Roadmap to fully-automated, pan-Antarctic, pack-ice seal surveys

A Dissertation presented

by

Bento C. Gonçalves

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Ecology & Evolution

Stony Brook University

December 2022

Stony Brook University

The Graduate School

Bento C. Gonçalves

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Dr. Heather J. Lynch – Dissertation Advisor
IACS Endowed Chair of Ecology & Evolution

Dr. H. Resit Akçakaya – Chairperson of Defense
Professor, Ecology & Evolution

Dr. Lesley Thorne
Associate Professor, School of Marine and Atmospheric Sciences

Dr. Dimitris Samaras
Professor, Computer Science

This dissertation is accepted by the Graduate School

Celia Marshik

Interim Dean of the Graduate School

Abstract of the Dissertation

Roadmap to fully-automated, pan-Antarctic, pack-ice seal surveys

by

Bento C. Gonçalves

Doctor of Philosophy

in

Department of Ecology & Evolution

Stony Brook University

2022

Antarctic pack-ice seals, through their ecological role as key Antarctic krill predators, are critical to the Southern Ocean ecosystem. Shifts in sea ice distribution caused by anthropogenic climate change and krill fisheries threaten their populations. While initially surveyed by vessel or aircraft transects, very high-resolution remote sensing imagery has emerged as a safer and potentially cheaper alternative. The sheer volume of imagery, however, limits the spatial and temporal scale for human annotation of satellite imagery. AI-based, fully-automated surveys offer true scalability and, while imperfect, provide consistent annotations unaffected by observer fatigue or other factors external to the image itself. However, a pan-Antarctic survey using remote sensing comes with a number of challenges: 1) detecting seals in very-high-resolution imagery is a daunting task even for trained experts and relies heavily on contextual clues, making proper statistical treatment pivotal to go from detections to population estimates; 2) variability in lighting, terrain, off-nadir angle, and sea ice conditions impose severe limitations on the reliability of validation and test sets; and 3) limitations in our understanding of seal haul out behavior hamper our efforts to estimate the portion of seals available for detection (i.e. not submerged) at any moment in time. Here I present the recent advances in AI-powered seal detection and outline a schematic of the fully-automated pipeline that would be needed for regular pan-Antarctic seal surveying, along with requirements in terms of cost of imagery, personnel, and computational power. I will also discuss auxiliary components developed in support of an automated seal census, including sea ice segmentation models that are able to restrict input imagery to potential seal habitat only, human-level seal detection models, and the HPC middleware required to apply this efficiently at scale.

"Wisdom tells me I am nothing. Love tells me I am everything. And between the two my life flows."

–Nisargadatta Maharaj

Contents

List of Figures

List of Tables

List of Abbreviations

CNN	**C**onvolutional **N**eural **N**etwork
GPU	**G**raphic **P**rocessing **U**nit
GFLOP	**G**iga (i.e. 1 billion) Floating Point Operation
UAV	**U**nmanned **A**erial **V**ehicle
HPC	**H**igh-**P**erformance-**C**omputing
CV	**C**omputer **V**ision
NLP	**N**atural **L**anguage **P**rocessing
AI	**A**rtificial **I**ntelligence
RSISU	**R**emote **S**ensing **I**mage **S**cene **U**nderstanding
DL	**D**eep **L**earning
VHR	**V**ery-**h**igh-**r**esolution (satellite imagery)
ADD	**A**ntarctic **D**igital **D**atabase
GIS	**G**eographic **I**nformation **S**ystem
WV-3	**W**orld**v**iew-3 satellite
WV-2	**W**ord**v**iew-2 satellite
SAR	**S**ynthetic **A**perture **R**adar
SO	**S**outhern **O**cean
APIS	**A**ntarctic **P**ack-**i**ce **S**eal project
SOS	**S**atellites **O**ver **S**eals project
PGC	**P**olar **G**eospatial **C**enter

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr. Heather Lynch, for her patience, guidance, and encouragement throughout my unorthodox Ph.D. journey. I could not have completed this work without her support, mentorship, and advice.

I am also grateful to the members of my dissertation committee Dr. Resit Akçakaya, Dr. Lesley Thorne, and Dr. Dimitris Samaras for their insightful feedback and valuable contributions to my research.

I would like to thank my colleagues in the Lynch Lab, Department of Ecology & Evolution, and Stony Brook in general, Salim El-Ayache, Sophie Kapitan, Alex Lengers, Laurel Yohe, Kash Bandaralage, Hieu Le, Alex Borowicz, Catie Foley, Maureen Lynch, Casey Youngflesh, Michael Schrimpf, Eliot Monaco, Mihir Umarani, Stoycho Velkovsky, Sebastian Dick, Michael Wethington, to name a few, for creating a rich and stimulating environment through my time in New York.

I am indebted to the National Science Foundation for providing financial support for my research (Awards 1740595, 1043681, and 1559691) to the Polar Geospatial Center for providing access to imagery resources, and to the Pittsburg Supercomputing Center for supplying all compute resources necessary to conduct my work.

Finally, I am forever grateful to my wife, Renata, for her patience, guidance, and unwavering support through this long and winding journey — your strength and kindness inspire me to prevail against all odds.

To Renata and Tom

Chapter 1

Introduction

The Southern Ocean (SO), a mass of water encircling the Antarctic continent, harbors major seasonal hotspots for primary productivity [5] and plays a key role in maintaining ocean biogeochemical cycles [58, 25]. The SO remains encased in sea ice through the Antarctic winter, with little to no sunlight – imposing a severe limitation on productivity during that time. As the ice melts through the spring and summer, and the Antarctic continent bathes in nearly permanent sunlight, massive phytoplankton blooms pop up across the SO ecosystem [125, 21]. Phytoplankton takes advantage of the abundant micronutrients and sunlight to capture CO_2, which, after predation by primary consumers, gets sunk into the bottom of the ocean, acting as a biological pump [25, 58].

Among primary consumers, Antarctic krill (*Euphasia superba*), a small euphausiid shrimp, through its complex, multi-year, sea-ice dependent life-cycle, is able to track phytoplankton blooms through space and time. At the peak of its abundance during the austral summer, krill attains what could be the largest biomass among multicellular animals (379 million tonnes, [6]). While relatively simple at its base, the SO foodweb fans out to sustain a plethora of krill consumers [137], from squid and fish to seabirds and marine mammals. However, its strong link with sea ice dynamics [143, 113] put krill in a delicate position when faced with the anthropogenic climate change crisis [43, 69]. To make things worse, industrial krill fisheries [100, 44] have been on the rise and add a layer of difficulty in surveying krill populations and, through them, the

viability of krill consumers [129, 137]. Thus, keeping a close eye on krill populations is pivotal for gauging the health of the SO ecosystem as a whole.

Surveying krill directly to estimate its stocks, however, is a daunting task. Krill populations follow patchy distributions [138] and can perform complex horizontal and vertical migrations to track resources. Krill mega-aggregations (e.g. [102]) follow unpredictable algal blooms, making them hard to track and hampering our efforts to extrapolate global stocks from local samples [7]. Some krill predators, however, spend a considerable portion of their life above the surface, at predictable times and places. This makes them far more amenable to repeated population surveys. Among those, pack-ice seals, a group of four sea-ice-dwelling species, the crabeater seal (*Lobodon carcinophaga*), Weddell seal (*Leptonychotes weddellii*), leopard seal (*Hydrurga leptonyx*) and the Ross seal (*Omnatophoca Rossi*), are especially good candidates for not only do they haul out on sea ice, but they have a pan-Antarctic distribution and, due to their large size (2 - 4m) and the fact that they do not form tight aggregations like typical pinnipeds, they can be individually identified in very-high-resolution (VHR) remote sensing imagery (e.g. [50]). Crabeater seals, the most abundant of the four species by a large margin [56], are specialist krill predators [128, 129]. Through this tight trophic link, regular monitoring of crabeater seal populations can play the role of an ecological barometer for the SO ecosystem.

This central position as indicator species for the SO ecosystem bred many attempts at surveying pack-ice seal populations. The earliest efforts to do so are reviewed in [103] but a more recent and highly structured attempt at estimating pan-Antarctic pack-ice seal population abundance was organized by the Scientific Committee for Antarctic Research under the umbrella of the International Antarctic Pack Ice Seals (APIS) program [1]. APIS was a large collaboration between six countries from 1994 to 2000 that employed vessel- and helicopter-based line transects to sample seal densities at several locations and times. Though such studies brought invaluable insights on seal biology (e.g., [128, 114, 56]), the spatial and temporal coverages are

limited to draw continent-wide population estimates. Aerial transect-based monitoring programs such as APIS, especially in remote locations such as the SO, are not only resource-intensive [1] but dangerous for field biologists [121], making them unfeasible as means to repeatedly survey pack-ice seal populations.

Fortunately, VHR satellite imagery may soon be a viable alternative for aerial surveys, providing greater spatial coverage and, due to its dramatically lower cost, increased repeatability. The use of VHR satellite imagery for wildlife surveys has exploded in recent years and includes demonstration projects for southern elephant seals [92], polar bears [133] and African ungulates [147, 150], emperor penguins [46], whales [16], as well as seabird species whose presence and abundance can be estimated indirectly using the guano stain at the colony [73, 84]. Even at the highest resolution for commercially available imagery (0.31m / pixel), however, a 2-4 meter pack ice seal occupies only a handful of pixels. This lack of detail erodes confidence in detected objects and may create a heavy dependency on contextual clues. Moreover, characteristics such as the off-nadir angle from the sensor, lighting conditions, and cloud cover expand the breadth of scenarios encountered when annotating seals beyond the already heterogeneous sea ice landscape they inhabit. As such, seal annotation in VHR imagery is extremely time-consuming and is a skill that takes a long time to master. Although VHR imagery covers enough area for comprehensive, continent-scale surveys [53], the bottleneck from laborious expert seal annotation imposes severe limitations on spatial and temporal scales for pack ice seal surveys.

Citizen scientist campaigns can drastically improve the potential area surveyed by recruiting masses of volunteer annotators [142]. The extra manpower delivered by unskilled annotators – given the difficulty of the task at hand – comes at the obvious cost of generating less reliable labels [117]. Thus, statistical tools to provide proper treatment to unreliable labels (e.g. [117]) are pivotal to drawing ecological insights. Despite these issues, citizen scientist campaigns have been successfully applied to pack ice seal surveys in VHR in a recent effort by LaRue and colleagues in the Satellites Over Seals

(SOS) project [76]. The SOS project, running from 2016 to 2018, recruited >325,000 volunteer annotators to go over 790 Worldview-2 (WV-2) VHR scenes, with an on-nadir resolution of 0.6m / pixel, covering a totality of 268,611 km^2. The project presented volunteer annotators with 500 m x 500 m cropped WV-2 images and queried for the presence/absence of seals in such images, and treated low-quality annotations from volunteers with the CrowdRank algorithm, a consensus-based approach that ranks observers based on how much they agree or disagree with their peers. CrowdRank-derived rankings are then used to weight volunteer annotations when aggregating labels for a cropped image. While SOS covered an unprecedented area for pack-ice seal studies in VHR and used it to create the most comprehensive global population estimate for the Weddell seal to date [75], its applicability as a long-term survey program is questionable given it has only been applied and tested in a constrained scenario of hand-selected locations over fast-ice in November 2011 [76]. Moreover, the project's simplification of the annotation process to binary classification on cropped images, while understandable given the experience constraints from annotators, hinders the resolution and therefore the usefulness of the resulting output to understand the spatial aggregation patterns for pack-ice seals.

The field of computer vision (CV), particularly deep learning approaches powered by modern GPUs and large annotated datasets (e.g., [116]), has demonstrated the potential for assisting in several laborious visual tasks across different application areas (e.g., agriculture [136], construction [146], and medicine [42]). With the popularization of commercial satellites, [10], extremely large amounts of remote sensing imagery are amassed on a daily basis. In order to convert imagery into actionable insights, CV solutions for remote sensing imagery have become commonplace (e.g., sea ice segmentation and classification [51, 18, 57]). In niche applications such as detecting wildlife, however, the lack of expert annotators to create large training datasets imposes a barrier to successfully applying such methods [141]. Nonetheless, there have been several proof-of-concept studies on the applicability of CV on VHR to detect/measure whales

[17], pack-ice seals [50], and penguin colonies [85]. Among the difficulties of moving from proof-of-concept studies on wildlife detection in VHR to fully automated survey programs is ensuring that conditions at which detection models are trained and validated capture the full range of scenarios encountered when making predictions.

1.1 Roadmap to a fully-automated seal detection monitoring program

With the success of CV at tackling a series of complex tasks in related fields, I embarked on the task of developing a fully-automated tool to survey pack-ice seals. The first step to do so is amassing a body of labeled images to be used for training, with a portion of those set aside for validation (i.e. hyperparameter tuning and model selection) and testing (i.e. estimating out-of-sample performance). To maximize the certainty of annotation labels, I restricted automated seal detection work to the highest resolution available for commercial satellite imagery (0.31m / pixel), at the cost of longer processing times and lower spatial coverage. While aiming at classifying images for the presence of seals simplifies the annotation process, image-level labels provide sparse information as to why a given image was labeled as such, hampering the usability of such annotations as training samples for ML models such as CNNs. On the other hand, annotating every pixel covered by a seal is extremely laborious and makes it challenging to create a corpus of images large (and diverse) enough to train seal detection CNNs. In a compromise between the rich signal given by pixel-level seal masks and the convenient, but sparse, image-level labels, I created my training datasets by annotating seals at their center-most pixel (i.e. centroid). Since the model also needs to learn with a seal "is not", images with seal centroid annotations were complemented with a large set of seal-free images covering Antarctic coastline features such as pack-ice, glaciers, as well as land features such as rocks and emperor

penguin colonies. I then use this weakly annotated dataset to train a U-Net [115] variant, the SealNet [50], designed to make the most of the relatively weak signal from centroid annotations by simultaneously segmenting out seal centroids, classifying images for the presence of seals and counting the number of seals in each image. While SealNet worked as a pioneer, proof-of-concept study for automated seal detection in VHR, it only worked properly on the simplest scenarios, with high contrast between seals and their surrounding, and was not fit to address the complexity and breadth presented by Antarctic coastline scenery [50].

Pack-ice seals, hence their name, have a tight link with sea ice dynamics [99], both directly by relying on sea ice to breed, molt, and avoid predation from killer whales, and indirectly through their relationship with Antarctic krill – also highly dependant upon sea ice conditions [143]. As such, keeping track of sea ice conditions is key to our efforts to model and understand seal populations through space and time [99]. Beyond its ecological link to pack ice seals, sea ice conditions affect the visual complexity of seal environments in VHR imagery, potentially hindering our ability to detect them. With the intent of collecting a pivotal environmental variable to model seal presence and detectability at the same resolution for which we detected seals, I built the first ice floe segmentation model for VHR imagery [51]. Taking advantage of substantial high-performance computing (HPC) resources for hyperparameter search, this CNN-based approach is capable of segmenting out ice floes surrounded by water bodies while avoiding other common features in the Antarctic coast such as glaciers, icebergs, and rocks [51]. Beyond providing high-resolution information on floe size and sea ice cover, the CNN was able to visually capture seal habitat at a broad level, which, in practice, allowed us to narrow down the focus of seal detection models [53].

Three years past the original SealNet study [50], with access to substantial HPC resources, superior CNN architectures available, and the potential to focus on sea ice only through the sea ice segmentation model, I performed a complete overhaul in SealNet. The SealNet2.0 [52] effort started with a complete revision of annotation datasets:

the now unnecessary – and potentially confounding – land features were removed, several more annotations on novel scenes were added, and every single scene with seal annotations was scoured to simultaneously increase the number of positive training samples and reduce the number of false-negatives in our training set. The last addition to our annotation dataset, was a set of 300 unique, randomly sampled, 1km^2 WV-3 scene crops, spanning across 87 scenes, with overlapping annotations by three human observers. The latter provides a trustworthy approximation for scenes encountered when processing new imagery, allowing for robust estimates of out-of-sample performance. Taking full advantage of this vastly improved dataset, SealNet 2.0 uses a tree-based model [28] to combine output from several U-Net CNNs into predictions that outperform those of two human observers in terms of out-of-sample precision and recall. The addition of a tree-based model also comes with the desired property of model-derived probabilities that are closer to "truth" annotation labels, making this approach more amenable for modeling detection errors [52].

In a comprehensive comparison with other available sampling methods for pack-ice seal surveys [53] able to provide output at a lower cost than every sampling method but helicopters, with the advantages of 1) storing permanent records for further verification; 2) avoiding the considerable dangers involved with operating aircraft in the Antarctic; 3) potential to cover higher spatial and temporal dimensions; 4) ease of scalability with the addition of more GPUs; and 5) several orders of magnitude fewer CO_2 emissions. At its current state, SealNet2.0 delivers reliable putative seal locations in scenarios where seals can be confidently found by human observers in VHR [52]. To obtain continental-scale estimates for seal densities over units of time and area, we need statistical models to address: 1) the probability of detecting seals given environmental features such as sea ice conditions and visual complexity of the surrounding environment; 2) seal haul-out behavior through space and time; and 3) the relationship between the presence of seals and environmental factors such as bathymetry, distance to the continental shelf and primary productivity. Further reducing the focus of

SealNet2.0 predictions by using an ancillary CNN classifier that flags locations where seals, if present, could be accurately spotted has the potential to reduce the considerable detectability-related difficulties. In the present work, I demonstrate the viability of a fully-automated, CNN-based, detection pipeline as an accurate, clean, and cost-efficient tool for regularly monitoring Antarctic pack-ice seals.

Chapter 2

SealNet: A fully-automated pack-ice seal detection pipeline for sub-meter satellite imagery

2.1 Abstract

Antarctic pack-ice seals, a group of four species of true seals (Phocidae), play a pivotal role in the Southern Ocean foodweb as wide-ranging predators of Antarctic krill (*Euphausia superba*). Due to their circumpolar distribution and the remoteness and vastness of their habitat, little is known about their population sizes. Estimating pack-ice seal population sizes and trends is key to understanding how the Southern Ocean ecosystem will react to threats such as climate-change-driven sea ice loss and krill fishing. I present a functional pack-ice seal detection pipeline using Worldview-3 imagery and a Convolutional Neural Network that counts and locates seal centroids. I propose a new CNN architecture that detects objects by combining semantic segmentation heatmaps with binary classification and counting by regression. My pipeline locates over 30% of seals, when compared to consensus counts from human experts, and reduces the time required for seal detection by 95% (assuming just a single GPU). While larger training sets and continued algorithm development will no doubt improve classification accuracy, my pipeline, which can be easily adapted for other large-bodied

animals visible in sub-meter satellite imagery, demonstrates the potential for machine learning to vastly expand my capacity for regular pack-ice seal surveys and, in doing so, will contribute to ongoing international efforts to monitor pack-ice seals.

2.2 Introduction

Antarctic pack-ice seals (crabeater seals [*Lobodon carcinophaga*], Weddell seals [*Leptonychotes weddelli*], leopard seals [*Hydrurga leptonyx*] and Ross seals [*Omnatophoca rossii*], within the Phocidae family), as a group, are key krill predators in the Southern Ocean (SO). Monitoring their populations through vessel-based and aircraft-based surveys has been a major task for Antarctic research programs (e.g. the APIS program [1]). While previous surveys provide important baselines for seal population sizes (e.g. [56]), very-high-resolution satellite imagery has been proposed as a potentially more cost-efficient and scalable tool for surveying large-bodied animals inhabiting remote locations such as southern elephant seals [92], polar bears [133] and African ungulates [147, 150]. While large enough to be seen in VHR imagery, pack-ice seals are particularly hard to detect since their preferred haul-out environment, pack ice [72, 11] changes on short (hourly) and long (seasonal) time scales, and the information content of each individual seal in an image is exceptionally low (Fig. 2.1).

Though it is possible to find seal-sized objects in VHR imagery manually, this laborious approach is only feasible at local scales (e.g., [77]), introduces observer biases [35], and is not easily scaled to allow annotation of VHR images captured within the range of pack-ice seals. Thus, repeatable, large-scale wildlife surveys require automated detection systems [31]. Traditional pixel or object-based methods for remote sensing scene understanding (RSISU) (e.g. [70, 93]), perhaps due to their reliance on hand-crafted features and spectral signatures, struggle at the increased granularity posed by high spatial resolution satellite imagery. As is the case for many fields such as computer vision [139] and natural language processing [36], deep learning, in the

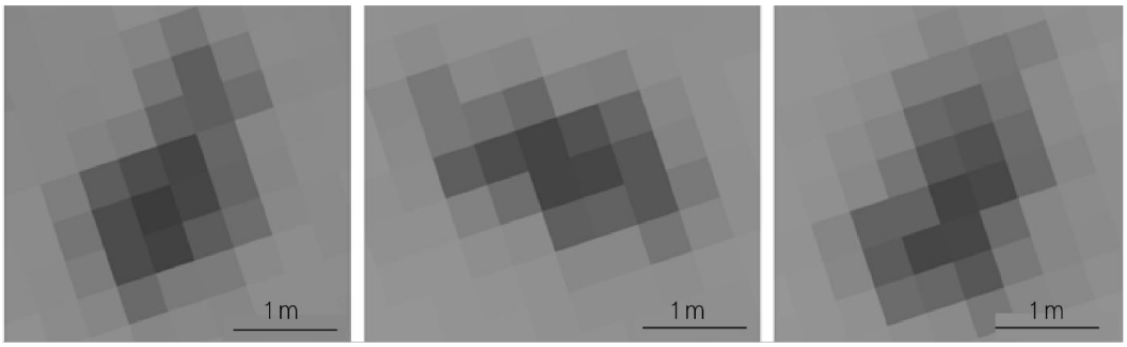

FIGURE 2.1: Individual pack-ice seals viewed in panchromatic WV-3 imagery. The scale bar shows one meter, with a typical seal about 2.5 m long covering 20–30 pixels (in total). All three seals in this image were extracted from the same scene, but the resolution may change depending on the angle at which the image was captured by the WV-3 sensor. Satellite imagery copyright DigitalGlobe, Inc. 2019.

specific flavor of Convolutional Neural Networks (CNNs), are now the state-of-the-art for RSISU [55] and is likely my best candidate for automated seal detection in high spatial resolution imagery. CNNs work by learning a series of convolution kernels – analogous to image processing kernels – as they learn to map inputs in the training data to their corresponding labels. CNNs have now been successfully employed in many ecological settings such as identifying whales [17, 10], finding mammals in the African Savanna with UAV imagery [67], and classifying animals in camera trap pictures [101].

In this work, I explore the viability of CNNs to locate pack-ice seals in Antarctica and the scalability of this approach, with the ultimate goal of facilitating continental-scale population counts for pack-ice seals and other large bodied animals. Like many other wildlife detection sampling schemes [67, 147], however, the vast majority of the VHR imagery contains no true positives (i.e. seals), creating the potential for significant false positives even if the false positive rate is low. I propose a seal detection pipeline that i) determines whether a portion of the image is occupied by seals; ii) counts seals in that portion of the image and; iii) locates the centroid of each identified

seal. All of the above is performed in a single pass with my proposed CNN architecture, SealNet. 1 In my validation and test sets, this approach is superior to pure regression or semantic segmentation approaches.

2.3 Materials and Methods

2.3.1 Selecting imagery

For this pipeline, I use Worldview 3 (WV-3) imagery provided by DigitalGlobe, Inc., which has the highest available resolution for commercial imagery with a 0.3 m at-nadir resolution in panchromatic imagery and 1.5 m with 16 multispectral bands (Red, Green, Blue, Red Edge, Coastal, Yellow, and 2 near-infrared bands). Only the panchromatic band was used for this work because individual seals are difficult to spot at lower resolutions and because the color information is not highly informative for classification (at least for human interpreters). Due to GPU memory limitations imposed by my CNN architecture, we subdivide WV-3 scenes into 224 × 224 pixel images (hereafter 'patches') (Fig. 2.2). Prior to prediction, each WV-3 scene is split into approximately 500,000 patches, keeping a 75% overlap between neighboring patches to ensure corners are not overlooked by the CNN classifier.

2.3.2 Building a training set

A training set with 75,718 raw training samples was manually assembled to train seal detection CNNs. Raw training samples are generated by extracting 450 × 450 pixel images (hereafter 'raw training patches'), roughly covering two hectares, at predefined locations (i.e. training points) on a total of 34 WV-3 scenes (Fig. 2.3) selected from the Polar Geospatial Center catalog. Training points were annotated by visually inspecting WV-3 scenes for locations with the 10 following features: seals on pack ice, seals on fast ice, emperor penguin colonies, marching emperor penguins, cracks on the sea ice,

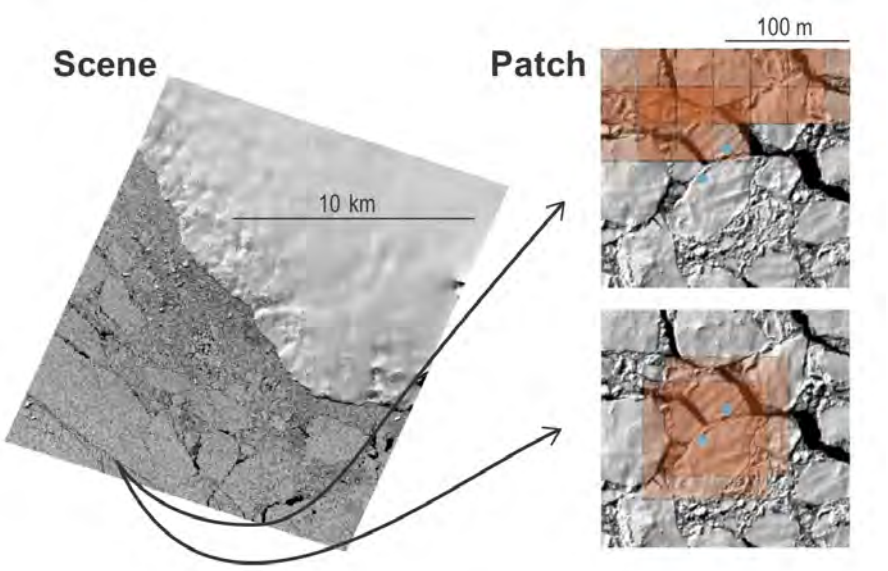

FIGURE 2.2: Sampling scheme. WV-3 scenes are split into smaller 'patches' to fit CNN requirements for input size. To create a training set, 450 × 450 training patches are extracted around features of interest (light-blue circles) on a scene, which may overlap depending on how close features are and CNN input size (orange squares on the bottom right). For prediction, whole scenes get chopped into 224 × 224 pixel patches using a sliding-window approach, with a stride that keeps a 75% overlap on both the x and y axes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) Satellite imagery copyright DigitalGlobe, Inc. 2019.

glaciers, fast-ice, pack ice, rock outcrops, and open water. For the last seven categories (background/non-target), I place an array of equidistant training points, separated by 100 m, over areas where a particular class was predominant, removing any occasional points that did not fit into that class. For emperor penguin colony points, I covered colonies with a similar array of equidistant points, with a 10 m distance between neighboring points. Groups of three or more emperor penguins arranged in lines were labeled as marching emperors, with training point annotations centered on one of the penguins. Since crabeater seals, Ross seals, and leopard seals are confined to pack-ice habitat [8, 12, 127], and the first species is far more abundant than the latter two [130, 131, 129], every seal on pack-ice in my training set is assumed to be a crabeater seal. Seals on fast ice are assumed to be Weddell seals since that is the only one of the four species that is strongly associated with fast ice habitat [12, 127]. To reduce the

annotation effort, my seal training points – both Weddell and crabeater – consist of a single point, placed at the centroid of each seal. When generating seal training images, I include the location of seal centroids within those images along with the image itself – necessary to derive ground truth seal locations and counts within training patches. Finally, my seal detection CNN, trained on the training set described above, was deployed on 18 new scenes, where seals could not be found upon visual inspection, generating a total of 10,766 training points, which were then added to the training set as a separate class for seal-shaped shadows. To evaluate and select models during training, my training data was split into training and validation sets. To prevent spatial overlap between training and validation images, I split entire groups of seals between training and validation, keeping roughly 90% of the seal training points for training and the remaining for validation. Background class training points were split by scene, where each scene with training points for a given background class is either used for training or validation. Background scenes were also split to keep roughly 90% of the training points for training and the remaining 10% for validation.

2.3.3 Setting up the convolutional neural network

Our seal detection pipeline (Fig. 2.4) detects seal centroids in VHR imagery following 4 steps: 1) tile input scene into 'patches' that can be classified using the CNN; 2) run each patch through the CNN to get a probability of harboring one or more seals (occupancy probability), a seal count, and a seal centroid intensity heatmap; 3) remove predictions below a predefined occupancy probability threshold; and 4) find the n greatest intensity peaks in the heatmap, where n is the seal count. My proposed CNN architecture was assembled by adding two branches to the U-Net [115] architecture: a branch for occupancy, branching out of the second U-Net max-pooling layer, based on the DenseNet [62] architecture and a branch for counting, branching out of the fourth U-Net max-pooling layer, based on the WideResnet architecture [152]. Apart from the

regular intensity heatmap output from U-Net, my architecture also outputs an occupancy probability and a seal count. All CNNs used here were implemented in PyTorch [107].

2.3.4 Training

Our U-Net variant is trained to minimize the difference between predicted seal counts and true seal counts, the difference between predicted occupancy and true occupancy ('1' if there is at least one seal in the patch, '0' otherwise), both measured with a Huber loss and the difference between the predicted heatmap, with a sigmoid transform, and an array with '1' over seal centroid pixels, smoothed around the centroid with a 5×5 Gaussian kernel, and '0' anywhere else – measured with a binary cross-entropy loss. To ensure that seal training images and seal centroids within those training images are as important during training as the more prevalent background training images and non-centroid pixels, binary cross-entropy losses were weighted using the ratio between the former and the latter. Training is performed using an AdamW optimizer [82] for 75 epochs (i.e. 75 complete runs through the training set), with an initial learning rate of 1×10 3, which was gradually tapered down to 1×10 5 using a cosine annealing learning rate scheduler [81], and a batch size of 64. Training images are sampled with replacement from the training set using a weighted sampler that ensures equal representation between training classes. Training images are normalized to have similar means and variances and augmented using left-right mirroring, bottom-up mirroring, random rotations (0–180), slight changes to brightness and contrast, random resized crops (0.675–1.2 of original scale, keeping the original aspect ratio) to the input size required by the CNN (224×224 in the current pipeline) and hide-and- seek transformations [124]. Whenever cropping and hide-and-seek transformations are applied to training images, seal locations within these, the number of seals on them, and whether they are occupied are updated to reflect those of the augmented sample. For

testing purposes, models were retrained with the same hyperparameter settings using all training and validation samples for 150 epochs.

2.3.5 Validation

Our model is validated at the end of each epoch. Prior to predictions, validation images are normalized and center-cropped to the dimensions required by the CNN. Similar to training, true counts and seal locations are adjusted to cropped validation images. Raw predictions on the validation set are converted to precision,

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

recall

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

and f1-score

$$F1 = precision * recall$$

where predicted seal centroids separated by no more than 5 pixels from a ground-truth seal centroid are considered true positives. At the end of each validation phase, validation losses, precision, recall and f1-scores are recorded. Whenever the f1-score surpasses the previous best score, a model checkpoint with the weights for that formulation is saved.

2.3.6 Testing

To test how SealNet generalizes to new imagery, I estimated out-of-sample precision and recall by comparing model-generated seal locations with those from the consensus of two experienced human observers on five novel scenes. First, test scenes were

independently counted by two observers with experience surveying seals in Antarctica and using VHR imagery (hereafter 'observer 1' and 'observer 2'). When looking for seals, observers followed a standardized counting procedure using a grid search system with 2 km × 2 km grid cells that were each exhaustively searched for potential seals. To create a consensus seal dataset for testing model performances, I started with seal points flagged by both observers. Points flagged by a single observer, after being stripped of observer ID, were independently reviewed by both observers, adding further seal points where both observers agreed upon to the final consensus dataset. Prior to model predictions on the test set, test scenes are tiled out into patches, with a 75% overlap between neighboring patches. Whenever multiple model predictions from overlapping tiles output seal centroids within 1.5 m of each other, the centroid with the highest heatmap intensity value is kept and the remaining centroids are discarded. My test set (Fig. 2.5) includes a pair of scenes over pack ice, with high (1.16 seals/km^2) and low (0.51 seals/km^2) seal densities, a pair of scenes over fast-ice, with high (4.06 seals/km^2) and low (0.30 seals/km^2) seal densities and a scene without seal detection by the observers. Apart from variations in seal density, test scenes were chosen to emulate scenarios likely to impact seal detectability, such as off-nadir angles and lighting conditions.

2.4 Model evaluation

To test the SealNet architecture, precision and recall obtained on test and validation sets with the full model are compared to those obtained with two simplified variants: i) the original U-Net architecture, trained only on heatmap matching; and ii) U-Net with a branch for counting. Due to the lack of a regression layer, counts on the original U-Net are obtained by applying a sigmoid transformation to the heatmap, thresholding sigmoid transformed values by 0.1 (i.e. any value lower than 0.1 is set to 0), and summing over all cells. Pure regression CNNs (e.g., CountCeption [108])

were not tested here because the lack of input-image-sized heatmaps for counts hampers localization and makes it difficult to match predicted centroids to ground-truth centroids. Finally, due to the relatively small size of my training set, the potential for improvement by acquiring more training data is investigated with a learning curve: I train my models with increasingly larger subsets (n = 100, n = 300, n = 1000, n = 3000, n = 10,000 and n = 30,000) of my training set for 15 epochs and plot the highest validation f1-scores – measured on the full validation set – against training set size. To maintain equal representation between training classes while generating the learning curve, training set subsets were sampled, without replacement, from the full training set using a weighted sampler. Though there are too few training images in some classes (e.g., Weddell seals, n = 981) to keep classes balanced at the largest subset (n = 30,000), the weighted sampler draws images with replacement. This ensures that, though the full training set itself may not be balanced, batches of training images still have equal class representation. Apart from the reduced number of epochs (75 vs. 15), CNNs on reduced training sets were trained with the same hyperparameters as their counterparts trained on the full training set.

2.5 Results

2.5.1 Validation

SealNet, with added branches for counting and occupancy, attained 0.887 precision and 0.845 recall at my validation set, outperformed base U-Net (precision = 0.250, recall = 0.993), but was slightly outperformed by U-Net + count (precision = 0.897, recall = 0.853) (Fig. 2.6a). Adding a counting branch to U-Net, when compared with heatmap thresholding approach, improved precision at my validation set more than threefold, at the cost of a small decrement in recall. Adding an occupancy branch to U-Net + count caused a slight decrease in precision and recall at my validation set.

My validation metrics for SealNet use an occupancy threshold of 0.1 (i.e., patches with a predicted occupancy probability lower than 0.1 are discarded by the model), which can be tuned to tradeoff recall for precision and vice-versa. The learning curve for SealNet (Fig. 2.6b) shows that validation f1-score increases as I add more training data, suggesting that adding new samples to my training set would be very beneficial to model performance.

2.5.2 Testing

Combining results from all five test scenes (Fig. 2.5) and comparing the results with consensus counts from two human experts, SealNet outperforms U-Net + count on f1-score, while both CNN architectures get better precision and recall than U-Net (Table 2.2). When deployed on an empty test scene (Fig. 2.5, subpanel e), SealNet was the only architecture to not produce a single false positive; U-Net produced 26 false positives and U-Net + count produced a single false positive. When aggregating predicted seals by group (using a 20-meter distance criterion to define group membership), U-Net + count has a superior recall on finding lone seals than the other two architectures (0.311 vs. 0.230 [SealNet] and 0.196 [U-Net]) on lone seals, while SealNet is superior on finding seals inside groups from 3 to 5 seals and more than 10 seals (Fig. 2.8a). Groups with a small number of seals were far more prevalent in my test scenes (Fig. 2.5) than larger ones (Fig. 2.8b).

2.5.3 Human observer performance vs. CNN performance

When compared with a consensus review from both human observers, individual human observers made a considerable number of mistakes and were inconsistent across different scenes (Table 2.2, see Observer 1 and 2). Even so, the human observer outperformed the CNNs tested in this analysis but at the expense of considerably more processing time (Table 2.2).

TABLE 2.1: Training set classes. For each of the first 10 classes, patches were manually annotated on WV-3 imagery following the annotation method listed for that label. Training patches under the 'shadow' label were extracted using an early iteration of SealNet to find seals on scenes without seals. Note that the total number of scenes is smaller than the added number for all labels since there are often several different labels in a single scene.

Class label	Annotation method	N patches	N scenes -
Crabeater	1 patch centered on each individual seal	4238	6
Weddell	1 patch centered on each individual seal	981	15
Emperor	Array of patches with 10 m gaps over colony	7124	19
Marching-emperor	1 patch centered on each penguin line	1064	18
Pack ice	Array of patches with 100 m gaps over area	17771	10
Ice-sheet	Array of patches with 100 m gaps over area	20694	9
Glacier	Array of patches with 100 m gaps over area	5762	4
Crack	Array of patches with 100 m gaps over area	1449	4
Rock	Array of patches with 100 m gaps over area	4836	6
Open-water	Array of patches with 100 m gaps over area	11799	5
Shadow	Extracted from CNN output in scenes with no seals	10766	518
Total	-	86483	52

2.6 Discussion

2.6.1 CNN performance

Even with a relatively small training set (Table 2.1), weakly-supervised training samples, and a test set with only 1168 seals distributed over 150,000 non-overlapping patches, my pipeline often produces reasonable predictions, including unmistakable seals missed by my double-observer count (Fig. 2.7). In contrast with typical usages of deep learning for RSISU, which relies on bounding box-based approaches (e.g. YOLO [112]), I explore instance-based approaches, in the form of U-Net variants, for object detection in remote sensing scenes. Apart from requiring lower annotation effort (i.e. centroids vs. bounding boxes), my approach excels at object localization with little to no post-processing non-maximum suppression efforts. Similar to Le et al. [85], this work highlights the potential for weakly-supervised approaches on remote sensing tasks.

When compared with test set counts from my double-observer consensus set, my most sensitive model finds over 35% of seals, generating 1321 false positives, while my most precise model finds 30% of seals, generating 604 false positives. Perhaps due to the lower information content, seals hauling out by themselves were more often missed by the CNNs than those in larger groups (Figs. 2.7, 2.8a). Although group size had a profound impact on recall (Fig, 2.8a), effects were not consistent across my 3 model architectures. These variations can be used to divide predictions between different models (e.g., U-Net + count on small groups and SealNet on large ones), model detection errors, and, in semi-automated pipelines, highlight predictions that will require more attention from human observers. SealNet predictions are still not as reliable as those obtained by an experienced human observer. However, using my pipeline with a single modern GPU is over 10× faster than counting by hand. While similar studies in wildlife samples from aerial or VHR imagery like Salberg [118] and Xue et al. [147] report higher performance scores, the first uses aerial imagery, with far superior spatial resolution, and the second relies on expert opinion for inference, in contrast to my fully automated pipeline. Moreover, both use less rigorous testing than demonstrated here for SealNet (i.e. small test sets, cross-validation on the training set, and overlap between training and testing scenes).

There are large differences between validation (Fig. 2.6a) and test performance metrics (Table 2.2), with test performance showing dramatically lower precision and recall. This outcome may be explained by the relatively small size of my validation set (8715 patches across 49 scenes) and by the small number of test scenes (n = 5), which may have caused my validation set to be insufficiently representative of the problem at hand (i.e. finding seals in WV-3 scenes from the Antarctic coastline) and/or my test set to be a biased sample of typical Antarctic scenes. Similar to results from Aich and Stavness [2], combining heatmap activation with counting by regression output improves precision and recall (Fig. 2.6a). Besides greatly improving precision at the cost of some recall at my chosen threshold of 0.1, my occupancy branch gives us the

TABLE 2.2: Test performance. Predicted count, precision, and recall using all model variants are shown for scenes a–d. I only include a predicted count for scene e because we cannot get meaningful precision or recall scores without ground-truth seal points. Performance metrics are obtained by comparing model-predicted seal locations with a consensus review from two experienced human observers. Patch counts reflect a stride that keeps a 75% overlap between neighboring cells (see Fig. 2.2).

Model	Scene a Consensus ct: 106 Patches: 127964	Scene b Consensus ct: 732 Patches: 127332	Scene c Consensus ct: 282 Patches: 138308	Scene d Consensus ct: 48 Patches: 78334	Scene e Consensus ct: 0 Patches: 173340
SealNet	Count: 57 Precision: 0.492 Recall: 0.277	Count: 809 Precision: 0.344 Recall: 0.377	Count: 58 Precision: 0.519 Recall: 0.133	Count: 33 Precision: 0.324 Recall: 0.224	Count: 0 - -
U-Net	Count: 461 Precision: 0.049 Recall: 0.207	Count: 4865 Precision: 0.049 Recall: 0.319	Count: 906 Precision: 0.0073 Recall: 0.223	Count: 246 Precision: 0.032 Recall: 0.163	Count: 26 - -
U-Net + count	Count: 191 Precision: 0.179 Recall: 0.315	Count: 1267 Precision: 0.240 Recall: 0.402	Count: 139 Precision: 0.430 Recall: 0.226	Count: 131 Precision: 0.134 Recall: 0.353	Count: 1 - -
Observer 1	Count: 50 Precision: 0.641 Recall: 0.373	Count: 1321 Precision: 0.527 Recall: 0.777	Count: 299 Precision: 0.569 Recall: 0.593	Count: 45 Precision: 0.639 Recall: 0.613	Count: 0 - -
Observer 2	Count: 168 Precision: 0.580 Recall: 0.784	Count: 732 Precision: 0.635 Recall: 0.635	Count: 218 Precision: 0.533 Recall: 0.437	Count: 72 Precision: 0.527 Recall: 0.716	Count: 1 - -

flexibility to tradeoff precision and recall by picking more strict or lenient threshold values. Learning curve results (Fig. 2.6b) suggest that acquiring more training data can lead to substantial improvements in prediction accuracy, though I caution that my validation set may not be sufficiently representative of future test settings. Though our use of dropout, hide-and-seek transforms during data augmentation and multi-task learning should help with reducing overfitting, the large gap between validation and test performance metrics hints that our models are overfitting the validation set, and cautions for more rigorous model selection approaches in such studies.

Similar to other ecological sampling settings with aerial imagery [19, 67], empty patches predominate in my test set, outnumbering those with seals by a factor of 500 – which is aggravated by my use of overlapping patches. Though the pervasiveness of false negatives and false positives in my model output calls for adequate statistical treatment before making any inferences about seal populations (e.g., [97, 110]),

raw model output from SealNet can also serve as an attention map for human observers, facilitating manual annotation. Besides being immediately applicable to semi-automated pack-ice seal surveys, lower annotation effort speeds up the acquisition of training data, which, as indicated by my learning curve (Fig. 2.6b), may boost prediction performance enough to bridge the gap between fully-automated approaches and manual surveys. When considering the performance of an automated classification pipeline, it is worth highlighting that there is a comprehensive literature on statistical methods for dealing with observation errors in wildlife surveying (e.g., [91, 97, 96]), and the existence of high error rates does not ipso facto preclude unbiased or reliable population estimates. While the vast majority of high spatial resolution imagery for the Antarctic is focused on terrestrial areas, there are no technical barriers to repeated sampling of key marine regions, and the development of a well-structured sampling regime for pack-ice seals could enable global population estimates on a yearly basis.

Besides exploring hyperparameter space and adding more training samples, predictive performance could be improved by adding environmental features as additional inputs to the CNN. This approach, akin to a habitat suitability model, takes advantage of the fact that all my training and input imagery is georeferenced and may be coupled with several biologically relevant measurements associated with seal presence/density in previous studies such as sea ice characteristics, bathymetry, water temperature and distance to shelf-break [8, 12, 14, 99]. Integrated habitat suitability and detection models provide a promising path forward and would likely improve classification accuracy while potentially informing on seal habitat selection and haul out patterns.

2.6.2 A path forward for a Southern Ocean pack-ice seal monitoring program

Past large-scale pack-ice seal surveys, both ship-based (e.g., [41]) and aerial-based [56], have provided initial clues on pack-ice seal distribution and population sizes. These estimates, unavoidably, rely on data aggregated from different sampling methods (e.g., [12, 56]) and from broad and discontinuous time windows (e.g., [1, 41]), or require an extrapolation for the entire coastline using seal density estimates derived from a single region [41]. Though mitigated by increasingly sophisticated statistical treatment, I argue that these limitations, due to the substantial expenditure required for Antarctic surveys, are inherent properties of sampling method choices. As my pilot study illustrates, satellite-based surveys, aided by automated detection with CNNs, can overcome these difficulties and provide large-scale pack-ice seal censuses, using a single sampling design that can be repeated yearly at spatial scales that cover a representative sample of the Antarctic coastline. This paper represents the first step towards a regular, cost-effective, pan-Antarctic seal monitoring program. Apart from algorithmic improvements and more training data, my pipeline would greatly benefit from a test set that is paired with concurrent observations by a ground-based observer or aerial photographs. A comprehensive pack-ice seal census would provide key information to understand how the SO ecosystem will react to threats such as climate-change-driven sea ice loss [78] and increasing krill fisheries [100]. An automated tool to survey pack-ice seals allows us not only to get a better idea about their abundance and long-term trends but also how their distribution is coupled to environmental features (e.g., sea ice conditions) or affected by external drivers (e.g., krill fisheries). This approach can be easily adapted for counting other large-bodied species visible from high spatial resolution satellite imagery, and I have provided the code to encourage other researchers to adapt the pipeline for their needs.

FIGURE 2.3: Training set scenes. Locations of the 52 WorldView-3 scenes used on the training set are marked with light-blue squares. Scenes with spatial overlap were captured at different times. Training set scenes range from October 2014 to February 2017. The scarcity of offshore scenes in my training set reflects the preponderance of coastal scenes on available WV-3 imagery. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.) Antarctic basemap extracted from Quantarctica (Matsuoka et al., 2018).

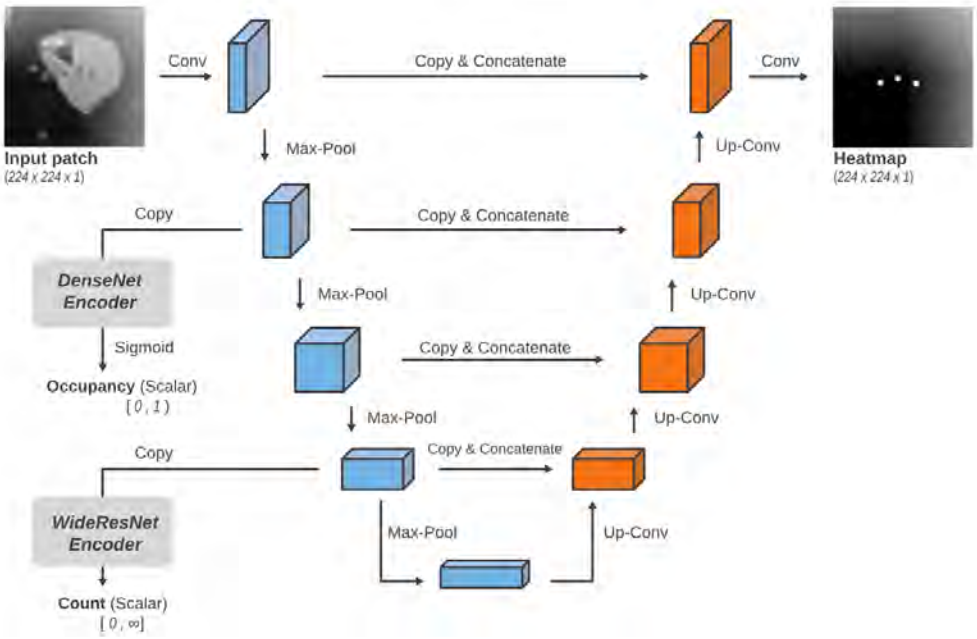

FIGURE 2.4: ealNet architecture. The CNN takes in a patch as input, generates an occupancy probability and a seal count with peripheral branches, and reconstructs a heatmap for pixel-wise probability of being a seal centroid. Predicted seal centroids are determined outside of the CNN by finding the n largest intensity peaks on a patch, where n is the regressed seal count for that patch multiplied by a Boolean (0 or 1) indicating whether the occupancy probability for that patch surpasses a predefined threshold. Model output is displayed in bold. Satellite imagery (upper left) copyright DigitalGlobe, Inc. 2019.

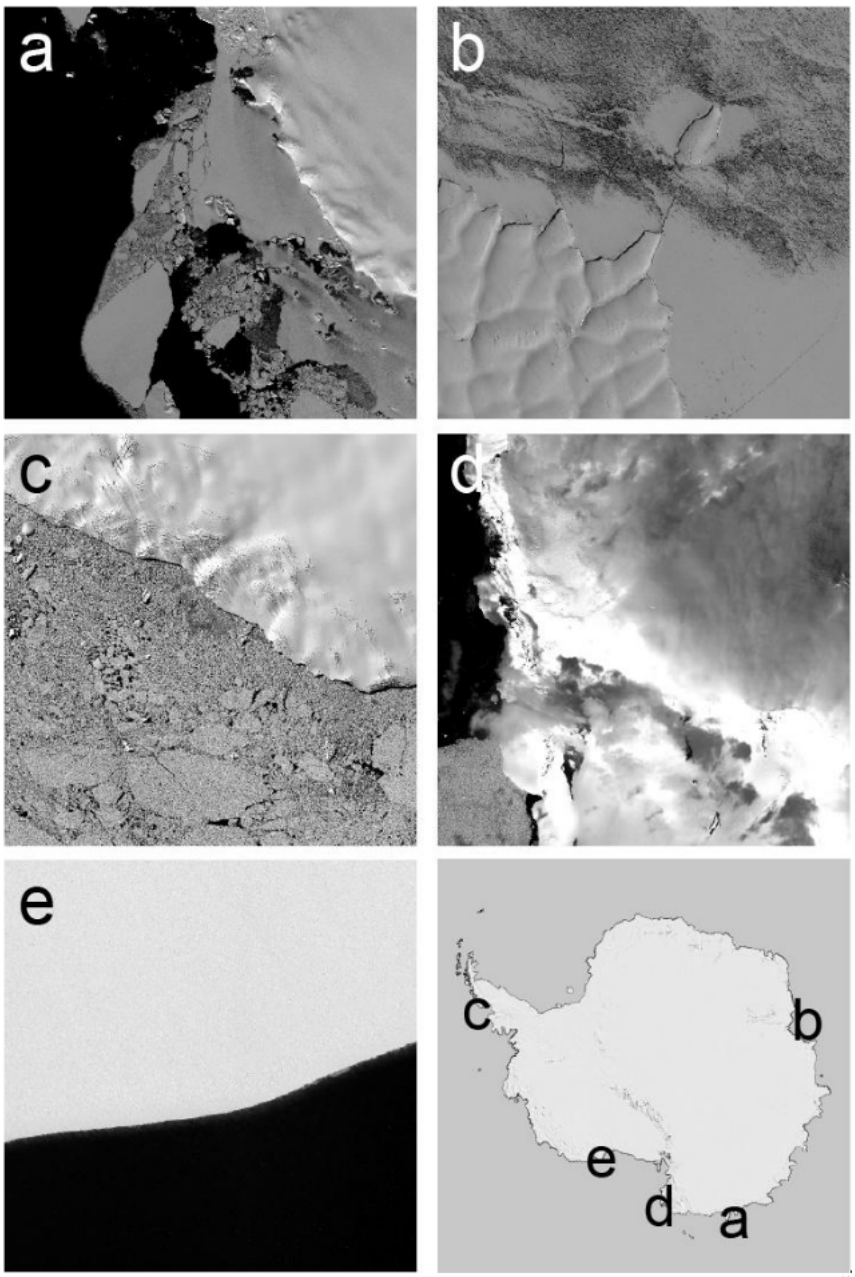

FIGURE 2.5: (a–e) Satellite imagery representing the test set. Test set scenes are not included in training or validation sets and serve as a way to get out-of-sample precision and recall over a range of scenarios that we are likely to encounter at deployment stages. Scenes a and b have seals over fast-ice, with low and high densities, respectively. Scenes c and d have seals over pack ice, with low and high densities, respectively. Scenes e covers the Antarctic coastline landscape without seals. All test scenes were obtained between February and March 2017 at the locations specified in the Antarctic continent thumbnail at the lower right of the panel. Antarctic basemap extracted from Quantarctica [90]. Satellite imagery copyright DigitalGlobe, Inc. 2019.

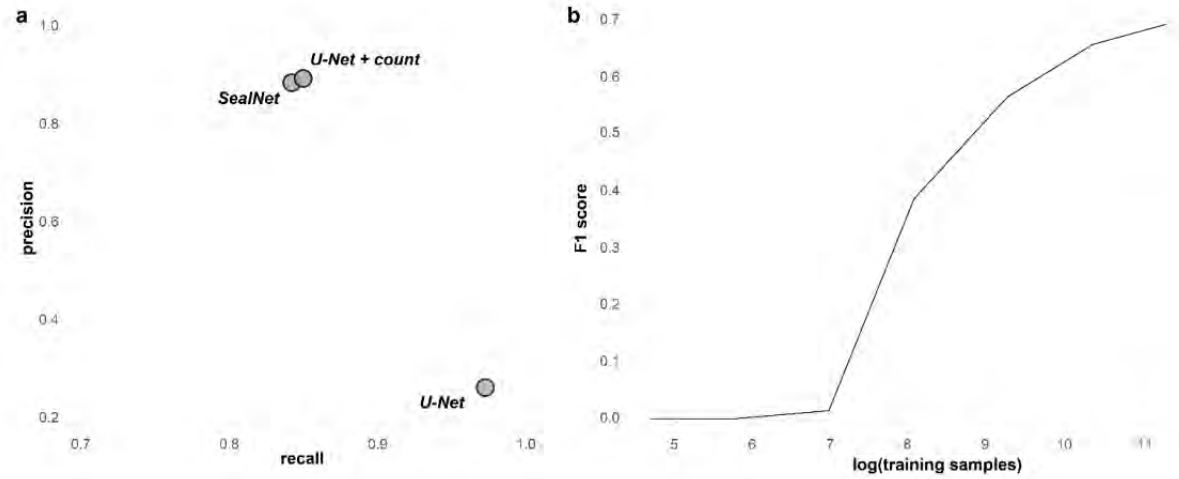

FIGURE 2.6: (a) Validation performance: Validation precision and recall reported here are the highest obtained for 75 training epochs. Predicted seal centroids are considered true positives if their location is within 5 pixels of a manually annotated seal centroid. (b) Learning curve for SealNet: Top validation f1-score obtained during training epochs is displayed for SealNet instances trained on increasing large random subsets of my training set. Training set subsets are generated using a weighted sampler that ensures a similar class representation regardless of the number of training samples on a subset.

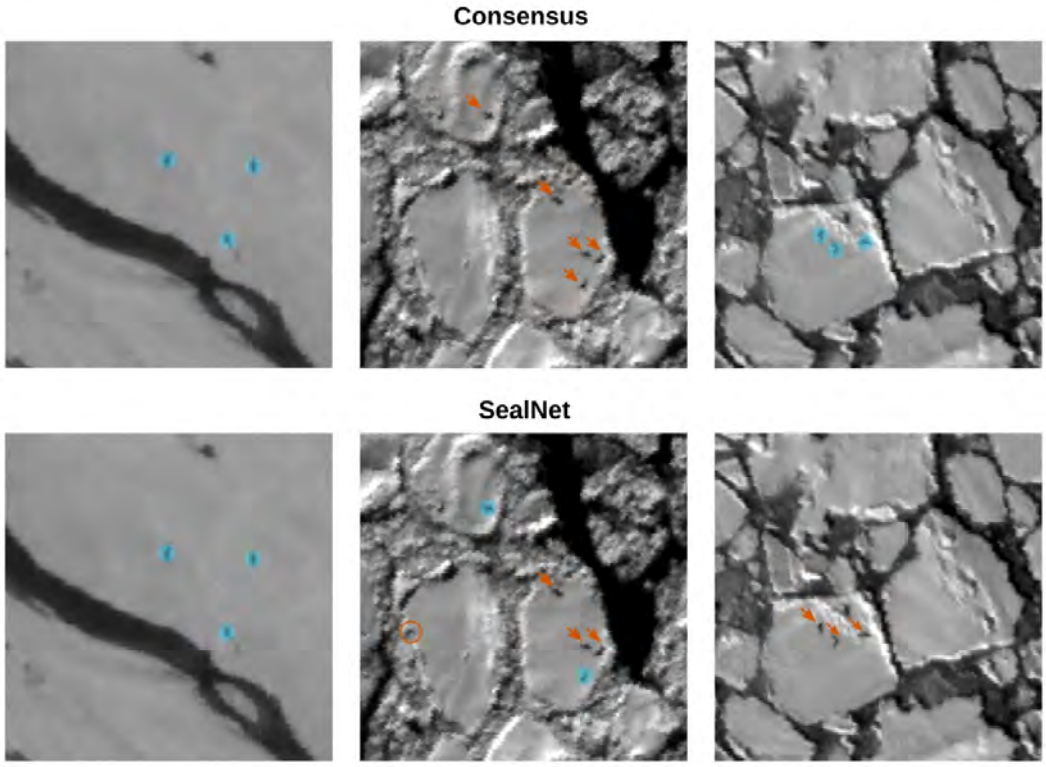

FIGURE 2.7: Sample SealNet output. Panels true-positives (light-blue circles) false negatives (orange arrows) and false positives (orange open circles), by a double-observer consensus (upper panel) and SealNet (lower panel). Examples, from left to right, show a case where both SealNet and the consensus set locate seals faultlessly, a case where SealNet outperforms the consensus set, and a case where it underperforms the consensus set. Crops were extracted at a 1:500 scale from the test scenes b, c, and d, respectively (Fig. 2.5). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
Satellite imagery copyright DigitalGlobe, Inc. (2019).

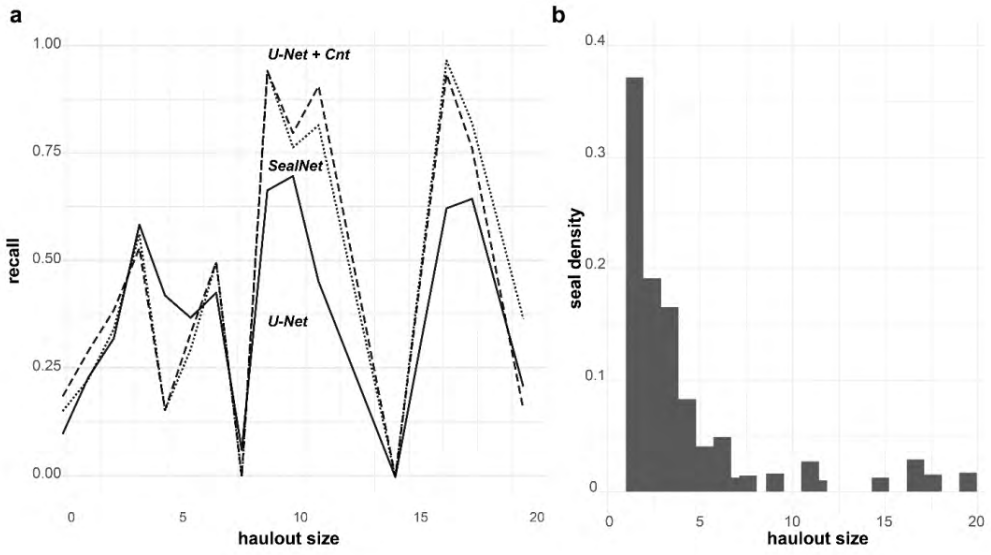

FIGURE 2.8: (a) CNN performance on different group sizes. Recall values are extracted by measuring the proportion of ground-truth seal points on specific-sized haul outs a model can recover. (b) Test group size distribution. The y-axis shows the proportion of seals across all test scenes that are located in groups of size x.

Chapter 3

Fine-Scale Sea Ice Segmentation for High-Resolution Satellite Imagery with Weakly-Supervised CNNs

3.1 Abstract

Fine-scale sea ice conditions are key to our efforts to understand and model climate change. I propose the first deep learning pipeline to extract fine-scale sea ice layers from high-resolution satellite imagery (Worldview-3). Extracting sea ice from imagery is often challenging due to the potentially complex texture of older ice floes (i.e. floating chunks of sea ice) and surrounding slush ice, making ice floes less distinctive from the surrounding water. I propose a pipeline using a U-Net variant with a Resnet encoder to retrieve ice floe pixel masks from VHR multispectral satellite imagery. Even with a modest-sized hand-labeled training set and the most basic hyperparameter choices, my CNN-based approach attains an out-of-sample f1-score of 0.698 – a nearly 60% improvement when compared to a watershed segmentation baseline. I then supplement my training set with a much larger sample of images weak-labeled by a watershed segmentation algorithm. To ensure watershed-derived pack-ice masks were a good representation of the underlying images, I created a synthetic version

for each weak-labeled image, where areas outside the mask are replaced by open-water scenery. Adding my synthetic image dataset, obtained at minimal effort when compared with hand-labeling, further improves the out-of-sample f1-score to 0.734. Finally, I use an ensemble of four test metrics and evaluated after mosaicing outputs from entire scenes. This mimics a production setting during model selection, reaching an out-of-sample f1-score of 0.753. My fully-automated pipeline is capable of detecting, monitoring, and segmenting ice floes at a very fine level of detail, and provides a roadmap for other use cases where partial results can be obtained with threshold-based methods but a context-robust segmentation pipeline is desired.

3.2 Introduction

Antarctic sea ice is an exceptionally dynamic habitat that plays an important role in climate feedback cycles [54, 144] and controls either directly or indirectly the Southern Ocean food web [5, 39, 135]. While coarse-grained maps of Antarctic sea ice have been available for several decades [30, 109], and have been critical to safe navigation [119, 95], climate modelling [64] and our understanding of sea ice-dependent predators [89], current sea ice products are primarily derived from passive microwave sensors operating at 25 km resolution and are therefore too coarse to resolve individual floes. Moreover, marine predators such as penguins and seals interact with sea ice on an extremely localized basis, and the characteristics of sea ice that might influence decisions about movement, foraging, or reproduction occur at scales far smaller than the resolution of typical sea ice imagery products [23, 9, 71]. Sub-meter resolution satellite imagery is now widely available for the Antarctic and this provides an opportunity to start mapping sea ice conditions over large spatial scales. The development and availability of fine-scale sea ice data products will radically expand our capacity to create high-resolution sea ice charts for navigation, link observed fine-scale sea ice conditions to climate models, and understand the detailed habitat requirements of

sea ice-dependent predators. Mapping fine-scale sea ice conditions at scale, especially within the highly-heterogeneous pack-ice zone, will require automated pipelines for sea ice segmentation.

Sea ice extraction is an active field in remote sensing. Typically, sea ice layers are extracted from Synthetic Aperture Radar (SAR) and optical sensors of low to medium resolution (e.g., MODIS, Sentinel-2, Landsat). Traditionally, sea ice was identified using pixel-based methods that used only the information contained in the spectral profile of each individual pixel to extract sea ice masks [61]. Other approaches explored the contrast between sea ice and the surrounding water bodies and threshold (e.g. watershed segmentation) or clustering-based methods (e.g. k-means clustering) to extract sea ice polygons [154, 155, 65], without the need of labeled datasets. More recently, machine learning models were trained to identify and predict different sea ice types from predetermined sets of sea ice polygons and expert-annotated class labels (e.g. [145, 18, 57]). Approaches using low to medium-resolution sensors bring the advantages of larger spatial and temporal coverage and, in the case of passive microwave and other non-optical sensors, the capability to extract useful information regardless of cloud cover and other factors that affect lighting. Although such methods have provided invaluable information on traits such as average sea ice cover, they are unfit to extract individual ice floes or fine-grained information on sea ice conditions. In imagery from very-high-resolution sensors such as Worldview-3, individual ice floes are several pixels large, and the classification and delineation of such super-pixel features are highly challenging for pixel-based solutions. Moreover, the extra detail adds a larger breadth of features that can hinder the performance of threshold-based methods. Fortunately, modern computer vision approaches exploiting deep learning (DL) are well suited to exactly such problems.

The rise of GPU-accelerated DL, marked by the first Imagenet challenge won by a Convolutional Neural Network (CNN) [116], has made DL affordable, brought the

field back as a hot research topic, and ultimately lead several ground-breaking improvements to the fields of CV and natural language processing (NLP). With the concomitant popularization of high-resolution sensors, DL solutions have largely replaced methods such as Support Vector Machines (SVM) and have already become a staple in some areas of remote sensing [86]. In contrast to other works that use DL for classifying sea ice at medium resolution (e.g. [18, 57]) and segment out sea ice in ship-borne images [38, 106], the goal of the present work is extracting precise ice floe masks from high-resolution imagery. More specifically, I am targeting ice floes only – a daunting task given a large number of potentially confounding fine-scale structures (e.g. slush, melt ponds, etc.) that emerge at very-high spatial resolutions. I do so by training a weakly-supervised CNN that learns from a small set of hand-labeled sea ice masks and a much larger set of weak annotations obtained with minimal effort using a watershed segmentation model. A fully automated pack-ice extraction tool would provide invaluable data for Antarctic ecology given the large number of ecosystem interactions mediated by sea ice.

3.3 Materials and Methods

3.3.1 Imagery and data annotation

Our datasets were extracted from a set of 43 multispectral Worldview-3 (WV-3) scenes (Table 3.1 and Fig. 3.1) covering 730.05 km^2 of coastal Antarctic scenery with an on-nadir resolution of 1.24m/pixel. I include three distinct types of annotation (Table 3.2): 1) pixel-level sea ice masks drawn by hand – my "hand-labeled" training set; 2) pixel-level sea ice masks extracted with watershed segmentation – my "watershed" training set; and 3) pixel level sea ice masks extracted with watershed segmentation and adapted to synthetic sea ice images – my "synthetic" training set. This multi-dataset

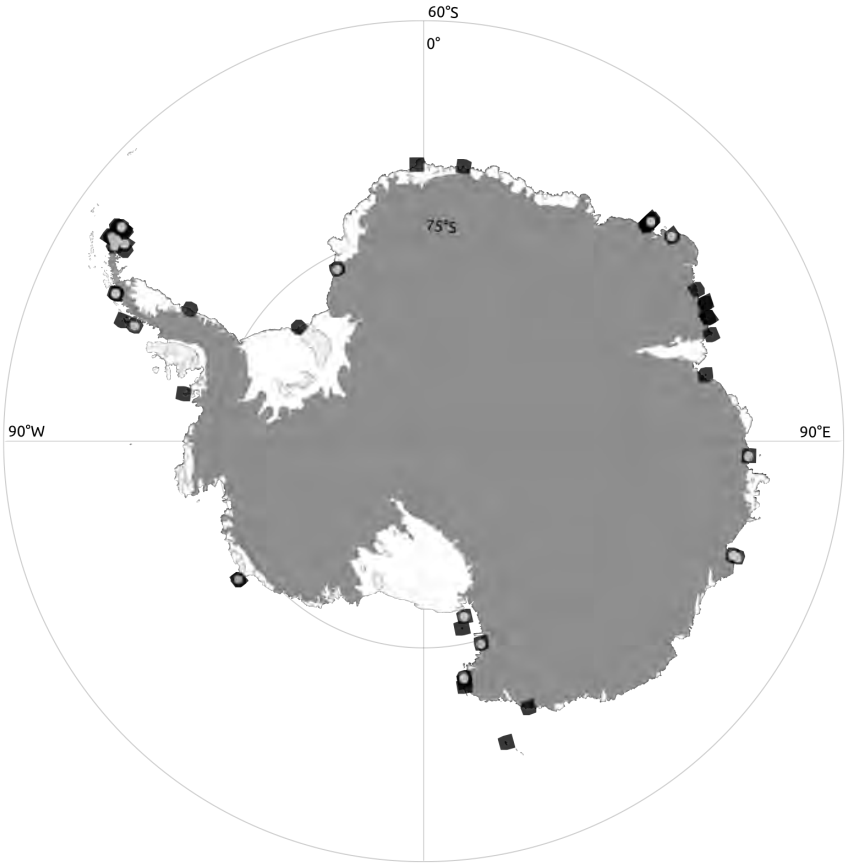

FIGURE 3.1: Training set scenes. Dark squares denote the location of each of the 36 scenes in my training set. Scene squares are marked with a light dot whenever I drew annotations by hand for that specific scene. Imagery copyright DigitalGlobe, Inc. 2021.

design allows us to take advantage of weak labels from watershed segmentation (watershed and synthetic training sets) during training but still get validation and test metrics on a set of careful manual annotations. Each scene consisted of the red, green, and blue bands of the WV-3 multispectral image tiled into 784×784 pixel patches with a 50% overlap between neighboring patches. I chose to extract patches that are bigger than my input size to generate a larger breadth of training images by leveraging random crops within my data-augmentation pipeline (described in the following section). Details on each method are supplied in the following sections.

TABLE 3.1: WorldView-3 imagery. I used a set of 43 multispectral WV-3 images to train, validate and test my ice floe segmentation models. To reduce GPU memory footprint during training and avoid further modifications to my CNN architectures, all imagery was converted from the native 8-band multispectral channels to three-channel images by extracting the red, green, and blue bands. Due to lighting limitations inherent to the poles and to capture the reproductive seasons of Antarctic megafauna, my imagery was acquired in a period ranging from November 20 to April 7 (summer - early spring) in the years of 2014, 2015, 2016, and 2017. All the imagery used in the study is cloud-free. Repeated consecutive catalog IDs indicate different scenes within the same strip.

Catalog ID	Lat-Lon	Cloud cover	Total area	Date
1040010005B62F00	-69.3327 158.4884	0.0	263.1 km^2	20/Nov/2014
1040010013346700	-76.9427 166.8715	0.0	212.6 km^2	26/Nov/2015
10400100156E6500	-63.1618 -54.9593	0.0	268.8 km^2	01/Jan/2016
10400100156E6500	-63.8006 -54.959	0.0	202.7 km^2	01/Jan/2016
10400100156E6500	-63.2718 -54.959	0.0	265.3 km^2	01/Jan/2016
10400100156E6500	-63.599 -54.9589	0.0	259.3 km^2	01/Jan/2016
1040010016234E00	-67.256 45.9485	0.0	266.5 km^2	02/Jan/2016
1040010016234E00	-67.668 45.9477	0.0	172.8 km^2	02/Jan/2016
1040010016234E00	-67.0437 45.9485	0.0	244.6 km^2	02/Jan/2016
1040010016234E00	-67.1471 45.9486	0.0	265.0 km^2	02/Jan/2016
1040010016234E00	-67.3652 45.9489	0.0	268.2 km^2	02/Jan/2016
1040010016234E00	-67.4748 45.9489	0.0	269.9 km^2	02/Jan/2016
1040010017265B00	-76.0 -26.6717	0.0	224.5 km^2	07/Jan/2016
1040010017A12200	-67.4771 164.6313	0.0	168.7 km^2	12/Jan/2016
10400100167EC800	-63.4564 -56.8695	0.0	282.7 km^2	17/Jan/2016
10400100167EC800	-63.3475 -56.8686	0.0	281.0 km^2	17/Jan/2016
10400100167EC800	-63.6757 -56.8695	0.0	287.3 km^2	17/Jan/2016
10400100167EC800	-63.2385 -56.8685	0.0	279.2 km^2	17/Jan/2016
10400100178F7100	-63.4235 -54.669	0.0	186.1 km^2	21/Jan/2016
104001001762AC00	-66.2365 110.1896	0.0	191.1 km^2	21/Jan/2016
10400100175A5600	-66.6168 -68.2485	0.0	122.0 km^2	25/Jan/2016
10400100175A5600	-67.575 -68.25	0.0	269.3 km^2	25/Jan/2016
104001001747E000	-64.2565 -56.6693	0.0	291.3 km^2	26/Jan/2016
104001001777C600	-69.0697 76.7836	0.0	220.4 km^2	28/Jan/2016
1040010018447F00	-67.6175 66.5771	0.0	296.5 km^2	28/Jan/2016
104001001844A900	-66.5325 92.5386	0.0	208.0 km^2	28/Jan/2016
1040010017764300	-74.7749 164.0267	0.0	225.3 km^2	29/Jan/2016
1040010017823400	-72.3657 170.2705	0.0	207.9 km^2	04/Feb/2016
1040010018694800	-72.0 170.5882	0.0	170.7 km^2	04/Feb/2016
10400100196BE200	-65.4111 -64.3911	0.0	274.8 km^2	25/Feb/2016
10400100196BE200	-65.4984 -64.3908	0.0	191.9 km^2	25/Feb/2016
10400100181F9B00	-66.8013 50.5412	0.0	215.6 km^2	27/Feb/2016
1040010018755100	-67.4705 61.0185	0.0	221.4 km^2	05/Mar/2016
1040010018046800	-65.938 110.2305	0.0	207.7 km^2	07/Mar/2016
1040010019529D00	-77.7016 -47.6769	0.0	183.9 km^2	13/Mar/2016
1040010019417700	-76.1377 168.3823	0.0	243.9 km^2	15/Mar/2016
104001001A625A00	-70.0097 -1.4187	0.0	163.3 km^2	16/Mar/2016
104001001A8FF900	-67.3803 63.9762	0.0	237.1 km^2	16/Mar/2016
104001001A27CC00	-64.5113 -57.4442	0.0	264.6 km^2	23/Mar/2016
104001001B448400	-69.9403 8.3095	0.0	163.1 km^2	25/Mar/2016
104001001A896700	-67.8698 69.7022	0.0	181.1 km^2	30/Mar/2016
104001001A6C8C00	-70.5887 -60.5685	0.0	234.1 km^2	07/Apr/2016
1040010028CD9C00	-73.2326 -126.7786	0.0	162.3 km^2	25/Jan/2017

TABLE 3.2: Training datasets. Number of scenes and total area covered by positive (i.e. patches with pack-ice) and negative (i.e. patches without pack-ice) patches within each of my datasets. Image annotations consisted of binary pixel masks that denote whether a pixel in a patch represents pack-ice. Hand-labeled masks were drawn over 3000×3000 meter crops at strategic locations whereas watershed derived masks were extracted by running a sliding window over scene regions marked by irregular polygons. Patches with watershed derived masks are used exclusively during training, whereas patches with hand-labeled masks are split equally between training, validation and test sets. Negative training patches were shared across all three training sets. To avoid inflation in my validation metric scores, I set aside a distinct set of negative images for validation.

Training set	Scenes	Area +	Area -
Hand-labeled[train]	19	20.8 km^2	240.9 km^2
Hand-labeled[valid]	18	20.2 km^2	17.85 $km2$
Hand-labeled[test]	19	20.4 km^2	16.8 $km2$
Watershed[train]	27	393.1 km^2	240.9 $km2$
Synthetic[train]	27	393.1 km^2	240.9 km^2

Hand-labeled training set

I employed hand-labeled pixel masks as my main tool to provide out-of-sample performance measurements to segmentation CNNs. My hand-labeled masks were created using the following steps: 1) extracting three RGB 3000×3000 pixel crops containing pack-ice at random, but with no overlap, from five different scenes; 2) opening the crops in Adobe Photoshop™ and creating a separate channel to store my sea ice mask; 3) using the magic wand and color selection tools to remove darker regions containing open water from my sea ice mask; and 4) manually filling holes created by darker areas inside floes. All my manual annotations were performed by a single individual and included multiple passes over the dataset to guarantee that annotations were as consistent as possible across different scenes. My crops and pixel masks were tiled using a sliding window approach with a patch size of 784×784 and 50% overlap between neighboring patches. I further supplemented this dataset by adding hard-negative patches (i.e. without sea ice) at the same proportion as the following two datasets.

The final hand-labeled dataset is drawn from 45 hand-labeled RGB crops split equally between training, validation, and test sets.

Watershed training set

I used a watershed segmentation algorithm as an inexpensive strategy to generate a large number of weak ground-truth masks from raw imagery with sea ice, as follows: 1) create georeferenced annotation masks by hand-drawing contour polygons over areas with pack-ice; 2) mask raw imagery and run a sliding window with a patch size of 784 and 50% overlap between neighboring patches to extract input patches; 3) create an annotation mask for each patch by running watershed segmentation sequentially; 4) draw contours for objects in the watershed mask and remove objects that are deemed too small to be a floe from the watershed mask (total area < $15m^2$; and 5) discard images where more than 15% of the total area has missing data or the watershed mask has a single contour (usually a contrast aberration around corners). I added extra patches in an equivalent manner using georeferenced polygons drawn in representative areas outside of pack ice to serve as hard negatives. The final Watershed training set contains a total of 6597 patches divided into 4085 pack-ice images and 2512 hard-negative patches.

Synthetic image training set

I built upon the previous dataset by creating synthetic versions of the imagery where the input image better matches its watershed mask as follows (Fig. 3.2): 1) taking a patch with pack-ice; 2) applying recursive watershed segmentation to the patch; 3) using the output of watershed segmentation to mask out all portions of the patch that did not contain sea ice; 4) pasting the resulting patch on top of an open water background patch to create a realistic synthetic image; and 5) removing the areas of greatest overlap between masked RGB channels from the segmentation mask to further individualize

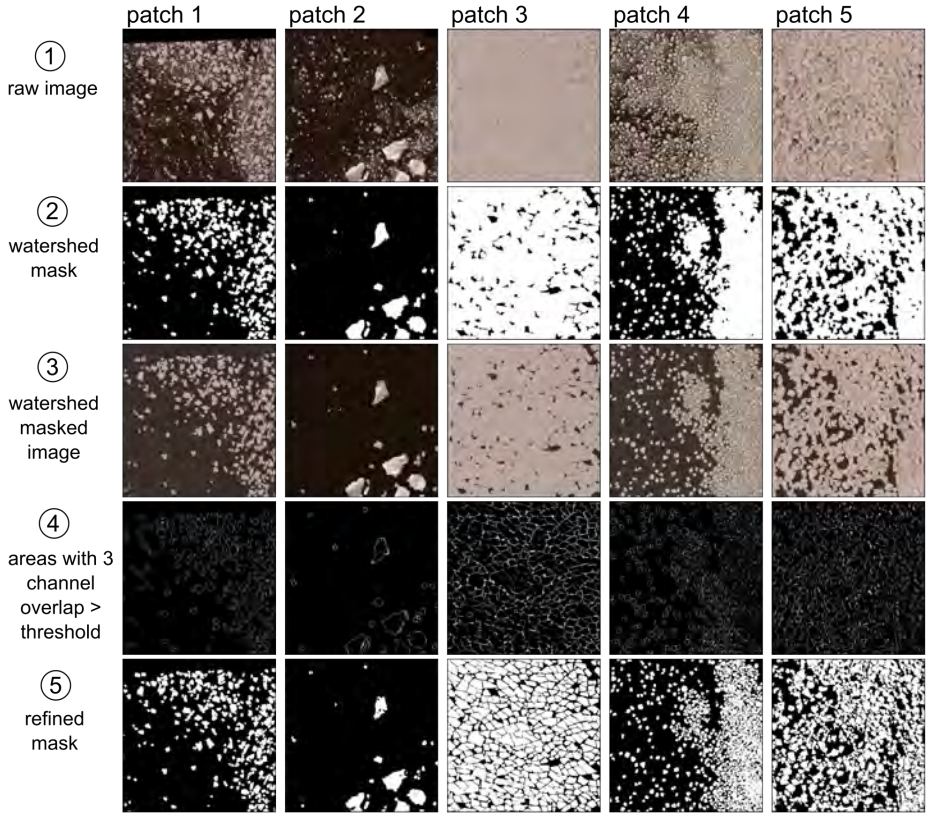

FIGURE 3.2: Synthetic image creation. Five examples of my synthetic image creation pipeline extracted from my training set images. My watershed segmentation algorithm in step 2 is applied sequentially for a total of three times. I fill masked-out areas in step 3 with open water images sampled at random. I find three channel overlaps in step 4 using an adaptive threshold. Refined masks in step 5 are obtained by subtracting overlapping areas from step 4 from watershed masks in step 2. Imagery copyright DigitalGlobe, Inc. 2021.

floes in the mask. The final Synthetic dataset has the same number of patches as the original Watershed dataset.

3.3.2 Segmentation CNNs

Our semantic segmentation experiments use a U-Net architecture variant [115], with the U-Net encoder branch replaced by a ResNet34 encoder [59](Fig. 3.3). I make this small modification in the U-Net encoder branch because a ResNet34 encoder generally outperforms the original encoder in terms of evaluation metrics with standard

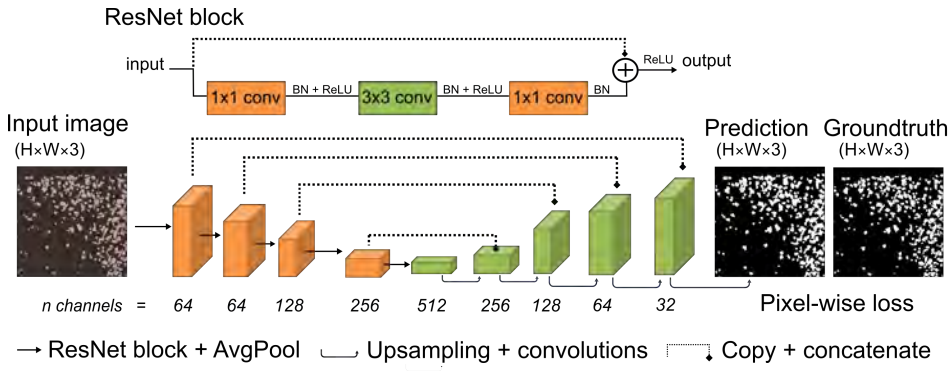

FIGURE 3.3: CNN architecture. My CNN architecture borrows from the U-Net architecture, with encoder and decoder branches connected by copy-and-concatenate operations, with the sole difference that the base U-Net encoder is replaced with a ResNet34 encoder. ResNet blocks within the encoder consist of a set of convolution operators intertwined by batch normalization and rectified linear unit (ReLU) operations followed by concatenation with the input features (i.e. skip-connection). After running through ResNet blocks, features get down-sampled after each ResNet block with a strided average pooling layer, reducing the height and width of each channel by a factor of 2. I do not provide numbers for height and width for input images and CNN blocks in the schematic because input size is a dynamic parameter in my study design.

datasets (e.g. [116]) and also to facilitate experiments with fine-tuning from a ResNet classifier. I trained my CNN to create pixel-level binary masks that represent areas of a patch covered by ice floes. I used the Dice coefficient [34] as my validation metric for model selection. I kept the best-performing model for each training set for comparison against the hand-labeled test dataset to get an out-of-sample measurement of model performance, boosted by test-time-augmentation [122]. Finally, I took the best-performing model according to test f1-score and retrained it on all samples from the synthetic and hand-labeled dataset to be used in production.

3.3.3 CNN training and validation

All my CNN training experiments were run on PyTorch v1.8.0 in Python [107], with an Adam optimizer [68] and a schedule where the learning rate is reduced by a factor of 10 whenever validation f1-score fails to improve after three consecutive epochs and training is interrupted after six epochs without improving validation F1. I searched

for optimal combinations hyperparameters running over 1000 random search experiments for input size (256, 384 and 512), loss function (see below), data augmentation (simple vs. complex, see below), and learning rate (log scale, from 1E-3 to 1E-5), using the greatest batch size allowed by GPU memory (200, 120 and 60 for input sizes 256, 384 and 512, respectively) and validation f1-score as the model selection metric. My training images are sampled with replacement to match a predefined ratio of negative to positive images in training batches, which was also explored as a hyperparameter. To explore the benefits of fine-tuning from a pre-trained model [151], I repeated my hyperparameter search experiments initializing model parameter weights to either the weights from one of my best-performing models according to validation metrics (picked at random from the top 100 models) or a CNN trained on binary classification for the presence of pack-ice in patches. My experiments were run at the Bridges-2 NSF supercomputer on GPU nodes with 8 Nvidia V-100 GPUs, each with 32GB of GPU memory. Model weights for my best-performing segmentation models are available in my GitHub code repository. I grouped my random search experiments with two options (i.e. data augmentation policy, fine-tuning, and test-time-augmentation), within 12 brackets defined by combinations of input size and training set and treated them as independent replicates for statistical analyses. More specifically, I extracted the best-performing model within each of my 12 brackets and tested whether the observed ratio of best-performing models given parameter values falls within the expectations of a binomial experiment with 12 trials and 50% probability of success.

3.3.4 Testing

I tested my CNNs using a routine that mimics the functioning of my models in production consisting of four steps: 1) use a sliding window approach to tile the input image into patches with a 50% overlap between neighboring patches, where the size of each patch matches the required input size of the CNN (i.e. 256, 384 or 512); 2)

generate predictions for each patch by applying a sigmoid transformation and binary threshold to the model output; and 3) create a mosaic of the predicted tiles and calculate metric scores by comparing predictions with ground-truth masks. Each of my CNN models from training was tested with and without test-time-augmentation, using a temperature-sharpen policy [13] to merge augmented predictions. To get robust out-of-sample performance estimates, my model selection used a consensus of four different metrics: 1) mean f1-score averaged across test scenes; 2) mean IoU averaged across test scenes; 3) f1-score across all pixels in the test set; and 4) average between the accuracy on background and foreground pixels in the test set. For each candidate model, I ran this pipeline over a set of 12 carefully labeled 3000×3000 meter areas with pack-ice and seven 3000×3000 meter areas of Antarctic scenery without pack-ice.

3.3.5 Loss functions

I experimented with a variety of loss functions that focus on different aspects of the segmentation output, largely borrowing from a recent comprehensive survey on loss functions for semantic segmentation CNNs [66]. Since the choice of loss function can have dramatic, non-obvious impacts on model performance [66], I chose to start with a broad set of candidate loss functions and use validation f1-scores during the hyperparameter search to find the ideal candidate for my use case. I initially used two pixel-based approaches, namely binary cross-entropy and Focal Loss [80]. While the former represents the simplest available loss function and is ideal for a baseline, the latter is often used for imbalanced datasets, as it puts more weight on pixels that are harder to classify. I then tested a number of region-based approaches that build upon the Dice coefficient [34] as they tend to preserve the shape of superpixel structures better than pixel-based solutions. In the context of semantic segmentation models, the Dice coefficient, a harmonic mean of precision and recall, is turned into a loss function by subtracting the Dice coefficient for a patch from 1, so that models can improve by

minimizing it through gradient descent optimization. Besides the original Dice Loss, I used three variants: 1) Log-Cosh Dice Loss [66], an attempt to improve the original Dice Loss by smoothing out its loss function; 2) Dice Perimeter loss [40], a variation of Dice Loss that uses the difference in the total perimeter of the predicted and ground truth masks as a regularization factor to the loss function; and 3) a weighted mixture of Dice and Focal Loss. Whenever available, I used native PyTorch implementations of my loss functions.

3.3.6 Data augmentation

To add more breadth to my training sets, and consequently make my models more robust to changes in scale, rotation, illumination, and position, I employed data augmentation pipelines tailored for satellite imagery, taking full advantage of rotations and random crops that would otherwise be unsuitable for non-aerial images. I use two data augmentation strategies: 1) a simple approach with random crops, vertical and horizontal flips, random shifts in position, random re-scaling, random 90-degree rotations (i.e., 90, 180, or 270 degrees), and brightness and contrast shifts; and 2) a more complex approach using the same transforms listed above plus noise reduction, RGB shifts, and random distortion effects. My data augmentation pipelines are applied continuously during training and use transform implementations from the Albumentations package [24]. The exact specifications for each can be found in my GitHub code repository.

3.3.7 Model baselines

I evaluated my sea ice extraction models using four baselines of increasing complexity: 1) watershed: extract directly with watershed segmentation (identical implementation from my watershed training set extraction); 2) basic U-Net: use the best-performing U-Net constrained to the simplest settings in my hyperparameter search (hand-labeled

training set, binary cross-entropy loss, simple augmentation pipeline, no model fine-tuning, no test-time augmentation, validation f1-score as model selection); 3) U-Net best validation: the best-performing U-Net from the hyperparameter search according to validation f1-scores; and 4) U-Net best test: the best-performing U-Net according to an ensemble of four different metrics measured after mosaicing model output. I obtained an out-of-sample performance estimate for each baseline as described in section 2.4.

3.4 Results

3.4.1 Model performance

Our first baseline, applying watershed segmentation to input images, attains a 0.464 f1-score in the test set after output mosaicing (Table 3.3). The simplest possible CNN-based model improves performance by >50%, reaching a test f1-score of 0.698. Adding more complex features and the synthetic dataset to the hyperparameter search further improves out-sample performance by 5%, reaching a test f1-score of 0.734. My final model, obtained by adopting a more elaborate model selection approach that mimics production settings, provides another modest improvement in terms of test f1-score, reaching 0.753.

3.4.2 Hyperparameter search

Our hyperparameter search experiments (Fig. 3.5) unanimously favored the use of test-time-augmentation ($\rho = 0.00024$, best performance in 12 out of 12 brackets), and showed slight, non-significant support for the use of my simple data augmentation pipeline over the complex one ($\rho = 0.07299$, best performance in 9 out of 12 brackets) and training from scratch instead of fine-tuning ($\rho = 0.07299$, best performance in 9

TABLE 3.3: Model performance. I show the f1-scores on validation and test sets of the best-performing model iteration across brackets input size and dataset, as well as the number of random search experiments, runs within each bracket trained from randomly initialized parameter weights (i.e., from scratch) or fine-tuning from a previous model, respectively. Validation f1-scores are obtained by averaging out the f1-scores from individual patches in the validation set. Test f1-scores reported are averages across the f1-score for all 19 test scenes obtained after output patches were merged into a mosaic, more akin to production settings, with the standard error as a measurement of spread. Test f1-scores from the same watershed segmentation approach I used to extract weakly-labeled images are provided as a baseline for U-Net-based models. My watershed segmentation model is implemented in Python using the numpy and OpenCV libraries and my U-Net CNN is implemented in PyTorch by swapping the original U-Net down-sampling layer for a ResNet34 encoder.

Model	Patch size	Dataset	F1 (val)	F1 (test)	N
U-Net	256	hand	0.842	0.727 ± 0.132	34, 12
U-Net	256	hand + synthetic	0.824	0.713 ± 0.087	36, 16
U-Net	256	hand + watershed	0.855	0.628 ± 0.174	34, 12
U-Net	256	synthetic	0.732	0.739 ± 0.126	42, 17
Watershed	256	-	-	0.464 ± 0.139	-
U-Net	384	hand	0.736	0.747 ± 0.142	31, 16
U-Net	384	hand + synthetic	0.822	0.713 ± 0.162	41, 19
U-Net	384	hand + watershed	0.848	0.633 ± 0.180	33, 10
U-Net	384	synthetic	0.769	0.727 ± 0.135	46, 21
Watershed	384	-	-	0.460 ± 0.141	-
U-Net	512	hand	0.776	0.733 ± 0.158	40, 13
U-Net	512	hand + synthetic	0.850	0.753 ± 0.113	32, 14
U-Net	384	hand + watershed	0.839	0.696 ± 0.176	39, 14
U-Net	512	synthetic	0.830	0.734 ± 0.133	37, 14
Watershed	512	-	-	0.459 ± 0.136	-

out of 12 brackets). In general, the f1-score differences between different parameter choices were much smaller when fine-tuning from previous models.

3.4.3 Qualitative model output

Model predictions obtained with watershed segmentation produce several false-positive and false-negative errors in scenes with pack-ice and produce an abundance of false-positive errors in background scenes (Fig. 3.5). My most basic CNN model has a greater recall than the previous baseline, at the cost of a lower precision in the third pack-ice scene, and successfully discards some icebergs and rocks from the predicted mask. Though it incurs substantially fewer false-positive errors than the previous baseline in background scenes, it does generate artifacts around the edges. The best model according to validation metrics produces sharp prediction masks inside pack-ice scenes but largely fails to discard icebergs and rocks from the predicted mask. Though this baseline achieves a higher overall f1-score than the previous one, it largely fails to ignore background imagery, incurring substantial false-positive errors. My final model, picked by my enhanced model selection scheme, has a lower recall but higher precision in pack-ice scenes when compared to the previous baseline and consistently discards icebergs and rocks from the predicted mask. Unlike the other three baselines, my final model generates few to no false positives when predicting outside of pack-ice (7 out of 7 background scenes had fewer than 0.5% false positives).

3.5 Discussion

3.5.1 Model out-of-sample performance

Even with a modest-sized hand-labeled training set, my CNN-based method largely outperforms threshold-based methods, represented here by a sequential watershed segmentation algorithm, quantitatively (Table 3.3) and qualitatively (Fig. 3.5). Even

after running a comprehensive hyperparameter search (Fig. 3.4), experiments using direct outputs of watershed segmentation as weak-labels (i.e. training set = hand + watershed) underperformed those with hand-labeled data only (Table 3.3). This result is expected if I take into account situations where there is lighter and darker pack-ice within the same patch, in which case the watershed algorithm will only retrieve the lighter-colored floes (e.g. Fig. 3.2, patches 2 and 4), creating misleading annotation masks. My synthetic image approach (Fig. 3.2), however, adds valuable supervision to my semantic segmentation CNNs, improving test f1-score by a considerable margin (Table 3.3), but incurs more false-positive errors when predicting outside of pack-ice (Fig. 3.5). With evaluation metrics to further penalize poor performance in background scenes and provide a better representation of true out-of-sample performance, I improved my test f1-score even further (Table 3.3), reaching over 0.75 in my comprehensive hand-labeled test set. Besides generating better prediction masks for ice floes, CNN-based methods are particularly advantageous because they are able to understand context, and thus produce considerably fewer false positives than threshold-based methods in at least three scenarios: 1) outside of pack-ice (Fig. 3.5, panels d and e); 2) in coastal areas; and 3) when icebergs are abundant. However, we need to exercise caution given the limited breadth of my test set. Thus, it would be beneficial to run qualitative and quantitative tests on a more representative set of randomized out-of-sample input images to rule out accidental good results from overfitting to the validation set during hyperparameter search studies and model selection. Additionally, introducing stronger regularization practices like one of the many approaches suggested in a recent review by Santos and Papa [120]. As one of my main goals was to evaluate CNNs as sea ice extraction tools, I did no post-processing on the output. There are several post-processing steps developed to improve the output from threshold-based or clustering-based methods that could also be beneficial if applied to my CNN-based pipeline (e.g. [155, 154]), especially with obtaining better ice floe boundaries when floes are tied together [156].

3.5.2 Hyperparameter search

Given the vast room for design choices with model architecture, loss functions, data augmentation routines, and training schedules and recent breakthroughs in GPU-accelerated parallel computing, the hyperparameter search has become a key step in developing ML pipelines and an active research field (e.g. [79, 4]). To allow an adequate exploration of design choices in a feasible time frame, my hyperparameter search (Fig. 3.4) focused on experimenting with input size, data augmentation routines, choice of loss function, choice of training set, ratio of negative to positive samples on training batches, and whether to fine-tune from a previous model. Surprisingly, with a few exceptions such as the underperforming LogCosh loss function [66] and the success of my mixture of Focal Loss and Dice Loss, there were no significant effects from my design choices in terms of validation f1-score (Fig. 3.4, middle panel in the lower part of the figure). Some settings, in particular data augmentation, would merit more experimentation, both in terms of further exploring the transformations adopted in this study by experimenting with their hyperparameters and by experimenting with novel transformations (e.g. [27, 49]). Another promising direction would be testing larger input sizes, as there seems to be an increasing trend in the median validation f1-score as I increase input size (Fig. 3.4, left panel in the upper part of the figure). I did not pursue that, however, because that would drastically reduce the size of my training batches since increasing input size has a quadratic effect on GPU memory usage. One design aspect that I did not touch in the present work and is particularly of interest to DL-based remote sensing applications is taking full advantage of multispectral bands. Apart from having a similar effect to GPU memory utilization as adding larger input sizes, taking 8-band images as inputs to my CNN model would require a series of modifications to the CNN architecture, making it less preferable than other important design choices included in my hyperparameter search when taking into account developer time allocation and computing resource utilization.

3.5.3 Fine-tuning experiments

Fine-tuning a model from previous model weights [151] obtained from training with a large, general-purpose dataset like the Imagenet challenge dataset [116] has become a staple when training computer vision CNNs. Such approaches are grounded on the generality of low-level structures like edges and simple shapes across applications and often focus on re-training only the last few layers in the CNN [140] that incorporate high-level structures. Fine-tuning is especially useful when there is a scarcity of labeled data. Existing model weights, however, are largely based on natural images from a frontal angle, hindering their usability for aerial or satellite imagery-based computer vision solutions, where the camera is always at an approximately 90°angle and the scale at which objects are presented is more or less fixed. Alternatively, fine-tuning for semantic segmentation models can be achieved by using patch-level labels to train a classifier model and swapping the weights from the original model backbone by the classifier parameter weights. Another approach is to fine-tune from a model trained at a different input size, aiming to be more scale-invariant. I experimented with both approaches and failed to obtain any improvement when fine-tuning from a classification model while obtaining some sparse improvements when fine-tuning from previous semantic segmentation models (best-performing models in 3 out of 12 of my hyperparameter brackets used fine-tuning from previous models). Interestingly, the trend line for the effect of learning rate in validation f1-scores changes sign for fine-tuning experiments (Fig. 3.4), potentially meaning that high learning rates could be breaking low-level feature representations from loaded model parameter weights. Since I decreased my learning rate during training whenever validation performance reached a plateau, results on the latter could have arisen by allowing the model to get out of local minima, similar to a warm-restart learning rate scheduling policy [83].

3.5.4 Conclusion

Though sea ice models at course resolution following the plastic continuum approach [33] can generate sensible predictions of several key features (e.g. sea ice thickness, sea ice cover) and will remain useful for climate modeling [15], their assumptions do not hold at finer-scale [32]. The added granularity provided by my solution allows better treatment of important phenomena such as the formation of fractures and leads which can substantially alter the structure of sea ice as it allows more short-wavelength absorption by the ocean [47]. Additionally, since tasked high-resolution satellite imagery (e.g. WV-3) can be retrieved at specified locations within hours, my approach can enhance sea ice detection for shipping and logistics with a broader range of action than ship-based camera approaches (e.g. [106, 38]. Because of its reliability outside of pack-ice areas (e.g. Fig. 3.5), my pipeline is capable not only of producing sharp ice floe segmentation masks but detecting the presence of floes in VHR imagery. My fully automated, context-robust approach allows us to leverage modern GPUs to monitor fine-scale sea ice conditions at a continental level. Finally, my semantic-segmentation approach could be expanded to segment and classify different fine structures in the Antarctic and Arctic landscapes provided we have plenty of labeled images at a passable quality standard.

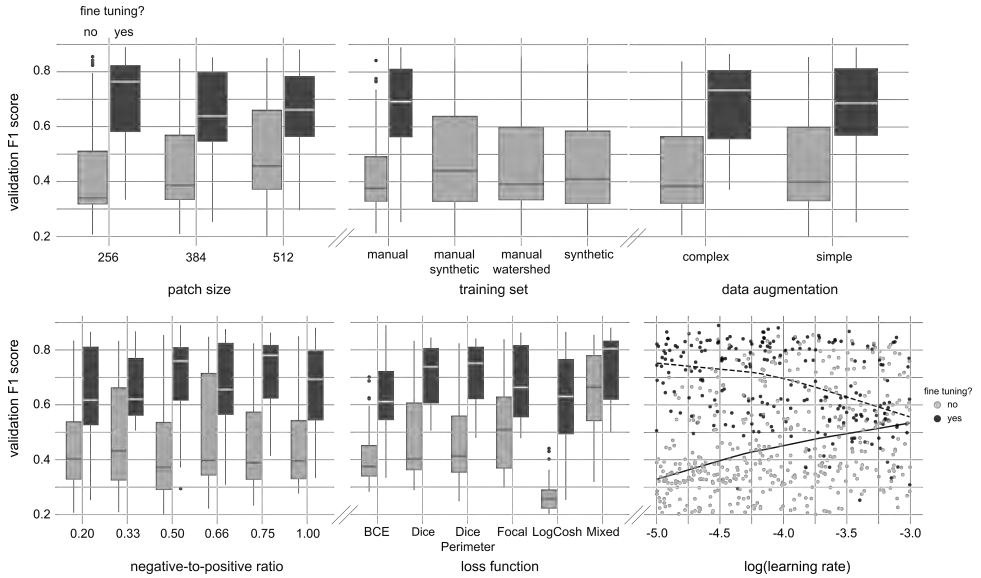

FIGURE 3.4: Hyperparameter tuning experiments. Validation f1-scores of 623 random search experiments across six different hyperparameters. I test the influence of input size, training set choice, data augmentation routine, ratio of negative to positive images within mini-batches, choice of loss function and learning rate on model performance, measured as the f1-score in the validation set. My training sets consist of combinations of a small set of hand-labeled images ("manual"), a larger set of images annotated using a watershed segmentation algorithm ("watershed"), and a set of synthetic input images created by modifying images from the previous set to be more consistent with their watershed-derived masks ("synthetic"). For loss functions, I tested binary cross-entropy loss (BCE), Focal Loss, three variants of Dice Loss, and a weighted mixture of Dice and Focal Losses. For each experiment, I split my runs between models trained from scratch and models that fine-tuned from a previous experiment, in which case initial parameter weights would be drawn from one of the top 100 models trained from scratch, selected at random. All my fine-tuning experiments were trained with manual labels, as the annotation masks within are closer to the output than I would wish during inference. The learning rate scatter plot shows each experiment as a dot and trend lines for models trained from scratch (continuous line) and fine-tuned models (dashed line).

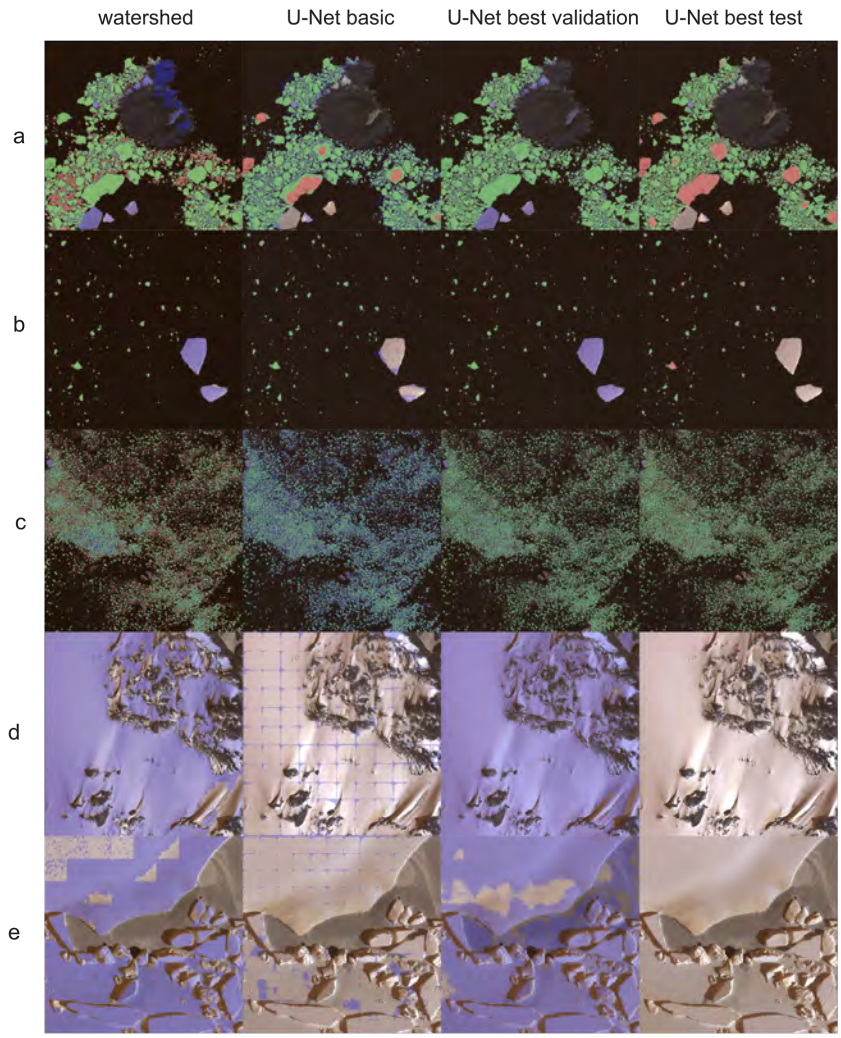

FIGURE 3.5: Output visualization. Model output at test scenes from four different sea ice extraction models left to right: watershed segmentation, a basic U-Net, the best U-Net according to validation metrics, and the best U-Net according to test metrics. Test scenes are 3000×3000 meter WV-3 multispectral scenes from the Antarctic coastline tiled with a 50% overlap at the input size required by each model. True positives, false positives, and false negatives are shown in transparent green, purple, and pink, respectively. My final model generates few if any false-positive errors in land and fast-ice imagery consistently avoids rock formations and icebergs, does not create artifacts at tiles edges, and still captures the majority of pack-ice within predicted masks. Imagery copyright DigitalGlobe, Inc. 2021.

Chapter 4

SealNet 2.0: Human level fully-automated pack-ice seal detection in very-high-resolution satellite imagery with CNN model ensembles

4.1 Abstract

Pack-ice seals are key indicator species in the Southern Ocean. Their large size (2–4 m) and continent-wide distribution make them ideal candidates for monitoring programs via very-high-resolution satellite imagery. The sheer volume of imagery required, however, hampers our ability to rely on manual annotation alone. Here, I present SealNet 2.0, a fully automated approach to seal detection that couples a sea ice segmentation model to find potential seal habitats with an ensemble of semantic segmentation convolutional neural network models for seal detection. My best ensemble attains 0.806 precision and 0.640 recall on an out-of-sample test dataset, surpassing two trained human observers. Built upon the original SealNet, it outperforms its predecessor by using annotation datasets focused on sea ice only, a comprehensive hyperparameter study leveraging substantial high-performance computing resources,

and post-processing through regression head outputs and segmentation head logits at predicted seal locations. Even with a simplified version of my ensemble model, using AI predictions as a guide dramatically boosted the precision and recall of two human experts, showing potential as a training device for novice seal annotators. Like human observers, the performance of my automated approach deteriorates with terrain ruggedness, highlighting the need for statistical treatment to draw global population estimates from AI output.

4.2 Introduction

Here, I present a human-level, fully automated solution to detect pack-ice seals in VHR imagery, built upon my previous proof-of-concept study [50]. My pipeline works by pre-processing Antarctic coastline VHR imagery through a sea ice segmentation model [51] that narrows down candidate imagery to scenes with relevant sea ice substrate. These selected scenes are scanned for seals using an ensemble of CNN models and ultimately converted to a database of geolocated predicted seals. Models in my pipeline were trained on a large annotated dataset obtained and curated over the course of five years. This extensive dataset and a massive deployment of GPU resources allowed us to train, validate, and test a wide range of different model solutions. Using annotations from an experienced observer as the basis for comparison, my automated approach outperforms two human observers with >100 h of experience when faced with novel VHR imagery randomly sampled from the existing imagery collection.

4.3 Materials and Methods

Our seal detection pipeline employs an ensemble of CNN models to derive georeferenced seal locations from very-high-resolution satellite imagery (Figure 4.1). Detailed information on my imagery datasets, CNN model training, and ensembling techniques can be found in the sections below.

4.3.1 Imagery and data annotation

For training, validation, model selection, model ensembling, and out-of-sample performance estimation, I employed three different annotated datasets (Figure 4.2): a training/validation set, an expert-selected test set, and a randomly selected test set. All three datasets comprised panchromatic Worldview-3 high-resolution satellite images (each one referred to as a 'scene') from the Antarctic coastline with an on-nadir resolution of 0.31 m/pixel. The annotation process consisted of manually browsing through Antarctic coastline imagery at a scale at which individual seals are detectable; at the centroid of each putative seal found, a geolocated point was added to a GIS (geographic information system) spatial point database (i.e., ESRI ® shapefile). I used a double-observer approach to create a consensus test dataset for out-of-sample performance estimation and model selection. The training/validation and test datasets used in this study were extracted from a set of 38 panchromatic scenes covering 8719.82 km^2. I selected training and test scenes that would represent a comprehensive range of environmental and sensor conditions, including images captured over a range of off-nadir angles, lighting conditions, and cloud covers. My training/validation and test sets represent a significant expansion and update of the datasets used in [50]. In addition to eliminating any putative seals that were re-classified on further consideration, I annotated several new scenes for training and validation. I also changed my

hard-negative sampling strategy from extracting crops at polygons that marked locations without seals to extracting random crops that did not overlap with any seal annotations in scenes for which I had seal annotations. This new hard-negative sampling approach reflects the introduction of a sea ice detection step prior to detecting seals, allowing us to focus on areas with sea ice conditions amenable to seals being hauled out and thus available for detection. I revised all hard-negative samples to remove potential false positives. To avoid a potential selection bias, I created an additional test set (heretofore, the 'random crops test set') composed of 300 non-overlapping 1 km^2 crops from 100 randomly selected WV03 scenes from the Antarctic coastline identified as having at least 5% sea ice cover by a sea ice segmentation CNN [51]. These crops were annotated by three different observers (BG, MW, HJL) with varying levels of experience classifying seals in WV03 imagery.

Training/validation set

I employed non-overlapping training and validation sets to train seal detection CNNs via gradient descent and run a comprehensive hyperparameter search optimized for out-of-sample performance. Using non-overlapping groups of seals from scenes in the training/validation set, I randomly assigned 80% of those groups of seal annotations to model training and 20% to model validation. Centered on each seal location, I extracted a 768×768 cropped image (i.e., patch) and saved a binary mask for that crop with '1' on pixels at seal centroids and '0' elsewhere. In addition, I recorded the total number of seals present in each patch. Since several seals may overlap in a single training patch, sampling training images at random would result in a bias towards seals situated within groups of seals (which are easier to detect [50]). To address that, I designed a weighted sampler that ensures that every seal is equally likely to be represented during training by down-weighting the probability of sampling individual seals based on the number of seals found within a radius of 50 m, making solitary seals and seals in larger groups just as likely to be represented during training. For

training models that require bounding box annotations (i.e., instance segmentation and object detection models), I generated 11 $\times$ 11 bounding boxes centered on each seal. For each scene in training/validation sets, I extracted 300 non-overlapping 'hard-negative' 768 $\times$ 768 patches by randomly drawing crops from regions without any seal annotation. To ensure that no false positives existed in negative patches, I manually reviewed each negative patch and excluded those potentially containing seals. The final training/validation set was composed of a total of 8735 patches with seal annotations and 7750 hard-negative patches.

Expert-selected test set

Our expert-selection test set builds on the approach from my previous study [50] by adding annotation revisions and replacing the original negative scenes with scenes that better reflect my new pipeline design with its sea-ice detection pre-processing step. I employed scene-wide annotations here to provide realistic out-of-sample metrics of model predictive performance in production (i.e., when predicting, a detection model has to go through entire scenes). This test set was used for both CNN model selection and model ensemble training. Moreover, I used predictions from the 10 top-performing models in the test scenes along with consensus annotations to train model ensembles that reclassify each point as a seal or a false positive based on the prediction from each model for that specific point (see below).

Random crops test set

I used a set of 300 non-overlapping 1 km^2 crops from 100 randomly selected WV03 scenes to validate models, optimize model ensembles, and remove selection bias from out-of-sample performance estimates. The 100 WV03 scenes were sampled at random from a set of 1948 scenes obtained by classifying a suite of 14,872 WV03 scenes through my sea ice segmentation model [51] and eliminating from consideration those

with <5% (predicted) sea ice cover. Because the vast majority of imagery contains neither seals nor features such as rocks and/or shadows that could be confounded with seals, I used my top-10 model ensemble along with a stratified sampling approach to select the 300 crops for the test set; specifically, I selected 100 crops where all 10 models predicted one or more seals (seals very likely present), 100 crops where none of the models predicted any seals (seals very likely absent), and 100 crops where there was disagreement among the models as to the presence of seals in the scene (seal presence ambiguous). To create a consensus dataset representing manual (human) annotation, three observers (BG, MW, and HJL) independently annotated these 300 crops. While all three observers had considerable experience annotating seals in WV03 imagery, there was a gradient in the amount of experience ($BG > MW > HJL$); so, the consensus annotation used to represent 'truth' (in the absence of true ground validation, which is impossible in this scenario) was constructed by having the most experienced observer (BG) review and edit (as appropriate) the union of all manual annotations and high-probability CNN predictions. To explore the benefits of AI-guided annotation, each observer had a 50% chance of having access to AI help in the form of ensemble model predictions with their associated probability. Finally, I used the random crops test set as a tool to calibrate and evaluate model ensembles on out-of-sample annotations.

4.3.2 CNN training and validation

CNN-model architecture choice focused on established rather than state-of-the-art methods to favor explainability, comparison with other studies, and ease of implementation instead of pure predictive performance. Seal detection CNNs tested in this study were designed for three different tasks: object detection (i.e., drawing bounding boxes around each object of interest and giving appropriate labels to each bounding box), instance-segmentation (i.e., object detection with segmentation masks inside

bounding boxes), and semantic segmentation (i.e., labeling every pixel in the image). For object detection and instance segmentation, I tested Fast R-CNN [48] and MaskR-CNN [60], respectively, as implemented in the native torchvision package [87]. In both cases, I modify the default anchor box sizes for predicted objects to a smaller size that better matches my 'truth' bounding boxes, and, in the case of MaskRCNN, I swapped the original binary cross-entropy (BCE) criterion from the segmentation loss for a region-based dice loss (see the section on loss functions below). For semantic segmentation models, I tested both the U-Net architecture [115], as implemented in the segmentation-models-pytorch package [149] but with an added regression head, and TransUnet, a transformer-based U-Net-like architecture, as implemented in the original paper [26], with the exception of an added regression head. To ensure a fair comparison between widely different CNN modalities, I used a unified validation metric across all models: f1-score between predicted and 'truth' seals in the validation set. Training epochs consisted of going through every training image exactly once, starting a validation round whenever 1/3 of the training images were processed, with a total of three validation rounds per training epoch. CNNs were trained with an AdamW optimizer [82] with a policy that reduces the learning rate by a factor of two whenever there is no improvement in terms of validation metrics for N consecutive validation rounds—where 'N' is a hyperparameter—and stops training whenever there is no improvement for 3 training epochs. Across all settings, I used 512×512 input images, either sampled at random from the training pool or processed sequentially from the validation pool, grouped into mini-batches with the greatest number of images allowed given constraints from the model architecture and GPU memory availability. All experiments were performed using pytorch [107] with mixed-precision training [94] on NVIDIA V100 GPUs with 32 GB of memory from the Bridges-2 supercomputer [22].

Data augmentation

To make my models more robust to scale, positioning, illumination, and other potential confounding factors, I employed a data augmentation pipeline during training, tailored to take full advantage of random crops and rotations given the nature of my imagery and dimensions of my objects of interest. I intentionally extracted larger patches (768×768) than my model input size (512×512) to keep training images diverse and non-obvious, and to reduce a potential bias for detecting seals at the center of input images. Whenever a seal was no longer present in the original patch after applying a random crop, the count for that patch (based on the 'truth' dataset) was adjusted to reflect that. For training, I used two data augmentation strategies: (1) a simple approach with random crops, vertical and horizontal flips, random shifts in position, random re-scaling, random 90-degree rotations (i.e., 90, 180, or 270 degrees), and brightness and contrast shifts; (2) a more complex approach using the transforms listed above plus noise reduction and intensity shifts. My training augmentations are integrated into the training loop using implementations from the Albumentations package [24]. As an additional step to make predictions more robust to orientation, I apply horizontal flips and 90, 180, and 270-degree rotations to each input image, and average out predicted masks and counts from all possible combinations.

Loss functions

To train regression heads on semantic segmentation models (i.e., U-Net and TransUnet), I used Huber loss [63], whereas segmentation heads used either Focal loss [80], Dice loss [34], or a combination of both. For instance segmentation and object detection models (i.e., Fast R-CNN and Mask R-CNN), I used the default torchvision losses for the Region Proposal Network classifier, the Bounding box classifier, and the Bounding box coordinates regression. I swapped the original Mask R-CNN BCE loss

for predicted masks with Dice loss since BCE is not suitable when few if any pixels fall into the positive class.

4.3.3 Hyperparameter search and model selection

I tested a wide range of scenarios to find optimal combinations of hyperparameters for seal detection models according to f1-score [34] for the expert-selected test set (i.e., test f1-score), running a total of 1056 full-length experiments. For all models, I tested the impact of learning rate, the number of epochs without improvement that would trigger learning rate reduction, and the ratio of negative to positive images in the data loader. For semantic segmentation models, I tested the impact of the segmentation loss function (Dice loss, Focal loss, or mixed Dice and Focal losses), backbone architecture (Resnet34 [59] and EfficentNet [134] variants), relative weight for regression and segmentation losses, and rate of dropout [132] applied to regression heads. When measuring test f1-score for semantic segmentation models, I tested the potential of using a threshold on predicted counts to remove false positives. To analyze the relative impact of each hyperparameter on test f1-score, I fit a CatBoost regressor model using hyperparameter values from each trial as dependent variables and f1-score as the outcome variable and calculated the relative importance of each hyperparameter using Shapley [123] scores. To save processing time, experiments that underperformed in terms of the maximum validation f1-score (>0.7 for instance segmentation and >0.5 for object detection and instance segmentation) were not carried into the testing stage. After an initial set of 372 experiments with semantic segmentation models, I narrowed down my hyperparameter pool to speed up convergence. For the latter 251 experiments, I also used a range of thresholds to explore the impact of using predicted count as a post-processing step to remove false positives, i.e., for each threshold, predicted points on patches where the predicted count was smaller than the threshold were discarded before comparison with ground truth annotations.

4.3.4 Model ensembling

Our examination of a large suite of models allowed us to deploy a model ensemble post-processing step. The first step to creating model ensembles was gathering the correspondent predicted counts and logits for predicted seal locations at the expert-selected test set and training/validation set using the 10 top-performing models in terms of test f1-score. Whenever an individual model did not predict a seal at a location where other model(s) did predict a seal, cells with model logits and counts for that location were left as missing values. Predicted counts and logits from each model were then used as dependent variables to predict whether each point was in fact a true seal according to the 'truth' annotations. I split my random crops test set between validation and testing to run a hyperparameter search for ensemble models, trained at binary classification for true-positive vs. false-positive seals, ranging from simpler linear models (i.e., logistic regression and ElasticNet [158]) to more intricate tree-based models (i.e., random forest, CatBoost [37], and XGBoost [28]). Though the training and validation sets are already captured by individual models, I added the potential usage of these annotations to train model ensembles as a hyperparameter. I ran 50 independent hyperparameter search studies with 1500 experiments each. An experiment in my hyperparameter search consisted of sampling a combination of hyperparameter values from posteriors, training an ensemble model using those hyperparameter values, and updating posteriors according to f1-score in the validation portion of the random crops test set. To measure the contribution of each individual model to ensemble predictions, I used relative feature importance for logits and predicted counts from each model, in the form of feature weights for linear models and Shapley scores for tree-based models. I used a Bayesian optimization routine [126], implemented in the optuna package [3], with multivariate normal priors for hyperparameters to find the best-performing ensemble model, using the f1-score of the validation half of the

random crops test set as my metric. The full range of hyperparameter choices for ensemble models can be found in the 's code repository.

4.3.5 Evaluation

During all training experiments, my validation metric is the instance-level (i.e., individual seal centroids) f1-score. I measured this metric directly for model ensemble predictions; however, for CNN outputs, I need to match each predicted seal with consensus annotations. Because different annotators may identify the seal centroid in slightly different locations, and given expected seal dimensions of roughly 2 m, I used a tolerance of 1.5 m to declare two seals a match. For semantic segmentation models, I did so by applying a sigmoid transform followed by a binary thresholding step, leaving us with seal mask polygons. I then extracted the centroid of each polygon and looked for a match with the centroids of the consensus dataset. For instance segmentation and object detection models, however, I simply extracted the centroid from each predicted bounding box for comparison with consensus centroids. I evaluated the out-of-sample performance of model ensemble predictions, the best individual performing models, the original SealNet model [50], and human observers against my random crops test set consensus annotations. Since model ensembles use the validation portion of the random crops test set for model selection, all models and model ensembles were evaluated on the test portion of the random crops dataset to ensure a fair comparison. For all evaluation steps, model predictions on land were masked out using a sea ice mask derived from the Antarctic Digital Database (ADD) high-resolution coastline polygons available on the Quantarctica project [90]. To evaluate the consistency of output model probabilities, I measured the correlation between the sum of logits around predicted seal centroids and their corresponding 'truth' label. Similarly, I measured the same correlation for ensemble models using model-derived logits and their corresponding 'truth' labels.

4.4 Results

Semantic segmentation models largely outperformed object detection and instance segmentation models (Figure 4.3) in terms of test f1-score (0.39 ± 0.08, 0.04 ± 0.02, and 0.04 ± 0.02, respectively), attaining a top test f1-score of 0.58. My initial set of experiments with semantic segmentation models (Figure 4.4, marked in orange) showed that test-time-augmentation is beneficial in terms of test f1-score, and that mid-range backbone architectures in terms of complexity (i.e., EfficientNet-b0, b1, and b2) had a slight edge over the extremes (i.e., ResNet34 and EfficientNet-b3); thus, I turned test-time-augmentation on by default and focused on mid-range backbone architectures for the later part of hyperparameter search experiments. First phase results also hinted that my ranges for the ratio of negative to positive images in training batches, learning rate, regression head weight, regression head dropout, and learning rate scheduler patience (Figure 4.4, marked in orange) could be adjusted to speed up convergence on better-performing models. Though there was a slight edge for simple augmentations over complex augmentations and random sampling over weighted group sampling, I opted to keep both options for later experiments. Even with fewer iterations, final experiments (Figure 4.4, marked in teal) largely outperformed initial ones in terms of test f1-score (0.35 ± 0.09 vs. 0.26 ± 0.13). Using a threshold on predicted counts as post-processing to remove false positives showed an average increase of 0.04 ± 0.05 in terms of test f1-score. Applying a threshold based on predicted counts as a post-processing step dramatically changed the distribution of test f1-scores (Figure 4.5); moreover, virtually all best-performing models had increases in f1-score by applying post-processing via the predicted count threshold.

The best-performing study for ensemble models achieved an f1-score of 0.69 in the validation portion of the random crops tests set (Table 4.1). The correlation between model logits and 'truth' labels was consistently higher for ensemble models when

compared with individual CNN models (Table 4.1). The vast majority of hyperparameter search studies (42 out of 50, binomial p-value: <0.001) converged on XGBoost tree ensembles as their model of choice, with varying internal settings. The single best-performing study and 2 out of the 10 best-performing studies, however, converged on CatBoost tree ensemble as their model of choice. All independent studies kept CNN model predictions from the validation set as training data and dropped those for the training set. Shapley values for feature importance on the best-performing models in independent studies (Figure 4.6) heavily favored features from the best-performing CNN in terms of f1-score in the random crops test set (CNN 3) and features regarding patch-level predicted counts, followed closely by model logits.

4.5 Discussion

Our best-fitting CNN ensembles (Figure 4.1) attain an f1-score of 0.71 on a randomly-sampled dataset, with double-observer coverage and no exposure during training or validation, outperforming two human observers with >100 h of experience and access to annotations from both a simpler ensemble (Table 4.1) and the previous SealNet CNN pipeline [50]. The improvement in predictive performance stems from three primary factors: (1) a larger and more carefully curated training dataset that focused on scenes with sea ice (Figure 4.2); (2) a comprehensive hyperparameter search study (Figures 4.2 and 4.4), only feasible in a multi-GPU setting; and (3) a new methodology using binary thresholding and regression counts followed by a model ensemble post-processing step. My SealNet 1.0 classifier used the regression output to dictate how many logit hotspots would be extracted from predicted segmentation masks [50]. Here, I use regression outputs as a post-processing step for segmentation masks, which removes many false positives and leads to an improved f1-score (Figure 4.5). This SealNet 2.0 approach is also preferable because it relies solely on pixel-level, centroid mask annotations for prediction, which hinge upon stronger supervision signals

TABLE 4.1: Out of sample performance for human observers (with and without the help of AI output), individual CNN models, and model ensembles measured at the random crops tests set. AI help is provided through the output of a simple ensemble model (i.e., an ElasticNet classifier, 'ensemble naive'), with a color gradient based on model certainty. Because whether an observer will have access to AI help is assigned independently at random, human observers had different sets of imagery processed with the aid of AI output. U-Nets 1–5 are ordered according to their ranking based on f1-score in the expert-selected test set. SealNet 1.0 predictions were obtained with the original SealNet. Similarly, ensemble models 1–5 are numbered in descending order of f1-score on the validation portion of the random crops test set. I include the correlation between model logits and 'truth' labels as a measurement of consistency.

Observer/Model	Precision	Recall	f1	AI help	Architecture	Logit correlation
HJL	0.35	0.56	0.43	No	-	-
HJL	0.58	0.69	0.63	Yes	-	-
MW	0.50	0.63	0.56	No	-	-
MW	0.55	0.69	0.61	Yes	-	-
CNN 1	0.60	0.63	0.62	-	UnetEfficientNet-b1	0.54
CNN 2	0.45	0.67	0.54	-	UnetEfficientNet-b1	0.33
CNN 3	0.71	0.67	0.69	-	UnetEfficientNet-b1	0.60
CNN 4	0.44	0.67	0.53	-	UnetEfficientNet-b1	0.36
CNN 5	0.68	0.53	0.60	-	UnetEfficientNet-b0	0.53
Sealnet 1.0	0.07	0.02	0.03	-	SealNet	0.07
ensemble 1	0.80	0.64	0.71	-	CatBoost	0.69
ensemble 2	0.74	0.67	0.70	-	XGBoost	0.67
ensemble 3	0.64	0.70	0.67	-	CatBoost	0.67
ensemble 4	0.73	0.67	0.70	-	XGBoost	0.68
ensemble 5	0.73	0.66	0.70	-	XGBoost	0.67
ensemble naive	0.59	0.69	0.64	-	ElasticNet	0.60

during training when compared with patch-level 'true' counts.

Surprisingly, though the problem at hand theoretically aligns better with instance segmentation/object detection frameworks, my experiments with MaskRCNN [60] and Fast R-CNN [48] showed lackluster results (Figure 4.3) when compared with U-Nets [115]—a considerably simpler semantic segmentation approach. The extremely poor precision scores obtained with these methods could derive from limitations for training without foreground objects, creating a bias for over-predicting seals. With my U-Net-based approach, I am not only able to train using background-only patches,

but I can also find optimal ratios of patches with and without foreground objects to maximize the balance between precision and recall through a hyperparameter search (Figure 4.4). This capability could give an edge to U-Net-based and other semantic segmentation approaches in 'needle-in-a-haystack' problems, which are ubiquitous in object detection applications for remote sensing imagery (e.g., [148, 17, 67]). The importance of showing negative examples during training in this kind of setting is supported by the relatively high negative-to-positive ratio found in my best-performing models (Figure 4.4, panel *d*).

Our top-10 individual CNN models, surprisingly, have a slightly lower out-of-sample recall than the global average for phase 2 experiments (0.54 vs. 0.55); however, on average, they are able to attain dramatically higher precision (0.52 vs. 0.33). This emphasis on avoiding false positives is also present when we look at the extremely high correlation between out-of-sample precision and f1-score ($r = 0.93$) and the strong negative correlation between out-of-sample recall and f1-score ($r = -0.43$). The relatively low correlation between f1-scores on the expert-selected test set performance and the random crops test set ($r = 0.49$)—and the dramatic performance decrease from SealNet 1.0 [50] on my more diverse test set Table 4.1—illustrates the importance of designing comprehensive test suites and cautions against over-relying on performance estimates on limited test sets. My random crops tests set was specifically designed to minimize the risk of over-relying on validation/model selection metrics for out-of-sample performance estimates — individual models that overfit to smaller datasets such as my validation set or expert-selected test set were very unlikely to attain good metrics by chance on a diverse, randomly-sampled, set of test images.

Having access to AI help in the form of output from a simple ensemble model leads to a substantial improvement in the f1-score of human observers, improving precision without sacrificing recall. Though I am not able to draw statistical insights given my limited observer pool, my results suggest that human supervision could be used as quality control for AI output, as in most human-in-the-loop AI approaches (HITL,

[153, 20]), and may also be used to guide inexperienced observers on challenging detection/classification tasks such as ours.

Though my ensemble models consistently outperformed individual CNN models (Table 4.1), I found the performance boost to be too small to justify the added computational cost of running imagery through ensembles when compared with the best-performing individual CNN (CNN 3, f1-score 0.69 vs. ensemble 1, f1-score 0.71). This similarity in performance is not surprising given the pronounced impact of features coming from CNN 3 on the best-performing ensembles (Figure 4.6), with minor contributions from a few other CNN models. Moreover, though I had a diverse set of hyperparameters within my 10 best-performing models, they hinged on the same datasets and model architecture (U-Net), which may have contributed to the high redundancy in including features from multiple CNNs. In contrast, several successful cases of applying ensemble models to CV rely on merging widely different individual components (e.g., [105, 111]). On the other hand, ensemble models consistently outperformed individual CNNs in terms of the correlation between model logits and true labels (Table 4.1), which makes them more desirable as an AI-guided annotation tool and could translate to a lower bias on novel input imagery.

While my approach performs extremely well in simple terrain, with large groups of seals, it does encounter difficulties with rough terrain and single seals (Figure 4.7). Notably, these settings also tend to be the most challenging for human observers, shown by the high correlation between AI and observer errors. These difficulties are unlikely to be surmounted directly by improvements in AI because I cannot reliably annotate ground-truth datasets in these settings. Although identifying the portions of VHR scenes where seals, if present, could be detected is a tractable problem for modern CV models, estimating seal densities in areas where they cannot be detected is a non-trivial problem and merits further investigation.

In addition to high out-of-sample performance (Table 4.1), when compared with

other candidate sampling methods for surveying pack-ice seals (i.e., fixed-wing airplanes, helicopters, UAVs, and human expert-based VHR surveys), AI-based approaches showed an operational cost comparable to that of the cheapest option available (helicopter-based flight transect surveys) [53], even taking into account the considerable cost of purchasing commercial VHR imagery. Cost-efficiency aside, GPU-accelerated AI-based approaches generate orders of magnitude fewer emissions than any other surveying method mentioned above [53]. The success of citizen science campaigns such as SOS [76], however, show the potential to utilize regular citizen science surveys as a validation method for fully automated pack-ice seal detection pipelines, especially given the difficulty of covering every potential real-world scenario during model evaluation.

Our results show compelling evidence for the immediate applicability of CNN-based, fully automated approaches for pack-ice seal surveys in VHR imagery. Moreover, they highlight the importance of comprehensive hyperparameter search studies and diverse training and evaluation datasets when employing AI methods to address complex tasks such as Antarctic pack-seal annotation in VHR imagery. With the addition of a pre-processing step to select VHR scenes where seals, if present, could be found and regular random checks by human observers for quality control, my approach with CNN ensembles is capable of delivering reliable, continental-scale putative Antarctic pack-ice seal locations.

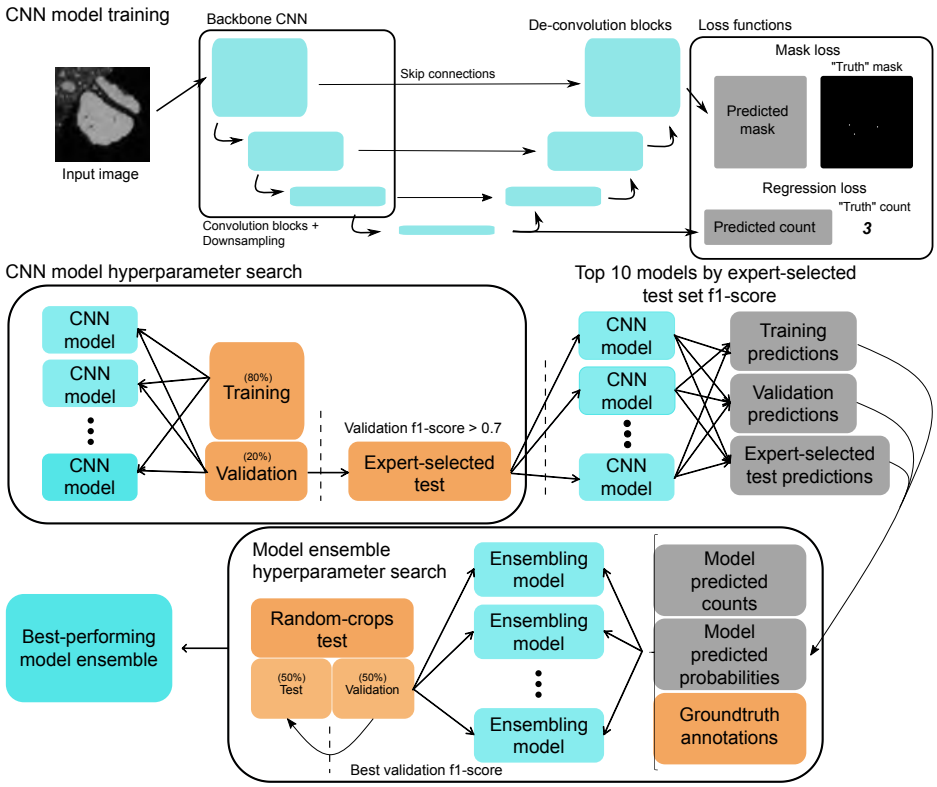

FIGURE 4.1: Simplified diagram for SealNet 2.0 showing the training of individual CNN models, model hyperparameter search, and model ensembling. Boxes colored in light-blue denote models, orange boxes denote datasets, and gray boxes denote model output. Thick black lines from datasets to models indicate model training. Dashed vertical lines indicate model selection steps. The best individual CNN models are trained on seal detection, including centroid segmentation and seal count regression, using a random search with training and validation, and the f1-score at the expert-selected test set for model selection. The best ensembling models are selected via Bayesian optimization, using top-10 CNN model predictions for the training set, validation set, and the expert-selected test set as dependent variables for training; true positive vs. false positive as the response variable; and the f1-score at the validation split from the random crops test set as a validation metric. Finally, I use the test portion of the random crops test set to estimate the out-of-sample performance of the best-performing model ensemble.

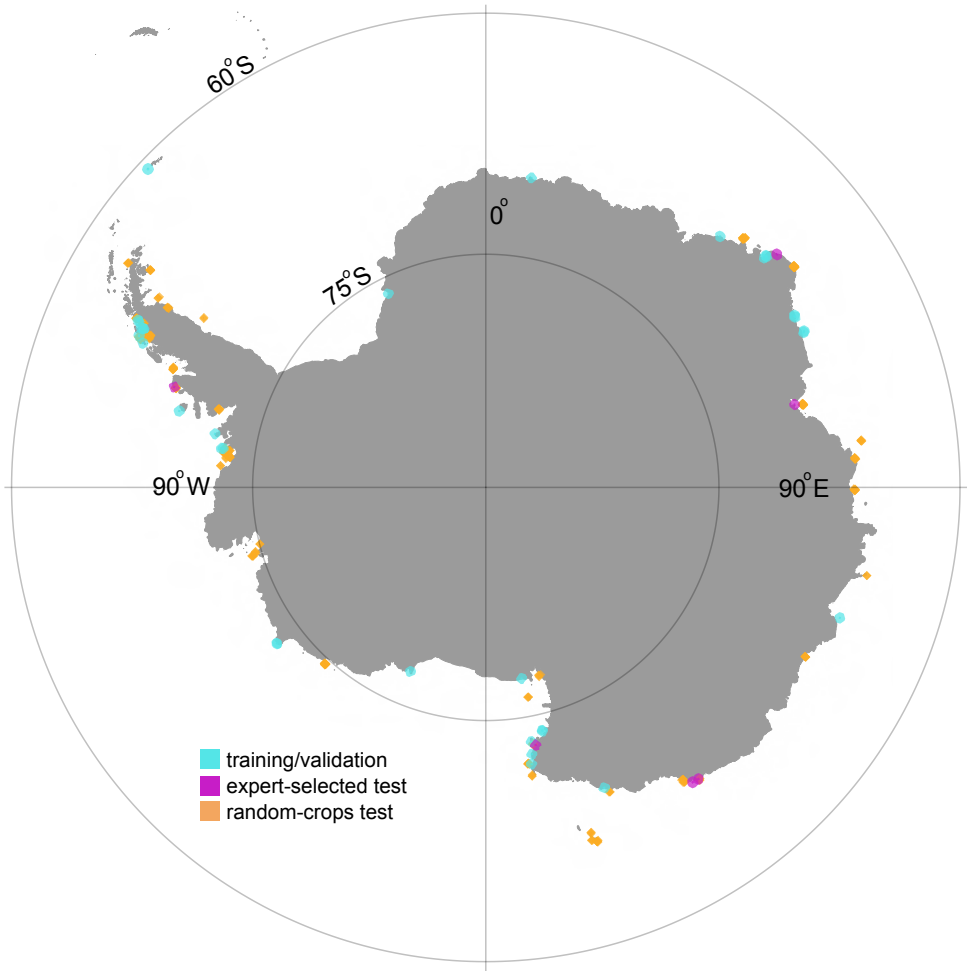

FIGURE 4.2: Training/validation set (light-blue), expert-selected test set (magenta), and random-crop test set (orange). Polygons denote entire Worldview-3 panchromatic scenes for the training/validation set and the expert-selected test set, and 1 km^2 crops within panchromatic Worldview-3 scenes for the random crops test set.

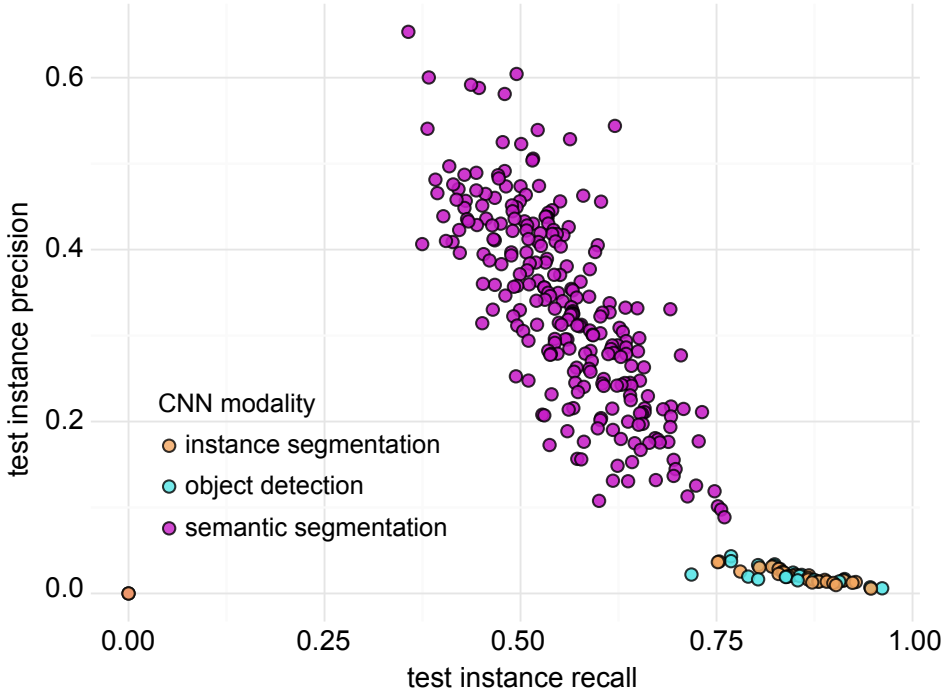

FIGURE 4.3: Expert-selected test set f1-score for hyperparameter search experiments from different computer vision domains. To ensure a fair comparison of models from different domains, semantic segmentation output masks are passed through a sigmoid transform and thresholded to extract mask centroids. Similarly, instance segmentation and object detection output bounding boxes are converted to centroids to evaluate matches with 'truth' centroids. To avoid unnecessary expenditure of GPU credits, experiments that did not perform well on the validation set (validation f1-score > 0.7 for semantic segmentation models and >0.5 for instance segmentation and object detection models) were not carried into testing.

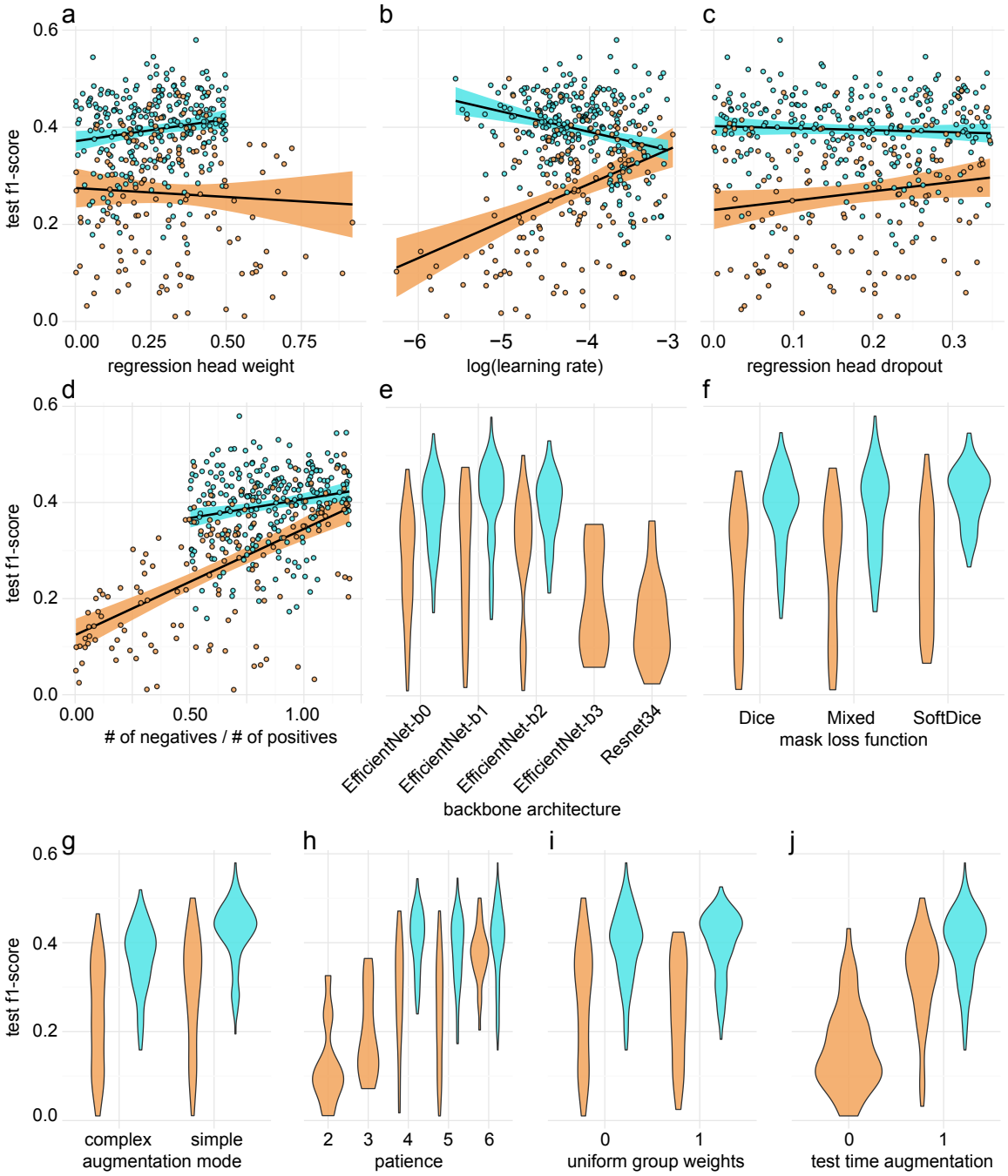

FIGURE 4.4: Results from random search hyperparameter study for semantic segmentation models, with phases one (orange) and two (teal). For continuous hyperparameters—namely, regression head weight, learning rate, regression head dropout, and negative-to-positive ratio—each circle corresponds to an independent random search experiment. For the continuous hyperparameters—regression head weight, learning rate, negative-to-positive ratio, and the discrete parameters—backbone architecture, patience, and test time augmentation, I narrowed down the range of options to speed up convergence on a best-performing model. Experiments for which test f1-score was below 0.01 are excluded from this plot.

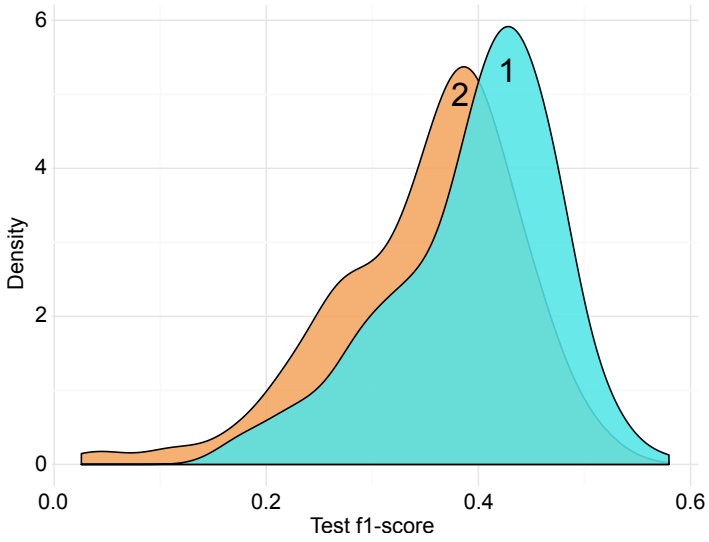

FIGURE 4.5: Side-by-side comparison between phase 2 experiment results with (1) and without (2) regression post-processing. Post-processing consisted of discarding predicted points within patches where regression output (i.e., predicted number of seals in a patch) is smaller than a specified threshold. For each model, I explore a range of thresholds to obtain the maximum possible test f1-score obtained after post-processing, using the same optimal threshold across the entire expert-selected test set.

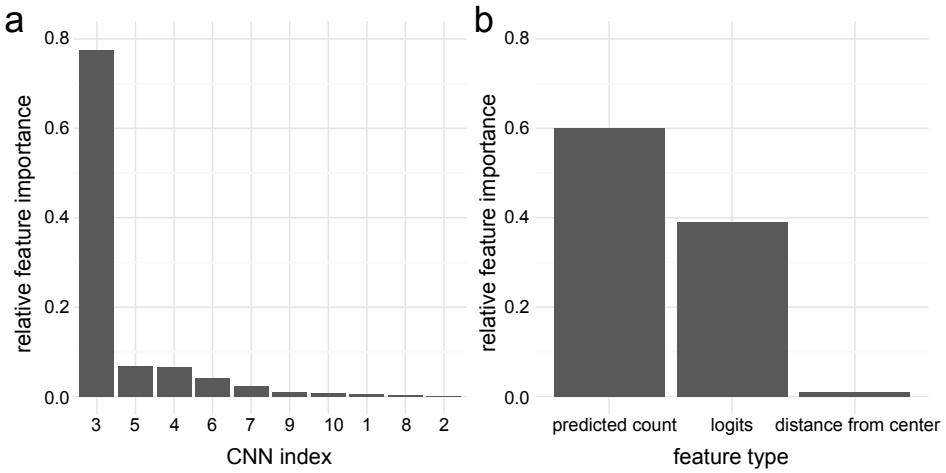

FIGURE 4.6: Feature importance for ensemble model features grouped by CNN index (**a**) and feature type (**b**). Ensemble models were either CatBoost or XGBoost tree-based ensembles trained for classifying false-positive and true-positive seal detections. Models were derived from a hyperparameter search using a training set with the logits, predicted seal counts, and distances from crop centers from the output of 10 U-Net CNNs at the validation and expert-selected test sets. Feature importances were obtained via Shapley scores at the validation portion of the random crops test set.

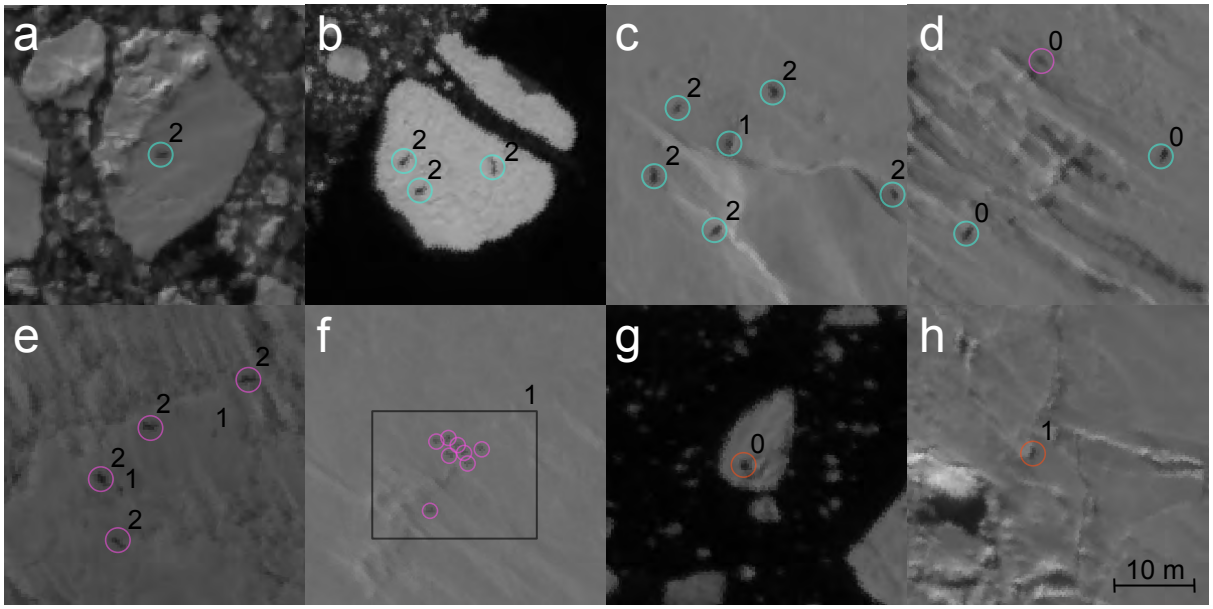

FIGURE 4.7: Prediction samples from my best ensemble model on eight scenes from the random crops test set. Samples were chosen to represent scenarios where the model predicts seal locations correctly (panels **a**–**d**), fails to find existing seals (panels **d**–**f**), and annotates background objects as seals (panels **g**,**h**). Seals marked with teal circles indicate true positives (i.e., predicted seal present in consensus dataset), whereas purple and orange circles indicate false-negative and false-positive seals, respectively. Numbers next to circle annotations indicate the number of human observers that agreed with that particular model annotation. Number '1' annotations unaccompanied by circles in panel *e* indicate edge cases where a human observer annotated a seal that was not present in the consensus dataset or model predictions. Imagery copyright Maxar Technologies Inc. 2022.

Chapter 5

A comprehensive review of pack-ice seal sampling methods

5.1 Abstract

Antarctic pack-ice seals, through their ecological role as key Antarctic krill predators, are critical to the Southern Ocean ecosystem. Shifts in sea ice distribution caused by anthropogenic climate change and krill fisheries threaten their populations. While initially surveyed by vessel or aircraft transects, very high-resolution remote sensing imagery has been proposed as a safer and potentially cheaper alternative. The sheer volume of imagery, however, creates a bottleneck in the availability of expert human annotators. AI-based, fully-automated surveys offer true scalability and, while imperfect, provide consistent annotations, and are unaffected by observer fatigue or other factors external to the image itself. Much progress has been made, including sea ice segmentation models that are able to restrict input imagery to potential seal habitat only, human-level seal detection models, and the HPC middleware required to apply this efficiently at scale. However, a pan-Antarctic survey using remote sensing comes with a number of challenges: 1) detecting seals in very-high-resolution imagery is a daunting task even for trained experts and relies heavily on contextual clues, making proper statistical treatment pivotal to go from putative seal locations to population

estimates; 2) variability in lighting, terrain, off-nadir angle, and sea ice conditions impose severe limitations on the reliability of validation and test sets; and 3) limitations in our understanding of seal haul-out behavior hamper our efforts to estimate the portion of seals available for detection (i.e. not submerged) at any moment in time. Here I outline a schematic of a fully-automated pipeline for regular pan-Antarctic seal surveying and compare it with other available surveying methods in terms of cost, coverage, and carbon emissions.

5.2 Introduction

The present study highlights recent advances (i.e. CNN-based seal detection models [50, 52] and sea ice segmentation models [51]) and the remaining obstacles to establishing a fully-automated pack-ice seal surveying program. Finally, I compare and contrast fully-automated approaches with citizen science surveys and vessel/aircraft-based surveys in terms of cost, emissions, and reliability, and discuss the future needs for a regular pan-Antarctic seal survey program.

5.3 Materials and Methods

5.3.1 Imagery

To scope out imagery requirements for surveying pack-ice seals I queried all VHR imagery IDs that followed these 5 criteria: 1) obtained with the Worldview-3 (WV-3) sensor; 2) south of 55 degrees of latitude; 3) at least partially covering the ocean surface; 4) at least partially cloud-free (i.e. cloud-cover < 100%); and 5) off-nadir angle from sensor < 30 degrees. My initial set of scene IDs was obtained from the Polar Geospatial Center (PGC) catalog and contains all available VHR commercial imagery up to April 2022. I used a high-resolution ADD-derived sea ice mask from the Quantarctica project [90] to mask out scenes on land. I used PGC cloud cover estimates to exclude scenes

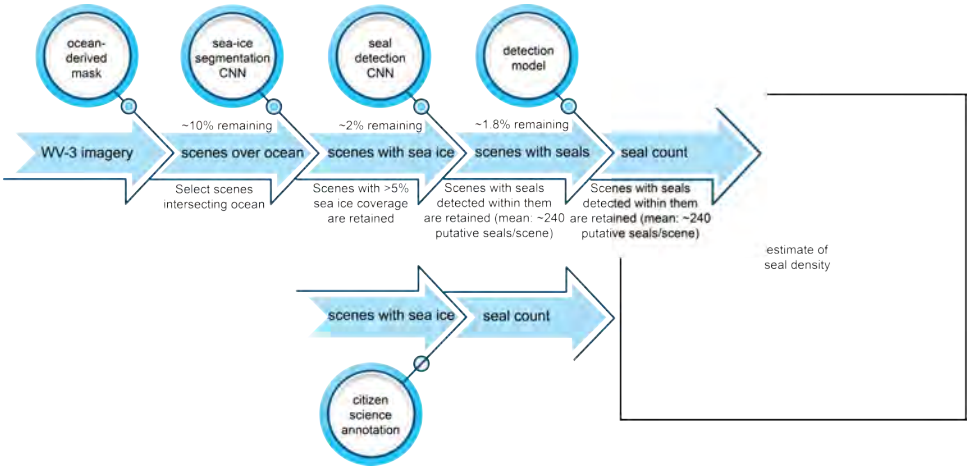

FIGURE 5.1: Schematic for the SealNet2.0 fully automated seal detection pipeline. Starting from the full corpus of suitable WV-3 scenes below 55 degrees of latitude, I follow a 5-step pipeline to obtain seal density estimates: a) find scenes that overlap with an ADD-derived ocean mask (i.e. scenes over the ocean); b) process scenes over the ocean with a sea ice segmentation CNN to obtain the subset of scenes with relevant seal habitat; c) process scenes with sea ice with an ensemble of seal detection CNNs to obtain georeferenced putative seals; d) use a detection model that uses haul out probability and model uncertainty to draw credible intervals for seal population sizes; e) aggregate by geography and time to get density estimates. The outside citizen science annotation loop serves as a guardrail to validate the quality of model output and flag potential abnormalities.

that are determined to be completely covered by clouds. My threshold for off-nadir angle ensures that the resolution is sufficiently high to detect potential seals.

5.3.2 Seal detection pipeline

To explore the performance and requirements of a fully-automated seal detection pipeline I use SealNet2.0 [52] CNN ensembles with a sea ice segmentation CNN [51] as a pre-processing step that subsets input imagery to those with potential seal habitat 5.1.

Sea ice pre-processing CNN

I utilize a sea ice segmentation CNN from my previous study [51], originally designed to segment ice floes, to calculate the area covered by ice floes for WV-3 scenes, which in turn can be used to filter scenes with potential seal habitat. This filtering step is

employed to mitigate false positive errors outside typical seal habitat and minimize the computational cost involved with the far heavier seal detection CNN ensembles. My filtering rule is fit using total floe area and percent floe cover for a set of 3000 WV-3 scenes labeled according to their potential for seal detection. A scene where seals could be present – and, if present, would be detectable – receives a label of "1" and a scene where seals are either extremely unlikely to be present or, if present, could not be accurately detected (e.g. extreme terrain roughness) would receive a label of "0". Scene labeling was performed by a single observer, with several hundred hours of seal annotation experience in VHR imagery, using a graphical interface to inspect scene thumbnails with overlaid ice floe maps annotated by the sea ice segmentation CNN, loading the full resolution scene and zooming in when necessary. I optimized my decision rule using the f1-score as a metric, evaluated at a stratified five-fold cross-validation split, where every split is sampled at random in a way that guarantees the original "truth" label distribution.

Seal detection CNN ensemble

Built upon the original SealNet study [50], my seal detection CNN ensemble ([52]) leverages my sea ice segmentation pre-processing step to focus solely on sea ice habitat for training and evaluation. Instead of predicting directly with a CNN, my approach utilizes an XGBoost [28] classifier to combine outputs from several seal centroid segmentation U-Net [115] models. Individual U-Net models were implemented using the Segmentation Models Pytorch package [149], with an EfficientNet [134] backbones and added regression heads trained to count seals in addition to their original segmentation head trained to segment out seal centroids. To estimate prediction overhead in terms of cost and CO^2 emissions, I calculate the average amount of floating point number operations (measured in billions of floating point operations, or GFLOPs) required to process 1 km^2 of input imagery using. My calculations for the cost of processing

imagery are based on NVIDIA®RTX 4090 GPUs with 24GB (quoted at $1599 in October 2022). To estimate the cost per hour, I spanned the retail price of that GPU across an estimated service life of four years. Due to the overhead from applying test-time-augmentation to the output (i.e. average the output across flipped and rotated versions of the input, 8x), mosaicing output from overlapping patches (i.e. tile with a 50% overlap between neighboring patches and average the output from overlapping areas, 4x), and generating output from three different U-Nets (3x), the cost of processing imagery through my seal detection pipeline increases from the 28776.05 GFLOPs (cost for single U-Net model, without post-processing) for processing a km^2 of WV-3 with a single U-Net to 2762501.25 GFLOPs per km^2, with all the ensembling steps for seal detection accounted for. The sea ice pre-processing step, however, does not ensemble from multiple U-Net models, reducing processing cost by a factor of three. I chose a consumer-grade GPU for my calculations instead of server-grade GPUs typically found in cloud provider AI nodes for their superior cost-efficiency when compared to the latter.

5.4 Comparison with other sampling methods

I compare the feasibility of VHR satellite imagery surveys to other potential seal sampling methods (i.e. aerial transects with helicopters, fixed-wing aircraft, and UAVs) by comparing resolution ($\frac{pixel}{m}$), costs (2022 US$), and greenhouse gas emissions (kg of CO_2) over a standardized unit of area. I estimate the operational cost by adding in the average price for fuel, personnel, and rental over a unit of time. When leasing a piece of equipment is non-trivial, I estimate cost over time by spanning retail price across an average service life for that piece of equipment. To calculate area coverage for helicopter and fixed-wing aircraft seal surveys, I based my estimates on the exact aircraft models and sampling strategies used at the APIS project surveys [56]. Operating costs

for the latter were obtained using Aircraft Cost Calculator®in October 2022. UAV operation cost estimates were obtained from a model specifically built the operate in the Antarctic, the PW-Zoom fixed-wing UAV [157]. Since this specially built model cannot be easily priced, I used a quotation for a fixed-wing UAV with similar capabilities spanned across a four-year service life to estimate its operational costs. Since UAV operations require the support of a research ship, I include operational cost estimates from a small ice-breaker in the US research fleet [104] to the cost of UAV surveys. Finally, I did not take the cost of re-fueling operations for vessels as these depend on a number of factors such as the availability of refueling bases.

Since the above sampling methods, with the exception of helicopter surveys, work by recording images for posterior annotation, I compare the feasibility of available annotation methods, namely, human observers, and fully-automated AI models in terms of their cost (2022 US$) and emissions (kg of CO_2) and predictive performance following out-of-sample performance for VHR imagery from my previous study. I did not include predictive performance for observers in helicopter-based transects because, to the best of my knowledge, there is no study providing standardized comparisons between all three methods. Finally, I compare the aforementioned survey methods in terms of cost, coverage, and emissions in a table including their efficiency per hour and during full, 1200-hour field seasons.

5.5 Results

Starting from all available WV-3 imagery below 55 degrees of latitude, and after filtering out a small portion of scenes under complete cloud cover or with an off-nadir angle $> 33^o$, I found that only 9.5 $\pm$ 1.7 % of WV-3 scenes were captured over the ocean, leading to an average yearly volume of 5790.62 $\pm$ 1708.25 scenes (excluding the incomplete 2022 season). The average number of scenes over the ocean remained more or less constant since the launch of the WV-3 satellite in 2014 (Figure **??**). With a

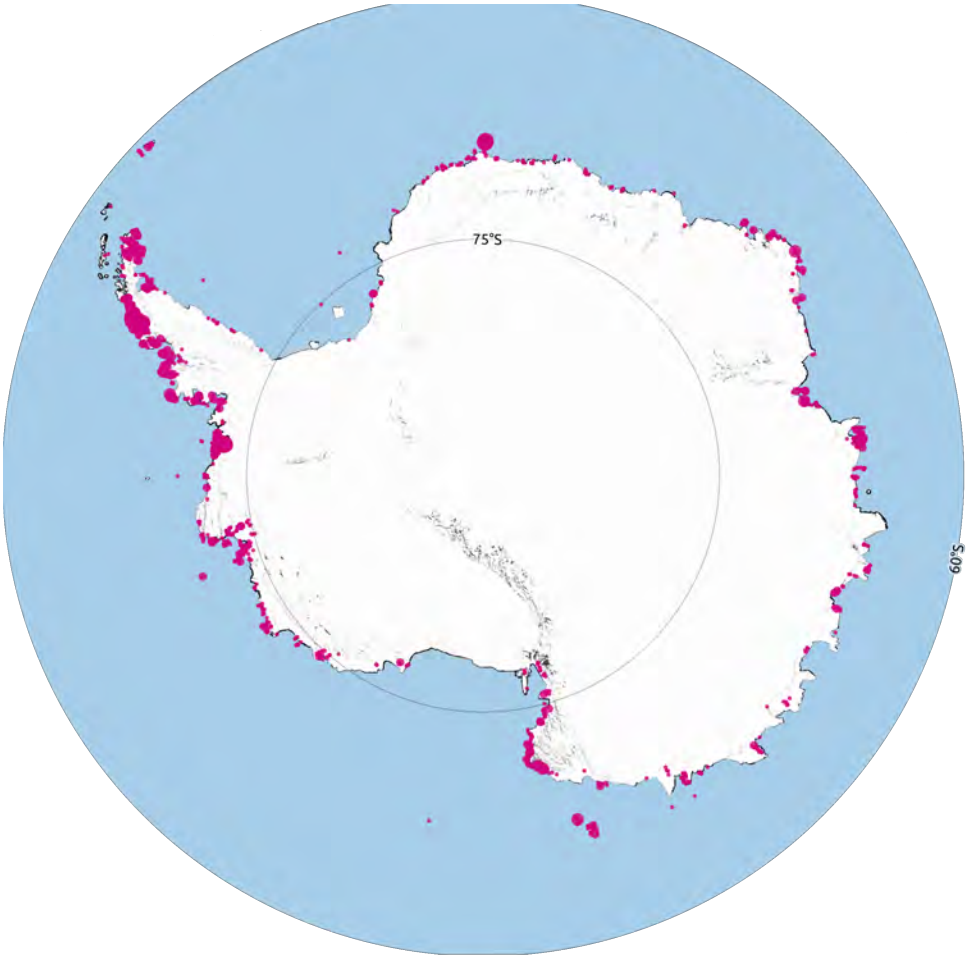

FIGURE 5.2: Putative seals from trial SealNet2.0 run. Every magenta spot indicates a putative seal detected with SealNet2.0 on a set of 14983 WV-3 images. Input images ranged from late September to early April in the years 2014 through 2021. Imagery copyright DigitalGlobe, Inc. 2021.

median area for scenes over the ocean of 241.31 km^2, the estimated yearly cost for purchasing a season's worth of imagery starts at 23.64 (million US$) and could reach as much as 43.42 (million US$). Using a simple threshold rule for segmentation-model-derived sea ice area (i.e. area covered by ice floes $\geq$ 1455 m^2), I am able to classify out-of-sample scenes for their seal detection potential with an accuracy of 0.92, a recall of 0.73 and a precision 0.76.

Our compiled results table (5.1) shows that helicopters, especially model Bell 206-L, offer the greatest cost-efficiency per area covered (0.1 km^2/$). Surveying WV-3 images with SealNet2.0, however, is the most efficient sampling method by several orders of

magnitude in terms of minimizing emissions, covering an area of 5072.22 km^2 under a single kilogram of CO_2. Moreover, an entire season of VHR imagery covers >10 times the upper bound for the area of sea ice that could be covered by helicopter in an entire field season, even under the unrealistic assumption of no re-fueling operations.

TABLE 5.1: Comprehensive seal survey method comparison including cost, coverage, and emissions involved. Sampling methods include helicopters (H), fixed-wing airplanes (P), research ships (RS), fixed-wing drones (UAV), human observers, and my fully automated pipeline (AI). Annual estimates for coverage, cost, and emissions assume a full 1200-hour Antarctic field season for on-site surveys and a 2000-hour full work year for human observers. An estimate for 384 human observers is included to illustrate the requirements for processing an entire season of VHR imagery manually. Because UAV operations require a support vessel I include an estimate of cost with six PW-ZOOM UAVs aboard the US Navy ice breaker RV Laurence M. Gould (LMG). Operating costs for the LMG do not include renting or buying the vessel itself. Similarly, the costs of annotating records obtained from cameras fixed-winged planes, and drones are omitted from this summary. Since I did not take refueling and maintenance operations into account in my calculations, coverage and cost for aircraft should be treated as an upper bound.

	Coverage (km^2 / hr)	Cost (\$ / km^2)	Emissions (kg CO_2 / km^2)	Coverage (km^2 / year)	Cost (\$ / year)
PW-ZOOM (UAV)	4.4	6.19	1.17	5280.0	32,688.1
Bell 206L (H)	118.0	9.9	3.14	141600.0	1,402,428.0
BO 105 (H)	118.0	13.14	3.62	141600.0	1,860,546.0
SealNet2.0 (AI)	1223.17	24.0	2×10^{-4}	1,397,760.0	33,536,187.02
UAV (x6) + LMG	26.4	31.89	1.71	31680.0	1,010,365.07
Human obs.	1.75	35.43	0.02	3640.0	128960.0
Human obs. (x384)	672.0	35.43	0.02	1,397,760.0	49,520,640.0
Polar 2 (P)	33.6	39.3	17.73	40320.0	1,584,528.0
LMG (RS)	6.0	198.51	270.62	7200.0	1,429,293.37

5.6 Discussion

Our study provides a baseline for the feasibility of candidate sampling methods for establishing a regular Antarctic pack-ice seal survey program. Taking into account a wide variety of factors such as fuel consumption, crew size, equipment lease, etc., I find that helicopter-based surveys with field biologists detecting seals in real-time,

especially those using smaller aircraft such as model Bell 206-L, outperform other sampling methods in terms of cost-efficiency by a wide margin (>2x more efficient than the runner up method, AI-based VHR surveys). However, when compared with the latter, AI-based surveys in VHR have the desirable properties of 1) keeping permanent records to allow verification; 2) avoiding dangers involved with operating aircraft in the Antarctic; 3) ease of scalability with the purchase of more imagery and addition of more GPUs, and 4) several orders of magnitude fewer CO_2 emissions (>125 times more efficient than the breathing of a human observer and > 18,000 times more efficient than a Bell 206-L helicopter). The puzzling low cost-efficiency of UAVs 5.1, given their relatively low fuel uptake and comparable cruise speed to helicopters, stems from mosaicing techniques requiring a large overlap (>60%) between neighboring images (e.g. [98])

Even adding a conservative overhead of 25% extra work for transferring imagery, tiling scenes into patches, and storing output, a single modern GPU makes it possible to go over a season's worth of imagery in 1142 hours (∼50 days) – highlighting the cost-efficiency and reduced carbon footprint of processing VHR imagery with fully-automated systems. For comparison, taking into account an average of 4x coverage for statistical post-processing, it would take 384 full-time human observers to cover the same extent. The elevated cost of commercial satellite imagery, however, remains a formidable obstacle to the implementation of regular VHR-based surveys, at an average yearly expense of $33,553,035.38. Alleviating these costs in international research areas such as the Antarctic would easily make VHR-based surveys more cost-efficient than the other available survey methods. Moreover, focusing on specific geographic locations and time-frames can greatly reduce this price tag. For instance, sampling the entire subset of imagery over fast-ice that is relevant during the Weddell seal breeding season, e.g. [75], reduces yearly cost by an order of magnitude.

CNN-based methods, however, introduce important additional caveats since they

often perform well in the exact context they were trained for but may fail to generalize to a broader context even with a clear separation between training, validation, and tests sets. Our proof-of-concept study in automated seal detection [50] illustrated this potential. In addition, the spatial resolution of VHR imagery is much lower than that provided by aerial imagery and though sensors such as WV-3 are adequate for seal detection, they do not provide enough detail for individual seals to be directly identified to species. Species identification is important for policymakers because, as has been the case for *Pygoscelis* penguins ([29]), each species might respond differently to external pressures such as krill fishing and sea ice loss (e.g. [45]). One way to circumvent this drawback to UAV-based detection methods would be to use contextual clues such as the spatial arrangement of groups of seals, substrate characteristics (e.g. fast-ice vs. pack-ice, floe size, etc.), and haul out phenology to infer putative species identification. Early work in this area has been promising and this remains an exciting area for future research.

A comprehensive comparison of performance and efficiency between citizen scientist campaigns and fully-automated AI pipelines for seal detection surveys merits further investigation. While I believe the latter provides compelling advantages in terms of general cost-efficiency, citizen scientist campaigns represent a promising generator of training data and AI validation. This is especially important given the complexity of the task at hand and the potential pitfalls of relying too much on performance estimates obtained on limited test sets given the considerable risk of CNN-based methods to overfit training and validation sets.

Though our adapting our sea ice segmentation model reliably predicted scenes with seal habitat upon evaluation, my pipeline could be further improved by replacing it with a potentially more efficient and more accurate classifier CNN, specialized in pinpointing locations where seals, if present, could be found. To convert putative

seals from a fully-automated seal detection model into credible intervals for population density, we will need to develop dedicated statistical tools that account for detection errors, seal haulout behavior, seasonality, and dependence on external environmental factors.

Fully automated, VHR-based, seal detection surveys stand out as clean, affordable, and fast survey methods to monitor the health of Antarctic pack ice seal populations and, through them, the entire SO ecosystem.

Chapter 6

Conclusion

In this thesis, I have developed an automated CNN-based detection pipeline for massive-scale seal and sea-ice surveys that progresses from an initial proof-of-concept (Chapter 2 [50]) through to a robust model (Chapter 4 [52]) that harnesses a novel sea ice segmentation algorithm (Chapter 3 [51]) and, when placed in the context of a techno-economic comparison, delivers several considerable advantages over alternative methods (Chapter 5 [53]). Beyond the application to pack-ice seal surveys explored in my dissertation, VHR imagery has been proposed as a general tool to survey other large-bodied animals (e.g. African megafauna [148], whales [17]), elephant seals [92] with applications for the design and monitoring of marine protected areas [74].

The utility of VHR imagery for animal surveys hinges on a few key aspects about the target organism and its spatial context: 1) how large is the animal, and how distinctive it is from its background and other co-occurring animal species? i.e. is there enough visual information to detect it with confidence?; 2) how often is the animal available for detection? is that predictable?; 3) are there other sampling tools that work for this context? (i.e. does it make sense to purchase VHR imagery or is the organism amenable to cheaper alternative surveys?); and 4) is there an existing body of experts to create annotations for training, model selection, and out-of-sample evaluation? The ideal organism for VHR-enabled survey is large-bodied (> 1.5 m), contrasts visually against its preferred environment, and displays a distinctive pattern of aggregation that allows it to be distinguished from other features in the landscape. Smaller-bodied

animals are also amenable to VHR surveys where they aggregate in dense clusters or when they modify the landscape (e.g. through the production of guano [85]) in ways that can be observed from space. VHR-based surveys have particular advantages over alternative methods in remote and inaccessible regions where the logistical burdens of conducting field operations limit direct access. In these cases, the cost of imagery and the laborious annotation process required to train automated detection systems easily outweighs the ongoing costs of long-term monitoring using more traditional means.

Far more than demonstrating the potential for using these tools for pack-ice seal monitoring, my contributions are twofold: A) a reliable tool for surveying Antarctic pack-ice seals that can immediately be used to extract putative seal locations in WV-3 imagery; and B) the largest corpus of expert-curated, geo-tagged seal annotations available. Finally, I provide a sample of > 350,000 putative seals obtained by processing 14983 WV-3 scenes through my SealNet2.0 pipeline (Figure 6.1). With statistical treatment to address imperfect detection and partial availability, such a dataset raises the opportunity to address a number of important ecological questions on Antarctic pack-ice seals population trends, breeding biology, phenology, and many other aspects. Moreover, the presence/absence of detected seals, much like penguin colonies (e.g. [75]), could be used as an environmental covariate when modeling other organisms and improving our understanding of niche partitioning in the SO. Finally, expanding my sea ice segmentation model to include broad categorical labels could provide granular information for habitat modeling applications and even improve the quality of the ice floe segmentation output given the more rich supervision signal for training the CNN.

One of the biggest drawbacks of adopting methods such as mine, as exposed in (Chapter 4, Table 5.1), is the substantial cost of acquiring commercial VHR imagery. I argue, however, that rather than excluding all but the wealthiest of research institutions, commercial satellite imagery may democratize Antarctic science. Researchers in countries without Antarctic research programs now have the possibility of obtaining

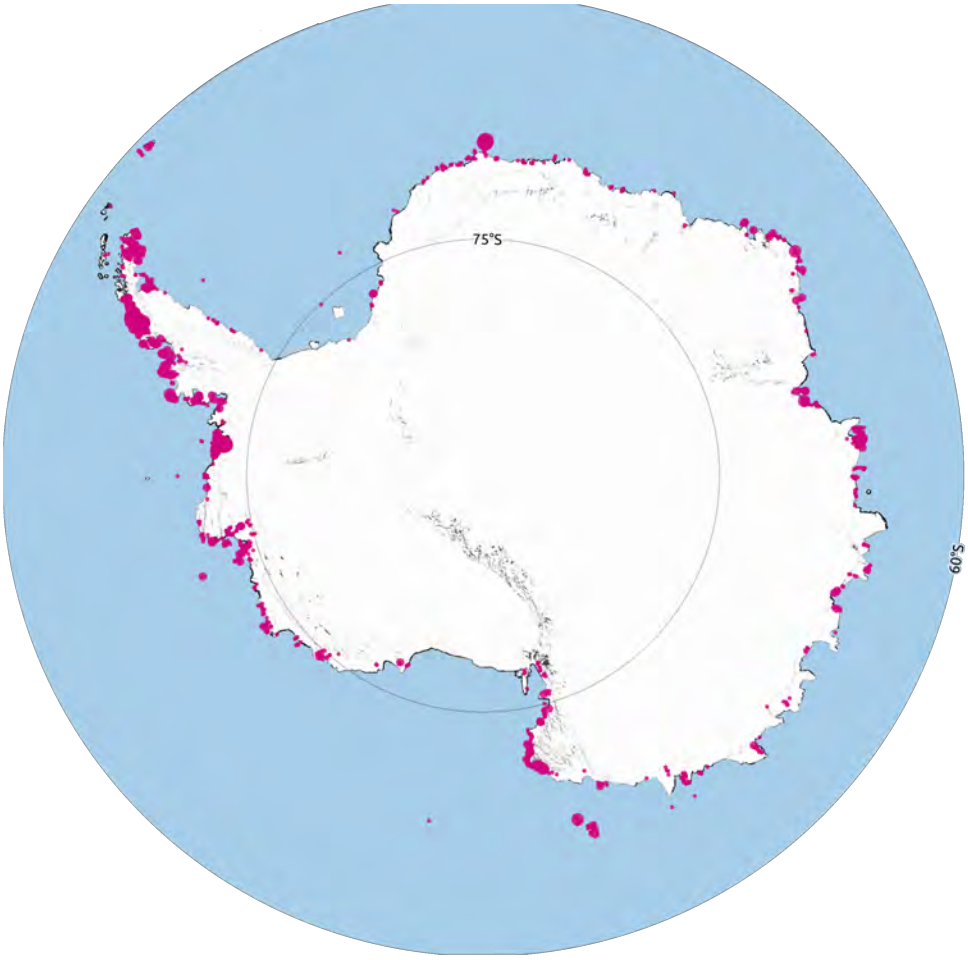

FIGURE 6.1: Putative seals from trial SealNet2.0 run. Every magenta spot indicates a putative seal detected with SealNet2.0 on a set of 14983 WV-3 images. Input images ranged from late September to early April in the years 2014 through 2021. Imagery copyright DigitalGlobe, Inc. 2021.

fine-grained information from a region of interest, often with repeated observations, while spending orders of magnitude less than it would take to build and operate ice-breaker ships [104] and Antarctic field stations [88].

I have paved the way for a fully-automated, reliable, and sustainable method for surveying Antarctic pack-ice seals. The next logical step would be developing statistical tools for true uncertainty estimation taking addressing two aspects: imperfect detection (i.e. false-positive and false-negative errors by the detection model) and partial availability (i.e. dealing with the portion of the population that is underwater at the time the input image was captured). One major challenge in integrating

this step into my seal detection pipeline (Chapter 4, Figure 5.1) is the dramatically lower spatial resolution at which relevant environmental features are obtained (e.g. bathymetry, 500m cells) without sacrificing granularity nor creating pseudo-replicate observations. To check for the validity of an environmental feature set for detection modeling would be including them within the tree-based CNN ensemble model and verify whether adding this new piece of information improves upon previous out-of-sample performance. In case they are informative, extracting Shapley scores (a game-theoretical measurement of the contribution of each feature to the final output [123]) for each detection could work as a surrogate for a habitat suitability model, hinting at combinations of environmental features that are conducive to the presence/absence of seals.

Bibliography

[1] S Ackley et al. "The International Antarctic Pack Ice Seals (APIS) Program. Multi-disciplinary research into the ecology and behavior of Antarctic pack ice seals. Summary Update". In: *The Expert Group on Seals (EGS); Scientific Committee on Antarctic Research (SCAR). Marthan N. Bester, D. Sc., Chief Officer, Brent S. Stewart, Ph. D., JD, Secretary (eds.)* (2006).

[2] Shubhra Aich and Ian Stavness. "Improving object counting with heatmap regulation". In: *arXiv preprint arXiv:1803.05494* (2018).

[3] Takuya Akiba et al. "Optuna: A Next-Generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, 2623–2631. ISBN: 9781450362016. DOI: 10.1145/3292500.3330701. URL: https://doi.org/10.1145/3292500.3330701.

[4] Razvan Andonie and Adrian-Catalin Florea. "Weighted random search for CNN hyperparameter optimization". In: *arXiv preprint arXiv:2003.13300* (2020).

[5] Kevin R. Arrigo and David N. Thomas. "Large scale importance of sea ice biology in the Southern Ocean". In: *Antarctic Science* 16.4 (2004), 471–486.

[6] A. Atkinson et al. "A re-appraisal of the total biomass and annual production of Antarctic krill". In: *Deep Sea Research Part I: Oceanographic Research Papers* 56.5 (2009), pp. 727–740. ISSN: 0967-0637. DOI: https://doi.org/10.1016/j.dsr.2008.12.007. URL: https://www.sciencedirect.com/science/article/pii/S0967063708002513.

[7] Angus Atkinson et al. "KRILLBASE: a circumpolar database of Antarctic krill and salp numerical densities, 1926–2016". In: *Earth System Science Data* 9.1 (2017), pp. 193–210.

[8] Grant Ballard et al. "Coexistence of mesopredators in an intact polar ocean ecosystem: the basis for defining a Ross Sea marine protected area". In: *Biological Conservation* 156 (2012), pp. 72–82.

[9] Grant Ballard et al. "Fine-scale oceanographic features characterizing successful Adélie penguin foraging in the SW Ross Sea". In: *Marine Ecology Progress Series* 608 (2019), pp. 263–277.

[10] Alan S Belward and Jon O Skøien. "Who launched what, when and why; trends in global land-cover observation capacity from civilian earth observation satellites". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 103 (2015), pp. 115–128.

[11] John L Bengtson and Brent S Stewart. "Diving and haulout behavior of crabeater seals in the Weddell Sea, Antarctica, during March 1986". In: *Polar Biology* 12.6 (1992), pp. 635–644.

[12] John L Bengtson et al. "Distribution, density, and abundance of pack-ice seals in the Amundsen and Ross Seas, Antarctica". In: *Deep Sea Research Part II: Topical Studies in Oceanography* 58.9-10 (2011), pp. 1261–1276.

[13] David Berthelot et al. "Mixmatch: A holistic approach to semi-supervised learning". In: *arXiv preprint arXiv:1905.02249* (2019).

[14] MN Bester, JWH Ferguson, and FC Jonker. "Population densities of pack ice seals in the Lazarev Sea, Antarctica". In: *Antarctic Science* 14.2 (2002), pp. 123–127.

[15] Ed Blockley et al. "The future of sea ice modeling: where do we go from here?" In: *Bulletin of the American Meteorological Society* 101.8 (2020), E1304–E1311.

[16] Alex Borowicz et al. "Aerial-trained deep learning networks for surveying cetaceans from satellite imagery". In: *PLOS ONE* 14.10 (Oct. 2019), pp. 1–15. DOI: 10.1371/journal.pone.0212532. URL: https://doi.org/10.1371/journal.pone.0212532.

[17] Alex Borowicz et al. "Aerial-trained deep learning networks for surveying cetaceans from satellite imagery". In: *PloS one* 14.10 (2019), e0212532.

[18] Hugo Boulze, Anton Korosov, and Julien Brajard. "Classification of Sea Ice Types in Sentinel-1 SAR Data Using Convolutional Neural Networks". In: *Remote Sensing* 12.13 (July 2020).

[19] Ismael V Brack, Andreas Kindel, and Luiz Flamarion B Oliveira. "Detection errors in wildlife abundance estimates from Unmanned Aerial Systems (UAS) surveys: Synthesis, solutions, and challenges". In: *Methods in Ecology and Evolution* 9.8 (2018), pp. 1864–1873.

[20] Steve Branson et al. "Visual recognition with humans in the loop". In: *European Conference on Computer Vision*. Springer. 2010, pp. 438–451.

[21] Andrew S. Brierley and David N. Thomas. "Ecology of Southern Ocean pack ice". In: (2002).

[22] Shawn T. Brown et al. “Bridges-2: A Platform for Rapidly-Evolving and Data Intensive Research”. In: *Practice and Experience in Advanced Research Computing*. New York, NY, USA: Association for Computing Machinery, 2021. ISBN: 9781450382922. URL: https://doi.org/10.1145/3437359.3465593.

[23] J.M. Burns et al. “Fine-scale habitat selection of crabeater seals as determined by diving behavior”. In: *Deep Sea Research Part II: Topical Studies in Oceanography* 55.3-4 (), pp. 500–514.

[24] Alexander Buslaev et al. “Albumentations: Fast and flexible image augmentations”. In: *Information* 11.2 (2020).

[25] EL Cavan et al. “The importance of Antarctic krill in biogeochemical cycles”. In: *Nature communications* 10.1 (2019), pp. 1–13.

[26] Jieneng Chen et al. *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation*. 2021. DOI: 10.48550/ARXIV.2102.04306. URL: https://arxiv.org/abs/2102.04306.

[27] Pengguang Chen et al. “Gridmask data augmentation”. In: *arXiv preprint arXiv:2001.04086* (2020).

[28] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. New York, NY, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: http://doi.acm.org/10.1145/2939672.2939785.

[29] Gemma V Clucas et al. “A reversal of fortunes: climate change ‘winners’ and ‘losers’ in Antarctic Peninsula penguins”. In: *Scientific reports* 4.1 (2014), pp. 1–7.

[30] Josefino C. Comiso et al. “Passive microwave algorithms for sea ice concentration: A comparison of two techniques”. In: *Remote sensing of Environment* 60.3 (1997), pp. 357–384.

[31] Paul B Conn et al. “Estimating multispecies abundance using automated detection systems: Ice-associated seals in the Bering Sea”. In: *Methods in Ecology and Evolution* 5.12 (2014), pp. 1280–1293.

[32] Max Coon et al. “Arctic Ice Dynamics Joint Experiment (AIDJEX) assumptions revisited and found inadequate”. In: *Journal of Geophysical Research: Oceans* 112.C11 (2007).

[33] MD Coon, GA Maykut, and RS Pritchard. “Modeling the pack ice as an elastic-plastic material”. In: (1974).

[34] Lee Raymond Dice. “Measures of the amount of ecologic association between species”. In: *Ecology* 26.3 (July 1945), pp. 297–302.

[35] Janis L Dickinson, Benjamin Zuckerberg, and David N Bonter. "Citizen science as an ecological research tool: challenges and benefits". In: *Annual review of ecology, evolution, and systematics* (2010), pp. 149–172.

[36] Hai Ha Do et al. "Deep learning for aspect-based sentiment analysis: a comparative review". In: *Expert systems with applications* 118 (2019), pp. 272–299.

[37] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. "CatBoost: gradient boosting with categorical features support". In: *ArXiv* abs/1810.11363 (2018).

[38] Benjamin Dowden et al. "Sea ice classification via deep neural network semantic segmentation". In: *IEEE Sensors Journal* (2020).

[39] Hajo Eicken. "The role of sea ice in structuring Antarctic ecosystems". In: *Weddell Sea Ecology*. Springer, 1992, pp. 3–13.

[40] Rosana El Jurdi et al. "A surprisingly effective perimeter-based loss for medical image segmentation". In: *Medical Imaging with Deep Learning*. 2021. URL: https://openreview.net/forum?id=NDEmtyb4cXu.

[41] Albert W Erickson and M Bradley Hanson. "Continental estimates and population trends of Antarctic ice seals". In: *Antarctic ecosystems*. Springer, 1990, pp. 253–264.

[42] Andre Esteva et al. "Deep learning-enabled medical computer vision". In: *NPJ digital medicine* 4.1 (2021), pp. 1–9.

[43] Hauke Flores et al. "Impact of climate change on Antarctic krill". In: *Marine Ecology Progress Series* 458 (2012), pp. 1–19.

[44] Jaume Forcada et al. "Responses of Antarctic pack-ice seals to environmental change and increasing krill fishing". In: *Biological Conservation* 149.1 (2012), pp. 40–50.

[45] Jaume Forcada et al. "Responses of Antarctic pack-ice seals to environmental change and increasing krill fishing". In: *Biological Conservation* 149 (2012), pp. 40–50.

[46] Peter T Fretwell and Philip N Trathan. "Discovery of new colonies by Sentinel2 reveals good and bad news for emperor penguins". In: *Remote Sensing in Ecology and Conservation* 7.2 (2021), pp. 139–153.

[47] Lucas Girard et al. "Evaluation of high-resolution sea ice models on the basis of statistical and scaling properties of Arctic sea ice drift and deformation". In: *Journal of Geophysical Research: Oceans* 114.C8 (2009).

[48] Ross Girshick. "Fast R-CNN". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[49] Chengyue Gong et al. “KeepAugment: A Simple Information-Preserving Data Augmentation Approach”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 1055–1064.

[50] B.C. Gonçalves, B. Spitzbart, and H.J. Lynch. “SealNet: A fully-automated pack-ice seal detection pipeline for sub-meter satellite imagery”. In: *Remote Sensing of Environment* 239 (2020), p. 111617. ISSN: 0034-4257.

[51] Bento C. Gonçalves and Heather J. Lynch. “Fine-Scale Sea Ice Segmentation for High-Resolution Satellite Imagery with Weakly-Supervised CNNs”. In: *Remote Sensing* 13.18 (2021). ISSN: 2072-4292. DOI: 10.3390/rs13183562. URL: https://www.mdpi.com/2072-4292/13/18/3562.

[52] C. Wethington M. Gonçalves Bento and Heather J. Lynch. “SealNet 2.0: human level fully-automated pack-ice seal detection in very-high-resolution satellite imagery with CNN model ensembles”. In: *Remote Sensing, in review* ().

[53] Wethington M. Gonçalves Bento C. and Heather J. Lynch. “Roadmap to a fully-automated, pan-Antarctic pack-ice seal monitoring program”. In: ().

[54] Hugues Goosse et al. “Quantifying climate feedbacks in polar regions”. In: *Nature Communications* 9.1 (2018), pp. 1–13.

[55] Yating Gu, Yantian Wang, and Yansheng Li. “A survey on deep learning-driven remote sensing image scene understanding: Scene classification, scene retrieval and scene-guided object detection”. In: *Applied Sciences* 9.10 (2019), p. 2110.

[56] Eliezer Gurarie et al. “Distribution, density and abundance of Antarctic ice seals off Queen Maud Land and the eastern Weddell Sea”. In: *Polar Biology* 40.5 (2017), pp. 1149–1165.

[57] Yanling Han et al. “Sea ice image classification based on heterogeneous data fusion and deep learning”. In: *Remote Sensing* 13.4 (Feb. 2021).

[58] Judith Hauck et al. “On the Southern Ocean CO2 uptake and the role of the biological carbon pump in the 21st century”. In: *Global Biogeochemical Cycles* 29.9 (2015), pp. 1451–1470.

[59] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015).

[60] Kaiming He et al. “Mask R-CNN”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

[61] Mohammad D Hossain and Dongmei Chen. “Segmentation for object-based image analysis (OBIA): A review of algorithms and challenges from remote sensing perspective”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 150 (2019), pp. 115–134.

[62] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[63] Peter J. Huber. “Robust Estimation of a Location Parameter”. In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73 –101. DOI: 10.1214/aoms/1177703732. URL: https://doi.org/10.1214/aoms/1177703732.

[64] Elizabeth C Hunke, William H Lipscomb, and Adrian K Turner. “Sea-ice models for climate study: retrospective and new directions”. In: *Journal of Glaciology* 56.200 (2010), pp. 1162–1172.

[65] Tolulope Bamidele Ijitona, Jinchang Ren, and Phil Byongjun Hwang. “SAR sea ice image segmentation using watershed with intensity-based region merging”. In: *2014 IEEE International Conference on Computer and Information Technology*. IEEE. 2014, pp. 168–172.

[66] Shruti Jadon. “A survey of loss functions for semantic segmentation”. In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)* (Oct. 2020).

[67] Benjamin Kellenberger, Diego Marcos, and Devis Tuia. “Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning”. In: *Remote Sensing of Environment* 216 (Oct. 2018), pp. 139–153. DOI: 10.1016/j.rse.2018.06.028. URL: https://doi.org/10.1016%2Fj.rse.2018.06.028.

[68] Diederik P. Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *ICLR* (2014). 3rd International Conference for Learning Representations, San Diego, 2015.

[69] Emily S Klein et al. “Impacts of rising sea temperature on krill increase risks for predators in the Scotia Sea”. In: *PloS one* 13.1 (2018), e0191011.

[70] Upama A Koju et al. “A two-scale approach for estimating forest aboveground biomass with optical remote sensing images in a subtropical forest of Nepal”. In: *Journal of Forestry Research* 30.6 (2019), pp. 2119–2136.

[71] Sara Labrousse et al. “Dynamic fine-scale sea icescape shapes adult emperor penguin foraging habitat in East Antarctica”. In: *Geophysical Research Letters* 46.20 (2019), pp. 11206–11218.

[72] SE Lake, HR Burton, and MA Hindell. “Influence of time of day and month on Weddell seal haul-out patterns at the Vestfold Hills, Antarctica”. In: *Polar biology* 18.5 (1997), pp. 319–324.

[73] MA LaRue et al. “A method for estimating colony sizes of Adélie penguins using remote sensing imagery”. In: *Polar Biology* 37.4 (2014), pp. 507–517.

[74] Michelle LaRue et al. “High-resolution satellite imagery meets the challenge of monitoring remote marine protected areas in the Antarctic and beyond”. In: *Conservation Letters* (2022), e12884.

[75] Michelle LaRue et al. “Insights from the first global population estimate of Weddell seals in Antarctica”. In: *Science Advances* 7.39 (2021), eabh3674.

[76] Michelle A LaRue et al. “Engaging ‘the crowd’in remote sensing to learn about habitat affinity of the Weddell seal inhttps://www.overleaf.com/project/632d97c1916aa3e70 Antarctica”. In: *Remote Sensing in Ecology and Conservation* ().

[77] Michelle A LaRue et al. “Satellite imagery can be used to detect variation in abundance of Weddell seals (*Leptonychotes weddellii*) in Erebus Bay, Antarctica”. In: *Polar Biology* 34.11 (2011), p. 1727.

[78] Jasmine R. Lee et al. “Climate change drives expansion of Antarctic ice-free habitat”. In: *Nature* 547.7661 (2017), pp. 49–54.

[79] Woo-Young Lee, Seung-Min Park, and Kwee-Bo Sim. “Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm”. In: *Optik* 172 (2018), pp. 359–367.

[80] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *CoRR* abs/1708.02002 (2017). arXiv: 1708.02002. URL: http://arxiv.org/abs/1708.02002.

[81] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).

[82] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[83] Ilya Loshchilov and Frank Hutter. “Sgdr: Stochastic gradient descent with warm restarts”. In: *arXiv preprint arXiv:1608.03983* (2016).

[84] Heather J Lynch et al. “Detection, differentiation, and abundance estimation of penguin species by high-resolution satellite imagery”. In: *Polar Biology* 35.6 (2012), pp. 963–968.

[85] Hieu M Le et al. “Weakly labeling the antarctic: The penguin colony case”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 18–25.

[86] Lei Ma et al. “Deep learning in remote sensing applications: A meta-analysis and review”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 152 (2019), pp. 166–177. ISSN: 0924-2716.

[87] Sébastien Marcel and Yann Rodriguez. “Torchvision the Machine-Vision Package of Torch”. In: *Proceedings of the 18th ACM International Conference on Multimedia*. MM ’10. New York, NY, USA: Association for Computing Machinery,

2010, 1485–1488. ISBN: 9781605589336. DOI: 10.1145/1873951.1874254. URL: https://doi.org/10.1145/1873951.1874254.

[88] Daniela Pawelski Amaro Marins et al. "Dealing With Challenges Developing a Gateway to the "End of the World"—The Brazilian Antarctic Station Case Study". In: *IEEE Internet of Things Journal* 9.16 (2022), pp. 15161–15168.

[89] Robert A. Massom and Sharon E. Stammerjohn. "Antarctic sea ice change and variability–physical and ecological implications". In: *Polar Science* 4.2 (2010), pp. 149–186.

[90] Kenichi Matsuoka et al. "Quantarctica, an integrated mapping environment for Antarctica, the Southern Ocean, and sub-Antarctic islands". In: *Environmental Modelling & Software* 140 (2021), p. 105015. ISSN: 1364-8152. DOI: https://doi.org/10.1016/j.envsoft.2021.105015. URL: https://www.sciencedirect.com/science/article/pii/S136481522100058X.

[91] Brett T McClintock et al. "Quantitative assessment of species identification in aerial transect surveys for ice-associated seals". In: *Marine Mammal Science* 31.3 (2015), pp. 1057–1076.

[92] Clive R McMahon et al. "Satellites, the all-seeing eyes in the sky: counting elephant seals from space". In: *PloS one* 9.3 (2014), e92613.

[93] Robert W McNabb et al. "Quantification and analysis of icebergs in a tidewater glacier fjord using an object-based approach". In: *Plos One* 11.11 (2016), e0164444.

[94] Paulius Micikevicius et al. "Mixed Precision Training". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=r1gs9JgRZ.

[95] James Millan and Jungyong Wang. "Ice force modeling for DP control systems". In: *Proceedings of the Dynamic Positioning Conference*. 2011.

[96] David A Miller et al. "Improving occupancy estimation when two types of observational error occur: Non-detection and species misidentification". In: *Ecology* 92.7 (2011), pp. 1422–1428.

[97] David AW Miller et al. "Determining occurrence dynamics when false positives occur: estimating the range dynamics of wolves from public survey data". In: *PLoS one* 8.6 (2013), e65808.

[98] Osama Mustafa et al. "Detecting Antarctic seals and flying seabirds by UAV". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2019), pp. 141–148.

[99] Dominik A Nachtsheim et al. “Habitat modelling of crabeater seals (Lobodon carcinophaga) in the Weddell Sea using the multivariate approach Maxent”. In: *Polar Biology* 40.5 (2017), pp. 961–976.

[100] Stephen Nicol, Jacqueline Foster, and So Kawaguchi. “The fishery for Antarctic krill–recent developments”. In: *Fish and Fisheries* 13.1 (2012), pp. 30–40.

[101] Mohammed Sadegh Norouzzadeh et al. “Automatically identifying wild animals in camera trap images with deep learning”. In: *Proceedings of the National Academy of Sciences*. Vol. 115.

[102] Douglas P Nowacek et al. “Super-aggregations of krill and humpback whales in Wilhelmina Bay, Antarctic Peninsula”. In: *PLoS One* 6.4 (2011), e19173.

[103] Torger øritsland. “Biology and population dynamics of Antarctic seals”. In: ed. by M.W. Holgate. Academic Press, 1970, pp. 361–366.

[104] Ronald O’Rourke. “Icebreaker Acquisition and the Need for a National Maritime Strategy”. In: (2018).

[105] Emanuela Paladini et al. “Two Ensemble-CNN Approaches for Colorectal Cancer Tissue Type Classification”. In: *Journal of Imaging* 7.3 (2021). ISSN: 2313-433X. DOI: 10.3390/jimaging7030051. URL: https://www.mdpi.com/2313-433X/7/3/51.

[106] Nabil Panchi, Ekaterina Kim, and Anirban 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”. In: *IEEE Sensors Journal* (2021).

[107] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035.

[108] Joseph Paul Cohen et al. “Count-ception: Counting by fully convolutional redundant counting”. In: *Proceedings of the IEEE International conference on computer vision workshops*. 2017, pp. 18–26.

[109] G. Peng et al. “A long-term and reproducible passive microwave sea ice concentration data record for climate studies and monitoring”. In: *Earth System Science Data* 5.2 (2013), pp. 311–318.

[110] Rajeev Pillay et al. “Accounting for false positives improves estimates of occupancy from key informant interviews”. In: *Diversity and Distributions* 20.2 (2014), pp. 223–235.

[111] Rishav Pramanik et al. “A fuzzy distance-based ensemble of deep models for cervical cancer detection”. In: *Computer Methods and Programs in Biomedicine* 219 (2022), p. 106776. ISSN: 0169-2607. DOI: https://doi.org/10.1016/

j.cmpb.2022.106776. URL: https://www.sciencedirect.com/science/article/pii/S0169260722001626.

[112] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[113] Christian S Reiss et al. "Overwinter habitat selection by Antarctic krill under varying sea-ice conditions: implications for top predators and fishery management". In: *Marine Ecology Progress Series* 568 (2017), pp. 1–16.

[114] Tracey L Rogers. "Age-related differences in the acoustic characteristics of male leopard seals, Hydrurga leptonyx". In: *The Journal of the Acoustical Society of America* 122.1 (2007), pp. 596–605.

[115] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional networks for biomedical image segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597.

[116] Olga Russakovsky et al. "ImageNet large scale visual recognition challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252.

[117] Leo A Salas et al. "Reducing error and increasing reliability of wildlife counts from citizen science surveys: counting Weddell Seals in the Ross Sea from satellite images". In: *bioRxiv* (2020).

[118] Arnt-Børre Salberg. "Detection of seals in remote sensing images using features extracted from deep convolutional neural networks". In: *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2015, pp. 1893–1896.

[119] Andrei Sandru et al. "A complete process for shipborne sea-ice field analysis using machine vision". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 14539–14545.

[120] Claudio Filipi Gonçalves Dos Santos and João Paulo Papa. "Avoiding overfitting: A survey on regularization methods for convolutional neural networks". In: *ACM Computing Surveys (CSUR)* 54.10s (2022), pp. 1–25.

[121] D Blake Sasse. "Job-related mortality of wildlife workers in the United States, 1937-2000". In: *Wildlife society bulletin* (2003), pp. 1015–1020.

[122] Divya Shanmugam et al. "When and why test-time augmentation works". In: *arXiv preprint arXiv:2011.11156* (2020).

[123] Lloyd S. Shapley. *A Value for N-Person Games*. Santa Monica, CA: RAND Corporation, 1952. DOI: 10.7249/P0295.

[124] Krishna Kumar Singh et al. *Hide-and-Seek: A Data Augmentation Technique for Weakly-Supervised Localization and Beyond*. 2018. DOI: 10.48550/ARXIV.1811.02545. URL: https://arxiv.org/abs/1811.02545.

[125] Walker O Smith and David M Nelson. "Phytoplankton bloom produced by a receding ice edge in the Ross Sea: spatial coherence with the density field". In: *Science* 227.4683 (1985), pp. 163–166.

[126] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical Bayesian Optimization of Machine Learning Algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

[127] C. Southwell et al. "A review of data on abundance, trends in abundance, habitat use and diet of ice-breeding seals in the Southern Ocean". In: *CCAMLR Science* 19 (2012), pp. 49–74.

[128] Colin Southwell. "Satellite-linked dive recorders provide insights into the reproductive strategies of crabeater seals (Lobodon carcinophagus)". In: *Journal of Zoology* 264.4 (2004), pp. 399–402.

[129] Colin Southwell et al. "Taking account of dependent species in management of the Southern Ocean krill fishery: estimating crabeater seal abundance off east Antarctica". In: *Journal of Applied Ecology* 45 (2 2008), pp. 622–631.

[130] Colin Southwell et al. "Uncommon or cryptic? Challenges in estimating leopard seal abundance by conventional but state-of-the-art methods". In: *Deep Sea Research Part I: Oceanographic Research Papers* 55.4 (2008), pp. 519–531.

[131] Colin J Southwell et al. "Estimating population status under conditions of uncertainty: the Ross seal in East Antarctica". In: *Antarctic Science* 20.2 (2008), pp. 123–133.

[132] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[133] Seth Stapleton et al. "Polar bears from space: assessing satellite imagery as a tool to track Arctic wildlife". In: *PLoS One* 9.7 (2014), e101513.

[134] Mingxing Tan and Quoc Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.

[135] David N Thomas and Gerhard S Dieckmann. *Sea ice: An introduction to its physics, chemistry, biology and geology*. John Wiley & Sons, 2008.

[136] Hongkun Tian et al. "Computer vision technology in agricultural automation—A review". In: *Information Processing in Agriculture* 7.1 (2020), pp. 1–19.

[137] Philip N Trathan and Simeon L Hill. "The importance of krill predation in the Southern Ocean". In: *Biology and ecology of Antarctic krill*. Springer, 2016, pp. 321–350.

[138] NM Voronina. "Comparative abundance and distribution of major filter-feeders in the Antarctic pelagic zone". In: *Journal of Marine Systems* 17.1-4 (1998), pp. 375–390.

[139] Athanasios Voulodimos et al. "Deep learning for computer vision: A brief review". In: *Computational intelligence and neuroscience* 2018 (2018).

[140] Xin Wang et al. *Frustratingly Simple Few-Shot Object Detection*. 2020. arXiv: 2003.06957 [cs.CV].

[141] Ben G Weinstein. "A computer vision for animal ecology". In: *Journal of Animal Ecology* 87.3 (2018), pp. 533–545.

[142] Marijke Welvaert and Peter Caley. "Citizen surveillance for environmental monitoring: combining the efforts of citizen science and crowdsourcing in a quantitative data framework". In: *SpringerPlus* 5.1 (2016), pp. 1–14.

[143] John Wiedenmann, Katherine A Cresswell, and Marc Mangel. "Connecting recruitment of Antarctic krill and sea ice". In: *Limnology and Oceanography* 54.3 (2009), pp. 799–811.

[144] Michael Winton. "Sea ice-albedo feedback and nonlinear Arctic climate change". In: *Arctic Sea Ice Decline: Observations, Projections, Mechanisms, and Implications*. Ed. by E.T. DeWeaver, C.M. Bitz, and L.-B. Tremblay. Vol. 180. AGU, Washington, D.C., 2008, pp. 111–131.

[145] Nicholas C Wright and Chris M Polashenski. "Open-source algorithm for detecting sea ice surface features in high-resolution optical imagery". In: *The Cryosphere* 12.4 (2018), pp. 1307–1329.

[146] Shuyuan Xu et al. "Computer vision techniques in construction: a critical review". In: *Archives of Computational Methods in Engineering* 28.5 (2021), pp. 3383–3397.

[147] Yifei Xue, Tiejun Wang, and Andrew K Skidmore. "Automatic counting of large mammals from very high resolution panchromatic satellite imagery". In: *Remote sensing* 9.9 (2017), p. 878.

[148] Yifei Xue, Tiejun Wang, and Andrew K. Skidmore. "Automatic Counting of Large Mammals from Very High Resolution Panchromatic Satellite Imagery". In: *Remote Sensing* 9.9 (2017). ISSN: 2072-4292. DOI: 10.3390/rs9090878. URL: https://www.mdpi.com/2072-4292/9/9/878.

[149] Pavel Yakubovskiy. *Segmentation Models Pytorch*. https://github.com/qubvel/segmentation_models.pytorch. 2020.

[150] Zheng Yang et al. “Spotting East African mammals in open savannah from space”. In: *PloS one* 9.12 (2014), e115989.

[151] Jason Yosinski et al. “How transferable are features in deep neural networks?” In: *arXiv preprint arXiv:1411.1792* (2014).

[152] Sergey Zagoruyko and Nikos Komodakis. “Wide residual networks”. In: *arXiv preprint arXiv:1605.07146* (2016).

[153] Fabio Massimo Zanzotto. “Human-in-the-loop artificial intelligence”. In: *Journal of Artificial Intelligence Research* 64 (2019), pp. 243–252.

[154] Qin Zhang and Roger Skjetne. “Image processing for identification of sea-ice floes and the floe size distributions”. In: *IEEE Transactions on Geoscience and Remote Sensing* 53.5 (2014), pp. 2913–2924.

[155] Qin Zhang, Roger Skjetne, and Biao Su. “Automatic image segmentation for boundary detection of apparently connected sea-ice floes”. In: *The proceedings of the 22nd International Conference on Port and Ocean Engineering under Arctic Conditions*. Port and Ocean Engineering under Arctic Conditions. 2013.

[156] Qin Zhang et al. “Image processing for ice floe analyses in broken-ice model testing”. In: *Cold Regions Science and Technology* 111 (2015), pp. 27–38.

[157] Anna Zmarz et al. “Application of UAV BVLOS remote sensing data for multi-faceted analysis of Antarctic ecosystem”. In: *Remote Sensing of Environment* 217 (2018), pp. 375–388.

[158] Hui Zou and Trevor Hastie. “Regularization and Variable Selection via the Elastic Net”. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67.2 (2005), pp. 301–320. ISSN: 13697412, 14679868. URL: `http://www.jstor.org/stable/3647580` (visited on 07/17/2022).