Modality Matters: A Sim-to-Real Study of Sonar-Based Object Detection and Tracking

Eric Huynh, and Paul Robinette
Department of Electrical and Computer Engineering
University of Massachusetts Lowell
quochuy_huynh@student.uml.edu, paul_robinette@uml.edu

***Abstract*— Deep learning for underwater object detection and tracking using sonar remains challenging due to limited labeled data and the noisy, modality-dependent nature of acoustic imagery. This paper presents a sim-to-real evaluation pipeline using two model architectures: YOLOv8 for detection and SiamRPN++ for tracking. Both models are pretrained on synthetic sonar data generated in the HoloOcean simulator and fine-tuned on real-world data from two types of sonar, a Ping360 scanning imaging sonar and an Oculus M750d multibeam sonar. We evaluate performance using mAP@50 for detection and average IoU for tracking. Results show that modality has a significant impact on generalization: YOLOv8 benefits from synthetic pretraining but struggles with domain shifts between sonar types, while SiamRPN++ offers more stable localization under noise but requires more diverse data to perform well. These findings underscore the need for modality-aware training and evaluation strategies when deploying sonar perception models in real-world underwater environments.**

I. Introduction

Robust perception is critical for underwater robotics tasks such as inspection, manipulation, and search-and-rescue, where visibility is often limited. In such conditions, optical sensors rapidly degrade in effectiveness in long range, making acoustic sensing a preferred modality [1]. In this work, we focus on a representative target object known as the task box (Fig. 1c), which is designed to simulate real-world underwater interaction tasks including valve turning, line cutting or attaching, and button pushing. The broader goal of our system is to detect the task box from long range using sonar, approach it using path planning, and then switch to vision-based perception once the object is sufficiently visible for precise manipulation.

However, sonar imagery poses unique challenges: it is often low in resolution, subject to beam-pattern distortions, and exhibits significant variability across different sensor types. These factors hinder the direct application of standard computer vision methods and complicate the use of data-hungry deep learning models.

Early sonar perception methods rely on hand-crafted features such as edge detection or template-matching techniques, which perform poorly under cluttered or noisy conditions. More recently, deep learning-based approaches such as YOLO have been adapted to sonar and demonstrated improved detection rates compared to traditional methods [2], [3]. Nonetheless, these methods operate on individual frames and often struggle when the signal-to-noise ratio (SNR) is low or object boundaries are ambiguous [4]–[6].

Tracking-based methods, such as Siamese Region Proposal Networks (SiamRPN) [7], offer an alternative by leveraging temporal consistency to maintain target localization across frames.

This paper presents a comparative study of detection-based (YOLOv8) and tracking-based (SiamRPN++) deep learning models in the context of underwater sonar imagery. We propose a sim-to-real pipeline in which both models are first trained on synthetic sonar images generated using the HoloOcean simulator [8] and then fine-tuned on real-world data collected using two distinct sonar types: a Ping360 scanning imaging sonar [9] and an Oculus M750d multibeam sonar [10]. By comparing the performance across both sensor modalities and model paradigms, we aim to highlight the importance of key design choices for future underwater perception systems.

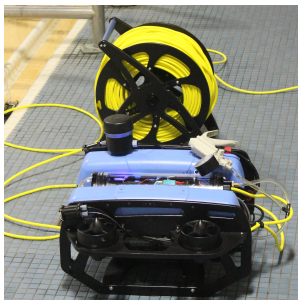
The rest of the paper is structured as follows. A short discussion of related works is provided in Sec.II. Methodology is described in Sec.III. Experimental results are presented in Sec.IV. Sec.V concludes the paper.

II. Related Works

Approaches such as edge detection and template matching perform adequately in controlled settings but struggle significantly with noisy acoustic data [1]. Valdenegro-Toro demonstrated that CNN-based patch matching vastly outperforms classical keypoint methods such as SIFT and SURF on forward-looking sonar imagery [11].

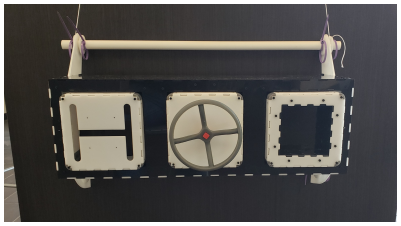
The advancement of deep learning leads to a growing number of sonar-specific adaptations of object recognition and detection models. Valdenegro-Toro et al. applied CNNs to target classification in forward-looking sonar and achieved around 99% accuracy in constrained datasets [12]. Similarly, Neupane and Seok surveyed deep-learning methods for sonar automatic target recognition (ATR) and noted strong performance in supervised settings. However, sonar-based deep learning models often struggle due to the lack of large annotated datasets and challenges from domain shifts [13].

Despite the success in sonar detection, models focusing on tracking, particularly in noisy, low-SNR environments, remain largely unexplored. In optical domains, Siamese Region Proposal Networks (SiamRPN) [7], SiamMask [14]



(a) BlueROV2

(b) DeepTrekker Revolution



(c) Task box

Fig. 1: Robotic platforms and task box used in this work.

and DaSiamRPN [15] have achieved state-of-the-art performance. These trackers leverage temporal consistency and template matching to maintain object identity across frames, which suggests strong potential for sonar-based application.

Beyond detection, large-scale sonar datasets are emerging. The UATD dataset by Xie et al. contains over 9,000 multibeam sonar images with ten classes of underwater objects [16]. Others like AquaYOLO have begun tailoring YOLOv8 for sonar imagery by incorporating residual blocks and spatial feature fusion modules [17].

This paper extends this body of work by directly comparing detection-based (YOLOv8) and tracking-based (SiamRPN++) deep learning models within a sim-to-real context across two different sonar modalities (scanning vs multibeam).

III. Methodology

A. Datasets

We use a combination of synthetic and real-world datasets to train and evaluate the models. Synthetic data is generated from the HoloOcean simulator, which provides realistic multibeam sonar images in a controlled pool-like environment. We generate 280 train, and 71 test images.

For real-world evaluation, two datasets are collected:

- **Ping360 Dataset:** The dataset was acquired using a BlueROV2 equipped with a Ping360 scanning imaging sonar. It consists of 40 training images and 5 test images. At a resolution of 1 degree per beam, the sonar completes a full 360-degree scan in under 40 seconds. We also experimented with a lower resolution of 4 degrees per beam; however, the result scans were significantly noisier, and the task box appearance became too degraded to serve as effective training data for object detection models.
- **Oculus Dataset:** Collected using a DeepTrekker Revolution equipped with an Oculus M750d multibeam sonar. It contains 40 train, and 5 test images. Unlike Ping360 sonar, the multibeam architecture enables near real-time scanning and allows us to acquire a large volume of data. In total, over 6,400 scans were recorded. However, to maintain consistency with the Ping360 dataset size, we randomly selected a subset of 45 images for training and evaluation.

The real-world datasets were captured in an indoor pool environment. To standardize annotation, all sonar frames were manually labeled using bounding boxes around the task box.

B. Model Initialization and Training

YOLOv8: We use a YOLOv8 model from a COCO-pretrained checkpoint (yolov8n.pt) as a detection backbone. The model is first adapted to synthetic sonar imagery by pretraining on HoloOcean data for 50 epochs using an image resolution of 512×512. We then perform separate fine-tuning runs on the Ping360 and Oculus datasets to evaluate the effect of sonar modality. During training, we applied standard YOLOv8 augmentations, including horizontal and vertical flipping, 90° rotations (clockwise, counter-clockwise, and upside down), and small-angle rotations in the range of ±15°. After augmentation, we have 93 train, 9 validation, and 5 test images for each of the real-world dataset. No additional hyperparameter tuning was performed. This straightforward training pipeline allows us to assess the model's ability to generalize from synthetic to real sonar imagery.

SiamRPN++: The SiamRPN++ tracking model is initialized with pretrained weights derived from the VID, YOUTUBEBB, DET, and COCO datasets. The architecture consists of a ResNet-50 backbone and a depthwise region proposal network. Like YOLOv8, the tracker is pretrained on HoloOcean and then fine-tuned independently on Ping360 and Oculus datasets. The backbone was partially frozen, with only the later layers (layer3 and layer4) fine-tuned after a specified number of epochs. The model was trained for 30 epochs for HoloOcean dataset, 20 epochs for Ping360 and Oculus dataset with a batch size of 16 using SGD with momentum and weight decay. A logarithmic learning rate schedule was applied, starting from 0.001 and decaying to 0.0001. Data augmentations included spatial shifts, scaling, and Gaussian blur on both the template and search images. The model was initialized from a pretrained checkpoint and trained end-to-end with multi-layer RPN heads and adjustment layers.

C. Evaluation Metrics

We report results using two distinct metrics tailored to the detection and tracking tasks:

- **Mean Average Precision at 0.5 IoU (mAP@50):** Used to evaluate object detection performance in YOLOv8. A detection is considered correct if the IoU with ground truth exceeds 0.5. This metric is standard for evaluating the precision-recall tradeoff in object detectors.

- **Average Intersection over Union (IoU):** Used to assess the SiamRPN++ tracker, which does not natively produce classification scores required for precision-recall curves. We compute frame-wise IoU between predicted and ground truth bounding boxes, then average across the sequence. This provides a direct measure of spatial alignment quality.

Performance is reported separately for each dataset (HoloOcean, Ping360, Oculus), and comparisons are made to assess the impact of modality-specific fine-tuning and domain shift.

IV. Experimental Results

A. Training Behavior

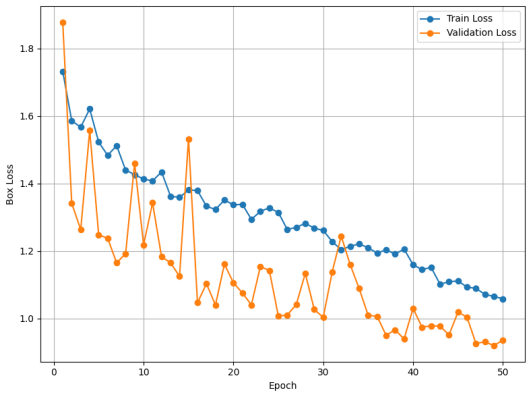


Fig. 2: YOLOv8 model train and validation loss for the HoloOcean dataset.

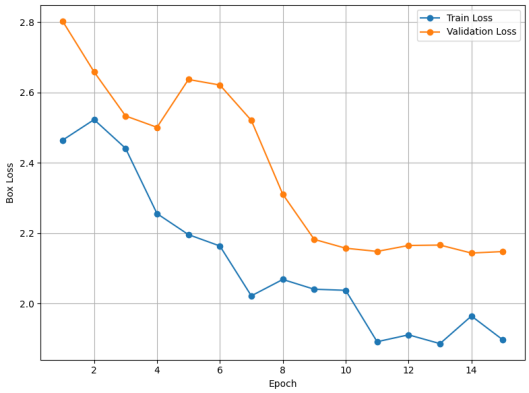


Fig. 3: YOLOv8 model train and validation loss for the Ping360 dataset.

The training loss curves underscore the varying difficulty of each dataset. On the HoloOcean dataset, the YOLOv8 model exhibits smooth convergence, thanks to the large quantity of clean synthetic data. In contrast, fine-tuning on the Ping360 dataset leads to overfitting around epoch 9, despite a short training duration of only 20 epochs.

For the Oculus dataset, while the training loss continues to decline, the validation loss plateaues, which signals

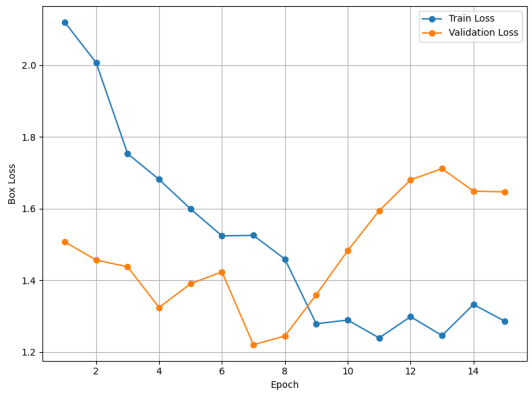


Fig. 4: YOLOv8 model train and validation loss for the Oculus dataset.

overfitting. These trends reveal the challenge of learning meaningful features from small real-world sonar datasets, especially when faced with low SNR and inconsistent target appearances.

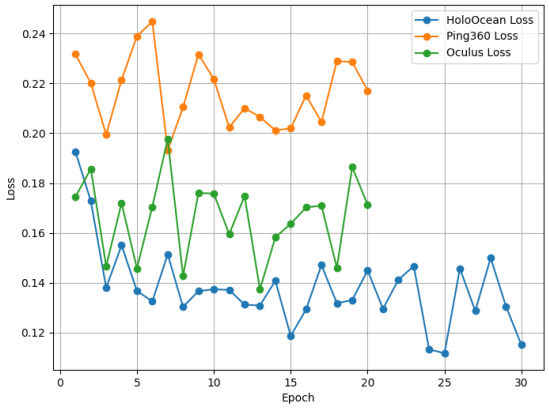


Fig. 5: Train loss for theHoloOcean, Ping360, and Oculus dataset with SiamRPN++.

As shown in Figure 5, the training loss curves for SiamRPN++ across the HoloOcean, Ping360, and Oculus datasets reveal limited convergence behavior. While the model achieves relatively low and stable loss on the synthetic HoloOcean data, the loss on the Ping360 and Oculus datasets remains high and fluctuates significantly across epochs. This indicates that the model struggles to fit the real-world data, particularly for the Ping360 dataset, where the loss shows no consistent downward trend. Despite SiamRPN++'s temporal modeling capabilities, the training dynamics suggest early signs of overfitting due to limited labeled data. These results highlight the need for more robust regularization or expanded datasets to support effective training in real sonar domains.

B. Quantitative Results

Table I summarizes the performance of YOLOv8 on both simulated and real sonar datasets. The model demonstrates

TABLE I: YOLOv8 object detection performance (mAP@50) across datasets.

Model	HoloOcean	Ping360	Oculus
Pretrained YOLOv8	0.01	0.00	0.00
YOLOv8 (HoloOcean)	**0.99**	0.00	0.37
YOLOv8 (HoloOcean + Ping360)	0.91	**0.20**	0.08
YOLOv8 (HoloOcean + Oculus)	0.98	0.01	**0.83**

strong detection accuracy on the synthetic HoloOcean data. When fine-tuned on real-world data, it generalizes moderately to the Ping360 dataset (achieving 0.20 mAP@50), but performs significantly better on the Oculus dataset with a score of 0.83 mAP@50.

This discrepancy stems from the difference in sonar modality. The Oculus and HoloOcean sonars are both multibeam systems that produce rich spatial structure and preserve object features that support generalization. In contrast, Ping360 is a mechanically scanning sonar with sparser, noisier returns that obscure object shape and make pattern recognition difficult.

Interestingly, the fine-tuned model on Ping360 data slightly improves its performance but causes a significant drop on the Oculus dataset. This suggests that the model has forgetten what it learned prior due to the dissimilarity in signal characteristics. This is a clear indication of domain shift. The model fine-tuned on Oculus data generalizes better to HoloOcean but performs poorly on Ping360, again due to the differences in sonar representation.

The results highlight that models trained on multibeam sonar data struggle when applied to a mechanical scanning sonar and vice versa. This underscores the need for sensor-aware training and data collection strategies in sim-to-real perception pipelines.

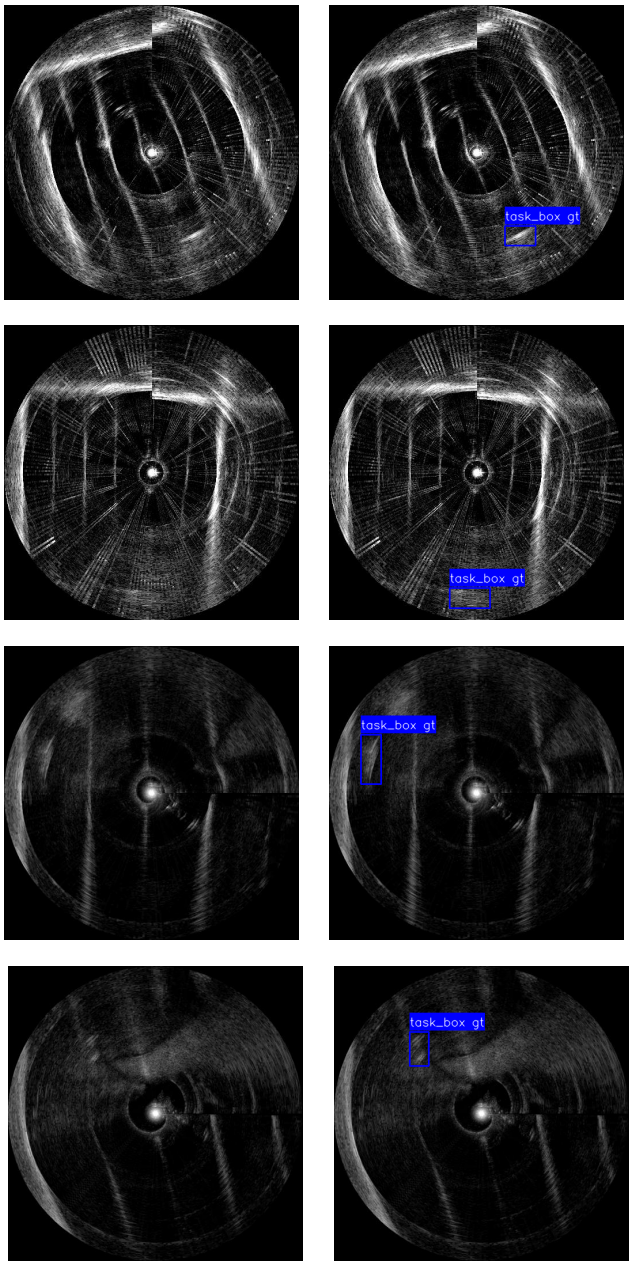
TABLE II: SiamRPN++ tracking accuracy (average IoU) across sonar datasets.

Model	HoloOcean	Ping360	Oculus
Pretrained SiamRPN++	0.11	0.22	0.37
SiamRPN++ (HoloOcean)	0.22	**0.26**	0.2
SiamRPN++ (HoloOcean+Ping360)	**0.28**	0.24	0.27
SiamRPN++ (HoloOcean+Oculus)	0.26	0.08	**0.41**

As shown in Table II, SiamRPN++ also achieves its highest performance on the Oculus dataset when fine-tuned with relevant data (IOU = 0.41). Performance on the HoloOcean dataset remains relatively stable across configurations, which suggests the model retains knowledge of the synthetic domain even after real-world fine-tuning.

However, performance on the Ping360 dataset is consistently lower. Unlike multibeam sonar, Ping360's noisier returns and lack of consistent geometric cues hinder the model's ability to track the object over time. Additionally, variations in sonar configurations and viewpoints as demonstrated in Figure 6 cause the appearance of the task box to change drastically and make generalization difficult with such limited training data.

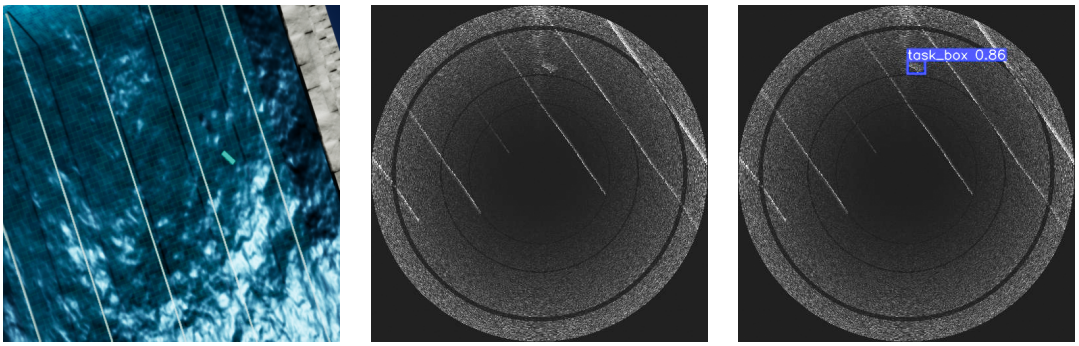
This result highlights the model's sensitivity to dataset variability and suggests that SiamRPN++ may require either larger, more diverse training data or input pre-processing to handle distortions in sonar tracking tasks.



(a) Ping360 sonar images (b) Task box ground truth

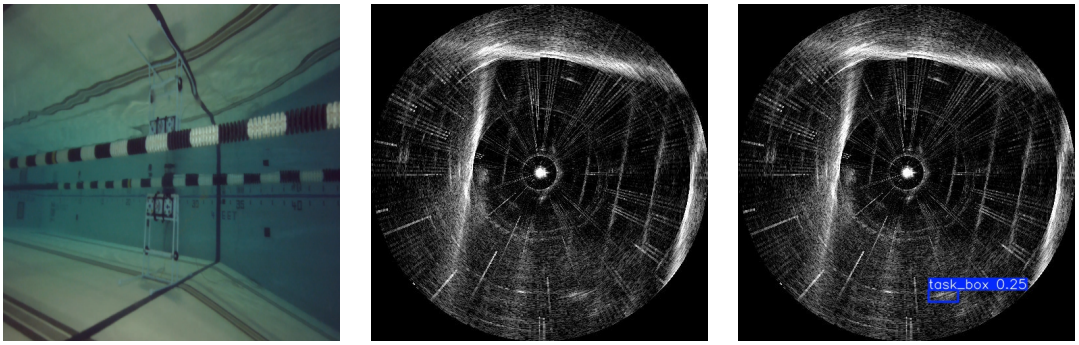
Fig. 6: Variations in range and viewpoint of Ping360 sonar dataset.

From both experiments, we observe that the choice of sonar modality significantly affects sim-to-real performance. To reduce the domain shift seen with the HoloOcean dataset and the Ping360 sonar dataset, we hypothesize that generating more representative synthetic data that closely matches the visual and noise characteristics of the Ping360 scans would improve model performance. This does not require simulating the full mechanical scanning process at 1-degree increments, but rather capturing key properties such as the radial beam pattern, sparsity, and characteristic noise artifacts



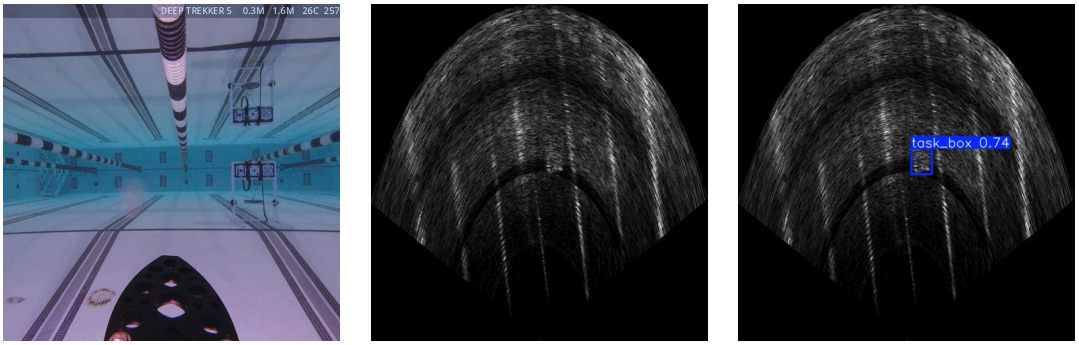
(a) Simulated environment (b) Simulated sonar image (c) Object detection result

Fig. 7: HoloOcean simulated environment and corresponding sonar image and detection result with YOLOv8.



(a) BlueROV2 view (b) Ping360 sonar image (c) Successful object detection

Fig. 8: Successful task box detection using fine-tuned YOLOv8 with Ping360 scanning sonar.



(a) DeepTrekker view (b) Oculus sonar image (c) Successful object detection

Fig. 9: Successful task box detection using fine-tuned YOLOv8 with Oculus multibeam sonar.

unique to the Ping360 or other mechanical scanning sonars. Developing a synthetic dataset with these traits would be an important step towards closing the gap between simulation and real-world sonar perception.

C. Qualitative Results

Figures 7–9 illustrate the qualitative performance of the YOLOv8 detection models across simulated and real-world sonar data.

Figure 7 shows a result from the HoloOcean simulator. The simulated indoor pool environment (Figure 7a) is used to generate synthetic sonar images (Figure 7b). The model accurately localizes the task box with a high-confidence prediction (Figure 7c).

In Figure 8, we evaluate the fine-tuned model on real-world data captured by the BlueROV2 platform with a Ping360 scanning imaging sonar. The ROV's onboard view is shown in Figure 8a, while the corresponding sonar image appears in Figure 8b. The model successfully detects the task box in Figure 8c.

Figure 9 demonstrates detection results using Oculus multibeam sonar on a DeepTrekker Revolution robot. The visual scene is shown in Figure 9a, and the corresponding sonar image in Figure 9b. The model achieves a precise detection of the target object in Figure 9c.

V. Conclusion

This study presents a sim-to-real evaluation of sonar-based object detection and tracking models trained on synthetic and real-world acoustic imagery. We compare detection-based (YOLOv8) and tracking-based (SiamRPN++) models across two sonar modalities, Ping360 scanning imaging sonar and Oculus M750d multibeam sonar, and analyze their performance under domain shift.

Our results show that models trained on synthetic multibeam sonar can generalize effectively to real multibeam sonar like Oculus but struggle with the Ping360 scanning imaging sonar data. The primary limiting factor is not dataset size, but sensor modality: multibeam sonars preserve shape and spatial continuity, whereas scanning sonars produce sparse and noisy representations.

This highlights that successful sim-to-real transfer in underwater perception depends critically on sensor compatibility. Careful selection of sonar hardware and modality-aware training pipelines are essential for deploying robust learning-based systems in underwater environments.

VI. Future Work

To address current limitations, we plan to substantially expand both real-world datasets especially the Ping360 sonar dataset. We aim to collect an additional 100–200 labeled samples. As described in Sec. III, it takes approximately 40 seconds to complete a single 360-degree scan with the Ping360 sonar. After each scan, the robot is repositioned to a new location or orientation, a process that typically takes at least one minute in total per scan. As such, acquiring 200 additional Ping360 scans would require approximately 200 minutes, or over three hours, of continuous operation. This will enable a more thorough analysis of how data volume and diversity impact generalization and domain adaptation.

We also intend to explore sonar-specific preprocessing techniques, such as beam pattern compensation, speckle noise filtering, and deep learning-based denoising. These enhancements could improve feature quality and model robustness, particularly for low-SNR scenarios like those seen with Ping360.

Ultimately, our goal is to determine whether mechanical scanning sonars can support end-to-end learning-based pipelines or are better suited for auxiliary roles. The outcome will guide future efforts toward developing a unified sonar perception system optimized for real-world autonomous underwater applications.

Acknowledgements

This work has been supported in part by the Office of Naval Research (N00014-21-1-2582 and N00014-23-1-2744).

References

[1] Y. Steiniger, D. Kraus, T. Meisen, "Survey on deep learning based computer vision for sonar imagery," *Engineering Applications of Artificial Intelligence*, vol. 114, 2022s.

[2] X. Cui, J. Zhang, L. Zhang, Q. Zhang, and J. Han, "Small object detection in side-scan sonar images based on SOCA-YOLO and image restoration," *Frontiers in Marine Science*, vol. 12, p. 1542832, 2025.

[3] Y. Sun and B. Yin, "CCW-YOLOv5: A forward-looking sonar target method based on coordinate convolution and modified boundary frame loss," *PLOS ONE*, vol. 19, no. 6, p. e0300976, 2024.

[4] X. Wang, Z. Zhang, and X. Shang, "Research on Improved YOLO11 for Detecting Small Targets in Sonar Images Based on Data Enhancement," *Applied Sciences*, vol. 15, no. 12, p. 6919, 2025.

[5] Z. Chen, G. Xie, X. Deng, J. Peng, and H. Qiu, "DA-YOLOv7: A Deep Learning-Driven High-Performance Underwater Sonar Image Target Recognition Model," *Journal of Marine Science and Engineering*, vol. 12, no. 9, p. 1606, 2024.

[6] K. S. Qin, D. Liu, F. Wang, J. Zhou, et al., "Improved YOLOv7 model for underwater sonar image object detection," *Journal of Visual Communication and Image Representation*, vol. 100, p. 104124, 2024.

[7] B. Li et al., "High performance visual tracking with Siamese region proposal network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] E. Potokar, S. Ashford, M. Kaess, and J. Mangelson, "HoloOcean: An Underwater Robotics Simulator," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2022, pp. 3040–3046.

[9] Blue Robotics, "Ping360 Scanning Imaging Sonar," https://bluerobotics.com/store/sensors-sonars/ping360-sonar/, accessed May 2025.

[10] Blueprint Subsea, "Oculus M750d Multibeam Imaging Sonar," https://www.blueprintsubsea.com/Oculus/, accessed May 2025.

[11] M. Valdenegro-Toro, "Improving sonar image patch matching via deep learning," arXiv preprint arXiv:1709.02150, 2017.

[12] M. Valdenegro-Toro, "Object recognition in forward-looking sonar images with convolutional neural networks," in *OCEANS 2016 MTS/IEEE Monterey*, 2016.

[13] D. Neupane and J. Seok, "A review on deep learning–based approaches for automatic sonar target recognition," *Electronics*, vol. 9, no. 11, p. 1972, 2020.

[14] Q. Wang et al., "Fast online object tracking and segmentation: A unifying approach," in *CVPR*, 2019.

[15] Z. Zhu et al., "Distractor-aware Siamese networks for visual object tracking," in *ECCV*, 2018.

[16] K. Xie et al., "A dataset with multibeam forward-looking sonar for underwater object detection," *Scientific Data*, vol. 9, no. 739, 2022.

[17] Y. Lu, J. Zhang, Q. Chen, C. Xu, M. Irfan, and Z. Chen, "AquaYOLO: Enhancing YOLOv8 for Accurate Underwater Object Detection for Sonar Images," *Journal of Marine Science and Engineering*, vol. 13, no. 1, p. 73, 2025.