Autonomous Labelling and Supervised Learning of a Deep Neural Network for Sea Ice Segmentation

Robert Nicholas Norfleet

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Ekaterina Kim
Prof. Maria Margarida Campos da Silveira

Examination Committee

Chairperson: Prof. João Manuel de Freitas Xavier
Supervisor: Prof. Ekaterina Kim
Member of the Committee: Prof. Alexandre José Malheiro Bernardino

November 2024

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

This work was created using LaTeX typesetting language
in the Overleaf environment (www.overleaf.com).

Acknowledgments

I want to thank my parents for their encouragement over these past two years through both the good and bad times. I never would have enjoyed this experience to the fullest degree without all my friends–both within the MIR programme and outside. I want to specifically thank my friends Rafael, Francisco, Max, Lara, and Leo for the continued friendship and support over the past two years and into the future. I also owe thanks to the wonderful communities in France, Portugal, and Norway that made me feel at home an ocean away, let me play volleyball with them, and showed me what beautiful countries they live in.

I would also like to acknowledge my dissertation supervisors Prof. Silveira and Prof. Kim for their support throughout the thesis writing process. Additionally, I want to specifically thank Ph.D candidate Oskar G. Veggeland for his help at every step of this process and for some excellent belays climbing in Trondheim.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the last two years. Thank you.

Abstract

The use of Deep Neural Networks (DNNs) for image segmentation shows promising results but requires large, labelled training datasets. These datasets are rare for specific domain studies and necessitate domain expertise and significant people hours to perform manual labelling. Using data acquired from a voyage in November 2023 aboard the *Kronprins Haakon*, an automated labelling technique was used to generate training data without any manual input. A camera-LiDAR payload designed by Oskar G. Veggeland captured 2,111 images with accompanying LiDAR point clouds. Three labelling methods were applied to the point cloud information, used to train three different models (U-Net, DeepLabV3+, SegFormer), and tested on a 361-image manually labelled dataset. Despite the point cloud information providing a semi-accurate map of sea ice in the images, the cloud suffered from excessive sparsity, necessitating the three preprocessing methods. Compared to the manually labelled ice masks, the Morphological and Otsu-Hybrid processed masks produced the best IoU scores at 0.60 each. After being trained on the Morphological dataset for 30 epochs, U-Net was able to achieve an IoU score of 0.76 on the manually labelled test set. Overall, the Morphological dataset-trained models demonstrated impressive recall scores whilst the Otsu-Hybrid dataset-trained models succeeded in precision. These results suggest that utilizing LiDAR information and image processing methods could represent a viable alternative to manual labelling of sea ice for binary segmentation. Code will be made available at: https://github.com/colenorfleet/sea_ice_segmentation.

Keywords

Deep neural networks; binary segmentation; LiDAR point cloud; sea-ice; image processing; automated labeling

Resumo

A utilização de Redes Neurais Profundas (DNNs) para segmentação de imagens mostra resultados promissores, mas requer grandes conjuntos de dados de treino rotulados. Estes conjuntos de dados são raros para estudos de domínios específicos e requerem experiência no domínio e horas de trabalho significativas para realizar a rotulagem manual. Utilizando dados adquiridos numa viagem em novembro de 2023 a bordo do *Kronprins Haakon*, foi utilizada uma técnica de etiquetagem automatizada para gerar dados de treino sem qualquer introdução manual. Uma carga útil de câmara-LiDAR concebida por Oskar G. Veggeland captou 2.111 imagens acompanhadas por nuvens de pontos LiDAR. Foram aplicados três métodos de rotulagem à informação da nuvem de pontos, utilizados para treinar três modelos diferentes (U-Net, DeepLabV3+, SegFormer) e testados num conjunto de dados rotulado manualmente de 361 imagens. Apesar da informação da nuvem de pontos fornecer um mapa semipreciso do gelo marinho nas imagens, a nuvem sofria de dispersão excessiva, necessitando dos três métodos de pré-processamento. Em comparação com as máscaras de gelo rotuladas manualmente, as máscaras processadas Morfológicas e Otsu-Híbridas produziram as melhores pontuações de IoU, 0,60 cada. Depois de ser treinado no conjunto de dados Morfológicas durante 30 épocas, o SegFormer conseguiu atingir uma pontuação IoU de 0,76 no conjunto de teste rotulado manualmente. No geral, os modelos treinados com conjuntos de dados morfológicos demonstraram pontuações de recuperação impressionantes, enquanto os modelos treinados com conjuntos de dados Otsu-Hybrid obtiveram precisão. Estes resultados sugerem que a utilização de métodos de processamento de imagem e informação LiDAR pode representar uma alternativa viável à rotulagem manual do gelo marinho para segmentação binária.

Palavras Chave

Redes neuronais profundas; segmentação binária; Nuvem de pontos LiDAR; gelo marinho; processamento de imagem; rotulagem automatizada

Contents

List of Figures

List of Tables

Acronyms

DNNs	Deep Neural Networks
DL	Deep Learning
NSR	Northern Sea Route
MASS	Marine Autonomous Surface Ships
LiDAR	Light Detection and Ranging
CNN	Convolutional Neural Network
PSPNet	Pyramid Scene Parsing Network
SPP	Spatial Pyramid Pooling
ViT	Vision Transformer
NLP	Natural Language Processing
MLP	MultiLayer Perceptron
ERF	Effective Receptive Field
FOV	Field-of-View
BCE	Binary Cross-Entropy
CBAM	Convolution Block Attention Module
CT	Computed Tomography
UAV	Unmanned Aerial Vehicle
SIPP	Sea Ice Pixel Proportion
SGD	Stochastic Gradient Descent
PSITRES	Polar Sea Ice Topography REconstruction System

1

Introduction

Contents

1.1 Motivation

In the machine learning field, the rapid development of Deep Neural Networks (DNNs) has yielded impressive results and promises further development. One area of rapid development is in computer vision and specifically, image segmentation. Deep Learning (DL) methods are able to avoid the traditional steps of semantic segmentation (manual feature definition, extraction, matching) and instead implement end-to-end learning [1]. These DL methods when applied to image segmentation have outperformed traditional algorithms across the board [2].

However, neural networks require large, labelled datasets for training. There are multiple available datasets for autonomous driving [3], pedestrian tracking [4], and object recognition [5]. For specific domain applications such as sea-ice segmentation, these datasets can be difficult to find. A significant amount of optical image datasets exist from icebreaker voyages, but the annotated counterparts are far fewer. Multiple publications [6] [7] remark on the lack of comprehensive datasets for sea ice segmentation and classification.

In many cases, neural networks that are used for specific domain knowledge studies are trained using transfer learning and custom datasets are created to meet the specific study's needs. A streamlined process to turn raw data and images into labelled ground truth data could drastically shorten the process of training DNNs and increase the amount of domain specific labelled images.

1.1.1 Environmental Factors

In the Arctic, sea ice is an important indicator for the state of climate change and Arctic well-being [8]. However, sea ice extent has been diminishing at a steady rate as seen in Figure 1.1. This vanishing of sea ice plays a pivotal role in enhancing further Arctic warming due to many effects, the most well-known being surface albedo feedback. Surface warming melts sea ice, which has a high albedo value and reflects significant amounts of heat from the sun back into space, replacing it with sea water. The sea water is less reflective and absorbs more heat than the sea ice, increasing the rate of warming [9]. This effect causes a positive feedback loop that continues to diminish Arctic sea ice year over year.

Due to the increasing loss in sea ice extent, monitoring the remaining sea ice year over year becomes more important to map the effects of climate change on the Arctic. Remote sensing and other methods of satellite imagery offer wide coverage [11], but their results can be supplemented and validated by shipboard observations. Closer perspectives can help not only segment between water and ice, but classify different types of ice such as new ice, grey ice, first-year ice, and multi-year ice [6] [12]. The age distinction within sea ice is important as older ice tends to be thicker and more resilient to changes in atmospheric and ocean forcings compared to younger ice. In 1985, 33% of the ice pack was made of ice older than four years, compared to 1.2% in March 2019 [13] as shown in Figure 1.2.

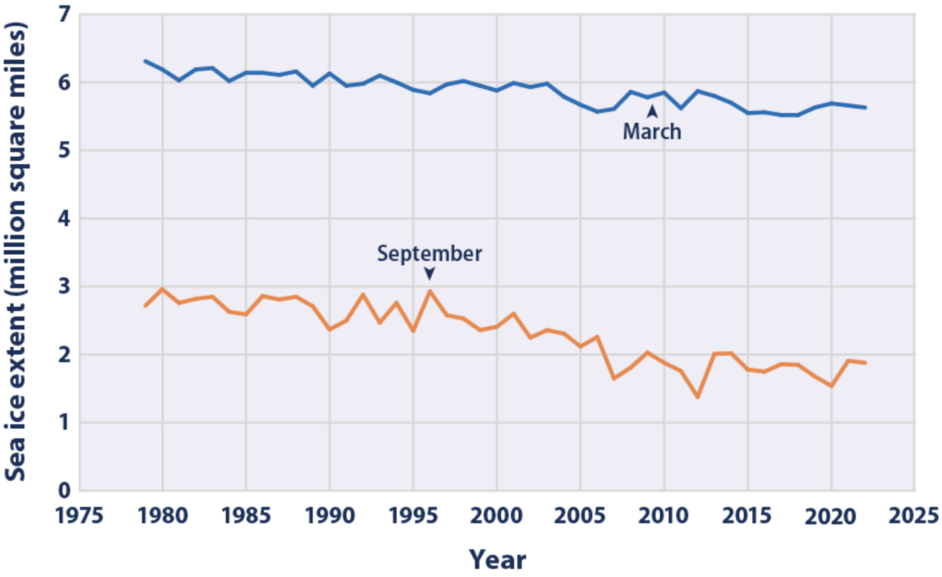


Figure 1.1: March and September Monthly Average Arctic Sea Ice Extent, 1979-2022 [10]

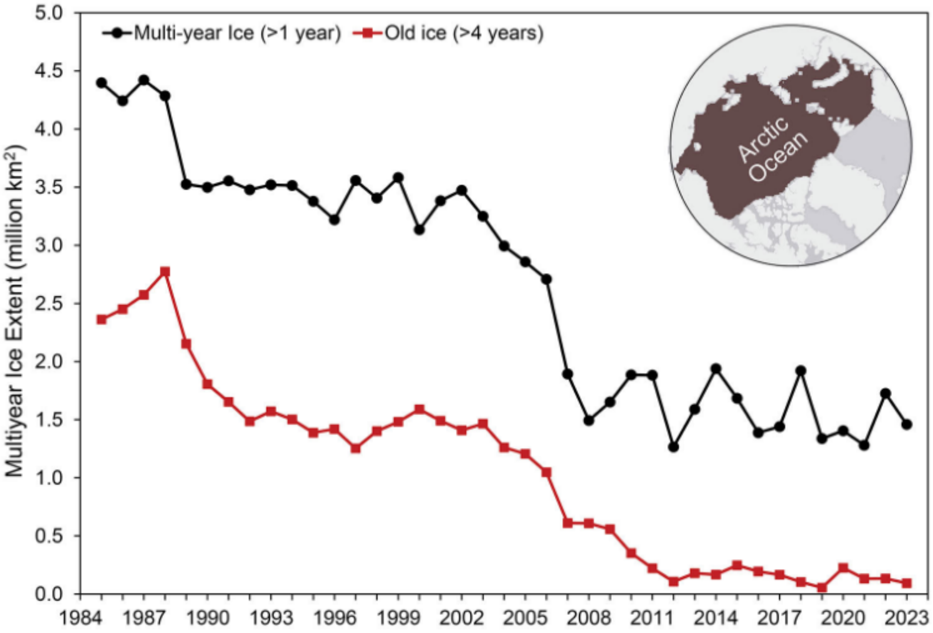


Figure 1.2: Extent of Multiyear Ice in the Arctic: Week of Minimum Total Extent, 1985-2023 [14]

1.1.2 Economic Factors

As Arctic sea ice continues to melt year over year, the effects have ceased to be purely environmental and now are branching into navigational and economic waters. A multi-modal ensemble model predicts that the Arctic could be seasonally ice-free as early as September 2055, depending on the scenario [15]. As sea ice year after year decreases, interest in the Arctic as a possible trading route increases. Whether using the Northwest Passage, the Transpolar Route, or the Northern Sea Route (NSR), these new pathways are between 30-50% shorter than the Suez or Panama canal routes [16].

With increasing numbers of ships entering the Arctic, navigation swells in importance, especially with the likely introduction of Marine Autonomous Surface Ships (MASS) in the future. Studies focused on using Light Detection and Ranging (LiDAR) to map near-ship sea ice have already shown promise for improving ships' situational awareness in ice fields [17] [18].

1.1.3 DigitalSeaIce Project

These factors have precipitated the development of the the DigitalSeaIce project, a partnership between Norway and China described as "a multi-scale integration and digitalization of Arctic sea ice observations and prediction models" [19]. Funded in part by The Research Council of Norway, the project pulls expertise from multiple universities aiming to improve ice forecasts in the Arctic with a focus on long-term variations and prediction.

The work for this thesis specifically falls under Work Package 2 (of five), entitled "Intelligent local-scale observations and analytics". The research question asked by this work package is essentially: "Can we develop autonomous, trustworthy, and time-efficient methods for monitoring and analysis of the large amounts of local-scale (100-500 m) in-situ sea ice data as required by work packages 1 and 4?" [19]. The context to which this work package fits into the larger DigitalSeaIce project is displayed in Figure 1.3

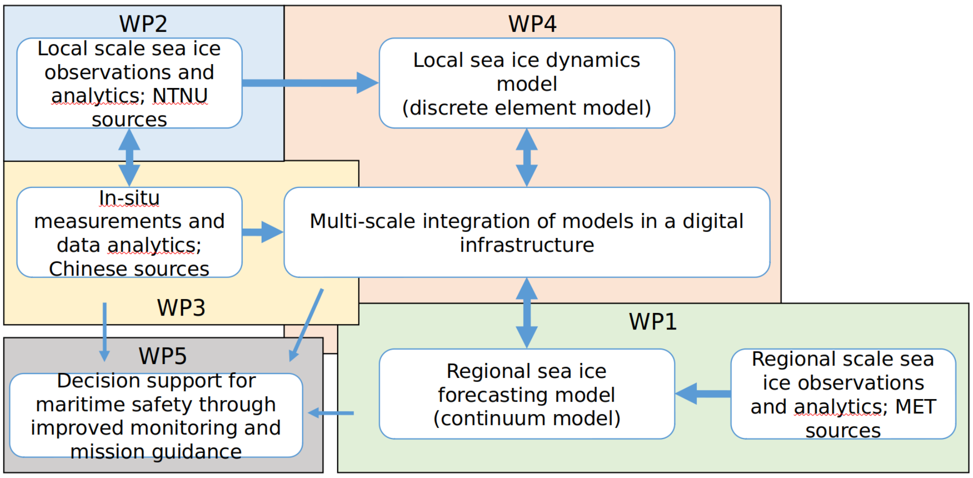

WP2
Local scale sea ice observations and analytics; NTNU sources
WP4
Local sea ice dynamics model (discrete element model)
In-situ measurements and data analytics; Chinese sources
Multi-scale integration of models in a digital infrastructure
WP3
WP1
WP5
Decision support for maritime safety through improved monitoring and mission guidance
Regional sea ice forecasting model (continuum model)
Regional scale sea ice observations and analytics; MET sources

Figure 1.3: DigitalSeaIce Work Package Relations [19]

1.2 Problem Definition

The problem presented in this work is the transformation of optical and LiDAR data into a dataset suited for training a deep neural network on the segmentation of water and ice. Using a payload designed by Oskar G. Veggeland containing a camera and LiDAR, raw data was captured during an icebreaker's voyage through the Arctic [20]. This apparatus captured continuous optical images and 3D point-cloud data of the sea ice as the ship traveled.

As sea ice is solid surface, it produces a high intensity point cloud in comparison to the sea water which tends to scatter LiDAR beams. Based on these assumptions, the returned point-cloud from the shipborne LiDAR array should consist only of sea ice. This point-cloud information can serve as a 'map' of the sea ice and can then be converted into 2D ground truth data for a deep neural network. The goal of this research endeavor is to determine if training a neural network using this automatically created ground truth data is feasible.

1.3 Challenges

The principal challenge for this research effort is the quality of the LiDAR data. Due to the differences in capture rates between the LiDAR and the optical camera, the 3D point-cloud often does not line up perfectly with the optical image. Specifically, there are differences in the capture rates of the LiDAR and optical camera. Multiple point clouds are sometimes attributed to a single optical frame, adding motion blur and distortion to the point clouds. Additionally, the returned point cloud is very sparse and does not accurately represent the edges of the sea ice floes. The sparsity of the LiDAR point cloud introduces large amounts of false negative pixels to the ground truth and increases the difficulty during training. These issues precipitated the need for preprocessing of the ground truth LiDAR data into three different datasets.

1.4 Aim & Scope

The aim of this thesis is to prove the feasibility of an automated data labelling pipeline to train a neural network in binary segmentation of water and sea ice. Work done by Oskar G. Veggeland converted the LiDAR point cloud into a 2D image taken from the point of view of the camera [20]. Various levels of preprocessing will be performed on the ground truth data and their effects investigated. A handful of different neural networks will then be trained using these datasets and their performances on a manually labelled dataset analyzed. Emphasis will be placed on keeping the process generalizable in the hopes that it can be applied to different camera configurations and weather conditions. Evaluation of the model will be performed with Intersection-over-Union (IoU), Dice score, pixel accuracy, precision, and recall.

1.5 Contribution

The main contribution of this thesis will be a proof-of-concept showing the feasibility of using LiDAR data to create an automated labelling pipeline for the supervised training of a neural network. This process will provide a method of acquiring labelled training data that avoids copious hours of manual labor. Ideally, the process can be applied to future voyages through the Arctic to create a more robust body of data on Arctic sea ice. The addition of the dataset to the public domain with future works should also help future research efforts related to the intersection of computer vision and Arctic sea ice. Hopefully, this process can assist scientists in their efforts to measure the effects of climate change and improve navigation for ships in the Arctic.

2

Background

Contents

2.1 Morphology

Mathematical morphology is a method within computer vision related to the shapes, sizes and other aspects of geometrical structures [21]. In the context of this thesis, morphology will be used to transform an image of the LiDAR point cloud into a binary ground truth mask usable by the deep neural network.

2.1.1 Erosion, Dilation, Opening, and Closing

Morphology can be broken down into two basic methods: erosion and dilation. In this case, the image will be assumed to be binary, i.e. back-scatter from LiDAR signifies a value of 1 (ice) with 0 occupying the rest of the image (water). A shape smaller than the image size is used to traverse the image, known as a structuring element [22]. Depending on if the structuring element fits, hits, or misses the desired pixels, an action will be performed. Generally, erosion removes pixels from objects bounded by the structuring element while dilation adds pixels [23].

Opening and Closing are two further methods that are combinations of erosion and dilation. Specifically, opening is erosion followed by dilation while closing is the reverse. Figure 2.1 demonstrates both processes on the same shape.

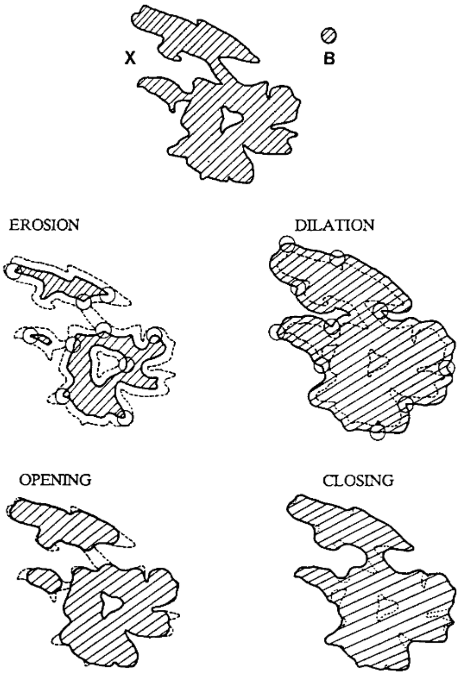


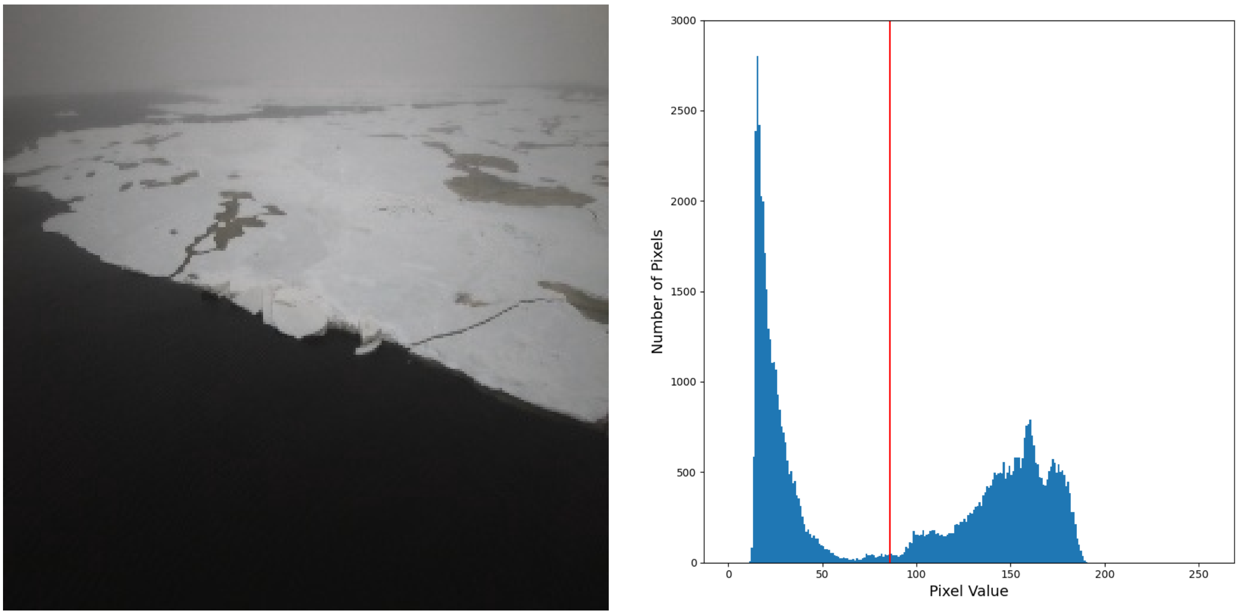
Figure 2.1: Example of Opening and Closing performed on the same shape [21]

In general, openings are used for removing small objects, protrusions, and thin connections between objects while closing removes small holes, smooths objects, and can fill gaps in the contour [24]. For this application, closing will be applied to the LiDAR point clouds to reduce sparsity.

2.2 Otsu's Method

In the context of this thesis, Otsu's method was used to perform automatic image thresholding on optical images to create a more representative ground truth dataset. Otsu's method is an automatic threshold selection algorithm for picture segmentation. At the highest level, the algorithm aims to find the maximum separability of two classes within an image.

In an ideal case, the histogram representing a gray image has two large peaks (representing the foreground and background) and the ideal threshold is the valley floor between the two peaks. This allows a segmentation of the fore and backgrounds of an image. However, in most images the peaks and valleys are not so distinct and are often imbued with noise. Otsu's thresholding method was designed to find this ideal 'valley' in any image. Figure 2.2a displays an image of sea ice while Figure 2.2b shows the grayscale image's histogram. The red line represents the pixel threshold chosen by Otsu's method to divide the image. The formulation of this method is summarized in the following section, as was published in [25].



(a) Original Image

(b) Gray Image Histogram

Figure 2.2: Otsu's Threshold Histogram
Red line represents Otsu's Threshold Value

2.2.1 Formulation

Otsu's method assumes an image is presented in L gray levels, in this case it is 256 bins for an 8-bit image. The number of pixels at any level i is denoted n_i and the total number of pixels becomes:

$$N = n_1 + n_2 + ... + n_L \tag{2.1}$$

The gray-level histogram is normalized and assumed to be a probability distribution:

$$p_i = \frac{n_i}{N}, \qquad p_i \geq 0, \quad \Sigma_{i=1}^{L} p_i = 1 \tag{2.2}$$

Then all pixels in the image are separated into two separate classes, C_0 and C_1 using a threshold value at level k. Class C_0 represents all pixels with intensities [1, .., k] whereas class C_1 denotes intensities [k+1, ..., L]. The equations for the probabilities and class mean levels are given below:

$$\omega_0 = \sum_{i=1}^{k} p_i = \omega(k) \tag{2.3}$$

$$\omega_1 = \sum_{i=k+1}^{L} p_i = 1 - \omega(k) \tag{2.4}$$

$$\mu_o = \frac{\sum_{i=1}^{k} i * p_i}{\omega_0} \tag{2.5}$$

$$\mu_1 = \frac{\sum_{i=k+1}^{L} i * p_i}{\omega_1} \tag{2.6}$$

Theoretically, the next step in this process is to find the within-class variance, however for a bi-modal histogram this is equivalent to finding the between-class variance, σ_B^2:

$$\sigma_B^2 = \omega_0 * \omega_1 * (\mu_0 - \mu_1)^2 \tag{2.7}$$

Thus, the optimal threshold k^* maximizes the between-class variance for all tested thresholds k:

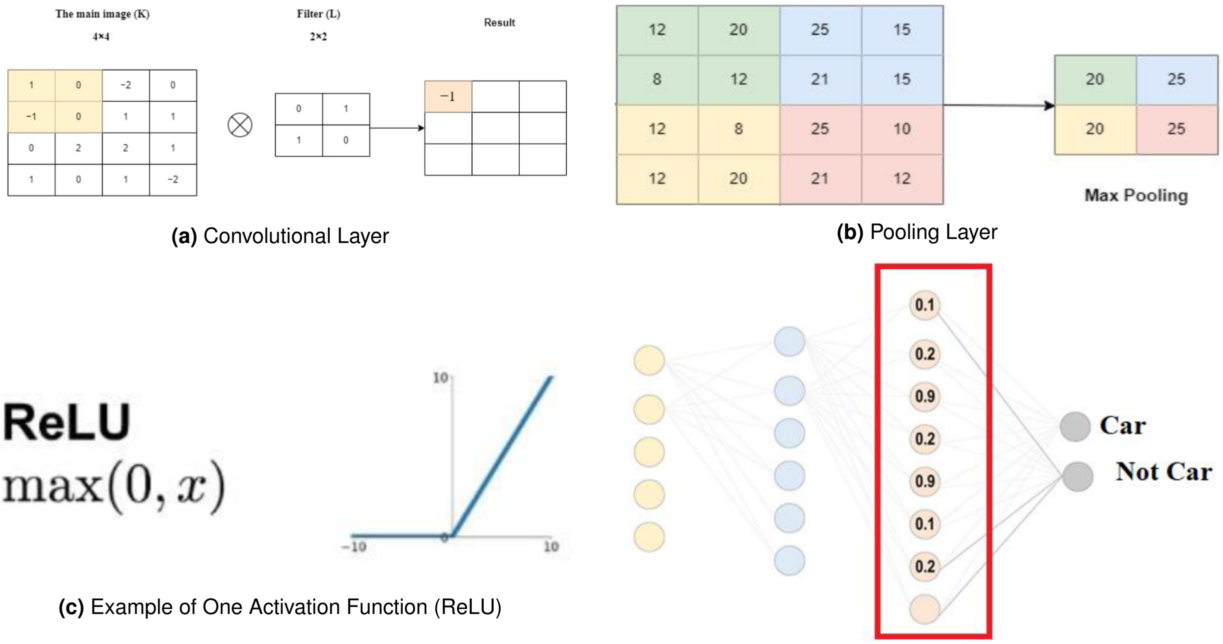
$$\sigma_B^2(k^*) = max_{(1 \leq k < L)} \sigma_B^2(k) \tag{2.8}$$

2.3 Deep Learning Models in Image Segmentation

Image segmentation algorithms have moved from traditional image processing methods such as thresholding, k-means clustering, and Markov random fields to now favoring deep learning models [2]. These deep learning models have shown impressive performance compared to these older models and have altered the course of development in the field. Within this broad category, there are a variety of different methods and network architectures.

2.3.1 Convolutional Neural Networks

A successful and often-used network architecture for deep learning in the field of computer vision is the Convolutional Neural Network (CNN). The key to the efficacy of these networks is their ability to extract features from input images. The basic structure of a CNN is composed of four layers: a convolutional layer, a pooling layer, an activation function, and a fully-connected layer [26]. In the convolutional layer, a weighted filter (known as a kernel) is passed over the input image. The weights are at first random and gradually learned during training. In the pooling layer, the dimensions of the feature map (created by the kernel in the previous step) are reduced while retaining as much information as possible. The activation function is essentially a test to determine whether or not a neuron should be fired based on the output of the pooling layer. Finally, the fully-connected layer connects all of the neurons determined by the activation function to those in the layers above and below [26]. Using a loss function and back-propagation, the network can learn which features correspond with their respective outputs and weight them to minimize the loss. Figure 2.3 displays diagrams representing each layer of a typical CNN.



(a) Convolutional Layer

(b) Pooling Layer

(c) Example of One Activation Function (ReLU)

(d) Fully-Connected Layer

Figure 2.3: Diagrams displaying each layer of a CNN [26]

2.3.2 Encoder-Decoder Architectures

Further development to convolutional methods is the addition of a decoder. In an encoder-decoder network architecture, an encoder compresses the input data into a representation in the latent space. Then the decoder network translates the latent space representation back into a prediction of the output [2]. The first widely-recognized neural network with an encoder-decoder architecture was the U-Net, developed in 2015.

The U-Net architecture consists of a contracting, downsampling path and an expanding, upsampling path. The combination of 3x3 convolutions, rectified linear units (ReLU), and 2x2 max pooling operations achieved state-of-the-art segmentation results at the time [27]. The architecture of this model is displayed below in Figure 2.4.

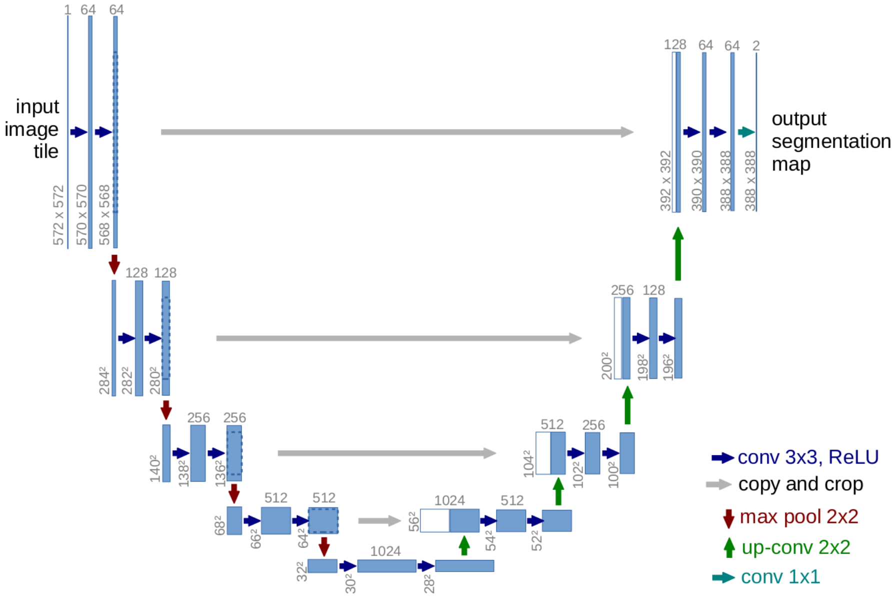


Figure 2.4: U-Net Model Architecture [27]

Another variation on the encoder-decoder network architecture was developed in 2017, called the Pyramid Scene Parsing Network (PSPNet). The goal of the pyramid scene parsing network was to avoid image segmentation errors that could be explained by context (e.g. a boat misclassified as a car despite being in the water). To improve the performance of scene parsing, the authors stacked multiple sub-regions with different sized receptive fields on top of the output feature map from an encoder. The novel method allowed for global contexts to be introduced to the existing feature maps and achieved 1st place performance at the time on datasets like PASCAL VOC 2012 and Cityscapes [28]. A diagram of this network is displayed below in Figure 2.5.

A further advancement along this developmental line is the latest iteration of the DeepLab family, DeepLabV3+. This updated version combines Spatial Pyramid Pooling (SPP) with an encoder-decoder

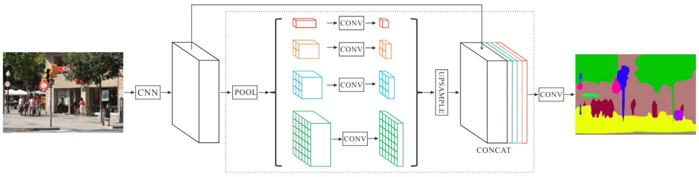


Figure 2.5: Pyramid Scene Parsing Network Architecture [28]

structure while applying atrous convolution to extract denser feature maps. The key difference from its previous iteration, DeepLabV3, was the addition of a decoder module that allows for detailed object boundary recovery. [29]. The architecture of this model can be seen in Figure 2.6.

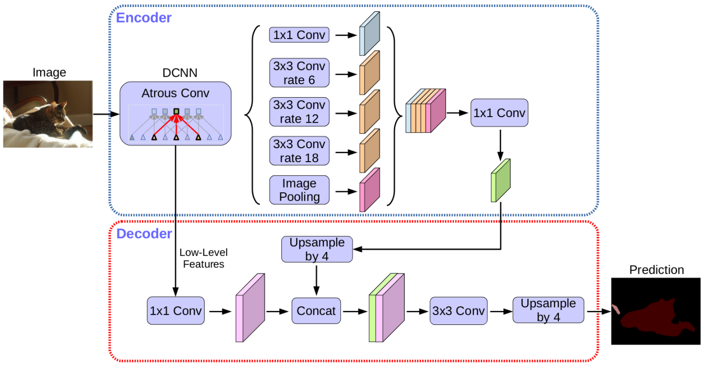


Figure 2.6: DeepLabV3+ Network Architecture [29]

2.3.2.A Atrous Convolution and Spatial Pyramid Pooling

Two of the methods mentioned previously that enabled increased levels of performance for these encoder-decoder structured models are explained in more detail below.

Atrous convolutions are a method that allows the user to manually adjust the filter's field-of-view [30]. The receptive field of the neural network is altered by inserting a certain amount of zeros between filter values [31]. The rate, r, changes the space between the weights of the kernel. Using this parameter, the size of the receptive field in the convolutional layer can be controlled. This means the filter can now look at larger areas of the input without a decrease in the spatial resolution or an increase in the kernel size [32]. This method is displayed in Figure 2.7 and is used in DeepLabV3+.

SPP is a method that removes the fixed size constraint of the network. In other words, with SPP the network can allow inputs of various sizes to be used within the same convolutional neural network. Before this development, convolutional neural networks required all images in a dataset to have the same size. If an image was not this size, it would either need to be cropped or warped to fit the input requirement and as a result, valuable data would be lost. By creating different spatial bins that are then pooled

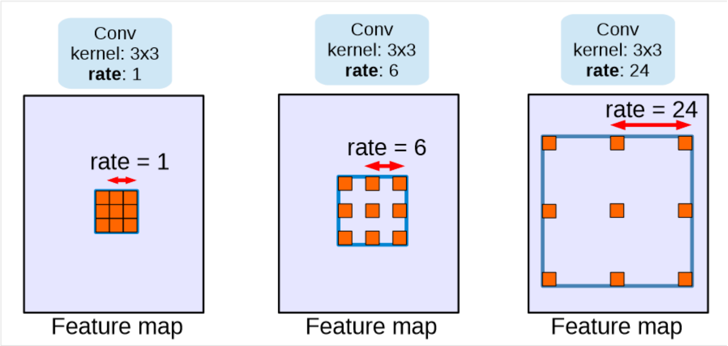


Figure 2.7: Atrous Convolution with kernel size 3x3 and differing rates [30]

in a single layer, images of any size can be input into a CNN [33]. Figure 2.8 shows the SPP architecture. In addition to enabling any physical size image to function as an input to a convolutional neural network, SPP allows networks to encode multi-scale contextual information to better inform predictions [29]. A visual representation of SPP is shown in Figure 2.8.

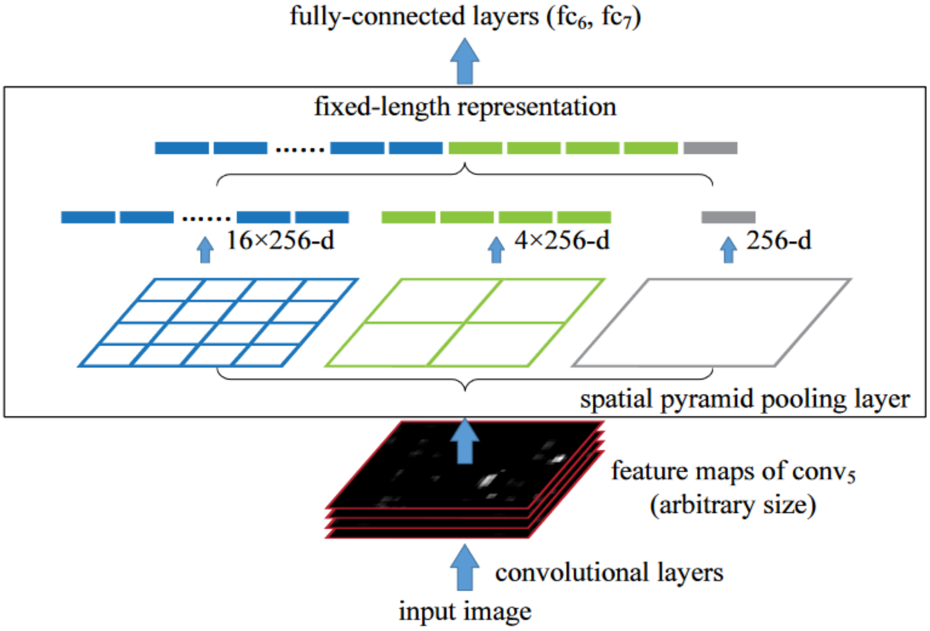

fully-connected layers (fc6, fc7)
fixed-length representation
16×256-d
4×256-d
256-d
spatial pyramid pooling layer
feature maps of conv5
(arbitrary size)
convolutional layers
input image

Figure 2.8: Diagram Demonstrating Spatial Pyramid Pooling in a CNN [33]

2.3.3 Transformers in Computer Vision

While the majority of neural network architectures used for computer vision are convolutional in nature, vision transformers are growing in popularity. Since the introduction of the classical Vision Transformer (ViT), the use of these transformers for classification, detection, segmentation, and compression has risen dramatically.

Transformers are typically used for Natural Language Processing (NLP), where words are converted to embeddings and are encoded according to their position in a sequence. The same concept is applied for ViT using images; an image is split into fixed-size patches, linearly embed, positionally encoded, and fed into the transformer. An image of the original Vision Transformer architecture is seen in Figure 2.9.

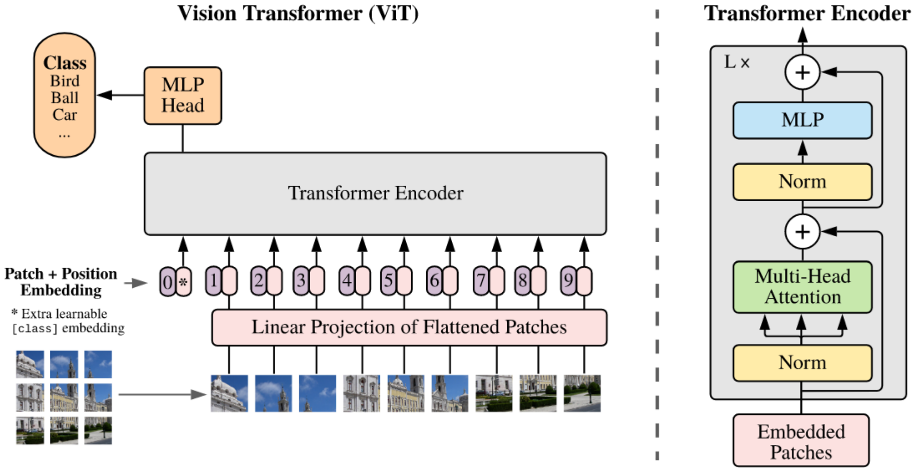


Figure 2.9: Vision Transformer Network Architecture [34]

In their comprehensive survey, Jamil *et al.* found that visual transformers are succeeding in classification and detection because of self-attention mechanisms and effective transfer learning but suffer from high computational costs, large training datasets, and lack of interpretability [35]. Tranformers' need for pre-training on large datasets arises as transformer architectures do not inherently encode inductive biases for visual data, unlike convolutional architectures which can encode prior image knowledge [36]. As for computational complexity, the cost of core self-attention in transformers increases at a quadratic rate with the number of patches therefore increasing the barrier to using these transformers for higher resolution tasks in object detection and image segmentation [36]. In general, transformers offer a very promising avenue in the field of computer vision but suffer from significant drawbacks. As a result of these hindrances, focus has been placed on making transformers' architectures more efficient. SegFormer is a simple, efficient yet powerful semantic segmentation model that combines a transformer encoder with MultiLayer Perceptron (MLP) decoders [37]. Its framework is displayed in Figure 2.10.

SegFormer benefits from a transformer encoder that generates a hierarchical feature representation with a focus on multi-level features without using Positional Embedding, lightening the computational load. The decoder only consists of MLP layers, keeping complexity low while maintaining an large Effective Receptive Field (ERF) [37]. For this thesis, SegFormer is used as a point of comparison against more traditional convolutional image segmentation methods as it represents a state-of-the-art vision transformer.

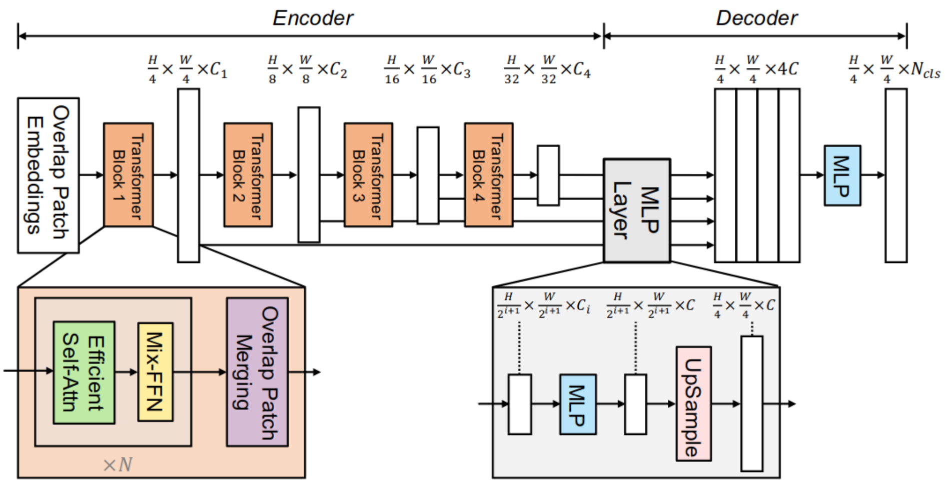

Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{cls}$
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
Efficient Self-Attn
Mix-FFN
Overlap Patch Merging
$\times N$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C$
$\frac{H}{4}\times\frac{W}{4}\times C$
MLP
UpSample

Figure 2.10: SegFormer Network Architecture [37]

2.4 Pretraining and Transfer Learning

A key prerequisite for high-performing neural networks is sufficient quantities of labelled training data. In supervised learning, it is difficult for a model to perform well if it is not trained with adequate ground truth data. To circumvent this problem, a model can be pretrained on data not specifically meant for the task at hand and then fine-tuned on applicable data in a general process known as transfer learning. Typically in pretraining, a network architecture is trained on a large dataset such as ImageNet [38] to teach a model general image features. The weights for each layer are kept to further improve upon with fine-tuning on a smaller, task-specific dataset [39]. When there is a lack of representative training data, transfer learning has been shown to greatly outperform common supervised learning methods [40].

2.5 Data Augmentation

One main goal of improving deep convolutional networks in computer vision tasks is increasing their generalizability. This refers to a model's ability to perform on information not previously seen before. Some models tend to overfit, meaning they become too attuned to the training data and struggle to make inferences on unseen testing data. One effective method of combating overfitting is data augmentation.

Augmented datasets represent a more comprehensive set of possible data points, shrinking the distance between training and validation datasets along with unseen testing sets [39]. Typical data augmentations for computer vision tasks include cropping, flipping, color space alterations, rotations, translations, and noise injections. In general, data augmentation artificially increases the size of the training dataset, allowing the model to become more generalizable especially when working with smaller datasets.

2.6 LiDAR Field-of-View Mask

As LiDAR measures light that is emitted and then back-scattered, there is a distance where no meaningful information will be returned. In the images used for this thesis, the distance occurs close to the horizon. In order to avoid the model classifying the sky water, a mask was applied to each image. This mask designated the area where information was returned for the LiDAR and was computed by Oskar G. Veggeland [20]. Loss was only back-propagated from areas inside the mask, or where the LiDAR was 'active'.

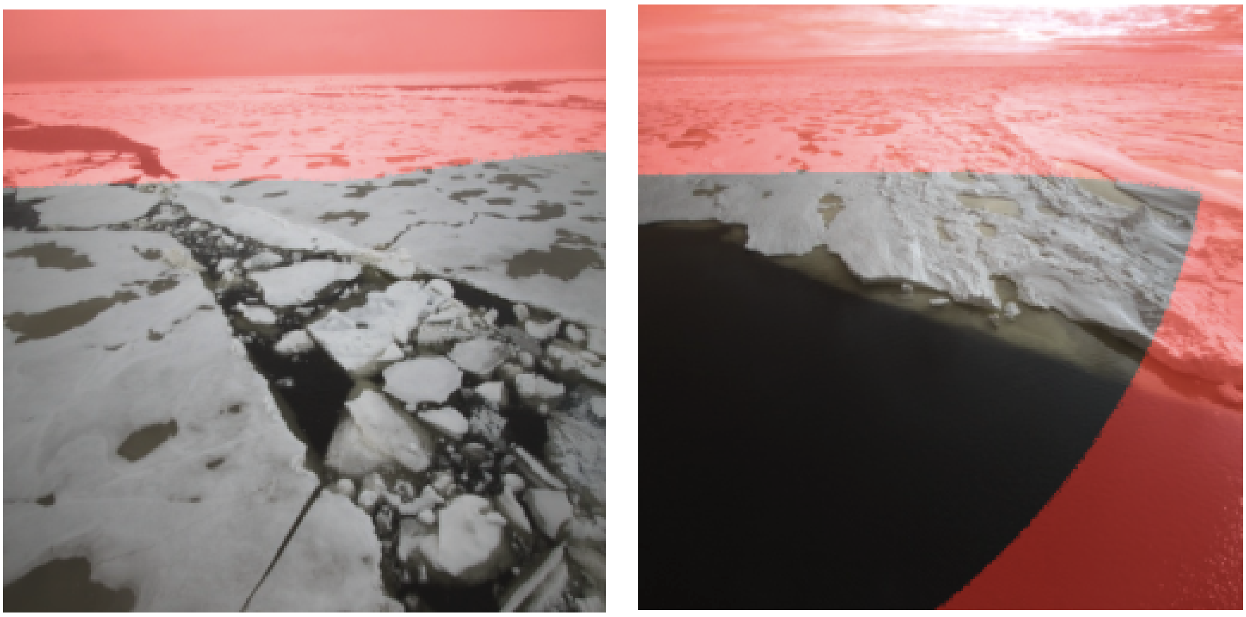


Figure 2.11: Optical Images with areas outside the LiDAR Field-of-View (FOV) Mask Highlighted in Red

Figure 2.11 displays two examples of optical images with areas outside the mask overlaid and highlighted. The mask usually removes areas in the distance and the sky, but occasionally clips areas towards the start and end of recording.

2.7 Evaluation

There are a variety of ways to evaluate the performance of image segmentation models. In this thesis, the Intersection-over-Union, pixel accuracy, precision, recall, and Dice score will be used. These metrics were chosen based off of previous studies in image segmentation on sea ice [1] [12] [41].

All measures of performance rely on the following definitions related to pixel classification.

- TP (True Positive): the pixel was correctly predicted as ice.
- TN (True Negative): the pixel was correctly predicted to be water.

- FP (False Positive): the pixel was incorrectly predicted to be ice.

- FN (False Negative): the pixel was incorrectly predicted to be water.

2.7.1 Intersection-over-Union

The Intersection-over-Union measures the ability of a model to correctly classify an object or shape. The metric measures the overlap between the model predicted object boundary and the actual ground truth. Equation 2.9 shows the method of calculating IoU assuming *A* represents the ground truth prediction while *B* represents a prediction. Figure 2.12 displays a conceptual diagram.

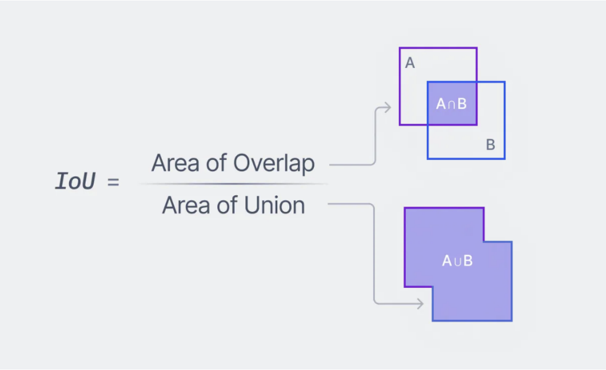


Figure 2.12: Visualization of Intersection over Union [42]

$$\text{IoU} = \frac{A \cap B}{A \cup B} = \frac{TP}{TP + FP + FN} \tag{2.9}$$

2.7.2 Pixel Accuracy

The pixel accuracy is a simple metric designed to provide an overall view of the model's performance. It's definition is simply the total number of correctly predicted pixels over the total number of predicted pixels. It is displayed in Equation 2.10.

$$\text{Pixel Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

2.7.3 Dice Score

The Dice score is another method of measuring the similarity of a predicted mask with its ground truth. For example, to assess the similarity of a ground truth mask, A, and its prediction, B, the equation can be defined as:

$$\text{DICE} = \frac{2 * \mid A \cap B \mid}{\mid A \mid + \mid B \mid} = \frac{2 * TP}{2 * TP + FP + FN} \tag{2.11}$$

A Dice score of 0 indicates zero similarity whilst a value of 1 indicates perfect overlap.

2.7.4 Precision

Precision aims to measure the percentage of positive pixels predicted accurately. It is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.12}$$

In a binary segmentation task, a high precision means that if a pixel is predicted to be positive it is likely that it is correct. On the other hand, low precision indicates that the positive class (ice) is over-predicted.

2.7.5 Recall

Recall is a measure of the percentage of the ground truth positive pixels that were predicted correctly and is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.13}$$

A high recall value in indicates that the model is correctly predicting most of the ground truth positive pixels. If recall is low, the model is missing object pixels and is over-predicting the background class (water).

2.7.6 Sea Ice Pixel Proportion (SIPP)

For the purposes of visualizing model performance on different types of images, sea ice pixel proportion was incorporated into the evaluation metrics. In the context of this thesis, SIPP is defined as the ratio of ice pixels within an image. Only the pixels within the LiDAR FOV mask are considered.

$$\text{SIPP} = \frac{N_{\text{ice}}}{N_{\text{total}}} \tag{2.14}$$

where N_{ice} denotes the number of pixels classified as ice and N_{total} is the total number of pixels within the LiDAR FOV mask.

2.8 Loss Function

The loss function chosen for the training of these neural networks was the Binary Cross-Entropy (BCE) Loss. BCE Loss is commonly used in binary classification tasks where a given sample should either be assigned 0 or 1. As it is a logarithmic loss, the value will increase exponentially the closer the predicted probability of the true class gets to 0. On the other hand, if a sample has a high probability of being correctly predicted as the true class, the value will be very low. As bad predictions are be penalized heavily compared to good ones, this serves as an acceptable metric to train models. The formulation is displayed in Equation 2.15.

$$\text{BCE} = \frac{-1}{N}\sum_{i=1}^{N} y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i)) \tag{2.15}$$

where y_i represents the class and $log(p(y_i))$ represents the probability of that class occurring. For this thesis, the loss will be calculated with logits instead of probabilities to avoid numerical instabilities, as shown in Equation 2.16.

$$\text{BCEWithLogits} = \frac{1}{N}\sum_{i=1}^{N} \left(\log\left(1 + e^{-z_i}\right) - y_i z_i\right) \tag{2.16}$$

where N is the number of samples, z_i is the logit from the model, and y_i is the true label (0 or 1).

3

State-of-the-Art

Contents

3.1 Deep Learning for Image Segmentation

While image segmentation using deep learning is a growing field, it is also one that has been studied extensively. In the last 10 years, rapid development has been performed in the form of different models and network architectures. Figure 3.1 displays a timeline of deep learning segmentation algorithms on 2D images from 2014 to 2020.

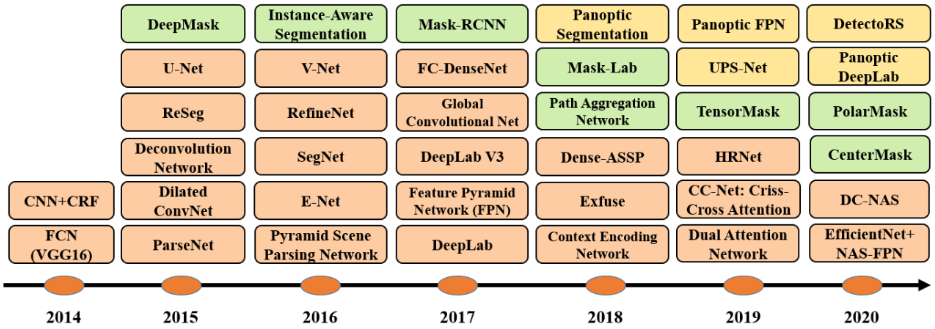


Figure 3.1: Timeline of Deep Learning Segmentation Algorithms, 2014-2020 [2]

Minaee *et al.* [2] measured the performance of the above algorithms on a variety of datasets. Table 3.1 below displays eight different models and their respective performances on the PASCAL-VOC dataset. It should be mentioned the asterisk indicates the network has been pre-trained on a different dataset. DeepLabV3+ scores high compared to previous convolutional methods and thus was chosen as one of the models for this study.

Method	**Backbone**	**mIoU**
FCN	VGG-16	62.2
RefineNet	ResNet-152	84.2
PSPNet	ResNet-101	85.4
DeeplabV3	ResNet-101	85.7
PSANet	ResNet-101	85.7
DeeplabV3+	Xception-71	87.8
EMANet	ResNet-152	88.2
DeeplabV3+*	Xception-71	89.0

Table 3.1: Image Segmentation Network Performances on PASCAL-VOC dataset [2]

In recent years, deep neural networks have been used for image segmentation in a variety of fields with success. When studying leaf spot disease in sugar beets, Adem *et al.* [43] were able to achieve a classification rate of 96.47% using Yolov4 with image processing on a dataset of 1040 images. In the medical field, image segmentation is commonly used to identify tumors and other aberrations in medical images. Gite *et al.* [44] utilized a U-Net architecture in lung segmentation with X-ray images to produce 98% segmentation accuracy and a mIoU score of 0.95.

Recently, transformers are being adapted to segmentation tasks and producing impressive results. The development of transformer models has been pushed forward by their self-attention which allows capture of 'long-term' dependencies between sequence events and their large scale pre-training combined with subsequent fine-tuning [36]. Thisanke *et al.* [45] conducted a state-of-the-art survey of vision transformers in the field of semantic segmentation. A summary of the relevant results is below in Table 3.2.

				Datasets	
Model	**Variant**	**Backbone**	**# Params (M)**	**ADE20K**	**Cityscapes**
SETR	SETR-MLA (16,160k)	ViT-L	310.57	48.64	
	SETR-PUP(16,80k)	ViT-L	318.31		79.34
Swin		Swin-L	234	53.5	
Segmenter		ViT-L	307	53.63	81.3
PVT	PVT v1	PVT-Large	65.1	44.8	
	PVT v2	PVT v2-B5	85.7	48.7	
HRFormer	OCRNet(7,150k)	HRFormer-B	50.3	46.3	
	OCRNet(15,80k)	HRFormer-B	50.3		81.9
Mask2Former		Swin-L-FaPN		**56.4**	
		Swin-B			**83.3**
SegFormer		MiT-B0	**3.4**	37.4	76.2
		MiT-B1	13.1	42.2	78.5
		MiT-B2	24.2	46.5	81
		MiT-B3	44	49.4	81.7
		MiT-B4	60.8	50.3	82.3
		MiT-B5	81.4	51	82.4

Table 3.2: State-of-the-Art Survey of Transformers in Semantic Segmentation [45]

While SegFormer does not represent the absolute best performer on either the ADE20K or the Cityscapes dataset, it maintains competitive performance with impressively low parameters. This justified its usage in this study as the models were trained on a laptop GPU. The performance of SegFormer compared to well-known convolutional methods on the same two datasets is displayed in Table 3.3. Despite fewer parameters, SegFormer maintains competitive performance compared to PSPNet and DeepLabV3+ on both datasets.

				ADE20K			Cityscapes	
Method	**Encoder**	**# Params (M)**	**Flops**	**FPS**	**mIOU**	**Flops**	**FPS**	**mIoU**
FCN	MobileNetV2	9.8	39.6	**64.4**	19.7	317.1	14.2	61.5
	ResNet-101	68.6	275.7	14.8	41.4	2203.3	1.2	76.6
PSPNet	MobileNetV2	13.7	52.9	57.7	29.6	423.4	11.2	70.2
	ResNet-101	68.1	256.4	15.3	44.4	2048.9	1.2	78.5
DeepLabV3+	MobileNetV2	15.4	69.4	43.1	34.0	555.4	8.4	75.2
	ResNet-101	62.7	255.1	14.1	44.1	2032.3	1.2	80.9
SETR	ViT-Large	318.3		5.4	50.2		0.5	82.2
SegFormer	MiT-B0	3.8	**8.4**	50.5	37.4	**125.5**	**15.2**	76.2
	MiT-B4	64.1	95.7	15.4	51.1	1240.6	3.0	83.8
	MiT-B5	84.7	183.3	9.8	**51.8**	1447.6	2.5	**84.0**

Table 3.3: SegFormer Compared to Traditional Methods in Semantic Segmentation [37]

3.2 Sea Ice Segmentation

With the development of deep learning models for image segmentation, the models have been increasingly applied to sea ice in a variety of ways. While impressive work has been done on segmenting sea ice from remote sensing information [46] [47], for the purpose of brevity the state-of-the-art was focused on shipborne optical studies.

3.2.1 Shipborne Optical Sensors

3.2.1.A Segmentation and Classification

As computer vision techniques have evolved, their usages for segmentation and object classification in the Arctic have grown. Most studies focus on the segmentation and classification of multiple ice features from optical cameras aboard icebreakers. Zhang *et al.* [41] developed a novel neural network architecture coined ”Ice-Deeplab” for multi-class sea ice segmentation. This model modifies the existing DeepLabV3+ architecture by adding a Convolution Block Attention Module (CBAM) and a modified decoder structure to achieve 1.5% higher mIoU and 3.1% better sea ice mIoU performance over the basleine DeepLabV3+ model. In a similar study, Li *et al.* [7] modified the DeepLabV3+ model architecture for sea ice detection aboard icebreakers. Other studies have combined studies on segmentation and classification. Dowden *et al.* [6] [48] achieved impressive results both segmenting and classifying ice features on a dataset captured from the *Nathaniel B. Palmer*. Kim *et al.* [49] both identified and located multiple ice objects within images aboard two icebreakers. Balasooriya *et al.* [1] compared PSPNet101 to DeepLabV3 in the semantic segmentation of sea ice and discovered an average inference speed of 0.08s for DeepLabV3 and 1.9s for PSPNet101 while maintaining similar performance (90.21 percent mIoU vs 90.1, respectively). By creating an ensemble of multiple networks (PSPNet, PSPDenseNet, DeepLabV3+, UPerNet), Panchi *et al.* was able to significantly outperform PSPNet in classification of 14 different classes [12].

As a part of this experiment, Panchi *et al.* investigated the model's ability to differentiate between ice and non-ice objects. The following values in Table 3.4 were computed using the mean ensemble approach and postprocessing. While this does not provide a 1-to-1 comparison as Panchi's model utilized manually labelled training data and postprocessing, it serves as a benchmark for the performances of the models in this study.

Dataset	**mIoU**	**Accuracy**	**F1 Score**
Avg. of 5-fold cross-validation	0.933	0.957	0.965
Clear Test	0.948	0.963	0.973
Grayscale Test	0.946	0.961	0.972
Vignette Test	0.758	0.829	0.862

Table 3.4: Binary Sea Ice Segmentation Results for Ensemble Network [12]

As weather conditions in the Arctic are variable and often result in image distortion, studies have focused on remedying the issue. Panchi *et al.* [50] investigated using deep learning as a 'de-weathering' algorithm to improve segmentation and classification performances. Pederson *et al.* [51] explored ice object classification with added distortion designed to emulate poor weather conditions. Focus has also been placed on the comparison of ice object classifications by a neural network compared to a human sea ice expert [52].

3.2.1.B Mapping & Awareness

Since cameras have been placed on ships navigating the Arctic, efforts have been made to study the impact they could have on mapping and detection sea ice. Sandru *et al.* [11] present a beginning-to-end analysis for shipborne sea-ice fields consisting of the capture of images, image pre-processing, orthorectification of the image, and identification of floes using K-means and dynamic thresholding algorithms. Sandru *et al.* [17] has further investigated this field using LiDAR sensors aboard the icebreaker *S.A. Agulhas II* to map and navigate through ice fields. Utilizing a 3D camera system known as the Polar Sea Ice Topography REconstruction System (PSITRES), Sorenson *et al.* [53] was able to create high-resolution 3D reconstructions of surrounding ice features. Further utilizing convolutional neural networks, the team could identify and segment features such as algae, meltponds, and polar bear prints on the ice. Finally, Veggeland *et al.* [20] utilized a system with both 3D LiDAR and optical cameras to create maps of surrounding sea ice with RGB values attributed to the 3D point cloud. These maps capture sea ice roughness and significant ice features.

3.3 Image Segmentation with Sparse Ground Truth Data

As fully annotated datasets are time-consuming to create, scientists have investigated strategies to emulate dense ground truth data with existing data. Often, multi-modal methods can provide incomplete or sparse data that can be modified to imitate a manually labelled dataset. Maggiolo *et al.* investigated this by purposely degrading dense ground truth data using morphological operations. The goal was to determine the change in performance on remote sensing data if ground truth data could be 'scribbled' instead of densely annotated. The authors compared results using a few different models (including their own CI-FC-CRF) and found that absolute differences in precision revealed roughly an average of 15% performance drop compared to the densely labeled ground truth [54].

These methods are very important in specific domain applications where fully annotated datasets are rare. In the medical field, Li *et al.* sought to minimize time spent annotating ground truth segmentations of Computed Tomography (CT) scans. CT scans contain multiple 'slices', each of which represent an image as the subject moves through the machine. At the end, the scans can be compiled into a 3D representation. The authors investigated two different ways of creating pseudo ground truth based on increasingly sparse manually labeled CT slices. The authors found that up to 95% of the manual workload could be eliminated without a significant sacrifice in accuracy [55]. In an application similar to this thesis, Alonso *et al.* aimed to improve binary segmentation of corals with sparse ground truth information. The authors experimented with multiple ways of augmenting the existing sparse ground truth to gain a better representation of the coral mask. The team utilized superpixel segmentation methods to increase the resolution of the ground truth masks and incorporated multi-modal information in the form of fluorescence channels. Combining these methods, the authors were able to improve segmentation results using a fine-tuned SegNet model and the augmented ground truth [56]. An example of their methods of ground truth augmentation is seen in Figure 3.2.

3.4 Existing Datasets

For the purposes of image segmentation in general, there are many datasets. 2D datasets such as PASCAL-VOC, Cityscapes, MS COCO, and Siftflow are often used for measuring object detection and image segmentation performance [2]. In addition, datasets like KITTI and nuScenes contain both images and 3D LiDAR point clouds to train networks on 3D segmentation [57].

For the purposes of segmenting images of sea ice, datasets are often captured from ships as they move through the Arctic. Zhang, Jin *et al.* [58] used an annotated 814 image dataset of river ice gathered for the purposes of their study using an Unmanned Aerial Vehicle (UAV). Zhang *et al.* [41] utilized a 320-image dataset captured from a Chinese ice-strengthened cargo ship, *Tian'en*, on a voyage through the Arctic. Dowden *et al.* [6] created two datasets of 1090 labeled images and 240 labeled images from

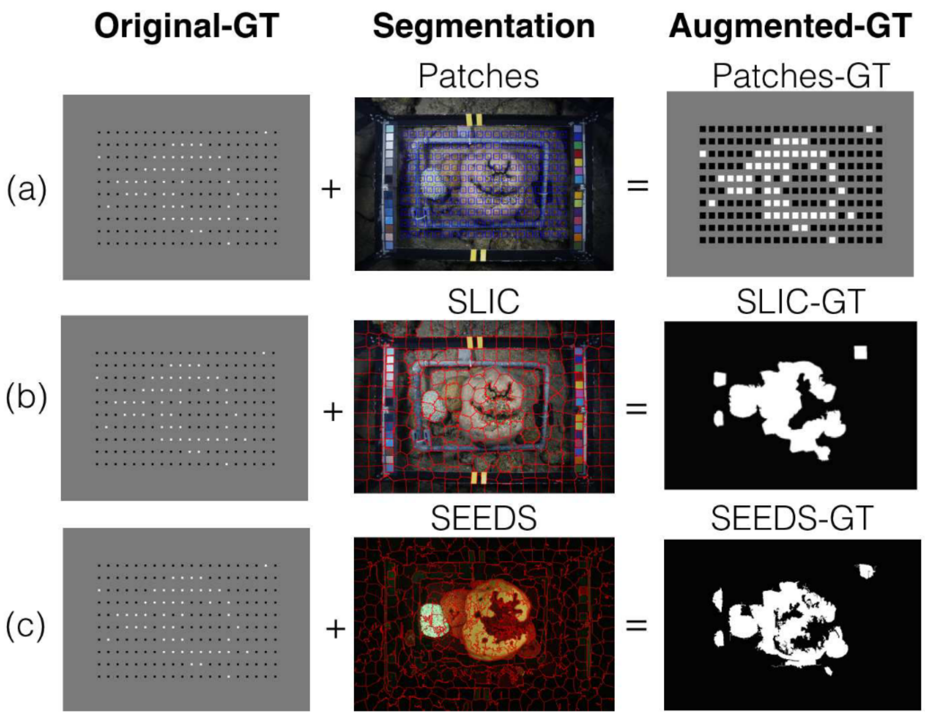
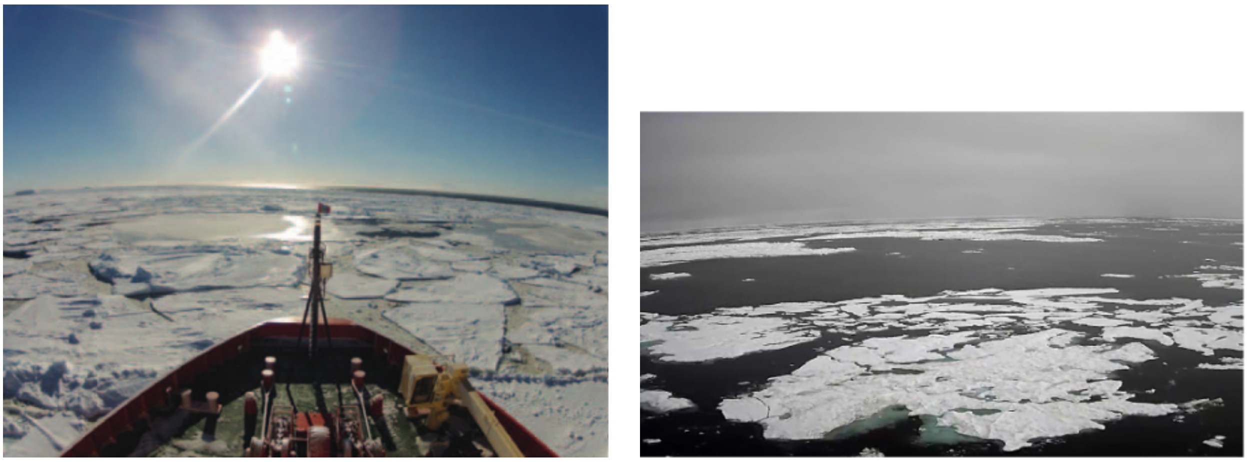


Figure 3.2: Coral Segmentationi Methods with Sparse Ground Truth: (a) Patch Method around existing GT pixels; SLIC Superpixel method; SEEDS Superpixel method [56]

two months of footage on the icebreaker *Nathaniel B. Palmer* as it completed an Antarctic expedition. Similarly, Li *et al.* [7] utilized a training dataset of about 1300 images taken from footage aboard the *Nathaniel B. Palmer*. Kim *et al.* [49] used a 370 image dataset for classification and a 390 image dataset for segmentation assembled from images taken aboard the US Coast Guard icebreaker *Healy* and the nuclear-powered icebreaker *50 Let Pobedy*.

Panchi *et al.* created a new dataset for their study with 338 unique images collected from Google, Yandex, and Baidu along with 37 additional images from the RV Lance during a research cruise in the Fram Strait [12]. In a similar method, Pederson *et al.* [51] assembled a dataset of 738 images from Google, Yandex, publicly available image streams from icebreakers, and private pictures. To map the surrounding ice fields using LiDAR and optical images, Veggeland *et al.* [20] utilized a 120-second trajectory taken from aboard the *Kronprins Haakon*. Sorenson *et al.* [53] utilized PSITRES to capture over 8 million images across three voyages and has made the dataset publicly available (although not all images are labelled) [59]. For the de-weathering algorithm developed in [50], Panchi *et al.* created the first open-source ice image dataset with both clean and weather degraded images [60].

Image datasets tend to differ due to the angle and location of cameras placed onboard the icebreakers. Two images from the voyages of the *Nathaniel B. Palmer* and *Tian'en* are included in Figure 3.3. While optical segmentation and LiDAR mapping have been implemented on sea ice, to the best of the



(a) Image from *Nathaniel B. Palmer* Cruise [6]

(b) Image from *Tian'en* Cruise [41]

Figure 3.3: Dataset Images from Sea Ice Segmentation Studies

author's knowledge there is no existing dataset that contains both images and LiDAR point cloud data of sea ice.

3.5 Literature Gaps

While there are investigations into image segmentation using sparse datasets, these are fewer and dedicated to specific use cases. Furthermore, the reasons why the data are sparse and the strategies to combat this differ in most papers. Specifically, a method of automated labelling using LiDAR point cloud data in the Arctic has not yet been pursued to the author's knowledge. However, the segmentation and classification of sea ice is a rapidly growing field as more focus is placed on the shrinking Arctic ice sheet.

4

Methodology

Contents

4.1 Dataset Creation

Using a payload containing an optical camera and LiDAR designed by Oskar G. Veggeland, data were captured during a cruise through the Arctic. The optical camera used for this dataset was a FLIR blackfly GigE optical camera with a resolution of 1440x1080. Coupled with this was a Mid-70 LiDAR from Livox. The payload was placed on the prow of the Norwegian icebreaker *Kronprins Haakon* on its voyage in the summer of 2023 [20]. Images of the apparatus and its placement are displayed in Figure 4.1, while the architecture of the payload is shown in Figure 4.2.

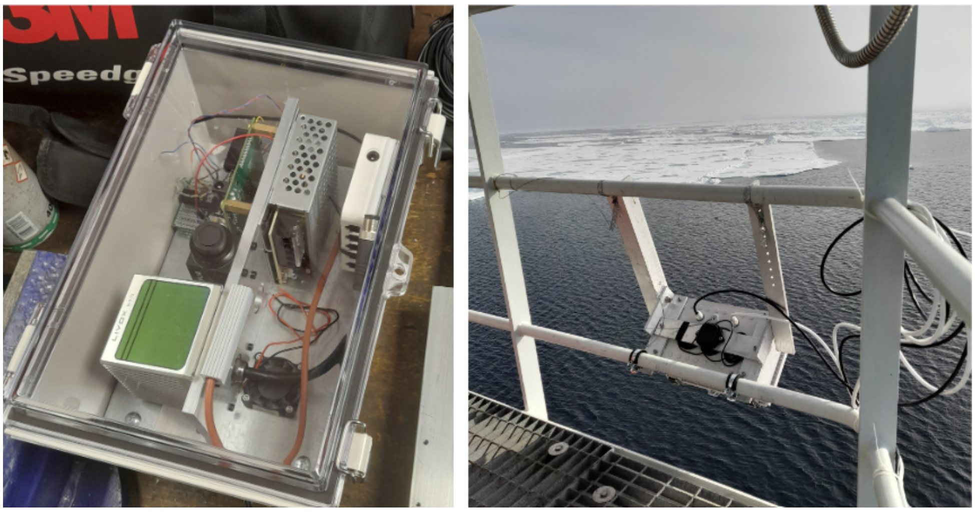


Figure 4.1: Camera-LiDAR Payload Designed by Oskar G. Veggeland [20]

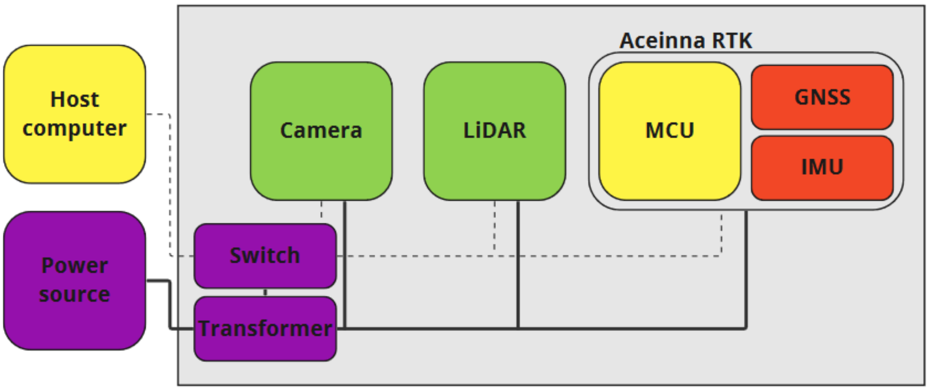
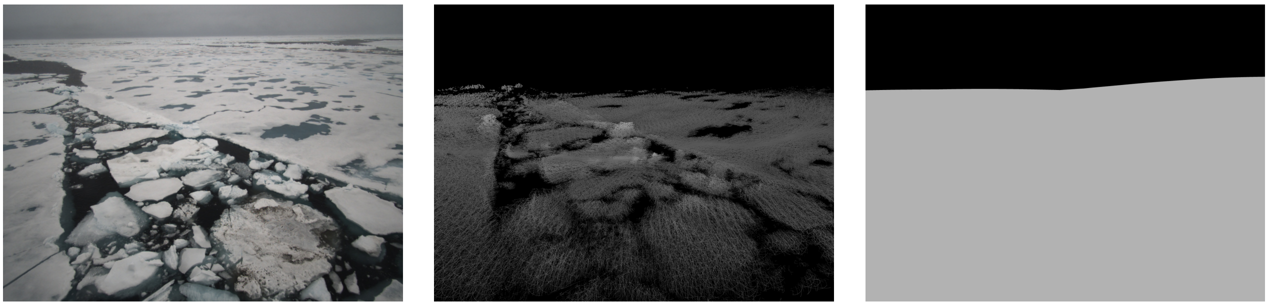


Figure 4.2: Architecture of Camera-LiDAR Payload [20]

4.2 Dataset Preprocessing

After the dataset was accumulated and the LiDAR point cloud translated to the camera's perspective, the LiDAR point clouds were processed in three different ways. One example of an optical image, the accompanying LiDAR point cloud, and the LiDAR FOV mask before any data processing is shown in Figure 4.3.



(a) Optical Image

(b) Unprocessed LiDAR Point Cloud

(c) LiDAR FOV Mask

Figure 4.3: Example of an Unprocessed LiDAR Point Cloud
The gray area in 4.3c represents the 'active' zone of the LiDAR

As seen in Figure 4.3, the LiDAR point cloud represents the sea ice as seen in the optical image relatively well. There are some features that can cause problems, such as the sparsity of the point cloud closer to the LiDAR apparatus and the classification of some meltponds as water. The size for all images and masks was chosen to be 256 x 256 pixels, as this is compatible with the SegFormer model architecture, is computationally light, and has precedence in the literature [49]. The data preprocessing was performed on the original image size of 1430 x 1063 pixels before being resized to 256 x 256 pixels using OpenCV's resize function with linear interpolation.

4.2.1 Raw Dataset

The Raw dataset represents the most basic preprocessing that is be done to create an ice mask from the LiDAR point cloud. This process forms the basis for the following two methods and is based on the assumption that if the LiDAR receives a point in the back-scatter, there is ice at that point. Therefore, if there is a LiDAR point (no matter the intensity) it is converted to an 'ice' pixel for the binary ground truth mask. The formulation is as follows.

Let $I(x, y)$ represent the intensity of a pixel at position (x, y) in the grayscale image of the LiDAR point cloud. A simple thresholding operation is applied, where each pixel is transformed based on its intensity.

The thresholded image $I_T(x, y)$ is defined as:

$$I_T(x,y) = \begin{cases} 1, & \text{if } I(x,y) > 0 \quad (4.1a) \\ 0, & \text{otherwise} \quad (4.1b) \end{cases}$$

Thus, if the pixel value $I(x, y)$ is greater than 0, it is set to 1 ('ice'). Otherwise, it is set to 0 ('water').

4.2.2 Morphological Dataset

The second dataset is based on the binary threshold implemented in the Raw dataset with the addition of one iteration of the morphological process closing. This step was taken to fill the holes that are present in the raw LiDAR point cloud. The kernel used is a rectangle with a 3x3 pixel size. This process involves two steps:

1. **Dilation**: The binary thresholded image's white regions are expanded by checking each pixel's 3x3 neighborhood. If any pixel in the neighborhood is white (1), the center pixel is set to white as well.

Mathematically, the dilation $D(I_T)(x, y)$ is:

$$D(I_T)(x,y) = \max\{I_T(x+i, y+j) \mid -1 \leq i, j \leq 1\} \tag{4.2}$$

Thus, the maximum value of the pixels in a 3x3 neighborhood around (x, y) is taken.

2. **Erosion**: After dilation, the white regions are shrunk by again checking each pixel's 3x3 neighborhood. Now, the center pixel is set to white only if all pixels in the neighborhood have a value of 1.

The erosion $E(D(I_T))(x, y)$ is:

$$E(D(I_T))(x,y) = \min\{D(I_T)(x+i, y+j) \mid -1 \leq i, j \leq 1\} \tag{4.3}$$

This takes the minimum value in the 3x3 neighborhood.

The final image after applying both dilation and erosion (the closing operation) is denoted as:

$$I_C(x,y) = E(D(I_T))(x,y). \tag{4.4}$$

4.2.3 Otsu-Hybrid Method

The third and most complex of the datasets was created to improve the quality of the ground truth label. This method is a combination of the binary threshold employed in the Raw dataset and an image threshold performed on the grayscale version of the input image with Otsu's binarization. The process is as follows.

A binary mask $M_{\text{topo}}(x, y)$ is first created from the LiDAR point cloud, similar to the Raw Dataset:

$$M_{\text{topo}}(x, y) = \begin{cases} 1, & \text{if topo}(x, y) > 0, \quad (4.5a) \\ 0, & \text{otherwise.} \quad (4.5b) \end{cases}$$

Applying Otsu's thresholding method on the grayscale input image gray_image(x, y) to return a binary image $M_{\text{otsu}}(x, y)$:

$$M_{\text{otsu}}(x, y) = \begin{cases} 1, & \text{if gray\_image}(x, y) > T_{\text{otsu}}, \quad (4.6a) \\ 0, & \text{otherwise,} \quad (4.6b) \end{cases}$$

where T_{otsu} is the threshold computed using Otsu's method.

A combined mask $M_{\text{ice}}(x, y)$ is then created by combining $M_{\text{topo}}(x, y)$ and $M_{\text{otsu}}(x, y)$. This is done by checking if both masks have a value of 1 at the same pixel location:

$$M_{\text{ice}}(x, y) = \begin{cases} 1, & \text{if } M_{\text{topo}}(x, y) + M_{\text{otsu}}(x, y) > 1, \quad (4.7a) \\ 0, & \text{otherwise.} \quad (4.7b) \end{cases}$$

Finally, the same morphological closing operation seen in the Morphological Dataset is applied:

The final closed binary mask $M_{\text{closed}}(x, y)$ is:

$$M_{\text{closed}}(x, y) = E(D(M_{\text{ice}}))(x, y) \quad (4.8)$$

Where E and D represent the erosion and dilation steps as a part of the closing operation.

This method aims to take advantage of the best aspects of each mask. The LiDAR point cloud is impervious to issues that can affect image thresholding like brightness, contrast, and blurriness. The image thresholding often results in well-defined object boundaries that the motion blur in the LiDAR point cloud lacks. Combined, the result is not perfect but tends to keep some of the positive aspects of each method. Figure 4.4 displays a good example of the strengths of this method. The thresholding operation defines the object boundaries but misclassifies the sky as ice, while the final result in the hybrid method removes the false positives.

Three examples of an optical image and the three preprocessing methods are displayed in Figure 4.5.

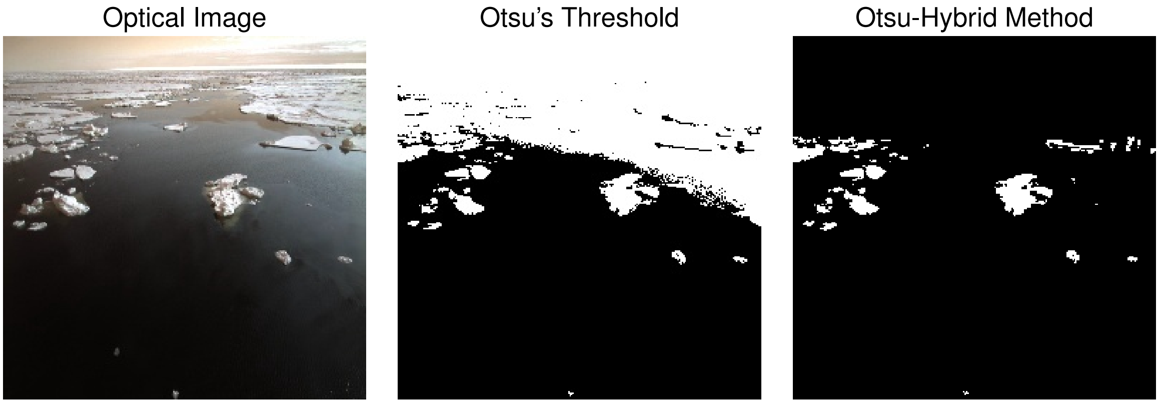


Figure 4.4: Otsu-Hybrid Method Example

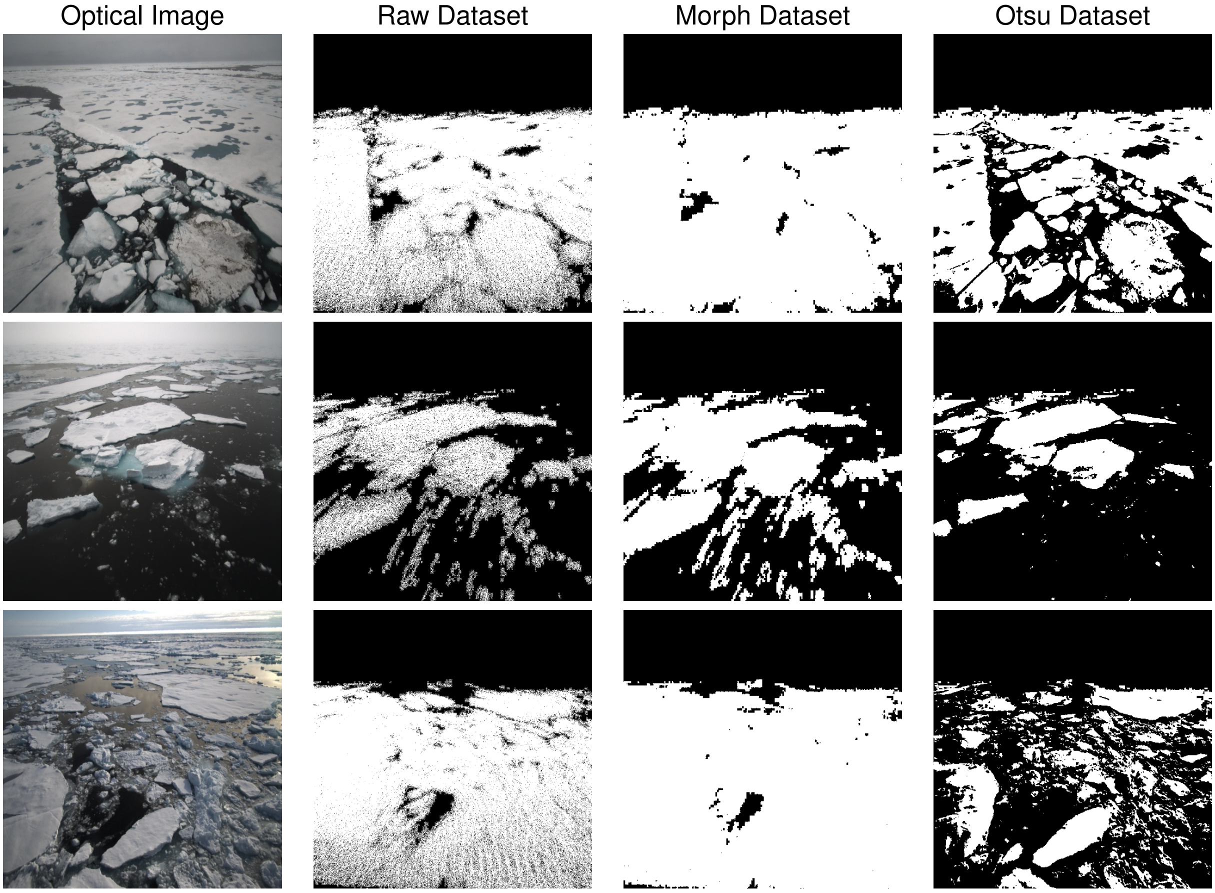
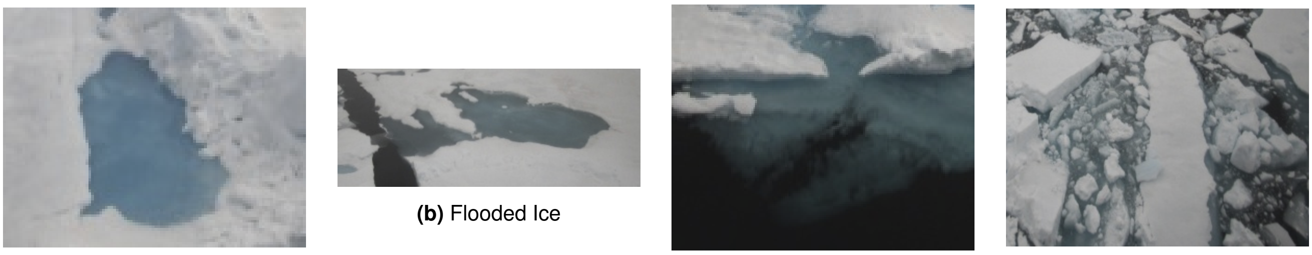


Figure 4.5: Dataset Preprocessing Examples

4.2.4 Manually Labelled Ground Truth

To provide a measure of comparison for the three datasets described, a manually labelled test set was assembled from two sources. A colleague working on similar research in computer vision, Nabil Panchi, provided a dataset of 186 labelled optical images, referred to as the GoNorth Dataset. These images are from the same cruise as the dataset used for this thesis and thus can be integrated to measure the performance of the preprocessing methods. The images are cropped, but still provided a valuable measuring point to evaluate how close the automated labelling processes come to the manually-labelled ground truth. The author of this paper additionally labelled a 175 image subset using the online software Roboflow, from here on referred to as the Roboflow Dataset.

The manually labelling process presented a few challenges in the classifications of ice features such as meltponds, flooded ice, undersea ice, and sea ice rubble. Ideally, these features should occupy separate classes but in this binary segmentation task they had to be delineated as either ice or water. For the purposes of this thesis, any feature that was not open water was labelled as ice. Figure 4.6 contains examples of the aforementioned ice features in the dataset. It is important to mention that even within binary classification, there are differences in labelling methods.



(a) Meltpond

(b) Flooded Ice

(c) Undersea Ice

(d) Ice Rubble

Figure 4.6: Manually Labelled Ice Features

4.2.5 Class Imbalances

Analyzing the class balances in any machine learning task is important, as unbalanced classes are likely to lead to incorrect inferences from a model. Table 4.1 displays the percentages of positive and negative pixels within the LiDAR FOV mask for each dataset.

There exists a majority of ice pixels for the Morphological dataset, while the sparsity of the Raw dataset and the image thresholding method of the Otsu dataset decreases the ratio of positive pixels. Both manually labelled datasets contain higher percentages of ice pixels than the Raw and Otsu datasets.

Dataset	Ice Pixels (%)	Water Pixels (%)
Raw	0.42	0.58
Morphological	0.61	0.39
Otsu	0.41	0.59
goNorth	0.51	0.49
Roboflow	0.59	0.41

Table 4.1: Class Imbalances

4.3 Dataset Split

As all of the images used are from the same research cruise, they are taken from the same angle aboard the ship with generally similar weather conditions. The images are taken frequently, meaning one image and the ones immediately adjacent in time are extremely similar. If the entire dataset is split randomly, it is likely that similar images could be present in both the training and testing sets. This can lead to data leakage, artificially improving test set performance.

4.3.1 Train-Validation-Test

One method of avoiding data leakage was splitting the training, validation, and test sets by trajectory. During the cruise, the payload was not recording for the entire time. Instead, the data were split into different 'chunks' representing different times when the payload was active, called trajectories. To promote the least data leakage possible, images from the same trajectory are not present in both the training and testing sets. A table with each trajectory and number of images, along with the distinction for each split, is displayed in Table 4.2.

This resulted in a total dataset of 2,111 images with 1,464 in the training set, 317 in the validation set, and 330 in the testing set. This is roughly a 70,15,15 split (69.35% train, 15.02% validation, 15.63% test). Table 4.3 aims to analyze the split based on the average sea ice pixel proportion.

As the Raw dataset is unmodified (no 'ice' pixels added), it is likely the best representation for the average amount of sea ice present in each dataset. It is clear the datasets are not perfectly balanced as the SIPP decreases from train to validation to test.

4.3.2 Data Augmentation

Data Augmentation was performed using the Albumentations package in Python. The training augmentations are listed in Table 4.4.

The VerticalFlip augmentation is added to prevent the model from disregarding the top section of

Trajectory Number	Number of Images	Split
21	98	train
22	20	validation
23	184	train
24	113	train
26	112	train
27	6	train
28	2	train
29	11	test
30	53	train
31	211	test
40	116	train
49	178	validation
72	260	train
73	37	train
75	108	test
76	180	train
77	5	train
78	3	train
79	9	train
80	12	train
81	215	train
82	59	train
91	119	validation

Table 4.2: Trajectory Numbers and Associated Dataset Split

			SIPP	
Split	# of Images	Raw	Morph	Otsu
train	1464	0.49	0.69	0.45
val	317	0.34	0.39	0.29
test	330	0.22	0.45	0.36

Table 4.3: Dataset Split SIPP Values

Augmentation	Probability	Description
HorizontalFlip	0.5	Randomly flips the sample horizontally
VerticalFlip	0.2	Randomly flips the sample vertically
RandomBrightnessContrast	0.3	Randomly changes the brightness and contrast, for exposure to different lighting conditions
RandomToneCurve	0.2	Randomly changes relationship between light and dark areas of the image by adjusting the tone curve
RandomResizedCrop	0.2	Randomly crops a portion of an image and resizes to original image size

Table 4.4: Training Data Augmentations

an image as it is normally masked out in the loss function by the LiDAR FOV. The brightness contrast and tone curve augmentations are efforts to increase the models' robustness to weather and lighting conditions which are variable in the Arctic. Finally, the resized crop is designed to give the models experience on different sized ice features.

4.4 Models

The three models used for this study were the U-Net [27], DeepLabV3+ [29], and the SegFormer visual transformer [37]. In each of these cases, the models were modified to have a single output class. The U-Net and DeepLabV3+ models were imported via *Segmentation Models Pytorch*. This package allows customization in the encoder, encoder weights, input channels, and output classes. For the sake of consistency in the U-Net and DeepLabV3+ models, a ResNet101 encoder [61] with pretrained ImageNet [38] weights was used along with three input channels and one output class. *Segmentation Models Pytorch* enables the customization in output classes by implementing a *SegmentationHead* module that contains one *Conv2d* layer convolving the output to the desired number of classes.

For SegFormer, the simplest version (MiT-B0 encoder) was used as it has the fewest parameters and thus the lowest memory usage. This SegFormer model was loaded from the *HuggingFace* repository, with num_labels parameters set to 1. In this case, the hierarchical transformer was pretrained on the ImageNet-1k dataset [38], then the decoder head was added and the model fine-tuned on the ADE20K dataset [62] at a 512x512 resolution [63]. A feature of the SegFormer model is an output size $\frac{1}{4}$ the size of the input. Thus, the output logits were upsampled from a size of 64x64 pixels to the input resolution of 256x256 pixels using the *torch.nn.functional.interpolate* function. This ensures proper loss calculation as the labels are also of size 256x256 pixels.

4.4.1 Loss Function

The loss function used for the training of these models was the BCE Loss with logits. An important part of this loss function was the addition of the LiDAR FOV mask. This FOV mask allows the model to be trained only on areas of the image where the LiDAR is 'active' and avoid class imbalance and incorrect labels. For implementation in code, the BCE loss was calculated and then masked prior to back-propagation. Implementation is displayed below, where $\mathbf{l}$ represents the logits (predicted values), $\mathbf{y}$ represents the true labels, $\mathcal{L}(\mathbf{l}, \mathbf{y})$ signifies the loss function (binary cross-entropy), and $\mathbf{m} \in \{0, 1\}$ the mask.

First, the loss is computed for each element of the batch:

$$\mathbf{L} = \mathcal{L}(\mathbf{l}, \mathbf{y}) \tag{4.9}$$

Then the LiDAR FOV mask is applied to the loss to ensure that only elements where the mask $\mathbf{m} = 1$ contribute to the loss:

$$\mathbf{L}_{\text{masked}} = \mathbf{L} \cdot \mathbf{m} \tag{4.10}$$

Finally, the mean loss is calculated then back-propagated, ensuring that the sum of the loss is normalized by the number of valid mask elements:

$$\text{mean\_loss} = \frac{\sum \mathbf{L}_{\text{masked}}}{\sum \mathbf{m}} \tag{4.11}$$

After the loss was back-propagated, a sigmoid function was applied to the output logits to calculate the probabilities. Then a threshold of 0.5 was applied to the output of the sigmoid to compute the predicted ice mask.

4.5 Training and Evaluation

PyTorch was utilized for training the models. A NVIDIA RTX 3060 Laptop GPU was the main device used for training equipped with CUDA 11.2. The Stochastic Gradient Descent (SGD) optimizer was utilized along with a learning rate scheduler, *ReduceLROnPlateau*.

The metrics used for training of the models can be found in Table 4.5.

For purposes of evaluating the training and performance of the models, metrics were recorded during training and test set evaluations. For the training of the models, BCE Loss, IoU, and Dice score were recorded and averaged across each epoch for both the training and validation sets. For the evaluation of the test set, BCE Loss, IoU, Dice score, pixel accuracy, precision, recall, and SIPP (of both label and prediction) were recorded for each sample. Each model was first trained on each of the three

Number of Epochs	30
Batch Size	16
Initial Learning Rate	$5x10^{-3}$
Criterion	BCEWithLogitsLoss(reduction='none')
Optimizer	SGD
Optimizer Momentum	0.9
Scheduler	ReduceLROnPlateau
Scheduler Factor	0.5
Scheduler Patience	5 epochs
Scheduler Metric	Validation Loss

Table 4.5: Training Parameters

preprocessed datasets. Then each trained model was evaluated on the manually labelled ground truth dataset to see if any of the preprocessed datasets could approximate the manually labelled equivalent.

5

Experiments, Results & Discussion

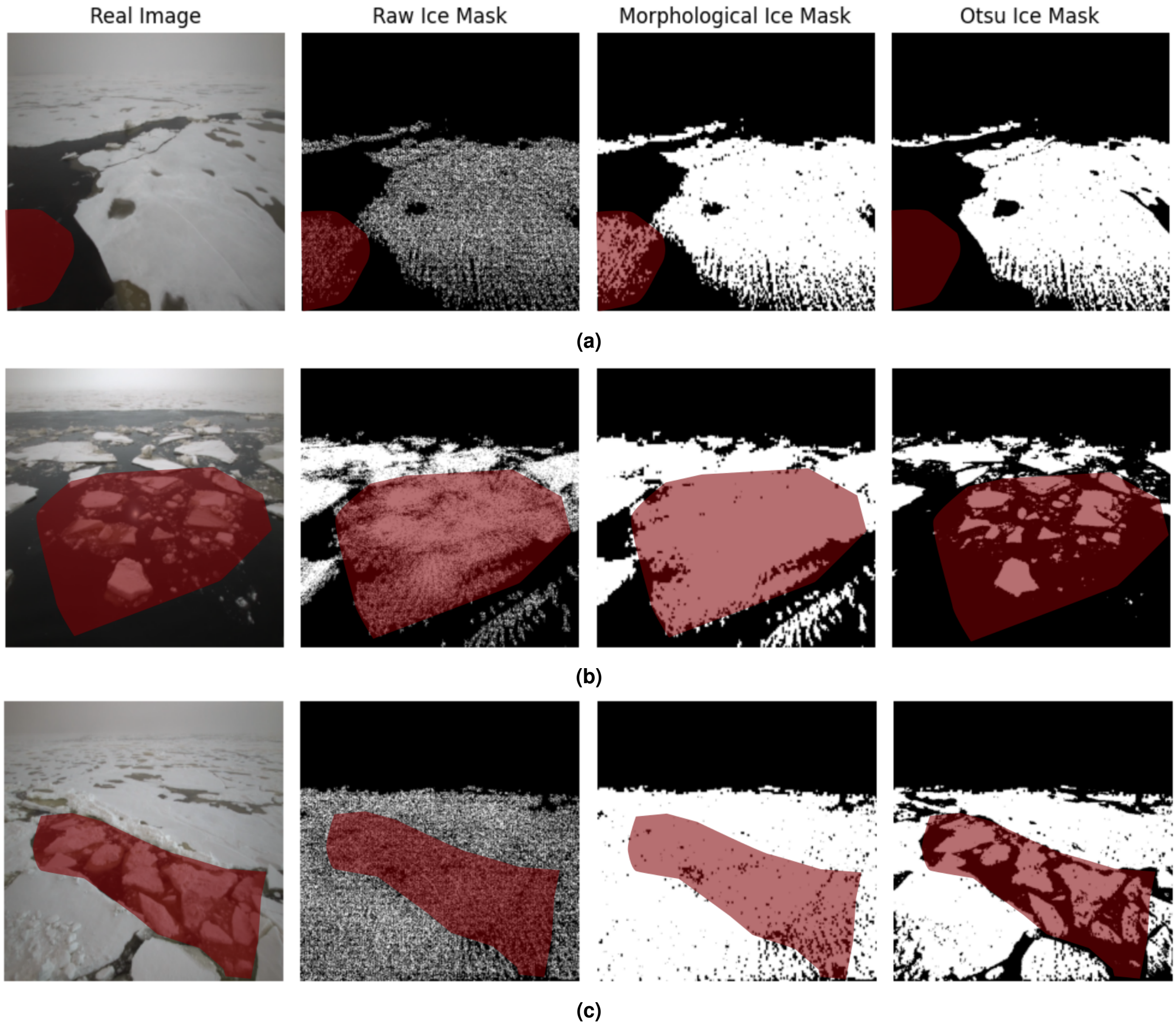
Contents

5.1 Evaluation of Preprocessed Datasets

5.1.1 Raw and Morphological Datasets

The Raw dataset is the most basic set of preprocessing done to the returned LiDAR point cloud. As the Morphological dataset is the same as the Raw dataset with the addition of one iteration of closing, the errors are largely the same in both sets.

Figure 5.1 shows 3 examples of the datasets where the shortcomings are most evident.



(a)

(b)

(c)

Figure 5.1: Raw & Morphological Dataset Shortcomings
Areas highlighted in red represent dataset shortcomings

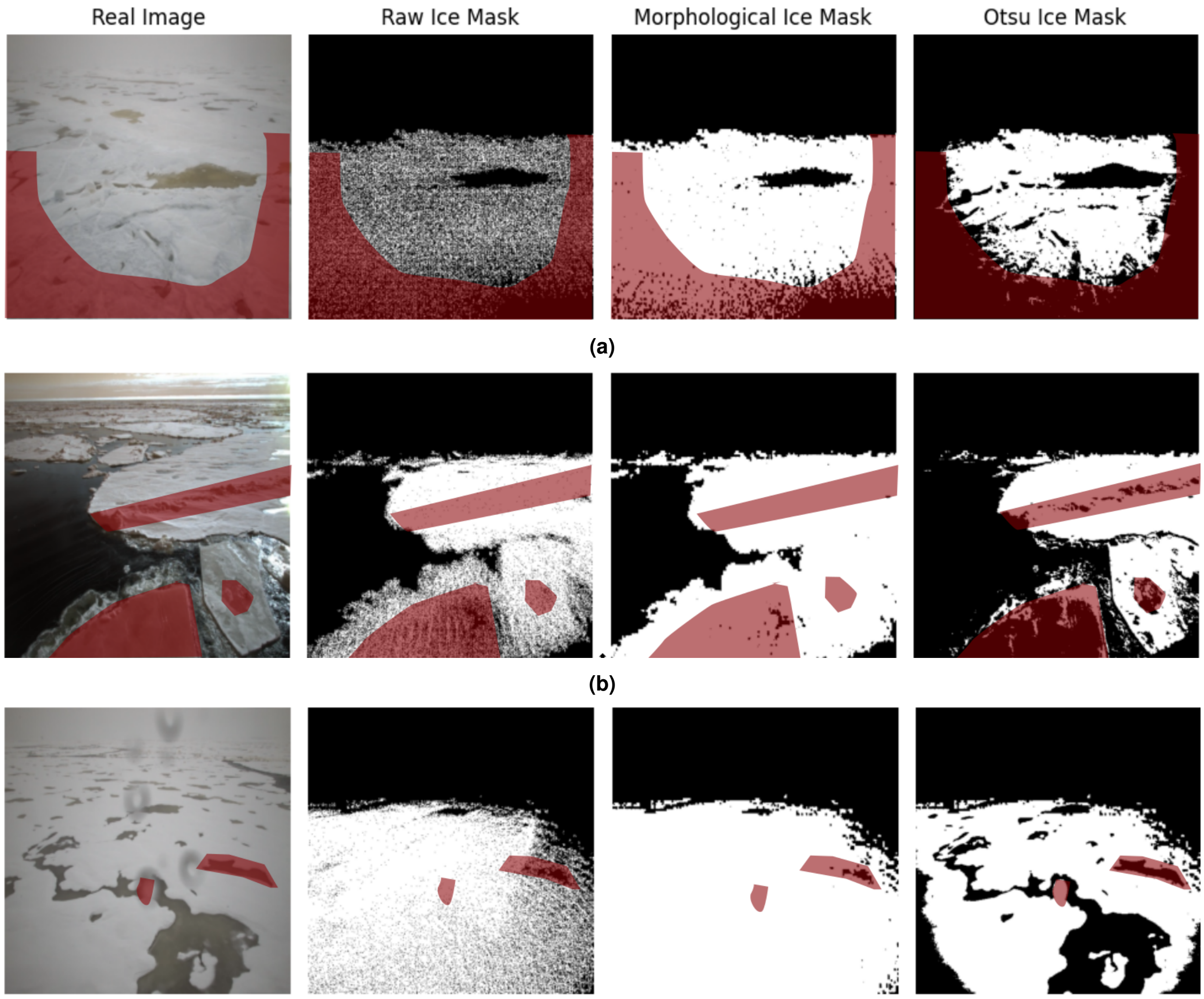
Figure 5.1a displays an example of motion blur. In the Raw and Morphogical datasets, there is an artifact from a piece of ice that has already passed through the camera frame, but remains in the LiDAR FOV because of the difference in capture rates. This introduces the chance that the model will learn to predict false positives. In addition to the motion blur, the sparsity of the Raw point cloud is evident in Figure 5.1a. Even with the addition of the closing morphological filter, there remains very sparse areas of ice especially close to the camera. Similar to Figure 5.1a, Figure 5.1b shows the negative effect motion blur can have on the ground truth labelling. Despite the area highlighted in red consisting of separate ice floes, the motion blur present in the LiDAR system blends them together. This gives the impression of one large floe and over-predicts ice, which again could influence false positives in model predictions. The last example in Figure 5.1c contains both the effects of motion blur and sparsity as well. In all of these examples, the Otsu dataset seems to perform a better job. The hybrid combination of point cloud and image threshold eliminates the false artifact in Figure 5.1a and better defines the edges of the ice floes in Figures 5.1b & 5.1b. However, this dataset still suffers from the point cloud sparsity mentioned earlier.

5.1.2 Otsu-Hybrid Dataset

Despite the differences in appearance of the ground truth samples in Figure 4.5, the Otsu dataset is still heavily related to the Raw dataset. Since mask points were only kept if they were present in both the Raw mask and the Otsu image threshold, the Otsu threshold succeeds in some areas that the Raw and Morphological datasets fail, but fall short in other samples. Figure 5.2 displays three examples of the weaknesses of the Otsu dataset while Figure 5.3 shows two images where the desired ground truth is achieved.

The first image Figure 5.2a displays a vignetting effect that is not uncommon in when using image thresholding methods. Usually, the edges of an image appear to be slightly darker than the center. Depending on the image and the severity of the vignetting, Otsu thresholding can interpret the edges as water. When it occurs, it impacts the Otsu mask to a great degree by adding in false negatives. This has previously been investigated, specifically a de-vignetting filter is used to some success in [11]. Figure 5.2b is another shortcoming of image thresholding methods. Namely, shadows and darker shades of ice or ice near the surface tend to be classified as water. This is problematic as it tends to classify ice features such as ridges as partly water, creating an excess of false negatives. In the final example, 5.2c, the Otsu image thresholding method classifies most meltponds as water. In addition, it classifies an raindrop on the camera frame as ice. This is partly due as well to the motion blur seen in the raw mask, but nevertheless remains a failure of the preprocessing method. The classification of meltponds as water is another example of an effect that will likely cause the predictions of false negatives.

However, this method succeeds in other samples. Figure 5.3a shows how the hybrid nature of the data preprocessing avoids classifying the sun glare as ice. Without the LiDAR point cloud, Otsu's binarization would classify this area and other bright areas as ice. Furthermore, both 5.3a and 5.3b demonstrate the ability of the Otsu preprocessing method to correctly define the edges of ice floes.



(a)

(b)

(c)

Figure 5.2: Otsu Dataset Shortcomings
Areas highlighted in red represent dataset shortcomings

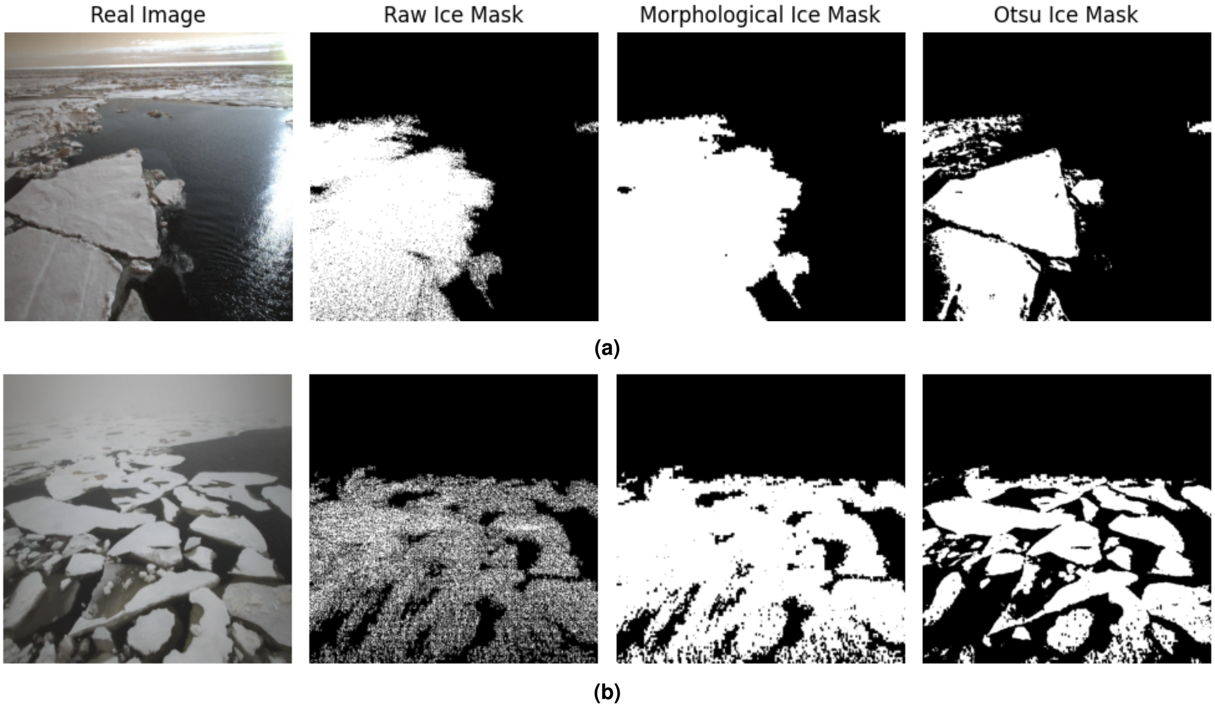


Figure 5.3: Otsu Dataset Successes

This proves especially useful in cases where the Raw LiDAR point cloud possesses significant motion blur. The hybrid nature of this dataset does produce a significant amount of false negatives, but clearly reduces the amount of false positives.

5.1.3 Comparison to Manually Labelled Datasets

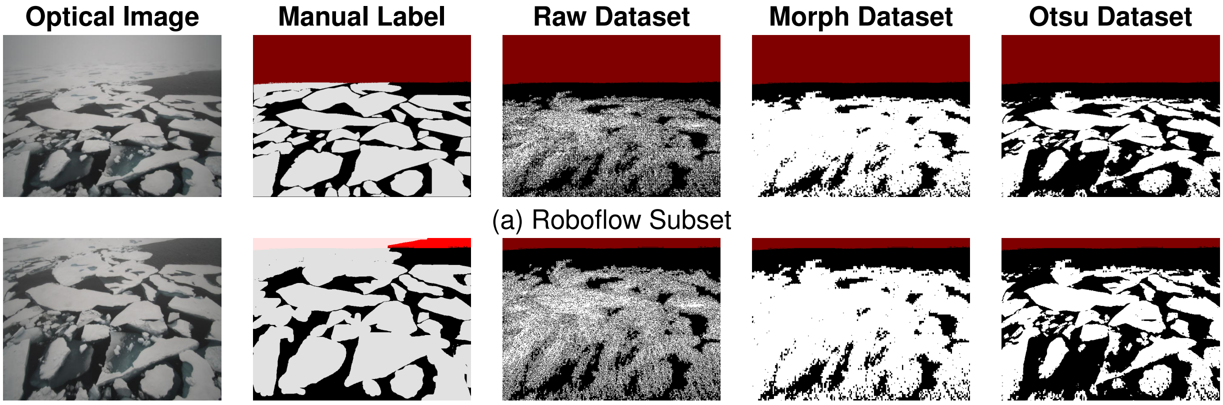
The two manually labelled datasets are referred to as GoNorth and Roboflow. The GoNorth dataset consists of 186 images from the *Kronprins Haakon* voyage that were cropped and labelled by Nabil Panchi. The Roboflow dataset consists of 175 images from same voyage labelled by the author in Roboflow. The 175 images are a subset of the testing split dictated in Table 4.2. To compare, the preprocessing methods were applied to the test set samples and the masks compared. The purpose of this section is to evaluate how well the three preprocessed datasets emulate manually labelled sea ice images.

Table 5.1 shows that all three datasets do not fully approximate manual labelling, but the Morphological and Otsu datasets come closer than the Raw. Their scores mainly differ in the classification of false positives and false negatives. The Morphological dataset scores the lowest in precision but the highest in recall. Thus, this dataset is predicting an excess of false positives but has the lowest amount of false negatives. The Otsu dataset succeeds in the precision category, eliminating most of the false positives but lags significantly in recall due to false negatives. Differences in performance between the

Subset	Dataset	IOU	DICE	Pixel Accuracy	Precision	Recall
GoNorth	Raw	0.54	0.68	0.77	0.83	0.62
	Morphological	0.66	0.78	0.83	0.81	**0.79**
	Otsu	**0.68**	**0.79**	**0.85**	**0.97**	0.69
Roboflow	Raw	0.29	0.42	0.58	0.79	0.30
	Morphological	**0.53**	**0.66**	**0.75**	0.78	**0.59**
	Otsu	0.52	0.65	**0.75**	**0.90**	0.53
Combined	Raw	0.42	0.55	0.68	0.81	0.47
	Morphological	**0.60**	**0.72**	0.79	0.80	**0.69**
	Otsu	**0.60**	**0.72**	**0.80**	**0.93**	0.61

Table 5.1: Manually Labelled Dataset Evaluation
Bold numbers represent the best performances for each subset between the three preprocessed datasets

two testing datasets are due to slight differences in the manual labelling. Figure 5.4 displays the two different labelling methods on the same input image. The main difference is cropping out of the majority of the LiDAR FOV mask, slightly changing the perspective.



(a) Roboflow Subset

(b) GoNorth Subset

Figure 5.4: Manual Label Dataset Differences with LiDAR FOV Mask Highlighted
Areas highlighted in red represent areas masked out by the LiDAR FOV mask

The confusion matrices presented in Figure 5.5 confirm the shortcomings seen in Figures 5.1 and 5.2. Namely, the Raw dataset contains a large amount of false negatives. This is expected due to the sparsity of the LiDAR point cloud; the few returned LiDAR points vastly under-represent the ice features in the images. The Morphological dataset is an improvement, increasing the number of true positives and reducing the false negatives. However, this comes at the cost of overestimating ice due to motion blur and increases the number of false positives. Finally, the Otsu dataset vastly decreases the excess false positives of the Morphological dataset as it correctly defines ice floe edges. The cost of performance is a significant increase in false negatives due to the misclassification of shadows and meltponds as water.

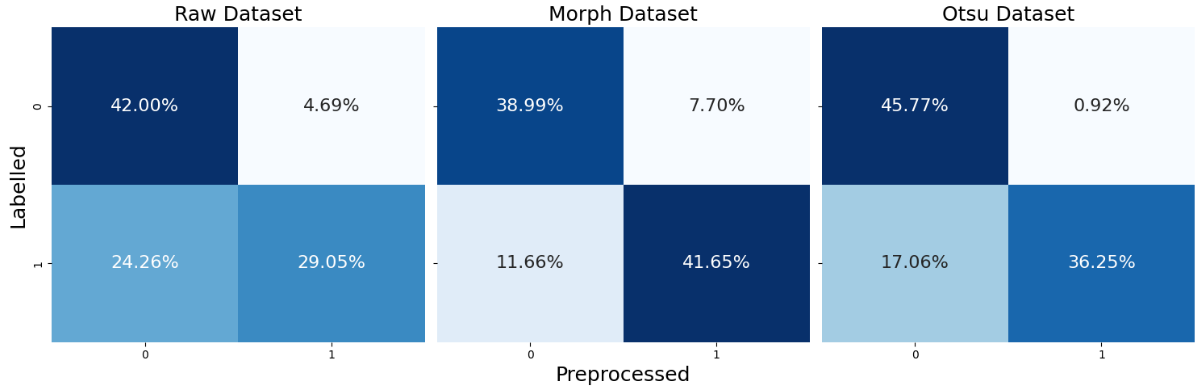


Figure 5.5: Preprocessed Dataset Confusion Matrices

5.2 Model Training

All three models were trained on each dataset for 30 epochs using a batch size of 16 along with the parameters mentioned in Table 4.5. Below is the measured BCE loss across each dataset. In this section and all of the following, the area masked out by the LiDAR FOV mask was not included in metric calculations.

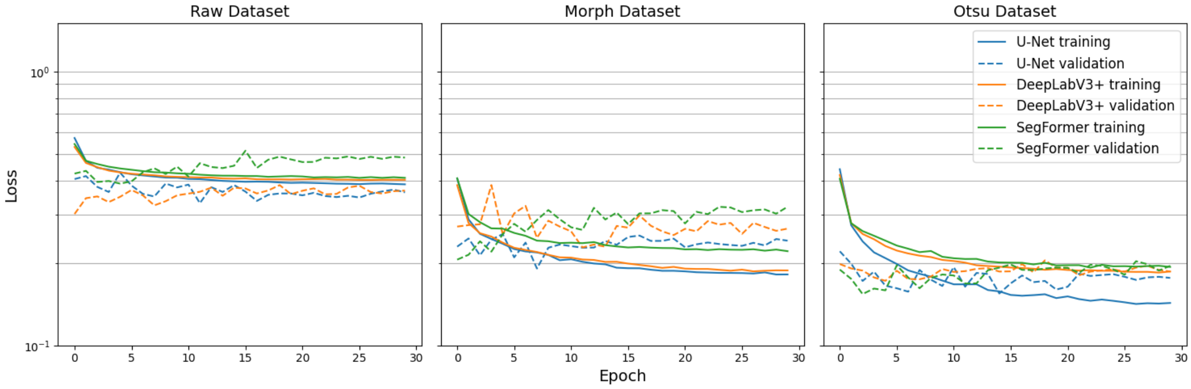


Figure 5.6: Training Losses for Models on each Dataset

It is clear that the models presented in Figure 5.6 learn more on the Morphological and Otsu datasets than the Raw. In all cases, validation loss begins lower but gradually converges with the training loss. This is likely due to the data augmentation performed on the training set that is absent in the validation split. Data augmentation artificially increases the difficulty of the training split to help the model generalize better. Most evident in the Morphological dataset, there seems to be overfitting as the training loss continues to decrease as the validation loss stays relatively constant. In the Otsu dataset, U-Net seems to minimize the training loss most effectively compared to its peers. In the other datasets, this difference is less severe.

The decrease in losses over the training period in the Morphological and Otsu datasets is likely due to better representation of the ground truth. With some of the sparsity of the point cloud filled in, the models are increasingly able to distinguish between sea ice and water. In simple terms, this binary segmentation task is finding a correlation between the 'lighter' sea ice pixels in the images and the preprocessed ground truth. Thus, the closer the preprocessed datasets emulate manually labelled ice masks, the quicker these models will learn to properly define the sea ice objects. The averaged results of the models' performance throughout training are displayed in Table 5.2.

		Loss		**IOU**		**DICE**	
Model	**Dataset**	**Train**	**Val**	**Train**	**Val**	**Train**	**Val**
U-Net	Raw	0.41	0.37	0.64	0.43	0.78	0.59
	Morph	0.21	0.23	**0.88**	0.71	**0.94**	0.83
	Otsu	**0.18**	**0.18**	0.85	**0.73**	0.92	**0.84**
DeepLabV3+	Raw	0.42	0.36	0.65	0.48	0.78	0.64
	Morph	**0.21**	0.27	**0.88**	0.69	**0.93**	0.81
	Otsu	**0.21**	**0.19**	0.82	**0.71**	0.90	**0.82**
SegFormer	Raw	0.43	0.45	0.64	0.36	0.78	0.49
	Morph	0.24	0.29	**0.86**	**0.70**	**0.93**	**0.82**
	Otsu	**0.22**	**0.18**	0.81	**0.70**	0.90	0.81

Table 5.2: Training Statistics by Model
Bold numbers represent the best performances for each model between the three preprocessed datasets

The averaged training results in Table 5.2 agree with the assertion that the models have a difficult time learning to minimize loss on the Raw dataset. Overall, performances are nearly identical in terms of training loss between the Morphological and Otsu datasets. In a significant number of cases, the best training IoU belongs to the Morphological dataset while the Otsu dataset provides the best validation IoU. This supports the overfitting seen in the Morphological dataset in Figure 5.6. Additionally, the models likely can learn the image thresholding methods implemented in the Otsu dataset. This is reinforced by the low and constant nature of the validation losses in the Otsu dataset training curves.

5.3 Evaluation of Model Predictions

The trained models' predictions were then tested on each of the two manually labelled datasets. Table 5.3 contains the averaged results for each combination of model and dataset. The best IoU scores for each subset are produced by models trained on the Morphological dataset. However, the differences from model to model on the Morphological and Otsu datasets are small, suggesting the different architectures do not have a significant impact on the performance. Confirming the datasets' respective strengths and weaknesses, the Morphological-trained models consistently score the highest in recall while the Otsu-trained models succeed in precision. In every case, the models IoU scores are higher on the goNorth subset than the Roboflow subset. The correlation suggests the models perform better when the camera perspective is shifted downward towards the ice, eliminating faraway ice features and the horizon. The overall best performer was the U-Net model trained on the Morphological dataset. Despite performing the lower in Precision, its Recall and IoU scores are the highest. The SegFormer model trained on the Otsu dataset scores impressively on precision, largely avoiding the prediction of any false positives.

Subset	Model	Dataset	IOU	DICE	Pixel Accuracy	Precision	Recall
Roboflow	U-Net	Raw	0.48	0.59	0.74	0.79	0.51
		Morph	**0.73**	**0.82**	**0.86**	0.86	**0.83**
		Otsu	0.61	0.72	0.82	0.90	0.62
	DeepLabV3+	Raw	0.56	0.67	0.78	0.82	0.59
		Morph	0.65	0.75	0.84	0.80	0.73
		Otsu	0.61	0.72	0.82	0.92	0.61
	SegFormer	Raw	0.42	0.54	0.66	0.85	0.44
		Morph	0.70	0.78	**0.86**	0.81	0.79
		Otsu	0.66	0.78	0.82	**0.98**	0.66
goNorth	U-Net	Raw	0.64	0.73	0.82	0.87	0.72
		Morph	**0.78**	**0.87**	0.88	0.82	**0.95**
		Otsu	0.76	0.85	0.89	0.96	0.78
	DeepLabV3+	Raw	0.73	0.82	0.87	0.91	0.81
		Morph	**0.78**	0.86	0.88	0.83	0.92
		Otsu	0.77	0.86	**0.90**	0.96	0.80
	SegFormer	Raw	0.68	0.76	0.83	0.89	0.74
		Morph	0.77	0.86	0.88	0.84	0.90
		Otsu	0.74	0.84	0.89	**0.97**	0.76
Combined	U-Net	Raw	0.56	0.66	0.78	0.83	0.62
		Morph	**0.76**	**0.85**	**0.87**	0.84	**0.89**
		Otsu	0.69	0.78	0.86	0.93	0.70
	DeepLabV3+	Raw	0.65	0.75	0.83	0.87	0.70
		Morph	0.72	0.80	0.86	0.81	0.83
		Otsu	0.69	0.79	0.86	0.94	0.71
	SegFormer	Raw	0.55	0.65	0.74	0.87	0.59
		Morph	0.73	0.82	**0.87**	0.83	0.84
		Otsu	0.70	0.81	0.85	**0.97**	0.71

Table 5.3: Trained Models' Performance on Manually Labelled Datasets
Bold numbers represent the best scores of any model/dataset combination on a manually labelled subset

The confusion matrices for U-Net model predictions in Figure 5.7 support the results in Table 5.3 and bear similarity to the dataset confusion matrices in Figure 5.5. The Morphological dataset trained U-Net has both the highest percentage of false positives but also the highest percentage of true positives. It scores significantly lower in the percentage of false negatives, whereas the Raw and Otsu trained U-Nets have higher values. The U-Net trained on its respective datasets struggles from the same issues as the datasets: false negatives due to sparsity in the Raw, misclassification of shadows and meltponds as water in the Otsu, and an over-representation of ice in the Morphological.

Despite the similarities between the dataset and the U-Net prediction confusion matrices, there are a few important differences. In all three datasets, the percentage of false negatives decreased from dataset to prediction. Additionally, there is an increase in true positives from dataset to prediction. This trend is encouraging and suggests that the models have a slightly better understanding of the scene than the automatically labelled datasets.

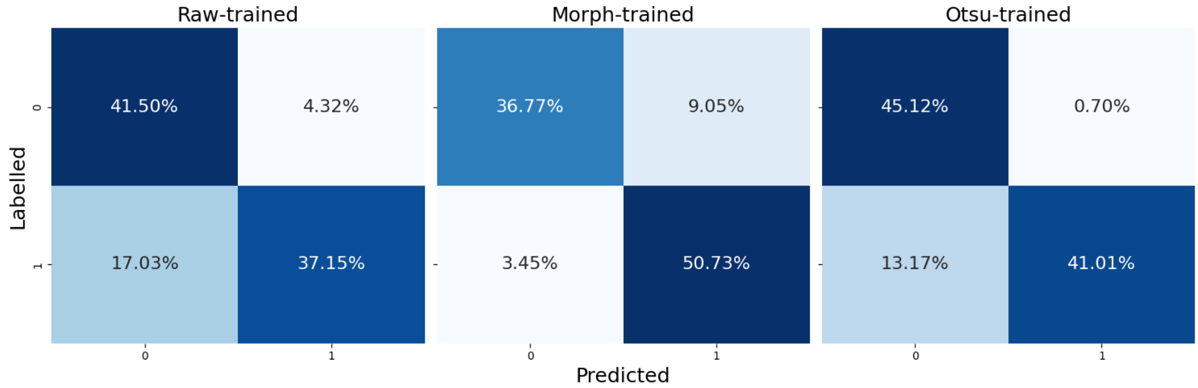


Figure 5.7: U-Net Prediction Confusion Matrices

Figure 5.8 shows three correlations between variables for a U-Net trained on each of the three datasets. Figure 5.8a presents the correlation between the proportion of ice in the input image and the corresponding loss on that image. In both the Raw and Otsu trained U-Net models, the loss increases with the proportion of sea ice. This could suggest a latent underprediction of ice or a difficulty in defining the ice. It is likely due to the to the fact that as the percentage of ice increases, so will the number of false negatives as was seen in the dataset confusion matrices (5.5). The same pattern is shown in the pixel accuracy vs. SIPP Label plots (5.8b). Specifically in the Raw-trained U-Net, there are cases with high SIPP that result in very low pixel accuracy scores. This behavior is not as severe in the Otsu-trained U-Net. The relationship between the sea ice proportion of the prediction and the label (5.8c) further confirm this behavior. The Otsu-trained U-Net predictions lie completely below the 1-to-1 trendline, confirming the high precision scores in all Otsu-trained models (5.3). The Raw-trained model displays similar behavior within the Roboflow subset, but it is much more accurate on the GoNorth subset. The Morphological trained U-Net model displays a 'triangular' behavior, reaching its max loss and lowest pixel accuracy scores near 0.5 SIPP. This model likely heavily overpredicts ice as the SIPP rises, a behavior that does not have the same affect on performance when the images have a very high percentage of ice. Figure 5.8c aligns with this assertion, with most of the samples above the 1-to-1 trendline.

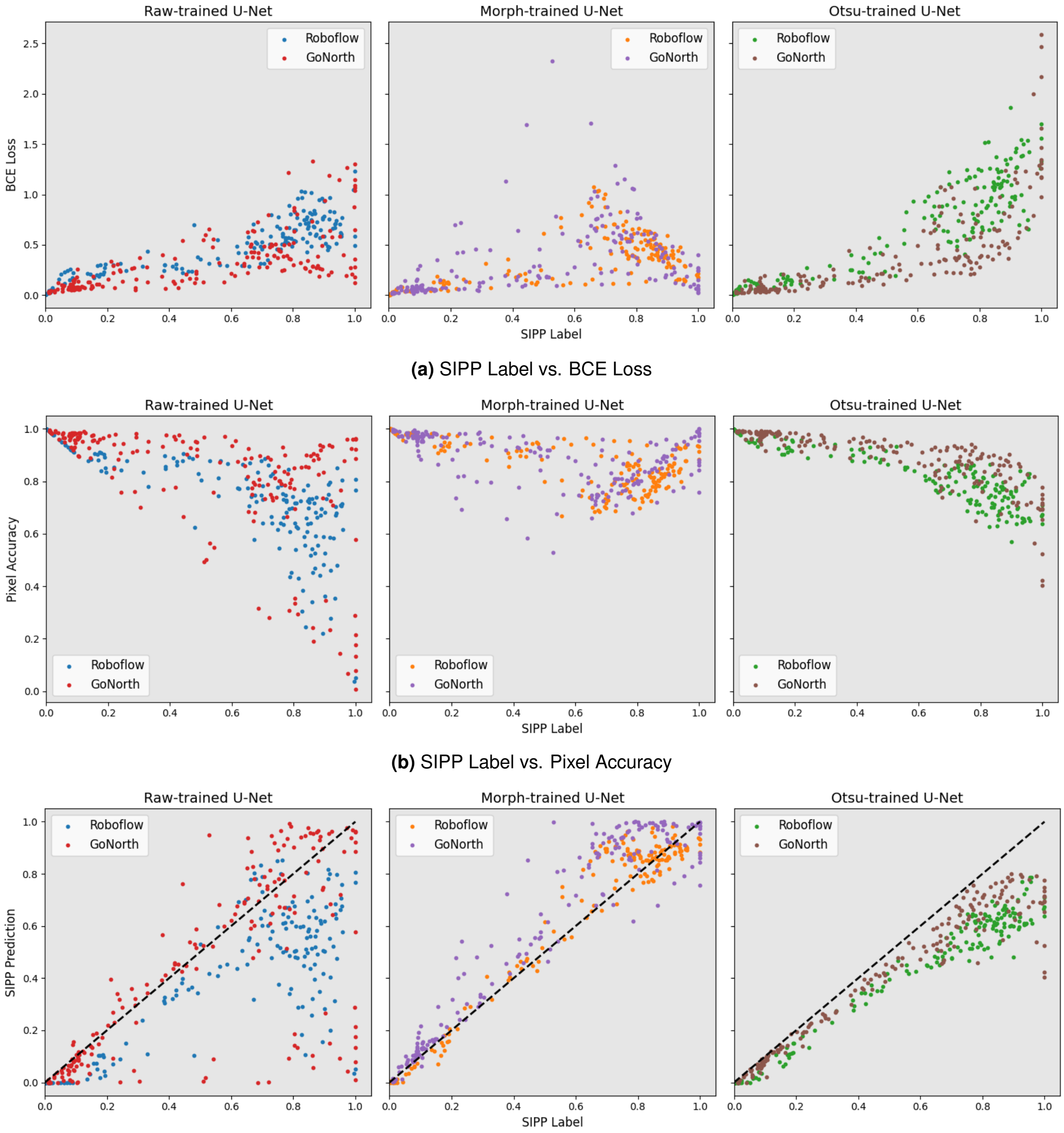
Conclusions can additionally be made within the manually labelled subsets. In the Raw- and Otsu-trained models, the predictions on the GoNorth subset lie much closer to the ideal 1-to-1 relationship between labelled and predicted SIPP. This behavior suggests that these models struggle with predicting areas close to the LiDAR FOV mask. When this area is cropped out of the input images, the models perform significantly better. This behavior is not entirely unexpected, as the models received no information from loss back-propagation on these areas. In the Morphological-trained model predictions, the GoNorth subset seems to suffer more severely from the overprediction of ice than its Roboflow counterpart. This suggests that the Morphological-trained U-Net tends to overpredict on areas within the LiDAR FOV mask, a behavior that is mediated with the addition of areas closer to the mask (further from the camera).

The model predictions on the Roboflow subset in Figure 5.9 visualize the assertions stated above. The predictions were taken from each dataset's best performing model: DeepLabV3+ for the Raw dataset, U-Net for the Morphological dataset, and SegFormer for the Otsu dataset. The overprediction of ice in the Morphological-trained model is especially evident, with the blending together of multiple ice floes into one mass. The Otsu-trained model correctly defines the floe borders but mislabels parts of floes that are slighly underwater. The Raw-trained model struggles in general with the prediction, but clearly performs worse close to the camera as a result of the severe sparsity present in the point clouds in this location. Despite areas far from the camera being removed from loss back-propagation by the LiDAR FOV mask, the models still predicted ice in these areas. The bright pixel values for foggy and sunny skies likely indicated to the models that ice was present, but they struggled to correctly define boundaries. Poor performance in this section of the images is to be expected, as the models did not learn on this portion of the input.

The predictions for the GoNorth dataset in Figure 5.10 offer more insight. With the camera focused more closely on the ice features, the Raw-trained model produces an accurate prediction, albeit with some false negatives and misclassification of undersea ice. The Morphological-trained model again combines ice floes together but correctly predicts areas of undersea ice. Finally, the Otsu-trained model incorrectly classifies meltponds as open water but correctly predicts the above-water ice floe edges. In general, the model predictions confirm the assertions made in Tables 5.1,5.3 and Figures 5.5, 5.7. To summarize, models trained with the Raw and Otsu datasets suffer from overprediction of false negatives but score highly in precision. Models trained on the Morphological dataset overpredict false positives but score highly in IoU and recall.

Overall, the metrics presented in this section suggest that models trained on the Morphological dataset produce the highest results. Yet, Otsu-trained model predictions have their merits: low prediction of false positives and correct boundary detection. The Raw-trained models struggle with predictions on the Roboflow subset but increase in performance when viewing the ice features closer in the GoNorth subset.

The choice of preprocessed dataset will ultimately come to a decision on use case. If a high recall is desired where few ice pixels are incorrectly classified as water, the Morphological dataset will be the

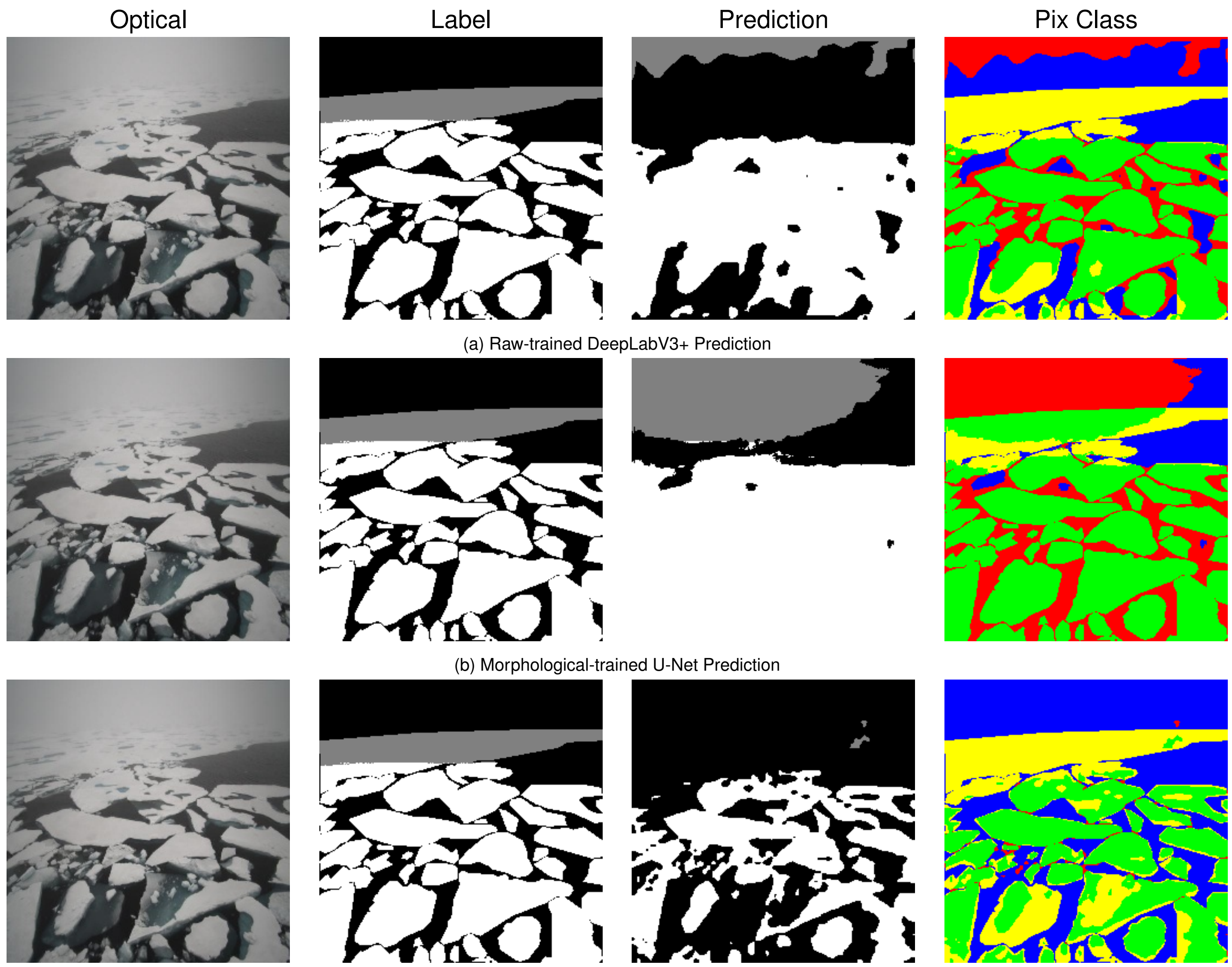


(a) SIPP Label vs. BCE Loss

(b) SIPP Label vs. Pixel Accuracy

(c) SIPP Label vs. SIPP Prediction

Figure 5.8: Performance Correlations
The black dotted line in 5.8c represents a 1-to-1 correlation trendline



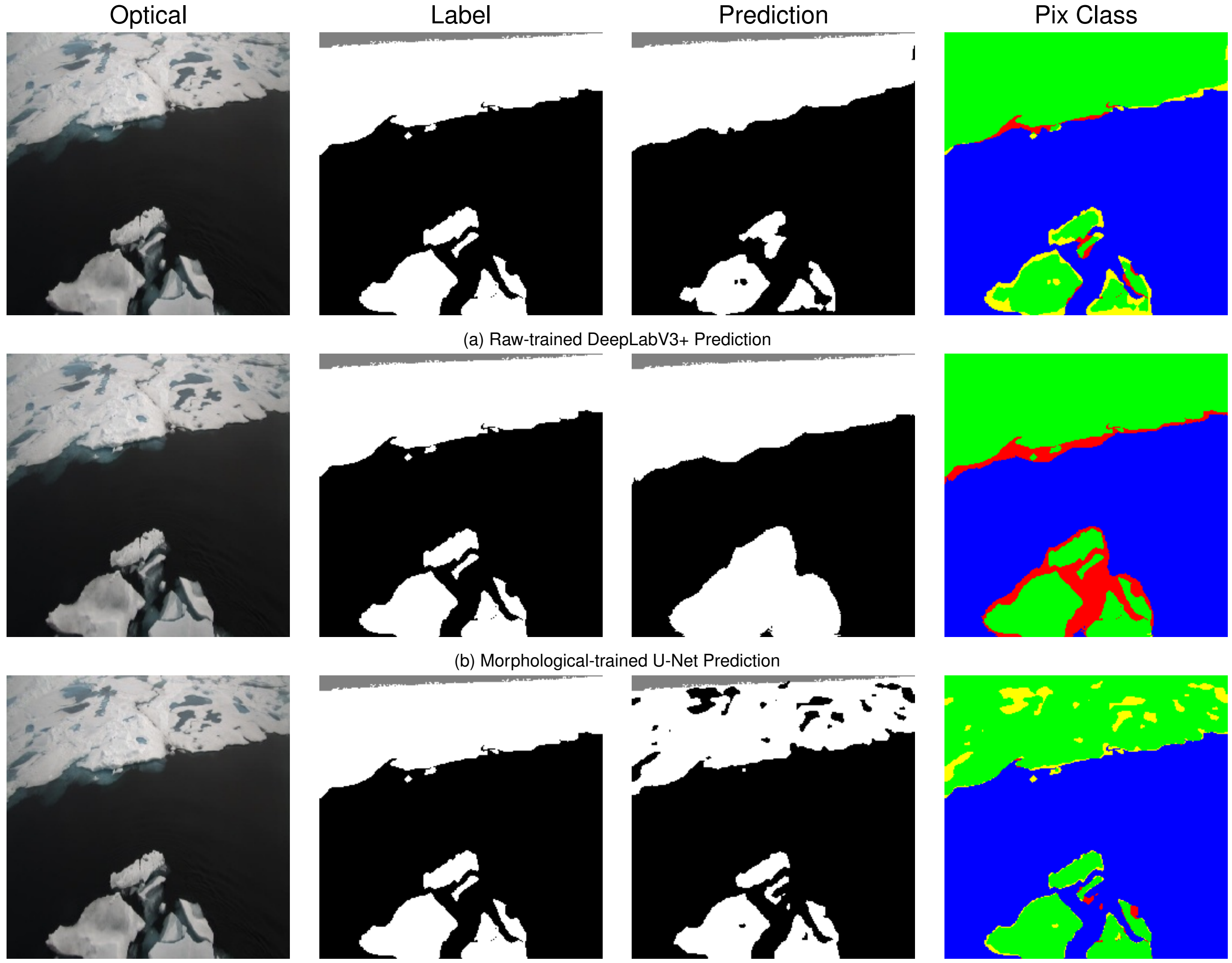
(a) Raw-trained DeepLabV3+ Prediction

(b) Morphological-trained U-Net Prediction

(c) Otsu-trained SegFormer Prediction

Figure 5.9: Roboflow Image Predictions

Gray shading represents predictions outside the LiDAR FOV mask. The last column colors the prediction based on pixel classification: Green:TP, Red:FP, Blue:TN, Yellow:FN



(a) Raw-trained DeepLabV3+ Prediction

(b) Morphological-trained U-Net Prediction

(c) Otsu-trained SegFormer Prediction

Figure 5.10: goNorth Image Predictions

Gray shading represents predictions outside the LiDAR FOV mask. The last column colors the prediction based on pixel classification: Green:TP, Red:FP, Blue:TN, Yellow:FN best. If instead precision is the priority, with its few false positives, the Otsu dataset is likely the best option.

5.4 Model Comparison

While the main purpose of this experiment was to compare automated labelling techniques, the performances of U-Net, DeepLabV3+, and SegFormer differ. Viewing the statistics for the combined labelled dataset in Table 5.4, the models display similar performance on each dataset. The largest differences in performance come from the Raw-trained DeepLabV3+ compared to its peers. This is likely due to the atrous spatial pyramid pooling present in DeepLabV3+ that allows multiple receptive fields to be incorporated into model predictions. Utilizing multiple rates within the atrous convolutions, greater spatial context could be incorporated into the model to overcome the sparsity of the Raw dataset. The Morphological-trained U-Net model is the best overall performer with an IoU of 0.76. The the skip connections present in the U-Net likely helped to maintain important information that is normally lost during down-sampling in the encoder. As it was originally designed for medical image segmentation, U-Net has proven to be exceptional at defining object boundaries and classifying images at the pixel level [27]. Despite using the lightest-weight SegFormer architecture, the Morphological- and Otsu-trained SegFormer models scored competitively on the combined manually labelled subset. Its hierarchal transformer encoder is able to capture both the local and global context to inform its predictions. Because of the lightweight MLP decoders and lack of positional embeddings, the SegFormer consistently scores the highest inference times.

Dataset	Model	IOU	DICE	Pixel Accuracy	Precision	Recall	Inference Time (imgs/s)
Raw	U-Net	0.56	0.66	0.78	0.83	0.62	48.15
	DeepLabV3+	**0.65**	**0.75**	**0.83**	**0.87**	**0.70**	49.40
	SegFormer	0.55	0.65	0.74	**0.87**	0.59	**57.58**
Morph	U-Net	**0.76**	**0.85**	**0.87**	**0.84**	**0.89**	49.38
	DeepLabV3+	0.72	0.80	0.86	0.81	0.83	47.89
	SegFormer	0.73	0.82	**0.87**	0.83	0.84	**58.88**
Otsu	U-Net	0.69	0.78	**0.86**	0.93	0.70	47.62
	DeepLabV3+	0.69	0.79	**0.86**	0.94	**0.71**	48.40
	SegFormer	**0.70**	**0.81**	0.85	**0.97**	**0.71**	**58.99**

Table 5.4: Combined Manually Labelled Subset Performance by Model
Bold numbers represent the best performance on the combined manually labelled test set for all models trained on the same preprocessed dataset

Overall, all three models produce similar performances. While each has its strengths, these results come from an extremely small sample size: 30 epochs with a batch size of 16 using a training dataset of under 2000 images. Therefore, meaningful conclusions about the general applicability of these three model types cannot be made solely from this study.

6

Conclusion

Contents

This graduate thesis presents three different methods of the automated labelling of Arctic sea ice using multi-modal information provided by LiDAR. Three different models were tested on these datasets to evaluate if the preprocessed datasets could emulate manually labelled ground truth images. While the performances were not near state-of-the-art sea ice segmentation as seen in Table 3.4 [12], they are encouraging enough to warrant further investigation. After minimal training on a small dataset, U-Net was able to acheive an IoU score of 0.76 on a 361 manually labelled image test set. SegFormer and DeepLabV3+ performed similarly with IoU of 0.73 and 0.72 respectively. SegFormer was able to produce the best inference times, at almost 59 images per second.

Quantitative comparisons show that the Raw dataset did not provide an accurate depiction of the labelled ground truth because of the excessive sparsity present in the Raw point cloud. The Morphological dataset trained models performed the best in IoU and recall, but suffered from overprediction of ice due to motion blur present in the point clouds. The Otsu-Hyrbid method defined ice object boundaries, but suffered from misclassifications of meltponds and shadows as open water. Otsu-trained models performed nearly as well as the Morphological trained models in IoU, but greatly outpaced them in precision due to the lack of false positives. There was a clear correlation between the the characteristics of each preprocessed dataset and the predictions from models trained on them. However, there was an encouraging pattern displayed in Figures 5.5 and 5.7 where U-Net's predictions improved from the preprocessed datasets. To choose a preprocessed dataset for training, care should be placed in the desired end usage. If the purpose of the model is not miss any positive pixels (e.g. a ship navigating the Arctic trying to avoid any contact with ice), the Morphological dataset should be chosen due to its high recall scores. If precise ice floe boundaries are the priority, the Otsu-dataset should be used. Furthermore, if segmentation of meltponds is desired (as there is a correlation between spring meltpond area and September sea ice extent [64]), the Otsu-trained models show potential in locating and defining their boundaries.

Overall, this study was designed to prove if LiDAR point cloud information can be transformed into accurate ground truth labels for the training of a neural network. Utilizing three different preprocessing methods and three different models, binary segmentation results show that this is a feasible proposition.

6.0.1 Recommendations and Future Work

For future studies in this field, effort should be placed on increasing the accuracy of the automated labelling system. For example, a de-vignetting algorithm similar to the one applied in [11] could greatly increase the accuracy of the Otsu-Hybrid dataset. Other image processing methods, such as superpixel segmentation as demonstrated in [56], could better emulate a manually labelled image of sea ice.

To improve training and inference scores, model architectures can be modified and hyperparameters tuned. While this study was designed as a proof-of-concept, modifying model architectures for sea ice segmentation and classification have shown promise in the literature [7] [12] [58].

Bibliography

[1] N. Balasooriya, B. Dowden, J. Chen, O. De Silva, and W. Huang, “In-situ sea ice detection using DeepLabv3 semantic segmentation,” in *OCEANS 2021: San Diego – Porto*, pp. 1–7, ISSN: 0197-7385. [Online]. Available: https://ieeexplore.ieee.org/document/9705801

[2] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey.” [Online]. Available: http://arxiv.org/abs/2001.05566

[3] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuScenes: A multimodal dataset for autonomous driving.” [Online]. Available: http://arxiv.org/abs/1903.11027

[4] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, “MOTChallenge 2015: Towards a benchmark for multi-target tracking.” [Online]. Available: http://arxiv.org/abs/1504.01942

[5] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft COCO: Common objects in context.” [Online]. Available: http://arxiv.org/abs/1405.0312

[6] B. Dowden, O. De Silva, W. Huang, and D. Oldford, “Sea ice classification via deep neural network semantic segmentation,” vol. 21, no. 10, pp. 11 879–11 888. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9225140?casa_token=2r8dnG-CtfMAAAAA:HHkuFhZeRWF1XCTPk2rOb7N0snXbAXQo-_612DjSLZRmUzFW-ZTHKkO6zvonOQluRm2OSeKklQ

[7] S. Li, M. Wang, J. Wu, S. Sun, M. Shi, and R. Ma, “Sea ice detection network for icebreakers in polar environments with attention-based deeplabv3+ architecture,” vol. 2718, no. 1, p. 012062, publisher: IOP Publishing. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2718/1/012062

[8] US EPA and OAR. Climate change indicators: Arctic sea ice. [Online]. Available: https://www.epa.gov/climate-indicators/climate-change-indicators-arctic-sea-ice

[9] J. A. Screen and I. Simmonds, “Increasing fall-winter energy loss from the arctic ocean and its role in arctic temperature amplification,” vol. 37, no. 16, https://onlinelibrary.wiley.com/doi/pdf/10.1029/2010GL044136. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1029/2010GL044136

[10] M. Tschudi, W. N. Meier, J. S. Stewart, C. Fowler, and J. Maslanik, “EASE-grid sea ice age.” [Online]. Available: http://nsidc.org/data/nsidc-0611/versions/4

[11] A. Sandru, H. Hyyti, A. Visala, and P. Kujala, “A complete process for shipborne sea-ice field analysis using machine vision,” vol. 53, no. 2, pp. 14 539–14 545. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S240589632031870X

[12] N. Panchi, E. Kim, and A. Bhattacharyya, “Supplementing remote sensing of ice: Deep learning-based image segmentation system for automatic detection and localization of sea-ice formations from close-range optical images,” vol. 21, no. 16, pp. 18 004–18 019. [Online]. Available: https://ieeexplore.ieee.org/document/9443178

[13] J. Richter-Menge, M. L. Druckenmiller, and R. L. Thoman, “Arctic report card 2019.” [Online]. Available: https://arctic.noaa.gov/Report-Card/Report-Card-2019/

[14] R. L. Thoman, T. A. Moon, and M. L. Druckenmiller, “NOAA arctic report card 2023: Executive summary,” publisher: NOAA Global Ocean Monitoring and Observing Program. [Online]. Available: https://repository.library.noaa.gov/view/noaa/56621

[15] T. Wei, Q. Yan, W. Qi, M. Ding, and C. Wang, “Projections of arctic sea ice conditions and shipping routes in the twenty-first century using CMIP6 forcing scenarios,” vol. 15, no. 10, p. 104079, publisher: IOP Publishing. [Online]. Available: https://dx.doi.org/10.1088/1748-9326/abb2c8

[16] A. H. Lynch, C. H. Norchi, and X. Li, “The interaction of ice and law in arctic marine accessibility,” vol. 119, no. 26. [Online]. Available: https://www.pnas.org/doi/10.1073/pnas.2202720119

[17] A. Sandru, A. Visala, and P. Kujala, “Shipborne sea-ice field mapping using a LiDAR,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4350–4357, ISSN: 2153-0866. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9636275?casa_token=S0cpBkxI0zQAAAAA:y7yHH4j0DSy53LaJEmFTGYgT1rbCpRN6KY4sGAosL1FydIORDY0juhIbhVs78V8jEo4tMJxe0Q

[18] M. W. M. Said, O. De Silva, G. K. I. Mann, C. Daley, J. R. Dolny, D. Oldford, and R. G. Gosine, “LiDAR and vision based pack ice field estimation for aided ship navigation,” num Pages: 18. [Online]. Available: https://research.library.mun.ca/14491/

[19] R. Skjetne, “DigitalSeaIce: Multi-scale integration and digitalization of arctic sea ice observations and prediction models.” [Online]. Available: https://www.ntnu.edu/digitalseaice

[20] O. G. Veggeland, E. Kim, and R. Skjetne, “Multi modal mapping of sea ice fields from remote shipborne instrumentation,” in *OMAE2024-125706*. ASME.

[21] P. Maragos, *Morphological Signal and Image Processing*. CRC Press, vol. 20091678, pp. 1–31. [Online]. Available: http://www.crcnetbase.com/doi/abs/10.1201/9781420046052-c16

[22] P. Chhikara. Understanding morphological image processing and its operations. [Online]. Available: https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756

[23] H. F. Mahmood. What is morphological image processing? [Online]. Available: https://www.educative.io/answers/what-is-morphological-image-processing

[24] Z. Yu, Y. Zhao, and X. Wang, “Research advances and prospects of mathematical morphology in image processing,” in *2008 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1242–1247, ISSN: 2326-8239. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4670786

[25] N. Otsu, “A threshold selection method from gray-level histograms,” vol. 9, no. 1, pp. 62–66, conference Name: IEEE Transactions on Systems, Man, and Cybernetics. [Online]. Available: https://ieeexplore.ieee.org/document/4310076/?arnumber=4310076

[26] M. M. Taye, “Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions,” vol. 11, no. 3, p. 52. [Online]. Available: https://www.mdpi.com/2079-3197/11/3/52

[27] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation.” [Online]. Available: http://arxiv.org/abs/1505.04597

[28] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, ISSN: 1063-6919. [Online]. Available: https://ieeexplore.ieee.org/document/8100143

[29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation.” [Online]. Available: http://arxiv.org/abs/1802.02611

[30] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation.” [Online]. Available: https://arxiv.org/abs/1706.05587v3

[31] H. He, D. Yang, S. Wang, S. Wang, and Y. Li, “Road extraction by using atrous spatial pyramid pooling integrated encoder-decoder network and structural similarity loss,” vol. 11, no. 9, p. 1015. [Online]. Available: https://www.mdpi.com/2072-4292/11/9/1015

[32] A. Sankar. A primer on atrous convolutions and depth-wise separable convolutions. [Online]. Available: https://towardsdatascience.com/a-primer-on-atrous-convolutions-and-depth-wise-separable-convolutions-443b106919f5

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” vol. 8691, pp. 346–361. [Online]. Available: http://arxiv.org/abs/1406.4729

[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale.” [Online]. Available: http://arxiv.org/abs/2010.11929

[35] S. Jamil, M. Jalil Piran, and O.-J. Kwon, “A comprehensive survey of transformers for computer vision,” vol. 7, no. 5, p. 287, number: 5 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2504-446X/7/5/287

[36] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” vol. 54, no. 10, pp. 200:1–200:41. [Online]. Available: https://doi.org/10.1145/3505244

[37] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “SegFormer: Simple and efficient design for semantic segmentation with transformers.” [Online]. Available: http://arxiv.org/abs/2105.15203

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, ISSN: 1063-6919. [Online]. Available: https://ieeexplore.ieee.org/document/5206848

[39] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” vol. 6, no. 1, p. 60. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0

[40] A. van Opbroek, M. A. Ikram, M. W. Vernooij, and M. de Bruijne, “Transfer learning improves supervised image segmentation across imaging protocols,” vol. 34, no. 5, pp. 1018–1030. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6945865

[41] C. Zhang, X. Chen, and S. Ji, “Semantic image segmentation for sea ice parameters recognition using deep convolutional neural networks,” vol. 112, p. 102885. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843222000875

[42] D. Shah. Intersection over union (IoU): Definition, calculation, code. [Online]. Available: https://www.v7labs.com/blog/intersection-over-union-guide,https://www.v7labs.com/blog/intersection-over-union-guide

[43] K. Adem, M. M. Ozguven, and Z. Altas, "A sugar beet leaf disease classification method based on image processing and deep learning," vol. 82, no. 8, pp. 12 577–12 594. [Online]. Available: https://doi.org/10.1007/s11042-022-13925-6

[44] S. Gite, A. Mishra, and K. Kotecha, "Enhanced lung image segmentation using deep learning," vol. 35, no. 31, pp. 22 839–22 853. [Online]. Available: https://doi.org/10.1007/s00521-021-06719-8

[45] H. Thisanke, C. Deshan, K. Chamith, S. Seneviratne, R. Vidanaarachchi, and D. Herath, "Semantic segmentation using vision transformers: A survey," vol. 126, p. 106669. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197623008539

[46] Y. Han, Y. Liu, Z. Hong, Y. Zhang, S. Yang, and J. Wang, "Sea ice image classification based on heterogeneous data fusion and deep learning," vol. 13, no. 4, p. 592, number: 4 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2072-4292/13/4/592

[47] A. S. Nagi, D. Kumar, D. Sola, and K. A. Scott, "RUF: Effective sea ice floe segmentation using end-to-end RES-UNET-CRF with dual loss," vol. 13, no. 13, p. 2460, number: 13 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2072-4292/13/13/2460

[48] B. Dowden, O. De Silva, and W. Huang, "Sea ice image semantic segmentation using deep neural networks," in *Global Oceans 2020: Singapore – U.S. Gulf Coast*, pp. 1–5, ISSN: 0197-7385. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9389229?casa_token=tEvCHLe3pCcAAAAA:SwGPWCxYKsLOzamYaFv4xFUprfhMbEMwiDgKRrfvVzdAMuCvCesKEWVMfS68uqpTHtijBbJKgA

[49] E. Kim, N. Panchi, and G. S. Dahiya, "Towards automated identification of ice features for surface vessels using deep learning," vol. 1357, no. 1, p. 012042, publisher: IOP Publishing. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1357/1/012042

[50] N. Panchi and E. Kim, "Deep learning strategies for analysis of weather-degraded optical sea ice images," vol. 24, no. 9, pp. 15 252–15 272, conference Name: IEEE Sensors Journal. [Online]. Available: https://ieeexplore.ieee.org/document/10480355

[51] O.-M. Pedersen and E. Kim, “Arctic vision: Using neural networks for ice object classification, and controlling how they fail,” vol. 8, no. 10, p. 770, number: 10 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2077-1312/8/10/770

[52] E. Kim, G. S. Dahiya, S. Løset, and R. Skjetne, “Can a computer see what an ice expert sees? multilabel ice objects classification with convolutional neural networks,” vol. 4, p. 100036. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590123019300362

[53] S. Sorensen, V. Veerendraveer, W. Treible, A. R. Mahoney, and C. Kambhamettu, “The polar sea ice topography reconstruction system,” vol. 61, no. 82, pp. 127–138. [Online]. Available: https://www.cambridge.org/core/journals/annals-of-glaciology/article/polar-sea-ice-topography-reconstruction-system/38F00E865E5F2BC93520D6116AE1EB53

[54] L. Maggiolo, D. Marcos, G. Moser, S. B. Serpico, and D. Tuia, “A semisupervised CRF model for CNN-based semantic segmentation with sparse ground truth,” vol. 60, pp. 1–15. [Online]. Available: https://ieeexplore.ieee.org/document/9497318/

[55] J. Li, J. K. Udupa, Y. Tong, L. Wang, and D. A. Torigian, “Anatomy segmentation evaluation with sparse ground truth data,” in *Medical Imaging 2020: Image Processing*, B. A. Landman and I. Išgum, Eds. SPIE, p. 51. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11313/2549327/Anatomy-segmentation-evaluation-with-sparse-ground-truth-data/10.1117/12.2549327.full

[56] I. Alonso, A. Cambra, A. Munoz, T. Treibitz, and A. C. Murillo, “Coral-segmentation: Training dense labeling models with sparse ground truth,” pp. 2874–2882. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2017_workshops/w41/html/Alonso_Coral-Segmentation_Training_Dense_ICCV_2017_paper.html

[57] L. Zhao, H. Zhou, X. Zhu, X. Song, H. Li, and W. Tao, “LIF-seg: LiDAR and camera image fusion for 3d LiDAR semantic segmentation,” pp. 1–11. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10128757?casa_token=wwsdcBuk7HAAAAAA:S4UBG9xy48TT9RPh8ZPi-BxXO6j2YjiytGbm5uBCKoA-AW7UrI4NYAD6Tv1OZ3HeVMiT9E8apg

[58] X. Zhang, J. Jin, Z. Lan, C. Li, M. Fan, Y. Wang, X. Yu, and Y. Zhang, “ICENET: A semantic segmentation deep network for river ice by fusing positional and channel-wise attentive features,” vol. 12, no. 2, p. 221. [Online]. Available: https://www.mdpi.com/2072-4292/12/2/221

[59] S. Sorenson, V. Veerendraveer, W. Treible, A. R. Mahoney, and C. Kambhamettu, “cPSITRES.” [Online]. Available: https://vims.cis.udel.edu/geo/ice/Home/

[60] N. Panchi, “SeaIceWeather.” [Online]. Available: https://ieee-dataport.org/documents/seaiceweather

[61] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition.” [Online]. Available: http://arxiv.org/abs/1512.03385

[62] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, “Semantic understanding of scenes through the ADE20k dataset.” [Online]. Available: http://arxiv.org/abs/1608.05442

[63] [Online]. Available: https://huggingface.co/nvidia/segformer-b0-finetuned-ade-512-512

[64] D. Schröder, D. L. Feltham, D. Flocco, and M. Tsamados, “September arctic sea-ice minimum predicted by spring melt-pond fraction,” vol. 4, no. 5, pp. 353–357, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/nclimate2203