Breaking the Ice: Video Segmentation of Close-Range Ice-Covered Waters

by

Corwin MacMillan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2024

© Corwin MacMillan 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The Arctic Ocean is experiencing significant ice recession, with projections indicating ice-free conditions during summer by 2060 [1]. This environmental shift opens new navigation routes, which could serve as a crucial trade route between the Pacific and Atlantic. Current ice navigation relies heavily on subjective decisions by ice experts, highlighting the need for tools that can assist in making objective, data-driven navigation decisions.

This dissertation explores methods for ice condition assessment using ship-borne optical data, focusing on the application of machine learning techniques. We investigate several neural network architectures for the semantic segmentation and classification of ice, specifically aiming to develop a network that is robust to occlusions (e.g., droplets on the lens) and capable of inferring ice condition in occluded regions. We create a novel medium-sized dataset of 946 images with fine annotations and provide our semi-automated approach in order to create large finely annotated dataset. We train an ensemble of traditional convolutional neural networks (CNNs) and show their performance ability on our finely annotated dataset. Finally, we use a modification of SegFlow [2], integrating features from PWCNet [3] and ResNet [4], to leverage temporal knowledge and decrease error from lens occlusions. By adjusting the network, we boast improved segmentation performance in both occluded and non-occluded data from baseline approach.

Our methodology demonstrates the potential of neural networks to provide robust ice condition assessments, aiding the pursuit of objective and reliable evaluations of ice conditions. By leveraging machine learning techniques, this research can contribute to safer and more efficient navigation in the increasingly accessible Arctic waters, thereby supporting the development of navigation tools that can keep pace with the changing environmental landscape.

Acknowledgements

I would like to extend my gratitude to the National Research Council Canada for their support and contributions to the development of this work. I wish to thank my supervisors Zhao Pan and Andrea Scott for their kindness, guidance, and patience. I would also like to thank Richard Duan, Michelle Chung-Si Chan, Maksym Veremchuk, and JD Zhu for their respective contributions to this project.

Table of Contents

Author's Declaration i

Abstract ii

Acknowledgements iii

List of Figures vi

List of Tables x

1 Introduction 1

1.1 Motivation 1

2 Dataset and Annotation 4

2.1 Dataset Overview 4

2.2 Related Works 5

2.3 Region-Based Annotation 9

2.3.1 Defining Regions 10

2.3.2 Semi-Manual Annotation 16

3 Classification and Segmentation Methodology 23

3.1 Related works 24

3.1.1 Semantic Segmentation . . . 24
3.1.2 Neural Networks for Ice Segmentation . . . 26
3.1.3 Optical Flow . . . 27
3.1.4 Video Semantic Segmentation . . . 29
3.2 Baseline Semantic Segmentation . . . 30
3.3 Video Semantic Segmentation . . . 32
3.3.1 Temporal Knowledge Distillation . . . 34
3.3.2 Joint Optical Flow and Segmentation Model . . . 35

4 Results and Discussion **38**
4.1 Semantic Segmentation . . . 38
4.1.1 Occlusion Mitigation . . . 40
4.2 Video Semantic Segmentation . . . 45
4.2.1 Temporal Knowledge Distillation . . . 45
4.2.2 Joint Optical Flow and Segmentation Model, UPerFlow . . . 49
4.2.3 Occlusion Mitigation . . . 52

5 Conclusion **55**
5.1 Future Works . . . 56

Bibliography **58**

APPENDICES **68**

A Training and Validation Curves **69**
A.1 Optuna Hyperparameter Validation Curves . . . 69
A.2 UPerFlow Training and Validation Curves . . . 70

List of Figures

2.1 GPS data (outlined in red) showing the capture locations of our dataset. The dataset tracks the Canadian Coast Guard icebreaker "Henry Larsen" off the coast of Newfoundland from April 23 to April 29, 2015. 5

2.2 Examples of images from our dataset with different lighting and ice conditions. (a) shows sparse ice floe coverage and (b) shows similar ice coverage with different lighting. (c) shows a common occlusion in our data, where a water droplet on the lens partially obscures the image. (d) shows an example frame from our test dataset with iceberg footage. (e) & (f) show examples from our training dataset, with varied ice floe coverage. (g) shows an example from our validation dataset. (h) shows (g) as it was shown in NVIDIA's 2021 keynote [41]. 6

2.3 A sample image from our dataset and part of its labeling process. (a) shows the original image, and (b) shows the image with the ship (red), sky (light green), and iceberg (purple) manually labeled (with corresponding color). (c) shows the different regions in the image, separated by dashed color lines (magenta line for near- and mid-field, and green line for mid- and far-field, respectively), and a centerline is included in the center of the image for clarity. 11

2.4 A diagram demonstrating the relation between distance in the real-world and how it is projected onto the image plane. Equal distances in the region of interest are represented with dashed colored lines, which all converge to the focal point of the lens. The distance between the image plane and the focal point is the focal length, and the angle of perspective of the camera is denoted as α. The visible span of the region of interest (ROI) is determined by the vertical field of view of the camera, represented by β. The furthest point visible in the ROI is denoted as z_∞, while the corresponding point in the image plane is denoted y_∞. The camera is included for illustrative purposes. 13

2.5 Region-based annotation for ice floes on the same sample image from our dataset. (a) shows the result of Otsu's algorithm on the different regions of our image, where the brighter classes are represented in yellow. (b) shows the result of picking the brightest classes from Otsu's algorithm to represent the ice floe class, shown in white. (c) shows the final ice floe classification, after the result is filtered to remove all floes under a certain area. 18

2.6 Region-based annotation for water on the same sample image from our dataset. (a) shows the threshold gradient used on the sample image in order to classify the water. In the near-field, pixels with a intensity less than 100 are labeled as water. This threshold gradually increases to 340 in the far-field, meaning that all unlabeled pixels in the far-field are labeled as water. (b) shows the final labeled image and (c) overlays this label on the original image for clarity. 20

2.7 Examples of labeled images from our dataset. The original images are on the left, and the labels are on the right and overlaid on the original images for clarity. (a) & (c) are examples from the training dataset. (e) is an example from our validation dataset. (g) is an example from our test dataset. . . . 22

3.1 Extreme examples of data augmentations from our data augmentation pipeline. (a) is the original unaltered image. (b) shows the randomly cropped image with augmentation factors of saturation 0.2, contrast 0.5, rotation -5, gaussian blur standard deviation 0.5, random erasing scale 0.33, random erasing ratio 2.5, random erasing gaussian blur standard deviation 15. (c) shows a different crop with saturation 1.8, contrast 1.5, rotation 20, gaussian blur standard deviation 2, random erasing scale 0.02, random erasing ratio 0.3, random erasing gaussian blur standard deviation 100 33

3.2 An illustration of the temporal consistency loss [93]. (a) shows a simplified flow chart of the method. (b) shows an example of the method on our dataset. Highlighted in the red box is an inconsistent prediction, where a classification for iceberg is shown in one frame but not the other. This classification is penalized, since it does not match the optical flow prediction, shown in the bottom. The arrows of the optical flow show the direction of the flow, and the color shows the magnitude in pixels per frame. 36

4.1 The confusion matrix for UPerNet. The network performs very accurate segmentations for almost all classes. However, performance sharply degrades when classifying brash ice, which causes average performance to drop considerably. 39

4.2 Examples of predictions from UPerNet for our dataset. The original images with the network predictions overlaid are on the left, and the labels are on the right. Figures (a)-(f) are examples from the test dataset. The results are accurate overall, but we see some common mistakes, such as misclassifying the iceberg reflection. 41

4.3 Examples of predictions from UPerNet in the presence of occlusion. The predictions for light occlusions are on the left, and the predictions for heavy occlusions are on the right. (g)-(h) show the performance of the network on unlabeled data with water droplets on the lens. 44

4.4 Flow features for a given image in our training dataset. The flow predictions are for a 2 second time difference. (a) shows the first image for the flow prediction. (b) shows the flow prediction. (c) shows the warped features given the predicted flow. (d) shows the occlusion mask calculated from the warped image. (e) shows the warped label for the following image, with dissimilar regions from the occlusion mask blacked out. (f) shows the masked warped label overlaid on the original image (a). 46

4.5 The confusion matrix for UPerNet trained with temporal consistency loss. Compared to Figure 4.1, the network performs better for most classes, but severe degradation in brash ice prediction results in a worse performance overall. 48

4.6 The confusion matrix for joint model UPerFlow. The network sees increased performance for the same classes as with the temporal consistency loss, without degradation of brash ice prediction, shown in Figure 4.5. The overall performance is better than baseline in Figure 4.1. 50

4.7 Examples of predictions from joint model UPerFlow for our dataset. We see similarly accurate predictions as baseline approach in Figure 4.2, but suffer significantly less from common errors, such as the reflection of the iceberg. 51

4.8 Examples of predictions from UPerFlow in the presence of occlusion. The predictions for light occlusions are on the left, and the predictions for heavy occlusions are on the right. Compared to baseline approach in Figure 4.3, we see less error as our approach is able to infer information in occluded regions using temporal context. (g)-(h) show UPerFlow on unlabeled data occluded with water droplets. UPerFlow is able to leverage temporal data to create predictions, even when presented with real-world occlusion. . . . 54

A.1 Validation mean Intersection over Union (mIoU) curves for our large bayesian parameter search [100] using Optuna [101]. The best performing network was UPerNet with ResNet101 backbone, with 0.828 mIoU on validation dataset. 69

A.2 The training and validation curve for our video segmentation network, UPerFlow. Our network completes training in in 637 epochs, with best validation performance 0.825 mIoU on validation dataset. Although our network performs slightly worse for validation dataset, it performs better on the test dataset, indicating that the network may have better generalization properties. 70

List of Tables

4.1 Performance metrics of UPerNet on the artificially occluded datasets. (Light) represents its performance on the lightly occluded dataset, and (Heavy) represents the heavily occluded dataset. 42

4.2 Performance metrics for the different UPerNet configurations. (F) signifies training augmented with temporal consistency loss. 49

4.3 Performance metrics of UPerFlow on the artificially occluded datasets. The performance of UPerNet baseline were included for reference. UPerFlow outperforms baseline approach for all occlusion levels, and we see less performance degradation as a result of occlusion. 52

Chapter 1

Introduction

1.1 Motivation

Arctic ice has shown a rapid and irrefutable trend of coverage recession and thinning over the past few decades. By some projections, we can expect to see ice-free conditions in the summer by as early as 2060 [1]. Although this may have negative environmental implications, it has the benefit of opening the Arctic for navigation. Of particular interest is the opportunity to use the Arctic as a trade route between the Pacific and Atlantic, providing alternatives to the currently relied upon Panama Canal through the Northern Sea Route, the transpolar route, and eventually the northwestern passage [5]. Currently, ice navigation relies on experts to interpret and integrate data from ship radar, satellite imagery, and other tools to make subjective decisions about safe navigation conditions. There exists a need for tools that can assist these experts by providing objective, data-driven assessments of navigational safety.

There are several factors that determine the overall "safeness" of the ice conditions surrounding a ship vessel. Intuitively, factors such as the amount of ice coverage and the size of ice pieces have a direct correlation to the safeness of the ice condition. However, additional factors such as the age of the ice (e.g., first-year ice versus multiyear ice), and the pressure caused by the movement and joining of ice floes pose perhaps a larger threat to the safety of the ship [6]. To properly assess these factors, researchers seek to use various tools to create an objective measure of the ice condition. One such tool is ice radar, where X-band radar can be used to identify icebergs, floes, and pressure ridges and determine the age of the ice [7]–[10]. Other works use accelerometer-based tools to measure ice concentration by recording the vibrations of the ship [11], [12]. Airborne vehicles such as helicopters or

drones equipped with LiDAR (Light Detection and Ranging), Electromagnetic Induction Sensors, or Ground Penetrating Radar have also been shown to be able detect and measure ice thickness, age, and concentration [13]–[19]. These tools provide comprehensive data that enable a more accurate and holistic assessment of the ice conditions, assisting ice experts in making informed and objective decisions regarding safe navigation routes.

However, while these physical tools have proven effective in certain contexts, they are not without limitations. One of the primary drawbacks is their speed and/or resolution, which can impede their suitability for real-time navigation decision-making in dynamic Arctic environments. LiDAR systems, for instance, may require significant processing time to generate detailed ice maps, particularly over large spatial extents. Similarly, radar-based techniques often exhibit lower resolution compared to optical imaging methods, limiting their ability to discern fine-scale features of ice cover. Furthermore, the operational constraints associated with these tools, such as the need for favorable weather conditions and line-of-sight visibility, can further hinder their utility in remote and harsh Arctic conditions.

To resolve the limitations of these techniques, many researchers employ optical sensors to assess the ice condition. Ship-borne camera data provides a large field of view to provide high-resolution information of the ice condition surrounding the ship. Optical data can also be captured at a high frame-rate, to provide dense temporal context. In order to classify the different ice objects in the images, many researchers employ unsupervised segmentation techniques, including K-means clustering, Otsu's method, and thresholding [20]–[30]. However, despite their utility, unsupervised segmentation techniques suffer from significant reliability issues that limit their effectiveness in practical applications.

One of the main challenges with unsupervised segmentation methods is their sensitivity to variations in image characteristics, such as illumination conditions, contrast levels, and noise levels. Since these techniques operate without prior knowledge of the underlying image content, they rely solely on statistical properties of the image data, which may not always correspond to meaningful distinctions between ice and non-ice regions. As a result, unsupervised segmentation algorithms often produce inconsistent and inaccurate results, particularly in complex scenes with heterogeneous ice cover and diverse surface conditions.

Furthermore, unsupervised segmentation methods typically require substantial manual tuning of parameters, such as the number of clusters in K-means clustering. The selection of appropriate parameters is subjective and can vary depending on the specific characteristics of the input data, leading to suboptimal segmentation outcomes. Moreover, these techniques lack the ability to incorporate contextual information or domain-specific knowledge, which is crucial for accurate ice detection in Arctic environments.

Neural networks, inspired by neuron activation receptors in the brain, were developed to address the lack of contextual information of unsupervised techniques. By layering multiple "neurons" in a network, neural networks were shown to be able to learn multi-dimensional patterns to create classes for data. In the field of optical data, researchers employ convolutional neural networks (CNNs) that learn convolutional weights to capture spatial information in an image [31]. A few works are interested in the segmentation and classification of ship-borne optical ice data using well-established techniques in the field of semantic segmentation [32]–[40]. However, their works share some limitations. Since there are no public datasets for sea ice segmentation, researchers annotate their own datasets, which tend to be small or use coarse label annotations. Furthermore, limited works are interested in the mitigation of erroneous predictions from lens occlusions, which present themselves frequently as water droplets on the lens. Finally, there are no currently available works that seek to leverage temporal information in order to improve accuracy in occluded regions. Our contributions in this work are as follows:

1. We develop a medium-sized finely annotated dataset that is interested in classifying ice floes, brash ice, water, ship, sky, and iceberg. We also outline our labeling methodology such that future works can create larger finely annotated datasets.

2. We assess pre-existing image semantic segmentation networks and create classifiers that are able to create fine annotations with 94% accuracy and 0.835 mean intersection over union.

3. We improve upon existing video semantic segmentation architecture to leverage temporal information and create accurate predictions in occluded regions.

Our work will be organized as follows: Section 2 outlines our dataset, our labeling methodology, and related works interested in ship-borne sea ice annotation. Section 3 outlines our methodology for training and evaluating our segmentation networks, and provides a brief overview of image and video segmentation techniques and previous approaches applied for ship-borne sea ice imagery. Finally, we provide our results and discussion in Section 4, and summarize our work and provide suggestions for future works in Section 5.

Chapter 2

Dataset and Annotation

2.1 Dataset Overview

Our dataset follows the Canadian Coast Guard's icebreaker "Henry Larsen" during her expedition off the Newfoundland coast in the spring of 2015. The areas where the footage was captured are shown in Figure 2.1 and outlined in red. It presents a realistic collection of data, featuring varying conditions that range from ideal to messy due to factors such as lighting, lens obfuscations, and environmental conditions. Notably, our dataset includes footage of an iceberg, and to our knowledge, is the first to provide and annotate in situ iceberg footage from onboard an icebreaker. Interestingly, a glimpse of this footage is featured in Jensen Huang's 2021 keynote [41], where it is highlighted as a new frontier application for machine learning, likening the challenge to that of lunar exploration. In the context of ice situation assessment, our dataset is an invaluable resource due to the diverse, realistic, and unique data it provides. Examples of our dataset can be seen in Figure 2.2.

Our images are taken with a GoPro camera and are all taken from the same elevated perspective on board the ship. This elevated perspective allows for a complete view of the bow of the ship and its immediate surroundings, and ice objects can be seen to extend to the horizon. near-field ice objects exhibit distinct features, facilitating easy identification and differentiation based on their shape and texture. However, as distance increases, details diminish, and individual ice objects become more challenging to discern due to size reduction and atmospheric effects. This variation in visibility emphasizes the need for region-based annotation, where labeling is tailored to the object's visibility and distance from the camera. Initially, we categorize our image into three regions dubbed the near-

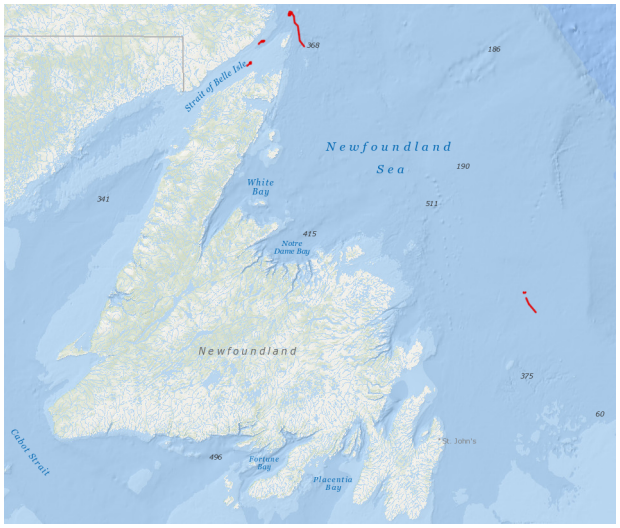


Figure 2.1: GPS data (outlined in red) showing the capture locations of our dataset. The dataset tracks the Canadian Coast Guard icebreaker "Henry Larsen" off the coast of Newfoundland from April 23 to April 29, 2015.

field, mid-field, and far-field; however, we adapt the number of regions for our images to accommodate significant changes in lighting or contrast.

2.2 Related Works

Limited works are interested in the annotation and classification of near-field ice objects, especially sea ice. This can be largely attributed to the lack of publicly available datasets, since the ability to record sea ice is limited to those in polar regions. However, another significant setback lies in the challenge of annotating sea ice data. In a single image, there can be hundreds of separate instances of ice classes in an image, causing ice data to be extremely expensive to annotate. When presented with expensive data to label, researchers often have to choose whether to allocate large resources to annotate expensive data by hand or use unsupervised methods that are quick but perhaps unreliable. In this work, we present an intuitive methodology that combines both approaches, aiming to be more reliable than many unsupervised techniques and faster than manual annotation.

To our knowledge, Dowden et al. [32], were the first to suggest deep learning models for sea ice segmentation and ship navigation. In their work, they manually annotated video data from the Nathaniel B. Palmer Antarctic expedition [42] to create a dataset of

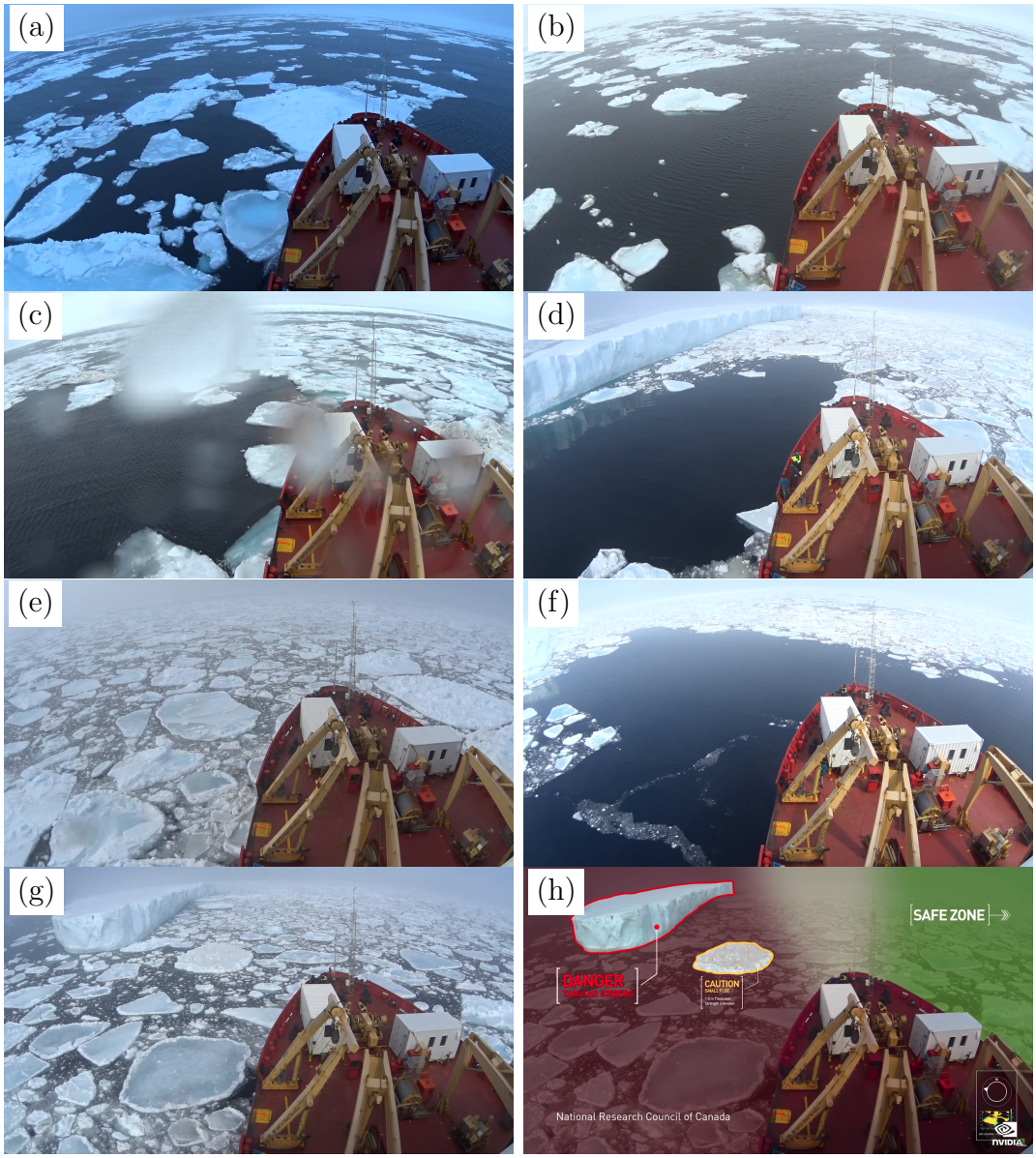


Figure 2.2: Examples of images from our dataset with different lighting and ice conditions. (a) shows sparse ice floe coverage and (b) shows similar ice coverage with different lighting. (c) shows a common occlusion in our data, where a water droplet on the lens partially obscures the image. (d) shows an example frame from our test dataset with iceberg footage. (e) & (f) show examples from our training dataset, with varied ice floe coverage. (g) shows an example from our validation dataset. (h) shows (g) as it was shown in NVIDIA's 2021 keynote [41].

1090 images using PixelAnnotationTool [43]. Although their dataset is relatively large, the high cost of manual annotation results in coarse annotations for some images, where entire fields of ice floes may be labeled as a single entity rather than segmenting each floe individually. Subsequent studies have attempted to classify the same Nathaniel B. Palmer expedition data, either using the original annotations [34], [35] or recreating their own. Asharay et al. [33] emphasized the importance of raindrop removal for model performance on the Nathaniel B. Palmer dataset, and proposed bilateral smoothing and canny edge detection as solutions. However, their segmentation dataset contained less than half the number of annotated images as Dowden et al.

Li et al. [36] re-annotated the dataset using the annotation tool LabelMe [44] to create a dataset of similar size, but their data also suffered from low resolution and did not differentiate between ice floes and new ice in expansive fields. Other studies have aimed to achieve similar results with their own annotated datasets. Zhang et al. [37] created a high-resolution dataset using LabelMe, but it comprised only 300 images. Zhao et al. [38] also used LabelMe, but their dataset comprised of 308 coarse annotations. Panchi et al. [39] used Pixel-Tool [45] to create a small dataset of 338 images, and Cai et al. [46] employed an unspecified open-access annotation tool for a dataset of 500 images.

In the domain of machine learning for sea ice segmentation, all current approaches struggle with datasets that either have coarse annotations or are too small. Our approach combines unsupervised techniques with manual labeling to create a mid-sized, high-resolution dataset of about 1,000 images. We outline this approach in the hopes for future works to generate more comprehensive data for sea ice segmentation.

There are several established methods for segmenting ice (sea ice or river ice) using unsupervised techniques, with most works utilizing remote sensing data. These data typically come from a bird's-eye perspective rather than a ship's perspective, which does not suffer from the same atmospheric effects. The top-down viewing angle ensures ice objects to have similar length-scales with similar lighting conditions, allowing for the use of global segmentation algorithms. While these methodologies have their advantages, they also have significant drawbacks that make them less appealing for ship navigation. Remote sensing data is often low-resolution because the images are taken from a large distance, especially in the case of satellite data. Additionally, the data has high latency, limiting its use for real-time navigation. For effective real-time navigation, shipborne optical data is necessary. Although methodologies for remote sensing are very similar to ours in terms of objective, the differences in lighting conditions and viewing angles limit their applicability to ship perspective data. However, due to the shared objective, we expect methodologies in both fields to be similar or inspired by each other. A brief outline of some common methodologies for remote and near-field ice imagery will be given.

A common algorithm used for remote sensing and near-field ice data is K-means [47]. K-means aims to organize the given data into K clusters by choosing clusters in the feature space based on their Euclidean distance to K randomly placed centroids. Since the algorithm creates clusters based on Euclidean distance, the algorithm works well when data classes in the feature space are closely related. Since ice objects tend to be white and distinctly colored from their surrounding environment, K-means is expected to work well. Extensive works have employed K-means for ice segmentation and have observed varying degrees of success [20]–[24]. As such, some works have applied the methodology for near-field ship data [25], [26]. However, near-range imagery are much higher resolution than remote applications, and ice exhibits a much larger range of texture features and color variations. This array of features presents a significant challenge for K-means, and results in inaccurate and/or coarse annotations.

To combat the unreliability of K-means clustering, researchers often leverage specific features of the ice to identify ice objects in an image. Stewart et al. [48] segment Synthetic-Aperture Radar (SAR) data by modeling the probability of ice classes as a function of intensity and the shape of the floes. Deng et al. [49] segment ice and differentiate types of ice using Markov Random Fields (MRF) that optimally leverage spatial information alongside pixel intensity and/or region texture. These methodologies require an accurate model to portray the probability distribution of the ice. Both Stewart et al. and Deng et al. suggest using a gamma distribution, which is intuitive since ice classes represent a large proportion of the bright classes in an image. However, the gamma distribution is ill-suited for oblique perspective data, where the bright classes in the far-field represent a disproportionate amount of the bright classes in the image, thereby degrading classification accuracy in the near-field. Accurate depiction of ice in oblique images requires a more sophisticated model that incorporates distance from the camera. In our work, we segment our image into multiple regions based on distance and segment each region separately. This approach allows us to address the varying brightness, contrast, and texture conditions within different parts of the image more effectively.

Otsu's algorithm is a histogram-based technique that creates classes using pixel intensity values [50]. The algorithm works on grayscale images and chooses thresholds in a histogram based on regions that minimize intra-class variance. In its simplest form, the algorithm is a binary classifier that operates under the assumption that the data is a mixture of two normal distributions with equal variance and sample size. It works well with clean homogeneous data with high inter-class variance and low intra-class variance [51]. Otsu's algorithm can also be extended to create multiple classifications [52], and we employ MATLAB's *multithresh()* function. Several works employ Otsu thresholding for ice segmentation, both in remote [27] and near-field [28], [29] applications. Kalke et al. [30]

trained Support Vector Machines (SVMs) to classify river ice and found that SVMs performed better than Otsu's thresholding and other automatic clustering techniques, while performing similarly to manual thresholding. In our approach, we use multiclass Otsu's thresholding to generate multiple proposals for the correct threshold and then manually select the best threshold. This allows us to leverage the efficiency of automated thresholding while ensuring the correctness of manual thresholding.

Similar to our work is that of Haverkamp et al. [53], who propose separating SAR images into multiple sub-regions and applying thresholding techniques in each region. They identify regions in the image that exhibit high degrees of bimodality and use maximum likelihood estimation to find the threshold that differentiate ice. Lu et al. [54] adopt this method but apply it to perspective ship data, aiming to locate ice floes and calculate concentration. They outline calibration parameters necessary to orthorectify their data and use this to calculate concentration and determine the region of the image that represents the far-field. Since their work focuses on ice concentration, they discard far-field information in favor of higher accuracy near-field segmentations. We argue that far-field information is still important and should be used for navigation and machine learning. Although individual ice floes may not be discernible in the far-field, pertinent information about concentration or the presence of dangerous icebergs can still be estimated. Additionally, our work definitively outlines a methodology to define any number of regions in an image for segmentation, without the need for calibration. Finally, we use multilevel Otsu thresholding rather than maximum likelihood estimation, for its ability to create multiple thresholds as a proposal for the correct threshold. We manually select the appropriate threshold that results in the most accurate segmentation.

2.3 Region-Based Annotation

The Henry Larsen dataset focuses on classifying and segmenting six distinct categories, four of which are various types of ice objects defined by the World Meteorological Organization (WMO) standards [55]. These categories are:

- Iceberg: A massive piece of ice of varying shape, protruding more than 5 m above sea-level
- Ice floe: Any contiguous piece of sea ice
- Water: A large area of freely navigable water

- Brash ice: Weakly frozen together clusters of ice crystals and small fragments of floating ice no more than 2 m across
- Ship: The boat present in the near-field of all images; including the artifacts and people on board
- Sky: All visible sky in the image

While our classification includes a category labeled "brash ice", it is worth noting that this designation deviates slightly from the WMO standards. Technically, our classification could be subdivided into two separate classes: "brash ice" which describes the individual ice fragments, and "new ice" which describes weakly frozen together ice crystals. However, for navigational purposes, this distinction is deemed negligible, as neither poses a significant threat to the ship. Therefore, we amalgamate these categories, treating them as the complement to the three primary ice-type classes (i.e. not iceberg, ice floe, nor water).

Additionally, we recognize that the ice floe class can be further separated into classes based on the age of the ice (e.g. first-year ice, multi-year ice), which each have properties (e.g., thickness and strength) that can impact the ship differently. However, precise age classification requires expertise from ice specialists, a field beyond our current scope. Consequently, we leave this classification for future works. Currently, we focus our work on the fine annotation and classification of ice while minimizing error from lens artifacts.

Many works leverage distinct features of the ice in order to segment it; including shape, texture, and color. Among these, color is the most commonly used feature. We postulate that color is the most important differentiating feature of ice, and color-thresholding techniques combined with heuristics can be used in combination sufficiently to segment the ice. For oblique ship perspective imagery, we note that features in an image exhibit similar characteristics based on their distance from the camera. Thus we define distinct regions in the image based on their distance and lighting and annotate our images in an approach we call Region-Based Annotation.

2.3.1 Defining Regions

Figure 2.3a shows an example of an image from our dataset that contains all 6 relevant classes. Our images are taken with a GoPro camera and are all taken from the same elevated perspective on board the ship. This elevated perspective allows for a complete view of the bow of the ship and its immediate surroundings, and ice objects can be seen to extend to the horizon. Near-field ice objects exhibit distinct features, facilitating easy

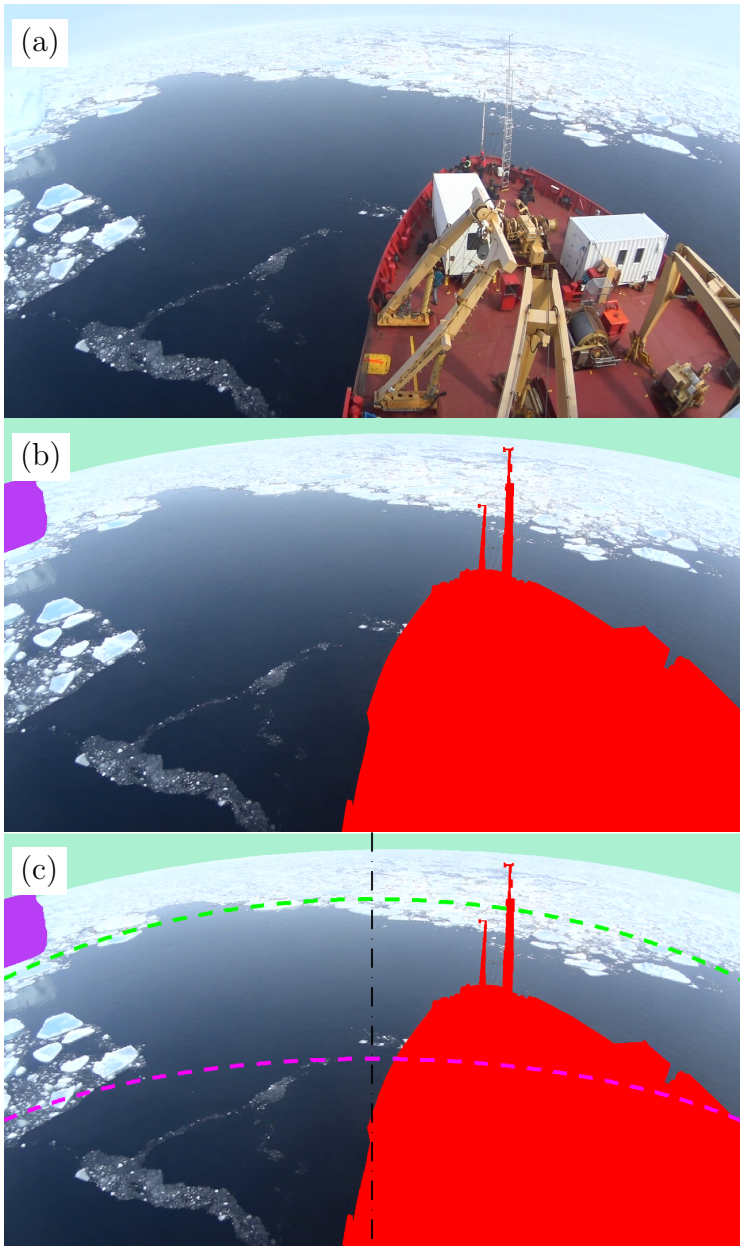


Figure 2.3: A sample image from our dataset and part of its labeling process. (a) shows the original image, and (b) shows the image with the ship (red), sky (light green), and iceberg (purple) manually labeled (with corresponding color). (c) shows the different regions in the image, separated by dashed color lines (magenta line for near- and mid-field, and green line for mid- and far-field, respectively), and a centerline is included in the center of the image for clarity.

identification and differentiation based on their shape and texture. However, as distance increases, details diminish, and individual ice objects become more challenging to discern due to size reduction and atmospheric effects. This variation in visibility emphasizes the need for region-based annotation, where labeling is tailored to the object's visibility and distance from the camera. Initially, we categorize our image into three regions dubbed the near-field, mid-field, and far-field; however, we adapt the number of regions for our images to accommodate significant changes in lighting or contrast.

In real-world coordinates, a system that defines regions based on their distance would be best defined using a polar coordinate system, where each point on the plane is described by its distance and angle from the origin. However, such a coordinate system may not be accurate with our images, since we expect our images to be a skewed representation of the real world. Factors such as the perspective angle of the camera as well as the physical properties of the camera lens both cause our captured images to be skewed [56]. Traditionally, to account for this skewing of images, one would use known measurements of multiple features in an image in order to create a linear mapping that describes the transformation between the real world and the image [57]. The mapping that depicts the skewing from perspective is described with the extrinsic properties of the camera, and the mapping from the physical properties of the lens is described with the intrinsic properties. Knowledge of this mapping allows for the size of all objects in an image to be known and would allow us to easily define regions based on distance. However, such properties are not provided for our dataset, which poses a challenge for labelling based on distance. To overcome this, we create a rough parametric model to describe the extrinsic and intrinsic properties of our model, then tune these parameters to fit what we consider appropriately characterizes the regions in our images.

Extrinsic Properties Model. We first construct a parametric model to describe the camera's extrinsic properties. Figure 2.4 depicts the relationship we expect between the real-world and the image plane. The focal point, situated within the camera lens, serves as the convergence point for incoming light rays. The image plane is where light is projected on to form the image, and is the mathematical representation of our dataset images. The distance between the focal point and the image plane is defined as the focal length (f). Within view of the camera lies the region of interest (ROI), which represents the landscape being captured. The angle formed between the camera and the farthest point within this region is termed the angle of perspective, represented as α. The visible span of the ROI is determined by the vertical field of view of the camera, represented by β. The furthest point visible in the ROI is denoted as z_∞, while the corresponding point in the image plane is denoted y_∞.

We are interested in describing the relationship between distance in the ROI and and its

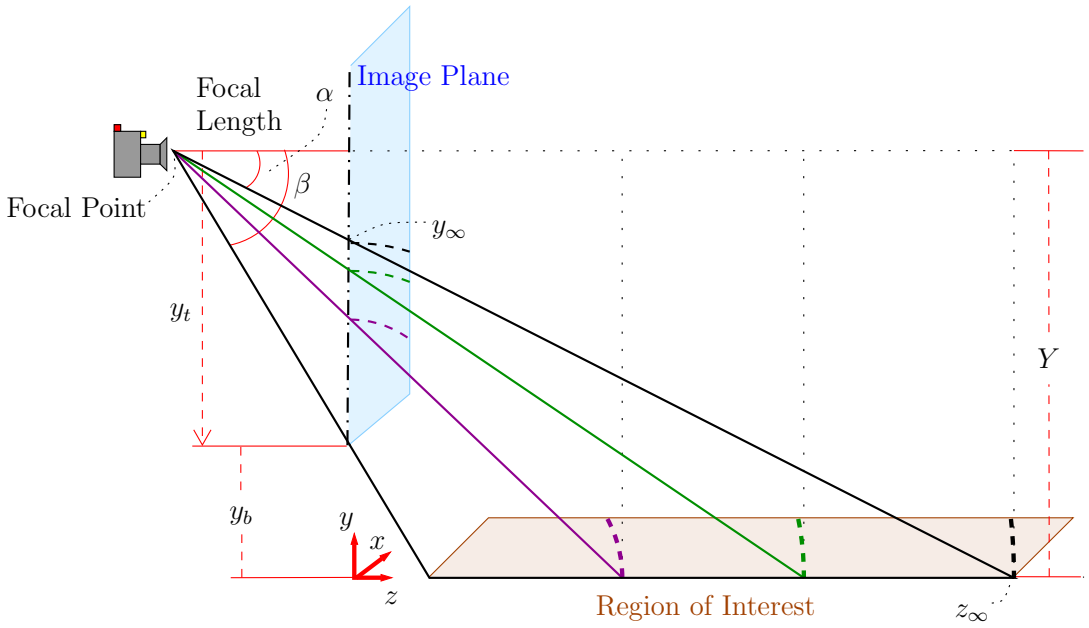


Figure 2.4: A diagram demonstrating the relation between distance in the real-world and how it is projected onto the image plane. Equal distances in the region of interest are represented with dashed colored lines, which all converge to the focal point of the lens. The distance between the image plane and the focal point is the focal length, and the angle of perspective of the camera is denoted as α. The visible span of the region of interest (ROI) is determined by the vertical field of view of the camera, represented by β. The furthest point visible in the ROI is denoted as z_∞, while the corresponding point in the image plane is denoted y_∞. The camera is included for illustrative purposes.

corresponding representation in the image plane. In Figure 2.4, we show multiple regions that are equally spaced apart within the ROI, and how they are warped in our image plane. Due to the perspective angle, these regions become compressed in the image plane and appear smaller the further away they are from the camera. Let y_t represent the vertical distance from the focal point in the image plane, Y represent the ROI's vertical distance from the camera, and z represent the ROI's horizontal distance from the camera. Using similar triangles, the relationship between y_t and z can be expressed as:

$$\frac{y_t}{Y} = \frac{f}{z+f}. \tag{2.1}$$

This equation connects coordinates in the image plane y_t with the real-world distance z, in terms of parameters Y and f. Redefining the image coordinates from the bottom of

the image and rearranging for the image coordinate y, we obtain:

$$y = Y(\gamma - \frac{f}{z+f}), \tag{2.2}$$

where $\gamma \in [0, 1]$ represents the ratio between Y and the vertical field of view of the camera. Rigorously, $\gamma := 1 - \frac{y_b}{Y}$, where $y_b \in [0, Y]$ is the vertical distance from the bottom of the image plane to the region of interest. Our final relation describes the projection of distance from the real-world to the image plane with three parameters, Y, γ, and f.

To simplify our relation, we introduce several assumptions. First, we normalize the coordinate system for easier analysis. Let z_0 and z_∞ denote the closest and furthest points in the region of interest (ROI), respectively, with corresponding image plane points y_0 and y_∞. Knowledge of the values of these points allow us to parametrize the mapping between the ROI and the image plane. Rather than working with their actual values, we normalize the coordinate system so that the coordinates of the closest and furthest points in both the ROI and the image plane range between 0 and 1. The normalized coordinates are denoted by y^* and z^*, defined as:

$$y^* = \frac{y}{y_\infty}, \tag{2.3}$$

$$z^* = \frac{z}{z_\infty}. \tag{2.4}$$

We then impose the following constraints:

$$y^*|_{z^*=0} = 0, \tag{2.5}$$

$$y^*|_{z^*=1} = 1. \tag{2.6}$$

In the normalized coordinate system, these constraints map the nearest and furthest points in the ROI to their corresponding locations in the image plane. By incorporating these constraints into Equation (2.2), we derive the following relations:

$$\gamma = 1, \tag{2.7}$$

$$f^* = Y^* - 1. \tag{2.8}$$

Here, $Y^* := \frac{Y}{y_\infty}$ and $f^* := \frac{f}{z_\infty}$, are the normalized versions of the original parameters Y and f, respectively, scaled by the distances in the image and real-world planes.

The final simplified relation describes the relation between real-world distance and how it is warped in the image plane. It mimics the physical extrinsic model given by

Equation 2.2 and depends only on a single parameter, Y^*. The final simplified relation is given by,

$$y^* = Y^*(\frac{z^*}{z^* + Y^* - 1}). \tag{2.9}$$

To create explicit regions in the image plane, we split the normalized ROI into equidistant regions before warping them with Equation (2.9). We then tune the parameter Y^* to fit our labeling needs. These labeling needs include the size of objects (their distance from the camera), as well as their location and lighting conditions. Figure 2.3c demonstrates an example of this process. In this image, the ice floes and water in the far-field are much brighter than the rest of the image, thus we tune our parameter to ensure that this region is explicitly defined for labeling. Although these regions may not actually correspond to "equal real-world distance", they fit our labeling needs while adequately describing the image's extrinsic properties. For this video, our images are separated into 3 regions for labeling, with parameter $Y^* := 3$.

Intrinsic Properties Model. Aside from perspective warping, we expect our images to be physically warped from the physical properties of the lens, known as the intrinsic properties of the camera. To account for this, we assume that the transformation for the lens's intrinsic properties can be mostly described with a linear scaling operation. This assumption requires ignoring the effects of multiple modes of lens distortion, which are usually modeled with higher order linear transformations [58]; however it provides a sufficient first order approximation for our application. If we assume the transformation between the real word and the camera coordinate system is a linear scaling transformation, then the distance to the camera can be represented by an elliptical coordinate system, which can be thought of as a skewed polar coordinate system. The equations of the elliptic coordinate system are given by [59]:

$$x = a \cosh \mu \cos \nu, \tag{2.10}$$

$$y = a \sinh \mu \sin \nu. \tag{2.11}$$

Where (μ, ν) represent curves of ellipses and the hyperbolic distance between them, respectively, (x, y) represent cartesian coordinates, and a represents the location of the ellipse foci. To convert cartesian coordinates to elliptical coordinates, we use the relation,

$$\mu = \frac{\operatorname{acosh}\left(\sqrt{(x+a)^2 + y^2} + \sqrt{(x-a)^2 + y^2}\right)}{2a}. \tag{2.12}$$

We ignore the hyperbolic parameter ν, since we are only interested in the skewed distances represented by ellipses, and the distances between the ellipses can be adequately modeled with our previous extrinsic property relation seen in Equation (2.2).

To construct our elliptical coordinate system, we first define the cartesian coordinate system origin to be at the bottom center of our image. We assume this point represents the closest point to the camera in our image. We then define every point in the image according to its position in pixel units. Finally, we tune our elliptical parameter a to fit our labeling needs, before normalizing the elliptical system to be between 0 and 1 as we did with our extrinsic model. In the case of Figure 2.3c, the parameter was tuned such that the ellipses spanned the width of the image and fit the approximate shape of the bright far-field and the ice floes within it. For this video, we chose an ellipse focus length of $a \coloneqq 800$, corresponding to a focus length around two-thirds of the 1280 pixel wide frame. Our final approach to describing regions for our data requires 2 transformations that approximate the extrinsic and intrinsic properties of the camera, each with a single tunable parameter.

2.3.2 Semi-Manual Annotation

After separating the images into different regions, we address the challenge of creating segmentations for each of our six classes. These classes are unique, each presenting distinct qualities that affect their ease of annotation. For instance, the ship is a large, singular, and distinct object in the image, and its red tint makes it easy to identify. However, the ship also contains difficult-to-label features, such as the see-through antenna or white containers that blend with the surrounding environment. Similarly, the sky and iceberg classes are singular and easy to identify but are quite similar in color to their surroundings.

In contrast, the ice floe, brash ice, and water classes appear multiple times throughout the image and have complex geometries that result from ice floes touching and separating. Despite these complexities, when comparing these three classes to each other, they are quite distinct, with bright pixels corresponding to ice floes and dark pixels to water. Therefore, we approach the annotation challenge with a multipass method: manually annotating the infrequent classes and using algorithms for the colorfully distinct classes.

While ice floes, brash ice, and water are abundant and complex, the ship, iceberg, and sky appear only once in each image, making them relatively easier to annotate manually. However, their geometries are not necessarily simple; for instance, the ship and the iceberg have intricate shapes that cannot be defined with regular polygons, increasing annotation costs. With large datasets, this cost can become significant as annotations need to be recreated for each image.

To mitigate these costs, we take advantage of the video format of our data by creating sparse annotations over time and interpolating between these frames. The positions of the ship and the sky change minimally between frames, allowing us to reuse a significant portion of the annotation labels. For our dataset, we use the Computer Vision Annotation Tool (CVAT) [60], labeling every 10th image. Any errors resulting from interpolation are manually corrected. This method efficiently balances manual accuracy with the automation of repetitive tasks, thereby reducing the overall annotation burden. An example of the manual annotation (with corresponding color) of the ship (red), sky (light green), and iceberg (purple) are shown in Figure 2.3b.

After the manual annotation of these classes, we may now annotate the water, brash ice, and ice floes. We employ a multipass approach and annotate these classes sequentially. Among these classes, ice floes in particular stand out as an ideal class to begin with. Ice floes are white and reflect light off their surface, causing them to be bright and identifiable. Similar to many previous approaches, we employ Otsu's algorithm to classify the ice floes for each region. We utilize MATLAB's *multithresh()* function, an extension of Otsu's algorithm capable of handling up to 20 classes. Using *multithresh()* for a single class allows us to create multiple recommendations for the correct threshold corresponding to the ice floes, rather than relying on the automatic label created by the binary Otsu classifier. We then manually select the threshold best corresponding to the ice floes.

In our approach we oversegment each region into 20 classes and pick the 6 brightest classes to label as ice floes. We found this number to be fairly consistent across regions and frames, however the number may vary by video. After ice floe annotation, we remove any cluster less than 700 pixels to avoid misclassifying small ice fragments. This is done using MATLAB's *bwareafilt()* function, which identifies all connected clusters in a binary image and filters them based on size. Figure2.5 shows an example of our approach. Figure 2.5a shows the multiclass otsu algorithm, with the brightest classes in yellow and the darkest in blue. Figure 2.5b shows the result after choosing the 6 brightest classes as ice floes. Figure 2.5c shows the final ice floe classification after filtering out small ice fragments.

Finally, we annotate the brash ice and water classes. The refractive properties of water make this annotation particularly challenging. As light travels from the air to the water, the amount of light that can pass through the boundary is determined by the incident angle between the light ray and the normal to the water surface. Generally, the steeper the incident angle, the more light is reflected, as described by the Fresnel equations [61]. For our context, this angle is determined by distance to the camera, since light reflected from the far-field must travel at a steep angle (relative to the water surface normal) to reach the optical sensor. This creates a color gradient of light along the water surface as the distance increases and more light is reflected. Since this color gradient changes rapidly, it becomes

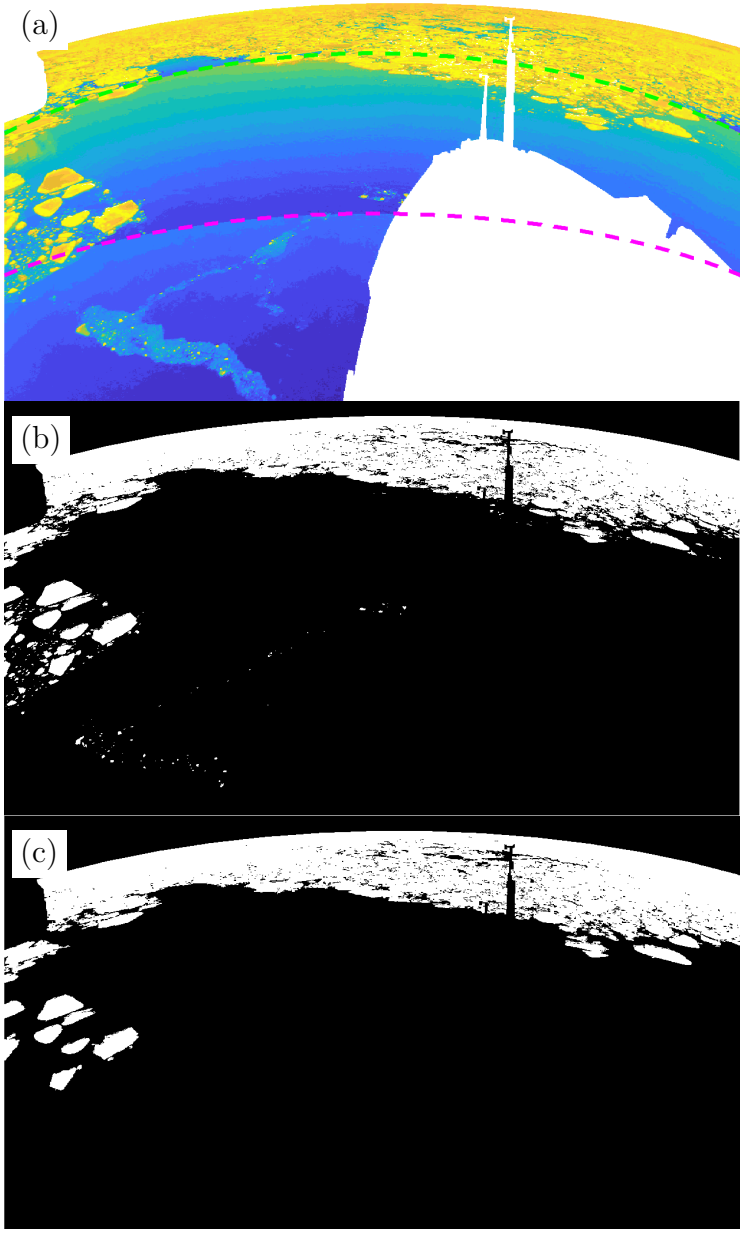


Figure 2.5: Region-based annotation for ice floes on the same sample image from our dataset. (a) shows the result of Otsu's algorithm on the different regions of our image, where the brighter classes are represented in yellow. (b) shows the result of picking the brightest classes from Otsu's algorithm to represent the ice floe class, shown in white. (c) shows the final ice floe classification, after the result is filtered to remove all floes under a certain area.

very difficult to define a finite number of regions to describe the change in light intensity based on distance. To overcome this, we interpolate a continuous range of color thresholds that describes the change of light reflection based on distance. In the near-field, the color of the water is dark and the intensity gradually increases as distance increases. For simplicity, we use a linear model to describe the relation between water pixel intensity and distance. This allows us to create a continuous range of thresholds using two parameters, the color threshold in the far-field and the threshold in the near-field. We threshold pixels along the Blue channel of the RGB image.

To tune these parameters, we initialize the values with random guesses, and change them until the profile of the water reflection is adequately described. We strongly prioritize the correct classification of water within the visible range, since contrast in the far-field is low and makes it difficult to definitively differentiate between water and brash ice. Figure 2.6 shows an example of our approach. Figure 2.6a shows the range of thresholds for the image. In the near-field, less light is reflected and all pixels with intensity less than 100 are labeled as water. We then choose the second threshold such that no instances of bright water are mislabeled. For this image, this requires a threshold of 340 in the far-field. The final annotated image is shown in Figures 2.6b and overlaid on the raw image as shown in 2.6c.

This approach was used to annotate 945 labeled images, from 3 separate videos. These videos were chosen based on their relatively clean data and balanced distribution of classes. Our dataset was split approximately 80:10:10 for training, validation, and testing, respectively, with two videos used for training and validation and one video being left out for testing. Figure 2.7 shows examples of the labels for the dataset. Our approach is able to create fine annotations through the annotation of three objects, the tuning of two geometric parameters, and five annotation parameters. This reduces annotation costs dramatically compared to manually annotating complex ice floes. We further reduce annotation costs by reusing annotations and parameters for multiple video frames. Our large, finely annotated dataset provides precise descriptions of the surrounding ice environment and enables large neural networks to learn effectively from the data.

Our annotation method has some minor shortcomings, which are a direct result of the naivety of color segmentation and require more nuanced algorithms to improve in the future. For example, achieving accurate classifications for water in the near-field using linear thresholding sometimes requires setting unrealistically high thresholds in the far-field. This can lead to erroneous predictions, where brash ice may be misclassified as water in the far-field. In Figure 2.6a, we use a threshold of 340, which is well above the maximum pixel value of 256, meaning all pixels in the far-field that may potentially be brash ice are labeled as water. However, contrast in the far-field is very low, making it challenging to

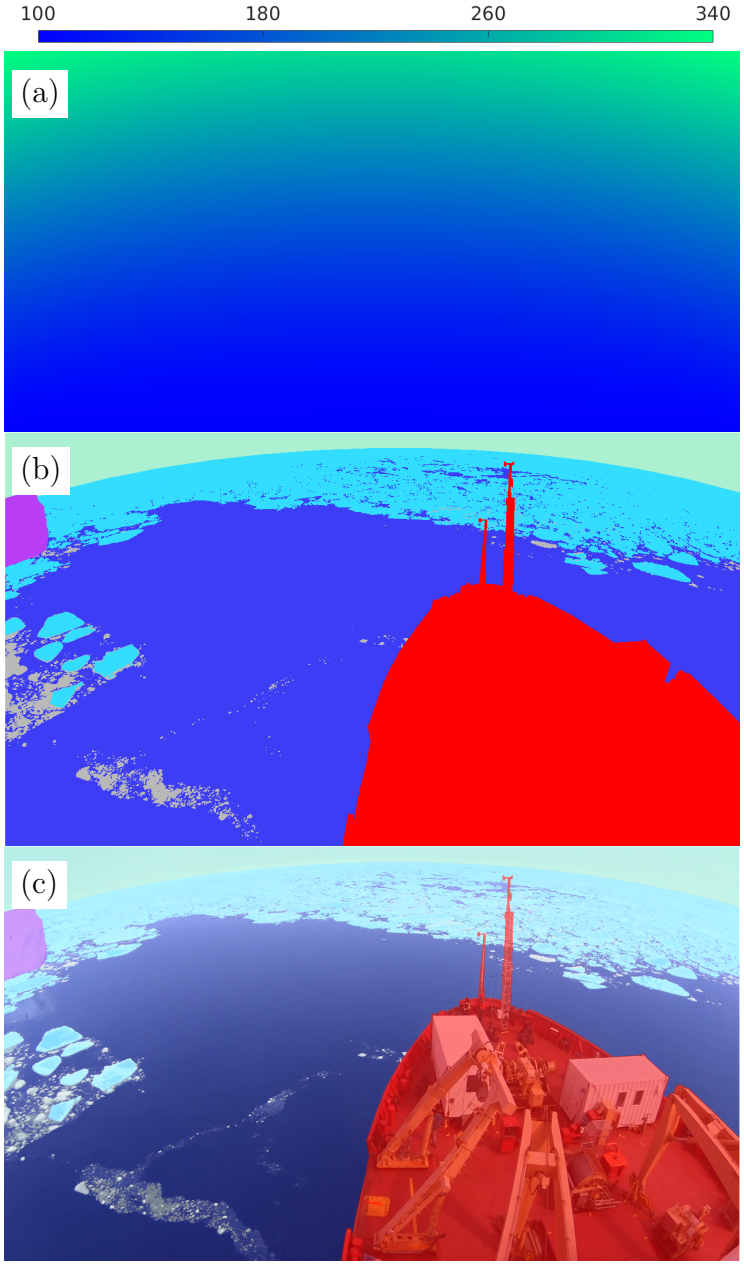


Figure 2.6: Region-based annotation for water on the same sample image from our dataset. (a) shows the threshold gradient used on the sample image in order to classify the water. In the near-field, pixels with a intensity less than 100 are labeled as water. This threshold gradually increases to 340 in the far-field, meaning that all unlabeled pixels in the far-field are labeled as water. (b) shows the final labeled image and (c) overlays this label on the original image for clarity.

definitively differentiate water from brash ice in that region. Regardless, this distinction between far-field brash ice and water may not be very important, as neither pose a threat to the safety of the ship.

Our algorithm also suffers from differentiating large clusters of brash ice from ice floes. Although the algorithm works well when brash ice is sparsely distributed, clumps of brash ice seem like one homogeneous region and cannot be identified by an algorithm that measures cluster areas. Similarly, large ice floes leaving the frame or split by the ship antenna in the image may become too small and mislabeled as brash ice. This can be seen, for example, in Figure 2.7b where part of the ice floe to the left of the short antenna is mislabeled as brash ice.

Finally, the texture of ice floes occasionally results in regions on the ice floe that are indifferentiable from brash ice in the water. Although this result is rare, an example can be seen in Figure 2.7h on the ice floe in the near field. This may be solved using algorithms that account for ice floe shape and texture, although it is unclear whether any such algorithm could work consistently. In the future, we hope to use the classifications from neural networks for the creation of new low-cost data, while correcting minor misclassifications by hand in an ever-improving feedback loop.

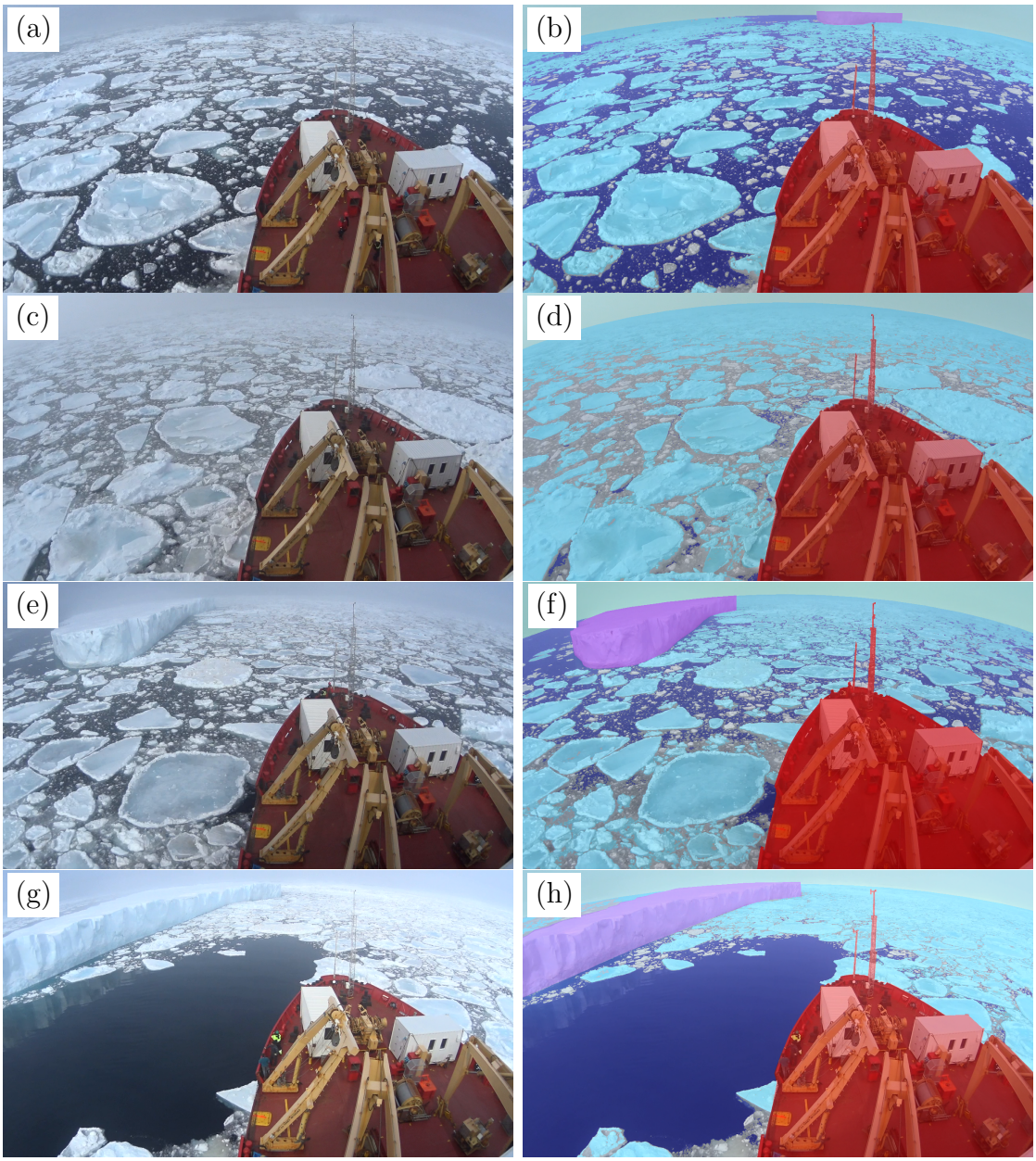


Figure 2.7: Examples of labeled images from our dataset. The original images are on the left, and the labels are on the right and overlaid on the original images for clarity. (a) & (c) are examples from the training dataset. (e) is an example from our validation dataset. (g) is an example from our test dataset.

Chapter 3

Classification and Segmentation Methodology

Our work addresses two main problems for near-range ice segmentation: high-resolution annotation and lens occlusion mitigation. We address the need for high-resolution annotation with our Region-Based methodology and Henry Larsen Dataset. However, our labeled dataset does not contain any lens occlusions, such as water droplets on the lens. This is a deliberate choice. Since we cannot definitively know what objects are behind a lens occlusion, manual annotation of occluded data would require that we physically label occluded regions as a new "occlusion" class. We emphasize that lens occlusions - such as water droplets - should be viewed purely as sources of error, and our methodology should not be interested in its classification. Instead, we should seek to use methodologies that can leverage information from multiple sources and reduce the error in the occluded region. For this, we turn to the field of video segmentation and optical flow.

However, prior to the application of some more advanced techniques, we must evaluate the performance of some well-established techniques on our dataset. Since our dataset is novel and high-resolution, it is not immediately clear which networks would perform best on our dataset. Thus, we reevaluate some of the best performing networks from previous works on our dataset, and view their performance as a new baseline to surpass. We then show that simple video segmentation networks can perform on par with image segmentation networks while creating reasonable predictions in heavily occluded regions.

3.1 Related works

3.1.1 Semantic Segmentation

Semantic segmentation, a process that involves classifying each pixel in an image into predefined categories, has seen substantial advancements with the advent of deep learning techniques. This task is critical in various applications, from medical imaging to autonomous driving, where understanding the precise location and boundaries of objects within an image is crucial. Neural networks have garnered significant attention in recent years due to their remarkable performance in these complex visual tasks, revolutionizing the field with their ability to learn from large datasets and generalize to new, unseen data.

The introduction of Fully Convolutional Networks (FCNs) by Long et al. [62] marked a significant milestone in the evolution of semantic segmentation techniques. FCNs replaced the fully connected layers in conventional Convolutional Neural Networks (CNNs) with convolutional layers, enabling pixel-wise classification and end-to-end training for segmentation tasks. This architectural change allowed for more accurate and efficient segmentation by leveraging the spatial hierarchies learned through convolutional operations.

Building upon these foundational models, SegNet [63] and U-Net [64] further improved segmentation accuracy by incorporating encoder-decoder architectures. These architectures featured symmetric connections between the encoder and decoder paths, facilitating better spatial information recovery and precise boundary delineation. The encoder captured abstract features while the decoder reconstructed spatial details, ensuring high-quality segmentation outputs.

ResNet, or Residual Networks, introduced by He et al. [4], has been a groundbreaking architecture in the field of deep learning, significantly improving the training of very deep networks. The key innovation of ResNet is the introduction of residual blocks, which allow the network to learn residual functions with reference to the layer inputs, rather than learning unreferenced functions. This addresses the problem of vanishing gradients, making it feasible to train much deeper networks.

ResNet has been widely adopted as a backbone for many segmentation models due to its strong feature extraction capabilities and ability to learn complex representations. By maintaining the flow of information through skip connections, ResNet enables the effective training of networks with hundreds or even thousands of layers. This robustness and depth make it an ideal foundation for advanced segmentation frameworks.

One significant advancement in segmentation models that builds upon ResNet is the DeepLab series by Chen et al. [65]–[67]. DeepLab introduced several innovative techniques

to enhance segmentation performance. Atrous Convolution, also known as dilated convolution, allowed for the expansion of the receptive field without losing resolution, thereby capturing multi-scale contextual information. The introduction of Spatial Pyramid Pooling aggregated features at multiple scales, significantly enhancing the ability to segment objects at different scales and improving the robustness of the segmentation models.

The Pyramid Scene Parsing Network (PSPNet), introduced by Zhao et al. [68], made a significant impact on the field by addressing the challenge of capturing both local and global context in segmentation tasks. PSPNet employs a pyramid pooling module that aggregates contextual information from different regions of the image at multiple scales. This approach enables the model to understand the scene's overall structure while preserving fine details, leading to improved segmentation performance, especially in complex scenes.

The Unified Perceptual Parsing Network (UPerNet), proposed by Xiao et al. [69], represents another advancement in semantic segmentation by integrating multiple perceptual parsing tasks into a single framework. UPerNet leverages a Feature Pyramid Network (FPN) to fuse features from different levels of a backbone network, enhancing the model's ability to capture both high-level semantics and low-level details. This multi-level feature fusion approach allows UPerNet to achieve state-of-the-art performance in various segmentation benchmarks, demonstrating its robustness and versatility.

To train these large networks, researchers often employ a combination of transfer learning and data augmentation, which allows them to leverage pre-existing models and enhance the diversity of their training data. Transfer learning involves fine-tuning models that have been pre-trained on large datasets, such as ImageNet [70], to adapt them to the specific task at hand. This significantly reduces the computational resources and time required to train deep networks from scratch and has been shown to imrpove network performance on the new dataset [71]. Data augmentation techniques, such as rotations, flips, and rescaling artificially increase the size of the dataset by creating varied versions of the original data. These strategies not only improve the network's ability to generalize to new data but also mitigate overfitting, leading to more robust and reliable segmentation performance [72].

In our approach, we evaluate FCN, DeeplabV3, DeepLabV3+, UNet, PSPNet, and UPerNet with different sized ResNet backbones as a baseline. We show that building off these models to include temporal information can improve performance, especially in the presence of occlusion.

3.1.2 Neural Networks for Ice Segmentation

The application of semantic segmentation for near-field ice identification is an emerging field. Limited existing works use a subset of the aforementioned frameworks with slight modifications for improved performance. Dowden et al. [32] were the first to suggest using deep learning for ship-based ice segmentation and train SegNet and PSPNet to demonstrate their capability to segment and classify ice. Following their work, several authors suggest improvements through their own machine learning approaches on the same dataset, by using more accurate or faster networks. Asharay et al. [33] improved the performance of PSPNet by removing water droplet artifacts on the lens through bilateral smoothing and canny edge detection. In a subsequent work [34], they employed conditional generative adversarial networks and show that the networks outperform without their previous data augmentation approach. Balasooriya et al. [35] utilized DeeplabV3+ for mobile applications, demonstrating that this network was both more accurate and faster than PSPNet. Li et al. [36] suggested integrating attention modules into the DeepLabV3+ architecture, improving the network's performance compared to the vanilla model. Other works seek to achieve similar results on their own annotated datasets. Zhang et al. [37] also used DeeplabV3+ with attention to segment ice on their self-annotated dataset. Zhao et al. [38] proposed using a modified UNet, while Panchi et al. [39] suggest using PSPNet, PSPDenseNet, DeepLabV3+, and UPerNet in an ensemble for ice segmentation, showing that the ensemble outperformed the individual.

In Section 2.2, we discussed the shortcomings of these works in terms of their datasets. Existing datasets are typically small and/or low in resolution to adequately encapsulate the variance and dynamics of sea ice data. In addition, some methodologies are also flawed in that they often seek to label idealistic data that are not subject to realistic sources of error, such as raindrop lens occlusions. Dowden et al. [32] was the first to try and address this issue. They classify any particles or objects on the camera lens that obstruct the image as "lens artifacts", and label these regions directly. However, there is no suggestion on how to address and mitigate the loss of information in this portion of the image. Asharay et al. [33] propose artificially removing raindrops from the image using a raindrop detection network, and a combination of smoothing and edge detection filters. They first train a CNN to detect whether raindrops are present in an image. If raindrops are present, they then apply a bilateral smoothing filter to remove raindrops and Canny edge detection to restore the boundaries of occluded ice floes. This methodology is a form of spatial interpolation, where neighbouring regions are used to smooth out missing information in occluded regions. Thus, this methodology's ability to infer occluded regions inherently relies on small translucent raindrops where hidden

information can still be partially discerned, or environments with simple geometry. In their most recent work [34], they demonstrate that conditional generative adversarial networks can be taught to create accurate classifications even with the presence of lens occlusions. To train the network, they present the network with ship data in a variety of weather conditions such that their network may learn to be robust against these occlusions. However, their approach is limited by the data they use to train their network: since their ground truth data contains occlusions, the accuracy of their model relies on the annotator's ability to infer the shape of the ice floes behind the occluded regions.

The work of Panchi et al. [40], is also interested in the segmentation of weather-degraded images. They create a dataset of artificially occluded shipborne images using a pair of adjacent cameras, one with water droplets on the lens and one without. They also artificially augment the clean images to simulate different arctic weather conditions. They test two approaches to remove the occlusions and segment the ice data: using pretrained weather-removal networks to first remove occlusions before passing the cleaned images to a segmentation network, and employing a single unified model that simultaneously removes occlusions and segments the images. Their findings show that the unified model slightly outperforms the pretrained approach on average, though both methods exhibit unique strengths under certain weather conditions. Notably, while the deweathering networks excel at removing occlusions and achieve high structural similarity to the clean images, the subsequent segmentation results are sometimes degraded. This degradation is attributed to the complexity of the environment and the extent of occlusion. This observation underscores the challenges inherent in inferring complex geometries in heavily occluded regions. Although it is possible to label lightly occluded images or occluded images with simple geometry, the task of inferring complex geometries in heavily occluded regions is challenging. For example, when looking at Figure 2.2c, it is almost impossible to infer what is blocked by the large water droplet without leveraging additional information or assumptions (e.g., images from multiple cameras or temporal information).

In our work, we propose to achieve more accurate predictions in occluded regions by leveraging temporal information. We infer predictions for the current frame from previous frames where the occlusion was not present at the same location in the current frame To connect the current and previous frames when the ship is moving, optical flow is perhaps the most straightforward solution.

3.1.3 Optical Flow

Optical flow tracks the motion of objects in adjacent frames and encodes the information through dense motion vectors that represent the displacement of pixels. The field of op-

tical flow in computer vision precedes CNNs, and traditional techniques use custom pixel encodings and leverage linear transformation characteristics in order to track features in an image and derive the flow [73]–[75]. However, edge cases in the field are especially difficult to overcome, and tracking the motion of objects using hard-coded features becomes especially difficult when they move in and out of view, e.g. by leaving the frame or becoming partially or totally obscured by another object.

With the development of CNNs, pioneer work [76] began to emerge that leverage artificial datasets [76]–[78] with ground truth optical flow to teach optical flow to neural networks. It was soon shown that features learned from CNNs outperform traditional handcrafted features while also showing superior degrees of generalization and robustness to occlusion. FlowNet [76] was the first approach to predict optical flow using a pure CNN architecture. Since then, the best-performing optical flow networks take inspiration from optimization algorithms, where networks predict the flow that minimizes the difference between the features of one image and the warped features of the other image. FlowNet2 [79] achieved this using multiple sequential FlowNet's specialized for different object displacements, creating finer flows by sequentially warping image features. PWCNet [3] refines on this idea by incorporating warping in a single network architecture by calculating the flow at multiple levels in the feature pyramid network, and calculating the correlation cost of the warped feature using the flow at each level. More recently, RAFT [80] used a recurrent network to sequentially create more refined flows using the previous flow prediction and the associated correlation pyramid cost. Although many different architectures for optical flow exist, they all incorporate similar structures that sequentially minimize correlation costs using more refined flows.

Recent optical flow works have shown that optical flow networks can learn to be robust to occlusions using data augmentation techniques [80], [81]. To simulate occlusions in their data, they take inspiration from a data augmentation technique from the field of object detection, called random erasing. In this approach, it was shown that partially obscuring an image lead to less network hyperfixation on specific object features and improved performance. In their work, Zhong et al. [82] randomly erase rectangular portions of an image, investigating the effects of the color in the erased region, the aspect ratio of the rectangle, the ratio of the rectangle's area to the image's area, and the probability of erasing. Optical flow methodologies leverage this technique by asymmetrically erasing their images, where one image has a rectangular portion randomly replaced by the image mean. We aim to use optical flow networks and their methodology to achieve occlusion robustness and mitigate error from water droplets and lens occlusions in our data.

We aim to leverage the robustness of optical flow networks to occlusions by combining their methodologies with video segmentation techniques. By integrating optical flow's

occlusion handling with the temporal information inherent in video data, we can better address the challenges posed by raindrops and other lens occlusions in our dataset.

3.1.4 Video Semantic Segmentation

Video semantic segmentation extends the principles of semantic segmentation, which aims to classify and segment all pixels in an image. While semantic segmentation focuses on segmenting individual images, video segmentation tackles the same task but considers temporal information across sequential frames to improve classification. The video segmentation approach is particularly relevant in datasets like Cityscapes [83], where images are captured sequentially from the viewpoint of a moving vehicle within an urban environment. The Cityscapes dataset serves as a rich resource for training video segmentation models, containing finely annotated images in sequential format.

Despite the availability of annotated video data, models trained on such datasets often underperform compared to those trained solely on single images. This performance gap is largely due to the greater abundance of labeled single image data available for training. For instance, although the Cityscapes dataset includes labeled video data, the annotations are temporally spaced such that many sequential labels are quite distinct from each other. Consequently, video segmentation models that leverage temporal information offer little to no advantage over image segmentation models in this context. In fact, the best performing video segmentation approach for Cityscapes does not use a video segmentation model at all. Instead, Zhu et al. [84] augment the Cityscapes dataset by propagating handmade labels to sequential unlabeled frames.

Conversely, in datasets where video data is labeled with a higher temporal resolution, such as Camvid [85], the top-performing models effectively utilize the temporal information of the videos to enhance performance [86]. Similar to Camvid, our dataset predominantly consists of video data, and we anticipate that models leveraging this temporal context will outperform traditional single image models. Furthermore, unlike the realm of autonomous driving, the relative speed of ships in Arctic waters is much slower, ensuring sequential video frames to be similar and rich temporal context to be available to the model. Therefore, we aim to harness the temporal information inherent in our video dataset to train video segmentation models capable of surpassing their single image counterparts.

Video semantic segmentation techniques tend to revolve around a few common approaches to distill temporal information in their models. Without knowledge of specific object instances within the image, as would be the case with panoptic segmentation [87], researchers need to use techniques that find correlations between the pixels of adjacent

frames. These techniques include the use of recurrent networks [88], conditional random fields [89], [90], and optical flow [91]–[94]. We are particularly interested in video segmentation methodologies that leverage optical flow, due to the ability of optical flow networks to handle occlusions and seamlessly integrate temporal information across video frames.

Among the various approaches in video segmentation that employ optical flow, two methodologies stand out for their practical application to our needs. We prefer these methodologies for their out-of-the-box functionalities that do not require retraining on optical flow datasets. First, Liu et al. [93] achieve near state-of-the art performance on the CamVid dataset by training traditional image segmentation networks with an additional flow loss. This method incorporates FlowNet2 [79] as a teacher network during training, distilling temporal knowledge into image segmentation networks by penalizing segmentation predictions that do not match the optical flow prediction. Our aim is to utilize this approach as a baseline to assess the potential of optical flow in enhancing network performance on our dataset and to explore the full capabilities of single image networks for video data. Subsequently, we evaluate whether these results can be improved by embedding optical flow features directly into the convolutional network.

To accomplish this, we adopt the network described by Cheng et al. [2] known as SegFlow. Cheng et al. leverage the analogous scale structures of FlowNet and ResNet to directly embed optical flow features into the segmentation network. In their methodology, they concatenate the feature maps of the final three layers of ResNet and FlowNet into separate decoder branches in order to estimate optical flow and object segmentation simultaneously. However, their methodology has some minor flaws. First, the features concatenated from ResNet and FlowNet are at different scales, which prevents direct concatenation. The original authors addressed this by pooling and cropping the larger FlowNet features to match the size of the ResNet features. We argue that this approach may result in the loss of valuable information, especially in low-dimensional feature spaces. We also augment their network to more closely resemble optical flow networks and segmentation networks by changing the number of inputs to the ResNet encoder, and changing the decoder architecture to more closely resemble UPerNet. We show that these changes allow for superior performance of video segmentation networks over image segmentation networks, without the need for retraining on optical flow datasets.

3.2 Baseline Semantic Segmentation

While CNNs have demonstrated impressive performance in semantic segmentation tasks, particularly on datasets featuring close-range ice objects such as ours [32], [39], our ap-

proach to labeling causes our work to be unique to conventional segmentation tasks. Instead of generating simplistic annotations manually, our labeling methodology enables the creation of highly detailed annotations. This is depicted in Figure 2.6, where brash ice in the near-field and water in the far-field are represented by just a few pixels in width. These fine annotations tend to not be present in traditional semantic segmentation datasets [83], [95], [96], since they tend to be interested in annotating commonplace objects, such as people or cars, which are large and distinct. Consequently, we anticipate that the performance of CNNs that excelled on previous datasets may not translate effectively to ours. Thus, we begin by reevaluating these high-performing networks on our dataset, viewing their performance as a new baseline to surpass.

To assess the efficacy of neural networks on our dataset and create our baseline, we employ six different architectures, each utilizing a ResNet encoder [4]. This ResNet encoder is pre-trained on the ImageNet dataset [70]. To explore the trade-offs between depth and generalization, we evaluate three ResNet backbones of varying depth (50, 101, and 152), considering that while deeper networks often yield superior performance, they may also exhibit a tendency to overfit on smaller datasets [97].

Across each backbone, we assess the performance of six distinct network architectures: a basic fully convolutional network (FCN) [62], DeeplabV3 [66], DeeplabV3+ [67], PSPNet (Pyramid Scene Parsing Network) [68], UNet [64], and UPerNet [69]. Additionally, before the training data is passed to each network, the data are augmented through the following transformations [72]:

- Random color jittering [98], [99] of the saturation and contrast in the range of [0.2, 1.8], and [0.5, 1.5],
- Random rotation in the range of $[-5°, 5°]$,
- Random Gaussian blur with kernel size 5 and standard deviation in the range of $[0.5, 2]$,
- Random horizontal flipping with a probability of $p = 0.5$,
- Random crop to the range of $[512, 512]$,
- Mean and Gaussian random erasing [82] with an area ratio of (0.02, 0.33), and an aspect ratio of (0.3, 2.5).

These data augmentations were selected to increase the variance of our dataset while remaining true to the variations we expect in the Arctic environment. One commonly

used data augmentation technique, random rescaling of the image before cropping, was intentionally omitted. This decision was influenced by the importance of ice size within our dataset, where distinguishing between ice floes and brash ice relies heavily on their size. For this reason, we also require a large crop size, as ice floes in the far field may resemble the size of brash ice in the near field. Additionally, we apply random color jittering to simulate changes in lighting conditions. To simulate occlusions, we use the random erasing technique adapted from optical flow methods. This technique involves randomly replacing a rectangular portion of the image with its mean value. To mimic the effect of water droplets on the lens, we also choose to replace the occluded portion with a Gaussian blur. The blur's kernel size is determined by the minimum possible length of the erased portion, and the standard deviation is chosen within the range of 15 to 100. A standard deviation of 15 simulates light occlusions, where water droplets partially obscure a region, while a standard deviation of 100 represents heavy occlusions, where the region is significantly blurred and obscured. We show examples of our data augmentation in Figure 3.1. The hyperparameter values were chosen according to the best performance on the validation dataset, using a Bayesian hyperparameter search [100] with Optuna [101].

To train our models, we use the class-balanced cross-entropy loss with softmax activation function, given by,

$$L_{CE} = \frac{1}{N} \sum_{n=1}^{N} -w_{y_n} \log \frac{\exp x_{n,y_n}}{\sum_{c=1}^{C} \exp x_{n,c}}, \tag{3.1}$$

where N is the number of samples determined by the dimension and batch size, w is the class weight, x is the model output, y is the target, and C is the number of classes.

3.3 Video Semantic Segmentation

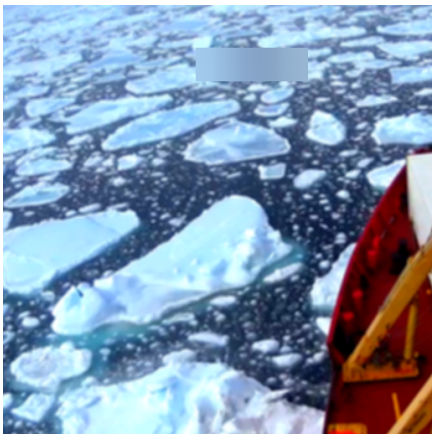
We employ methods from video segmentation to leverage temporal information in our dataset and improve performance. Namely, we are interested in video segmentation works that leverage optical flow, due to the ability of optical flow networks to handle occlusions and seamlessly integrate temporal information across video frames. We explore two methodologies: temporal knowledge distillation using optical flow teachers [93], and a joint optical flow and segmentation network [2].



(a)



(b)



(c)

Figure 3.1: Extreme examples of data augmentations from our data augmentation pipeline. (a) is the original unaltered image. (b) shows the randomly cropped image with augmentation factors of saturation 0.2, contrast 0.5, rotation -5, gaussian blur standard deviation 0.5, random erasing scale 0.33, random erasing ratio 2.5, random erasing gaussian blur standard deviation 15. (c) shows a different crop with saturation 1.8, contrast 1.5, rotation 20, gaussian blur standard deviation 2, random erasing scale 0.02, random erasing ratio 0.3, random erasing gaussian blur standard deviation 100

3.3.1 Temporal Knowledge Distillation

We accomplish temporal knowledge distillation through the temporal consistency loss [93]. The temporal consistency loss evaluates the performance of a single image network on two adjacent frames and penalizes predictions that do not align with the predicted optical flow. Figure 3.2a illustrates a simplified flowchart of this methodology. Specifically, for adjacent video frames (I_n, I_{n+1}), each frame is processed individually by a single image segmentation network, resulting in predictions (s_n, s_{n+1}). Concurrently, these two frames are fed into an optical flow network to obtain the backward optical flow prediction, $f_{n+1\to n}$. The initial prediction s_n is then reconstructed from s_{n+1} using a bilinear warping operation [79] to create the warped prediction $\hat{s}_{(n+1)\to n}$ $(\mathcal{W}(s_{n+1}, f_{n+1\to n}) = \hat{s}_{(n+1)\to n})$. This allows the creation of two predictions for a single frame: one predicted by the segmentation network and the other propagated by the optical flow. As such, the optical flow prediction can be used as the ground truth to inform a temporal loss and penalize the predictions that are temporally inconsistent. Additionally, to avoid penalizing regions where the optical flow prediction is inaccurate due to occlusions or objects exiting the frame, an occlusion mask $V = \exp\left(-|I_n - \hat{I}_{(n+1)\to n}|\right)$ is employed. The occlusion mask warps the second image to the first, and marks pixels with different RGB values as occlusions. The temporal consistency loss, denoted as L_{tc}, is calculated as follows:

$$L_{tc} = \frac{1}{N}\sum_{i=1}^{N} \|V \odot (s_n^i - \hat{s}_{n+1\to n}^i)\|_2^2, \tag{3.2}$$

where $\odot$ is the element-wise product and N is the total number of pixels in the image. Intuitively, this loss penalizes pixels that fail to align with the optical flow prediction. An illustration of this concept is provided in Figure 3.2b, where two sequential predictions are shown side-by-side with the optical flow prediction below. We highlight in the red box an erroneous prediction, where a classification for iceberg is shown in one frame but not the other. Since the optical flow predicts slow-moving pixels in this area, the sudden disappearance of a classification would be impossible and is consequently penalized.

To account for varying speeds of the ship and/or water, we wish to vary the time interval between the sequential frames that are used to train the model. This functions to vary the temporal context passed to the network and prevent the network from overfitting to a particular time interval. In our study, we randomly choose our second image to be within 4 seconds of the first, $I_{n+1} \in (I^n, I^{n+4}]$. We find the 4 second window to be sufficiently spaced to incorporate temporal variation while ensuring features are present in sequential images. Additionally, we use the better performing RAFT network [80] as the optical flow teacher, rather than FlowNet2.

The incorporation of the temporal loss in our framework involves the use of two sequential images for optical flow estimation, but only the ground truth label for the segmentation of the first image. We integrate the temporal loss with the cross-entropy loss, and assign it a weight of 0.2, as per [93]. The total loss is given by,

$$\mathcal{L} = L_{ce} + 0.2L_{tc}, \tag{3.3}$$

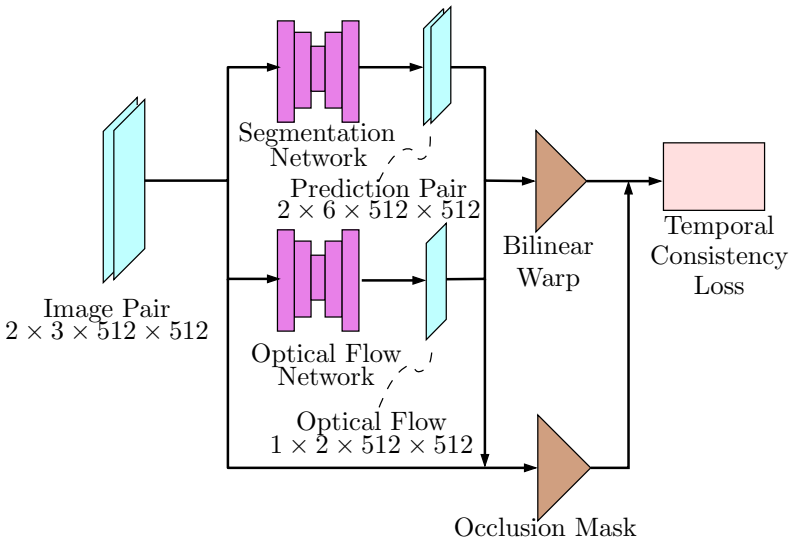
Where L_{ce} is the cross-entropy loss, as per Equation 3.1.

3.3.2 Joint Optical Flow and Segmentation Model

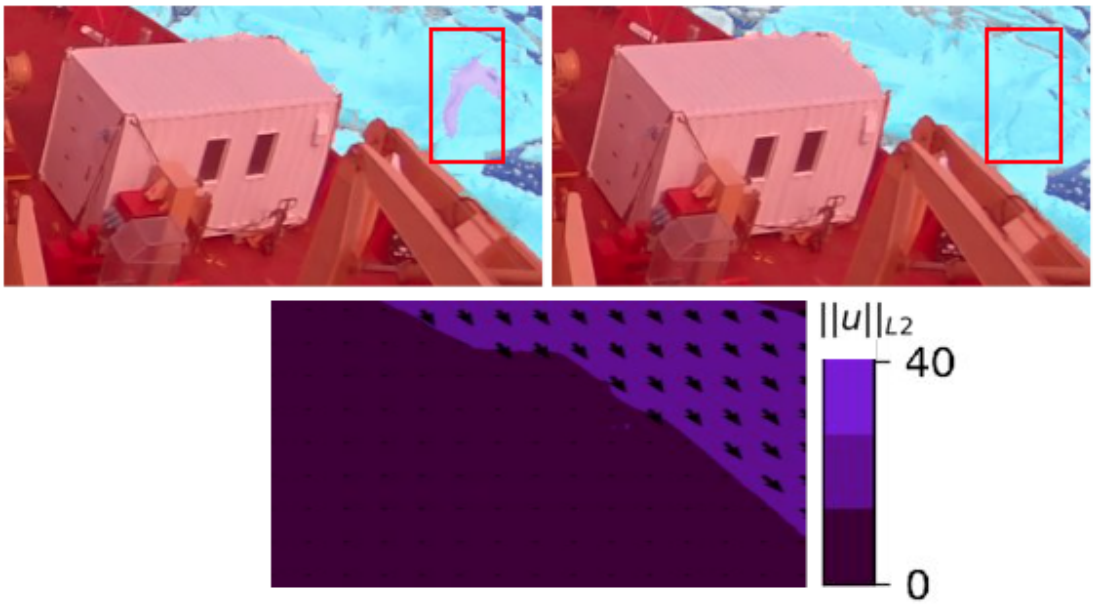
In our work, we suggest the use of a unified optical flow and segmentation network to leverage temporal information in the presence of occlusion. Our work is motivated by SegFlow [2] that combines ResNet and FlowNet into a single video object segmentation model. The network consists of a segmentation encoder branch, an optical flow encoder branch, a segmentation decoder, and an optical flow decoder, with skip connections between the branches and the decoders. The segmentation encoder uses a ResNet encoder with 101 layers, the optical flow branch uses FlowNet, and the decoders concatenate features from the final three layers of both networks to create an optical flow prediction and an object segmentation prediction. Our network is inspired by the structure of SegFlow but we make changes to each of its components.

Our work is interested in improving semantic segmentation in occluded areas using temporal information, and our changes to SegFlow reflect this. First, we change the network structure to create segmentation predictions for two images rather than one. This allows our network to leverage information from both frames in the case where occlusion is present in the first or the second frame. Correspondingly, we change the network to leverage temporal information from *bi-directional* optical flow (i.e. backward $f_{(n+1)\to n}$ and forward $f_{n\to(n+1)}$ flows). This allows the network to learn how features in one frame correspond to the other in both directions. We also change components based on developments in the semantic segmentation and optical flow fields since SegFlow. Our adapted model shares many features with UPerNet, and we call our model UPerFlow. Below, we detail the changes we make to the segmentation encoder branch, segmentation encoder, optical flow branch, and skip connections.

Segmentation Encoder Branch. For segmentation encoder, we adopt the pretrained ResNet encoder with 101 layers. In order to encode features from both images, we change the first layer of the network to accept six channels instead of three. These channels



(a)



(b)

Figure 3.2: An illustration of the temporal consistency loss [93]. (a) shows a simplified flow chart of the method. (b) shows an example of the method on our dataset. Highlighted in the red box is an inconsistent prediction, where a classification for iceberg is shown in one frame but not the other. This classification is penalized, since it does not match the optical flow prediction, shown in the bottom. The arrows of the optical flow show the direction of the flow, and the color shows the magnitude in pixels per frame.

correspond to the concatenation of the RGB channels of both images, consistent with the approach used in many optical flow architectures, such as FlowNet and RAFT.

Segmentation Decoder. We adopt the UPerNet decoder for the segmentation branch of our network. The UPerNet decoder concatenates hierarchical encoder features into a feature pyramid network to increase spatial context passed to the network and creates accurate segmentations. We anticipate the rich spatial context provided by the UPerNet will be well suited for our dataset and we adopt its structure for our segmentation decoder. However, we can swap our choice of decoder according to the best performing image segmentation network. We use two decoders that create segmentations for each image.

We do not create a separate decoder for optical flow since we are not interested in optical flow prediction.

Optical Flow Branch and Skip Connections. We use PWCNet as the flow branch instead of FlowNet, for its enhanced optical flow capability while being contained in a simple CNN structure. PWCNet provides rich contextual information at each level of its feature pyramid decoder, with a warping layer and cost volume layer corresponding to every low-level flow feature. The warp layer provides additional context to the flow prediction by warping the feature from the other image that corresponds to the predicted flow. Subsequently, the warped feature is compared to the original feature using a cost volume, CV, which compares their similarity by the relation,

$$CV_{i,j}(l_1, l_2) = \frac{1}{N}(c_i(l_1))^T c_j(l_2), \tag{3.4}$$

where N is the number of channels in the ith and jth column vectors of latent features l_1 and l_2, denoted as $c_i(l_1)$ and $c_j(l_2)$, respectively.

We integrate the flow, warp, and cost features from various decoder levels of PWCNet into the segmentation decoders. These features are concatenated with the corresponding ResNet feature at the same scale from the segmentation branch and then passed to the decoder. For the decoder processing the first image, we concatenate the flow features calculated from the backward flow, and for the other decoder, we use the forward flow features. This approach ensures that each decoder has context from the other image and an understanding of how features between the two images are related through flow information. Additionally, the warp feature and cost volume offer insights into the accuracy of the flow information to the model, where areas with a high cost volume may correspond to incorrect flow predictions.

Chapter 4

Results and Discussion

4.1 Semantic Segmentation

We evaluate the performance of our models using the mean Intersection over Union (mIoU) performance and mean pixel-wise accuracy (mAcc) on the test dataset. The models were trained for maximum of 800 epochs, with an AdamW learning rate optimizer [102] and a cosine annealing learning rate scheduler [103] with learning rate from 1E-4 to 1E-6. We employ early stopping if the validation loss does not improve for 150 epochs, and save the best performing model for evaluation. The best performing model and learning rate were found alongside the data augmentation hyperparameters (Section 3.2) using a broad Bayesian search [100] with Optuna [101]. The training curves from the hyperparameter tuning are included in Appendix A.1. All models were trained with a batch size of 16 using 1-2 NVIDIA A100 GPU's, depending on memory constraint.

For our dataset, the optimal network was identified as UPerNet with a ResNet101 backbone. On our test dataset, this network achieved a mean Intersection over Union (mIoU) of 0.838 and a mean pixel-wise accuracy (mAcc) of 0.946. The architecture of UPerNet proved to be well-suited for our dataset, and we hypothesize that the skip connections played a crucial role in propagating the rich spatial context of our complex scenery to the final layers of the decoder.

The confusion matrix for UPerNet is shown in Figure 4.1. The network performs accurate segmentations for almost all classes. Distinct classes such as the ship and iceberg are classified with perfect accuracy, and the network also performs well at classifying ice floes, water, and sky. However, performance degrades when classifying brash ice, leading

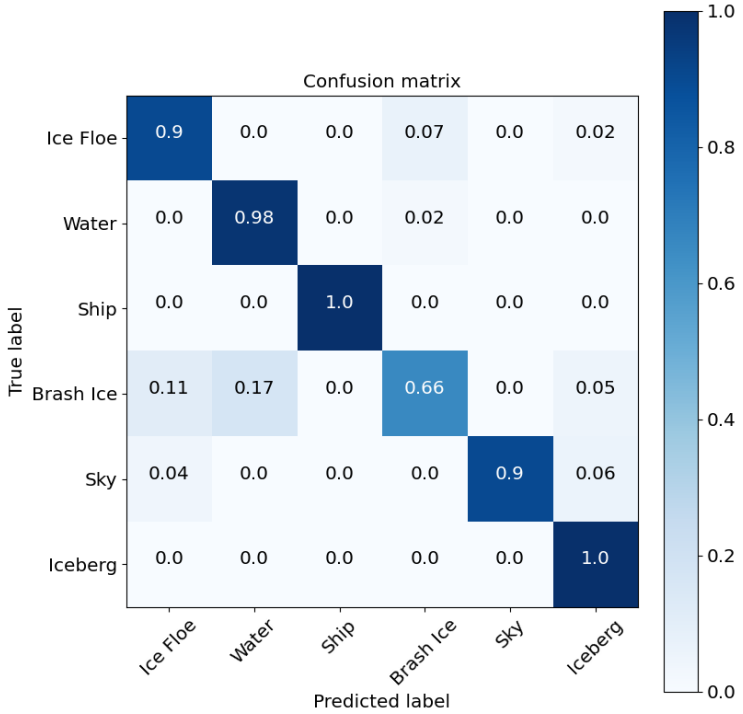


Figure 4.1: The confusion matrix for UPerNet. The network performs very accurate segmentations for almost all classes. However, performance sharply degrades when classifying brash ice, which causes average performance to drop considerably.

to a noticeable drop in average performance. The network frequently confuses the brash ice class with water or ice floe. This is an understandable result, as these classes are visually similar, and brash ice can be physically considered a transition class between ice floe and water.

Moreover, it is possible that some of our ground truth data are incorrect since classes in the far field can be challenging to identify accurately. Small fragments of ice must be classified as brash ice, but seemingly small ice fragments in the far field may become larger as they approach the ship, causing their classification to change. This inherent difficulty in distinguishing between brash ice, water, and ice floes, especially in the far-field, can contribute to the observed classification errors.

Some sample inference results are shown in Figure 4.2. Our test set follows Henry Larsen as she approaches slowly approaches an iceberg. In the near-field is a large, still area of water with some dispersed ice floes. The water is very still and creates a clear reflection

of the iceberg, shown in Figures 4.2d and 4.2f. The iceberg itself stretches from the mid to far-field, occupying more of the image as the vessel gets closer. As proximity increases, the upper portion of the iceberg becomes visible, revealing a distant field of brash ice and ice floes, which extends the ice-covered landscape seen in the far-field of the image. The model successfully identifies and segments the objects in our dataset, effectively narrating the ice scenery through its segmentation results. However, some notable misclassifications are observed.

One notable error is the region above the iceberg. Rather than predicting brash ice and ice floes, the network prefers to classify the whole region as an iceberg. This issue is illustrated in Figure 4.2d, where an attempt to classify brash ice is made, but the region is predominantly labeled as an iceberg. This is perhaps a result of our training data, where all iceberg instances are adjacent to the sky and other ice objects are never found above the iceberg. Thus, the network prefers to create iceberg classifications that touch the sky, rather than classify the identifiable ice floes present above the iceberg.

Additionally, the network predicts large clusters of brash ice as ice floes. However, this is a result of our own mislabelling, since our semi-supervised approach cannot differentiate between large clusters of brash ice and ice floes. This can be seen in Figure 4.2c where a large cluster of brash ice in the near-field is classified as a single ice floe, matching the erroneous label in Figure 4.2d.

Another notable misclassification involves the reflection of the iceberg in the still water. As shown in Figures 4.2b and 4.2f, the reflection is frequently mislabeled as either brash ice or iceberg. This error likely arises due to the visual similarity between the reflection and the actual iceberg, as both share similar color, texture, and shape. The model's difficulty in differentiating between the reflection and the actual object underscores a common challenge in image segmentation tasks, where reflections and shadows can be confused with the objects they mirror. In Section 4.2, we highlight how optical flow may help the network create accurate predictions in these regions. However, prior to this, we investigate UPerNet's capability to create predictions in occluded regions after training with random erasing.

4.1.1 Occlusion Mitigation

We evaluate UPerNet's performance in the context of occlusions to determine if random erasing can effectively enhance the network's robustness to such challenges. As our test dataset lacks natural occlusions, we simulate occluded regions by artificially introducing occlusions into the test images. Since the test dataset has ground truth, this approach

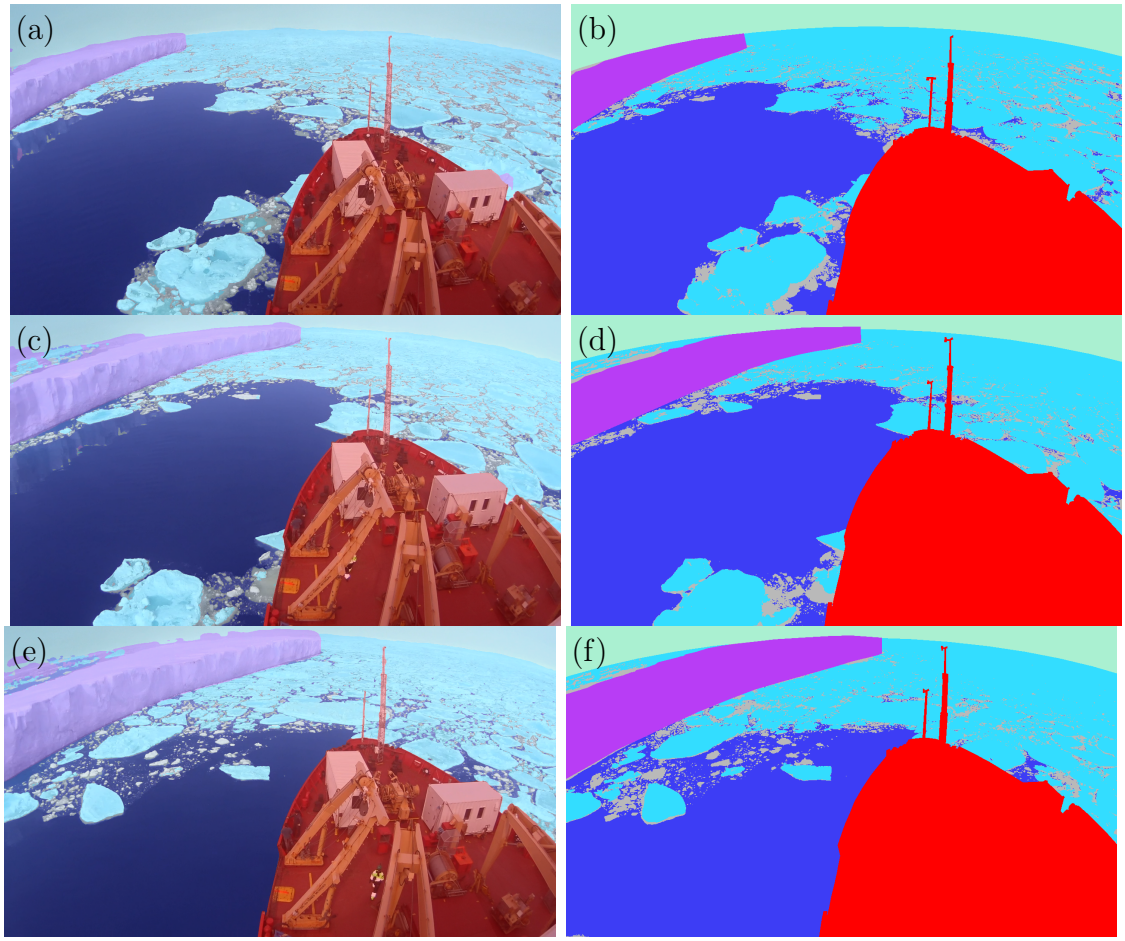


Figure 4.2: Examples of predictions from UPerNet for our dataset. The original images with the network predictions overlaid are on the left, and the labels are on the right. Figures (a)-(f) are examples from the test dataset. The results are accurate overall, but we see some common mistakes, such as misclassifying the iceberg reflection.

provides an objective measure of the model's ability to infer information in these artificially occluded areas. Additionally, we incorporate samples of unlabeled data to showcase the network's performance under real-world occlusion scenarios. By assessing these aspects, we aim to understand how well the network generalizes to occlusions and the fundamental capabilities and limitations of single image networks.

To evaluate our model's performance under occlusion, we generate two artificial datasets: one simulating light occlusions where the underlying data remains partially visible, and another simulating heavy occlusions where the data is fully obscured. To create these datasets, we occlude every other image in the test dataset using the random erasing technique described in Section 3.2. Random square areas of the image are occluded with an area ratio of 0.15. For light occlusions, we apply a Gaussian blur with a standard deviation of 15 in the occluded region, while for heavy occlusions, we use a standard deviation of 100. This testing approach mirrors the occlusions introduced during training, allowing the network to demonstrate its familiarity with such occlusions and the extent of its predictive capabilities in occluded regions.

The performance of the network on the two datasets is shown in Table 4.1. Results for the lightly occluded dataset are denoted as (Light), and for the heavily occluded dataset as (Heavy). Given the network's familiarity with this type of occlusion, it experiences only a modest decline in performance. On the lightly occluded dataset, performance decreases to 0.829 mIoU and 0.941 mAcc, reflecting an approximate 1% drop in mIoU. On the heavily occluded dataset, performance further declines to 0.819 mIoU and 0.935 mAcc, corresponding to an approximate 2% decrease. Although the decrease in performance is modest, it demonstrates that the network is less capable of inferring information in heavily occluded regions.

Network Architecture	**MIoU**	**MAcc**
UPerNet (None)	0.838	0.946
UPerNet (Light)	0.829	0.941
UPerNet (Heavy)	0.819	0.935

Table 4.1: Performance metrics of UPerNet on the artificially occluded datasets. (Light) represents its performance on the lightly occluded dataset, and (Heavy) represents the heavily occluded dataset.

Inference results of our network on the occluded data are illustrated in Figure 4.3, with the lightly occluded data on the left and the heavily occluded data on the right. Overall, the network performs well on lightly occluded data, effectively utilizing partially

visible features to generate reasonable predictions in occluded regions. For instance, in Figures 4.3a and 4.3e, the network is able to create reasonable predictions for the shape of the ship and ice floes. However, it struggles with iceberg predictions, producing spotty and inconsistent results, as observed in Figure 4.3c. This is likely due to the infrequent appearance of iceberg in the training data, especially from the new perspective presented in the test data. These findings indicate that even when partial information is available through occlusions, the predictive capability of single-image networks is constrained by their learned ability to interpolate shapes. Since the iceberg from the test dataset perspective was not present in the training data, the network cannot reliably infer the shape of the iceberg behind the occlusion.

The trend of spotty and inconsistent results is exacerbated on the heavily occluded data. Without any discernible features in the occluded regions, the network is forced to guess the shape and location of objects, leading to significant inaccuracies. In these cases, the network's predictions often become extremely coarse and unreliable. For instance, in Figures 4.3b and 4.3f, the predicted shape of the ship in the occluded region is severely distorted from our expectations. The network also produces large, coarse ice floe predictions, as it lacks specific knowledge of individual ice floe shapes. Additionally, in Figures 4.3b and 4.3d, the network tends to predict icebergs in occluded regions, even when not near an iceberg, demonstrating the unreliability of the network's results in occluded regions. This highlights the limitations of the network when dealing with severe occlusions, emphasizing the need for more advanced techniques to handle such challenging scenarios.

Finally, Figures 4.3g-4.3h illustrate the network's performance when faced with real occlusion, such as water droplets on the lens. In Figure 4.3g, two small droplets — one in the left far-field and another in the near field — are present. In Figure 4.3h a large water droplet obscures the left side of the image, while a smaller raindrop occludes part of the left side of the ship. Despite being trained with random erasing, the network struggles with these real-world occlusions, producing nearly random predictions in the affected areas. Similar to the iceberg scenario, the data behind the occluded regions is unfamiliar to the model, leaving the network unable to accurately interpolate the shape of the obscured objects.

The performance of image segmentation networks in the presence of occlusions and reflections leads us to explore video segmentation, where temporal information from sequential frames can be leveraged to improve classification in the presence of occlusion. Section 4.2 will delve into how video segmentation techniques can enhance the robustness and accuracy of our model in dynamic environments.

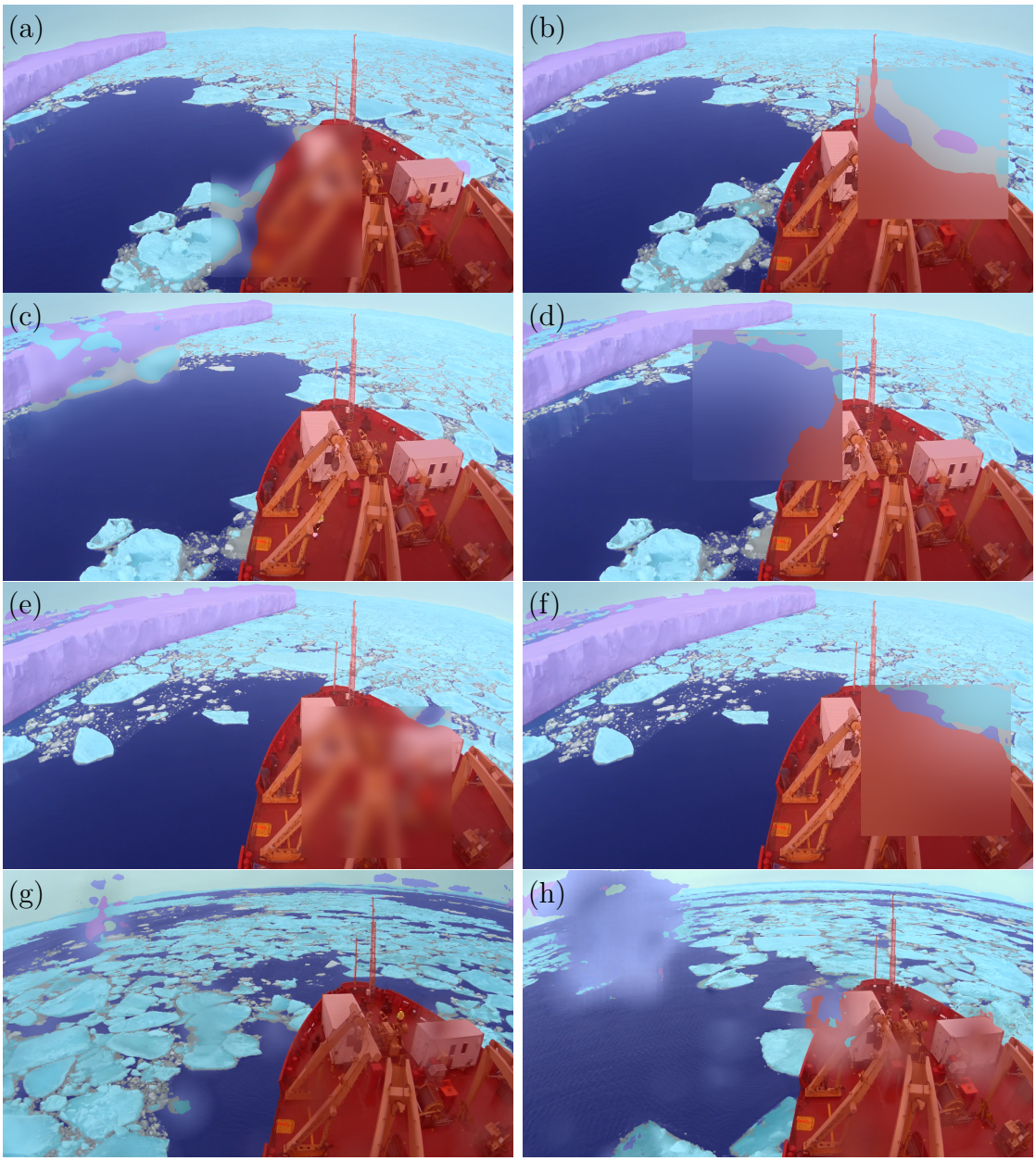


Figure 4.3: Examples of predictions from UPerNet in the presence of occlusion. The predictions for light occlusions are on the left, and the predictions for heavy occlusions are on the right. (g)-(h) show the performance of the network on unlabeled data with water droplets on the lens.

4.2 Video Semantic Segmentation

To objectively compare video segmentation models to image segmentation models, we train all models with the same hyperparameters and learning rates, mentioned in Sections 3.1.1 and 4.1. The only modification made is for the UPerFlow training, where random erasing is applied asymmetrically to the image pair (i.e., only one image is subjected to random erasing, not both). This setup allows us to fairly compare the impact of temporal consistency loss and the joint optical flow model to the baseline approach, which we will explore in the following subsections.

4.2.1 Temporal Knowledge Distillation

When retraining the model with temporal consistency loss 3.2, the network achieved a mean Intersection over Union (mIoU) of 0.818 and a mean pixel-wise accuracy (mAcc) of 0.945. This marks a sharp decrease in the performance of the network, with a 2% decrease in mIoU from baseline. The results suggest that the temporal consistency loss may not be a suitable loss function, potentially due to inaccuracies in optical flow prediction or issues in the warping process.

To better understand the implications of these findings, it's crucial to assess the reliability of the optical flow predictions themselves, especially since real-world datasets do not come with ground truth optical flow. In our case, we obtain a measure of flow prediction validity through the warping function. By warping one image to another using the predicted optical flow, we can gauge the accuracy of the flow prediction by comparing the warped image to the original image. Additionally, the occlusion mask from 3.3.1 provides a measure of the similarity between the images, which can further validate the optical flow prediction.

Figure 4.4 shows an example image from our dataset and its associated flow features from a two second window. The optical flow network used is RAFT. Figure 4.4a shows the original image and Figure 4.4b shows the predicted optical flow. When using the predicted flow to create the warped feature in Figure 4.4c, the visual similarity between the warped feature and the original image suggests a high level of accuracy in the flow prediction. The ice floes especially remain consistently positioned across the images, reflecting the model's effective capture of motion. This is further corroborated by the occlusion mask in Figure 4.4d, where most pixels are near 1 (represented by white), indicating strong agreement between the warped and original images. However, there are areas where the images diverge. This is especially notable next to the ship, where optical flow prediction in

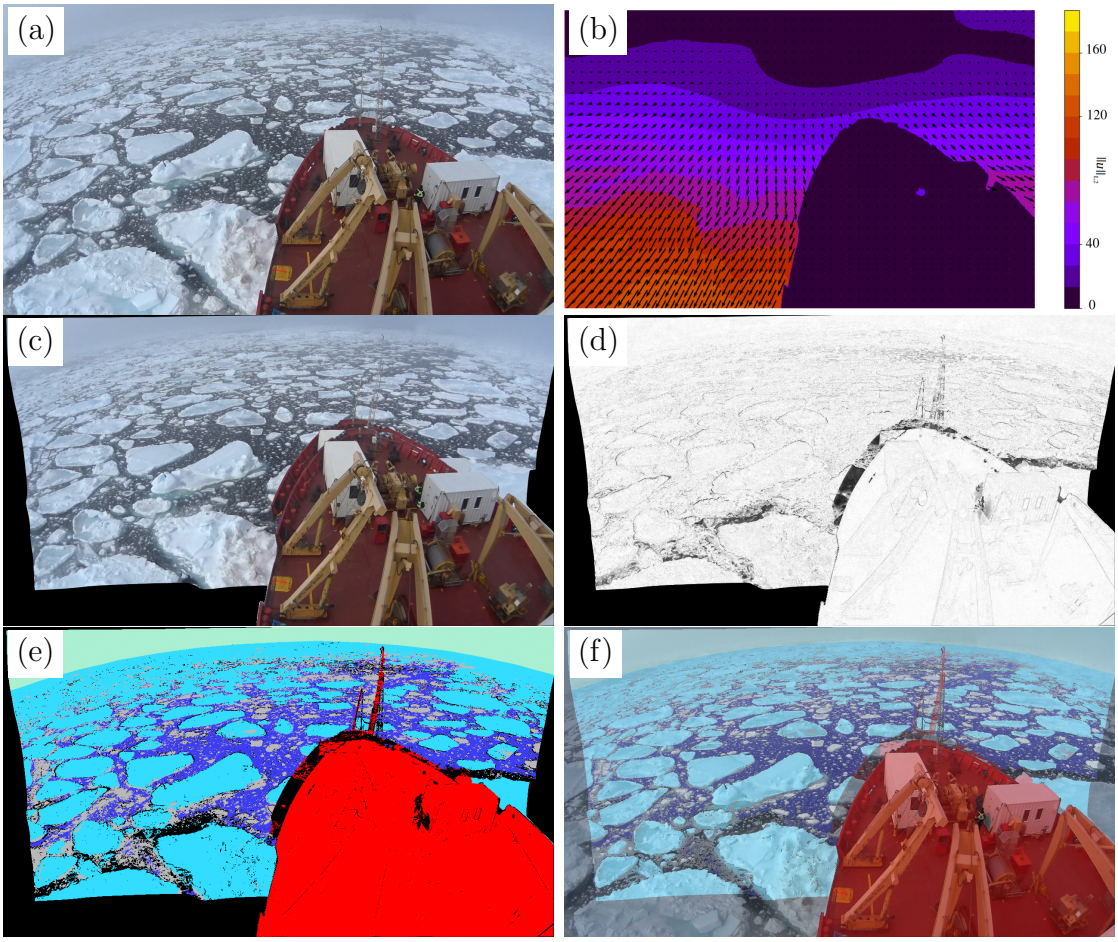


Figure 4.4: Flow features for a given image in our training dataset. The flow predictions are for a 2 second time difference. (a) shows the first image for the flow prediction. (b) shows the flow prediction. (c) shows the warped features given the predicted flow. (d) shows the occlusion mask calculated from the warped image. (e) shows the warped label for the following image, with dissimilar regions from the occlusion mask blacked out. (f) shows the masked warped label overlaid on the original image (a).

an occluded [1] area leads to duplication of ship features when warping. The darker regions in the occlusion mask highlight these discrepancies, showing occluded regions, such as near the ship's boundary, as well as areas where the flow prediction fails, such as at the edges of ice floes.

However, regions with accurate flow in the image space do not always align with the correct labels in the label space. The temporal consistency loss penalizes predictions that deviate from the warped prediction, relying on visual similarity in the image space to identify incorrect warped labels. Visual similarity between images may not be the best metric to determine occluded areas in labels.

To assess the validity of optical flow predictions within the label space, we conducted a similar test by warping the labels and overlaying them onto the original image. We then used the occlusion mask to black out invalid warped regions in the label. By visually checking whether the labels accurately correspond to the respective features in the image, we can gain a subjective measure of the optical flow's validity and the effectiveness of the occlusion mask within the label space.

Figure 4.4e illustrates the warped label with visually dissimilar regions from the occlusion mask blacked out, and Figure 4.4f displays the warped label overlaid on the original image. Many of the warped labels correctly correspond to the relevant features, which can be observed particularly with the warped ice floe labels. However, the occlusion mask fails to accurately identify several regions where the warped label is invalid. For example, the ship's white sections, which resemble ice floes, are not properly classified as occlusions by the occlusion mask. Additionally, in the image space, the ship's antenna appears see-through, but the label does not reflect this, preventing the antenna from being correctly identified as an invalid region. As a result, these mislabeled regions are not masked, teaching the network to produce incorrect predictions in areas frequently subject to occlusion.

Although training with temporal consistency loss 3.2 performs worse than baseline approach on average, it does offer certain advantages. Figure 4.5 shows the confusion matrix for UPerNet trained with temporal consistency loss. Despite the overall decline in performance, the network produces more accurate predictions for ice floe, water, and sky. This enhancement is particularly expected for ice floes, where optical flow has shown strong capabilities in tracking these features over time. This suggests that the temporal consistency loss effectively leverages the optical flow network's knowledge of moving objects, aiding the network in maintaining consistent predictions for features that exhibit distinct movement

[1] Although previously we use the term occlusion to only refer to water droplets on the lens, in actuality occlusion refers to any object in an image that obscures information. In the case of the ship, ice floes that go next to the ship and out of view are in an occluded area.

from their surroundings.

However, the model struggles with identifying brash ice, leading to a noticeable decrease in mIoU. Unlike other classes, brash ice lacks distinct boundaries and is dispersed among ice floes, making it a more fluid category. The optical flow network may have difficulty accurately tracking brash ice due to its subtle and variable movement. This challenge is further compounded by the limitations of the occlusion mask, which relies on visual similarity to identify invalid flow regions. The distinction between ice floes, brash ice, and water is often subtle and varies significantly with distance from the camera. The occlusion mask's inability to capture these nuances may lead to confusion, causing brash ice to be misclassified as water or ice floes due to similar coloration.

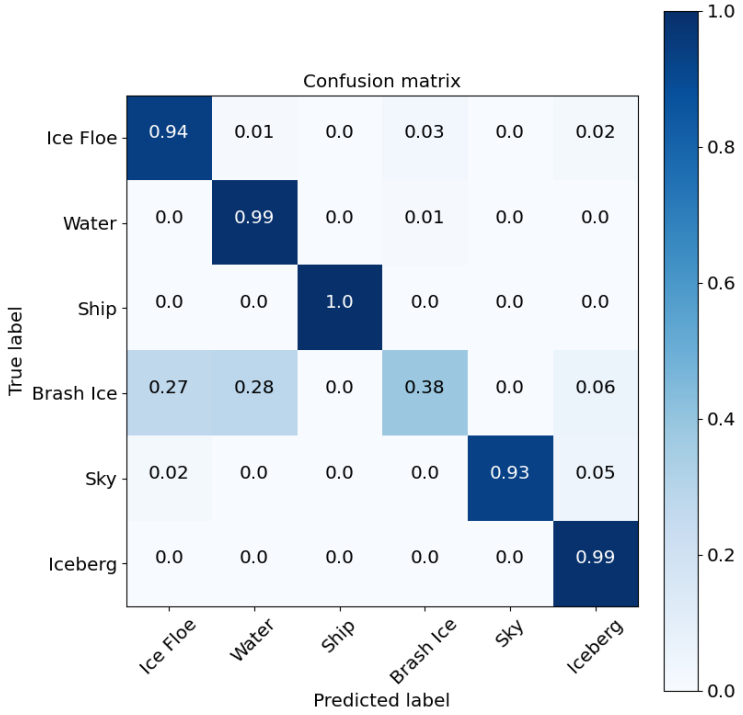


Figure 4.5: The confusion matrix for UPerNet trained with temporal consistency loss. Compared to Figure 4.1, the network performs better for most classes, but severe degradation in brash ice prediction results in a worse performance overall.

To address the limitations identified with the temporal consistency loss, particularly in detecting nuanced classes like brash ice, we explore the potential of a joint optical flow and segmentation model. A joint model could effectively leverage temporal information

while simultaneously learning the subtleties of invalid flow regions for a stronger overall performance. This leads us to the next section, where we explore the capabilities of our joint model.

4.2.2 Joint Optical Flow and Segmentation Model, UPerFlow

We summarize the results of our experiments in Table 4.2. The training and validation curves for UPerFlow are shown in Appendix A.2. UPerNet trained with temporal consistency loss is denoted as (F), and our approach, UPerFlow, is shown in the final row.

Network Architecture	Mean Intersection over Union	Mean Pixel-wise Accuracy
UPerNet	0.838	0.946
UPerNet (F)	0.818	0.945
UPerFlow (ours)	**0.844**	**0.948**

Table 4.2: Performance metrics for the different UPerNet configurations. (F) signifies training augmented with temporal consistency loss.

In contrast to the temporal consistency loss results, when flow features are embedded directly into the network, we observe no loss in performance. In fact, our model sees an increase in mIoU to 0.844 and mAcc to 0.948, which is the best performance out of all models. This shows that flow features can be beneficial for a network's performance, so long as they are provided to the network in the right format. Embedding the accurate flow features from PWCNet directly into the network supplies temporal context without enforcing incorrect predictions in specific regions that are difficult to measure with an occlusion mask.

The confusion matrix for UPerFlow is shown in Figure 4.6. We see an improvement from baseline in the ice floe and sky classes, with almost no degradation in brash ice accuracy. This indicates that the joint model successfully leverages optical flow while also understanding the nuances in regions of inaccurate flow. This balanced performance underscores the potential of joint models to improve segmentation accuracy across a variety of classes, even when dealing with complex and subtle variations in the data.

Figure 4.7 shows the performance of UPerFlow on the test dataset, compared to baseline in Figure 4.2. Overall, the performance remains quite similar and accurate, maintaining a strong ability to segment the various classes within the dataset. Notably, there are significant improvements in areas where the baseline model previously struggled. The UPerFlow model demonstrates a marked reduction in misclassifications caused by the reflection of

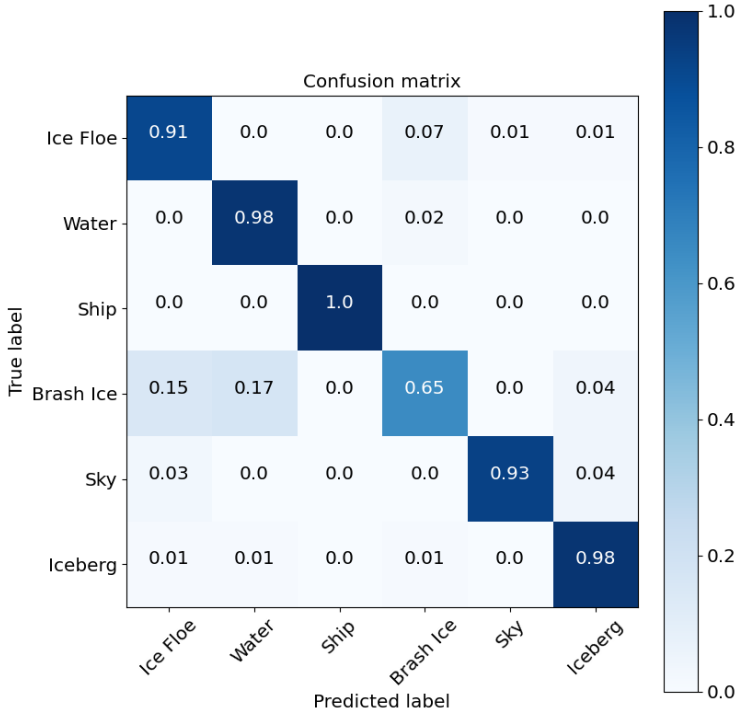


Figure 4.6: The confusion matrix for joint model UPerFlow. The network sees increased performance for the same classes as with the temporal consistency loss, without degradation of brash ice prediction, shown in Figure 4.5. The overall performance is better than baseline in Figure 4.1.

the iceberg in the water. Additionally, it better classifies the region above the iceberg, correctly identifying the ice in that area, though there is still room for improvement.

This suggests that the optical flow network's understanding of coherent objects plays a critical role in aiding the segmentation process. By recognizing the iceberg as a distinct object, the optical flow network likely helps the segmentation network generate more accurate predictions in challenging areas, such as above the iceberg and within its reflection. This capability enhances the model's overall performance in distinguishing and correctly labeling these complex regions.

However, the model still exhibits some errors. Due to inaccuracies in optical flow, the network tends to generate erroneous predictions in occluded areas and near image boundaries where objects exit the frame. For instance, in Figure 4.7a, the network misclassifies the ice floe adjacent to the ship. Similarly, Figure 4.7b illustrates how the network misclas-

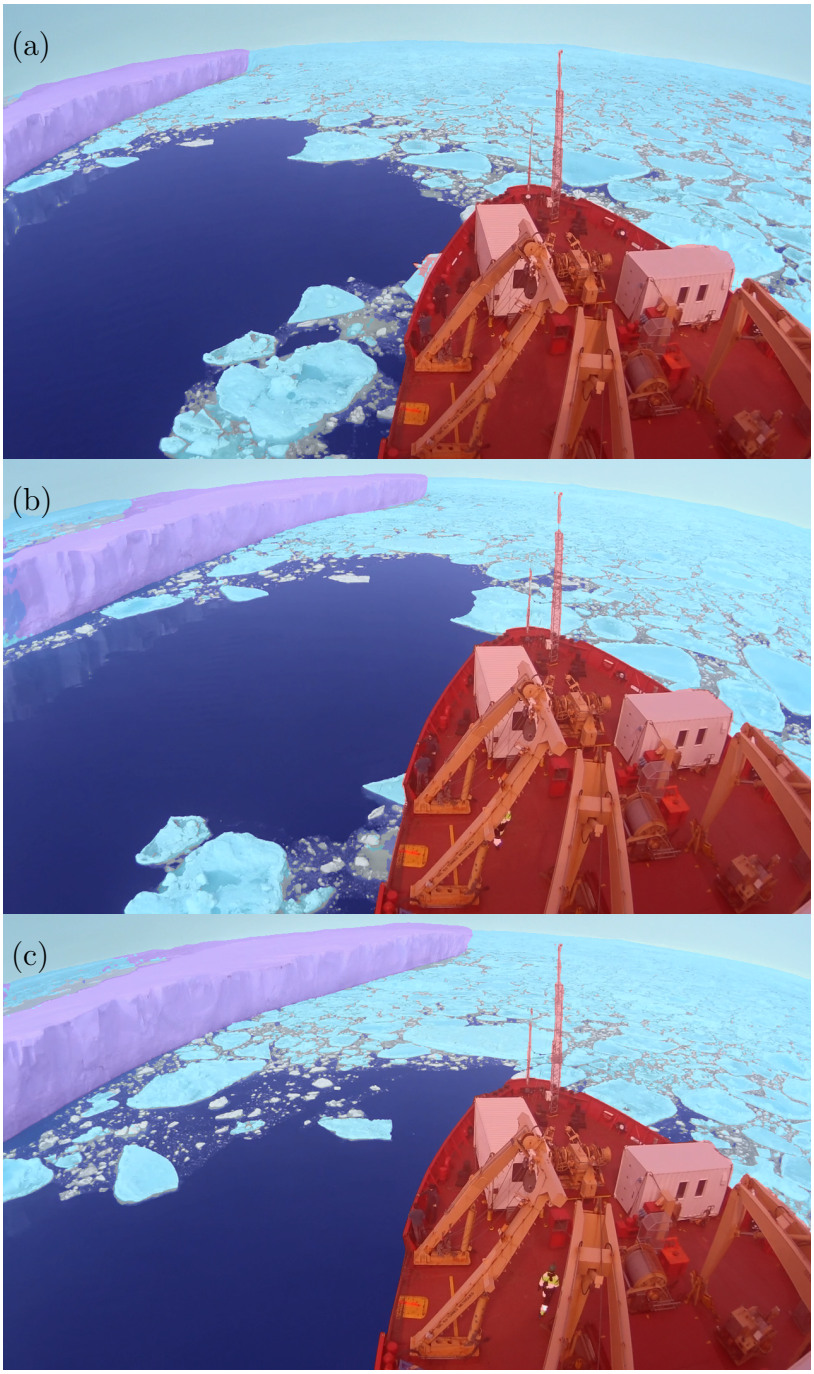


Figure 4.7: Examples of predictions from joint model UPerFlow for our dataset. We see similarly accurate predictions as baseline approach in Figure 4.2, but suffer significantly less from common errors, such as the reflection of the iceberg.

sifies the iceberg at the edge of the image. Despite these errors, they are relatively minor compared to the network's improved ability to handle sources of error, such as reflections or occlusions. The enhanced performance on occluded data is discussed in the next section.

4.2.3 Occlusion Mitigation

To test the performance of video segmentation models in the presence of occlusion, we use the same test performed in Section 4.1.1. In this section, we discuss only the performance of UPerFlow as it relates to the baseline UPerNet, since it has the best performance overall.

The performance of UPerFlow on the artificial datasets is shown in Table 4.1. Baseline results are included for reference. Compared to the image segmentation approach, video segmentation experiences performs significantly better in the presence of occlusion, particularly heavy occlusion. On the lightly occluded dataset, performance decreases to 0.837 mIoU and 0.945 mAcc, reflecting less than 1% drop in mIoU, but only slightly worse than UPerNet's performance on the non-occluded dataset. On the heavily occluded dataset, performance declines marginally more to 0.835 mIoU and 0.943 mAcc, which still corresponds to an approximate 1% decrease in mIoU. By effectively leveraging temporal information, UPerFlow manages to minimize the errors caused by occlusions, even in scenarios with substantial occlusion.

Network Architecture	**Occlusion Level**	**MIoU**	**MAcc**
UPerNet	None	0.838	0.946
	Light	0.829	0.941
	Heavy	0.819	0.935
UPerFlow	None	**0.844**	**0.948**
	Light	**0.837**	**0.945**
	Heavy	**0.835**	**0.943**

Table 4.3: Performance metrics of UPerFlow on the artificially occluded datasets. The performance of UPerNet baseline were included for reference. UPerFlow outperforms baseline approach for all occlusion levels, and we see less performance degradation as a result of occlusion.

Inference results of our network on the occluded data are illustrated in Figure 4.8, in comparison to baseline performance in Figure 4.3. In both light and heavy occlusion scenarios, UPerFlow exhibits superior performance compared to UPerNet in handling and inpainting the occluded regions. In the lightly occluded images, Figures 4.8a , 4.8c, and 4.8e,

the network demonstrates enhanced capability to refine annotations within occluded areas. It retains a clearer representation of the shapes of ships and ice objects, showing a notable improvement in accuracy. Notably, in Figure 4.8c, the network's predictions for the iceberg are no longer spotty, highlighting its ability to leverage temporal information to accurately inpaint regions obscured by occlusion.

Similarly, the performance on the heavily occluded data does not exhibit a noticeable decline. In Figures 4.3b, 4.8d, and 4.8f, the network maintains a coherent representation of the ship's shape and provides detailed annotations, even though no ship features are visible in the heavily occluded region. The network also generates plausible predictions for the boundaries of ice objects, demonstrating an ability to distinguish and distribute brash ice between adjacent ice floes, rather than producing large, inaccurate blobs of ice floe predictions. Finally, the network avoids random iceberg predictions, ensuring that all predictions are reasonable approximations of the occluded areas. These results underscore the advantage of video segmentation networks over image segmentation models, as they can utilize temporal information to more effectively infer details in occluded regions.

Finally, Figures 4.8g-4.8h illustrate the network's performance when faced with real-world occlusion. In contrast to UPerNet, UPerFlow produces more accurate predictions under real-world conditions. The video segmentation model creates reasonable predictions in occluded regions caused by water droplets, demonstrating its superior ability to handle real occlusions compared to traditional image segmentation approaches. This shows that random erasing effectively teaches UPerFlow to leverage temporal information and create reasonable predictions in occluded areas, rather than teaching the network to create best-guess interpolations, as it does for image segmentation.

Overall, our work has shown the advantages of video segmentation approaches over traditional image segmentation. Using optical flow, we can teach networks to leverage temporal information that is robust to environmental artifacts, such as reflections or occlusions, and boast improved performance overall. By embedding optical flow features directly into our model, the network learns temporal relations between coherent objects, and avoids misclassifications in regions where optical flow may fail, such as with the brash ice class.

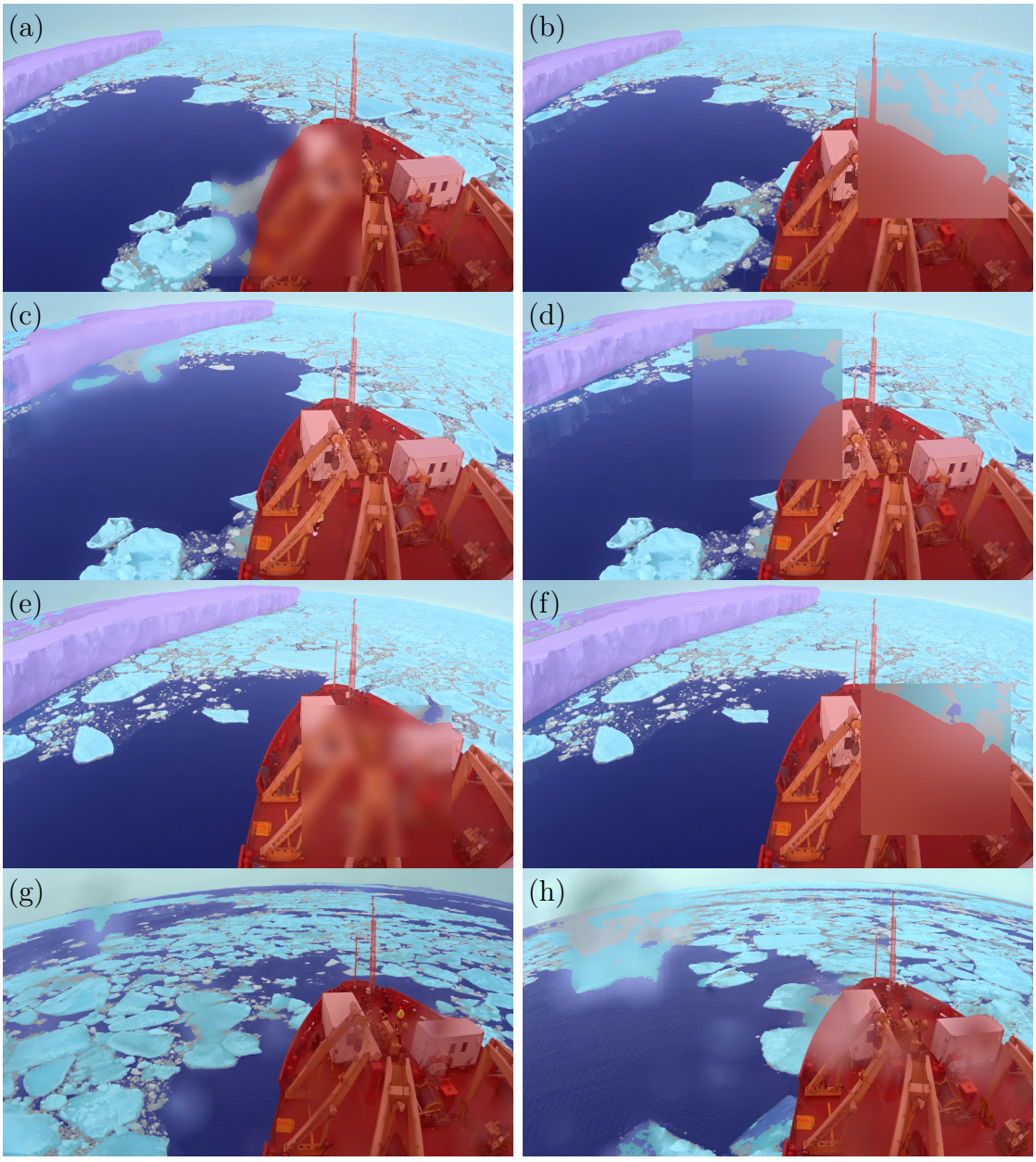


Figure 4.8: Examples of predictions from UPerFlow in the presence of occlusion. The predictions for light occlusions are on the left, and the predictions for heavy occlusions are on the right. Compared to baseline approach in Figure 4.3, we see less error as our approach is able to infer information in occluded regions using temporal context. (g)-(h) show UPerFlow on unlabeled data occluded with water droplets. UPerFlow is able to leverage temporal data to create predictions, even when presented with real-world occlusion.

Chapter 5

Conclusion

In this work, we present our contributions to the field of ice condition assessment and navigation safety in the Arctic. One of our key contributions is the development of a medium-sized, finely annotated dataset that classifies various ice and non-ice elements, namely ice floes, brash ice, water, ships, sky, and icebergs. This dataset, along with our detailed labeling methodology, provides a foundation for future works to create larger and similarly annotated datasets, facilitating further research and development in this area.

We also assessed existing image semantic segmentation networks and determined that UPerNet outperforms other architectures on our finely annotated dataset, achieving 94.6% accuracy and an mIoU of 0.838. This architecture benefits from skip connections to the decoder, which enhance feature propagation and improve overall performance compared to other models. Despite its strong performance, UPerNet still encounters challenges with artifacts common in realistic sea ice data, such as reflections in the water and droplets on the lens. These artifacts can lead to misclassifications and highlight the need for further refinement to handle such realistic conditions effectively.

We improve upon this baseline result using techniques from video segmentation. We show the limitations of training image segmentation networks with temporal consistency loss, and that embedding optical flow features directly into the network can improve performance. By modifying existing video semantic segmentation architecture SegFlow using more modern optical flow and segmentation architectures, our approach achieved 94.8% accuracy and mIoU of 0.844. Additionally, our approach suffers significantly less from misclassifications that appear from realistic data, such as reflections in the water and water droplets on the lens. On artificially occluded data, our method exhibits only a 1% performance decrease in heavily occluded regions, compared to a 2% degradation in the

baseline, and maintains accurate performance that is comparable to baseline approach on *non-occluded* data.

These contributions collectively highlight the potential of neural networks to provide robust ice condition assessments and support researchers and navigators in making objective, data-driven decisions. By enhancing the safety and efficiency of navigation in the increasingly accessible Arctic waters, this research supports the broader goal of improving decision-making capabilities in this critical and evolving region.

5.1 Future Works

While our research significantly advances the field of ice condition assessment and navigation safety in the Arctic, there are several avenues for future work that could build upon our findings:

Enhanced Data Annotation. Although our labeling methodology enabled fine annotations for a medium-sized dataset, color thresholding has some limitations. In Section 2.3.2, we list these limitations, as they pertain to differentiation between ice floes, brash ice, and water in specific contexts. Additionally, although our dataset is large, it only annotates frames from three videos, which limits the diversity of environmental conditions presented to the model. Expanding the dataset with additional annotations that encompass a broader range of ice conditions, lighting scenarios, and weather variations could provide a more comprehensive training ground for segmentation models. We hope that that we can use our trained classifiers for the creation of new low-cost data, while correcting minor misclassifications by hand in an ever-improving feedback loop. Another potential limitation of our dataset is with the ice floe labels. While our approach accurately annotates ice floe pixels, it does not differentiate between individual floes. This could significantly impede the implementation of trained models for Arctic navigation, since it may not be able to identify specific ice floes that are dangerous because of their strength or age. Incorporating a panoptic approach [87] that identifies and classifies specific ice floes could improve the model's ability to recognize and assess dangerous or noteworthy ice floes.

Refinement of Optical Flow Techniques. Our approach utilizes PWCNet as the optical flow network due to its convolutional structure, which allows for straightforward integration with the segmentation network. However, modern optical flow methods like RAFT leverage recurrent layers to iteratively refine optical flow predictions. These methods propagate image features and compare similarities across frames to enhance accuracy. It is plausible that adopting such optical flow architectures for segmentation could fur-

ther improve the results, potentially leading to even better performance in ice condition assessment and occlusion mitigation.

Improved Occlusion Handling. While our network performs effectively in reducing occlusion errors, predictions in occluded regions remain approximations rather than precise representations of the underlying data. Future improvements could be achieved by integrating techniques from video inpainting [104], where networks fill in missing regions by leveraging temporal information from adjacent frames. For instance, ProPainter [105] employs optical flow predictions combined with transformer blocks over a 10-frame window, achieving state-of-the-art results in this domain. Similarly, in the domain of semi-supervised video object segmentation, Segment Anything 2 [106] utilizes cross-attention mechanisms over seven frames and boasts impressive capability for zero-shot annotation and object tracking. Incorporating transformer blocks across the temporal domain and expanding the receptive temporal field could enhance our network's capability to leverage temporal information and handle occlusions more accurately, leading to better performance in complex, real-world scenarios.

Collaborative Research and Data Sharing. Collaborating with other research institutions and sharing datasets could accelerate advancements in this field. Building a collaborative framework for data sharing and joint research could help develop more generalized and robust models applicable to various contexts.

By addressing these areas, future research could significantly enhance the accuracy, reliability, and applicability of ice condition assessments, contributing to safer and more efficient navigation in the Arctic.

Bibliography

[1] J. C. Stroeve *et al.*, "Trends in arctic sea ice extent from cmip5, cmip3 and observations," *Geophysical Research Letters*, vol. 39, no. 16, 2012. DOI: https://doi.org/10.1029/2012GL052676. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2012GL052676. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2012GL052676.

[2] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, "Segflow: Joint learning for video object segmentation and optical flow," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[3] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, *Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume*, 2018. arXiv: 1709.02371 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1709.02371.

[4] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[5] W. Ostreng *et al.*, *Shipping in Arctic waters: a comparison of the Northeast, Northwest and trans polar passages*. Springer Science & Business Media, 2013.

[6] M. Garvin, "Review of technologies for real-time shipboard ice severity sensing," 2020.

[7] B. Lund *et al.*, "Arctic sea ice drift measured by shipboard marine radar," *Journal of Geophysical Research: Oceans*, vol. 123, no. 6, pp. 4298–4321, 2018. DOI: https://doi.org/10.1029/2018JC013769. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018JC013769. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JC013769.

[8] A. Parsa, "Ice age classification using cross-polarization measurement with x-band radar," in *ISOPE International Ocean and Polar Engineering Conference*, ISOPE, 2013, ISOPE–I.

[9] H. Murthy and S. Haykin, "Bayesian classification of surface-based ice-radar images," *IEEE Journal of Oceanic Engineering*, vol. 12, no. 3, pp. 493–502, 1987. DOI: 10.1109/JOE.1987.1145277.

[10] Oct. 2023. [Online]. Available: https://rutter.ca/ice-navigation/.

[11] H.-M. Heyn, M. Blanke, and R. Skjetne, "Ice condition assessment using onboard accelerometers and statistical change detection," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 3, pp. 898–914, 2020.

[12] M. Johnston, R. Frederking, G. Timco, and M. Miles, "Motan: A novel approach for determing ice-induced global loads on ships," *Proceedings MARI-TECH*, vol. 2003, pp. 1–17, 2003.

[13] N. T. Kurtz *et al.*, "Sea ice thickness, freeboard, and snow depth products from operation icebridge airborne data," *The Cryosphere*, vol. 7, no. 4, pp. 1035–1056, 2013. DOI: 10.5194/tc-7-1035-2013. [Online]. Available: https://tc.copernicus.org/articles/7/1035/2013/.

[14] A. A. Petty *et al.*, "Characterizing arctic sea ice topography using high-resolution icebridge data," *The Cryosphere*, vol. 10, no. 3, pp. 1161–1179, 2016. DOI: 10.5194/tc-10-1161-2016. [Online]. Available: https://tc.copernicus.org/articles/10/1161/2016/.

[15] M. Rivas, J. Maslanik, J. Sonntag, and P. Axelrad, "Sea ice roughness from airborne lidar profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3032–3037, 2006. DOI: 10.1109/TGRS.2006.875775.

[16] R. Forsberg, K. Keller, and S. Jacobsen, "Airborne lidar measurements for cryosat validation," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, 2002, 1756–1758 vol.3. DOI: 10.1109/IGARSS.2002.1026244.

[17] C. Haas, J. Lobach, S. Hendricks, L. Rabenstein, and A. Pfaffling, "Helicopter-borne measurements of sea ice thickness, using a small and lightweight, digital em system," *Journal of Applied Geophysics*, vol. 67, no. 3, pp. 234–241, 2009, Airborne Geophysics, ISSN: 0926-9851. DOI: https://doi.org/10.1016/j.jappgeo.2008.05.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926985108000682.

[18] R. Galley, M. Trachtenberg, A. Langlois, D. Barber, and L. Shafai, "Observations of geophysical and dielectric properties and ground penetrating radar signatures for discrimination of snow, sea ice and freshwater ice thickness," *Cold Regions Science and Technology*, vol. 57, no. 1, pp. 29–38, 2009, ISSN: 0165-232X. DOI: https:

//doi.org/10.1016/j.coldregions.2009.01.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165232X09000068.

[19] P. Maijala, J. Moore, S.-E. Hjelt, A. Pälli, and A. Sinisalo, "Gpr investigations of glaciers and sea ice in the scandinavian arctic," English, in *GPR98 - 7th International Conference on Ground-Penetrating Radar Proceedings*, 1998, pp. 143–148.

[20] Q. Remund, D. Long, and M. Drinkwater, "Polar sea-ice classification using enhanced resolution nscat data," in *IGARSS'98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings.(Cat. No. 98CH36174)*, IEEE, vol. 4, 1998, pp. 1976–1978.

[21] J. H. Coggins, A. J. McDonald, and B. Jolly, "Synoptic climatology of the ross ice shelf and ross sea region of antarctica: K-means clustering and validation.," *International journal of climatology*, vol. 34, no. 7, 2014.

[22] N. S. Fučkar, V. Guemas, N. C. Johnson, F. Massonnet, and F. J. Doblas-Reyes, "Clusters of interannual sea ice variability in the northern hemisphere," *Climate Dynamics*, vol. 47, pp. 1527–1543, 2016.

[23] J. Sobiech and W. Dierking, "Observing lake-and river-ice decay with sar: Advantages and limitations of the unsupervised k-means classification approach," *Annals of Glaciology*, vol. 54, no. 62, pp. 65–72, 2013.

[24] P. Bharathi and P. Subashini, "Texture based color segmentation for infrared river ice images using k-means clustering," in *2013 International Conference on Signal Processing, Image Processing & Pattern Recognition*, IEEE, 2013, pp. 298–302.

[25] B. Weissling, S. Ackley, P. Wagner, and H. Xie, "Eiscam — digital image acquisition and processing for sea ice parameters from ships," *Cold Regions Science and Technology*, vol. 57, no. 1, pp. 49–60, 2009, ISSN: 0165-232X. DOI: https://doi.org/10.1016/j.coldregions.2009.01.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165232X09000032.

[26] A. Sandru, H. Hyyti, A. Visala, and P. Kujala, "A complete process for shipborne sea-ice field analysis using machine vision," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 539–14 545, 2020, 21st IFAC World Congress, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2020.12.1458. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240589632031870X.

[27] Q. Zhang and R. Skjetne, "Image techniques for identifying sea-ice parameters," 2014.

[28] O. De Silva, C. Daley, J. R. Dolny, G. K. Mann, R. G. Gosine, and D. Peters, "Vision analysis of pack ice for potential use in a hazard warning and avoidance system," 2016.

[29] W. Lu, Q. Zhang, R. Lubbad, S. Løset, and R. Skjetne, "A shipborne measurement system to acquire sea ice thickness and concentration at engineering scale," Oct. 2016. DOI: 10.4043/27361-MS.

[30] H. Kalke and M. Loewen, "Support vector machine learning applied to digital images of river ice conditions," *Cold Regions Science and Technology*, vol. 155, pp. 225–236, 2018, ISSN: 0165-232X. DOI: https://doi.org/10.1016/j.coldregions.2018.08.014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165232X17304251.

[31] Z. Li, W. Yang, S. Peng, and F. Liu, *A survey of convolutional neural networks: Analysis, applications, and prospects*, 2020. arXiv: 2004.02806 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2004.02806.

[32] 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*, 2020, pp. 1–5. DOI: 10.1109/IEEECONF38699.2020.9389229.

[33] N. M. Alsharay, Y. Chen, O. A. Dobre, and O. De Silva, "Improved sea-ice identification using semantic segmentation with raindrop removal," *IEEE Access*, vol. 10, pp. 21 599–21 607, 2022. DOI: 10.1109/ACCESS.2022.3150969.

[34] N. M. Alsharay, O. A. Dobre, Y. Chen, and O. De Silva, "Sea-ice classification using conditional generative adversarial networks," *IEEE Sensors Letters*, vol. 7, no. 4, pp. 1–4, 2023. DOI: 10.1109/LSENS.2023.3259202.

[35] 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*, 2021, pp. 1–7. DOI: 10.23919/OCEANS44145.2021.9705801.

[36] 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," *Journal of Physics: Conference Series*, vol. 2718, no. 1, p. 012 062, Mar. 2024. DOI: 10.1088/1742-6596/2718/1/012062. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2718/1/012062.

[37] C. Zhang, X. Chen, and S. Ji, "Semantic image segmentation for sea ice parameters recognition using deep convolutional neural networks," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, p. 102 885, 2022, ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2022.102885. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843222000875.

[38] J. Zhao, L. Chen, J. Li, and Y. Zhao, "Semantic segmentation of sea ice based on u-net network modification," in *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2022, pp. 1151–1156. DOI: 10.1109/ROBIO55434.2022.10011899.

[39] 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," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 18 004–18 019, 2021. DOI: 10.1109/JSEN.2021.3084556.

[40] N. Panchi and E. Kim, "Deep learning strategies for analysis of weather-degraded optical sea ice images," *IEEE Sensors Journal*, vol. 24, no. 9, pp. 15 252–15 272, 2024. DOI: 10.1109/JSEN.2024.3376518.

[41] "Gtc november 2021 keynote with nvidia ceo jensen huang 2021," Nov. 2021. [Online]. Available: https://www.youtube.com/watch?v=jhDiaUL_RaM.

[42] C. Brooks. "Two months breaking ice (in under five minutes)," Youtube. [Online]. Available: https://youtu.be/BNZu1uxNvlo.

[43] A. Bréhéret, *Pixel Annotation Tool*, https://github.com/abreheret/PixelAnnotationTool, 2017.

[44] K. Wada, *Labelme: Image polygonal annotation with python*, https://github.com/labelmeai/labelme, 2018.

[45] S. Kim, *Label-tool*, https://github.com/Slava/label-tool, 2019.

[46] J. Cai, S. Ding, Q. Zhang, R. Liu, D. Zeng, and L. Zhou, "Broken ice circumferential crack estimation via image techniques," *Ocean Engineering*, vol. 259, p. 111 735, 2022, ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2022.111735. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801822010861.

[47] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, 2020, ISSN: 2079-9292. DOI: 10.3390/electronics9081295. [Online]. Available: https://www.mdpi.com/2079-9292/9/8/1295.

[48] D. Stewart, D. Blacknell, A. Blake, R. Cook, and C. Oliver, "Optimal approach to sar image segmentation and classification," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 147, no. 3, pp. 134–142, 2000.

[49] H. Deng and D. Clausi, "Unsupervised segmentation of synthetic aperture radar sea ice imagery using a novel markov random field model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 528–538, 2005. DOI: 10.1109/TGRS.2004.839589.

[50] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. DOI: 10.1109/TSMC.1979.4310076.

[51] S. U. Lee, S. Yoon Chung, and R. H. Park, "A comparative performance study of several global thresholding techniques for segmentation," *Computer Vision, Graphics, and Image Processing*, vol. 52, no. 2, pp. 171–190, 1990, ISSN: 0734-189X. DOI: https://doi.org/10.1016/0734-189X(90)90053-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0734189X9090053X.

[52] S. Arora, J. Acharya, A. Verma, and P. K. Panigrahi, "Multilevel thresholding for image segmentation through a fast statistical recursive algorithm," *Pattern Recognition Letters*, vol. 29, no. 2, pp. 119–125, 2008.

[53] D. Haverkamp, L. K. Soh, and C. Tsatsoulis, "A comprehensive, automated approach to determining sea ice thickness from sar data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 1, pp. 46–57, 1995. DOI: 10.1109/36.368223.

[54] P. Lu and Z. Li, "A method of obtaining ice concentration and floe size from shipboard oblique sea ice images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 7, pp. 2771–2780, 2010. DOI: 10.1109/TGRS.2010.2042962.

[55] 2014. [Online]. Available: https://library.wmo.int/records/item/41953-wmo-sea-ice-nomenclature.

[56] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[57] F. Remondino and C. Fraser, "Digital camera calibration methods. considerations and comparisons," en, H.-G. Maas and D. Schneider, Eds., 5, ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'; Conference Location: Dresden, Germany; Conference Date: September 25-27, 2006, vol. XXXVI, Rio de Janeiro: ISPRS, 2006, pp. 266–272. DOI: 10.3929/ethz-b-000158067.

[58] J. Weng, P. Cohen, M. Herniou, *et al.*, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 10, pp. 965–980, 1992.

[59] P. Moon and D. E. Spencer, *Field theory handbook: including coordinate systems, differential equations and their solutions*. Springer, 2012.

[60] B. Sekachev *et al.*, *Opencv/cvat: V1.1.0*, version v1.1.0, Aug. 2020. DOI: 10.5281/zenodo.4009388. [Online]. Available: https://doi.org/10.5281/zenodo.4009388.

[61] M. Born, E. Wolf, and A. Bhatia, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, 1999, ISBN: 9780521642224. [Online]. Available: https://books.google.ca/books?id=aoX0gYLuENoC.

[62] J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation*, 2015. arXiv: 1411.4038 [cs.CV].

[63] V. Badrinarayanan, A. Kendall, and R. Cipolla, *Segnet: A deep convolutional encoder-decoder architecture for image segmentation*, 2016. arXiv: 1511.00561.

[64] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, 2015. arXiv: 1505.04597 [cs.CV].

[65] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs*, 2017. arXiv: 1606.00915.

[66] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, *Rethinking atrous convolution for semantic image segmentation*, 2017. arXiv: 1706.05587 [cs.CV].

[67] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, *Encoder-decoder with atrous separable convolution for semantic image segmentation*, 2018. arXiv: 1802.02611 [cs.CV].

[68] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, *Pyramid scene parsing network*, 2017. arXiv: 1612.01105 [cs.CV].

[69] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, *Unified perceptual parsing for scene understanding*, 2018. arXiv: 1807.10221 [cs.CV].

[70] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[71] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML workshop on unsupervised and transfer learning*, JMLR Workshop and Conference Proceedings, 2012, pp. 17–36.

[72] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[73] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, J. Bigun and T. Gustavsson, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 363–370.

[74] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI'81: 7th international joint conference on Artificial intelligence*, vol. 2, 1981, pp. 674–679.

[75] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[76] P. Fischer *et al.*, *Flownet: Learning optical flow with convolutional networks*, 2015. arXiv: 1504.06852 [cs.CV].

[77] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conf. on Computer Vision (ECCV)*, A. Fitzgibbon et al. (Eds.), Ed., ser. Part IV, LNCS 7577, Springer-Verlag, Oct. 2012, pp. 611–625.

[78] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[79] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, *Flownet 2.0: Evolution of optical flow estimation with deep networks*, 2016. arXiv: 1612.01925 [cs.CV].

[80] Z. Teed and J. Deng, *Raft: Recurrent all-pairs field transforms for optical flow*, 2020. arXiv: 2003.12039 [cs.CV].

[81] G. Yang, J. Manela, M. Happold, and D. Ramanan, *Hierarchical deep stereo matching on high-resolution images*, 2019. arXiv: 1912.06704 [cs.CV].

[82] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, *Random erasing data augmentation*, 2017. arXiv: 1708.04896 [cs.CV].

[83] M. Cordts *et al.*, *The cityscapes dataset for semantic urban scene understanding*, 2016. arXiv: 1604.01685 [cs.CV].

[84] Y. Zhu *et al.*, *Improving semantic segmentation via video propagation and label relaxation*, 2019. arXiv: 1812.01593 [cs.CV].

[85] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. xx, no. x, pp. xx–xx, 2008.

[86] T. Zhou, F. Porikli, D. Crandall, L. V. Gool, and W. Wang, *A survey on deep learning technique for video segmentation*, 2022. arXiv: 2107.01153 [cs.CV].

[87] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, *Video panoptic segmentation*, 2020. arXiv: 2006.11339 [cs.CV].

[88] D. Nilsson and C. Sminchisescu, *Semantic video segmentation by gated recurrent flow propagation*, 2017. arXiv: 1612.08871 [cs.CV].

[89] A. Kundu, V. Vineet, and V. Koltun, "Feature space optimization for semantic video segmentation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3168–3175. DOI: 10.1109/CVPR.2016.345.

[90] S. Chandra, C. Couprie, and I. Kokkinos, *Deep spatio-temporal random fields for efficient video segmentation*, 2018. arXiv: 1807.03148 [cs.CV].

[91] R. Gadde, V. Jampani, and P. V. Gehler, *Semantic video cnns through representation warping*, 2017. arXiv: 1708.03088 [cs.CV].

[92] M. Ding, Z. Wang, B. Zhou, J. Shi, Z. Lu, and P. Luo, *Every frame counts: Joint learning of video segmentation and optical flow*, 2019. arXiv: 1911.12739 [cs.CV].

[93] Y. Liu, C. Shen, C. Yu, and J. Wang, "Efficient semantic video segmentation with per-frame inference," *ECCV*, 2020.

[94] P.-Y. Huang, W.-T. Hsu, C.-Y. Chiu, T.-F. Wu, and M. Sun, *Efficient uncertainty estimation for semantic segmentation in videos*, 2018. arXiv: 1807.11037 [cs.CV].

[95] B. Zhou *et al.*, *Semantic understanding of scenes through the ade20k dataset*, 2018. arXiv: 1608.05442 [cs.CV].

[96] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[97] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[98] C. Szegedy *et al.*, *Going deeper with convolutions*, 2014. arXiv: 1409.4842 [cs.CV].

[99] A. G. Howard, *Some improvements on deep convolutional neural network based image classification*, 2013. arXiv: 1312.5402 [cs.CV].

[100] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016. DOI: 10.1109/JPROC.2015.2494218.

[101] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[102] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, 2019. arXiv: 1711.05101 [cs.LG].

[103] I. Loshchilov and F. Hutter, *Sgdr: Stochastic gradient descent with warm restarts*, 2017. arXiv: 1608.03983 [cs.LG].

[104] W. Quan, J. Chen, Y. Liu, D.-M. Yan, and P. Wonka, *Deep learning-based image and video inpainting: A survey*, 2024. arXiv: 2401.03395 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2401.03395.

[105] S. Zhou, C. Li, K. C. K. Chan, and C. C. Loy, *Propainter: Improving propagation and transformer for video inpainting*, 2023. arXiv: 2309.03897 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2309.03897.

[106] N. Ravi *et al.*, *Sam 2: Segment anything in images and videos*, 2024. arXiv: 2408.00714 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2408.00714.

APPENDICES

Appendix A

Training and Validation Curves

A.1 Optuna Hyperparameter Validation Curves

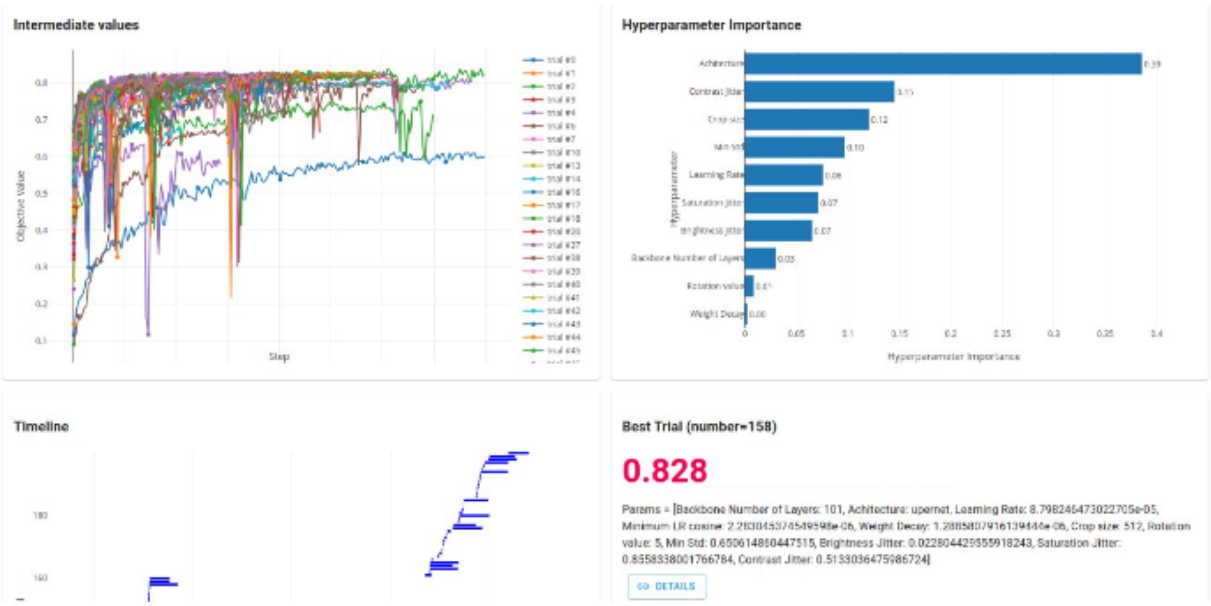


Figure A.1: Validation mean Intersection over Union (mIoU) curves for our large bayesian parameter search [100] using Optuna [101]. The best performing network was UPerNet with ResNet101 backbone, with 0.828 mIoU on validation dataset.

A.2 UPerFlow Training and Validation Curves

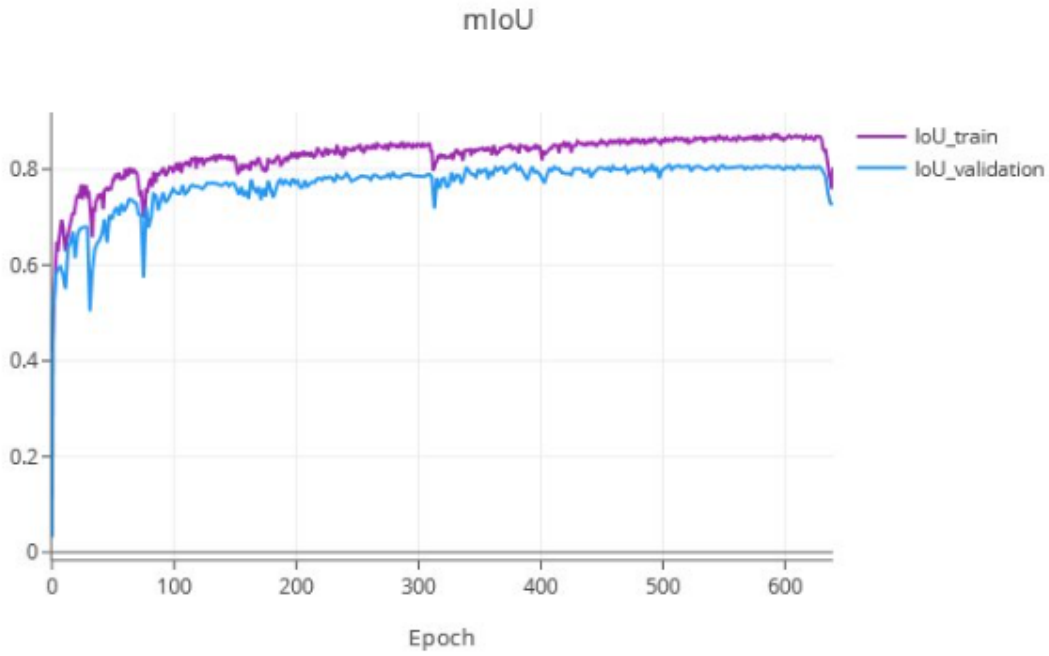


Figure A.2: The training and validation curve for our video segmentation network, UPerFlow. Our network completes training in in 637 epochs, with best validation performance 0.825 mIoU on validation dataset. Although our network performs slightly worse for validation dataset, it performs better on the test dataset, indicating that the network may have better generalization properties.