PROXY-LABEL SEMI-SUPERVISED DEEP LEARNING FOR OBJECT DETECTION AND MAPPING IN SYNTHETIC APERTURE SONAR IMAGERY

S Steele Kraken Robotics

1 INTRODUCTION

Seabed geohazard detection is a requirement for many different types of surveys including hydrographic, habitat, and infrastructure installation (such as cable, pipeline, and offshore wind). In many applications geohazards as small as 20 cm must be detected over large areas. The resolution of side looking sonar imagery strongly impacts the quantity of boulders that can be detected in imagery[1], thus the range independent resolution of synthetic aperture sonar (SAS) makes it an ideal sensor for geohazard detection. Manually identifying and classifying geohazards in SAS imagery is time consuming, creating a bottleneck in project delivery. The application of deep learning for seabed segmentation has the potential to substantially reduce the cost associated with geohazard detection. Supervised deep learning demands an extensive collection of training images, which requires considerable time and resources to generate. In contrast, semi-supervised learning attempts to train a model through the use of a dataset consisting of a small set of labelled training samples and a large number of unlabelled samples. Here, we explore applying the proxy-label semi-supervised learning method to SAS imagery. Previous work in the application of semi-supervised learning to object detection in SAS imagery has focused on autoencoders or architectures analogous to autoencoders, which are self-supervised models that first encode an image into a lower dimensional latent representation, then decode the representation to reconstruct the image. These learned representations can be used to perform semi-supervised learning for segmentation or classification tasks. In the SAS domain these tasks have been trained jointly using architectures such as the ladder network (performs classification, reconstruction, and segmentation)[2] or as two separate steps where the model is first trained to reconstruct SAS imagery and then later fine-tuned to perform classification[3]. Our approach, proxy-label semi-supervised learning, is more akin to bootstrapping. With this technique no image reconstruction step is required, instead we rely on a teacher convolutional neural network (CNN) to generate proxy-labels (i.e. noisy or imperfect labels) for a large set of unlabeled SAS images, which can then be used to train another CNN model referred to as the student. There are numerous categories of proxy-label techniques including self-training, co-training, and multi-view learning[4]. Even though the architecture presented in our work trains a new model with the proxy-labels, we still consider it to be a form of self-training. Our approach can be considered a variation of the pseudo-label technique[5], which trains on labeled and unlabeled data simultaneously with two forward passes. In this text we will refer to any model architecture that assigns proxy labels to unlabelled data, which are used as targets for learning, as proxy-label techniques, and reserve the term pseudo-label for the original paper that introduced the pseudo-label architecture. The pseudo-label and other proxy-label semi-supervised learning techniques have been shown to increase model robustness and generalisation, even when training data is abundant[5,6].

The primary differences between our proxy-label technique and the pseudo-label technique are the num-

ber of forward passes (ours only requires one, making it faster to train) and the treatment of noisy labels. To minimize the impact of noisy labels, most proxy-label methods either: 1) weight the unlabeled data training loss such that their contribution increases over time (i.e.,model epochs)[5], or 2) utilize only high confidence predictions for training by choosing a confidence threshold, which can either be absolute or relative (i.e., the top N unlabeled training samples are used in training the next iteration)[7]. Our proxy-label approach does not require a proxy-label weighting parameter or cutoff. Here, we utilize a CNN pre-trained to identify objects in optical imagery (VGG16[8]) as the teacher model and a U-Net[9] model architecture for the student model. In previous work, the teacher model was fine tuned to detect geohazards in SAS imagery using transfer learning on a small set of labeled SAS images[10]. Thus, no confidence threshold or weighting is required because, with the transfer learning, we can consider all predictions to be high confidence. We also utilize data augmentation and dropout to add noise to the student model, which increases generalization by reducing overfitting (i.e., the model is "forced to learn harder"[6] from the proxy-labels). The teacher model was used to generate a large set of labeled SAS images to train a student network with the specific task of detecting geohazards in SAS imagery. In this paper we will explore how different combinations of SAS derived features (such as bathymetry and vertical uncertainty) impact the student model performance. To the author's knowledge, the only SAS feature utilized in the literature to train deep learning models for object detection has been imagery (both intensity and complex). This paper will also explore the impact of using the teacher model confidence to assign a weighting to each pixel during student network training. Results from our proxy-label learning framework will be demonstrated on imagery collected during numerous surveys using a Kraken Robotics miniature interferometric SAS (MINSAS).

2 METHODS

The proxy-label semi-supervised learning technique consists of a teacher network trained on a small manually labeled (i.e., ground truth) dataset and a student network trained on a large set of proxy-labels produced by the teacher network. A generalized workflow of our implementation of the proxy-label technique can be found in Figure 1. We leverage a previously developed CNN trained to segment geohazards in SAS imagery through transfer learning on a small dataset of ground truth labels[10]. We used this transfer learning model to generate predictions and confidence weights for thousands of unlabelled SAS images from numerous surveys. These predictions were utilized as noisy image labels to train a new student network that incorporates other SAS derived features not typically utilized for CNN training such as bathymetry and vertical uncertainty.

2.1 Teacher Model

The teacher model follows the U-Net architecture[9], which consists of a contracting and expansive path. The contracting path can be any CNN, while the expansive path uses up-convolutions and concatenations to combine the CNN features with spatial information to produce a high resolution segmentation map. We chose the VGG16 architecture pre-trained on ImageNet[11] data as the base CNN (contracting path) because it has been shown to be highly effective at detecting objects in SAS imagery with the application of transfer learning[12]. The ImageNet dataset consists of over 1.2 million training images belonging to 1000 different classes. The VGG16 model filters learned from the ImageNet data can be modified to detect boulders through the fine tuning transfer learning technique. To fine tune a model some layers are frozen such that their weights don't update during training. The layers that are not frozen utilize the weights as a starting point and the model adjusts the weights based on our training data. We found the model performed best when all the convolutional layers of the VGG16 network were frozen except the last two. To help prevent overfitting during fine tuning we applied the following augmentations: flip (horizontal and vertical), brightness, and zoom. The teacher model on its own yields satisfactory predictions; it

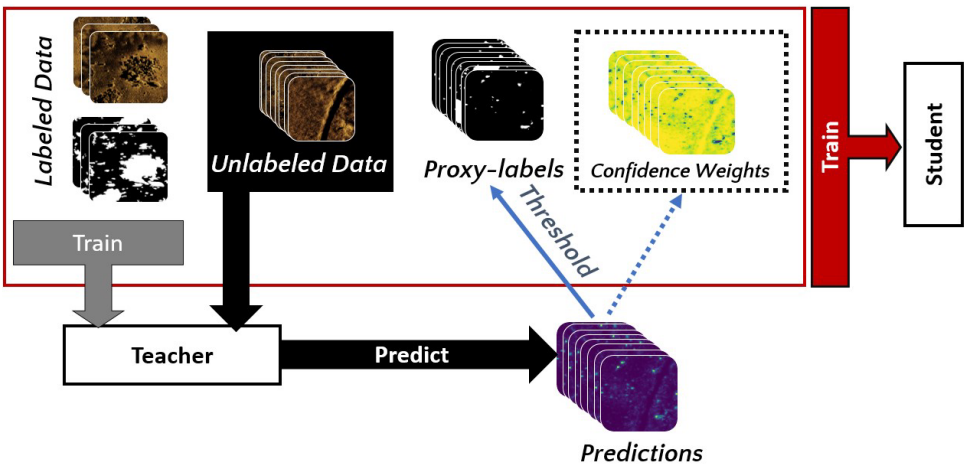


Figure 1: Flow diagram for the proxy-label technique using a teacher model to train a student model. Dotted lines indicate optional calculations and inputs.

achieved a mean IoU of 0.78 on the validation data during training. The teacher outputs the target class (geohazard) probability of each pixel, and thus these must be converted to binary proxy-labels using a threshold during preprocessing before training the student model. We experimented with target class thresholds between 0.2 and 0.7, but found a target class threshold of greater than or equal to 0.3 yields the best mean IoU on the validation data. As shown in Figure 1, we can also use the teacher model predictions to calculate the confidence for each pixel so that weights can be applied to the training data. The intention of this technique is to minimize the impact of low confidence predictions on the student model training. Since the teacher model outputs the probability of the target class, we cannot use the predictions directly as confidence (this would cause all background pixels to be given low confidence). The confidence weights (C) can be calculated from the model predictions (probability map, P) and target class threshold, T, with

$$C = \begin{cases} 1 - P, & \text{if } P < T \\ P, & \text{otherwise.} \end{cases}$$

2.2 Student Model

Like the teacher model, the student model follows the U-Net architecture[9] to produce semantic segmentation predictions. The student model utilizes a similar contracting path CNN architecture to that presented in the U-Net publication[9]. The U-Net contracting path consists of the repeated application of two 3x3 convolutions followed by a rectified linear unit (ReLU), and a 2x2 max pooling operation with stride two down sampling, which reduces spatial resolution. After each down sampling, there is a doubling of the number of convolution feature channels, causing the model's capacity for feature information to go up as the spatial resolution decreases. The final layer of the U-Net model is a 1x1 convolution used to map each feature vector to the desired number of classes. The U-Net model was originally developed for bio-medical imaging with a single channel input. Here, we will compare multiple versions of the student model utilizing various combinations of one to three channels. In particular we compare the student model trained with the following channel combinations: 1) image intensity, 2) image intensity and bathymetry, 3)

image intensity and vertical uncertainty, and 4) image intensity, bathymetry, and vertical uncertainty. In Tables 1 and 2, we will use the following naming conventions for different feature combinations: dB refers to the use of the SAS intensity image, bathy refers to the inclusion of bathymetry feature, and vu refers to the inclusion of the vertical uncertainty feature. The VGG16 base model of the teacher has over 25 million parameters and is likely larger than necessary for our binary semantic segmentation task. Thus, it is desirable to make the student model as small as possible, without sacrificing performance (knowledge distillation[13]). We can achieve knowledge distillation by ensuring the student network is smaller (has fewer layers and/or feature channels) than the teacher network. The student model was trained with one less application of convolutions than the original U-Net architecture, yielding 18 convolutional layers. For comparison, there are 23 convectional layers in the original U-Net architecture (10 in the contracting path, 13 in the expansive path), while the VGG16 model has 13 convolutional layers and three fully connected layers. We chose to reduce the number of starting feature channels from 64, as done with the original U-Net model and VGG16, to 16 feature channels. In total, the student model has 535,793 parameters, which corresponds to about 2 % of the size of the teacher model. The student model was trained with a batch size of 32, learning rate of 1×10^{-4}, and a dropout rate of 0.1.

2.3 Data

The Kraken MINSAS consists of two vertically separated receiver arrays, which provide co-registered seabed imagery and bathymetry at 3 cm and 25 cm resolution, respectively. From the bathymetry, vertical uncertainty maps can also be generated at 25 cm resolution. The MINSAS has a modular configuration, enabling the selection of different array lengths to accommodate desired vehicle, size, speed, and area coverage rates. The MINSAS has a modular array that can be configured with one to four receiver modules (MINSAS 60, 120, 180, and 240). The MINSAS operates at a centre frequency of 337 kHz with a bandwidth of 40 kHz. The MINSAS can optionally include gap reducer imagery. The gap reducer narrows the SAS nadir gap through the use of an additional transmitter operated at a lower frequency (105 kHz, with 40 kHz bandwidth) and a relatively wider beampattern[14]. Not all MINSAS surveys include the gap reducer, but where applicable we include the gap reducer data in our datasets. Data collected by the MINSAS over numerous surveys was used to train the student and teacher network. The teacher model was trained with 350 ground truth labels with a a training and validation split of 85% and 15%, respectively. The student model was trained with the ground truth labels and approximately 4000 images proxy-labeled by the teacher. The manually labeled and proxy labeled data were combined into one dataset. To help control for luck and randomness induced by the proxy labeled dataset, we utilize k-folds cross validation for comparing the performance of the various different student model input channels described in Section 2.2. In k-folds cross validation the training dataset is randomly divided into k subsets (folds). The model is trained and evaluated k times, using a different fold as the validation set each time, while the remaining folds are used as the training data. The performance metrics can be averaged over all the folds to get a more complete and unbiased performance estimate. We chose to use five folds for our analysis because this allows for each model run to use 20% of the data for validation. Additionally, using five folds has been empirically shown to yield estimates that do not suffer from excessively high bias or high variance[15]. In this paper we will present results from two different test sets. The first test dataset (Test A) was used for comparing the different student models and was generated by manually sorting the training model predictions into high accuracy (no or very few incorrect pixel labels observed) and low accuracy. A portion of the high accuracy predictions representing 15% of the entire proxy-labeled dataset (approximately 40% of the high accuracy predictions) was set aside as a test dataset. The remaining data was recombined into one dataset to be later split into five folds of training and validation data. The second test dataset (Test B) utilizes the teacher validation set to evaluate the student model on ground truth labels and allow us to assess if the student can outperform the teacher. Test A and Test B datasets consist of pools of imagery from different surveys, with no overlap (i.e., data from a given survey is included in only one of Test A or B).

Table 1: Test A mean IoU for all cross validation folds and model runs, which includes all feature combinations with confidence weights applied ($W = C$) or no weights applied ($W = 1$). Top row is the mean computed across all five k-folds for each student model.

	dB		dB_bathy		dB_vu		dB_bathy_vu	
	$W = 1$	$W = C$	$W = 1$	$W = C$	$W = 1$	$W = C$	$W = 1$	$W = C$
Mean	0.80	**0.82**	0.78	**0.82**	0.80	**0.81**	0.80	**0.82**
k=0	0.82	0.80	0.82	0.82	0.81	0.82	0.78	0.80
k=1	0.82	0.84	0.79	0.81	0.82	0.82	0.80	0.83
k=2	0.82	0.80	0.80	0.80	0.78	0.80	0.78	0.83
k=3	0.74	0.83	0.74	0.83	0.77	0.80	0.81	0.80
k=4	0.82	0.81	0.76	0.83	0.80	0.82	0.83	0.83

For both the teacher and student models the SAS intensity images were preprocessed with a gaussian blur to minimize the impact of speckle. Following this, all SAS features were downsampled to a size of 224 by 448 pixels. This size was chosen as it generally preserves the aspect ratio of the original SAS data and preserves enough detail to detect geohazards as small as 20 cm in diameter. The factor by which each SAS feature is downsized is variable because each SAS image can vary in size (depends on the survey settings) and not all SAS features are reported at the same resolution.

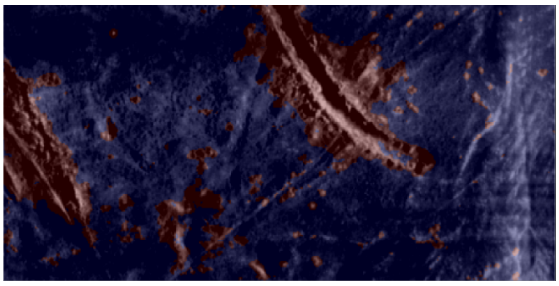
3 RESULTS AND DISCUSSION

A summary of the test set (Test A) mean IoU for all the cross validation folds for each student model feature combinations are shown in Table 1. All versions of the student model are high performing, with all achieving a mean IoU 0.80-0.82. The application of the confidence weights consistently improves the average model performance by 0.01-0.04. The inclusion of the low resolution features (bathymetry, and vertical uncertainty) does not significantly impact the student model performance. It is likely that the lower resolution features lack any attributes to denote smaller objects such as boulders, effectively adding either no additional information or even adding noise. The student model performance on the Test B dataset has been summarized in Table 2. The teacher model can achieve a mean IoU of 0.76 over the Test B dataset, while all the student models can achieve a mean IoUs of 0.78-0.81, indicating the student model is highly skilled at knowledge distillation; the student models can maintain and even increase model performance while using only 2 % of the parameters. Similar to Test A, we see that, for all feature combinations tested, on average the student model performs better when the teacher confidence weights are used. Interestingly, we observe that the best performing model includes both the image intensity and bathymetry features. The inclusion of the bathy feature along with the image intensity feature only marginally increases the performance and thus may not be considered significantly better than just using the intensity feature, especially if one considers the additional processing and complexity introduced by doubling the number of features. In addition to being a skilled knowledge distiller, the student models have also shown evidence of improved generalization in comparison with the teacher; the student models are capable of distinguishing geohazards from noise artifacts and other seabed features (such as ridges) where the teacher can't (Figure 2).

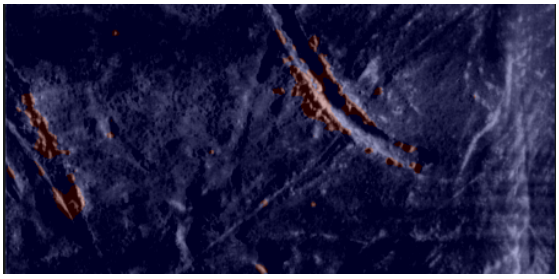
Overall the mean IoU values achieved by the model may seem relatively low overall, especially in terms of the application to commercial boulder detection. In semantic segmentation it is challenging to find a metric that can accurately quantify the accuracy of the labels, especially for applications such as boulder

Table 2: Test B mean IoU for all cross validation folds and model runs, which includes all feature combinations with confidence weights applied ($W = C$) or no weights applied ($W = 1$). Top row is the mean computed across all five k-folds for each student model.

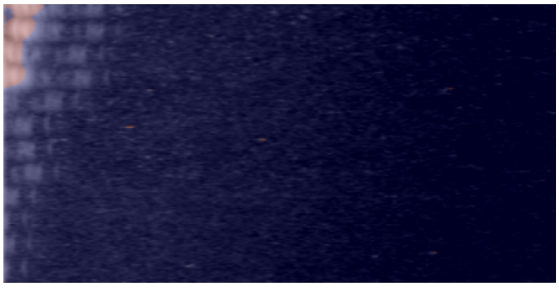
	dB		dB_bathy		dB_vu		dB_bathy_vu	
	$W = 1$	$W = C$	$W = 1$	$W = C$	$W = 1$	$W = C$	$W = 1$	$W = C$
Mean	0.78	**0.79**	0.79	**0.81**	0.78	**0.79**	0.78	**0.80**
k=0	0.80	0.78	0.76	0.82	0.79	0.80	0.81	0.81
k=1	0.75	0.80	0.74	0.81	0.77	0.77	0.79	0.78
k=2	0.79	0.77	0.79	0.79	0.78	0.78	0.77	0.81
k=3	0.80	0.82	0.79	0.80	0.79	0.79	0.77	0.82
k=4	0.79	0.77	0.80	0.81	0.78	0.80	0.77	0.77



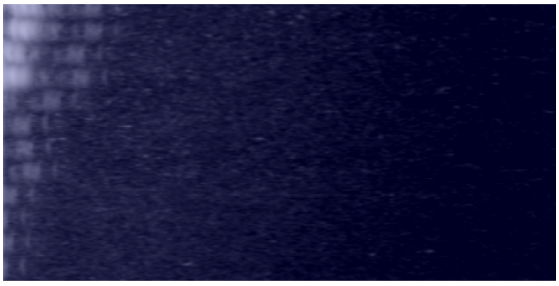
(a) Teacher



(b) Student (dB feature only)

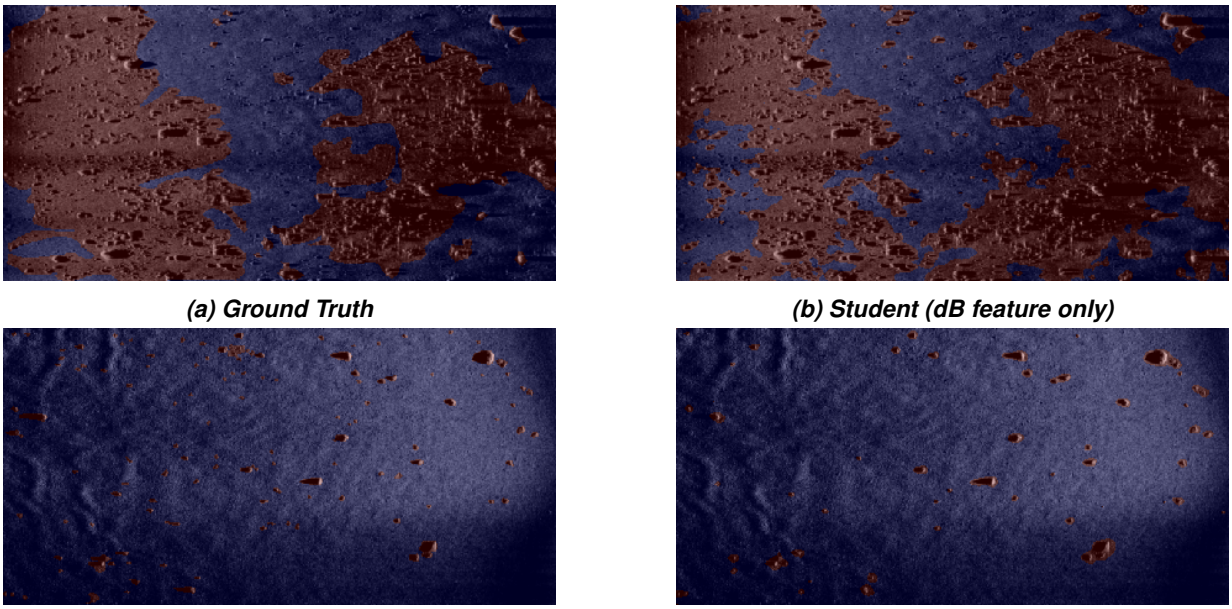


(c) Teacher



(d) Student (dB feature only)

Figure 2: Comparison of predictions from teacher model (left) and student model (right) on a SAS image featuring ridges and scours as well as geohzards (top) and an image featuring noise artifacts (bottom). SAS intensity images are overlaid with model predictions, where red is the target class and blue is the background class.



(a) Ground Truth ***(b) Student (dB feature only)***

(c) Ground Truth ***(d) Student (dB feature only)***

Figure 3: Comparison of ground truth labels (left) with predictions from dB student model (right). SAS intensity images are overlaid with model predictions, where red is the target class and blue is the background class.

detection where there may be many adjacent or overlapping small objects, and thus the exact boundaries of the label can be subjective and although quantitatively different, make no real difference qualitatively. For example in Figure 3 (top), there is no qualitative difference in the teacher and student labels, they both effectively capture the regions containing boulders; however the mean IoU score between these two segmentation labels is 0.68. Thus we can still get high quality predictions, even if the score seems quantitatively low. Mean IoU and other similar metrics such as dice score are highly sensitive to small errors when a given class represents a small portion of the image, causing the metric to remain low, while still providing qualitatively satisfactory labels. For example, if an image label has only a single pixel for a class but the model predicts three pixels for that class the IoU for that class would be 0.33. This can have a significant impact on our dataset, where some images may only have a few boulders scattered across the entire image, resulting in a limited number of target class pixels. This is demonstrated in Figure 3 (bottom), these labels are qualitatively similar but have a mean IoU of 0.74.

4 CONCLUSION

Using the proxy-label technique, we can improve model performance by up to 5 % while reducing the model size by 98%. The inclusion of the teacher confidence weights in the student model consistently improved the student performance by 1-4%. The inclusion of additional features (bathymetry or vertical uncertainty) did not significantly improve the model performance. It is likely that the resolution of these additional features are too low to provide useful features for detecting small objects. We observed evidence that the proxy-label technique helped increase model generalization across all model runs. The teacher network often falsely identified noise artifacts and portions of seabed features such as ridges as geohazards; however, the student network was able to generalize and distinguish noise and other seabed features from geohazards. Future work will focus on implementing and comparing variations of

the proxy-label technique demonstrated in the literature, which could include using the student network to update the teacher network as well as applying the original pseudo-label technique. Future work may also explore further knowledge distillation.

REFERENCES

1. Gitta von Rönn, Klaus Schwarzer, Hans-Christian Reimers, and Christian Winter. Limitations of boulder detection in shallow water habitats using high-resolution sidescan sonar images. *Geosciences*, 9(9):390, 2019.
2. Johnny Chen and Jason E. Summers. Deep convolutional neural networks for semi-supervised learning from synthetic aperture sonar (sas) images. *Proceedings of Meetings on Acoustics*, Jun 2017.
3. Oscar Bryan, Tom S. Haines, Alan Hunter, Roy Edgar Hansen, and Narada Warakagoda. Automatic recognition of underwater munitions from multi-view sonar surveys using semi supervised machine learning: A simulation study. *Proceedings of Meetings on Acoustics*, Jun 2022.
4. Massih-Reza Amini, Vasilii Feofanov, Loic Pauletto, Emilie Devijver, and Yury Maximov. Self-training: A survey. *arXiv:2202.12040 [cs.LG]*, 2023.
5. Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, July 2013.
6. Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.
7. Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv:2006.05278 [cs.LG]*, 2020.
8. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
9. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention-MICCAI 2015*, 2015.
10. Shannon-Morgan Steele. A unified semantic segmentation and object detection framework for synthetic aperture sonar imagery. *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, June 2023.
11. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
12. John McKay, Isaac Gerg, Vishal Monga, and Raghu Raj. What's mine is yours: Pretrained cnns for limited training sonar atr. *Proceedings of the OCEANS 2017 - Anchorage Confrence*, 2017.
13. Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
14. Jeremy Dillon, Shannon-Morgan Steele, Richard Charron, David Shea, Nathan Smith, Jan Albiez, and Alex Duda. Synthetic aperture sonar nadir gap coverage with centimetric resolution. *Global Oceans 2020: Singapore – U.S. Gulf Coast*, Oct 2020.
15. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.