The Version of Record of this manuscript has been published and is available in Canadian Journal of Remote Sensing; March 29, 2021; https://www.tandfonline.com/doi/10.1080/07038992.2021.1901221

Peter Q. Lee [a]*ǂ, Keerthijan Radhakrishnan [a]ǂ, David A. Clausi [a], K. Andrea Scott [a], Linlin Xu [a], Marianne Marcoux [b]

[a]*Department of Systems Design Engineering, University of Waterloo, Waterloo, Canada;* [b]*Fisheries and Oceans Canada, Winnipeg,Canada*

*Corresponding author: pqjlee@uwaterloo.ca

ǂ Authors contributed equally

Beluga Whale Detection in the Cumberland Sound Bay using Convolutional Neural Networks

Détection des bélugas dans le détroit Cumberland Sound à l'aide de réseaux de neurones à convolution

Abstract

The Cumberland Sound Beluga is a threatened population of belugas and the assessment of the population is done by a manual review of aerial surveys. The time-consuming and labour-intensive nature of this job motivates the need for a computer automated process to monitor beluga populations. In this paper, we investigate convolutional neural networks to detect whether a section of an aerial survey image contains a beluga. We use data from the 2014 and 2017 aerial surveys of the Cumberland Sound, conducted by the Fisheries and Oceans Canada to simulate two scenarios: 1) when one annotates part of a survey and uses it to train a pipeline to annotate the remainder and 2) when one uses annotations from a survey to train a pipeline to annotate another survey from another time period. We experimented with a number of different architectures and found that an ensemble of 10 CNN models that leverage Squeeze-Excitation and Residual blocks performed best. We evaluated scenarios 1) and 2) by training on the 2014 and 2017 surveys respectively. In both scenarios, the performance on 1) is higher than 2) due to the uncontrolled variables in the scenes, such as weather and surface conditions.

RÉSUMÉ

Le béluga de Cumberland Sound est une population de bélugas menacée et l'évaluation de la population se fait en partie par un examen manuel d'image de relevés aériens. La nature de cette tâche, qui prend du temps et demande beaucoup de travail, motive la nécessité d'un processus informatique automatisé pour surveiller les populations de bélugas. Dans cet article, nous étudions les réseaux de neurones à convolution (RNC) afin de détecter si une section d'une image de relevé aérien contient un béluga. Nous utilisons les données des relevés aériens de 2014 et 2017 du détroit Cumberland Sound, réalisés par le ministère Pêches et Océans Canada, pour simuler deux scénarios : 1) lorsque l'on annote une partie d'un relevé et qu'on l'utilise pour entraîner un pipeline à annoter le reste et 2) lorsque l'on utilise les annotations d'un relevé pour entraîner un pipeline à annoter un autre relevé d'une autre période. Nous avons expérimenté plusieurs architectures différentes et avons constaté qu'un ensemble de 10 modèles RNC qui exploitent des blocs « Squeeze-and-Excitation » ainsi que des blocs résiduels donnent les meilleurs résultats. Nous avons évalué les scénarios 1) et 2) par un entraînement sur les survols aériens de 2014 et 2017. Dans les deux scénarios, la performance de 1) est supérieure à 2) en raison des variables non contrôlées dans les scènes, telles que la météo et les conditions de surface.

Introduction

The Cumberland Sound (CS) beluga is a distinct population that stays in Cumberland Sound, Nunavut, year-round (de March et al. 2004, Richard and Stewart 2009). Male and female CS belugas measure an average length of 3.6 m and 4.3 m, respectively, and have a mass from 800 to 1000 kg (Brodie 1971). Currently, the CS beluga is listed as threatened under the Species at Risk Act (SARA) in Canada (COSEWIC 2004) and a recovery strategy is being developed by the Minister of Fisheries and Oceans Canada (DFO) who is the competent minister for the recovery of the CS Beluga (DFO Unpublished). The DFO performs aerial surveys of the

Cumberland Sound Bay during late July and August in order to obtain population counts of the CS belugas, which are instrumental to providing advice on sustainable hunt for this population (Marcoux and Hammill 2016).

Population counts are obtained from a manual review process of the aerial images. This process is not ideal due to the required time and labour. Therefore, the creation of a computer automated prediction model to detect belugas would be beneficial. Designing such a method is a challenging task due to the few belugas in the images relative to open water areas and the heterogeneous appearance of the images from varying water depth, water surface patterns, and solar illumination. Solar illumination and weather conditions can also create wave patterns that resemble belugas.

The objectives of this study are to:

- Develop an automated model to identify belugas in aerial images to assist in population counts.
- Propose a numerical score, called PTPVSA, to gauge the usefulness of a whale detection pipeline for a human annotator.
- Develop techniques when constructing and training automated models that can overcome the challenge of the heterogenous appearance of the images in the dataset.

We propose a deep learning model, specifically a convolutional neural network, that strides through aerial images and extracts patches that are 65 x 65 pixels and assigns the probability of a whale being in that patch. We first define a baseline model of random forest as a benchmark for comparison. Then deep learning architectures, ensemble models, and data augmentation are explored. Using images from two separate surveys taken during 2014 and 2017, we consider two scenarios: one where a practitioner labels a subset of scenes in a survey and uses the proposed pipeline to aid in labelling remaining scenes, and another where the

practitioner uses labelled data from a previous year to automatically aid in labelling a new data set. For the former scenario, we divided the 2014 survey into two partitions such that the first partition is used to train the pipeline and aid in labelling the second partition. For the second scenario, we used the first partition from the former scenario to train the pipeline and apply it towards labelling the entirety of the 2017 survey. We also train our model with the 2017 survey and evaluate with the 2014 survey. Overall, we found that deep learning models with more advanced architectures, ensemble methods, and augmentation were beneficial. We also note that there is a performance gap between the first and second scenarios, indicating that the environmental changes between the two time periods are significant enough to affect performance.

Related Work

Prediction of objects within aerial images is a standard task within remote sensing. Applications of this range from sensing man-made objects or natural features, fauna, or flora. In terms of identifying whales, Fretwell et al. (2014) utilized the ArcGIS framework to identify southern right whales using WorldView2 satellite images. Maire et al. (2013) studied the application of using unmanned aerial vehicles for detecting marine mammals (dugongs). They noticed that the shape and colour of the dugongs varied with respect to the depth and turbidity of the water. They also found that solar glare increases the difficulty of detecting dugongs. Maire et al. (2013) leveraged the fact that dugongs are rare in the image and developed regions of interest based on colour rarity. This approach would not work for our problem because of the variations in tone for the different image scenes and whales. Maire et al. (2015) had very similar objectives to our own work as they also leveraged deep learning to identify dugongs. They used a Simple Linear Iterative Clustering (SLIC) region proposal algorithm followed by a deep convolutional model. Using this model, they achieved a recall of 80% with a precision of 27%. The dataset used for their problem consisted of images taken at lower altitudes. The dugongs have much different

scale in their images than the belugas from our images, which make it more difficult to identify features of belugas in our images.

As the availability of unmanned aerial vehicles (UAV) have increased, so too have their applications towards counting and surveying animals. Kellenberger et al. (2018) investigated the benefits of applying deep learning to identify regions with mammals within the African savanna for screening images captured by UAVs. Although similar goals existed for their study, the data set is very different. Belugas occupy a smaller number of pixels than animals in the African savanna. The visibility of belugas is also affected by factors, such as water depth, and they can easily become obstructed by water ripples and glare. A key similarity between the two tasks is the threat of false positives, which were very common in the animal detection in the African Savanna due to other objects in the scene. The same problem exists for beluga detection due to changing surface conditions of the water. Overall, Kellenberger et al. (2018) found that deep learning provided a good avenue for screening potential regions of the images. In the work by Barbedo et al. (2020), various deep learning models were evaluated in order to count cattle in UAV surveys. The authors adapted a tiling strategy that subdivided the large image into smaller subimages. Overall, the authors found that a large selection of off-the-shelf CNNs could be trained to achieve high prediction performance. Guirado et al. (2019) also investigated whale detection using satellite and aerial images with deep learning. They used Faster-RCNN based on RenNet v2. They chose this architecture based on performance in other object detection problems.

In object detection challenges, one will typically have to grapple with two challenges: predicting the size or bounding-box of an object and predicting the class of an object. In terms of deep learning, such tasks can be implemented with what are called region-proposal networks (Ren et al. 2015; Girshick 2015; Girshick et al. 2014; Redmon et al. 2016). While some authors utilized region-proposal networks (Kellenberger et al. 2018; Andrew et al. 2017; Guirado et al. 2019), Barbedo et al. (2020) avoided these architectures due to their difficulty in recognizing

"small groups of objects and to generalize aspect ratios". It is worth mentioning that the problem of object detection specifically for remote sensing images is still a very active area of research, with many novel approaches being developed to tackle the challenges of object detection in remote sensing data. In the work by Ghorbanzadeh et al. (2020) the issue of detecting refugee dwellings from high resolution satellites was tackled using an object-based image analysis approach along with convolutional networks. The work by Courtrai et al. (2020) examined the utility of a super-resolution pipeline for detecting small objects by way of generative adversarial networks.

Data

The data used in this study consists of images taken from aerial surveys of the Cumberland Sound Bay, Nunavut during the summer observation periods of 2014 and 2017 during late July and early August. Of the total images taken, only those within the area of the Clearwater Fiord, as shown in Figure 1, were used for this study. This region is the major congregation point of Cumberland Sound belugas during summer months (Richard and Stewart 2009).

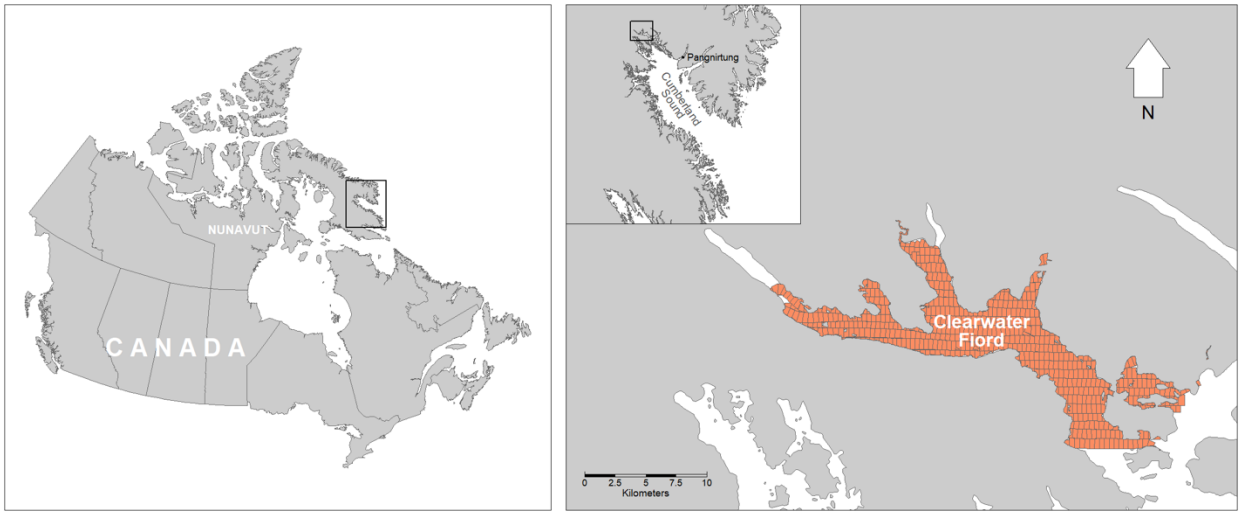


Figure 1: Map of the study area in Clearwater Fiord within Cumberland Sound, Nunavut, Canada. Area in orange shows the individual outlines of the photos taken on August 7, 2017.

The aerial geo-coded images were acquired from a Nikon D810 camera equipped with a 25 mm lens and flown at height of 610 m at a speed of 204 km/h with each image containing 4912 × 7360 pixels, representing an approximate pixel spacing of 10 cm. During flight, the camera continuously acquired images with an interval of about 4 seconds. Images outside the study area were excluded. In addition, images of scenes that captured areas entirely above sea level were excluded through registration with the Canadian Digital Elevation Model (CDEM), since whales are sea dwelling creatures. This inclusion criteria resulted in a total of 3888 images in the 2014 dataset and 2712 images in the 2017 dataset. Since the two datasets were taken from different time periods, the weather conditions were different. In Table 1, we compare the weather conditions that occurred during the survey periods in each day of the 2014 and 2017 dataset. From this comparison, we can see that the scenes from the 2014 dataset have more clear conditions while the scenes from the 2017 dataset have more overcast conditions.

Table 1: Weather during aerial surveys. The most prevalent weather condition each day was declared as the weather for the day. Data acquired from Pangnirtung A, Nunavut weather station (Government of Canada, 2019).

Dataset	**Date**	**Weather**	**Wind Speed (km/h)**
2014	Aug 3, 2014	Clear	8.95
	Aug 4, 2014	Clear	12.04
	Aug 10, 2014	Clear	18.96
	Aug 11, 2014	Cloudy	24.04
2017	Jul 29, 2017	Foggy	7.17
	Aug 7, 2017	Cloudy	7.71
	Aug 8, 2017	Clear	11.41
	Aug 12, 2017	Cloudy	8.33

We partitioned the data into two separate scenarios, scenario A where the model was trained with data from 2014 and scenario B where the model was trained with data from 2017. Each scenario splits the data into three mutually exclusive groups:

Scenario A: *Train-A 2014*, *Eval-A 2014*, and *Eval-A 2017*

Scenario B: Train-B 2017, *Eval-B 2014*, *Eval-B 2017*.

A breakdown of the datasets is shown in Table 2.

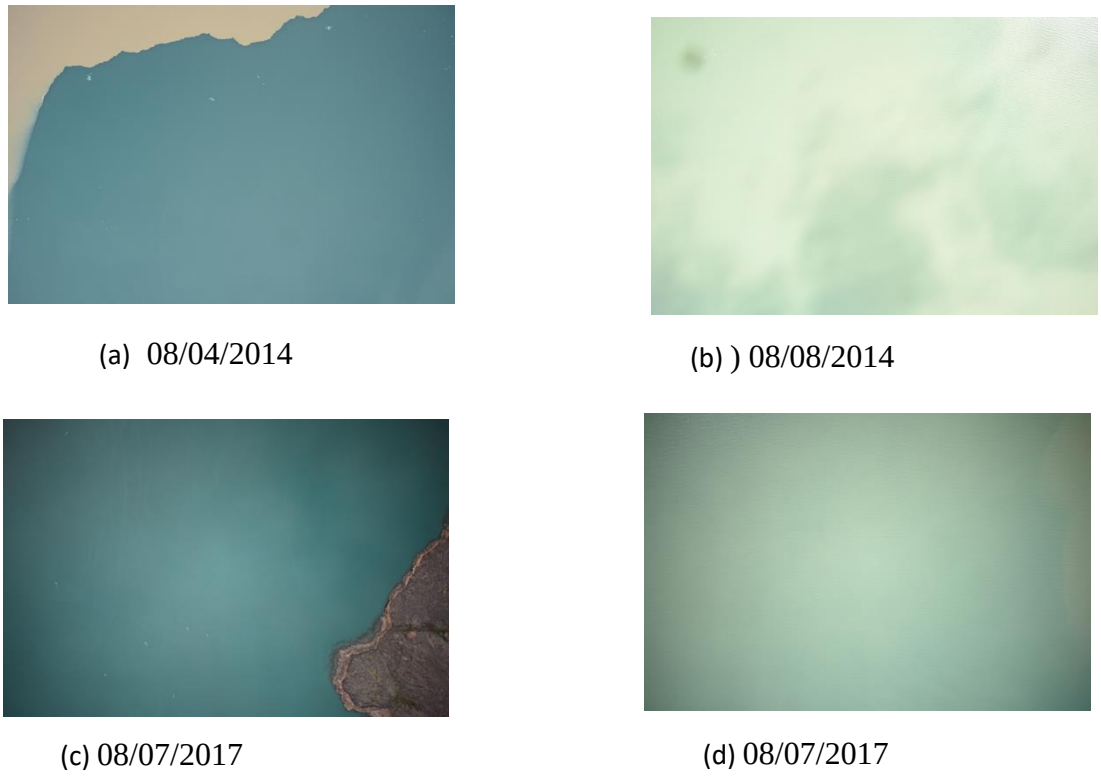
Table 2: Breakdown of beluga findings by category. Note that the numbers do not account for duplicate counts of belugas that appear from overlapping images.

	2014 Dataset		**2017 Dataset**	
Group	*Train-A 2014*	*Eval-A 2014*	*Eval-A 2017*	
Total number of belugas	1190	447	3691	
Total number of images	373	3515	2712	
Total number of images without belugas	92	3444	2482	
	Eval-B 2014		*Train-B 2017*	*Eval-B 2017*
Total number of belugas	1637		2781	910
Total number of images	3888		186	2526
Total number of images without belugas	3536		3	2479

Data Challenges

Training a deep learning model to predict whales on aerial surveys is a challenging task due to the heterogeneous appearance of the images and relative size of belugas.

The appearance of the background in the images varies significantly throughout the dataset because the appearance of the water depends on environmental variables such as the time of day, wind speed, weather conditions, sediment, and viewing angle. Figure 2 shows some examples of the heterogeneous backgrounds in images within the study zone. These conditions cause difficulties from a classification perspective because the appearance of a beluga with respect to the background will vary based on these factors. Exposure or compensation for different tones of water are thus required to build an effective classifier. Also, regions with high amounts of specular reflection (i.e., glare) from the sun greatly reduce visibility. Consequently, these factors place the onus on the classification model to have enough complexity or capability to handle these variations. As will be shown in the following sections, regions with high glare were a significant source of false positives created by the classifiers.



(a) 08/04/2014 (b)) 08/08/2014

(c) 08/07/2017 (d) 08/07/2017

Figure 2: Images of heterogenous appearance due to solar illumination. The coverage for each image was roughly 900 m and 600 m in terms of image width and height respectively.

The size of belugas is another challenge when working with aerial surveys. While dependent on the physical size of the beluga, the aircraft altitude, and viewing angle, adult belugas have a length of approximately 30 to 40 pixels, which make them small components of the image. The younger belugas are smaller and more difficult to identify. The combination of the heterogeneous backgrounds, different appearances of the belugas, and the small size of belugas make it difficult to identify belugas within an image.

Methodology

The backbone of our pipeline is a machine learning classifier that is capable of mapping a subimage to a desired label: whale or non-whale. The pipeline is designed to produce likelihood predictions for overlapping patches of the original images to aid a human operator in recognizing regions that contain whales. We consider a baseline random forest (RF) model, a base convolutional neural network (CNN) model, and enhanced CNN models with advanced deep learning techniques. In this section we discuss the classifier as well as our patch extraction procedure. A summary of the entire pipeline is visualized in Figure 3.

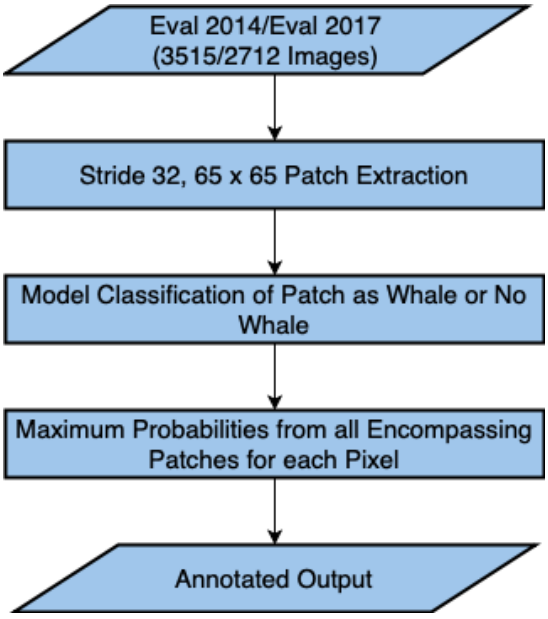


Figure 3: Evaluation pipeline to obtain annotated output from Eval 2014/Eval 2017 scenes.

Patch Extraction

The machine learning models used in this study require uniform patches as input. Patches were obtained using a tiling strategy to break the original 7360 × 4912 pixel input images into overlapping 65 × 65 pixel patches, with the patch size being a tradeoff between having sufficient pixels to capture the beluga and the prediction granularity. The tiling strategy differed for training set (*Train-A 2014* and *Train-B 2017*) and the evaluation sets (*Eval-A/B 2014* and *Eval-A/B 2017*).

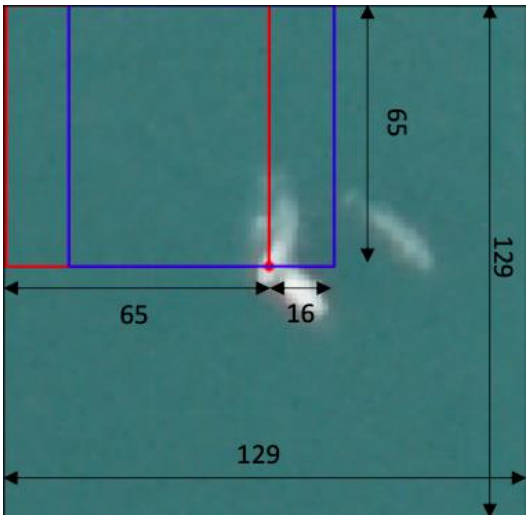


Figure 4: 129 × 129 patch surrounding a whale sighting extracted from image with two sample 65 × 65 sub-patches used for training (red/blue).

We create sets of patches called *TrainPatches-A* and *TrainPatches-B* composed of an equal number of patches labelled as whale and non-whale that are extracted from *Train-A 2014* and *Train-B 2017*. The subimages labelled as whale were created by selecting 129 × 129 large patches surrounding the beluga coordinate, then dividing this into 65 × 65 patches using a stride of 16 in both x and y directions. We do this to ensure that our model is shift invariant. An example of the patch extraction procedure is shown in Figure 4. Next, we obtain patches labelled

as non-whale by taking an equivalent number of 65 × 65 patches, not adjacent to belugas, randomly selected from *Train-A/B*. An additional set, *TrainPatches-A/B + High Glare*, adds onto *TrainPatches-A/B* by selecting patches with high glare, by a process explained in more detail in the Hardmining section. Overall, this extraction process resulted in a total of 35,950 patches containing whales, 35,950 patches that do not contain whales, and 8000 patches from high glare regions away from whales. An overview of the patch extraction pipeline for training is shown in Figure 5.

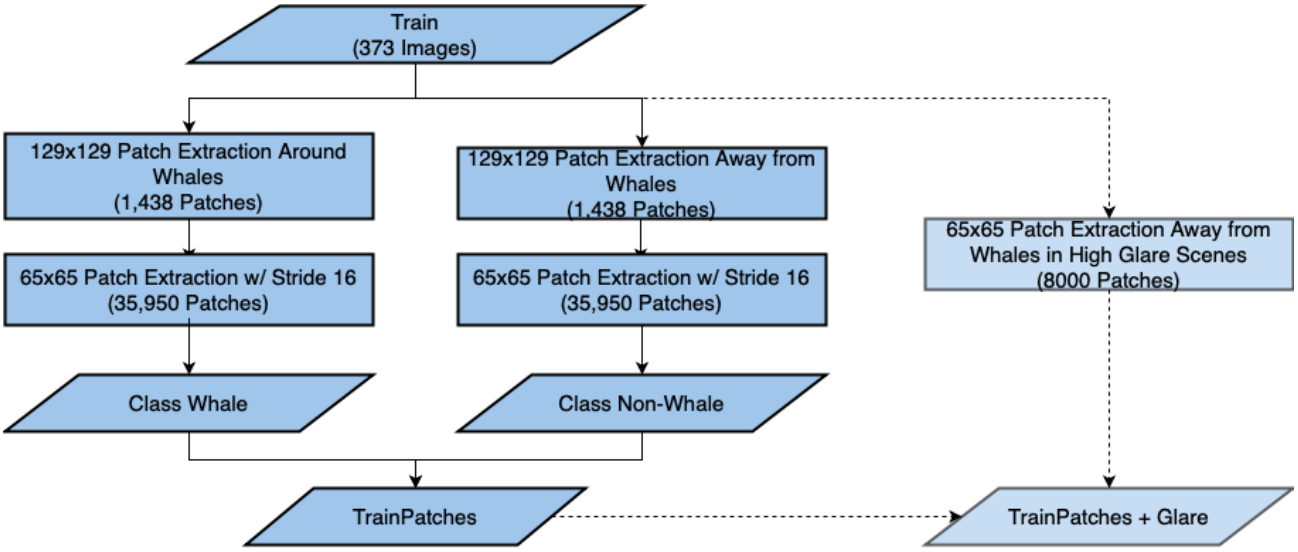


Figure 5: Patch extraction pipeline to obtain *TrainPatches* and *TrainPatches+Glare* from *Train*

For evaluation, *Eval-A/B 2014* and *Eval-A/B 2017*, the tiling strategy was adapted so that it would cover the entire image, using a stride spacing of 32. The stride size was increased to make evaluation on the full image more manageable. These patches are ultimately what are passed into the evaluation pipeline, as previously shown in Figure 3, that uses the trained models to predict labels for the entire images.

Baseline Machine Learning Model

The field of image recognition has proposed many different methods for the problem of classifying images, which vary in performance depending on the problem's characteristics. With the popularity of deep learning, much development has been focused on applying deep learning models, namely CNNs, towards image recognition problems because of their abilities to learn high-level features without requiring to specifically engineer features for the problem. To validate previous studies and confirm that deep learning models are better suited for our problem, we develop a baseline RF classifier.

We choose an RF classifier as the baseline because when compared to other traditional machine learning algorithms, an RF classifier has fewer hyper parameters to tune. Furthermore, previous studies have shown that RF classifiers perform just as well on remote sensing scene classification as other traditional machine learning algorithms (Pelletier et al. 2016; Pal 2005). The inputs to our RF classifier are histograms of oriented gradients (HOG) vectors (Dalal and Triggs 2005), which are transformations of the subimages. In terms of parameters, the HOG vectors were generated with 8 orientations with cells of 8×8, and the RF had 100 trees with Gini impurity. Prior to HOG transformation, a non-linear brightness and contrast adjustment (The Gimp Development Team, 2020) was applied to better discriminate the whales from the darker background. Our interest in this RF classifier is only as a baseline classifier method for comparing deep learning models. While there are likely improvements that could be made from other feature extraction algorithms and traditional machine learning classifiers, it is outside the scope of our work. We implemented HOG with the skimage python package and RF with the scikit-learn Python package (Walt et al. 2014; Pedregosa et al. 2011).

Convolutional Neural Network

Convolutional neural networks (CNNs) are neural networks that consist of convolutional and fully connected layers (LeCun et al. 1995). CNNs perform image classification by assigning a class likelihood to an input image and selecting the class with the highest likelihood score. CNNs

have become the primary choice for image classification tasks because of their ability to learn complex image patterns. The inputs to the CNN are the image channels. For an RGB image, such as the ones used in this study, the image channels are the red, green, and blue channels, which we normalized to be between -0.5 and 0.5. Through a series of convolutional filters, the CNN is able to identify and preserve features of interest. After the convolutional layers, the images are lexicographically stacked to 1-dimensional vectors. These vectors are passed to the fully connected layers, where non-linear functions are applied to linear combinations of previous layers. The output from the fully connected layers is the probability of the existence of a beluga. The parameters in the network are optimized by means of a training algorithm, such as gradient-based algorithms. Often in CNN models, pooling operations are performed to downsample the image. Max pooling is a pooling operation where blocks of pixels are replaced with the pixel with the greatest intensity.

Our proposed CNN architecture for whale detection takes 65 x 65 pixel patches as input and determines the probability of a whale existing in that patch. This is different from the region proposal networks and semantic segmentation models that are popular for object detection tasks.

We decided not to use region-proposal networks, which are commonly applied for object-detection purposes, for the following reasons:

(1) Belugas in the images do not appear as distinct objects in the image. Often, they appear in groups where multiple belugas appear to overlap within the view. In addition, they are similar to water ripples in the image and they are often hidden among the water ripples. This would likely cause regions containing belugas to go undetected, which is unacceptable in our context.

(2) In the mainstream approaches, the regions are typically rescaled to a common size for the classifier to operate. The appearance of belugas is relatively similar in size throughout our dataset, so the added complexity of dynamically generating different sized regions is

unnecessary. Furthermore, such a method would skew the relative scale of the belugas in the original images and consequently destroy the prior information of the common size of belugas.

We decided not to use semantic segmentation models for the following reasons:

(1) Semantic segmentation requires pixel-wise labels of the categories. Our dataset only contained coordinates of whale findings and converting these to pixel-wise maps is a non-trivial task.
(2) Our evaluation of this task is based on the number of whales that are successfully detected. As converting pixel-wise predictions to a discrete count of whales is non-trivial, we determined it more straightforward to do a patch-wise classification approach.

Most of the state-of-the-art models for computer vision are built-for and tested on popular benchmark datasets, such as Canadian Institute for Advanced Research 10 (CIFAR-10) (Krizhevsky, 2009), ImageNet (Deng et al., 2009), Microsoft Common Objects in Context (MS COCO) (Ling et al., 2014), and many more, that represent natural images encountered by typical multimedia sources (e.g., images appearing from social networks). The visual characteristics of these images differ from those in remote sensing tasks, so it is unestablished that the state-of-the-art models on popular datasets will generalize to this setting.

For this study, convolutional networks of varying complexity were used. First, a custom baseline CNN, hereafter referred to as BaseCNN, was implemented. The CNN architecture, as shown in Figure 6(a), consists of three convolution operations followed by flattening and two fully connected layers. 2x2 max pooling was performed between each convolutional layer to reduce computational cost and reduce the likelihood of overfitting. This architecture was chosen based on computational cost and accuracy on a validation set. Non-linear activation functions were used at the end of each layer. The activation functions for all layers except the last was the

rectified linear unit (ReLU) (Glorot et al. 2011). ReLU activation function returns zero for all values less than zero and performs a linear operation (y = x) for all values greater than or equal to zero. ReLU is a standard activation function for computer vision applications. A softmax activation function was used for the final layer. The softmax function converts outputs from the fully connected layer to class likelihood values that sum to one. From these values, a probability of a beluga occurrence can be determined for each patch.

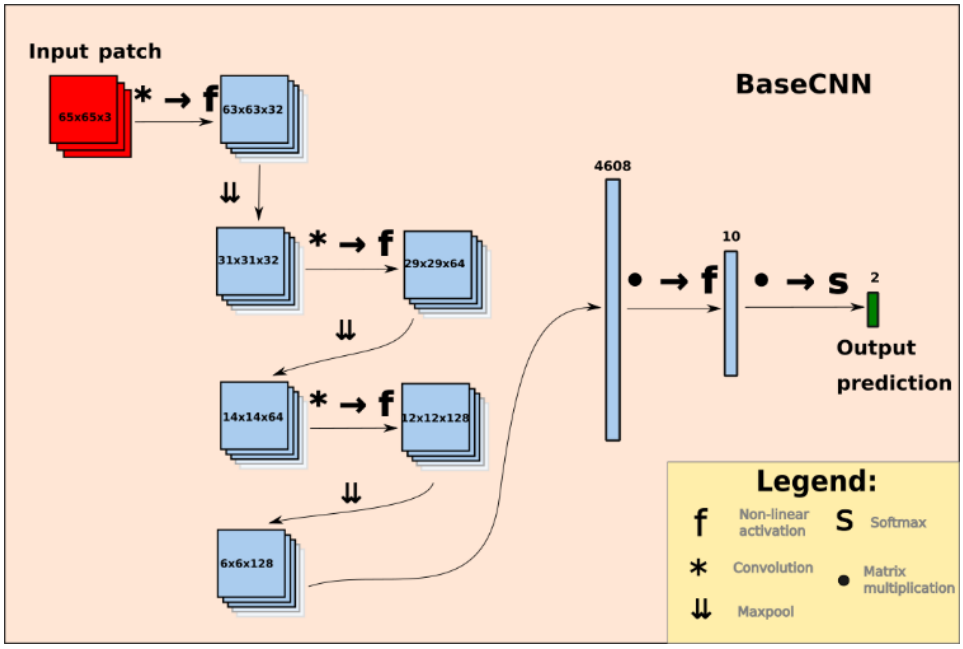
The CNN was trained using a learning rate of 1×10^{-3} with the Adam optimizer (Kingma and Ba 2014). The model was trained for 100 epochs as it was found that the training and validation loss plateaus at this point.

Additional CNN Components

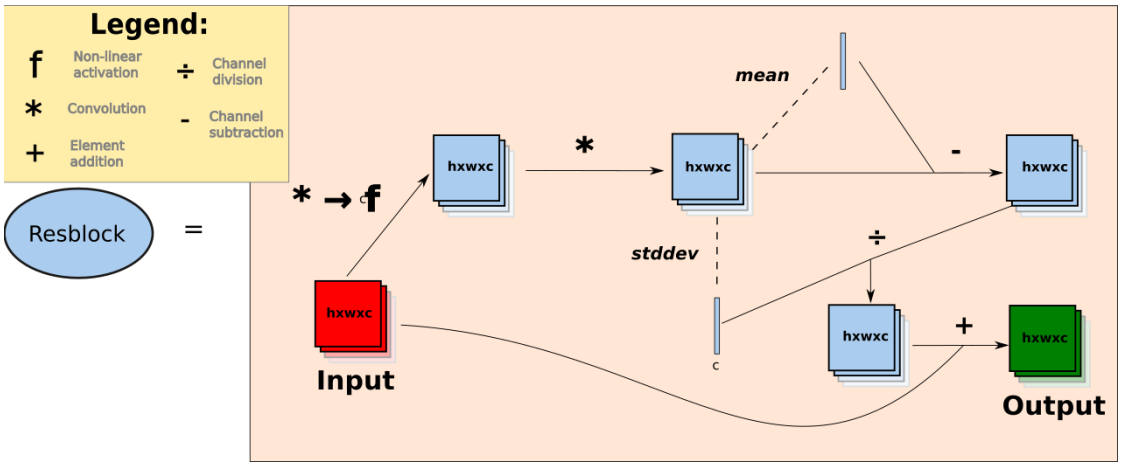
For our study, we looked to the current literature to inspire methods for creating our deep learning architectures. Two architecture components we examined are residual blocks (He et al., 2016) and squeeze-excitation (SE) blocks (Hu et al., 2018). Residual blocks, originally proposed by He et al. (2016), are modules that are leveraged to extend the depth and hence the predictive capacity of the network. Residual blocks, as visualized in Figure 6(b), facilitate this by adding the input after the non-linear activation in order to improve the flow of gradients in the network. Batch normalization is built into the residual block, which normalizes inputs to hidden layers by subtracting a mean and dividing by standard deviation in order to better condition activation outputs during training (Ioffe and Szegedy 2015). SE blocks, proposed by (Hu et al., 2018), are used to model non-local channel interdependency by way of mean reducing the height and width dimensions, taking a trainable linear transformation of the channels (i.e., matrix multiplication), and then multiplying the response back into the hidden layer. An illustration of this model is shown in Figure 6(c). We created two more models that leverage residual blocks and SE blocks respectively.

The models leveraging SE and residual blocks were created in a similar architecture as the BaseCNN to evaluate the effects of the aforementioned blocks upon our task. The first model, hereafter referred to as the RESNET model, adds four additional residual blocks per level, thus making the model much deeper. Residual blocks utilize skip connections to prevent the vanishing gradient problem, which makes typical deep networks difficult to train (He et al. 2016). The second model, hereafter referred to as SE-RESNET, adds SE blocks to the RESNET model. For model training, the CNNs were initialized with random weights and zero valued bias vectors. These models were also optimized using the Adam optimizer with a learning rate of 1×10^{-3} (Kingma and Ba 2014). Both models are visualized in Figure 6(d).

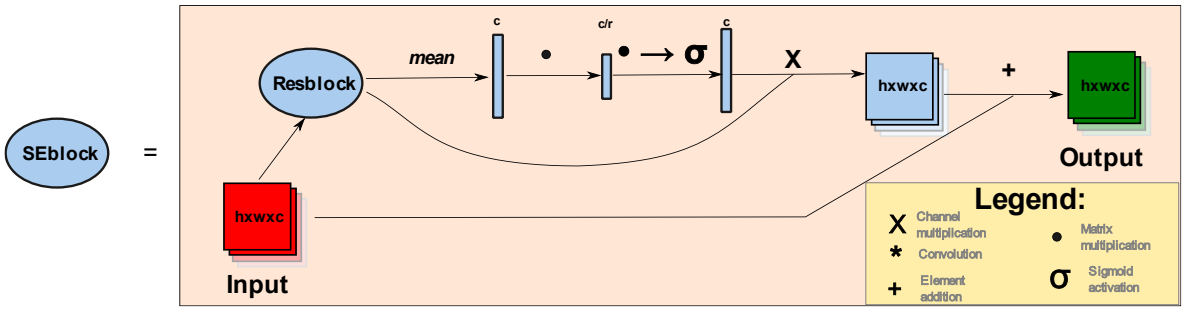
In computer vision problems, there are state-of-the-art models that are retrained for different applications. This is referred to as transfer learning and some of these models leverage the squeeze-excitation and residual blocks. We do not use transfer learning because these state-of-the-art models are built for applications that are not detecting objects from aerial images. They are most often built for classifying images. These images are more complex than images of open water with whales. Therefore, the complexity of these models is unnecessary. Furthermore, the architectural design choices that are beneficial for image classification may not be beneficial for whale detection. Note that we attempted to apply transfer learning and we did not achieve performance superior to our BaseCNN.



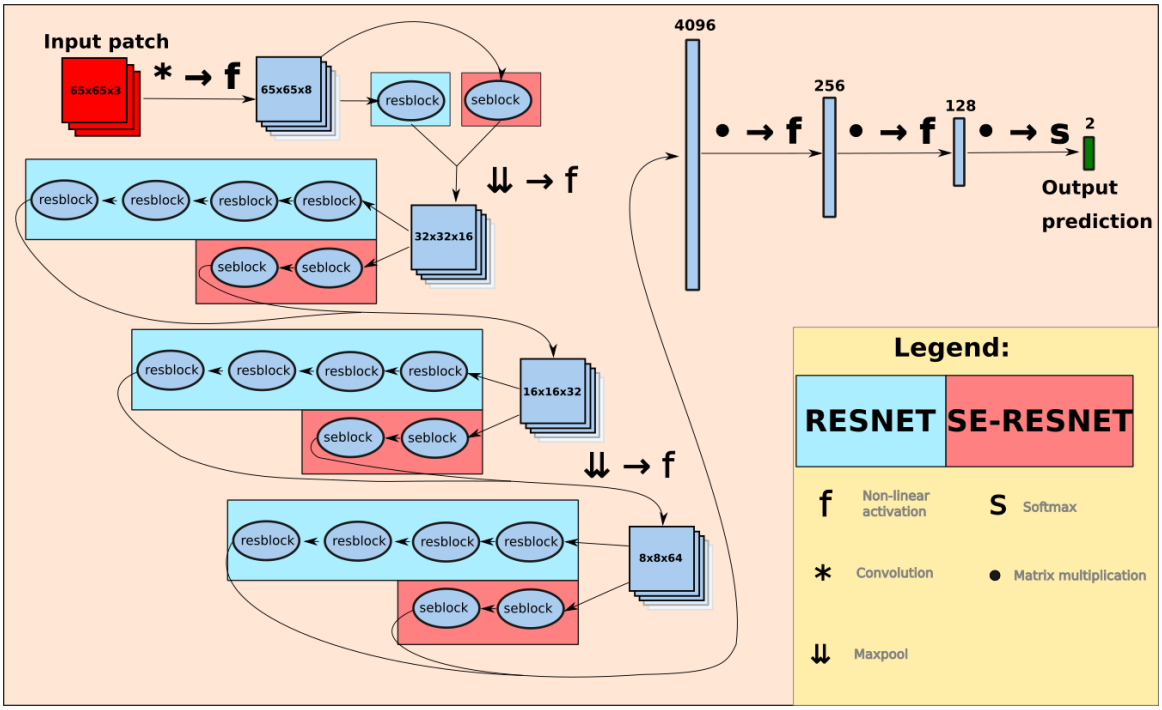
(a) Architecture of BaseCNN model.



(b) Construction of residual block with batch normalization as used in this study. Note that the dotted lines indicate that mean and standard deviation are only actively computed for samples during training. During evaluation, the mean and standard deviation are reused from those encountered during training, as is standard practice when implementing batch normalization.



(c) Construction of squeeze excitation residual blocks, according to Hu et al. (2018)



(d) Architecture of RESNET and SE-RESNET models.

Figure 6: Architectures of CNN models used in the experiments. The **conv** indicates that a convolution is applied, **maxpool** indicates maxpooling is applied (thereby reducing spatial dimensions), **f** indicates a non-linear activation function, **s** indicates a softmax function is applied, and **σ** indicates a sigmoid function is applied.

Ensemble Models

Due to the high variability in the appearance of the images, the performance of a model on certain scene types can be dependent on initialization conditions and the training order of the dataset. To make our model robust to these variations, we propose an ensemble of CNNs. In this paper, ensemble models refer to the creation of multiple separate models that are independently trained. The ensemble's prediction is the aggregate, in our case the mean average, of the predictions of their contained separate models. Ensembles are attractive as they reduce the variability that comes from parameter initialization and batch selection during training (Ju et al. 2018; Nanni et al. 2018). The main drawback is the time required to train several identical models. We chose 10 as a reasonable number of models to be contained in an ensemble. Through experimentation, we found that ensembles with any fewer models did not provide the same level of stability and ensembles with additional models did not provide any additional benefit while increasing computational complexity.

Hardmining

To reduce the false positives in regions of high glare, we propose a "hard-mining" strategy, a method used in the machine learning community that involves modifying a classifier's training set with samples it classified incorrectly. In our implementation of hard-mining, we selected 65 images with a high number of false positives caused by the glare. Specifically, we took images where the proportion of the image predicted as whales by the BaseCNN , which we call the surface area, was over 0.7. From these, a total of 8000 patches of size 65 × 65 were randomly selected from the 65 full images and added to *TrainPatches-A/B* to make a new set *TrainPatches-A/B + Glare*, which are used to train the classification models.

Identification of Regions Containing Belugas

All models discussed in the aforementioned sections output the probability of the presence of a Beluga for 65 by 65 pixel patches of the images, with a stride size of 32 by 32. After these probabilities are calculated, a threshold is determined which is used to identify regions as containing belugas. The output prediction map then takes the union of all patches with probabilities above the threshold. Since some regions will be overlapped by multiple patch predictions due to the stride size, the maximum probability is considered in these regions for thresholding purposes. Then, if a whale appears within a region that is above the threshold, it is considered detected. Likewise, total predicted area is the proportion of all pixels that have a probability above the threshold.

Evaluation

Using the strategy described above, a predicted probability score was generated for each patch using the trained model. Each overlapping patch was assigned a probability estimate by the model that indicates the model's confidence that a beluga was present. Finally, the last element that is required is evaluating the effectiveness of the pipeline using the dataset annotations. There are two elements of concern: the number of belugas that are correctly predicted and the amount of area that is predicted as containing a whale in each region. Maximizing beluga predictions is essential for the pipeline to be considered trustworthy by a human observer, however, predicting too many false positives will decrease the effectiveness for the observer by raising the time requirements to look through output. Given that classifiers produce a probability estimate of the presence of a beluga, a probability threshold is needed to make a concrete decision on whether a beluga is or is not present.

When one has a classifier that can generate probabilities for a discrete classification task, there exists several traditional measures for evaluating how well a classifier performs at different probability thresholds. Some of these include area under the receiver operating characteristic curve (AUROC – measure true positive rate vs. false positive rate) and area under the precision

recall curve (AUPRC – measures precision vs. recall) that evaluate the ratio of different confusion matrix metrics over all thresholds. However, as shown in the work by Carrington *et al.* (2020), these metrics can be skewed by factors such as class imbalance because they integrate over portions of the curve that are not always relevant and can provide a false sense of utility. This is relevant to our study because model predictions are to be used to guide a human observer and therefore there is a lower bound of whales that must be detected and an upper bound of total surface area that can predicted in order for the model to be useful.

In this vein, we define a metric called **partial true positives vs surface area (PTPVSA)** in order to gauge how well the models performed in terms of detecting whales and keeping surface area low. By evaluating at probability thresholds in the range of 0.005 to 1.0 with increments of 0.005 we have PTPVSA defined as

$$PTPVSA = \frac{1}{\nu\gamma}\int_0^{\nu} \max(f(x) - \gamma, 0)\, dx$$

where *f(x)* is the percentage of total belugas predicted for *x* surface area, ν is the upper bound of acceptable percentage of total surface area of the image and γ is the lower bound of acceptable percentage of belugas predicted. PTPVSA is normalized between 0 and 1, where 0 indicates that f(x) was below γ for all x between 0 and ν, and 1 indicates that all belugas are predicted for all x between 0 and ν. The values of $\nu = 0.1$ and $\gamma = 0.8$ were chosen because results within these thresholds were deemed most useful to an observer. In other words, we felt that a pipeline that predicted more than 10% average surface area or less than 80% of the total number of belugas would not be useful to an observer. It is noted that *f(x)* was linearly interpolated between evaluating at different probability threshold points. Note that PTPVSA only evaluates over a portion of the curve and should not be interpreted as other metrics such as area under the receiver operating characteristic curve.

Experimental Setup

With these evaluation metrics now defined, we conduct four different experiments to evaluate our methods:

- Experiment 1 investigates a selection of baseline classifiers.
- Experiment 2 explores the effect of applying ensembles of classifiers.
- Experiment 3 explores effect of hardmining to inject high glare examples into the training set with the ensemble classifiers in experiment 2.
- While the first three experiments are focused towards scenario A by training on patches from *Train-A 2014*, experiment 4 takes the best performing model and evaluates its performance and trains it on *TrainPatches-B+HighGlare and evaluates on Eval-B 2014 and Eval-B 2017.*

Results

Here we will present the results of the four experiments described above. A summary of our results can be viewed in Table 3. The runtime for each of the models were as follows: RF=20 s/image, BaseCNN=1.2 s/image, RESNET=1.5 s/image, SE-RESNET=2.0 s/image, BaseCNN =12 s/image, RESNET-x10=15 s/image, and SE-RESNET-x10=20 s/image. It should be noted however that our pipeline was written in Python and not fully optimized for runtime.

Experiment 1 evaluated four classification models: RF, BaseCNN, RESNET, and SE-RESNET. A plot showing the trade-off between predicted surface area and the fraction of detected belugas is shown in Figure 7. While RF had higher PTPVSA in *Eval 2014*, this was a

result of its predictions being more sensitive to threshold, so it can predict a larger range of surface areas, despite it being below the curves for the deep learning models. For *Eval 2017*, the PTPVSA was much lower. Based on the position of the curves in Figure 7, the models follow similar characteristics where SE-RESNET is on top, followed by RESNET, BaseCNN, and RF. The minimum predicted surface area differs greatly for each of the models, causing significant differences in PTPVSA despite the aforementioned ordering.

Experiment 2 compared the performance of the deep learning models when they were assembled into ensembles of size 10. It is noted that RF is already an ensemble of decision trees, so evaluating an ensemble of RF models would be redundant. The results are visualized in Figure 8. Compared to the models in Experiment 1, the PTPVSA for each deep learning model was greater than the non-ensemble variants. Lastly, when deep learning models are assembled into ensembles, they are also able to make predictions at lower surface areas. Therefore, we can more genuinely compare the PTPVSA of RF and deep learning models.

In Experiment 3, we evaluated our trained models on images with a high amount of glare that did not contain belugas in order to reduce the number of false positives in the data set, creating the set *TrainPatches-A + Glare* as described in Glare Adjustment. This increases the number of whales identified while decreasing the predicted surface area for each threshold, which increases the PTPVSA, as shown in Figure 9.

In Experiment 4, we chose SE-RESNET-x10 as the evaluation model based on the results from the previous experiments. We trained this model using *TrainPatches-B + HighGlare* and evaluated it on *Eval-B 2014* and *Eval-B 2017* respectively. There was a large difference in evaluation performance between the two years. As shown in Figure 10, it is clear that the model

has significantly lower prediction scores in 2014 than in 2017. The model only achieved a PTPVSA of 0.0008 in 2014 while it achieved 0.624 in 2017.

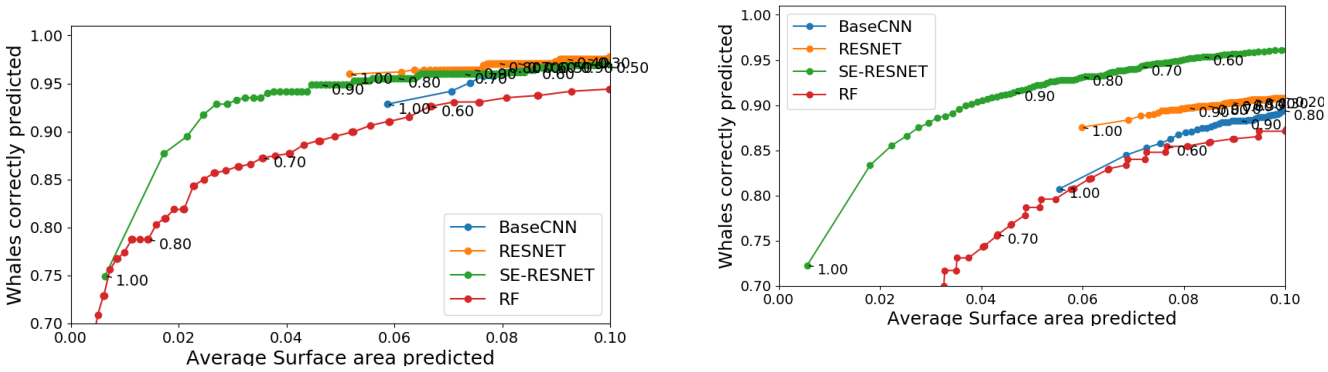


Figure 7: Experiment 1 prediction results. Shows classification results with baseline models trained on *TrainPatches-A* with probability thresholds between 0.005 and 1.0. This figure shows the ability of the models to generalize, with SE-RESNET performing better than the others. (a) Eval-A 2014 (b) Eval-A 2017. The annotations below the observed points indicate the threshold that was chosen to obtain the given point.

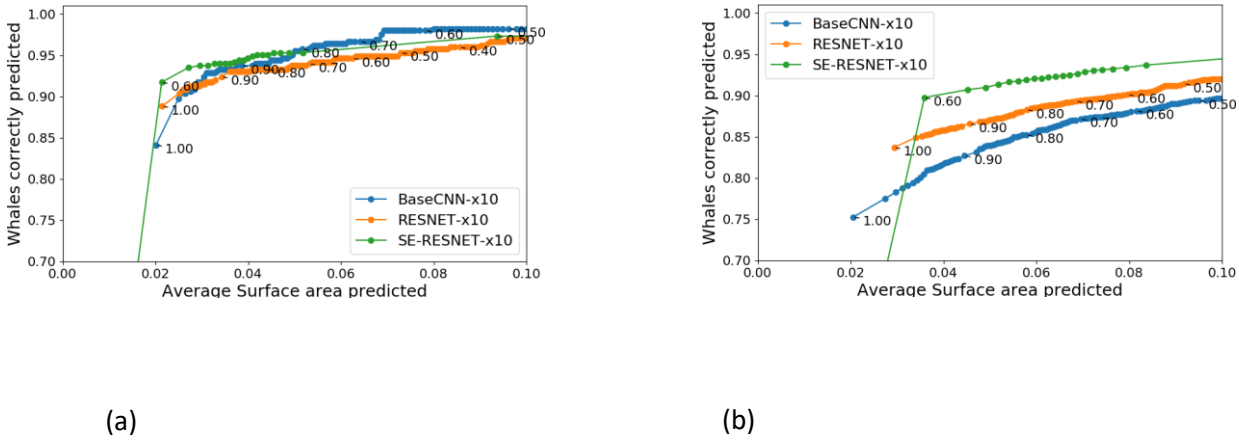


Figure 8: Experiment 2 prediction results. Shows classification results when ensembles of 10 CNN models are trained on *TrainPatches*-A with probability thresholds between 0.005 and 1.0. (a) Eval-A 2014 (b) Eval-A 2017

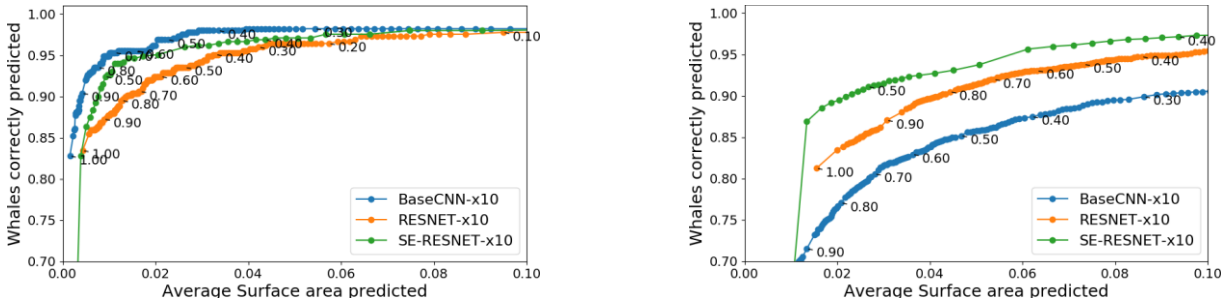


Figure 9: Experiment 3 prediction results. Shows classification results with ensemble models trained on *TrainPatches-A + Glare* with probability thresholds between 0.005 and 1.0 inclusive. (a) *Eval-A 2014* (b) *Eval-A 2017*

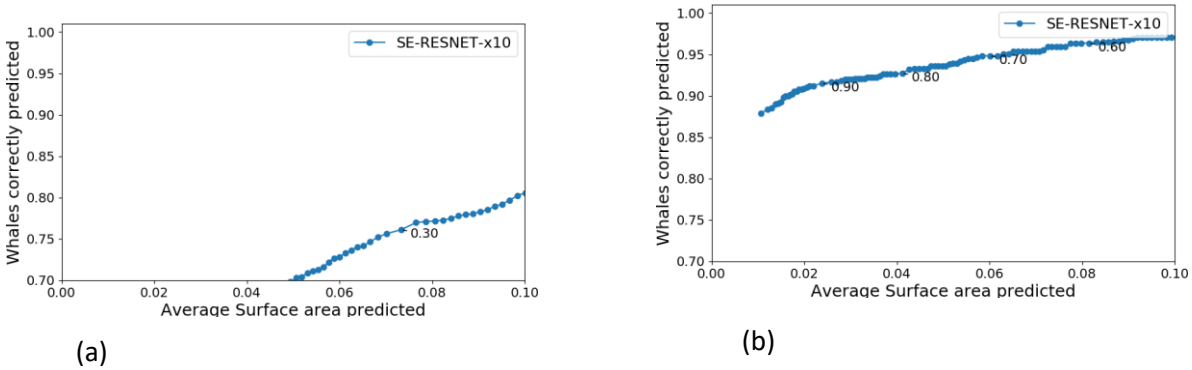


Figure 10: Experiment 4 prediction results. Shows classification results with ensemble models trained on *TrainPatches-B + Glare* with probability thresholds between 0.005 and 1.0 inclusive. (a) Eval-B 2014 (b) Eval-B 2017. The results on *Eval-B 2014* are much worse than previously seen due to it having visual characteristics that less represented in *TrainPatches-B + Glare.*

Table 3: Numerical PTPVSA Results

Training dataset	**Model**	**Eval 2014**	**Eval 2017**
Experiment 1: Baseline comparison			
TrainPatches-A	RF	0.4227	0.1026
TrainPatches-A	BaseCNN	0.3219	0.1308
TrainPatches-A	RESNET	0.4047	0.1891
TrainPatches-A	SE-RESNET	0.6200	0.5065
Experiment 2: Ensemble			
TrainPatches-A	BaseCNN-x10	0.6178	0.1991
TrainPatches-A	RESNET-x10	0.5578	0.3022
TrainPatches-A	SE-RESNET-x10	0.6666	0.4769
Experiment 3: High glare			
TrainPatches-A + *Glare*	BaseCNN-x10	0.8487	0.2566
TrainPatches-A + *Glare*	RESNET-x10	0.7222	0.4700
TrainPatches-A + *Glare*	SE-RESNET-x10	0.7913	0.6320
Experiment 4: Training on 2017			
TrainPatches-B + *Glare 2017*	SE-RESNET-x10	0.0008	0.624

Discussion

In the first experiment we evaluated a selection of our models on *TrainPatches-A* with neither ensemble (except for RF) nor glare augmentation. From the graphs in Figure 7, it is apparent that the predictions from RF create a lower curve than the deep learning models, indicating that it under-performs the deep learning models with higher surface areas predicted. The deep learning models having a higher curve means they can predict more belugas for the same amount of surface area predicted. However, it must be noted that the PTPVSA value for BaseCNN and RESNET are lower for *Eval 2014* due to their inability to make predictions at lower surface areas. Ultimately, it is clear that SE-RESNET was able to outperform all of the other models, as it was able to make predictions at low surface areas and produce a higher curve in the graph. The performance improvement of SE-RESNET is greatest in *Eval 2017*. Since *Eval 2017* contains scenes of very different appearance than the *TrainingPatches-A*, which were extracted from the 2014 set, this shows squeeze-excitation blocks allow the model to learn more general features of belugas.

In the second experiment, we evaluated the usage of ensemble models while training with *TrainPatches-A.* We found that in the cases of BaseCNN-x10 and RESNET-x10, the introduction of ensembles raised the PTPVSA by widening the distribution of the network predictions and enabling the models to predict at lower surface areas. The effect of ensembles on SE-RESNET-x10 was less clear, as the PTPVSA improved in *Eval-A 2014* but reduced in *Eval-A 2017* compared to SE-RESNET. Part of this is due to a sharp change at the 0.6 threshold, which seems to indicate that some of the ensembles did not train consistently and 4/10 of the models in the ensemble made vastly different predictions. Nonetheless, SE-RESNET-x10 still outperforms the other ensemble CNN models. Correct predictions made by SE-RESNET-x10 are shown in Figure 11. Ultimately, we found better performance from all models when training with an ensemble of CNNs. This observation is expected because with a wide variety of scene types,

depending on initialization, CNNs can learn different features. Using an ensemble of CNNs allows the limitations of each individual CNN to be minimized.

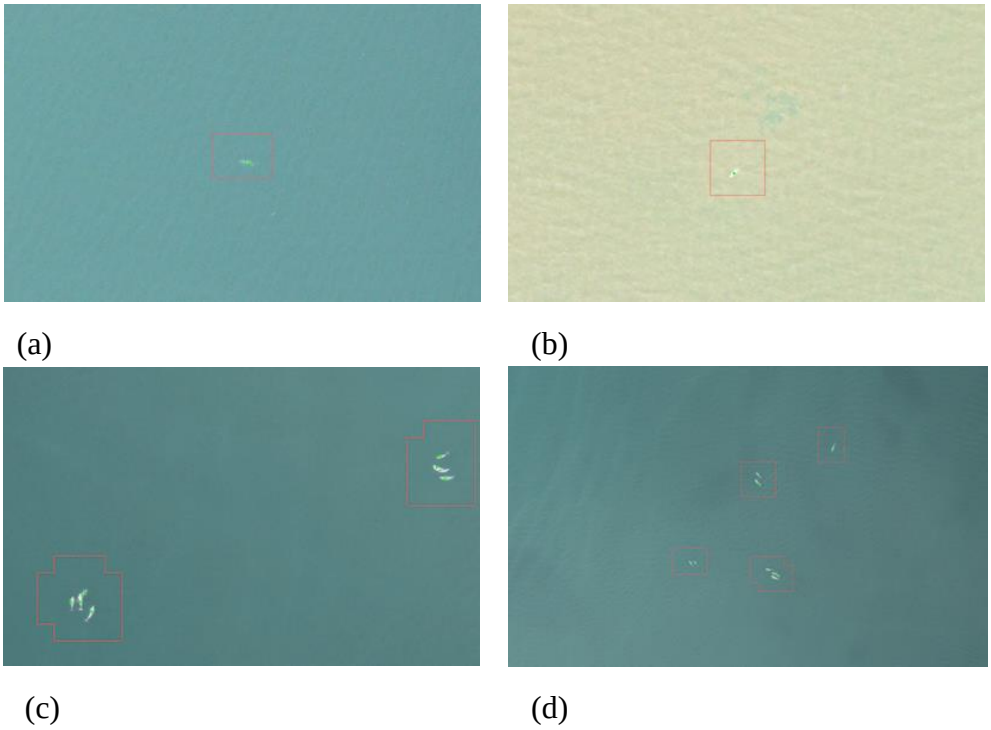


Figure 11:Predictions made correctly by SE-RESNET-x10 on scenes taken (a) 08/03/2017 (b) 08/03/2017 (c)08/07/2017 (d)08/07/2017. Image dimensions are a) 430 x 688 pixel, b) 339 x 542 pixel, c) 386 x 615 pixel, and d) 758 x 1213 pixel. Regions with high probability of containing a whale in the 65 x 65 neighborhood are outlined in red.

In the third experiment, we evaluated training ensemble CNNs with a training set that was augmented with high glare examples, *TrainPatches-A + Glare*. Overall, the augmentation greatly improved performance by lowering the ratio of surface area per beluga detected, as evidenced by both higher PTPVSA values and higher curves in for all models. While BaseCNN-x10 performed the best for *Eval 2014* (PTPVSA=0.8487), its performance gap for *Eval-A 2017* (PTPVSA=0.2566) was large. In contrast, SE-RESNET-x10 performed best for *Eval-A 2017* (PTPVSA=0.6320) and was a close second place for *Eval-A 2014* (PTPVSA=0.7913). The large

performance improvement shows the impact of expanding the dataset to handle different environmental variables.

We choose an ensemble of 10 SE-RESNETs as our best model for its ability to generalize and have better performance on a data set from a different year. The SE-RESNET identifies ~95% of belugas with ~2% of images labelled from *Eval-A 2014* and ~90% of belugas with ~2% of images labelled from *Eval-A 2017*. It must also be noted that all models have lower performance on *Eval-A 2017* which is due to the different appearances of the two datasets.

The fourth experiment also highlights the discrepancy of visual appearances between 2014 and 2017. It is clear from the numerical results that the model trained on 2017 generalized extremely poorly towards the images in 2014. The difference in PTPVSA seems to indicate that there is subset of images in 2014 that are entirely dissimilar to those found in the training set from 2017. One aspect that supports this the change in weather conditions between the years because, as mentioned in Table 1, most of the observation days in 2017 were cloudy while 2014 contained significantly more days with clear conditions. The fact that training on 2014 achieved better results could indicate that appearance of *TrainPatches-A* was more diverse and therefore better able to handle the conditions in 2017.

Now we discuss the contrast in results between *Eval-A 2014* and *Eval-A 2017* and the implications on their respective scenarios. In all scenarios, the performance on *Eval-A 2014* was higher than *Eval 2017*, indicating that there were significant factors between the two time periods that inhibited equivalent performance between the scenarios for evaluation in the same time period and evaluation in future time periods. From Table 3, the performance gap between the two datasets is largest for BaseCNN, narrowed for RESNET, and best for SE-RESNET. Since SE-RESNET also typically had higher PTPVSA than the other models, it is apparent that

the squeeze-excitation blocks were very effective for this task. As the novelty of the squeeze-excitation blocks is the interaction and control between channels non-locally, this could indicate they are more effective at handling background changes. However, Experiment 4 showed the squeeze-excitation blocks have limits in terms of handling background changes.

Background analysis

Given the difference in results between *Eval-A/B 2014* and *Eval-A/B 2017*, we investigate whether the background colour is correlated with the difference in performance of the models as an indirect way of assessing whether the environmental variables are a cause for the discrepancy in performance. We specifically examine the distribution of background colour between true positives and false negatives for *Eval 2014* and *Eval 2017*. For each beluga coordinate, we extract the background colour of a 65 × 65 pixel box by the following method. First, we convert the images from red-green-blue (RGB) colour space to the hue-saturation-value (HSV) colour space, as it is more informative for explaining the natural effects of illumination on the images. In order to detect the background, we assume that the beluga is brighter than the background. As the value channel is indicative of brightness, we select all pixels that are less than or equal to the 25th percentile of the value channel in the 65 × 65 patch as the set **S**. We then record the mean hue, saturation, and value for all pixels in **S**.

We examine the predictions from the models RESNET-x10 and SE-RESNET-x10 after training with *TrainPatches-A + Glare*. We chose the thresholds based so that the mean surface area predicted would be the same for both methods for both RESNET-x10 (0.8 for 2014 and 0.95 for 2017) and SE-RESNET-x10 (threshold=0.8 for 2014 and 0.95 for 2017) for the purpose of examining the distribution of colour values. Table 4 shows the hue, saturation, and value for true positives and false positives for *Eval 2014-A* and *Eval 2017-A* via the process listed above. From

these figures, we can observe that the distributions of *Eval-A 2014* and *Eval-A 2017* follow distinct trends. In *Eval-A 2014* using RESNET-x10, the false negatives appear to be slightly skewed towards lower saturation and higher value. This corresponds to backgrounds with high glare, as low saturation values and high value produces a washed-out effect that is produced by solar glare. An example of this is shown in Figure 12, in which it is difficult to identify the beluga, even to a human observer. The cumulative effect of the glare according to the HSV plots was less clear in SE-RESNET, indicating that the squeeze-excitation blocks allowed better compensation for these backgrounds.

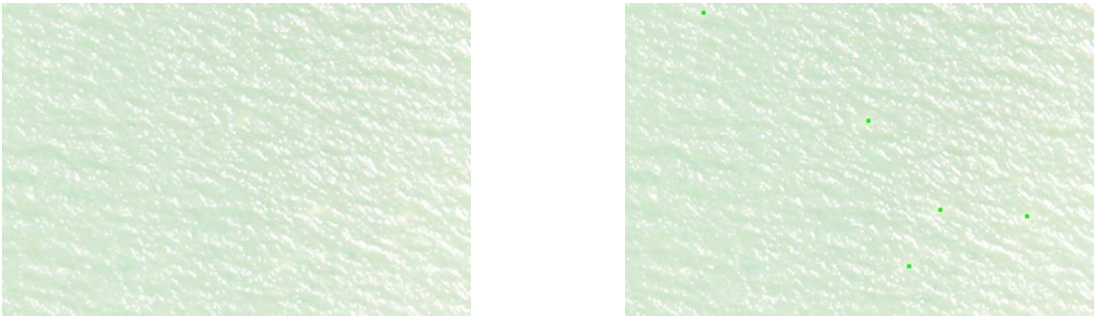
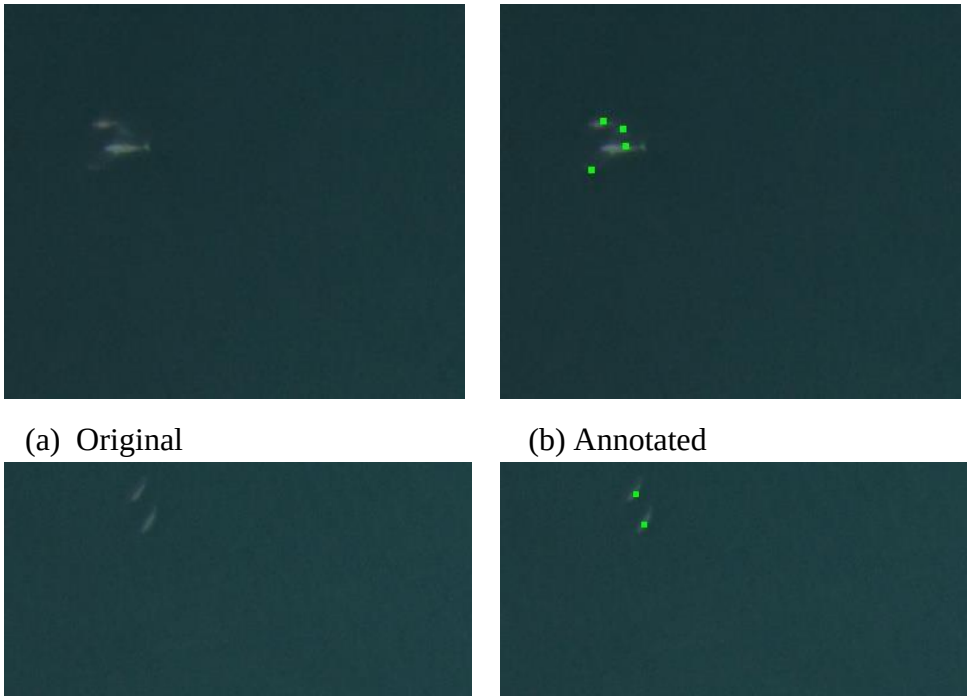


Figure 12: Belugas missed by SE-RESNET-x10. Original high glare image (left) and annotated image (right) from 2014 data set with the green dots being ground truth labels of belugas. (600 x 400 pixel region)

For *Eval-A 2017*, false negatives are concentrated with value measurements much lower than those in *Eval-A 2014* for both RESNET-x10 and SE-RESNET. This indicates that the classifiers had issues in correctly predicting beluga patches that have dark backgrounds. An example of this is shown in Figure 13, where the classifiers fail to detect belugas on dark backgrounds. These examples are very easy to spot for a human observer. The fact that these false negatives appear at value intensities much lower than the background of any beluga in *Eval*

2014 suggests that there a deficiency in the training set where examples of belugas with dark backgrounds are not present, and subsequently the classifiers have difficulty in predicting these samples in *Eval 2017*.

In the context of training with the *TrainPatches-B + High Glare*, Table 5 shows the HSV distributions for true positives and false negatives for SE-RESNET-x10. It can be observed that for *Eval-B 2014*, there is a larger concentration of false positives associated with the mid-level hues, low saturation, and high value intensity. This appears to demonstrate a converse relationship that occurred when training on 2014 because the model has become accustomed to the darker images captured during the cloudy days of 2017 while missing the brighter examples in 2014.



(a) Original (b) Annotated

(c) Original (d) Annotated

Figure 13: Two outlier examples where the model misses the belugas in the 2017 data set due to the darker background that is not present in the 2014 data set. a, b) 350 x 300 pixel sample c, d) 400 x 200 pixel sample.

Thus, it appears that variables such as weather, illumination, and surface conditions significantly differ between datasets. These variables have a significant impact on in the appearance of the images that cause discrepancies between the training set and testing set that are problematic for generalizing models between different time periods. Unfortunately, obtaining a training set that contains beluga patches with varying weather, illumination, and surface conditions is not be immediately feasible because it would require multiple surveys taken over a

large period of time. There are two approaches that could be taken to compensating for these environmental variables that could be explored for future work. The first would be implementing an augmentation algorithm that is capable of simulating changes in the environmental variables within the training set in order to make the model more robust. The second would be directly accounting for these environmental variables within the models such as by training separate models to match specific environments, such as cloudy and sunny, and then combining their predictions as a mixture model. When we obtain additional surveys in future work, we plan to adapt our training strategy to merge images from multiple surveys to evaluate on a single unseen survey. This will allow us to monitor the sensitivity of our methods to new survey conditions when additional training data from multiple time periods is considered.

Table 4: True positives and false negatives of model predictions according to HSV colour analysis with TrainPatches + Glare for 1 = (Eval 2014, RESNET-x10), 2 = (Eval 2014, SE-RESNET-x10), 3 = (Eval 2017, RESNET-x10), and 4 = (Eval 2017, SE-RESNET-x10)

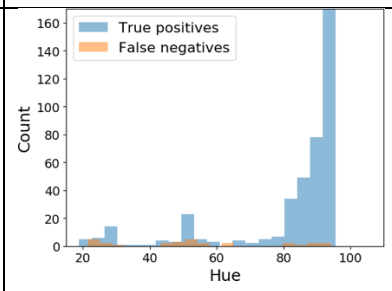
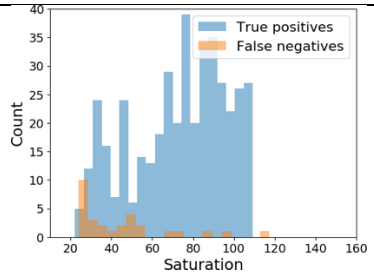
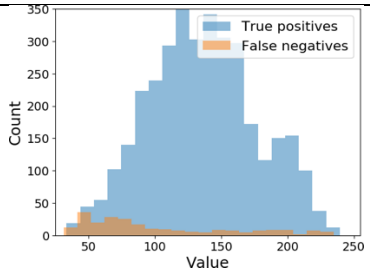
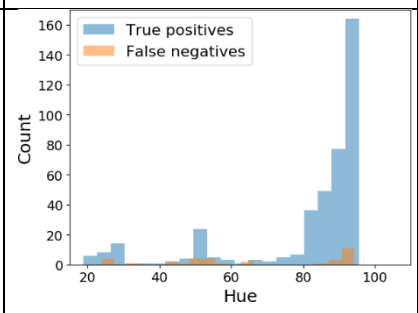
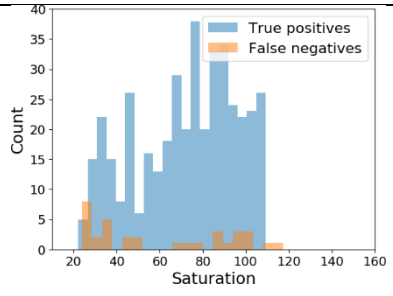
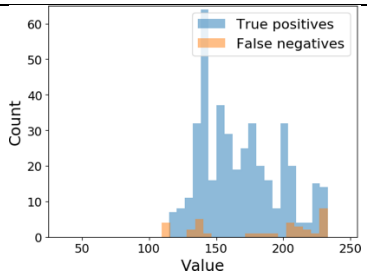
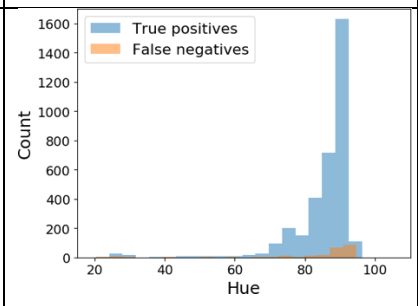
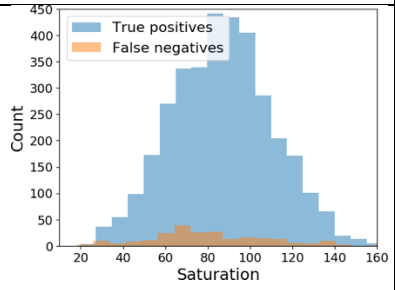
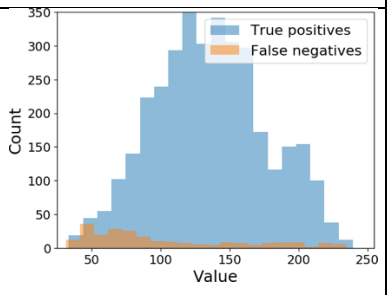
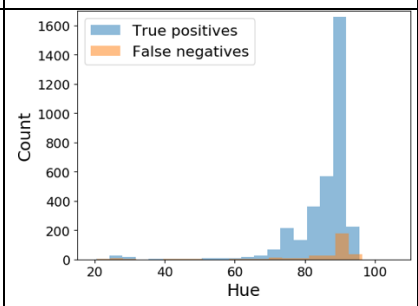
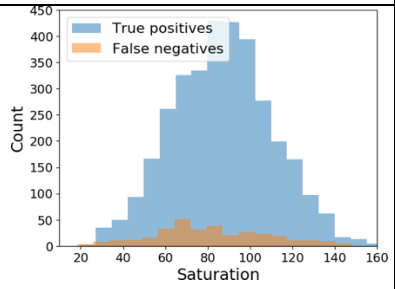
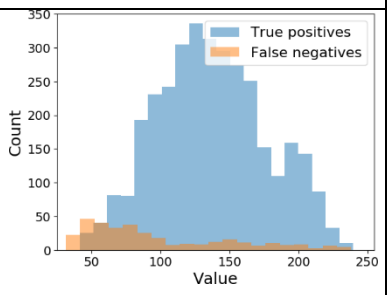
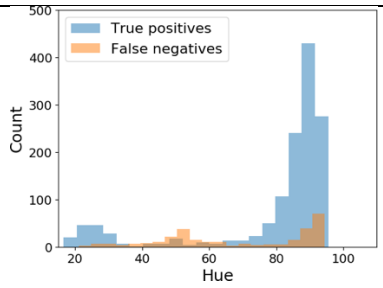
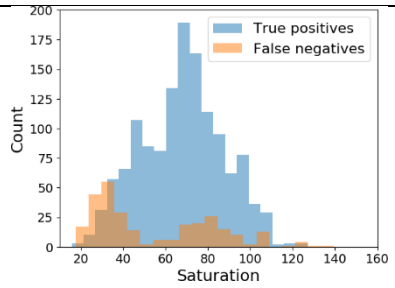
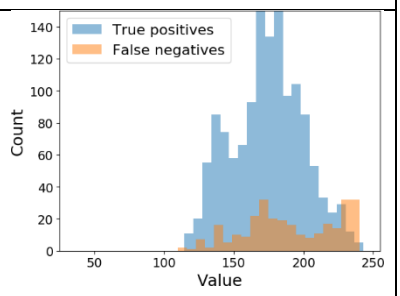
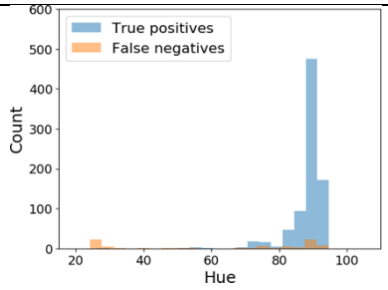
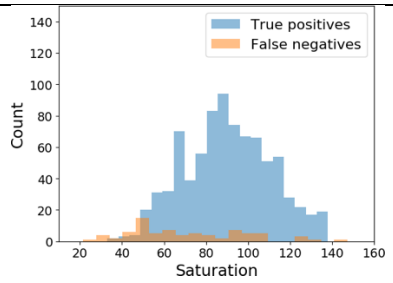
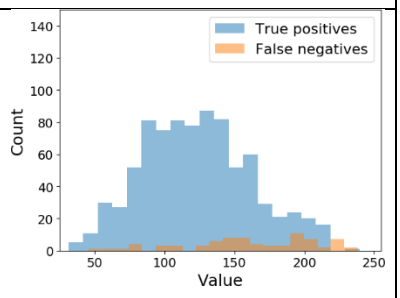
	Hue	Saturation	Value
1			
2			
3			
4			

Table 5: True positives and false negatives of model predictions according to HSV colour analysis with Train2017Patches + Glare for A = (Eval 2014, SE-RESNET-x10), B = (Eval 2017, SE-RESNET-x10),

	Hue	Saturation	Value
A			
B			

Conclusion

In this study, we applied CNNs to identify Cumberland Sound beluga belugas from aerial images. Aerial images and beluga sightings used in this study were part of an aerial survey of Cumberland Sound Bay, Nunavut during the summer observation periods of 2014 and 2017.

Comparing with our baseline classifier, a RF classifier, we show that deep learning is a viable method for identifying belugas in aerial images. We support this by showing CNNs are capable of identifying more belugas in the images, while labelling less surface area. Three CNNs were used for this study and they were a base CNN consisting of three convolutional layers and two fully connected layers (BaseCNN), a RESNET which is the BaseCNN with residual blocks, and a SE-RESNET which is the BaseCNN with residual and squeeze-excitation blocks. We also investigated ensemble models and data augmentation through glare adjustment. We found that all CNNs have superior performance to the RF classifer, but SE-RESNET as an ensemble of 10 models using glare adjustment had the best performance. Squeeze-excitation blocks and residual network blocks assisted the model in learning general features of whales while creating ensembles allowed more stable predictions.

Therefore, from this study, we conclude that through deep learning we can attain automation of beluga identification from aerial images. To make our methods production ready, more data is required to provide more exposure to different types of weather and lighting conditions.

Acknowledgements

The authors of this work received funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) (RGPIN-2017-04869, DGDND-2017-00078, RGPAS-2017-50794, RGPIN-2019-06744), the Marine Environmental Observation Prediction and Response Network, and from the University of Waterloo. The aerial survey was supported by Fisheries and Oceans Canada, Polar Continental Shelf Program, and the Nunavut Wildlife Management Board. Additional thanks are extended to the Aerial survey team and to L. Montsion for manual photo analysis and to the Pangnirtung Hunters and Trappers Association for their support and guidance during the aerial surveys.

References

Andrew, W., C. Greatwood, and T. Burghardt. 2017. "Visual localisation and individual identification of Holstein Friesian Cattle via deep learning." *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2850–2859.

Barbedo, J. G. A, L. V. Koenigkan, P. M. Santos, and A. R. B. Ribeiro. 2020. "Counting cattle in UAV images—Dealing with clustered animals and animal/background contrast changes." *Sensors*, no. 7 (20): 2126.

Carrington, André M., Paul W. Fieguth, Hammad Qazi, Andreas Holzinger, Helen H. Chen, Franz Mayr, Douglas G. Manuel. 2020. "A new concordant partial AUC and partial c statistic for imbalanced data in the evaluation of machine learning algorithms." *BMC Medical Informatics and Decision Making* 20 (4).

COSEWIC. 2004. COSEWIC Assessment and Update Statuis Report on the Beluga Whale Delphinapterus leucas in Canada. Technical report. Ottawa. https://www.sararegistry.gc.ca/virtual_sara/ files/cosewic/sr_beluga_whale_e.pdf.

Courtrai, L., M. T. Pham, and S. Lefèvre (2020). Small Object Detection in Remote Sensing Images Based on Super-Resolution with Auxiliary Generative Adversarial Networks. Remote Sensing, 12(19), 3152.

Dalal, N., and B. Triggs. 2005. "Histograms of oriented gradients for human detection." *In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'05), 1:886–893. IEEE.

de March, B.G.E., Stern, G., and Innes, S. 2004. The combined use of organochlorine contaminant profiles and molecular genetics for stock discrimination of white whales (Deplphinapterus leucas) hunted in three communities on southeast Baffin Island. J. Cetacean Res. Manag. 6: 241-250.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale

hierarchical image database." In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248-255. IEEE, 2009.

DFO. Unpublished. Recovery strategy for the beluga whale (Delphinapterus leucas) in Cumberland Sound. Species at Risk Act Recovery Strategy Series, Fisheries and Oceans Canada, Ottawa.

DFO. 2002. Sci. Stock Status Rep. E5-32. Technical report. *Government of Canada*. https://waves-vagues.dfo-mpo.gc.ca/Library/274059.pdf.

Fretwell, P. T., I. J. Staniland, and J. Forcada. 2014. "Whales from space: Counting Southern Right Whales by satellite." *PLOS ONE 9*, no. 2 (February): 1–9. doi:10.1371/journal.pone.0088655. https://doi.org/10.1371/journal.pone.0088655.

Ghorbanzadeh, O., D. Tiede, L. Wendt, M. Sudmanns, and S. Lang (2020). Transferable instance segmentation of dwellings in a refugee camp-integrating CNN and OBIA. European Journal of Remote Sensing, 1-14.

Girshick, R.. 2015. "Fast r-cnn." *In Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.

Girshick, R., J. Donahue, T. Darrell, and J. Malik. 2014. "Rich feature hierarchies for accurate object detection and semantic segmentation." *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

Glorot, X., A. Bordes, and Y. Bengio. 2011. "Deep sparse rectifier neural networks." *In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, edited by Geoffrey Gordon, David Dunson, and Miroslav Dud´ık, 15:315–323. *Proceedings of Machine Learning Research*. Fort Lauderdale, FL, USA: PMLR, November. http://proceedings.mlr.press/v15/ glorot11a.html.

Government of Canada. 2019. "Historical climate data." https://climate.weather.gc.ca/historical_data/search_historic_data_stations_e.html?hlyRange=2013-03-05%7C2020-06-24&dlyRange=2018-10-29%7C2020-06-24&mlyRange=%7C&StationID=51217&Prov=NU&urlExtension=_e.html&searchType=stnProx&optLimit=specDate&StartYear=2014&EndYear=2015&selRowPerPage=25&Line=81&lstProvince=NU&timeframe=1&Year=2014&Month=8&Day=6&txtRadius=25&optProxType=navLink&txtLatDecDeg=66.145&txtLongDecDeg=65.713611111111&station=PANGNIRTUNG%20A (accessed June 2020)

Guirado, Emilio, Siham Tabik, Marga L. Rivas, Domingo Alcaraz-Segura, and Francisco Herrera. "Whale counting in satellite and aerial images with deep learning." *Scientific reports* 9, no. 1 (2019): 1-12.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep residual learning for image recognition." *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hu, J., L. Shen, and G. Sun. 2018. "Squeeze-and-excitation networks." *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7132–7141.

Ioffe, S., and C. Szegedy. 2015. “Batch normalization: Accelerating deep network training by reducing internal covariate shift.” arXiv preprint arXiv:1502.03167.

Ju, C., A. Bibaut, and M. van der Laan. 2018. “The relative performance of ensemble methods with deep convolutional neural networks for image classification.” *Journal of Applied Statistics* 45 (15): 2800–2818.

Kellenberger, B., D. Marcos, and D. Tuia. 2018. “Detecting mammals in UAV images: Best practices to address a substantially imbalanced dataset with deep learning.” *Remote Sensing of Environment* 216:139–153. issn: 0034-4257. doi:https://doi.org/10.1016/j.rse.2018.06.028. http://www.sciencedirect.com/science/article/pii/S0034425718303067.

Kingma, D. P., and J. Ba. 2014. “Adam: A method for stochastic optimization.” arXiv preprint arXiv:1412.6980.

Krizhevsky, A. 2009. “Learning Multiple Layers of Features from Tiny Images.”

LeCun, Y., Y. Bengio, et al. 1995. “Convolutional networks for images, speech, and time series.” *The Handbook of Brain Theory and Neural Networks* 3361 (10): 1995.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO: Common objects in context." In *European conference on computer vision*, pp. 740-755. Springer, Cham, 2014.

Maire, F., L. M. Alvarez, and A. Hodgson. 2015. “Automating marine mammal detection in aerial images captured during wildlife surveys: a deep learning approach.” *In Australasian Joint Conference on Artificial Intelligence*, 379–385. Springer.

Maire, F., L. Mejias, A. Hodgson, and G. Duclos. 2013. “Detection of dugongs from unmanned aerial vehicles*.” In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2750–2756. IEEE.

Marcoux, M., and Hammill, M.O. 2016. Model estimates of Cumberland Sound beluga (Delphinapterus leucas) population size and total allowable removals. Canadian Science Advisory Secretariat Res. Doc. 2016/077: iv + 35 p.

Nanni, L., S. Ghidoni, and S. Brahnam. 2018. “Ensemble of convolutional neural networks for bioimage classification.” *Applied Computing and Informatics*.

Pal, M.. 2005. “Random forest classifier for remote sensing classification.” *International Journal of Remote Sensing* 26 (1): 217–222.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. “Scikit learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12:2825–2830.

Pelletier, C., S. Valero, J. Inglada, N. Champion, and G. Dedieu. 2016. “Assessing the robustness of Random Forests to map land cover with high resolution satellite image time series over large areas.” *Remote Sensing of Environment* 187:156–168.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. 2016. “You Only Look Once: Unified, Real-Time Object Detection.” *In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* June.

Ren, S, K. He, R. Girshick, and J. Sun. 2015. “Faster r-cnn: Towards real-time object detection with region proposal networks.” *In Advances in Neural Information Processing Systems*, 91–99.

Pierre, R., and D.B. Stewart. 2009. Information relevant to the identification of critical habitat for Cumberland Sound belugas (Delphinapterus leucas). DFO Can. Sci. Advis. Secr. Res. Doc. 2008/085: iv + 24 p.

The Gimp Development Team. 2020. “GIMP”. https://www.gimp.org

Van der Walt, S., J. L Schonberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. 2014. “Scikit-image: image processing in Python.” PeerJ 2 (June): e453. issn: 2167-8359. doi:10.7717/peerj.453. https: //doi.org/10.7717/peerj.453.