information challenging as it requires a significant amount of manual work to incorporate them into scientific research.

One such historical imagery archive is the TMA archive, where TMA stands for TriMetrogon Aerial, a system of cameras that takes vertical, left oblique, and right oblique images simultaneously for topographic mapping. The U.S. Navy collected this archive of historical imagery of Antarctica between 1946 and 2000, with a particular focus on the Antarctic Peninsula (USGS, 2018). These photographs were primarily used for topographic mapping and provide a historical snapshot of many parts of Antarctica. Fig. 1 shows an example image from this dataset.

This data set holds valuable information on historical ice topography and coverage in this area, for which few other data sources are available. In combination with recent observations, the TMA archive provides a unique opportunity to study multi-decadal changes in the state of the Antarctic Ice Sheet (Cook et al., 2016; Kunz et al., 2012; Cook and Vaughan, 2010). The data set, however, presents several challenges to its use. For example, as can be seen in Fig. 1, parts of the images can be obstructed by clouds or suffer from degradation due to the vinegar syndrome (decomposing of the film when stored for a longer time (Allen et al., 1987), see lower left part of the image). Furthermore, the archive consists of around 330.000 images, without any additional information on the content or quality of the images and with only an approximate geo-localization provided, making it difficult to find specific features in the image archive. Adding semantic information to the TMA image archive would significantly increase its usability. For example, this would allow researchers to find images on the boundary of ice and water to study areal changes of ice shelves and marine-terminating glaciers, detect rock outcrops, and provide information about the usability of individual images for specific research purposes, e.g. in terms of cloud

[*] Corresponding author.
E-mail addresses: F.Dahle@tudelft.nl (F. Dahle), R.C.Lindenbergh@tudelft.nl (R. Lindenbergh), Bert.Wouters@tudelft.nl (B. Wouters).

https://doi.org/10.1016/j.ophoto.2023.100056
Received 12 July 2023; Received in revised form 16 November 2023; Accepted 18 December 2023
Available online 25 December 2023

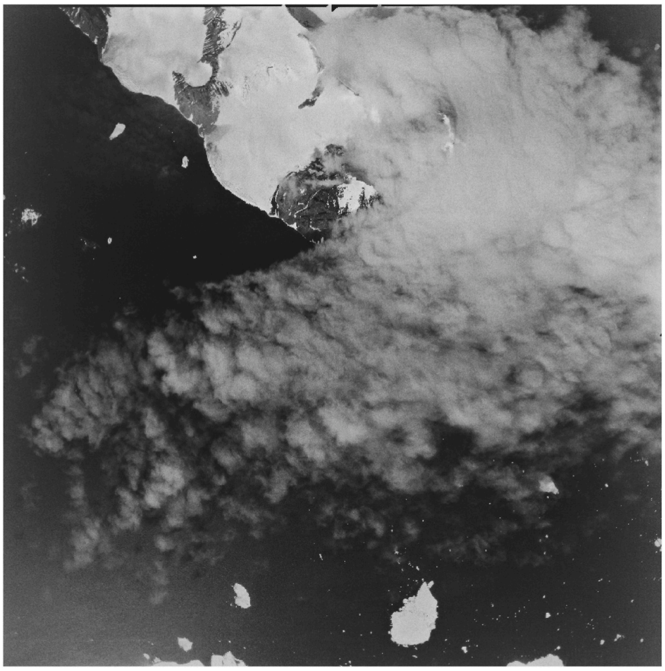

Fig. 1. Example of a historical image from the TMA archive (Antarctic Peninsula 1969).

cover or variability of the image.

Recent examples of adding semantic information can be found in (Nambiar et al., 2022; Wang et al., 2022; Heidler et al., 2021), where satellite imagery is segmented. In modern satellite imagery, information is available in multiple bands and with high contrast, so that segmentation is easily applicable. However, semantic segmentation of historical imagery is more challenging. The images are only available in grayscale with less contrast, making them less informative for segmentation. As a result, the algorithms developed for satellite imagery segmentation cannot directly be applied to historical imagery. Due to these problems, semantic segmentation of historical imagery is rare but some successful examples have been reported, such as (Mboga et al., 2020) and (Dias et al., 2020). In both cases, machine learning algorithms were used to apply semantic segmentation. However, these examples targeted a very diverse environment with very distinct classes, unlike the more monotone scenes in Antarctica.

Another big challenge for the semantic segmentation of historical imagery is the lack of training data: labelled data is often only available for modern data sources and cannot be used for historical images. Thus, all training data must be manually created beforehand, which is time-consuming, resulting in limited availability.

To address these challenges, we propose using a U-net for the semantic segmentation of the TMA archive. A U-net, originally developed for medical purposes by (Ronneberger et al., 2015), is a type of neural network specifically designed for image segmentation with a small amount of training data. Recently, U-nets gained popularity and are also extensively used for semantic segmentation in geo-science (Hartmann et al., 2021; Baumhoer et al., 2019; Heffels and Vanschoren, 2020; Kattenborn et al., 2019). In a previous paper (Dahle et al., 2022), we were able to create a semantic segmentation of part of the TMA historical imagery, even under challenging conditions, with an average accuracy of 74% over six classes using 67 images.

In this contribution, we build upon the use case and establish a fully operational workflow for the semantic segmentation of historical imagery of the cryosphere. To achieve this, we investigate the impact of different model parameters on the quality of the segmentation. It is worth noting that so far, most studies use default parameter settings and standard losses for training. However, adapting these parameters can lead to significantly improved results as shown in (Kugelman et al., 2022; Solórzano et al., 2021; Jadon, 2020). Nevertheless, these parameter comparisons often focus on a single parameter, and a holistic approach considering multiple parameters simultaneously is absent. Moreover, such comparisons are typically conducted on larger datasets with better image quality, making it challenging to extrapolate the findings to historical imagery segmentation.

To demonstrate the performance of our model, successive to the parameter evaluation, we apply the semantic segmentation to a geographical subset of the TMA archive, specifically Adelaide Island (see Fig. 2). The island is situated in the eastern part of the Antarctic Peninsula and is an enclosed area with a variety of different classes. As it features multiple flight paths, images are taken by different cameras.

2. Data

As input data, we utilize aerial images from the TMA archive. All pictures within the archives were made in triples, as can be seen in Fig. 3. Each image is associated with a unique identification number that comprises the flight line, roll, and frame. The roll indicates the direction

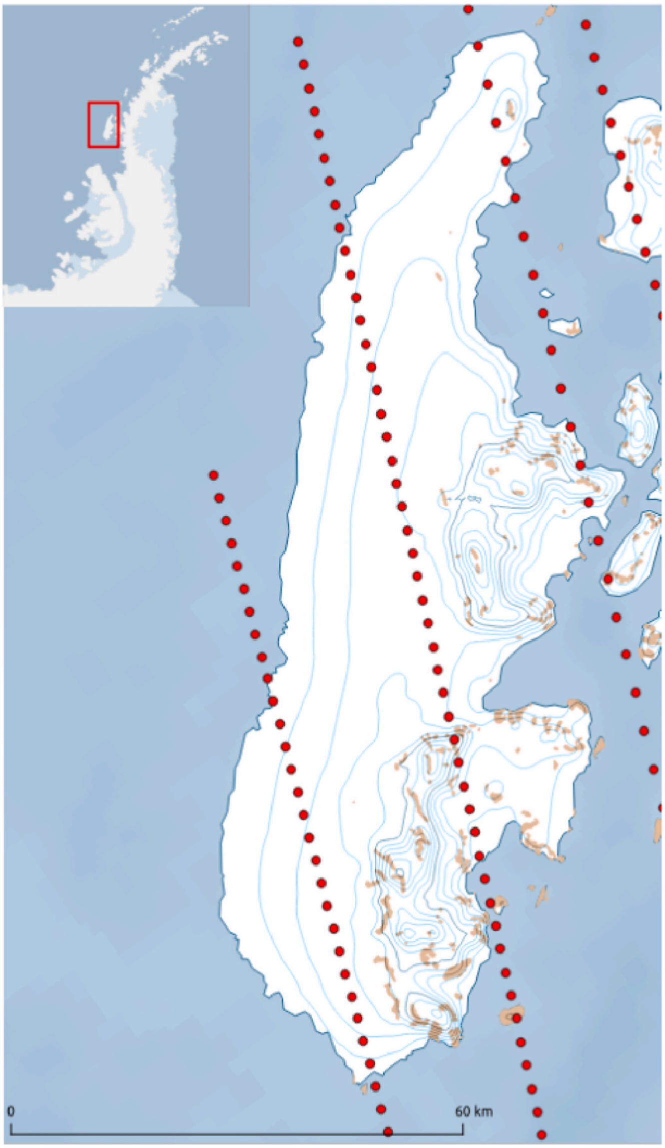

Fig. 2. Adelaide island with the position of cameras (red dots). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

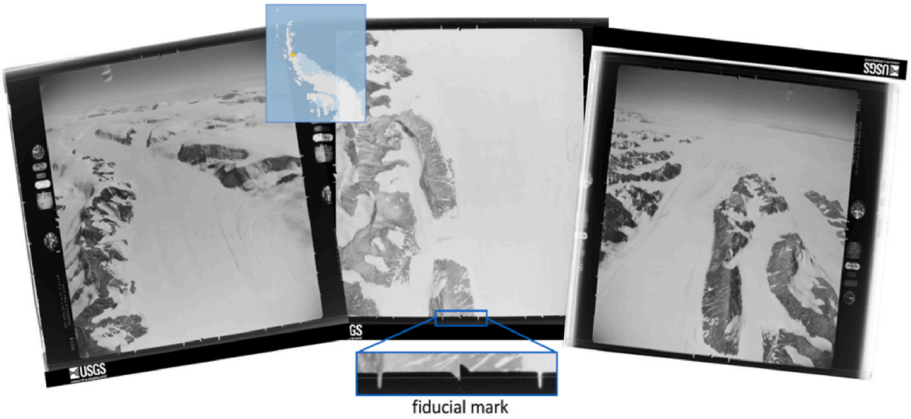

Fig. 3. Example image triple from Antarctic TMA, consisting of a left-oblique, a vertical and a right-oblique image.

the camera was facing, while the frame is a unique identifier for each image position on a flight line. For example, the identifier “CA026433R0058″ corresponds to an image captured during flight 0264, using the right roll (33R[1]), and has a frame number of 0058.

In total, this archive consists of around 330.000 grayscale images, which were digitally scanned in 2009 by the United States Geological Survey (USGS) at a resolution of 25 μm/1000dpi and made publicly available at their website (University of Minnesota, 2023) and their ftp-server (Polar Geospatial Center, 2023). With this resolution, all features on the historical images are recognizable and even small objects or objects in the distance can be identified. However, the scanning process was not flawless, as in many images scanning artefacts (e. g. Newton rings) are introduced.

The images are available as TIFF files with an image size of around 10000 × 10000 pixels and an 8-bit depth. It is important to note that these images are not geo-referenced, and only have a manually estimated indication of their position. Although the photo centre (position of the camera) is reported for every picture in the Antarctic Polar Stereographic coordinate system (EPSG:3031), a visual inspection has revealed that these coordinates can be inaccurate by several kilometres.

We define six different classes to be segmented in the images, as described in Table 1, together with some notes of their influence on the segmentation. Examples of the classes are shown in Fig. 4.

To train our model, we selected random images from the Antarctic Peninsula. This region is one of the most varying areas of the Antarctic with a diverse landscape, resulting in the most number of classes during segmentation. However, as there was no labelled training data available, we created the training data ourselves, as will be described in section 3.2.

Table 1
Description of classes.

Class	Notes
Ice	Only close to the water
Snow	Most dominant class and can be found on almost every image
Rocks	Small structures scattered in the images, usually easy to segment
Water	Second most dominant class
Clouds	Most difficult to segment; can contain traces of other classes beneath
Sky	Can only be found on oblique imagery
Unknown	Used when a pixel cannot be attributed to another class. Not existing in the final segmentation & no loss will be calculated for this class.

3. Methodology

In the following, the data pre-processing, the creation of the training data, and different attributes and design decisions of the segmentation process are explained. We compare the training and validation performance of different model parameters, and based on this, choose the best-performing combination of parameters. These are used to train a final segmentation model for more epochs. The parameters of this model can be found in subsection 4.3.

3.1. *Data pre-processing*

During pre-processing, the prevalent borders in the images, as can be seen in Fig. 3, 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 (King, 2009) and computer vision algorithms (e.g canny edge detection or Hough transform), these fiducial points can automatically be recognized and used to separate the inner part of the images from the borders.[2] Contrast enhancement, like used by (McNabb et al., 2020) for historical images cannot be used on the data: for some images, it improves the segmentation quality, but for other images with scanning errors it actually decreases the quality of the image and therefore does not improve the general quality of the model.

3.2. *Training data*

To generate the training data, we applied an unsupervised neural network for image segmentation to the raw images. This process produced preliminary image segments by identifying and grouping similar regions within the images. Various models of unsupervised segmentation are available, such as those described by (Kirillov et al., 2023) and (Kanezaki, 2018). For this study, we adopted the approach outlined by Kanezaki, which follows three main criteria: (1) pixels with comparable features are aggregated under the same label; (2) pixels that are spatially contiguous are also grouped under a single label; and (3) the overall number of unique labels is minimized to simplify the segmentation.

However, the unsupervised segmented images must be further processed to use them for training as they contain misclassified pixels and do not always match the images perfectly. Furthermore, these segments only have consecutive numbers as labels and contain no semantic information. The following steps were applied to improve the unsupervised images: (1) Renumbering segments: The segments created by the unsupervised segmentation could consist of multiple, non-connected parts. These parts are assigned a new number so that every unique

[1] 33 is a number describing that the camera is looking right, similar to a left (31L) and a vertical (32V) roll.

[2] Code is available on https://github.com/fdahle/Antarctic_TMA.

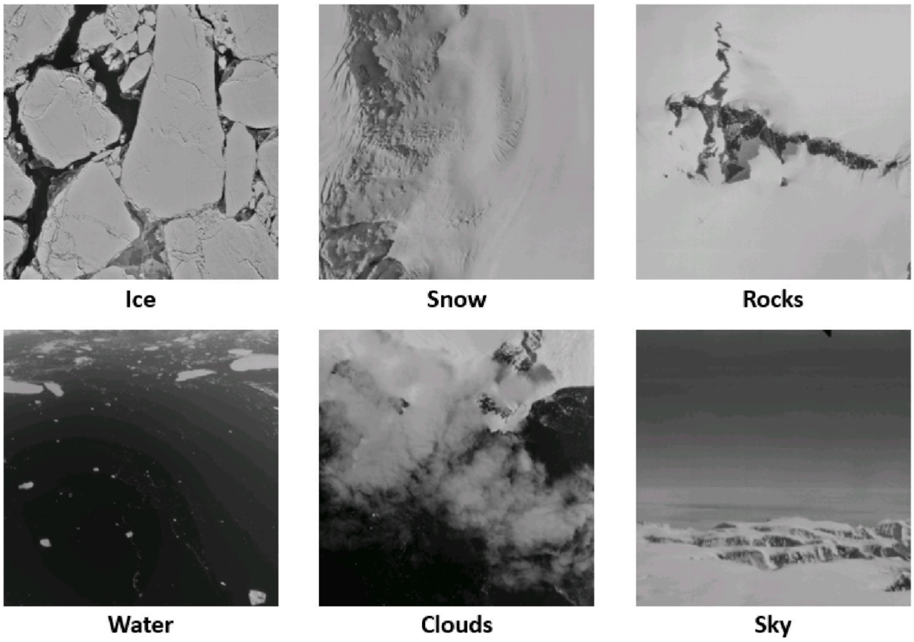

Fig. 4. Example for the classes with the class being the prominent feature in the image.

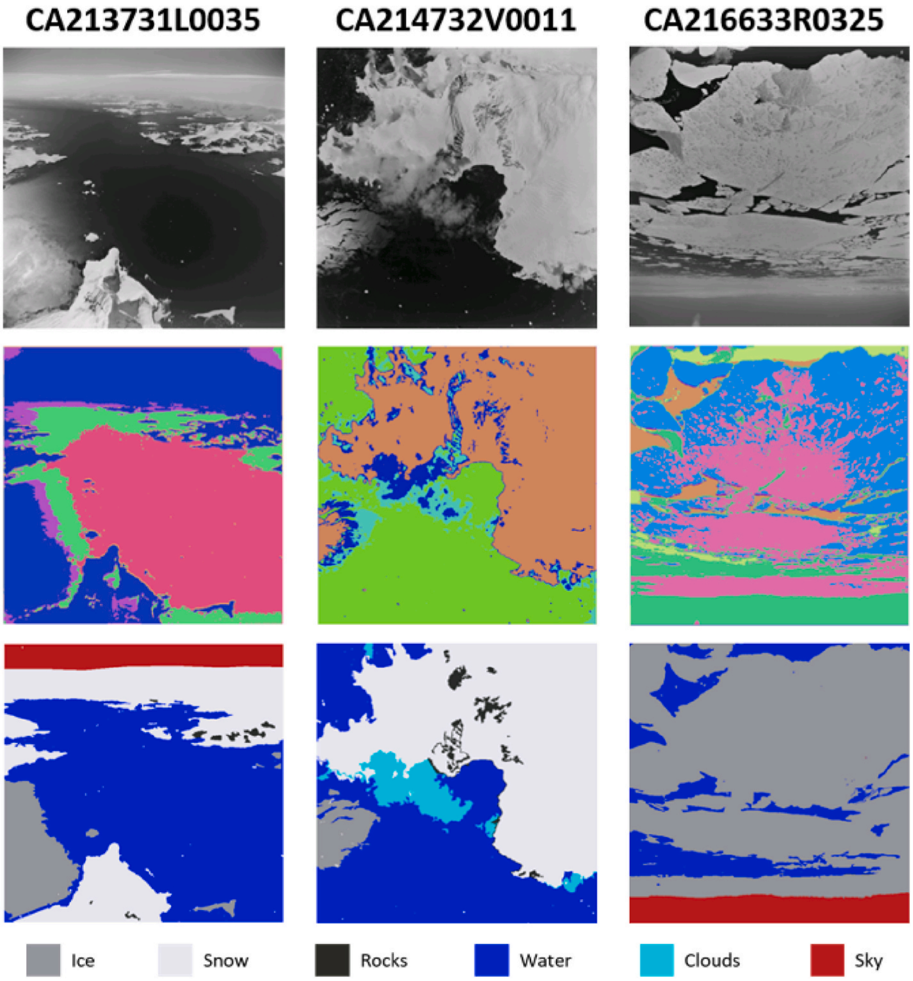

Fig. 5. Examples of self-labelled images with raw images at the top, the unsupervised segmentation in the middle (colours are assigned randomly) and the final obtained ‘ground truth’ at the bottom. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

segment has its own number as a label. (2) Removing small segments: we remove segments under a threshold size of 20 pixels to simplify the segmentation. Smaller segments are often considered to be noise or irrelevant to the classification task. (3) Filling voids: The removed segments from the previous step are filled with their surrounding pixels by using a watershed algorithm (Kornilov and Safonov, 2018). (4) Separating segments: Sometimes, only one segment is created, where in reality two different classes are present. For instance, in Fig. 5, which displays some examples of self-labelled training data, the classes sky and snow at the top of the left middle image are incorrectly merged in one single segment. These segments are separated manually after visual inspection.

To facilitate efficient labelling, we developed a tool in Python that allows segmenting images using the steps mentioned above. This enables for example relabelling of already labelled images together with adapting segments (e.g. separating).[3]

To distinguish the segments and assign the correct classes, the spatial context of the images must be known. However, due to the limited quality of the images, a correct assignment is not always possible. Creating labelled images is very time-consuming so the number of images for training and validation was limited to a total of 80 images. Of these, 80% was used as a training set and 20% as a validation set. As the data is highly imbalanced in class occurrences (see Table 2), the images cannot be assigned randomly to one of the two sets. This would have resulted in an unequal class distribution in both sets, making the model biased towards certain classes. Instead, the classes are divided using iterative stratification (Sechidis et al., 2011). This is a technique where a data set is divided into smaller subgroups all containing a similar composition of classes as the whole data set. Using this technique ensures an equal class distribution for both the training and validation set.

3.3. U-net

In this work, we utilize a U-net (Ronneberger et al., 2015), a model popular for image segmentation, which was adapted successfully for geospatial tasks (e.g., (Heidler et al., 2021), (Baumhoer et al., 2019)). In this model, convolutional neural networks (CNN) and fully convolutional networks (FCN), two special types of neural networks (Jiang et al., 2019), are combined in a U-formed structure. This architecture offers several advantages that are particularly relevant to our study. It can accommodate input images of varying sizes and is specifically designed to perform well even when training data are scarce; a situation we encounter with our set of 80 training images. Such a constrained dataset typically limits the efficacy of other well-established segmentation models like FCNs, Deeplab (Chen et al., 2017), and SegNet (Badrinarayanan et al., 2016), particularly when segmenting small structures within the images. Despite the computational intensity required by the U-net during the training phase, the resulting model is computationally efficient.

Our baseline U-net model consists of four layers, as depicted in Fig. 6. This structure is based on (Kattenborn et al., 2019) and contains multiple encoders (the contracting path) and multiple decoders (the expansive path). The encoders, situated on the left half of the figure, are a classical classification network where convolution blocks are applied followed by a max-pool downsampling to encode the input image into feature representations at multiple different levels. The decoders, located on the right half of the figure, semantically project the discriminative features learnt by the encoder onto the higher resolution pixel space to achieve dense classification at the pixel level. The decoder consists of up-sampling and concatenation followed by regular convolution operations.

Table 2
Class composition of train and validation set in percentage.

	Ice	Snow	Rocks	Water	Clouds	Sky	Other
Train	6.99	67.74	2.52	7.6	7.34	6.27	1.55
Validation	2.98	60.9	0.37	10.12	20.62	4.45	0.55
Complete	6.04	66.12	2.01	8.2	10.5	5.84	1.31

Each encoder/decoder in the network is built using the same components with the same attributes. The components include Conv2D, BatchNorm, Dropout, ReLU, MaxPool2D, and ConvTranspose2D. Conv2D is a convolutional layer that convolutes to additional feature maps with a kernel size of 3, followed by a stride of 1 to maintain image size. BatchNorm normalizes the input batch of 4 images by re-centering and scaling to make the network more stable and to converge faster (Ioffe and Szegedy, 2015). Dropout temporarily disables 20% of the nodes in the block during training, making the learning process more challenging, but reducing the chance of overfitting. ReLU is the activation function used in our network, which is short for rectified linear units (Goodfellow et al., 2016). It essentially removes all negative values from the output by setting them to zero. MaxPool2D downsizes the image to reduce the computational cost, using a kernel size of 3 with a stride of 2 to halve the image. ConvTranspose2D is a transposed convolutional layer that doubles the output image size compared to the input image size, also using a kernel size of 3 and a stride of 2.

The U-net model used in this work reduces the image size while increasing the number of feature maps in each encoding block, except for the first encoder/decoder, effectively reducing the computational cost. 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 a value between 0 and 1 describing how likely it is for each pixel to belong to that particular class. The segmented image is generated by applying the sigmoid function (Goodfellow et al., 2016) to the data and selecting the class with the highest probability for each pixel.

The parameters in Table 3 are commonly used in neural networks and remain consistent for all combinations of different segmentation models. The Adam-optimizer, a popular optimization algorithm in machine learning, is used to adapt the learning rate during run-time for faster convergence and better performance (Kingma and Adam, 2014). As the memory size of the used GPU (NVIDIA Tesla P100 with 16 GB RAM) is limited, all data is split up into batches of the maximum possible size and the results of all batches together are averaged. None of the models of this study were pre-trained, as then we can ensure complete control over the structure and parameters of the segmentation models.

3.4. Tests

Different parameters of the U-net will be tested for their performance in image segmentation. For every parameter, a new model is trained. All models are applied on the same dataset, having the same images in the training and validation set. The model with the parameters described in Table 4 is used as a baseline. For every test category, only the specific parameter of this category is changed. Tests for parameters will be done within six categories: additional layer components, learning rate, losses, number of layers, input size and augmentation, as elaborated below. Due to time and performance constraints, every model is trained for exactly 500 epochs with no early stopping.

3.4.1. Additional layer components

Next to the model components that are required for the model to learn, it is common to add additional components to each layer of the model. These components can help against overfitting as well as improve the quality of the model. For our segmentation, we use dropout and batch normalization, two commonly deployed elements in modern CNN architectures (Garbin et al., 2020b). Dropout layers, as utilized by

[3] The tool can be found at https://github.com/fdahle/Antarctic_TMA.

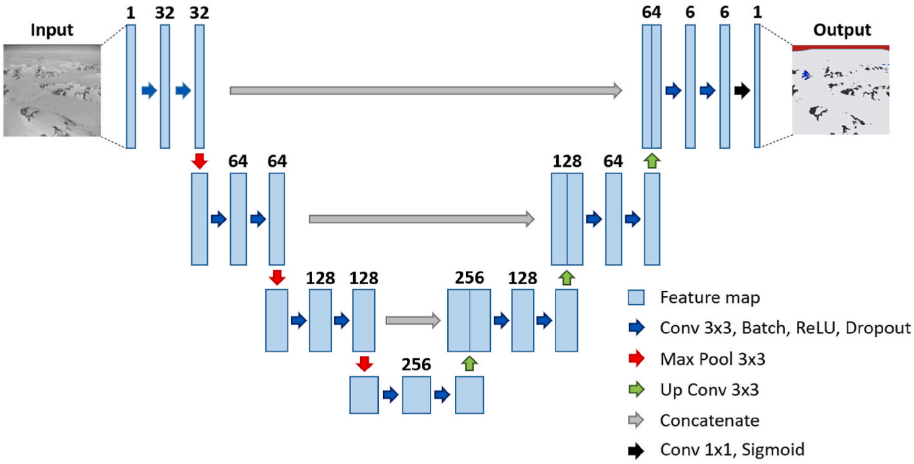

Fig. 6. The U-net takes a grayscale image as input and returns for each pixel and class a probability, which can be used to create a segmented output image.

Table 3
Model parameters that remain consistent during all trainingsa

Type	Value
Optimizer	Adam
Kernel size	3
Batch size	4
Training percentage	80%
Validation percentage	20%

Table 4
Parameters of the baseline model.

Parameter	Setting
Nr. of layers	normal (4 layers)
Learning rate	0.001
Loss	Cross entropy
Input-size	1200 × 1200 (resized)
Augmentation	No Augmentation
Overfitting	Dropout & Batch normalization

(Baumhoer et al., 2019), randomly set a percentage of neuron weights to 0 during training, withholding information from the model. This technique improves the model's robustness against overfitting but may also make it more challenging for the network to interpret the data. Alternatively, (Garbin et al., 2020a) recommends using batch normalization, which re-centres and re-scales the inputs to the layers, resulting in improved generalization and faster training. In order to quantify the influence of these components, we are testing both components as a first step, each separately and also using none of these components.

3.4.2. *Learning rate*

The learning rate is a number between 0 and 1 and influences how quickly a model can adapt to new data and determines how much the weights of a model are changed at every training iteration. A very small learning rate may result in very long training times because the weights only change by a very small amount. Increasing the learning rate will speed up the process, but comes with the danger of learning sub-optimal weights too fast and, therefore, leading to worse model quality. In our tests, we compare learning rates of 0.0001, 0.001, 0.01 and 0.1.

3.4.3. *Losses*

In machine learning, a loss function is an essential component for the learning process to accomplish its assigned task. The loss function allows describing numerically how well the prediction fits the model. During training, this information is used to change the parameters of the model. The loss is usually expressed as a numeric value starting from 0, where a value of 0 indicates perfect predictions of the data with no deviations, and higher values indicate worse predictions of the model. As the class 'unknown' should not be prevalent in the final classification, no loss is calculated for this class. Numerous loss functions are available for different applications, including image segmentation as documented in (Jadon, 2020). Here, we compare three common loss functions for semantic segmentation. In the following equations x_i describes the input, y_i is the target and C is the number of classes.

1. Weighted Cross-Entropy loss

The cross-entropy loss is one of the most commonly used loss functions in machine learning, with the weighted cross-entropy loss being an adaption for imbalanced data sets, as we deal with in our study. Based on a term from information theory, cross-entropy measures the entropy between two different probability class distributions. It is calculated with equation (1). Here w_i is the weight of a class and p_i its probability. The weight of the classes is the inverse probability of each class.

$$Weighted\ Cross\ Entropy\ Loss = \frac{1}{C}\sum_{i=1}^{C} w_i y_i log(p_i) \tag{1}$$

2. Focal loss

The focal loss is another loss especially suited for imbalanced datasets. It was originally designed by (Lin et al.) for object detection but was used with success for semantic segmentation as well. The loss function is a dynamically scaled cross-entropy loss, where the scaling factor decays to zero as confidence in the correct class increases so that the model is focusing on harder examples. In equation (2), α_i is a weighting factor for each sample, γ is a tunable focusing parameter and p_i is the probability of a class. The value of these parameters depends on the dataset and typically involves setting the weighting factor higher for the minority class and experimenting with different values of gamma to balance the model's ability to learn from hard examples and generalize to new data. Gamma is set to 2, and the weight of the classes is again the inverse probability of each class.

$$Focal\ Loss = \frac{1}{C}\sum_{i=1}^{C} -\alpha_i (i - p_i)^{\gamma} log(p_i) \tag{2}$$

3. DICE Loss

This loss is based on the Sørensen–Dice coefficient, which is used to estimate the similarity of two different samples. Like focal loss, it is mainly used to address a class imbalance in images, mainly due to a common imbalance of foreground and background pixels. It is calculated with equation (3). Here *N* describes the mini-batch size, a further and smaller subdivision of the batches. The choice of mini-batch size depends on factors such as available memory, dataset size, and model complexity, and is typically determined empirically by starting with a moderate size and adjusting based on performance and memory requirements.

$$Dice\ loss = \frac{2\sum_i^N x_i y_i}{\sum_i^C x_i^2 + \sum_i^C y_i^2} \tag{3}$$

3.4.4. Model depth

This term describes how 'deep' the model is, so how many encoding and decoding layers the model has. The model size has a direct influence on the number of parameters/weights that are trained. Typically, increasing the number of layers improves the accuracy of a model. However, as more parameters need to be calculated, the training time and model size will increase. Furthermore, it is possible to over-fit the data. On the other hand, it is also important not to have too few layers, as complex scenarios may not be learned, which means the model is underfitting.

To test the impact of model size, four different models with varying numbers of layers will be examined, including 2, 3, 4, and 5 layers.

3.4.5. Input size

Although the U-Net allows for flexibility in input image size, the images were resized prior to training. The images have an original size of around 10000 × 10000 pixels, which is too large to be used for training due to time and memory constraints. To address this issue, two different options were examined during parameter testing:

1. Resizing images: all images are resized to 1200 × 1200 pixels. This method is easily applicable, does not change the total number of images and the complete image is still depicted. However, information in the image is lost, especially texture, which is required for distinguishing classes with similar intensity values, like snow and clouds.
2. Cropping images: instead of using the complete image for training, a crop of 256 × 256 is taken from the image. This can either be a crop from the same location in every image or a crop based on a random location. In this way, no information on texture is lost. Although this method preserves information on texture, it presents some challenges for the dataset used in this study. Due to the class imbalance, taking a random crop means that the crops may also be imbalanced. In this case, the dominant class would be present in the majority of random crops, whereas other classes would only be present in a few crops. To mitigate this issue, we use the inverted random crop. A weight is assigned to every class, based on the occurrence of the classes in the dataset (see Table 2; classes with a smaller occurrence get higher weights), and a random crop is taken with a weighting.

3.4.6. Augmentation

In some cases, there may not be sufficient data to train a model, i.e. due to the lack of training data or the creation of additional training data being too time-consuming. A prime example of these cases is medical imagery, e.g. where some forms of cancer are too rare to have an adequate number of examples (Ayalew et al., 2021). For these cases, it is possible to apply data augmentation, a technique that is used to synthetically increase the amount of data, in order to provide a model with more samples and therefore increase the quality of the model. It is possible to either copy and adapt existing data or create completely synthetic data. An example of data augmentation in geo-sciences applications can be found in (Feng et al., 2022) for hyperspectral image classification. The augmented data are not created before the training nor added to the pool of available images. Instead, whenever the images are required for training an epoch of the model, they are randomly augmented with any of the augmentation methods, each applied with its own probability.

For this paper, we focus on the first option. We test different augmentation methods, used individually and all combined. The following methods are tested: (1) Flipping: The images are randomly (p = 0.5) flipped vertically and/or horizontally. (2) Rotation: The images are randomly (p = 0.5) rotated 90° for a random number (1–3) of times. (3) Brightness: The pixel values in the images are randomly (p = 0.5) increased or decreased by a random number (from 1 to 10). (4) Noise: Gaussian noise is added randomly (p = 0.5) to the images. (5) Normalize: The image values are normalized from 0 to 255 to a range from −1 to 1. Fig. 7 illustrates a visualization of these augmentation methods (except for normalization, as there is no visual change for this augmentation).

In addition to the benefits conferred by an expanded training dataset, the implementation of image augmentations by flipping and rotating offers further advantages. These augmentations can simulate common scanning errors, such as images being captured upside-down or rotated by 90°. By introducing these variations during training, the model can be conditioned to effectively process and segment such erroneously scanned images. Similarly, augmentations that adjust brightness and apply Gaussian noise enable the model to better handle images that are underexposed, overexposed, or affected by scanning artefacts. This preparatory step enhances the model's robustness and its ability to generalize from a broader range of input conditions.

3.5. Post-processing

Another important step to further improve the segmentation quality is post-processing. Our post-processing consists of the following steps: First, the images are resized to 2000 × 2000 pixels to speed up the post-processing. Subsequently, patches with a size smaller than 50 pixels are removed from the segmentation and filled with the values of surrounding pixels via the watershed algorithm. Next, the images made with cameras facing down vertically are checked for the presence of segments of the class sky. If present, these segments will be replaced with the value of the surrounding pixels via a watershed algorithm, since it is impossible for down-looking images to observe the sky. In non-vertical images, the class sky is enlarged to fill complete rows, when a row exceeds 50% of sky pixels. Furthermore, some combinations of the class sky are physically impossible. Examples would be small patches of the class sky that are located far away from the sky at the top of an image or small patches of the class snow inside the sky. As can be seen in Fig. 8, these patches are automatically recognized with computer vision methods, are removed and then again filled with the value of the surrounding pixels.

Finally, some logical criteria are applied to handle the confusion of the classes rock and water. Whenever a cluster is smaller than a threshold of 100.000 pixels and does not neighbour any cluster of its class, the class is changed (i.e., a rock cluster gets changed to water and vice versa).

3.6. Evaluation metrics

The evaluation of the image segmentation results can be complex as both the accuracy and the correct localization of the segmented images must be considered. Furthermore, an imbalanced distribution of the prominent classes during segmentation can lead to a statistical bias with incorrect high evaluation scores (Müller et al., 2022). Finally, not all incorrect segmentation labels are equally wrong. For example, confusion between the classes 'Ice' and 'snow' may have fewer implications than confusion between the classes 'sky' and 'snow'.

Almost all commonly used metrics are based on a computation of a

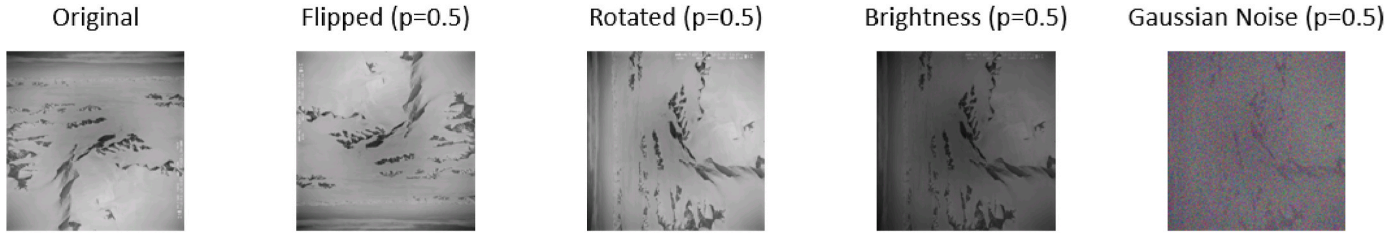

Fig. 7. Four different augmentation methods of the training images together with their probability.

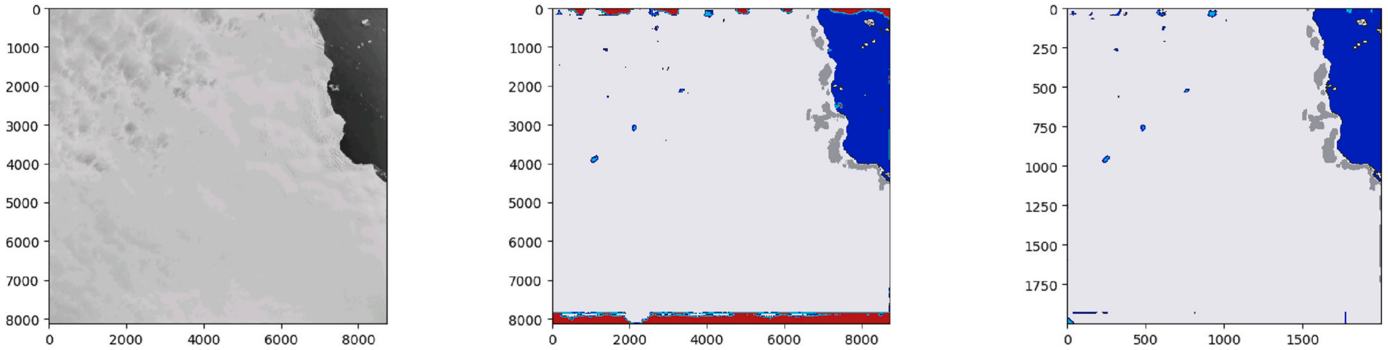

Fig. 8. For the initial segmented image (middle) post-processing is applied to improve the quality of the final segmentation (right). Here the sky is removed from the borders of the image.

confusion matrix, in which a pixel has one true class and one predicted class, which can result in four different outcomes: true positive (TP), true negative (TN), false positive (FP) and false negative (FN). Here, the models will be evaluated with the loss, accuracy and the F1-score. Precision and recall are important metrics required for the F1-score and therefore explained as well: (1) Accuracy: The most classical metric, which gives the percentage of pixels segmented correctly. However, this parameter is skewed by imbalanced datasets. If for example, 90% of an image contains the class ‘snow’, a model that learns to always classify every pixel to ‘snow’ will have an accuracy of 90%, even though the model is not useful. (2) Precision: a measure of quality, this parameter shows the number of correctly segmented pixels in relation to all pixels with this class attributed. It shows the ability of the model to segment a class with only pure results, with as few false segmented pixels as possible. Yet, it does not reflect the ability to capture all pixels of this class category. (3) Recall: a quantitative measure which shows the number of correctly segmented pixels in relation to all pixels with this class in reality. It indicates the ability of the model to segment as many pixels of a certain class correctly but does not take into account whether other classes are wrongly predicted as this class. (4) F1-Score: This parameter is a combination of both precision and recall using a harmonic mean. Only when both values have a high score, does the F1-score as well have a high score. It is generally seen as a more accurate score than accuracy for imbalanced datasets, even though it is less intuitive.

4. Results & discussion

In this section, we present the results of our study on optimizing hyper-parameters for semantic segmentation. We analyze the performance of the model using various hyper-parameters and present our findings. Based on these results, we identify the optimal set of hyper-parameters and train a segmentation model with them. We evaluate the performance of the optimized model on a set of 20 test images and provide a detailed discussion of the results. Additionally, we apply the model in a use-case to a larger set of images of Adelaide Island in Antarctica to demonstrate its effectiveness in real-world scenarios.

4.1. *Performance of the base-model*

Fig. 9 displays the training and validation performance of the base model with the most basic parameter settings. Even with these most basic settings, the model is observed to be learning, as evidenced by the decreasing training loss and increasing evaluation values.

During training, an interesting pattern emerges for the base model until around 500 epochs: The validation loss remains constant or even increases slightly, while the training loss is constantly decreasing. This is typically an indication of overfitting, where the model learns to just predict the training data, but not to create generalizations over the data. However, the evaluation parameter values continue to improve for testing and, importantly, for validation as well.

Further experimentation with the base model using more unseen data, however, reveals that this model is still learning, as it can segment better than the same model with fewer iterations. Therefore, classical overfitting does not appear to be the case here. Instead, this phenomenon can be explained by characteristics of the cross-entropy and the input data. Whereas most predictions improve (better scores for the evaluation parameters), other already bad predictions worsen (in this case the prediction of ‘ice’ as ‘snow’) and have a higher influence on the rising loss. Furthermore, the prediction probabilities for all classes may be increasing, for both incorrect and correct classes, which can further increase the loss. However, the right class is still maintaining the highest probability, so that the segmentation is not changing. Both phenomena combined can lead to a situation, in which the loss is rising, but the performance of the models is still improving.

However, around epoch 500, the model is showing signs of overfitting, with an increase in validation loss and lower values for the evaluation parameters.

4.2. *Training & validation performance of individual parameters*

Fig. 10 displays the training and validation performance of the individual parameters. The evaluation performance of the models (accuracy & F1-score) generally increases for each combination, indicating that the models are learning to segment the model. However, depending on the exact parameters, the performance can differ drastically. The

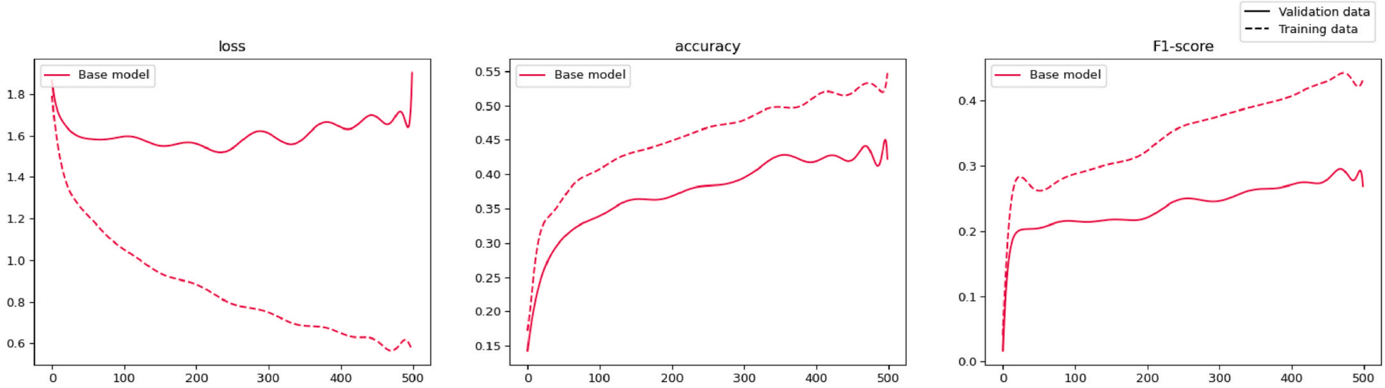

Fig. 9. Results for base model with loss, accuracy and F1-score for both validation and training data.

average validation values over all different parameter tests are 0.47 for accuracy and 0.48 for the F1-score.

The following can be noticed for the performance of the individual parameters:

4.2.1. Additional components

The results for the test with additional components align with the expectations. Dropout, in which a random percentage of the neurons is disabled so that they do not transmit information while learning, makes it harder for the model to train, leading to higher loss values for both training and validation. Even though the model's performance is decreasing, the chance of overfitting is smaller. On the other hand, batch normalization increases the model's performance, resulting in a smaller loss and increased performance for all evaluation parameters. For these reasons, both components were used in the subsequent tests.

4.2.2. Learning rate

In general, higher learning rates lead to worse model performance. In the extreme case of a learning rate of 0.1, the model is hardly learning anything and the loss remains nearly constant. The model is converging too fast to a sub-optimal solution and gets stuck iterating around it (see (Ketkar and Ketkar, 2017)). For lower learning rates, the model is improving and learning. With smaller learning rates, the weights are upgraded with smaller steps, allowing the model to converge to more optimal solutions at the cost of increased learning time. Notably, with a minimal learning rate of 0.0001, the model progresses in learning but requires a more extended period to achieve the performance level of a model trained with a learning rate of 0.001.

4.2.3. Loss type

All three loss functions allow the model to learn the image segmentation and the evaluation values are improving. For the cross-entropy loss, the training loss is decreasing, but the validation loss remains constant and even increases again at the end, as explained earlier. The focal loss yields minimal loss values in comparison to the other two losses. However, this is not related to a better performance of this loss, instead, the loss values are calculated differently. The evaluation performance of this loss is the lowest of all losses. Whereas the cross entropy loss yields higher accuracy, the dice loss performs better for the F1-score.

4.2.4. Model depth

Regardless of the model depth, the segmentation model is able to improve, but the two-layer model is less capable of learning segmentation, resulting in higher loss and lower evaluation parameters. To a limited extent, this also holds for three and five layers, whereas four layers are the sweet spot with the best loss and evaluation parameters. With more layers, too many parameters are included, so the model is overfitting. With fewer parameters, the model lacks the complexity to capture the relationship between the input and output variables for segmentation (underfitting).

4.2.5. Input size

When comparing the performance of the different input sizes in Fig. 10, an interesting behaviour similar to the lost type can be noticed. For training and validation, the model with resized input data has a higher accuracy, while cropping the input data yield better F1-scores for both. These conflicting results suggest that neither input size outperforms the other in terms of overall evaluation metrics. However, considering the better validation loss values and that the F1-score is better suited to evaluate imbalanced data sets, cropped images seem to be a better choice. Comparing the models for the different input sizes visually reveals that the resized model is better at giving the correct classes, whereas the cropped model is more accurate in extracting boundaries between classes.

4.2.6. Augmentation

Including augmentation data in the model did not have the desired effect, as most single methods have a higher loss and yield worse evaluation performance than the model without augmentation. The only exception is the 'noise' augmentation, which delivers better performance at the end of the model training. Combining all the augmentation methods leads to the worst results in terms of both loss and evaluation metrics. Using a single augmentation method only has a small influence. However, some augmentation methods, like applying 'rotation' and, to a smaller extent, 'flipping' decrease the performance of the model even further. This is in line with the findings of (Engstrom et al., 2019), who reported that perturbations such as translations and rotations can degrade the performance of a neural network.

4.3. Numerical quantification

The insights from the tests for individual parameters are used to train a new model with the combination of the best-performing hyper-parameters. These parameters are described in Table 5.

Both dropout and batch normalization are used against overfitting. The learning rate of 0.001 and model depth of 4 are used as the best-performing parameters. For the augmentation parameters, only noise was included, as it brings the most performance gain. For the loss type and input size, there is no single best-performing hyper-parameter or optimal solution, as it depends on whether accuracy or F1-score is more important.

During the training of some initial models, it was noticed through visual inspections that the model using cropped input data is better at detecting the boundaries between classes, whereas the model with resized input data excels at detecting the correct classes. This effect can even be enhanced by combining the different input sizes using different losses: dice loss for cropped input data and cross-entropy loss for resized

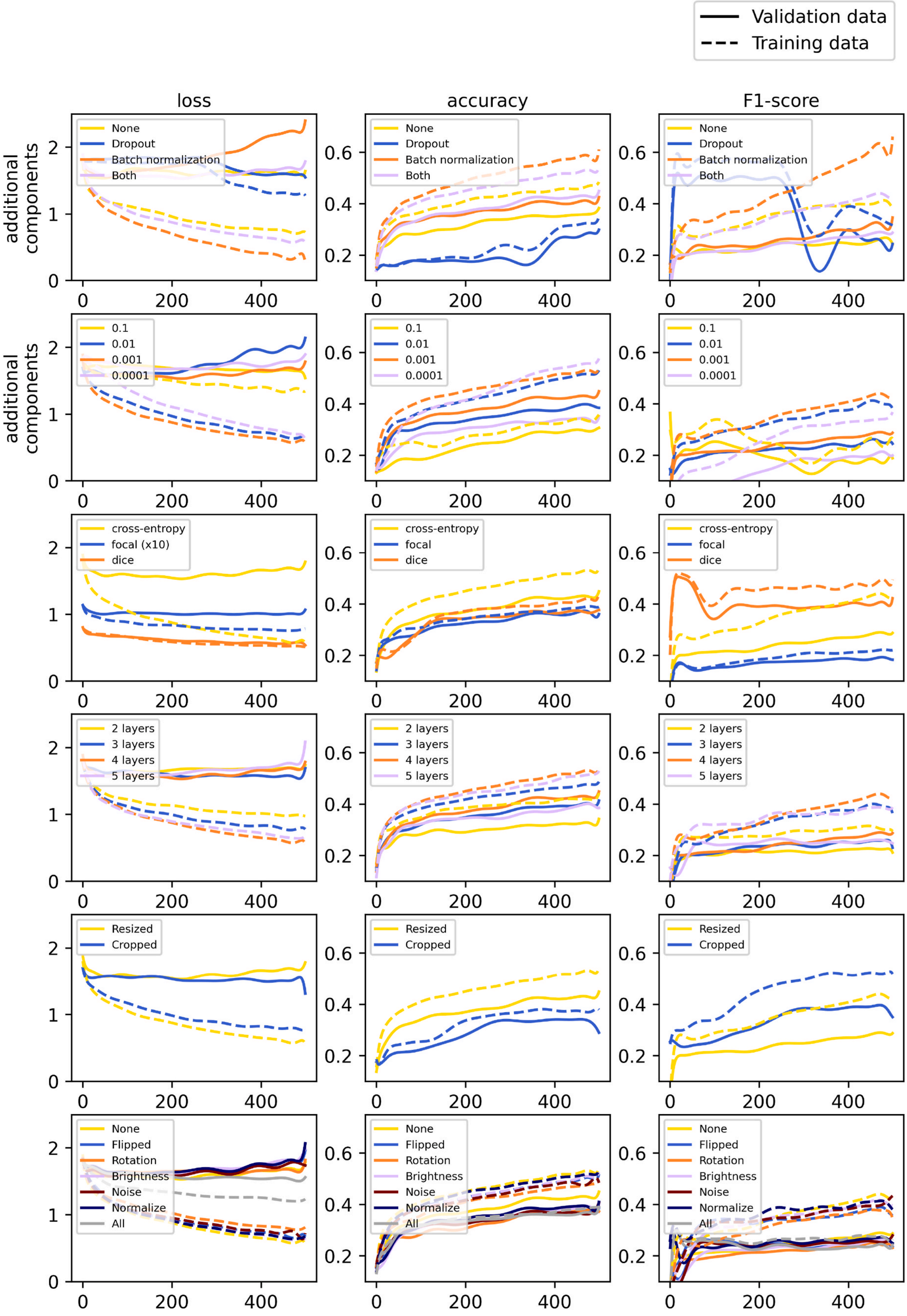

Fig. 10. Results for the models with the different parameters discussed in subsection 3.4 with each test showing loss, accuracy and F1-score for both validation and training data. For displaying purposes, the loss values for focal loss are exaggerated by factor 10.

Table 5
Parameters of the final model used for the numeral quantification.

Parameter	Value
Additional components	Dropout & Batch normalization
Learning rate	0.001
Loss	Dice Loss & Cross entropy
Model depth	4 layers
Input size	Cropped & Resized
Augmentation	Noise

input data. Therefore, we utilize the strengths of both combinations to further improve the quality of the segmentation by using both models together and merging the results. The segmented from cropped image serves as the base segmentation, and some clusters (enclosed segments of the same class values) from the segmentation of the resized images are used as well: The position and class of a cluster from the resized image replace the classes at the same position in the cropped image, as the segmentation performs better for the classes sky, ice, and clouds in the resized model. Fig. 11 visualizes this combining step: The cropped segmentation is the base image, whereas clouds & ice from the resized segmentation are taken from the resized image.

The final segmentation is therefore based on a combination of two models with the best-performing parameters for each cropped and resized imagery. The first model utilizes crops of 512 × 512 pixels and the dice loss, while the second model utilizes resized images of 1200 × 1200 pixels and the cross-entropy loss. The cropped model was trained for 500 epochs, while the resized model was trained for 360 epochs. The total training time was 74 h for the cropped model and 45 h for the resized model.

The model's numerical quantification is based on an evaluation of 20 semi-manually labelled (see subsection 3.2) images as ground truth. These images are randomly selected from the complete archive. Fig. 12 visualizes the segmentation of all twenty evaluation images with the optimal model. The accuracy, precision, recall, and F1-score are 0.73, 0.84, 0.72, and 0.71, respectively, based on the right confusion matrix of Fig. 13.

The difference between the confusion matrix of the base model and the optimized model is evident. In the base settings, the model tends to classify pixels as snow with a high probability due to the imbalance of the dataset, where snow is the most dominant class. However, the only two exceptions are the classes water and sky, which have opposite colour values (black instead of white). In contrast, the model with optimized hyper-parameters performs better in terms of correct classifications. Although the accuracy for some classes may have decreased, such as sky from 0.69 to 0.66 or snow from 0.9 to 0.81, this is likely due to the introduction of noise into the data (In return the model performs better on under- or overexposed images) Nonetheless, the model with additional hyper-parameters is more stable and better suited to handle new, unseen data.

When looking at the example images of Fig. 12, for every image the general semantic meaning can be extracted successfully. However, for some scenarios and images, the model underperforms. Notably, in images T4 and T15, rock structures are accurately recognized, but the model fails to distinguish between snow and sky, particularly when snow covers a significant portion of the ground. This can possibly be attributed to the underexposure of the camera, in which the ground is too dark and resembles the sky of the training data. The augmentation could not help to mitigate this, so more underexposed imagery could be included in the training data to address this limitation, as it would teach the model to differentiate correctly between underexposed snow and sky. Another challenge for the model is distinguishing between water and rocks, as both share similar attributes for the network. However, with post-processing, most of these confusions can be cleared and the correct class can be assigned.

It is worth noting that mismatches between classification and evaluation may be due to errors in the ground truth data. Generating ground truth data is a time-consuming process, and limitations in the accuracy of the data may occur, especially when dealing with small structures. Additionally, there may be limitations in correctly assigning classes. For example, clouds may not completely obstruct the surface beneath them, and distinguishing between the clouds and the surface beneath can be challenging. Ice often gets covered with snow, leading to confusion between the two classes. In image T10, the ground truth data identifies the entire image as snow, although the model correctly identifies small rock structures. This leads to lower evaluation scores as it is counted as an error. Nonetheless, it is essential to acknowledge these limitations when interpreting evaluation results.

When compared to the evaluation conducted in (Dahle et al., 2022), the current assessment, at first sight, does not demonstrate significant improvement in quantitative performance. However, the current study utilized a new set of images that featured more complex scenery, with a higher proportion of cloud cover and greater underexposure. This poses significant challenges for semantic segmentation models as they obscure important details and create inconsistencies in image quality. Despite these more challenging conditions, the new model performed well, maintaining its performance on even more difficult images.

Our model's segmentation approach closely resembles that of human labellers. It performs well in scenes that are easily recognizable by humans while encountering similar difficulties in challenging scenes. However, the model surpasses human capabilities in segmenting smaller structures with greater accuracy (for example T10 or T14 of Fig. 12). On the other hand, the model struggles with under- or overexposed images. Despite this limitation, the model's speed is a significant advantage over human labellers, taking only 10 s on a standard computer to segment an image compared to 20 min on average for a human (based on the creation of the training data for this model).

Overall, while the model shows promising results in adding semantic information to the images, there is still room for improvement,

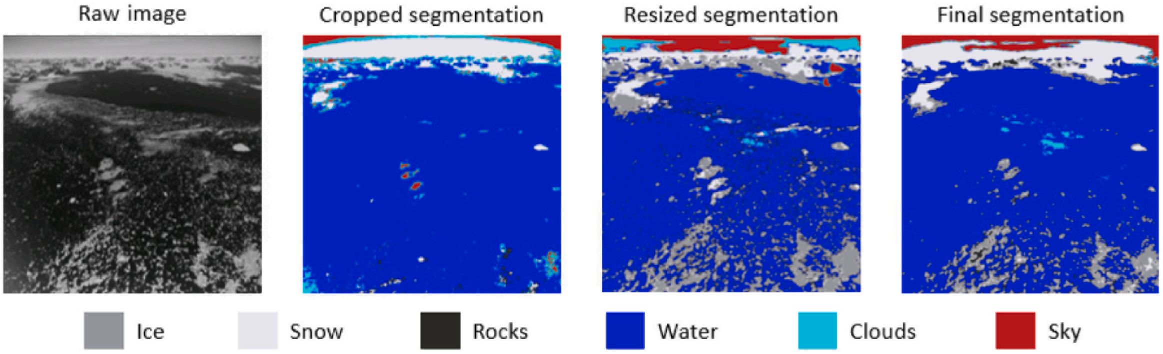

Fig. 11. Example for combining two models with the images from left to right: raw image, cropped segmentation, resized segmentation and final segmentation after post-processing.

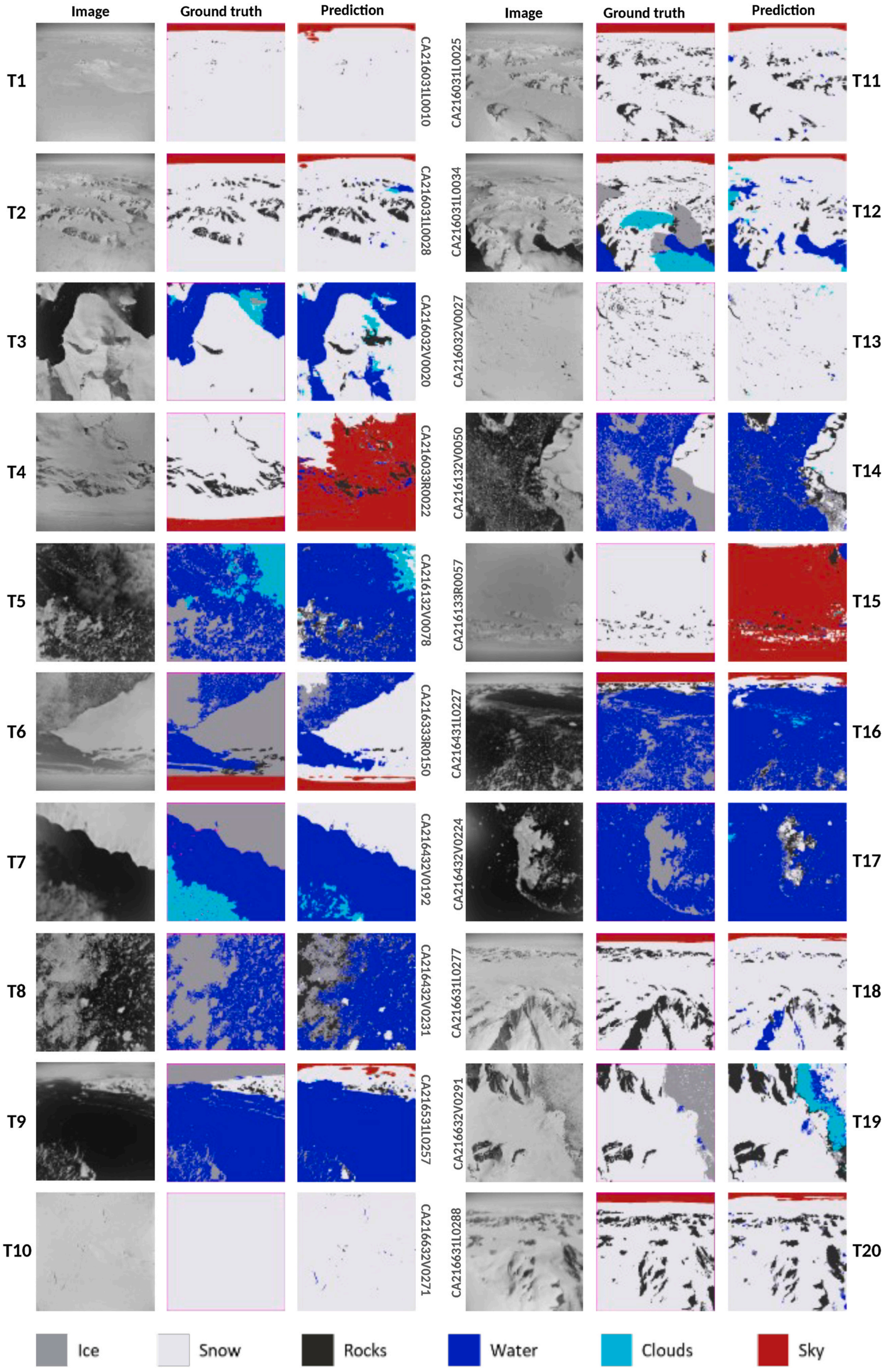

Fig. 12. Segmentation of twenty randomly selected images (Test image T1 - T20). For each triplet, the raw image is on the left, the manually created ground truth in the middle and the results of the segmentation on the right.

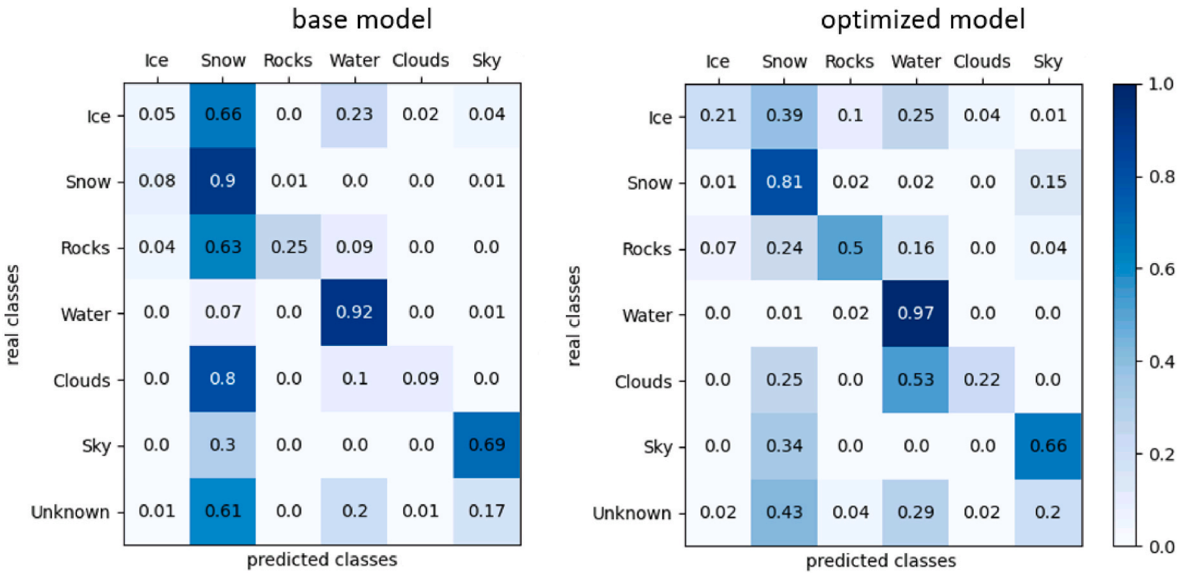

Fig. 13. Normalized confusion matrix for the base model (left) and the optimized model (right) with real classes on the left and predicted classes on the top.

particularly in distinguishing between snow and clouds. Future research could focus on improving the model's performance and incorporating validation data to evaluate its accuracy further.

4.4. Use-case Adelaide island

To demonstrate the capabilities of our optimized segmentation model beyond the validation set of 20 images, we applied the model to the vertical images of Adelaide Island from 1969 and quantified the results. Fig. 14 shows the 118 segmented vertical images of Adelaide Island, that were roughly geo-referenced based on the camera positions shown in Fig. 2. It is important to note that the position is only an approximation based on the (uncertain) reported camera positions, which is evident from the difference in coastline positions between the segmentation and the base map.

Due to the lack of validation data for Adelaide Island, we cannot provide a numerical assessment of the segmentation model's performance in this area. However, a visual inspection of the segmented images reveals that the model successfully adds semantic information to the images, particularly in differentiating between land, water, and ice, as is shown by the segmentations in bounding boxes A and B. Nevertheless, the model has limitations, as can be seen in bounding box C where it struggles to differentiate between snow and clouds, especially in the centre of the clouds, where there is significant visual overlap with snow. However, due to the limited data quality of these images, these areas are also difficult to segment for humans.

5. Conclusion

In this work, we successfully segmented an archive of historical aerial imagery of Antarctica using a U-net-shaped neural network. We compared and discussed the training and validation performance of different hyper-parameters. Combining the best-performing parameters, we applied a model on a test set and segmented images of Adelaide Island.

Several other segmentation methods exist, e.g., k-means clustering, random forest approaches, or other deep-learning-based approaches (Lateef and Ruichek, 2019). However, these methods seem unsuitable for this dataset, as they require either better quality data or much more labelled training data.

The proposed model is able to get the semantic meaning of a scene for the historical images of the TMA archive, even when using grayscale images with low contrast, conditions for which many other segmentation models would fail. The created model successfully learns to distinguish between most of the different classes with a certain confidence and does not get disrupted by unfavourable conditions, like poor image quality, limited spectral information, difficult semantic classes and only a few training images.

To our knowledge, no other semantic segmentation model exists that can work under these conditions. Based on the added information through segmentation, specific images with certain attributes can be selected, such as images located at the ice-ocean boundary, or images containing sky. This greatly facilitates case-specific inquiries into local conditions within the archive, thereby enabling various applications related to the historical conditions of Antarctica.

In theory, almost every combination of model parameters allows training a model for semantic segmentation and getting some meaningful output. However, the tests demonstrate that selecting the correct parameters can impact the segmentation's quality and even can account for training with a limited number of images and/or images with suboptimal quality. The findings can be extended to the segmentation of other historical imagery to enable a better selection of hyper-parameters at the start of the segmentation.

When it comes to parameter selection for semantic segmentation, we suggest the following guidelines: parameters such as model depth, learning rate, and the choice of loss function have a direct impact on the model and therefore have the biggest influence on the quality of the model. While data augmentation is beneficial, particularly when dealing with a limited number of images, its impact is not as strong. However, it is important to keep in mind that there is no universally optimal parameter that can be applied across all scenarios; the choice of parameters must be tailored to the specific characteristics of the dataset and the objectives of the study.

Even though the training time of both models (cropped and resized) takes a significant amount of time, the segmenting of an image itself is fast, with an average time of around 10 s per image, including post-processing. As a next step, we aim to extend the application of the model beyond Adelaide Island and use it to segment the entire TMA archive of 300.000 images, which will allow easy access to the data and encourage its further usage by the community. Besides filtering for specific classes, segmentation allows to correct oblique images which were scanned upside-down, so that the sky is at the top. It also enables the detection of rock outcrops in the images, which can then be used to address the poor geo-location of the imagery by matching with present-

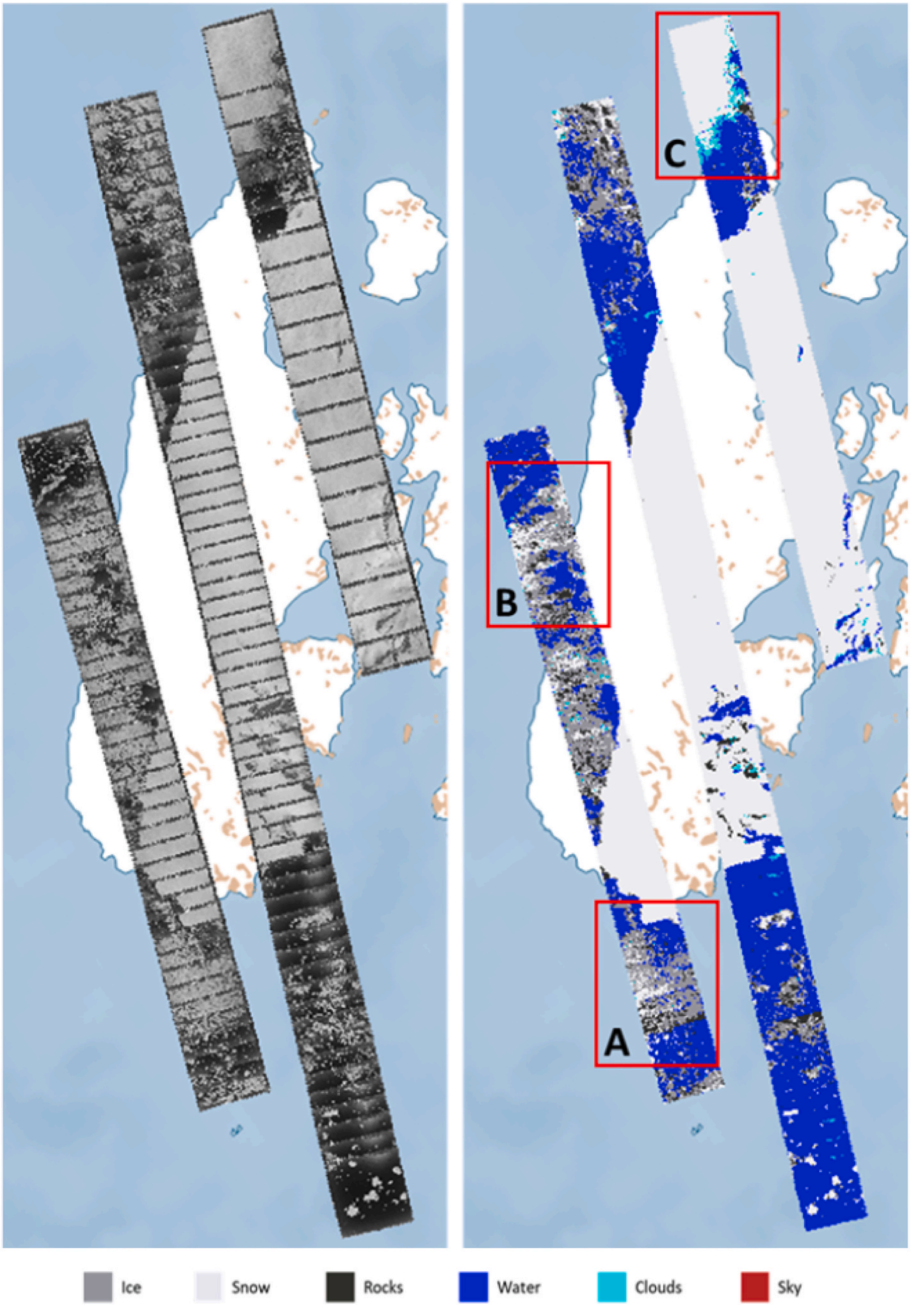

Fig. 14. 118 raw images (left) and segmented images (right) for Adelaide island.

day outcrop data sets (Burton-Johnson et al., 2016). Furthermore, it can help improve applications, such as Structure-from-motion to create 3D models (Child et al., 2020), by excluding irrelevant tie points (i.e. identical points in overlapping images) in the classes "water" and "sky". Tie-points detected in these classes are often of worse quality due to movements inside the class. The removal of these points enhances the quality of the 3D models and is a necessary step to automatize the process so that in future efforts the TMA Archive can be used to create historical 3D models at a large scale.

To enhance the model's robustness and generalization capabilities, it is possible to employ an even broader spectrum of augmentation techniques. For instance, incorporating affine transformations can effectively simulate various forms of image deformation. The application of Gaussian blur is beneficial for mimicking atmospheric disturbances or the blurring effects associated with cameras being out of focus. However, we expect that the most effective option would be to increase the size of the training data set. However, even though other historical cryospheric datasets exist, e.g., for Svalbard, Greenland or Alaska (Girod et al., 2018; Bjork et al., 2012; Knuth et al., 2023), these archives do not currently contain segmented images. Therefore, they cannot be used as training data without further manual work. A viable alternative is to acquire labelled training data from modern satellite images (e.g., Sentinel-2) and artificially degrade them to resemble historical data. Another possible option would be to include metadata of the images as additional data sources, such as whether an image is taken with an oblique- or vertical-facing 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 the use of this model for other images where these additional metadata are not available.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

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

6 Annex

Table 6 displays all images that were used during the training and validation of the models and the respective class composition per image.

Table 6
Class composition per image

Image	Sea ice	Snow	Rocks	Water	Clouds	Sky	Other	set
CA135431L0337	4.6	20.8	0	6.8	36.6	12.8	18.4	train
CA135431L0343	0	0	0	0	72.2	10.9	16.9	train
CA135433R0343	0.5	0	0	19.2	65.9	9.6	4.9	train
CA135631L0036	29.8	0	0	8.7	59.6	0	1.9	train
CA135632V0032	23.3	0	0	42.9	33.4	0	0.4	train
CA135633R0037	0	2.8	0	1.1	83	11.1	2.2	train
CA139132V0154	0	98	2	0	0	0	0	train
CA179231L0038	70.8	16.4	3.5	0.8	0	8.6	0	train
CA180031L0060	0	89	2.2	0	0	08.8	0	train
CA181331L0123	0	86.5	0.8	0	0	10	2.7	train
CA181332V0125	0	97.5	0	0	0	0	2.5	train
CA181333R0125	0	97.3	0.1	0	0	2.7	0	train
CA182433R0047	0	90.4	0.2	0	0	8.1	1.3	train
CA182433R0050	0	85.3	0.4	0	0	12.3	2	train
CA182933R0037	0	88.4	0	0	0	11.6	0	train
CA183431L0012	0	99.2	0	0	0	0	0.8	train
CA183432V0034	0	100	0	0	0	0	0	train
CA183432V0045	0	95	3.7	0	0	0	1.3	train
CA183532V0067	0	98.9	0	0	0	0	1.1	train
CA183533R0058	0	86.2	1.7	0	0	12.1	0.1	train
CA184333R0078	0	89.2	0	0	0	9.8	1	train
CA184431L0143	0	84.5	4.6	0	0	9.1	1.8	train
CA184432V0094	0	99.7	0	0	0	0	0.3	train
CA184432V0113	0	100	0	0	0	0	0	train
CA184432V0115	0	100	0	0	0	0	0	train
CA184432V0154	0	99.6	0	0	0	0	0.4	train
CA184531L0226	0	83.1	1.4	0	0	15.1	0.4	train
CA184532V0199	0	99.6	0	0	0	0	0.4	train
CA184532V0201	0	100	0	0	0	0	0	train
CA184532V0219	0	100	0	0	0	0	0	train
CA184532V0229	0	96.5	3.5	0	0	0	0	train
CA184532V0231	0	87.1	12	0	0	0	0.9	train
CA184533R0206	0	88.9	1.8	0	0	9.3	0	train
CA184533R0229	0	83.4	4.1	0	0	12.6	0	train
CA184533R0238	0	84.5	2.5	0	0	11.8	1.1	train
CA184733R0095	0	94	0	0	0	6	0	train
CA212333R0050	0	83.6	0.4	0	0	14.3	1.7	train
CA213731L0035	8.9	26	0.5	55.3	0	7.7	1.6	train
CA213731L0038	0	20.9	1.6	53.8	08.6	5.9	9.3	train
CA213733R0050	3.5	27.3	0.1	52.2	0	14.8	2.1	train
CA214732V0011	3.9	49.2	2.2	37.4	6	0	1.3	train
CA214831L0099	0.1	23.2	1.9	48.9	13.5	10.2	2.1	train
CA214832V0090	0	97.1	2	0	0	0	0.9	train
CA214833R0100	0.1	55	3.5	25.5	0.7	12.3	2.9	train
CA214932V0146	0	89.9	9.6	0	0	0	0.4	train
CA215032V0257	0	78.8	20.6	0	0	0	0.6	train
CA215131L0274	0	76.6	10.2	0	0	11.8	1.3	train
CA215131L0288	0	73.4	16.3	0	0	10.2	0.1	train
CA215132V0275	0	37.3	7.8	0	54.2	0	0.6	train
CA215331L0411	0	74.2	11.6	0	0	11.9	2.3	train
CA215333R0402	0	83.2	1.5	0	0	12.9	2.4	train
CA215731L0063	0	85.4	6.7	0	0	7.7	0.2	train
CA216631L0328	27.2	55.6	4.3	5.9	0	6.1	0.8	train
CA216632V0331	85.6	0	0	13.9	0	0	0.4	train
CA216633R0325	78.4	0.1	0.1	14.1	0	7.4	0	train
CA216633R0332	88.6	0	0	4.1	0	7.3	0	train
CA216731L0333	0	82.7	7.3	0	0	9.9	0	train
CA216733R0338	0	89.4	0.3	1.3	0	9	0	train
CA216733R0346	0	90.9	0.1	0.8	0	8.2	0	train
CA216733R0367	0	90.5	0.5	0	0	9	0	train
CA512933R0013	1.1	0	0	70.6	14.2	13.5	0.6	train

(*continued on next page*)

Table 6 (*continued*)

Image	Sea ice	Snow	Rocks	Water	Clouds	Sky	Other	set
CA035131L0077	45.8	0	0	40.2	0	14	0	val
CA135431L0352	3.2	8.2	0.2	3.9	83.8	0	0.8	val
CA135432V0337	0	0	0	48.6	50.3	0	1.1	val
CA135433R0350	5.1	0	0	32	49.9	9.1	3.9	val
CA135632V0031	26	0	0	24.5	72	0	0.8	val
CA168431L0207	0	0	0	43.1	45.5	11.3	0	val
CA172032V0190	0	100	0	0	0	0	0	val
CA172733R0183	0	92.2	1	0	0	6.8	0	val
CA180031L0079	0	0	0	0	90.3	9.6	0.1	val
CA182033R0051	0	87.7	1.8	0	0	10.5	0	val
CA182431L0059	0	93	2	0	0	5	0	val
CA183032V0009	0	99.5	0	0	0	0	0.5	val
CA183432V0005	0	99.6	0	0	0	0	0.4	val
CA183432V0041	0	99.8	0.2	0	0	0	0	val
CA183433R0044	0	92.2	0	0	0	7.3	0.5	val
CA183531L0087	0	86.8	0.4	0	0	11	1.8	val
CA183532V0060	0	99.7	0	0	0	0	0.3	val
CA184332V0060	0	98.4	1.5	0	0	0	0.2	val
CA184432V0105	0	100	0	0	0	0	0	val

References

Allen, N., et al., 1987. Degradation of historic cellulose triacetate cinematographic film: the vinegar syndrome. In: Polymer Degradation and Stability, vol. 19, pp. 379–387. https://doi.org/10.1016/0141-3910(87)90038-3, 4.

Ayalew, Y.A., Fante, K.A., Mohammed, M.A., 2021. Modified U-net for liver cancer segmentation from computed tomography images with a new class balancing method. In: BMC Biomedical Engineering, vol. 3, p. 4. https://doi.org/10.1186/s42490-021-00050-y, 1.

Badrinarayanan, V., Kendall, A., SegNet, R. Cipolla, 2016. A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation arXiv: 1511.00561 [cs.CV].

Baumhoer, C.A., et al., 2019. Automated extraction of antarctic glacier and ice shelf fronts from Sentinel-1 imagery using deep learning. In: Remote Sensing, vol. 11, p. 2529. https://doi.org/10.3390/rs11212529, 21.

Bjork, A.A., et al., 2012. An aerial view of 80 Years of climate-related glacier fluctuations in Southeast Greenland. In: Nature Geoscience, vol. 5, pp. 427–432. https://doi.org/10.1038/ngeo1481, 6.

Burton-Johnson, A., et al., 2016. An automated methodology for differentiating rock from snow, clouds and seain Antarctica from Landsat 8 imagery: a new rock outcrop map and areaestimation for the entire antarctic continent. In: The Cryosphere, vol. 10, pp. 1665–1677. https://doi.org/10.5194/tc-10-1665-2016, 4.

Chen, L.-C., et al., 2017. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs arXiv: 1606.00915 [cs.CV].

Child, S.F., et al., 2020. Structure-from-motion photogrammetry of antarctic historical aerial photographs in conjunction with ground control derived from satellite data. In: Remote Sensing, vol. 13, p. 21. https://doi.org/10.3390/rs13010021, 1.

Cook, A.J., Vaughan, D.G., 2010. Overview of areal changes of the ice shelves on the antarctic Peninsula over the past 50 years. In: The Cryosphere, vol. 4, pp. 77–98. https://doi.org/10.5194/tc-4-77-2010, 1.

Cook, A.J., et al., 2016. Ocean forcing of glacier retreat in the western antarctic Peninsula. In: Science, vol. 353, pp. 283–286. https://doi.org/10.1126/science.aae0017, 6296.

Cowley, D.C., Stichelbaut, B.B., 2012. Historic aerial photographic archives for European archaeology. In: European Journal of Archaeology, vol. 15, pp. 217–236. https://doi.org/10.1179/1461957112Y.0000000010, 2.

Dahle, F., et al., 2022. Semantic segmentation of historical photographs of the antarctic Peninsula. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-2-2022, pp. 237–244. https://doi.org/10.5194/isprs-annals-V-2-2022-237-2022.

Dias, M., et al., 2020. Semantic segmentation and colorization of grayscale aerial imagery with W-net models. In: Expert Systems, vol. 37. https://doi.org/10.1111/exsy.12622, 6.

Engstrom, L., et al., 2019. Exploring the landscape of spatial robustness. In: Proceedings of the 36th International Conference on Machine Learning. PMLR, pp. 1802–1811.

Feng, F., et al., 2022. Small sample hyperspectral image classification based on cascade fusion of mixed spatial-spectral features and second-order pooling. In: Remote Sensing, vol. 14. https://doi.org/10.3390/rs14030505, 3.

Garbin, C., Zhu, X., Marques, O., 2020a. Dropout vs. Batch normalization: an empirical study of their impact to deep learning. In: Multimedia Tools and Applications, vol. 79, pp. 12777–12815. https://doi.org/10.1007/s11042-019-08453-9, 19-20.

Garbin, C., Zhu, X., Marques, O., 2020b. Dropout vs. batch normalization: an empirical study of their impact to deep learning. In: Multimedia Tools and Applications, vol. 79, pp. 1–39. https://doi.org/10.1007/s11042-019-08453-9.

Girod, L., et al., 2018. Precise DEM extraction from svalbard using 1936 high oblique imagery. In: Geoscientific Instrumentation, Methods and Data Systems, vol. 7, pp. 277–288. https://doi.org/10.5194/gi-7-277-2018, 4.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT Press. http://www.deeplearningbook.org.

Hartmann, A., et al., 2021. Bayesian U-net for segmenting glaciers in sar imagery. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS. IEEE, Brussels, Belgium, pp. 3479–3482. https://doi.org/10.1109/IGARSS47720.2021.9554292.

Heffels, M., Vanschoren, J., 2020. Aerial imagery pixel-level segmentation. Tech. rep. Department of Mathematics and Computer Science, Eindhoven University of Technology.

Heidler, K., et al., 2021. HED-UNet: combined segmentation and edge detection for monitoring the antarctic coastline. In: IEEE Transactions on Geoscience and Remote Sensing, pp. 1–14. https://doi.org/10.1109/TGRS.2021.3064606.

Heisig, H., Simmen, J.-L., 2021. Re-engineering the past: countrywide geo-referencing of archival aerial imagery. In: PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science, vol. 89, pp. 487–503. https://doi.org/10.1007/s41064-021-00162-z, 6.

Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning. https://doi.org/10.48550/ARXIV.1502.03167.

Jadon, S., 2020. A Survey of loss functions for semantic segmentation. In: 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). Via del Mar, Chile. IEEE, pp. 1–7. https://doi.org/10.1109/CIBCB48159.2020.9277638.

Jiang, X., et al., 2019. CapsNet, CNN, FCN: comparative performance evaluation for image classification. In: International Journal of Machine Learning and Computing, vol. 9, pp. 840–848. https://doi.org/10.18178/ijmlc.2019.9.6.881.

Kanezaki, A., 2018. Unsupervised image segmentation by backpropagation. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, Calgary, AB, pp. 1543–1547. https://doi.org/10.1109/ICASSP.2018.8462533.

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. In: Scientific Reports, vol. 9, 17656. https://doi.org/10.1038/s41598-019-53797-9, 1.

Ketkar, N., Ketkar, N., 2017. Stochastic gradient descent. In: Deep Learning with Python: A Hands-On Introduction, pp. 113–132.

King, D.E., 2009. Dlib-ml: a machine learning toolkit. In: Journal of Machine Learning Research, vol. 10, pp. 1755–1758.

Kingma, D.P., Adam, J. Ba, 2014. A Method for Stochastic Optimization. https://doi.org/10.48550/ARXIV.1412.6980.

Kirillov, A., et al., 2023. Segment Anything arXiv: 2304.02643 [cs.CV].

Knuth, F., et al., 2023. Historical structure from motion (HSfM): automated processing of historical aerial photographs for long-term topographic change analysis. In: Remote Sensing of Environment, vol. 285, 113379. https://doi.org/10.1016/j.rse.2022.113379.

Kornilov, A.S., Safonov, I.V., 2018. An overview of watershed algorithm implementations in open source libraries. In: Journal of Imaging, vol. 4. https://doi.org/10.3390/jimaging4100123, 10.

Kugelman, J., et al., 2022. A comparison of deep learning U-net architectures for posterior segment OCT retinal layer segmentation. In: Scientific Reports, vol. 12, 14888. https://doi.org/10.1038/s41598-022-18646-2, 1.

Kunz, M., et al., 2012. Multi-decadal glacier surface lowering in the antarctic Peninsula: Peninsula glacier surface lowering. In: Geophysical Research Letters, vol. 39. https://doi.org/10.1029/2012GL052823, 19.

Lateef, F., Ruichek, Y., 2019. Survey on semantic segmentation using deep learning techniques. In: Neurocomputing, vol. 338, pp. 321–348. https://doi.org/10.1016/j.neucom.2019.02.003.

T.-Y. Lin et al. "Focal Loss for Dense Object Detection". In: arXiv:1708.02002 [cs] (Feb. 2018). arXiv: 1708.02002 [cs]..

Mboga, N., et al., 2020. Fully convolutional networks for land cover classification from historical panchromatic aerial photographs. In: ISPRS Journal of Photogrammetry and Remote Sensing, vol. 167, pp. 385–395. https://doi.org/10.1016/j.isprsjprs.2020.07.005.

McNabb, R., et al., 2020. An Open-Source Toolset for Automated Processing of Historic Spy Photos: sPyMicMac. https://doi.org/10.5194/egusphere-egu2020-11150 other.

Müller, D., Soto-Rey, I., Kramer, F., 2022. Towards a guideline for evaluation metrics in medical image segmentation. In: BMC Research Notes, vol. 15, p. 210. https://doi.org/10.1186/s13104-022-06096-y, 1.

Nambiar, K.G., et al., 2022. A self-trained model for cloud, shadow and snow detection in Sentinel-2 images of snow- and ice-covered regions. In: Remote Sensing, vol. 14, p. 1825. https://doi.org/10.3390/rs14081825, 8.

Polar Geospatial Center, 2023. Public HTTP data repository. https://data.pgc.umn.edu/aerial/usgs/tma/photos/. (Accessed 3 January 2023).

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. In: Navab, N., et al. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, vol. 9351. Springer International Publishing, Cham, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.

Sechidis, K., Tsoumakas, G., Vlahavas, I., 2011. On the stratification of multi-label data. In: Gunopulos, D., et al. (Eds.), Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 145–158.

Solórzano, J.V., et al., 2021. Land use land cover classification with U-net: advantages of combining sentinel-1 and sentinel-2 imagery. In: Remote Sensing, vol. 13, p. 3600. https://doi.org/10.3390/rs13183600, 18.

University of Minnesota, 2023. Aerial photography - antarctic single frames (1946-2000). https://www.pgc.umn.edu/data/aerial/. (Accessed 3 January 2023).

USGS, 2018. USGS EROS archive - aerial photography - antarctic single frame records. https://www.usgs.gov/centers/eros/science/usgs-eros-archive-aerial-photographyantarctic-single-frame-records. (Accessed 7 January 2022).

Wang, Y., et al., 2022. Snow coverage mapping by learning from sentinel-2 satellite multispectral images via machine learning algorithms. In: Remote Sensing, vol. 14, p. 782. https://doi.org/10.3390/rs14030782, 3.