A Comparative Study of Real-Time, Deep-Learning-Based Object Detection Techniques for Underwater Litter Detection

Kaya ter Burg
Delft Center for Systems and Control
Delft University of Technology
Delft, Netherlands
k.terburg@tudelft.nl

Athina Ilioudi
Cognitive Robotics
Delft University of Technology
Delft, Netherlands
a.ilioudi@tudelft.nl

Eline P. M. Troquay
Delft Center for Systems and Control
Delft University of Technology
Delft, Netherlands
eline.troquay@gmail.com

Amala Mary Vincent
Delft Center for Systems and Control
Delft University of Technology
Delft, Netherlands
a.m.vincent@tudelft.nl

Meichen Guo
Delft Center for Systems and Control
Delft University of Technology
Delft, Netherlands
m.guo@tudelft.nl

Bart De Schutter
Delft Center for Systems and Control
Delft University of Technology
Delft, Netherlands
b.deschutter@tudelft.nl

***Abstract*—Marine litter pollution is a major environmental threat due to the widespread presence of plastics and their detrimental impact on marine life and human health. There is a need for autonomous systems with computer vision to help clean the oceans. This study compares the latest state-of-the-art You Only Look Once (YOLO) models YOLOv9 - YOLOv12 in an underwater object detection setting in terms of accuracy, computational speed, and architecture complexity. We specifically focus on the smallest versions of these architectures, due to the real-time constraints of the setting. Multiple underwater datasets are combined to obtain a wide representation of underwater conditions and marine objects. The findings provide valuable insights into selecting and optimizing object detection architectures for underwater litter detection, contributing to monitoring marine ecosystems and addressing marine pollution. This work can be used as a building ground for further improving underwater object detection systems.**

***Index Terms*—computer vision, underwater object detection, deep learning, marine pollution, YOLO**

I. Introduction

Marine pollution is one of the major current global environmental problems: it poses a growing threat with the potential to damage the natural ecosystem, the economy, and human health, as documented by various studies [21], [39]. Multiple field studies [5], [11], [40], and recent models and global simulations [20] have revealed that a substantial proportion of marine litter resides beneath the water surface, sinking through the water column to the seabed.

Collecting this litter from the seabed via human diving operations are inefficient and unrealistic for long-term deep-sea clean-up [35]. Through this it becomes evident that an autonomous cleaning system is highly valuable. The described vast amount and widespread distribution of litter in water bodies further emphasize the need for systems with computer vision that can automatically localize litter. Accurate detection and classification also enable the differentiation of debris from flora and fauna, ensuring that autonomous clean-up operations do not harm the surrounding environment.

Deep-learning-based computer vision for underwater applications is gaining traction rapidly as a key technology for underwater robotics [10]. Novel procedures are emerging from the latest advancements in the field of artificial intelligence and robotics, seeking to revolutionize how underwater operations are implemented. Computer vision techniques significantly improve system performance and ensure high levels of autonomy by enabling robots to perceive their environment. The benefits of employing such technology in underwater operations are clear: greater safety for human operators, access to previously unreachable locations, and faster task completion.

However, underwater object detection is one of the most challenging research topics in the field of computer vision [45]. Detection is made less effective by factors that significantly degrade image clarity, such as absorption, scattering, and water turbidity. These difficulties are coupled with the time constraints associated with real-time operations, where detection results must be produced nearly instantaneously on specialized and often resource-limited hardware. This makes the task of real-time underwater object detection specifically challenging. The necessity for fast inference is crucial in tasks such as underwater litter detection, which requires not only precise but also real-time detection to guide immediate intervention or collection efforts. This task includes a wide range of shapes, sizes, and appearances that pose challenges to object detection algorithms, and it offers a representative benchmark for evaluating both accuracy and real-time performance in

This publication has been supported by the funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101093822 (SeaClear2.0).

underwater scenarios.

Despite recent advancements in deep learning-based vision, the literature is scattered when it comes to explicitly addressing the real-time requirements in underwater detection scenarios. Moreover, much existing work focuses on older architectures, as well as only a single dataset, see e.g. [4], [33], [46]. As such, there is no current comparison between the most recent state-of-the-art architectures across a broad selection of datasets. The present study also considers inference speed alongside accuracy and model complexity. Through this, it becomes possible to identify the most suitable architectures for realistic underwater applications.

This paper thus conducts a comprehensive comparative analysis of the state-of-the-art neural networks for real-time object detection for underwater litter detection. Specifically, our main contributions are:

- We conduct a comparative analysis of lightweight, real-time object detection architectures, focusing on recent versions of the YOLO architecture (v9-v12) [32].
- We evaluate each model using curated underwater datasets, comparing accuracy, inference speed, model size, training time, and number of Floating Point Operations (FLOPs) to identify trade-offs between performance and efficiency for real-time applications in underwater robotics.

We focus on the two smallest versions of these architectures, due to the real-time constraints of the underwater application setting. All candidate networks are trained on curated underwater datasets according to a fixed scheme.

The paper is organized as follows: Section II presents an overview of necessary preliminary knowledge. In Section III, the proposed training pipeline is presented and analyzed. Section IV discusses the results, and Section V provides concluding remarks and an outlook for future work.

II. Preliminaries

A. Underwater Object Detection

In recent years, many significant contributions to deep learning-based underwater object detection have been made. Various works [25], [28], [31], [44] employ a two-stage object detection scheme, such as Faster R-CNN [34]. In the first stage, a region-proposal network scans the image and suggests candidate bounding boxes, while in the second, a separate network classifies each candidate and refines its coordinates. Although these methods show promising results w.r.t. target detection, they face limitations in achieving real-time performance. In contrast, [2], [3], [16], [36] use a one-stage detector to perform underwater object detection. These approaches are able to achieve competitive results, yet their implementation remains challenging due to the underwater settings, as accuracy often degrades in turbid water, under low light, or when targets are partially occluded, while few results are demonstrated on resource-constrained embedded hardware. Moreover, the newest YOLO versions are still underexplored for the underwater setting.

Overall, two-stage detectors are shown to have higher accuracy compared to one-stage detectors, but they typically have slower inference times. On the other hand, one-stage detectors like YOLO often provide faster results and in some cases, can match or even outperform two-stage detectors in both speed and accuracy [1], both of which are crucial for real-time applications [45].

B. YOLO Architectures

YOLO refers to a family of one-stage object detection architectures employing a Convolutional Neural Network (CNN). These architectures predict the bounding boxes and class probabilities directly from an image in a single pass of the CNN. Due to this streamlined design, YOLO models are highly efficient and thus widely implemented in real-time applications.

The YOLO architecture that forms the base for all subsequent iterations was first presented in [32]. At its core, YOLO performs object detection by dividing the input image into an $S \times S$ cell grid. Each grid cell is responsible for predicting B bounding boxes, including their positions, dimensions, objectness score, and conditional class probability. In other words, for each bounding box, five values are predicted: the probability $\Pr(\text{object})$ of containing an object in the grid by the underlying bounding box, the center coordinates (b_x, b_y), and the dimensions (b_w, b_h) of the bounding box. In addition, for each grid cell, the class probabilities $\Pr(\text{class}_i|\text{object})$, conditioned on the grid cell containing an object, are predicted. The total output for each grid cell is therefore $B{\cdot}5{+}n$, where n represents the number of class categories. The final predictions are encoded as an $S \times S \times (B \cdot 5 + n)$ tensor.

In order to associate detections with ground truth objects, the objectness score is multiplied with the Intersection over Union (IoU) between the predicted and the ground truth bounding boxes, resulting in a confidence score c_s per each bounding box. In addition, a class-specific score c_ss is computed for each class i for each bounding box for all the grid cells. The class-specific score is calculated by multiplying the objectness score by the probability of the predicted class $\Pr(\text{class}_i|\text{object})$. The class-specific score reflects both the likelihood that the box contains an object of class i and how well the box aligns with the object.

YOLO applies a confidence threshold filter to filter out bounding boxes with low c_ss. Finally, Non-Maximum Suppression (NMS) is applied as a post-processing technique. NMS selects the box with the highest c_ss, and removes overlapping boxes based on a predefined IoU threshold.

III. Methodology

A. Datasets

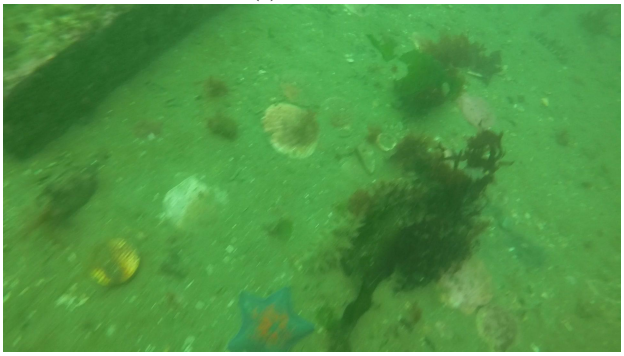
For the experiments, we curated a diverse set of marine litter and marine life datasets, each chosen for its relevance and suitability in evaluating the performance of the models under study. The selected datasets include SeaClear [9], TrashCan (Instance 1.0) [15], RUOD [12] and, Brackish [29], as summarized in Table I. An example image from each of these datasets

TABLE I: Summary of the bounding box annotated image datasets for underwater marine life and/or litter detection used for the comparison.

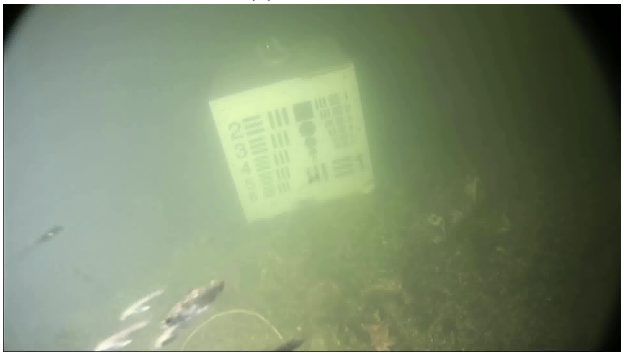
Dataset	Environment (Class examples)	#Images	#Annotations	#Classes	Year
SeaClear [9]	Mixed clear and brackish underwater (Bottle_plastic, plant, animal_urchin, rov_tortuga)	8 610	31 549	40	2022
TrashCan-Instance 1.0 [15]	Medium turbidity underwater (Bag, clothing, rope, wreckage, etc.)	7 212	12 127	22	2020
RUOD [12]	Clear underwater (Fish, sea urchins, corals, starfish, sea cucumbers, etc.)	14 000	74 904	10	2022
Brackish [29]	Brackish underwater (Fish, crabs, starfish, jellyfish, shrimp)	14 518	35 565	6	2019

(a) SeaClear

(b) TrashCan



(c) RUOD



(d) Brackish

Fig. 1: A randomly selected image from each of the chosen datasets.

is shown in Figure 1. Note that there is more variation within each dataset w.r.t. underwater conditions and objects than a single image can represent.

Several other datasets, as listed below, were excluded from our study due to various reasons:

- Datasets without bounding box annotations or any annotations at all (e.g. WildFish [47], Deep-Sea Debris database [17]).
- Datasets consisting solely of above-water trash, which do not represent the underwater environment (e.g. TACO [30], PlastOPol [6]).
- Datasets that represent smaller subsets of the TrashCan [15] dataset (e.g., Trash-ICRA19 [13], DeepPlastic [37], Deep-Sea Debris database [17]).
- Datasets with a relatively limited diversity of marine life (e.g., URPC datasets [14], UDD [24], DUO [23], UODD [18]).

In what follows, we discuss the selected datasets in more detail.

1) SeaClear: The SeaClear dataset [9] encompasses 8 610 bounding-box annotated images captured using an Remotely Operated Vehicle (ROV) at multiple underwater locations, including the coast of Lokrum and Bistrina in Croatia and Marseilles, France. This dataset features diverse classes, portraying various marine life, litter, and the ROV itself.

The images in the SeaClear dataset offer a comprehensive representation of both clear and brackish underwater environments, with clear environments primarily from the Lokrum location and brackish environments from Bistrina and Marseille. The inclusion of images from different underwater conditions, characterized by varying turbidity levels, provides a realistic and challenging test bed for evaluating various underwater object detection architectures.

In the current study, the SeaClear dataset is leveraged to validate the effectiveness of the models we trained, as the data aligns closely with the real-world conditions typically encountered in diverse underwater operations. By evaluating the performance of the models on the SeaClear dataset, we gain valuable insights into their capability to detect and to classify marine objects and litter in various underwater environments.

2) TrashCan: The TrashCan dataset [15] consists of 7 212 bounding box annotated images, encompassing diverse observations of trash, an ROV, and a wide variety of undersea flora and fauna. Two versions of the dataset exist: TrashCan-Material and TrashCan-Instance. In this study, we exclusively utilize the TrashCan-Instance version, which focuses on object-level annotations, as our research aims to identify and categorize different underwater object types rather than distinguishing between material types.

The imagery in the TrashCan dataset originates from the J-EDI (JAMSTEC E-Library of Deep-sea Images) dataset [17], curated by the Japan Agency of Marine Earth Science and Technology (JAMSTEC). These images are predominantly sourced from videos captured by ROVs operated by JAMSTEC since 1982, primarily in the sea of Japan.

Our selection of the TrashCan dataset is driven by multiple factors. Primarily, it is one of the largest underwater marine litter dataset to date. Additionally, the dataset's curation and proven usage in other scientific studies reinforces its credibility [7], [27]. Compared with the SeaClear dataset, the TrashCan images are collected in deeper waters, which causes the underwater conditions to differ significantly between the two datasets. This enables a thorough evaluation of both the quality and the challenges of both datasets in underwater litter detection.

3) RUOD: The RUOD dataset [12] comprises 14 000 bounding-box annotated, high-resolution images, showcasing a rich variance of marine objects and clear underwater environments. It was created to address the limitations of existing underwater object detection datasets, which often suffer from a scarcity of images, limited categories, low resolution, and insufficient representation of environmental challenges.

RUOD is an underwater, natural-light dataset carefully curated for underwater detection. The dataset is a collection of images from various websites, supplemented by images from the URPC2020 dataset [14]. The images thus come from a variety of locations and contain a wide range of viewing conditions. The dataset focuses on ten carefully selected marine flora and fauna objects with the highest overall frequency of occurrence in day-to-day life, presenting a diverse range of types, appearances, and scales.

Due to its distinctive features, particularly the abundant variety of marine objects and clear underwater environments, RUOD has been selected to enhance the robustness of the models not only in general underwater scenes but also specifically in clear underwater environments.

4) Brackish: The Brackish dataset [29] encompasses 14 518 images, out of which 12 444 contain objects of interest with bounding box annotations. These images feature a diverse collection of marine animals captured in brackish or unclear water conditions, characterized by varying but mostly high turbidity.

The dataset is captured using stationary cameras positioned at the bottom of Limfjorden in Denmark, approximately nine meters below the water surface. This setup ensures the portrayal of authentic underwater conditions, reflecting the real-world challenges of underwater litter detection in brackish environments.

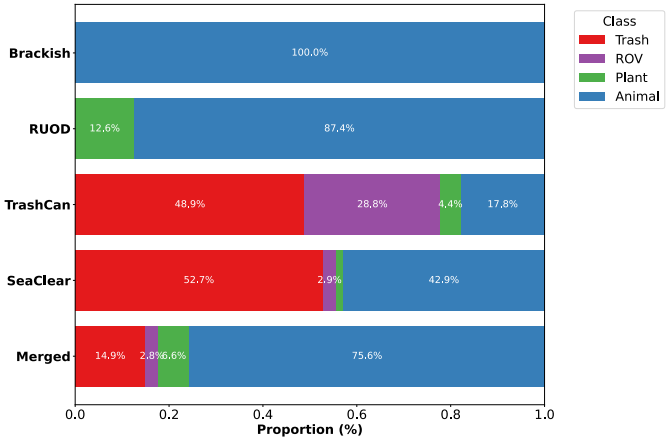


Fig. 2: The class distribution per dataset after mapping the class labels.

Similar to the RUOD dataset, the Brackish dataset includes the most common marine animals found in day-to-day life in the given area, represented across six distinct classes.

We have deliberately selected the Brackish dataset for this study to not only bolster the overall robustness of our model through its abundant training data, but also to specifically address the complexities of underwater object detection in brackish/unclear water with varying levels of turbidity.

B. Data Processing

To ensure consistency and comparability across all datasets, a standardized data pre-processing pipeline was applied. The original class labels in each dataset were mapped to a unified set of four classes: ROV, plant, animal, and trash. The datasets can then straightforwardly be merged together. The class distribution for each of the datasets (including the merged dataset) after the remapping is visualized in Figure 2.

The training and validation split for all datasets followed an 80-20 ratio. For the Brackish dataset, which was split 80-10-10 (train-validation-test), the validation and test datasets were merged to create the validation set. Lastly, a key consideration to prevent time-dependent data leakage, especially in datasets containing videos, involves the careful partitioning of entire videos into either the training or validation set. By treating complete videos as cohesive entities during the split, we mitigate the risk of overlapping frames between the two sets and safeguard against inadvertent contamination of the validation set with information from the training set. By preserving the independence and integrity of each set throughout the training and validation processes, we enhance the robustness and reliability of our model evaluation.

C. Object Detection Models

The YOLO architectures encompass multiple versions, each introducing improvements over its predecessors. We include YOLO version 9 up to and including version 12, the most

TABLE II: Performance of YOLO models on the COCO benchmark dataset as reported in the literature.

Model	$\mathbf{mAP}^{0.5-0.95}_{\text{val}}(\%)$
YOLOv9-T [43]	38.3
YOLOv9-S [43]	46.8
YOLOv10-N [42]	38.5
YOLOv10-S [42]	46.3
YOLOv11-N [19]	39.5
YOLOv11-S [19]	47.0
YOLOv12-N [38]	40.6
YOLOv12-S [38]	48.0

recently released one at the time of writing. Notable improving features for each of these versions are:

- YOLOv9 [43]: introduces two new techniques to deal with data loss and computational efficiency issues.
- YOLOv10 [42]: incorporates NMS-free training and various ways of reducing latency, FLOPs, and parameter counts without sacrificing accuracy.
- YOLOv11 [19]: uses a more efficient architecture with special blocks and attention [41] mechanisms.
- YOLOv12 [38]: proposes an attention-centric framework whilst keeping the speed of CNN-based approaches.

For each of these versions, particular emphasis was placed on the two smallest versions of the architectures, i.e. nano (called "tiny" for YOLOv9) and small. These smallest architectures have the highest inference speed as well as the lowest storage requirements due to their reduced number of parameters. Both inference speed as well as storage space are important limitations to consider when integrating an object detection model into an ROV for real-time detection [45].

Table II presents the performance of these selected models on the object detection benchmark dataset COCO [22] as reported in the literature. The COCO dataset is a large-scale object detection dataset for above water scenarios. It is often used for benchmarking object detection architectures. The reported mean Average Precision (mAP) scores provide insight into the respective performance characteristics of the models. The $\text{mAP}^{0.5-0.95}$ is the standard benchmark metric for the COCO dataset, where mAP is averaged over multiple IoU thresholds of 0.5 - 0.95 instead of a single threshold.

Each subsequent YOLO version improves upon previous architectures, demonstrated by the increased benchmark performance. This shows the relevancy of each new YOLO version for the general above-water object detection setting. In the current study, we evaluate whether this same principle holds for the underwater setting.

D. Training Scheme

All evaluated networks are trained according to a fixed scheme, to enable a fair comparison. The YOLO models were trained for 100 epochs, using a batch size of 32, an image size of 640 by 640 pixels, and the AdamW optimizer [26] with an initial learning rate of 0.01. Default hyperparameter configurations were applied, ensuring consistency for comparison across the trained models. These training parameters were chosen to strike a balance between computational efficiency and model convergence, enabling the YOLO models to effectively learn and generalize to the underwater litter detection task. All models were initialized with weights from pre-training on the COCO dataset [22].

Validation of each model was performed with a batch size of 16, a confidence threshold of 0.001 and an NMS IoU threshold of 0.7. For each model, we report the mAP score with an IoU threshold of 0.5. mAP is a widely used metric to evaluate and benchmark object detection models. It captures the trade-off between precision and recall. Precision measures the proportion of relevant objects predicted within the set of all predictions and recall measures the proportion of relevant objects that were found among all predictions. The IoU threshold determines when a detection is a true positive, where IoU refers to the degree of overlap between two bounding boxes, in this case the predicted and ground truth bounding boxes.

The training of the models with the selected datasets was conducted on the computational resources of the DelftBlue supercomputer GPU partition, provided by the Delft High Performance Computing Centre [8]. DelftBlue is equipped with NVIDIA V100S GPUs, where one GPU was requested per job, allowing high-performance computing for our training tasks.

Validation and inference processes were performed locally on a Lenovo Thinkpad P16 Gen 2, which features an NVIDIA RTX 2000 Ada Generation Laptop GPU with 8MB memory and a 13th-generation Inter i9 processor with 64GB of RAM.

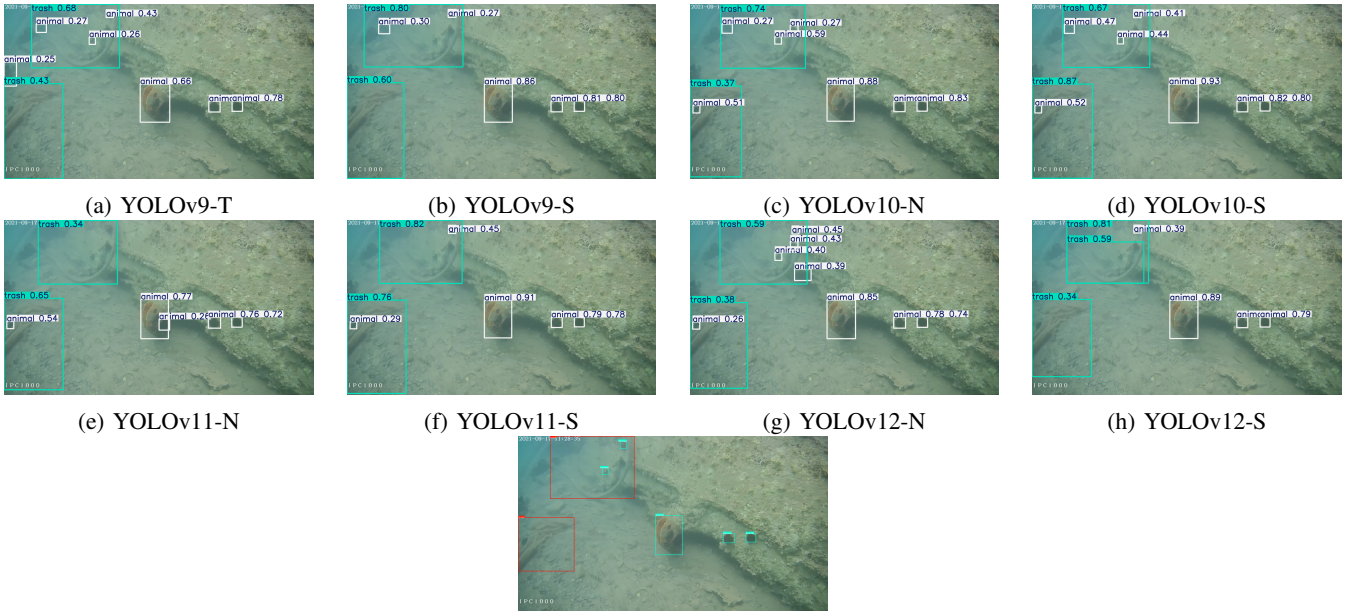
Our code and trained models can be found on: https://github.com/kayatb/real-time_underwater_litter_detection.

IV. Experimental Results

Our main results are summarized in Table III. Overall, the best performing model is YOLOv10-S with an mAP score of 85.8. Of the smaller architectures, YOLOv12-N performs the best with an mAP score of 84.7.

The performance from the small models compared to the nano models are better across all versions. This is straightforward, since the small models have more parameters than the nano models. However, the difference in performance decreases from 3% for YOLOv9 to 0.8% for YOLOv12. This indicates that the newer YOLO models are more powerful with a reduced number of parameters compared to older versions. In Figure 3, the detections from all evaluated models are shown for a representative validation set image fromt the SeaClear dataset. The image contains two litter items and five animals as per the ground truth annotations. All models are able to predict the object locations and classes with reasonable accuracy. The nano models overall have more false positives for the small animals, except for YOLOv11-N, which has some false negatives. The small models have good performance overall, with less false positive and negatives for the small animals (sea urchins in this image). Notably, YOLOv12-S has a false positive litter detection overlapping the true positive.

The performance per dataset is shown in Table III. For the SeaClear dataset, YOLOv10-S and YOLOv12-S are the best



(a) YOLOv9-T (b) YOLOv9-S (c) YOLOv10-N (d) YOLOv10-S

(e) YOLOv11-N (f) YOLOv11-S (g) YOLOv12-N (h) YOLOv12-S

(i) Ground Truth

Fig. 3: Detections from all models and the ground truth annotations on a representative validation image from the SeaClear dataset.

TABLE III: $mAP^{0.5}$ scores per dataset across all classes.

Model	Merged	SeaClear	RUOD	Brackish	TrashCan
YOLOv9-T	82.3	86.8	76.1	96.7	66.6
YOLOv9-S	84.9	92.3	**78.8**	98.4	70.3
YOLOv10-N	83.1	90.2	75.3	97.3	70.7
YOLOv10-S	**85.8**	**93.7**	78.2	98.7	74.4
YOLOv11-N	83.6	89.9	75.4	97.1	72.7
YOLOv11-S	85.3	93.5	78.5	**98.8**	73.9
YOLOv12-N	84.7	90.2	76.7	97.5	**75.1**
YOLOv12-S	85.4	**93.7**	78.5	**98.8**	73.7

TABLE IV: $mAP^{0.5}$ scores per class for the merged dataset.

Model	All	ROV	Plant	Animal	Trash
YOLOv9-T	82.3	90.5	68.4	87.0	83.2
YOLOv9-S	84.9	90.7	**70.5**	90.5	87.8
YOLOv10-N	83.1	92.5	67.4	87.7	84.9
YOLOv10-S	**85.8**	93.2	69.9	90.9	89.1
YOLOv11-N	83.6	94.0	67.4	87.9	85.1
YOLOv11-S	85.3	91.6	70.2	90.9	88.6
YOLOv12-N	84.7	**94.2**	69.5	88.4	86.6
YOLOv12-S	85.4	91.0	70.1	**91.1**	**89.3**

performing architectures. For the RUOD dataset, YOLOv9-S performs the best. For the Brackish dataset, YOLOv11-S and YOLOv12-S perform the best. For TrashCan, YOLOv12-N outperforms all models, even the bigger architectures.

In Table IV, the performance per class for the merged dataset are presented. For both the animal and the trash class, YOLOv12-S performs the best, closely followed by YOLOv10-S. Remarkably, YOLOv12-N outperforms all other models for the ROV class. YOLOv9-S performs the best on the plant class, which is arguably the most difficult class to detect in these datasets due to the limited number of annotations and the variable appearance.

Table V shows the runtime metrics for each YOLO architecture. Naturally, the nano architectures have considerably less parameters, a lower number of FLOPs, and less latency during inference than their small counterparts. They also require less time to complete training and use less memory. Generally, the more recent architectures have more parameters, but require a lower number of FLOPs, due to a more efficient design. YOLOv10 has the highest inference speed and YOLOv11 requires the least training time. Regarding storage, the newer models require more space, which is directly related to their corresponding larger number of parameters.

In Figure 4, the trade-offs between latency/FLOPs and mAP scores are shown for each model. From this visualization, it becomes clear that YOLOv10 is well-performing in terms of the latency-mAP trade-off. The YOLOv10-N architecture has the lowest overall latency and even the small architecture has a lower latency than all of the other architectures. On top of that, YOLOv10-N achieves the highest overall mAP score across datasets and classes. As for the FLOPs-mAP trade-off, YOLOv11-N has the lowest number of FLOPs, but is outperformed by YOLOv12-N, which has a negligible increase of 0.1 G in the number of FLOPs compared to YOLOv11-N. For the small architectures, YOLOv12-S has the lowest number of FLOPs and is outperformed only by YOLOv10-S.

V. Conclusions and Future Work

In this study, we have performed an extensive comparison between various state-of-the-art real-time object detection YOLO-based architectures. We focused specifically on the challenging setting of underwater litter detection. To do this we

TABLE V: An overview of architectural and runtime metrics for all tested models.

Model	#Params (M)	FLOPs (G)	Inference (ms)	Train (h)	Storage (MB)
YOLOv9-T	2.0	7.9	1.35	7.7	4.6
YOLOv9-S	7.3	27.4	1.65	10.5	15.2
YOLOv10-N	2.7	8.4	0.7	6.2	5.7
YOLOv10-S	8.1	24.8	1.1	8.3	16.5
YOLOv11-N	2.6	6.4	1.1	5.1	5.5
YOLOv11-S	9.4	21.6	1.4	7.0	19.2
YOLOv12-N	2.6	6.5	1.2	6.7	5.5
YOLOv12-S	9.3	21.5	1.8	9.9	18.9

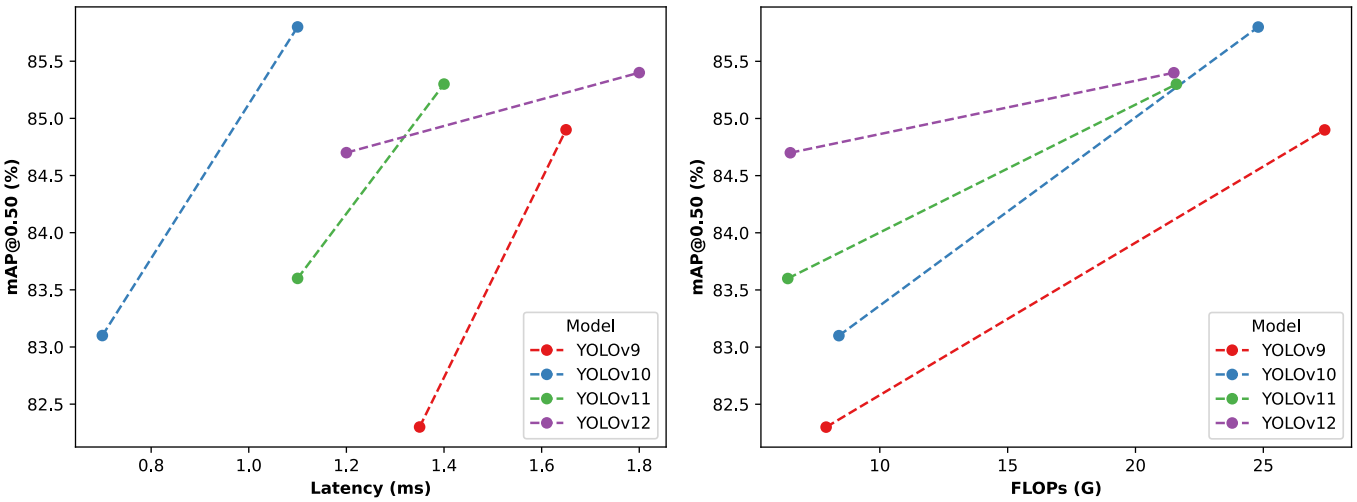


Fig. 4: The mAP-latency (left) and mAP-FLOPs (right) trade-offs for all evaluated models. The leftmost points of each line refer to the nano models and the rightmost points to the small models.

selected four underwater datasets, representing a wide variety of underwater environments and marine objects.

Overall, all evaluated models achieve good detection results across all datasets. Depending on what performance metric is the most important, a different model may be the better choice. YOLOv11 has a notably lower training time than the other versions. YOLOv9 has the lowest number of parameters and thus requires the least storage space. In terms of FLOPs, YOLOv12 is a good choice, having just a minor increase in the number of FLOPs compared to YOLOv11, but achieving more competitive detection results. YOLOv10-S achieves a good balanced performance in terms of latency as well as detection accuracy, having the lowest latency and the highest overall performance across datasets and classes.

This work demonstrates that the more recent YOLO versions are highly usable for underwater detection applications, especially those with real-time constraints. Our results can be used as a starting points for new developments within the field of underwater object detection. Building on top of this first comparison of recent architectures, there are several direction for future work:

- Broad hyperparameter tuning. For this work, performing an extensive search for the optimal hyperparameters was out of the scope. However, specifically tuning the hyperparameters for the underwater environment or even a particular dataset can yield an extra performance boost.
- Training and testing on additional datasets. In this work, we focused on underwater litter detection specifically, but other relevant real-time underwater applications exist, such as ecological monitoring.
- Extending the comparative analysis to include non-YOLO models. The present analysis focused exclusively on YOLO-based architectures. However, a comparison with other types of object detection architectures can yield additional insights.
- Enhancing the base architectures further. Although the base architectures already give promising results, there is still room for improvement. The underwater environment has many challenges that need to be addressed more explicitly, such as the detection of small, overlapping, and clustered objects. The suboptimal vision circumstances due to e.g. turbidity also require extra attention. Special techniques for dealing with these issues need to be developed and integrated to further the field of underwater object detection.

References

[1] K. Ali, M. Moetesum, I. Siddiqi, and N. Mahmood. Marine object detection using transformers. In *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 951–957, Islamabad, Pakistan, Aug. 2022.

[2] S. Cai, G. Li, and Y. Shan. Underwater object detection using collaborative weakly supervision. *Computers and Electrical Engineering*, 102:108159, 2022.

[3] L. Chen, Z. Liu, L. Tong, Z. Jiang, S. Wang, J. Dong, and H. Zhou. Underwater object detection using invert multi-class adaboost with deep learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.

[4] B. C. Corrigan, Z. Y. Tay, and D. Konovessis. Real-time instance segmentation for detection of underwater litter as a plastic source. *Journal of Marine Science and Engineering*, 11(8), 2023.

[5] A. Cozar, M. Sanz-Martín, E. Martí, J. I. González-Gordillo, B. Ubeda, J. A. Galvez, X. Irigoien, and C. M. Duarte. Plastic accumulation in the Mediterranean Sea. *PLOS ONE*, 10(4):e0121762, Apr. 2015.

[6] M. Córdova, A. Pinto, C. C. Hellevik, S. A.-A. Alaliyat, I. A. Hameed, H. Pedrini, and R. d. S. Torres. Litter Detection with Deep Learning: A Comparative Study. *Sensors*, 22(2):548, Jan. 2022.

[7] L. Dai, H. Liu, P. Song, and M. Liu. A gated cross-domain collaborative network for underwater object detection. *Pattern Recognition*, 149:110222, 2024.

[8] Delft High Performance Computing Centre (DHPC). DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2, 2024.

[9] A. Đuraš, B. J. Wolf, A. Ilioudi, I. Palunko, and B. De Schutter. A dataset for detection and segmentation of underwater marine debris in shallow waters. *Scientific data*, 11(1):921, 2024.

[10] M. Elmezain, L. Saad Saoud, A. Sultan, M. Heshmat, L. Seneviratne, and I. Hussain. Advancing underwater vision: A survey of deep learning models for underwater object recognition and tracking. *IEEE Access*, 13:17830–17867, 2025.

[11] M. Eriksen, L. C. M. Lebreton, H. S. Carson, M. Thiel, C. J. Moore, J. C. Borerro, F. Galgani, P. G. Ryan, and J. Reisser. Plastic pollution in the world's oceans: More than 5 trillion plastic pieces weighing over 250,000 tons afloat at sea. *PLoS ONE*, 9(12):e111913, Dec. 2014.

[12] C. Fu, R. Liu, X. Fan, P. Chen, H. Fu, W. Yuan, M. Zhu, and Z. Luo. Rethinking general underwater object detection: Datasets, challenges, and solutions. *Neurocomputing*, 517:243–256, Jan. 2023.

[13] M. Fulton, J. Hong, M. J. Islam, and J. Sattar. Robotic detection of marine litter using deep visual detection models. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5752–5758, Montreal, QC, Canada, May 2019.

[14] F. Han, J. Yao, H. Zhu, and C. Wang. Marine organism detection and classification from underwater vision based on the deep CNN method. *Mathematical Problems in Engineering*, 2020:1–11, Feb. 2020.

[15] J. Hong, M. Fulton, and J. Sattar. Trashcan: A semantically-segmented dataset towards visual detection of marine debris, 2020.

[16] K. Hu, F. Lu, M. Lu, Z. Deng, and Y. Liu. A marine object detection algorithm based on ssd and feature enhancement. *Complexity*, 2020(1):5476142, 2020.

[17] JAMSTEC. Deep-sea debris database. Japan Agency for Marine Earth Science and Technology, 2018.

[18] L. Jiang, Y. Wang, Q. Jia, S. Xu, Y. Liu, X. Fan, H. Li, R. Liu, X. Xue, and R. Wang. Underwater species detection using channel sharpening attention. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4259–4267, Virtual Event China, Oct. 2021.

[19] G. Jocher and J. Qiu. Ultralytics YOLO11, 2024.

[20] D. Klink, A. Peytavin, and L. Lebreton. Size dependent transport of floating plastics modeled in the global ocean. *Frontiers in Marine Science*, 9:903134, July 2022.

[21] L. Lebreton, B. Slat, F. Ferrari, B. Sainte-Rose, J. Aitken, R. Marthouse, S. Hajbane, S. Cunsolo, A. Schwarz, A. Levivier, K. Noble, P. Debeljak, H. Maral, R. Schoeneich-Argent, R. Brambini, and J. Reisser. Evidence that the Great Pacific Garbage Patch is rapidly accumulating plastic. *Scientific Reports*, 8(1):4666, Mar. 2018.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[23] C. Liu, H. Li, S. Wang, M. Zhu, D. Wang, X. Fan, and Z. Wang. A dataset and benchmark of underwater object detection for robot picking. In *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6, Shenzhen, China, July 2021.

[24] C. Liu, Z. Wang, S. Wang, T. Tang, Y. Tao, C. Yang, H. Li, X. Liu, and X. Fan. A new dataset, Poisson GAN and AquaNet for underwater object grabbing. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2831–2844, May 2022.

[25] J. Liu, S. Liu, S. Xu, and C. Zhou. Two-stage underwater object detection network using Swin Transformer. *IEEE Access*, 10:117235–117247, 2022.

[26] I. Loshchilov and F. Hutter. Fixing weight decay regularization in Adam. *CoRR*, abs/1711.05101, 2017.

[27] S. Majchrowska, A. Mikołajczyk, M. Ferlin, Z. Klawikowska, M. A. Plantykow, A. Kwasigroch, and K. Majek. Deep learning-based waste detection in natural and urban environments. *Waste Management*, 138:274–284, 2022.

[28] R. Mandal, R. M. Connolly, T. A. Schlacher, and B. Stantic. Assessing fish abundance from underwater video using deep neural networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2018.

[29] M. Pedersen, J. Haurum, R. Gade, T. Moeslund, and N. Madsen. Detection of marine animals in a new underwater dataset with varying visibility, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2019.

[30] P. F. Proença and P. Simões. TACO: Trash annotations in context for litter detection, Mar. 2020. arXiv:2003.06975.

[31] S. Qi, J. Du, M. Wu, H. Yi, L. Tang, T. Qian, and X. Wang. Underwater small target detection based on deformable convolutional pyramid. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2784–2788, 2022.

[32] J. Redmon, S. Kumar Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[33] F. Rehman, M. Rehman, M. Anjum, and A. Hussain. Optimized yolov8: An efficient underwater litter detection using deep learning. *Ain Shams Engineering Journal*, 16(1):103227, 2025.

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 91–99, Cambridge, MA, USA, 2015.

[35] R. Rodineliussen. Caring for water: Underwater waste, trash diving, and publicity in Stockholm. *Swedish Journal of Anthropology*, 4(2):73–92, 2021.

[36] M. Sung, S.-C. Yu, and Y. Girdhar. Vision based real-time fish detection using convolutional neural network. In *OCEANS 2017 - Aberdeen*, pages 1–6, 2017.

[37] G. Tata. DeepPlastic: An Open Source Image Dataset for Epipelagic Marine Plastic Detection, Apr. 2021.

[38] Y. Tian, Q. Ye, and D. Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.

[39] T. van Emmerik and A. Schwarz. Plastic debris in rivers. *WIREs Water*, 7(1):1398, 2020.

[40] E. van Sebille, C. Wilcox, L. Lebreton, N. Maximenko, B. D. Hardesty, J. A. van Franeker, M. Eriksen, D. Siegel, F. Galgani, and K. L. Law. A global inventory of small floating plastic debris. *Environmental Research Letters*, 10(12):124006, Dec. 2015.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[42] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, et al. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, 37:107984–108011, 2024.

[43] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao. YOLOv9: Learning what you want to learn using programmable gradient information. In *European Conference on Computer Vision*, pages 1–21. Springer, 2024.

[44] F. Xu, H. Wang, J. Peng, and X. Fu. Scale-aware feature pyramid architecture for marine object detection. *Neural Computing and Applications*, 33(8):3637–3653, apr 2021.

[45] S. Xu, M. Zhang, W. Song, H. Mei, Q. He, and A. Liotta. A systematic review and analysis of deep learning-based underwater object detection. *Neurocomputing*, 527:204–232, Mar. 2023.

[46] X. Zhang, D. Zhu, and W. Gan. YOLOv7t-CEBC network for underwater litter detection. *Journal of Marine Science and Engineering*, 12(4), 2024.

[47] P. Zhuang, Y. Wang, and Y. Qiao. WildFish: A large benchmark for fish recognition in the wild. In *Proceedings of the 26th ACM International Conference on Multimedia*, pages 1301–1309, Seoul Republic of Korea, Oct. 2018.