ODYSSEE: Oyster Detection Yielded by Sensor Systems on Edge Electronics

Xiaomin Lin[1]*, Vivek Mange[2]*, Arjun Suresh[1]*, Bernhard Neuberger[3], Aadi Palnitkar[1],
Brendan Campbell[4], Alan Williams[5], Kleio Baxevani[2], Jeremy Mallette[8], Alhim Vera[6]
Markus Vincze[3], Ioannis Rekleitis[7], Herbert G. Tanner[2], Yiannis Aloimonos[1]

***Abstract*— Oysters are a vital keystone species in coastal ecosystems, providing significant economic, environmental, and cultural benefits. As the importance of oysters grows, so does the relevance of autonomous systems for their detection and monitoring. However, current monitoring strategies often rely on destructive methods. While manual identification of oysters from video footage is non-destructive, it is time-consuming, requires expert input, and is further complicated by the challenges of the underwater environment.**

To address these challenges, we propose a novel pipeline using stable diffusion to augment a collected real dataset with photorealistic synthetic data. This method enhances the dataset used to train a YOLOv10-based vision model. The model is then deployed and tested on an edge platform; Aqua2, an Autonomous Underwater Vehicle (AUV), achieving a state-of-the-art 0.657 mAP@50 for oyster detection.

I. INTRODUCTION

Oyster reefs are crucial benthic environments, providing numerous ecosystem services, including water filtration, enhanced species richness, shoreline stabilization, and velocity attenuation [1]–[3]. Unfortunately, standing oyster stocks in areas such as the Chesapeake Bay [4] and the North Sea [5] have decreased significantly due to over-fishing, global warming, and disease. To mitigate this devastating ecological loss, massive efforts are being made to monitor and restore oyster habitats across Europe [5], Aisa [6], [7] and the United States [8]–[11].

Despite this effort, efficiently monitoring oyster beds for guided and enhanced restoration remains a major challenge [8], [12], [13]. Traditional monitoring methods, such as those conducted annually in Delaware Bay, New Jersey [14], and biannually in Chesapeake Bay, Maryland [15], rely heavily on destructive sampling techniques (dredging or tonging) and trained technicians to process samples. Among these reports, universal parameters such as “reef areal dimensions, reef height, oyster density, and oyster size-frequency distribution” are commonly assessed [5], [9]–[11], [16]. The overall process is costly and labor-intensive, limiting the scalability of surveying efforts.

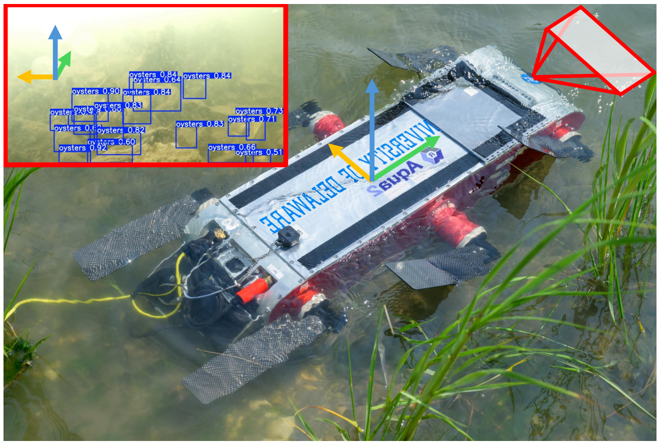

Fig. 1: Oyster detection system deployed on an Autonomous Underwater Vehicle (AUV). The main image shows oyster detection in real-time in a shallow marine environment on the Aqua2 robot. The inset highlights the output bounding boxes around the oysters detected in situ

Similar challenges exist in on-bottom oyster aquaculture, where larval oysters are set on clutch and left to grow for years before harvest. Despite global growth for this economic sector [17], the industry still relies on legacy equipment [18]. Advances in autonomy, which have improved terrestrial farming with real-time detection and precision techniques [19] could help underwater farming. By using these methods, farmers could better track oyster health, abundance, and size, optimize harvesting, and avoid areas with low yields or undersized oysters [20].

In fact, by collecting videos or photographs of the beds and assessing the quantity of live and dead oysters, a census of a localized region can be produced to determine the quality of the reef and the harvestable biomass. Similar video-based methods were successfully used to assess habitat provisioning in oyster reefs and aquaculture gear to provide valuable estimates of relative abundance, species composition, and behavior [21]–[24]. However, these

* Equal Contributors and Corresponding Authors.

[1] Maryland Robotics Center, University of Maryland, College Park, MD 20742, USA. Emails: {`xlin01`, `arjsur`, `apalnitk`, `jyaloimo`}`@umd.edu`.

[2] Center for Autonomous and Robotic Systems, University of Delaware, Newark, DE, 19711, USA. Emails: {`vivekm`, `kleiobax`, `btanner`}`@udel.edu`.

[3] Automation and Control Institute, TU Wien, 1040 Vienna, Austria. {neuberger, vincze}@acin.tuwien.ac.at

[4] School of Marine Science and Policy, University of Delaware, Lewes, DE, 19958, USA. Email: `bpc@udel.edu`.

[5] University of Maryland Center for Environmental Science, Horn Point Laboratory, Cambridge, MD 21613, USA Email: `awilliams@umces.edu`.

[6] College of Engineering and Applied Science, University of Cincinnati, OH 45221, USA. Email: `veragoaa@mail.uc.edu`.

[7] Molinaroli College of Engineering and Computing, University of South Carolina, Columbia, SC, USA. Email: `yiannisr@cse.sc.edu`

[8] Indepedendent Robotics, Montreal, QC, Canada. Email: `jeremy.mallette@independentrobotics.com`.

methods require highly trained individuals to manually identify and count animals over a large timespan, limiting the depth, scale, and applicability of monitoring for surveying. Autonomous systems can detect and assess live oysters in real time using vision models, offering a promising approach to monitoring. These systems enable on-site decision-making based on live feedback, but training such models requires large datasets of oyster imagery, which are often scarce.

To address this, we leverage diffusion models to generate photorealistic synthetic images, augmenting real datasets. These combined datasets are used to train models that are then deployed in autonomous underwater vehicles (AUVs) for real-time oyster population surveys. The models trained on these combined datasets are deployed on AUVs, enabling autonomous surveys in oyster fields, and providing accurate counts of oyster populations within their biomes. A key aspect of our approach is that the detection computation is performed in real-time onboard the AUV.

The contributions of this paper are as follows:

- The development and deployment of a state-of-the-art model for oyster detection with on edge implementation on the Aqua2 biomimetic AUV, to carry out tests on the oysters in real environment;
- A novel scheme for the generation of photorealistic oyster imagery;
- Open-sourcing the datasets and associated code to facilitate future research.

The remainder of this paper is organized as follows. First, we place this work in the context of previous work in Sec. II. Then, we describe the pipeline used to create synthetic images and the details of the edge device within Sec. III. We then present extensive quantitative and qualitative evaluations of our approach in Sec. IV. Finally, we conclude the paper in Sec. V with parting thoughts on future work.

II. RELATED WORK

Robotic monitoring of aquatic environments presents technical challenges due to low visibility, complex terrains, and the need for autonomy and real-time decision-making [25]–[27]. Edge computing, as a framework that brings data processing applications closer to data sources, can support robot autonomy in environmental monitoring missions [28], [29]. When implemented in autonomous underwater vehicles (AUVs), edge computing allows real-time and *local* data processing in support of a range of robotic missions, from object detection [30], [31] to autonomous navigation [32] and exploration [33] under resource constraints.

Within marine environmental monitoring, oyster detection, in particular, presents unique additional challenges due to the heavily occluded marine environment these creatures inhabit, which makes accurate manual labeling especially difficult. Sadrfaridpour et al. [34] used Mask-R-CNN [35] for oyster detection; however, their dataset was small and collected on an oyster farm, which lacked the variety of a wild reef setting —needed for robust detection in real-world environments. The collection and annotation of sufficient marine data, essential for achieving satisfactory results with vision models, has always been a challenge.

To save time and resources associated with collecting real data, we follow the conceptual approach proposed by Lin et al. [36], which uses 3D models to generate synthetic data. This technique has been successfully applied in various object detection tasks, such as whale detection with synthetic satellite images [37], aerial object detection [38], real and virtual segmentation [39], and olive detection [40]. We improved upon OysterNet [36] to generate realistic oyster assemblages and using stable diffusion for sim-to-real transfer.

While Generative Adversarial Networks (GANs) have been used for sim-to-real tasks such as underwater image restoration [41], synthetic oyster generation [36], and AUV pose estimation [42], diffusion models provide an attractive and often advantageous alternative. Indeed, diffusion models [43], which generate images by refining random inputs, have proven effective for high-quality image synthesis [44]–[46]. Recent advances, such as ControlNet [47], enable better control over the output by integrating external conditions like depth and segmentation map. This added control not only improves the fidelity of the generated images, but also expands the utility of diffusion models, enabling them to be used for tasks like synthetic data augmentation and domain-specific image generation.

Given the capability of these models to generate representative synthetic data, many have used diffusion models to aid in the training of vision models. He et al. [48] demonstrated the use of synthetic data in training vision models, stating its efficacy in improving the performance of models in zero-shot and low-shot environments. Bansal et al. [49] used synthetic data to achieve higher accuracy with ImageNet models, and Trabucco et al. Off-the-shelf stable diffusion models have also been used [50] to enhance existing data and apply it to few-shot domains. Despite the obvious advantages that diffusion models afford for dataset enhancement, few efforts attempt to create stable diffusion-enhanced datasets for the *underwater domain*.

Our work combines (i) edge computing for real-time underwater exploration and (ii) advanced synthetic data generation. While previous efforts faced challenges due to small and homogeneous datasets, our approach leverages diffusion models and geometric modeling to create diverse and realistic oyster datasets and deploy state-of-the-art machine learning models such as YOLOv10 [51] running locally on the Aqua2 AUV, ensuring robust performance with edge computation in real-world scenarios. The resulting pipeline is crucial to developing robust robust oyster detection systems and improving the effectiveness of autonomous, real-time monitoring of complex marine environments.

III. System Overview

To efficiently monitor oyster habitats in real time, we utilize the Aqua2 robotic platform (Figs 1, 3), designed for nondisruptive environmental monitoring tasks. We will

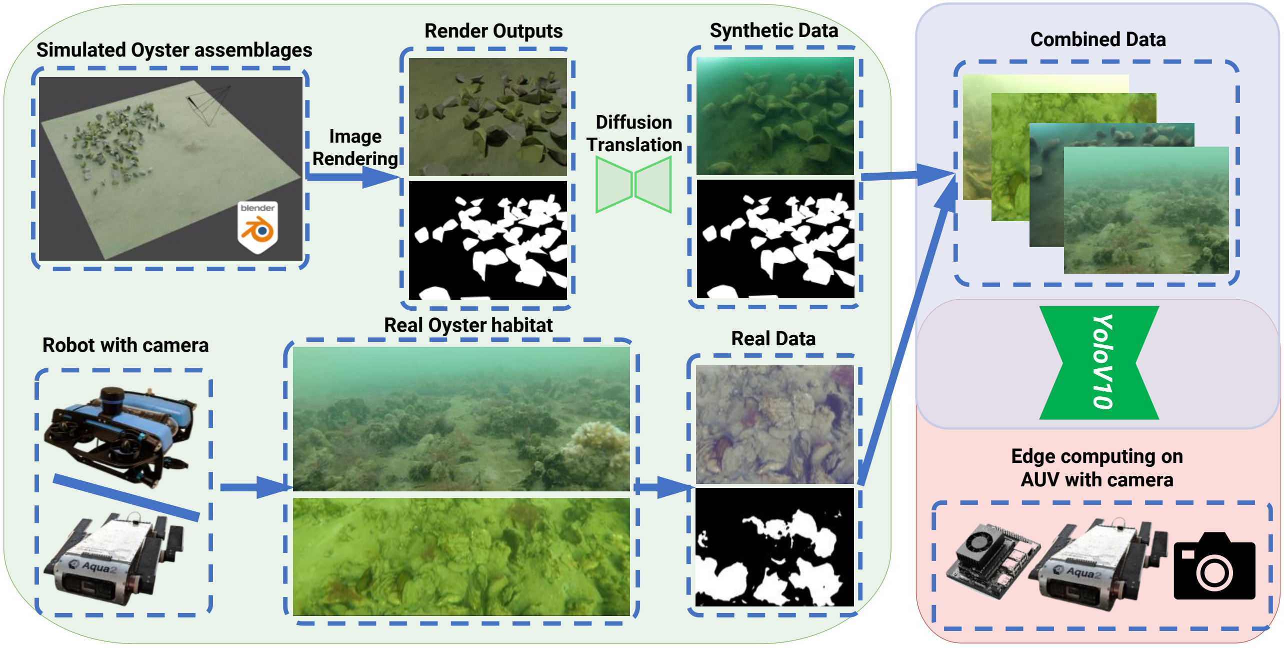

Fig. 2: Overview of the oyster detection system. Simulated oyster assemblages are rendered using Blender and paired with ground-truth segmentation masks. Diffusion models are employed to generate synthetic oyster imagery, ensuring consistency with real oyster habitats. Real-world data are collected by the Aqua2 and BlueROV, equipped with a camera for capturing oyster images in the field. The combined dataset is used to train a YOLOv10, which is deployed on an edge computing platform on Aqua2 for real-time oyster detection.

first provide an overview of the Aqua2 robot, followed by a detailed explanation of the software deployed on it. To further address the challenge of limited real-world oyster imagery in reef environments, we supplement our dataset with synthetic data, which will be discussed in the section III-C of this overview.

A. Hardware Overview

Aqua2 is a biomimetic hexapedal Autonomous Underwater Vehicle (AUV), designed for minimal environmental disturbance in sensitive habitats such as coral [52], [53] and oyster reefs [54]. Unlike traditional AUVs that use thrust-based propulsion, Aqua2 employs a unique biomimetic reciprocating paddle motion with its six fins, significantly reducing sediment disruption by avoiding thrust wash. This feature is crucial to maintain water clarity in underwater environments, as sediment plumes can obscure sensors and cameras [55]. The Aqua2 vehicle is equipped with a NVIDIA Jetson Xavier NX processor, which features a 384-core NVIDIA Volta GPU with 48 Tensor Cores, delivering up to 21 TOPS (Tera Operations Per Second) of performance at 15W of power. The processor is complemented by a 6-core NVIDIA Carmel ARM®v8.2 64-bit CPU, providing robust processing capabilities essential for complex, real-time image processing tasks. In addition, the vehicle is equipped with three 2-megapixel cameras optimized for low-light underwater imaging and with scene-based white balancing. The two front E-CAMs capture full HD video at 65 fps, while the rear Blue Robotics camera records at 30 fps. Both have focal lengths of about 3mm, but the front cameras offer a much wider field of view and overall better performance for surveying underwater environments. On a full load, Aqua2 is capable of running for 5 hours in real world environments.

B. Software Overview

Aqua2's software architecture leverages the Docker containers for easy management and deployment. NVIDIA's Container Runtime, compatible with the Open Containers Initiative (OCI), provides GPU access as outlined in III-A. Based on Dustynv/L4t-ml v-r36.2.0[1] container, we integrate Robot Operating System (ROS2), OpenCV, PyTorch v2.2.0 and Ultralytics. The system runs on ROS2 Humble middleware for real-time communication capabilities, modular design, and simplicity. OpenCV handles basic computer vision tasks and Ultralytics installs YOLOv10 and all its dependencies. We developed a custom ROS package that executes the YOLOv10 ROS node to establish the vision pipeline. During operation, Aqua2's vision pipeline captures images from a downward-facing camera. Pre-trained YOLOv10 submodels, cached on the robot, are quickly loaded to process and compare predictions, highlighting regions of interest and publish these images on the ROS network. These results can be analyzed in real-time by the operator for decision-making or used for autonomous monitoring tasks. With the hardware and software infrastructure in place, we now turn our attention to addressing the challenge of limited real-world oyster imagery by leveraging synthetic data generation to further enhance the model's performance.

[1] https://hub.docker.com/r/dustynv/l4t-ml

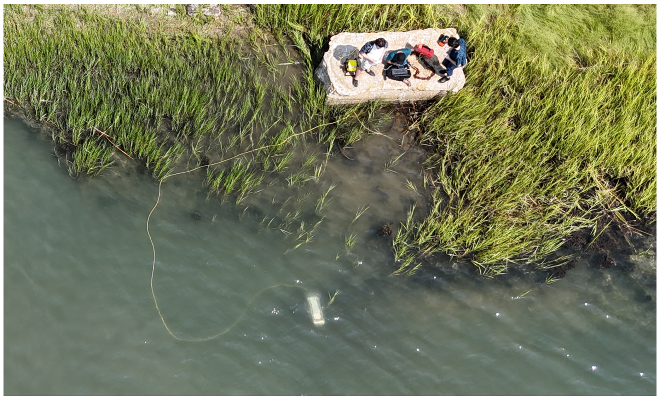

Fig. 3: Experimental deployment of the system, surveying a wild near-shore oyster reef in Lewes, DE. Operators can be seen teleoperating the Aqua2 and analyzing the feedback.

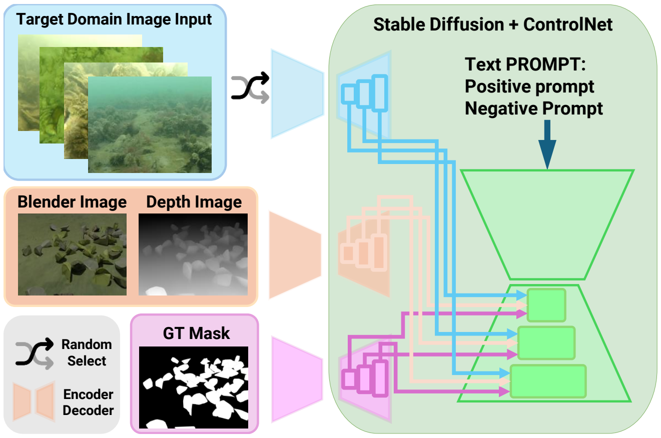

Fig. 4: Synthetic image generation pipeline using Stable Diffusion and ControlNet. ControlNet uses randomly sampled real underwater images combined with Blender-geneated images, depth images, and ground truth masks to ensure consistency. Stable Diffusion model, guided by text prompts, and ControlNet, refines the synthetic data to closely match real-world oyster environments for vision model training.

C. Synthetic Image Generation

Given the challenges of acquiring real-world data, synthetic image generation has become a valuable tool for extending datasets. By using diffusion models with ControlNet, we transform a synthetic dataset, comprising segmentation masks and depth images generated in Blender, into realistic oyster imagery that closely resembles the target underwater environment, enhancing vision model performance. This method effectively bridges the gap between synthetic and real-world data, producing more representative images while retaining the flexibility of synthetic generation. The data generation pipeline in Fig. 4 illustrates the ControlNet inputs on the left side, consisting of target domain images, the synthetic blender images, depth images and segmentation ground truth masks rendered with blender. For further guiadance a positive and negative prompt is provided. The details of the synthetic data generation pipeline are described in the following sections.

1) Simulation Image Rendering: To model the geometry of an oyster, we first 3D scanned ten oysters to build a mathematical model that approximates the shape in two stages: a 2D perimeter model and a stratified 3D model.

$$B_{i,0}(t) = \begin{cases} 1 & \text{if } t_{i+1} \geq t \geq t_i \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

$$B_{i,k}(t) = \tfrac{t-t_i}{t_{i+k}-t_i} B_{i,k-1}(t) + \tfrac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} B_{i+1,k-1}(t). \tag{2}$$

The 2D perimeter is represented using two cubic B-splines shows in (1) and (2) for the top and bottom shell halves. This model is then extended to 3D by adding depth changes across layers. High-frequency details are simplified for computational efficiency, with visual textures compensating for intricate surface variations. This approach provides a scalable way to generate realistic oyster models. More details of the modeling can be found in OysterNet [36]

The 3D modeled oysters are then randomly distributed within the Blender™ 3D game engine. The camera view was angled slightly to better emulate the point of view of an AUV observing the oysters in situ (Fig. 1), providing a more accurate visual representation of the AUV monitoring the oyster habitat.

The simulation is used to generate depth maps and the corresponding segmentation masks, which are used as the source domain in the synthetic dataset generation. Providing geometric consistency in ControlNet through the depth map indicates the locations of individual oysters within the segmentation mask.

2) Real Data Collection: To facilitate data collection, the BlueROV platform was equipped with a GoPro camera, and the Aqua2 robot was also employed to capture video of oysters in situ at Chesapeake Bay (Neavitt, MD; 38° 44’ 41.0”N, 76° 18’ 22.6”W) and Delaware Bay (Lewes, DE; 38°47’20.4”N 75°09’44.6”W) respctively. During the experiments, the robot was teleoperated to decouple the detection and classification results from the challenge of navigating around oyster structures, ensuring more precise data collection. From the video, high-quality frames were extracted and manually annotated by trained experts to identify the specific locations of the oyster within each frame. This data was subsequently combined with the earlier OysterNet dataset to obtain the final combined, fully real dataset of more than 2000 manually annotated images. Furthermore, real images are used as ControlNet inputs that represent the target domain to generate synthetic images.

3) Stable Diffusion and ControlNet: Images generated as detailed in Section III-C.1 are provided as the source domain and iteratively selected as ControlNet inputs. Depth maps and segmentation are kept in corresponding pairs such that the segmentation and geometry complement each other. Therefore, guiding the image generation to align with the desired structure and spatial characteristics. Additionally, four randomized reference images from the real data collection are used as image prompts for ControlNet to fine-tune the stylistic and environmental attributes of the generated images. A carefully selected text prompt ensures that Stable Diffusion focuses on the specific target object, namely, the oysters, and it includes negative prompts to mitigate certain biases in the diffusion models. Each generated image from the diffusion model is paired with the corresponding segmentation mask and then used as one of the ControlNet inputs. These masks serve to help generate annotations for the synthetic oyster images, enabling them to act as the ground truth.

Combining real and synthetic data ensures a sufficiently large dataset for training. With the system reviewed, we proceed to assess the performance of the system through a series of experiments.

IV. Experiments And Results

The mixed datasets obtained via the method described in III-C were then subsequently used to train all of the available YOLOv10 models and comparisons were run between these augmented models and the models which had only seen the real data.

A. Training Details

To train the various models, 4000 oyster images were synthetically generated as described in Section III-C with the size of 640×480, which were then combined with the existing dataset of 2025 real images of 1920×1680 resolution. To this end, we used only 30% of the real dataset for training combined with all the synthetic dataset. We tested our method on the remaining 70% of the real dataset to report our result. We train all networks for 300 epochs with an initial learning rate of 0.01 with a cosine annealing learning rate scheduler. The Adam optimizer was used with momentum set to 0.937 and a weight decay of 0.0005. To avoid overfitting. The training batch size was 16, and the model was trained on resized images to 640×640 pixels. In terms of data augmentation, we utilized several techniques, such as flipping, scaling, and random cropping. Close mosaic augmentation was applied for the first 10 epochs, with the mosaic probability set to 1.0. The model was further fine-tuned using automated augmentations, such as *randaugment* and random erasing with a probability of 0.4. During training, the object detection model was evaluated against a validation set, with an IoU threshold of 0.7. The maximum number of detections per image was capped at 300 to ensure efficient processing of the oyster habitat scenes.

B. Experimental Results

Extensive testing of various YOLOv10 submodels was carried out at the Marine Operation boat basin facility of the University of Delaware in Lewes, Delaware Fig. 3. This test focused on assessing the performance, accuracy and adaptability of these submodels in real-world scenarios, ensuring their robustness in handling the specific environmental and operational challenges presented by the location.

Lighter models like Yolov10-N achieved faster inference times and higher frequencies, while more complex models like Yolov10-X were slower. For example, Yolov10-N processed a frame in 55 ms with a frequency of 9.2Hz (Table Ia), while the Yolov10-S model, although more accurate, had a frequency of 8.9Hz with a processing time of 71 ms (Table Ib).

TABLE I: Quantitative Comparison of All YOLOv10 Sub-Model Performance

Model	Inference Time (ms)	Frequency (Hz)	AP_{50}	mAP50-95
YOLOv10-N	54.9	**9.2**	0.458	0.234
YOLOv10-S	71.0	8.9	0.557	0.314
YOLOv10-M	158.1	4.8	0.642	0.468
YOLOv10-B	214.8	3.8	0.645	0.416
YOLOv10-L	264.1	3.2	**0.657**	0.430
YOLOv10-X	375.2	2.3	0.642	0.424

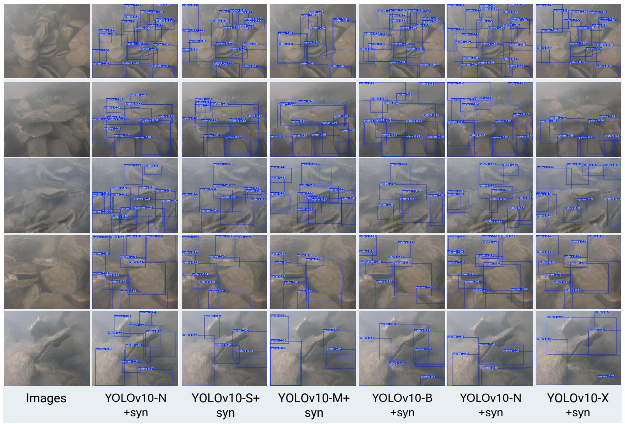

Fig. 5: Qualatative Comparison of All YOLOv10 Sub-Model Performance. *This image is best seen in color on a computer screen at 400% zoom.*

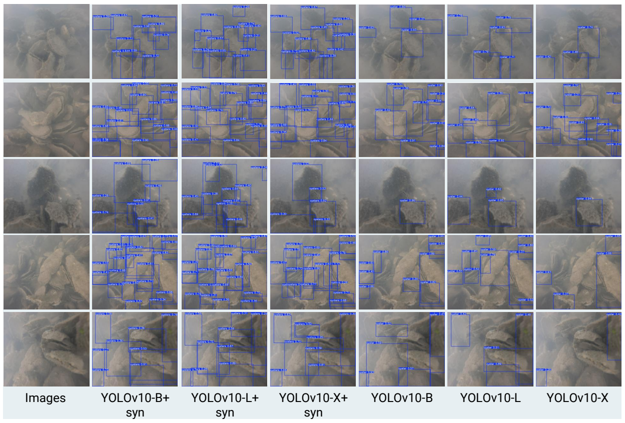

Fig. 6: Qualatative Comparison of YOLOv10-B, YOLOv10-L, and YOLOv10-X Models with and without Synthetic Data. *This image is best seen in color on a computer screen at 400% zoom.*

The YOLOv10-M model, suitable for general detection, processed frames in 158 ms at 4.8Hz (Table Ic). YOLOv10-B

TABLE II: Comparison of YOLOv10-B, YOLOv10-L, and YOLOv10-X Models with(S) and without(R) Synthetic Data. (S): the model is trained on combined synthetic and real data. (R): the model is trained on real data only

Model (S)	AP_{50}	mAP50-95	Model (R)	mAP50	mAP50-95
YOLOv10-B	0.645	0.416	YOLOv10-B	0.639	0.472
YOLOv10-L	**0.657**	0.430	YOLOv10-L	0.638	0.478
YOLOv10-X	0.642	0.424	YOLOv10-X	0.630	**0.480**

offered a good compromise, with a processing time of 214 ms and a frequency of 3.8Hz (Table Id). For more accuracy, YOLOv10-L took 264 ms per frame at 3.2Hz (Table Ie). Lastly, YOLOv10-X, the most computationally intensive model, was processed at 372 ms with a frequency of 2.3Hz (Table If).

The qualitative results in Fig. 5 from testing multiple YOLOv10 sub-models demonstrate a clear trend: generally, larger models achieve better detection accuracy, as indicated by higher mAP50 and mAP50-95 values. As depicted in Table I, YOLOv10-L outperforms smaller models like YOLOv10-N, with mAP50 scores of 0.657 compared to 0.458. However, this trend does not hold for the largest model, YOLOv10-X, where the performance slightly drops, suggesting that the size of the dataset might not be sufficient to fully leverage the complexity of the larger network. This indicates a point of diminishing returns with increasing model size, where the added complexity does not necessarily translate into better detection performance.

When comparing models trained with and without synthetic data in Fig. 6, it is evident that incorporating synthetic data improves the overall detection results across the board. As depicted in Table II, YOLOv10-L with synthetic data achieves an mAP50 of 0.657 compared to 0.638 without synthetic data. However, this improvement comes with a trade-off: the confidence in these detections, as reflected in the mAP50-95 scores, is slightly lower with synthetic data. This suggests that while synthetic data can enhance detection capabilities, it might introduce variability or noise that affects the precision of the model's predictions.

V. Conclusion and Future Work

In this work, we present a comprehensive system for real-time, on-edge, underwater oyster monitoring, which integrates advanced object detection models and synthetic data generation via stable diffusion on the Aqua2 underwater robotic platform. We demonstrated that systematically combining robotic capabilities with domain-specific synthetic data enhances model accuracy and detection performance for underwater environments. This system not only improves oyster monitoring efficiency but also establishes a framework for broader applications in marine environmental monitoring.

Looking ahead, the system can be improved both from hardware and software perspectives. On the hardware side, we could introduce additional sensor modalities, such as multi-beam ultrasound and single-beam ultrasound, for depth estimation, which would significantly enhance performance in murky water conditions. From a software standpoint, incorporating advanced navigation and exploration algorithms would enable the robot to autonomously navigate and perform surveillance tasks. Online learning algorithms could allow the system to adapt in real-time to different environmental conditions, such as variations in water color. Finally, introducing collaborative multi-robot systems would make monitoring more efficient and comprehensive, allowing for larger-scale underwater environmental monitoring.

Acknowledgment

This work was supported by USDA NIFA Sustainable Agriculture System Program under award number 20206801231805, NOAA's Project ABLE via award number NA22OAR4690620-T1-01, the NSF awards 1943205 and 2024741, the EU-program EC Horizon 2020 for Research and Innovation under grant agreement No. 101120823, project MANiBOT, and the US Army Corps of Engineers, ERDC Contracting Office under Contract No. W912HZ-22-2-0015. We thank Dr. Christopher Rasmussen and Dr. Arthur Trembanis for their support during the 2024 Autonomous Systems Bootcamp at the University of Delaware, Independent Robotics for assisting with Aqua AUV operations, and Rileigh Hudock, Dr. Edward Hale, and Max Collins for help with data collection.

References

[1] L. D. Coen, R. D. Brumbaugh, D. Bushek, R. Grizzle, M. W. Luckenbach, M. H. Posey, S. P. Powers, and S. G. Tolley, "Ecosystem services related to oyster restoration," *Marine Ecology Progress Series*, vol. 341, pp. 303–307, 2007.

[2] J. H. Grabowski and C. H. Peterson, "Restoring oyster reefs to recover ecosystem services," *Ecosystem engineers: plants to protists*, vol. 4, pp. 281–298, 2007.

[3] B. Campbell, C. Gurdisz, J. Testa, E. Hale, and M. W. Gray, "Assessing the spatial variability of cage movement and velocity attenuation of an off-bottom oyster farm and its influence on eastern oyster (crassostrea virginica) growth performance," *Available at SSRN 4858563*, 2024.

[4] R. I. Newell, "Ecological changes in chesapeake bay: are they the result of overharvesting the american oyster, crassostrea virginica," *Understanding the estuary: advances in Chesapeake Bay research*, vol. 129, pp. 536–546, 1988.

[5] B. Pogoda, "Current status of european oyster decline and restoration in germany," *Humanities*, vol. 8, no. 1, p. 9, 2019.

[6] W. Quan, R. Fan, Y. Wang, and A. T. Humphries, "Long-term oyster recruitment and growth are not influenced by substrate type in china: implications for sustainable oyster reef restoration," *Journal of Shellfish Research*, vol. 36, no. 1, pp. 79–86, 2017.

[7] M. J. Bishop, B. S. Lanham, J. R. Esquivel-Muelbert, V. J. Cole, K. M. Faelnar, C. Jenkins, J. Keating, F. Martínez-Baena, and W. A. O'Connor, "Oyster reef restoration-aquaculture interactions: maximizing positive synergies," *Frontiers in Marine Science*, vol. 10, p. 1162487, 2023.

[8] L. P. Baggett, S. P. Powers, R. D. Brumbaugh, L. D. Coen, B. M. DeAngelis, J. K. Greene, B. T. Hancock, S. M. Morlock, B. L. Allen, D. L. Breitburg, *et al.*, "Guidelines for evaluating performance of oyster habitat restoration," *Restoration Ecology*, vol. 23, no. 6, pp. 737–745, 2015.

[9] B. N. Blomberg, T. A. Palmer, P. A. Montagna, and J. B. Pollack, "Habitat assessment of a restored oyster reef in south texas," *Ecological Engineering*, vol. 122, pp. 48–61, 2018.

[10] K. McFarland and M. P. Hare, "Restoring oysters to urban estuaries: Redefining habitat quality for eastern oyster performance near new york city," *PloS one*, vol. 13, no. 11, p. e0207368, 2018.

[11] S. J. Theuerkauf, D. B. Eggleston, and B. J. Puckett, "Integrating ecosystem services considerations within a gis-based habitat suitability index for oyster restoration," *PloS one*, vol. 14, no. 1, p. e0210936, 2019.

[12] W. Pine III, F. Johnson, P. Frederick, and L. Coggins, "Adaptive management in practice and the problem of application at multiple scales—insights from oyster reef restoration on florida's gulf coast," *Marine and Coastal Fisheries*, vol. 14, no. 1, p. e10192, 2022.

[13] J. A. Fitzsimons, S. Branigan, C. L. Gillies, R. D. Brumbaugh, J. Cheng, B. M. DeAngelis, L. Geselbracht, B. Hancock, A. Jeffs, T. McDonald, *et al.*, "Restoring shellfish reefs: Global guidelines for practitioners and scientists," *Conservation Science and Practice*, vol. 2, no. 6, p. e198, 2020.

[14] D. Bushek, J. Gius, and J. Morson, "2024 stock assessment workshop report," Port Norris, NJ, 2024, final Report. [Online]. Available: https://hsrl.rutgers.edu/wp-content/uploads/2024/04/SAW2024.pdf

[15] J. Haddaway-Riccio, "2020 maryland oyster stock assessment update summary," July 2020. [Online]. Available: https://dnr.maryland.gov/fisheries/Documents/SummaryUpdateReport_Oysters.pdf

[16] M. W. Beck, R. D. Brumbaugh, L. Airoldi, A. Carranza, L. D. Coen, C. Crawford, O. Defeo, G. J. Edgar, B. Hancock, M. C. Kay, *et al.*, "Oyster reefs at risk and recommendations for conservation, restoration, and management," *Bioscience*, vol. 61, no. 2, pp. 107–116, 2011.

[17] R. Botta, F. Asche, J. S. Borsum, and E. V. Camp, "A review of global oyster aquaculture production and consumption," *Marine Policy*, vol. 117, p. 103952, 2020.

[18] G. Kumar, C. Engle, and C. Tucker, "Factors driving aquaculture technology adoption," *Journal of the world aquaculture society*, vol. 49, no. 3, pp. 447–476, 2018.

[19] Y. Yang, Y. Zhou, X. Yue, G. Zhang, X. Wen, B. Ma, L. Xu, and L. Chen, "Real-time detection of crop rows in maize fields based on autonomous extraction of roi," *Expert Systems with Applications*, vol. 213, p. 118826, 2023.

[20] K. D. Stokesbury, E. P. Baker, B. P. Harris, and R. B. Rheault, "Environmental impacts related to mechanical harvest of cultured shellfish," *Shellfish aquaculture and the environment*, pp. 319–338, 2011.

[21] R. Mercaldo-Allen, P. J. Auster, P. Clark, M. S. Dixon, E. Estela, Y. Liu, L. Milke, G. Phillips, D. Redman, B. C. Smith, *et al.*, "Oyster aquaculture cages provide fish habitat similar to natural structure with minimal differences based on farm location," *Frontiers in Marine Science*, vol. 10, p. 1058709, 2023.

[22] K. B. Veggerby, M. D. Scheuerell, B. L. Sanderson, P. M. Kiffney, and B. E. Ferriss, "Shellfish aquaculture farms as foraging habitat for nearshore fishes and crabs," *Marine and Coastal Fisheries*, vol. 16, no. 2, p. e10282, 2024.

[23] R. Connolly, C. Herrera, J. Rasmussen, C. Buelow, M. Sievers, K. Jinks, C. Brown, S. Lopez-Marcano, C. Sherman, F. Martínez-Baena, *et al.*, "Estimating enhanced fish production on restored shellfish reefs using automated data collection from underwater videos," *Journal of Applied Ecology*, vol. 61, no. 4, pp. 633–646, 2024.

[24] A. D. Armbruster, R. Mercaldo-Allen, J. M. Rose, K. Seda, P. Clark, G. Phillips, D. Redman, and C. W. Conroy, "Territorial and occupancy behavior of black sea bass on oyster aquaculture gear and boulder habitat," *Frontiers in Marine Science*, vol. 11, p. 1380484, 2024.

[25] S. Manjanna, N. Kakodkar, M. Meghjani, and G. Dudek, "Efficient terrain driven coral coverage using gaussian processes for mosaic synthesis," in *2016 13th Conference on Computer and Robot Vision (CRV)*. IEEE, 2016, pp. 448–455.

[26] J. Hansen, S. Manjanna, A. Quattrini Li, I. Rekleitis, and G. Dudek, "Autonomous marine sampling enhanced by strategically deployed drifters in marine flow fields," in *OCEANS 2018 MTS/IEEE Charleston*. IEEE, 2018, pp. 1–7.

[27] N. Karapetyan, J. Moulton, and I. Rekleitis, "Meander-based river coverage by an autonomous surface vehicle," in *Field and Service Robotics*. Springer, 2021, pp. 353–364.

[28] J. Roostaei, Y. Z. Wager, W. Shi, T. Dittrich, C. Miller, and K. Gopalakrishnan, "Iot-based edge computing (iotec) for improved environmental monitoring," *Sustainable computing: informatics and systems*, vol. 38, p. 100870, 2023.

[29] M. Modasshir, A. Quattrini Li, and I. Rekleitis, "Deep neural networks: a comparison on different computing platforms," in *Canadian Conference on Computer and Robot Vision (CRV)*, 2018, pp. 383–389.

[30] A. Abdullah, T. Barua, R. Tibbetts, Z. Chen, M. J. Islam, and I. Rekleitis, "Caveseg: Deep semantic segmentation and scene parsing for autonomous underwater cave exploration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3781–3788.

[31] M. Modasshir, S. Rahman, O. Youngquist, and I. Rekleitis, "Coral Identification and Counting with an Autonomous Underwater Vehicle," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2018, pp. 524–529.

[32] M. Xanthidis, N. Karapetyan, H. Damron, S. Rahman, J. Johnson, A. O'Connell, J. M. O'Kane, and I. Rekleitis, "Navigation in the presence of obstacles for an agile autonomous underwater vehicle," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 892–899.

[33] X. Lin, N. Karapetyan, K. Joshi, T. Liu, N. Chopra, M. Yu, P. Tokekar, and Y. Aloimonos, "Uivnav: Underwater information-driven vision-based navigation via imitation learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5250–5256.

[34] B. Sadrfaridpour, Y. Aloimonos, M. Yu, Y. Tao, and D. Webster, "Detecting and counting oysters," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2156–2162.

[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[36] X. Lin, N. J. Sanket, N. Karapetyan, and Y. Aloimonos, "Oysternet: Enhanced oyster detection using simulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5170–5176.

[37] A. Gaur, C. Liu, X. Lin, N. Karapetyan, and Y. Aloimonos, "Whale detection enhancement through synthetic satellite images," in *OCEANS 2023-MTS/IEEE US Gulf Coast*. IEEE, 2023, pp. 1–7.

[38] X. Lin, C. Liu, A. Pattillo, M. Yu, and Y. Aloimonous, "Seadronesim: Simulation of aerial images for detection of objects above water," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 216–223.

[39] J. Wu, X. Lin, S. Negahdaripour, C. Fermüller, and Y. Aloimonos, "Marvis: Motion & geometry aware real and virtual image segmentation," *arXiv preprint arXiv:2403.09850*, 2024.

[40] Y. Karabatis, X. Lin, N. J. Sanket, M. G. Lagoudakis, and Y. Aloimonos, "Detecting olives with synthetic or real data? olive the above," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 4242–4249.

[41] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, "Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images," *IEEE Robotics and Automation letters*, vol. 3, no. 1, pp. 387–394, 2017.

[42] B. Joshi, M. Modasshir, T. Manderson, H. Damron, M. Xanthidis, A. Quattrini Li, I. Rekleitis, and G. Dudek, "Deepurl: Deep pose estimation framework for underwater relative localization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1777–1784.

[43] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[44] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens, "Exploring the structure of a real-time, arbitrary neural artistic stylization network," *arXiv preprint arXiv:1705.06830*, 2017.

[45] P. T. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, "Style augmentation: data augmentation via style randomization." in *CVPR workshops*, vol. 6, 2019, pp. 10–11.

[46] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[47] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control

to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.

[48] R. He, S. Sun, X. Yu, C. Xue, W. Zhang, P. Torr, S. Bai, and X. Qi, "Is synthetic data from generative models ready for image recognition?" in *arXiv preprint arXiv:2210.07574*, 2022.

[49] H. Bansal and A. Grover, "Leaving reality to imagination: Robust classification via generated datasets," 2023. [Online]. Available: https://arxiv.org/abs/2302.02503

[50] B. Trabucco, K. Doherty, M. Gurinas, and R. Salakhutdinov, "Effective data augmentation with diffusion models," *ArXiv*, vol. abs/2302.07944, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:256900870

[51] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," 2024. [Online]. Available: https://arxiv.org/abs/2405.14458

[52] M. Modasshir and I. Rekleitis, "Augmenting coral reef monitoring with an enhanced detection system," in *IEEE International Conference on Robotics and Automation*, Paris, France, 2020, pp. 1874–1880.

[53] M. Modasshir, S. Rahman, and I. Rekleitis, "Autonomous 3d semantic mapping of coral reefs," in *Results of the 12th International Conference of Field and Service Robotics*. Springer, 2021, pp. 365–379.

[54] G. Dudek, M. Jenkin, C. Prahacs, A. Hogue, J. Sattar, P. Giguere, A. German, H. Liu, S. Saunderson, A. Ripsman, *et al.*, "A visually guided swimming robot," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3604–3609.

[55] B. Joshi, M. Xanthidis, M. Roznere, N. J. Burgdorfer, P. Mordohai, A. Quattrini Li, and I. Rekleitis, "Underwater exploration and mapping," in *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*. IEEE, 2022, pp. 1–7.