Beyond Model-Centric Marine Debris Object Detection: Towards A Data-Centric Approach

G029 (s1707068, s2737499)

Abstract

Ocean pollution caused by marine debris has become a growing environmental concern. Object detection systems using deep learning methods present a promising approach for automated marine debris identification. However, insufficient labelled data is a major bottleneck to train robust deep neural networks. This research seeks to address this scarcity of annotated datasets in marine debris detection by leveraging related datasets to enhance object detection performance. To this end, we fine-tuned the YOLOv11s model on the *Underwater Plastic Pollution Detection* dataset containing 15 categories of underwater debris. We then explored semi-supervised learning, sequential fine-tuning, oversampling, and data augmentation techniques to improve upon this baseline. Our best model achieved validation mAP50 of 0.834 and test mAP50 of 0.809, outperforming the latest results in the literature. The contributions of this work include establishing a standardised benchmark for marine debris detection and advancing the performance of autonomous underwater vehicles through data-driven methods. Since automated marine debris detection systems require real-time processing, we benchmarked the inference speeds alongside accuracy metrics.

1. Introduction

Plastic, once proclaimed as "the material of 1,000 uses" (Pilapitiya & Rathnayake, 2024), has become one of our ocean's most pervasive pollutants. Every year an estimated 12 million tonnes of plastic enter our oceans (Eunomia Research and Consulting, 2024) comprising 80% of all studied marine debris (International Union for Conservation of Nature, 2022). Traditional approaches to monitoring this pollution have relied on labour intensive manual sampling and visual identification methods (Jia et al., 2023). However, recent advances have sparked significant interest in autonomous underwater vehicles (AUVs) capable of both identifying and collecting marine debris (Sánchez-Ferrer et al., 2023) or conducting automated debris surveys (Merlino et al., 2020).

Object detection of marine debris represents a fundamental component of these systems. While deep neural networks have demonstrated remarkable success in object detection across various domains (Szegedy et al., 2013), their application to marine debris detection faces a significant challenge: the scarcity of labelled training data. Previous work has explored synthetic data generation to enhance existing datasets (Musić et al., 2020; Sánchez-Ferrer et al., 2023), though its benefits diminish as real data volumes increase. Alternative approaches, to be explored in this study, include the use of semi-supervised learning to leverage unlabelled data (Noroozi & Favaro, 2016; Misra & van der Maaten, 2020) and sequential fine-tuning to leverage auxiliary labelled data, were identified as a gap in marine debris detection by a most recent comprehensive view of the field (2016-2023) (Jia et al., 2023). The review highlights that while various CNN-based architectures have been evaluated, the lack of standardised benchmarks and consistent experimental settings makes direct comparisons of these methodologies challenging. We sought to address this lack of unity across the field by establishing a baseline to measure all future experiments against.

Given that most prior research on this task has primarily focused on model-centric approaches such as novel algorithms and model architectures, our work instead investigated data-centric approaches to address the labelled data scarcity problem in the context of marine debris object detection. More specifically, we employed semi-supervised learning, sequential fine-tuning, oversampling, and data augmentation techniques to leverage both auxiliary labelled and unlabelled data. While marine debris exists across all orders of magnitude, from the Great Pacific Garbage Patch spanning millions of square kilometres (Parker, 2015), to nanometre sized microplastics and microfibres (United States Environmental Protection Agency, 2024), this study focuses specifically on the detection and segmentation of macro-scale waste items that retain their original form.

Our contributions in this work include the following:

1. Established an improved baseline for the *Underwater Plastic Pollution Detection* dataset using YOLOv11s.
2. Evaluated the recently released YOLOv12s model.
3. Investigated semi-supervised learning to leverage unlabelled data and sequential fine-tuning to leverage related labelled data of underwater garbage.
4. Explored oversampling and data augmentation techniques for underwater debris detection.
5. Benchmarked the inference speed of YOLOv8s, YOLOv11s, and YOLOv12s.

Figure 1. Examples of *plastic bottle* images from PlastOPol (left) and Underwater Plastic (right) datasets

2. Datasets and Task

2.1. Main Dataset

The *Underwater Plastic Pollution Detection* dataset consists of ocean debris images labelled with 15 classes including: mask, can, cellphone, electronics, glass bottle, glove, metal, miscellaneous, net, plastic bag, plastic bottle, plastic, rod, sunglasses and tire. There are 5127 images in total with a 70:20:10 train/validation/test split. Some preprocessing was already applied to the dataset, including resizing all images to 416x416 and using common data augmentation methods such as random flipping and mosaic augmentation. In our experiments, we resized the images to 640x640 for optimal performance on YOLO models and implemented additional data augmentation techniques.

2.2. Auxiliary Datasets

PlastOPol is a non-marine rubbish dataset that we leveraged for semi-supervised learning to enhance our main dataset (Córdova et al., 2022). This dataset comprises 2418 images containing 5300 labelled instances of litter across various terrestrial environments. All annotations in this one-class labelled dataset belong to the "litter" super-category. Our intent was to incorporate PlastOPol to strengthen our model's capacity to recognise waste across different contexts, hypothesising that litter recognition capabilities developed on land could transfer to underwater environments. Figure 1 shows sample images from PlastOPol and the main dataset.

TrashCan was utilised for our investigation into semi-supervised learning and sequential fine-tuning to enhance our main dataset. This supplementary dataset consists of underwater images captured by ROVs primarily in Japanese waters. The dataset contains 7,218 images of various categories including ROVs, unknown objects, marine plants, animals, and debris (with some debris categories overlapping with our main dataset). However, TrashCan differs from our main dataset in hue & saturation, aspect ratio (480x270), and types of objects, as the frames were captured by a different camera in a different geographical region. The dataset maintains an approximate 84:16 train/validation split. We did not require a separate test set as our model evaluation was conducted on the test split of the main dataset. Figure 2 shows sample images from TrashCan and the main dataset.

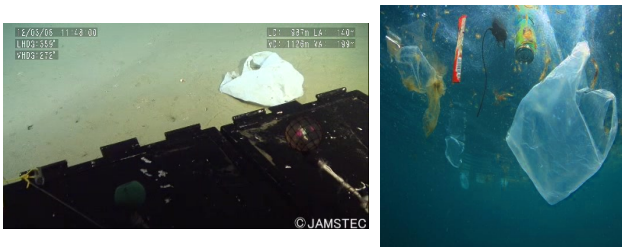


Figure 2. Examples of a *bag* image from TrashCan (left) and a *plastic bag* image from Underwater Plastic (right) datasets

2.3. Task and Evaluation Method

From the review of recent works, it is apparent that the literature is not unified across the domain of marine debris classification and that there is a call to reduce the need for large amounts of labelled data. Once a baseline was established, this study addressed the following question: Can semi-supervised learning and sequential fine-tuning effectively leverage auxiliary datasets of underwater debris images to improve model performance? Building on this, we investigated if data augmentation techniques could improve object detection performance and benchmarked inference speeds for real-time deployment on AUVs.

To assess the models, we employed mean Average Precision (mAP) as the main evaluation criteria, a widely used metric for object detection. mAP provides a balanced evaluation of precision and recall across all categories in the dataset. mAP uses the Intersection over Union (IoU) score which measures the overlap between the predicted bounding box and the ground truth. First, the Average Precision (AP) for each class is computed as the area under the Precision-Recall curve based on a certain IoU threshold. Then, mAP is calculated as the average AP score across all classes. Two common variants of mAP are mAP50 and mAP50-95. mAP50 uses a fixed IoU threshold of 50%, while mAP50-95 is a stricter metric which takes the average scores from multiple IoU thresholds ranging from 50% to 95% with a step size of 5% (Terven et al., 2023).

3. Methodology

YOLO, which stands for "You Only Look Once", was a novel approach for the object detection task proposed by (Redmon, 2016). Previous methods utilised a two-step framework involving both regression and classification, which first predicted potential bounding boxes around an object and then classified the object in those boxes. In contrast, YOLO was successful in performing object detection as a single regression problem by a single neural network. This drastically reduced the model complexity and increased training and inference speeds while maintaining competitive accuracy. Moreover, a key advantage of YOLO was its capability to generalise object representations, making it easily adaptable to unseen objects and domains in downstream tasks.

Different versions of YOLO have been released over the

years since the model was first proposed in 2016, ranging from versions 1 to 12. Each model iteration brought potential enhancements over its predecessor, including improved training strategies, data augmentation techniques, and extension to additional computer vision tasks (Terven et al., 2023). The common objective among these refinements is not only accuracy but also high inference speed, as real-time object detection is an important application of YOLO. Most YOLO models are available in five sizes: nano, small, medium, large, and extra-large, making them flexible to various computational settings. In this report, we primarily employed "YOLOv11s", the *small* variant of version 11 which consists of significantly fewer parameters than the previous versions, resulting in improved processing speeds without exhibiting a degradation in accuracy (Khanam & Hussain, 2024). In addition, we evaluated YOLOv12, the latest version which uses an optimized attention architecture to achieve competitive results on the COCO dataset (Tian et al., 2025).

Transfer Learning is a commonly utilised technique in computer vision, where a model first is pre-trained on large amounts of labelled or unlabelled data, then fine-tuned on a smaller, task-specific dataset for related task. YOLO was pre-trained on the Microsoft Common Objects in COntext (COCO) corpus (Lin et al., 2014), a diverse collection of images of people, vehicles, animals, household items. The model can then be effectively fine-tuned in various real-world applications such as autonomous vehicles, wildlife monitoring, and pill detection (Terven et al., 2023). In this case, we primarily used the pre-trained YOLOv11s model and fine-tuned it for the task of detecting marine debris.

Sequential Fine-tuning (SFT) is a method commonly employed in NLP to train domain-specific language models such as *BloombergGPT* which specialises in finance (Wu et al., 2023) and *Med-PaLM* which specialises in clinical knowledge (Singhal et al., 2023). We extended this idea to marine garbage detection by first training on the TrashCan dataset to adapt the model to the specific trash domain before fine-tuning on our main underwater debris dataset, as depicted in Figure 3.

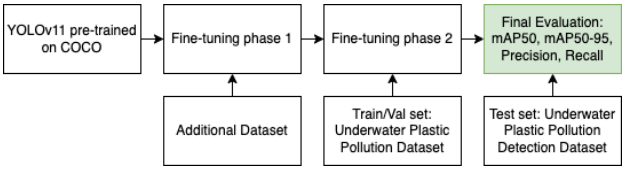


Figure 3. Sequential fine-tuning (SFT) pipeline

Semi-supervised Learning (SSL) is a method which typically employs large amounts of unlabelled data in cases where annotated data is difficult to obtain. As a first step, an initial model is trained using the small amount of available labelled data. This model is then used to generate pseudo labels for the unlabelled data. Pseudo labels with high confidence are combined with the original labelled dataset and used to retrain the model (Xu et al., 2021). Figure 4 shows training and evaluation pipeline of this process. Since there is a limited amount of annotated images of ocean debris, semi-supervised learning is a promising method to leverage unlabelled images.

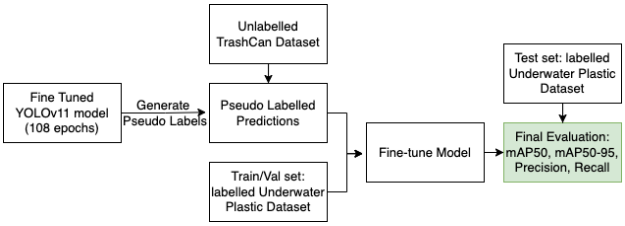


Figure 4. Semi-supervised learning (SSL) pipeline

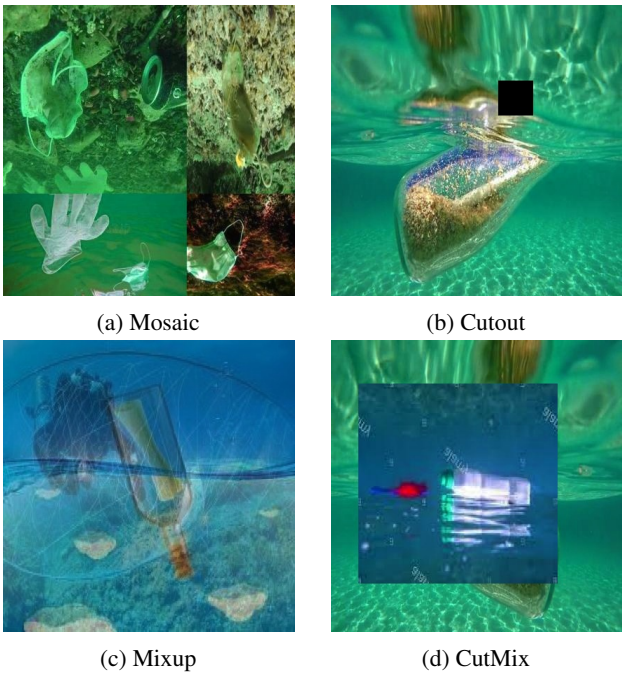
Oversampling is a simple method to address imbalanced datasets by increasing the representation of minority classes. It involves randomly duplicating existing minority class samples (Mohammed et al., 2020). This is a straightforward approach, however it creates multiple identical images and does not leverage synthetic variations.

Data Augmentation Techniques are essential to improve the robustness and generalisation of computer vision models. For our task with a small training set, data augmentation is beneficial to create additional training examples. Mosaic augmentation has already been applied to the original training set, while random flipping, scaling, translating, and colour jittering are applied by default by the *ultralytics*[1] package used for training. *Mosaic* combines 4 images in a 2x2 grid, which helps the model to detect small objects in diverse contexts (Bochkovskiy et al., 2020). In addition, we implemented three data augmentation methods which have been shown to be effective for object detection, namely Cutout, Mixup, and CutMix. *Cutout* applies square or rectangular masks at random locations in the image, forcing the model to learn the full context of the image instead of relying on specific features (DeVries & Taylor, 2017). *Mixup* creates new training examples using convex combinations of two randomly selected images, essentially blending the images and merging their labels (Zhang et al., 2017). *CutMix* works by cutting patches of an image, pasting the patches onto another image, and mixing their labels, resulting in enhanced model accuracy and robustness (Yun et al., 2019). These data augmentation methods are illustrated in Figure 5.

4. Related Work

It is challenging to identify unified benchmarks in the literature for marine debris detection, as various datasets were used for training and evaluation, including JAMSTEC Debris (Fulton et al., 2019; Bajaj et al., 2021; Xue et al., 2021; Huang et al., 2023), Trash-ICRA19 (Hipolito et al., 2021), TrashCan (Zhou et al., 2022), Debris Tracker (Teng et al., 2022), and web-based images (Musić et al., 2020; Bhanumathi et al., 2022). Even when the same dataset was used, the authors employed different annotation schemes, trash categories, and train/test split. However, there is clear pro-

[1]https://docs.ultralytics.com/



(a) Mosaic (b) Cutout

(c) Mixup (d) CutMix

Figure 5. Illustrations of different data augmentation methods

gression in model architecture over time in the studies on underwater garbage detection. Earlier papers employed variants of ResNet, Mask R-CNN, and Faster R-CNN, then from 2020 there was a shift towards YOLO models, ranging from YOLOv3 to YOLOv8. YOLO was reported to outperform ResNet and R-CNN both in terms of accuracy and inference speed (Musić et al., 2020; Xue et al., 2021; Huang et al., 2023), making it highly suitable for real-time detection tasks. Following this trajectory, we decided to explore YOLOv11 and YOLOv12, the latest versions of this model architecture.

The most recent and closely related paper to our task is (Rehman et al., 2025). The authors evaluated variants of YOLOv8, from nano to extra-large, on the *Underwater Plastic Pollution Detection* dataset which is our main dataset of interest. YOLOv8s (*small*) with extensive hyperparameter tuning led to the best validation mAP50 score of 0.827. We replicated this result as one of the baselines for our experiments.

The majority of previous studies on underwater debris detection focused on proposing and evaluating different algorithms (Fulton et al., 2019), loss functions (Huang et al., 2023), or model architectures (Musić et al., 2020; Bajaj et al., 2021; Xue et al., 2021; Zhou et al., 2022). However, the effectiveness of data augmentation techniques as well as leveraging related datasets via semi-supervised learning and sequential fine-tuning remains largely understudied for marine debris detection. Semi-supervised learning was reported to improve performance for object detection on the COCO dataset (Sohn et al., 2020), but this has yet to be investigated in the context of underwater imagery. Some data augmentation techniques were proposed for the detection of marine organisms (Huang et al., 2019) and marine vessels (Shin et al., 2020). However, these augmentation methods were not evaluated on detecting marine debris and employed older model architectures such as ResNet and Faster R-CNN. Recognising this research gap, we aimed to evaluate these data-centric approaches for the specific task of detecting ocean debris, for which there is a limited amount of annotated data available.

5. Experiments

To ensure fairness and reproducibility, we used NVIDIA GeForce RTX 4060 Ti and *ultralytics* v8.3.75 across all experiments. We observed that using either a different GPU or a different version of *ultralytics* can significantly impact results.

5.1. Hyperparameter Tuning

As an initial experiment, we tested the pre-trained YOLOv11s model on the underwater pollution dataset of interest without any further training. However, it either failed to detect the ocean debris in the images or misclassified them as other objects, which demonstrated the need for fine-tuning the model to our specific task. This result was not surprising, as COCO was not trained on any underwater objects (Lin et al., 2014).

Next, we compared the performance of YOLOv11s and YOLOv12s, the newest versions in the YOLO family. For both models, we varied the batch sizes of [16, 32, 64] and trained for 20 epochs using the *auto* learning rate (LR) automatically determined by the optimizer *AdamW*. As reported in Table 1, YOLOv11s outperformed YOLOv12s across all batch sizes. Therefore, we conducted hyperparameter search for YOLOv11s to determine the best combination of LR and batch size.

Model	**Batch=16**	**Batch=32**	**Batch=64**
YOLOv1ls	0.671	**0.748**	0.738
YOLOv12s	0.629	0.647	0.663

Table 1. Validation mAP50 scores of YOLOv11s and YOLOv12s with varying batch sizes at 20 epochs

We tested all combinations between learning rates of [1e-3, 1e-4, 1e-5, *auto*] and batch sizes of [16, 32, 64] for YOLOv11s. The results showed that a batch size of 32 was optimal across all learning rates (Table 2) and the best two learning rates were 1e-4 and *auto*.

	Batch=16	**Batch=32**	**Batch=64**
LR = 1e-3	0.531	0.711	0.614
LR = 1e-4	0.729	**0.799**	0.769
LR = 1e-5	0.460	0.583	0.509
LR = *auto*	0.671	**0.748**	0.738

Table 2. Validation mAP50 scores from YOLOv11s Hyperparameter Search at 20 epochs

5.2. Establishing Baselines

To obtain the latest baseline from the literature, we reproduced the results from (Rehman et al., 2025) in which YOLOv8s was trained for 108 epochs on our main dataset, *Underwater Plastic Pollution Detection*. In addition, we trained YOLOv11s with the best configurations from our hyperparameter search as well as YOLOv12s[2] for 108 epochs to compare against YOLOv8s. The results in Table 3 clearly show that YOLOv11s with LR=*auto* performs best overall and has smallest generalisation gap. Here, our YOLOv11s model was able to outperform the latest results in the literature on this particular dataset. We established this as our new baseline with validation mAP50 of 0.834 and test mAP50 of 0.809 and our future experiments aimed to improve upon this result.

Model	LR	mAP50		mAP50-95	
		Val	Test	Val	Test
YOLOv8s	1e-4	0.826	0.783	0.528	0.474
YOLOv11s	auto	0.834	**0.809**	0.550	**0.499**
YOLOv11s	1e-4	0.837	0.764	0.548	0.496
YOLOv12s	auto	0.827	0.750	0.538	0.493

Table 3. Results of different YOLO *small* versions at 108 epochs and batch size of 32

As we needed to quickly test various methods, it was inefficient to run every experiment for 108 epochs. Thus, we trained YOLOv11s with the best hyperparameters from above (LR=*auto*, batch=32) for 30 epochs as a "proxy" baseline result as shown in Table 4. For the different methods tested in our experimentation, we trained the models for 30 epochs to compare against this "proxy" baseline as summarised later in Table 7, and only fine-tuned the best performing models for the full 108 epochs at the very end.

Split	Precision	Recall	mAP50	mAP50-95
Validation	0.796	0.711	0.776	0.506
Test	0.785	0.629	0.747	0.484

Table 4. Baseline results of YOLOv11s trained for 30 epochs with best hyperparameters

5.3. Dataset Scaling

Using the baseline parameters, experiments were run to compare the performance with increasing training set size, shown in Figure 6. As the training set size increased, mAP50 scores also increased, indicating potential for the model performance to improve with larger volumes of data. With more manually annotated data unavailable and expensive to collect, this justified the need for applying data augmentation and leveraging auxiliary datasets through semi-supervised learning and sequential fine-tuning.

[2]Batch size of 64 performed best for YOLOv12s but we used batch size of 32 due to GPU constraints.

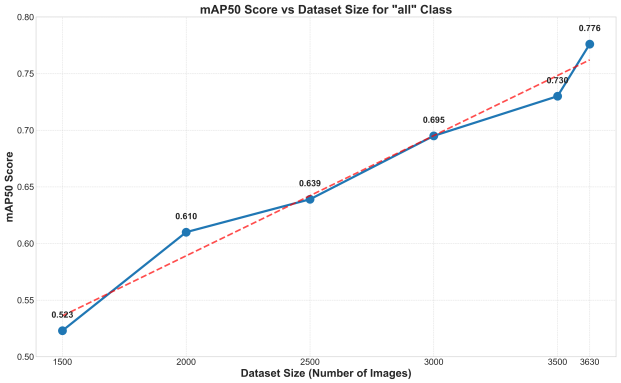


Figure 6. mAP50 score increases as dataset size increases from 1500 to full dataset size (3630 images) at 500 image intervals

5.4. Semi-Supervised Learning

As a preprocessing step for semi-supervised learning, we applied the class labels provided in the TrashCan dataset to keep only trash instances and remove images of animals, plants, and ROVs. This resulted in 2928 images from various "trash" categories which could be utilised for semi-supervised learning. To generate pseudo labels, our best YOLOv11s model from §5.2 was used to predict the classes and bounding boxes for the images in TrashCan.

We experimented with confidence thresholds of 90%, 85%, and 80% during the prediction step and observed a trade-off between prediction quality and quantity. The 90% threshold resulted in very few but highly-confident pseudo labels, while the 80% threshold generated more predictions but of lower quality. We found a good balance with the 85% threshold, resulting in to 171 high-quality predicted images which were used to enhance the original training set. We then fine-tuned YOLOv11 with the extended data for 30 epochs and obtained a validation mAP50 score of 0.772, which is on par with the baseline mAP50 score of 0.776.

We also attempted to perform semi-supervised learning using the PlastOPol dataset. However, our experiments revealed significant domain gap issues, as the model fine-tuned on images of underwater garbage was ineffective in detecting trash instances on land. The contextual differences between land-based and marine litter proved too substantial for effective knowledge transfer, hindering our model's ability to generalise across these distinct environmental domains.

5.5. Sequential Fine-tuning

To address class imbalance shown in Figure 7, we removed the dominant ROV and unknown categories in TrashCan (75% of training, 79% of validation data) from our training process. This prevented model bias toward classes absent from out main dataset. We also removed the "trash_" prefix from debris labels in the TrashCan dataset for better alignment with our main dataset. YOLOv11s was fine-tuned on this modified TrashCan dataset for 50 epochs and then further fine-tuned on our main dataset for 30 epochs (results in

Table 5). Sequential fine-tuning slightly decreased overall mAP50, with varied performance across object categories as some classes improved (*cellphone*, *tire*, *misc*) while others declined (*electronics*, *mask*). The *sunglasses* class maintained high performance, demonstrating robust detection for images of glasses present in the original COCO dataset.

We then refined class definitions, renaming *pipe* to *rod* and relabelling plastic items more specifically (*bag* to *pbag*, *bottle* to *pbottle*). We also consolidated *tarp*, *snack_wrapper*, *container*, and *cup* into a broader *plastic* class. These class modifications (SFT-CLM) slightly improved overall mAP50 to 0.736, though still below the baseline. The *rod* class showed the most significant improvement (0.272 to 0.511), exceeding its baseline performance. While *pbag* and *pbottle* maintained strong performance, the consolidated *plastic* class decreased from 0.584 to 0.490, suggesting excessive class variability despite our intentions to mirror the diversity already present in the main dataset's *plastic* category.

Class	Base	SFT	SFT-CLM
All	0.776	0.732	0.736
mask	0.891	0.794	0.808
can	0.808	0.617	0.693
cellphone	0.982	0.986	0.979
electronics	0.805	0.743	0.702
gbottle	0.815	0.749	0.745
glove	0.854	0.841	0.865
metal	0.355	0.274	0.364
misc	0.577	0.623	0.595
net	0.940	0.936	0.911
pbag	0.977	0.961	0.973
pbottle	0.817	0.792	0.793
plastic	0.629	0.584	0.490
rod	0.401	0.272	0.511
sunglasses	0.995	0.995	0.806
tire	0.797	0.811	0.799

Table 5. Performance comparison (mAP50) across model versions. Base: Baseline; SFT: Sequential Fine Tuning; SFT-CLM: SFT with Class Label Modifications.

5.6. Oversampling

For multi-class classification with imbalanced datasets, one of the simplest solution is oversampling the minority classes. We used a frequency based approach that balances the dataset by duplicating images in the four most under-represented classes. From Figure 7 we can see that *can*, *metal*, *rod* and *sunglasses* are the least represented with only 20, 22, 9 and 3 instances respectively. The *can*, *metal*, and *rod* classes were multiplied by two while the *sunglasses* class was multiplied by 5, resulting in a total of 192 additional images to the training set. The oversampling technique enhanced validation performance, increasing the mAP50 score from 0.776 (baseline) to 0.791. An improvement was also seen in two of the target classes: *can* and *rod* with a validation mAP50 increase from 0.808 to 0.889 and 0.401 to 0.744 respectively. Oversampling *metal* harmed its validation performance with its mAP50 validation score decreasing from 0.355 to 0.25. Interestingly, the mAP50 score of *sunglasses* remained unchanged at 0.995, despite being the most oversampled class. This may be attributed to the fact that *glasses* is a class present in the COCO dataset (Lin et al., 2014), on which YOLO models are pre-trained.

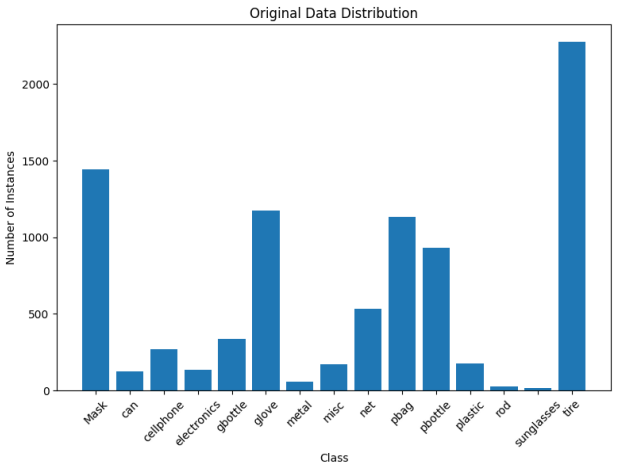


Figure 7. Data distribution of the *Underwater Plastic Pollution Detection* dataset training set

5.7. Data Augmentation Techniques

Data augmentation methods allow us to extend our original dataset without using an auxiliary dataset. As described in §3, we experimented with three augmentation techniques suitable for object detection: Cutout, Mixup, and CutMix. To implement Cutout, we followed the original paper and applied a random square mask to 50% of the images in the training set. In Mixup, the hyperparameter α controls the strength of blending of the images. We tested α=[1.0, 5.0] and applied Mixup to 20% and 50% of the dataset. For CutMix, we used α=1.0 as in the original paper to obtain a random patch size which is cut from an image and pasted onto another image. We experimented with applying CutMix to 50% and 100% of the training set. For each augmentation method, we trained the model for 30 epochs and compared against the baseline of 30 epochs.

The results on the validation of the different augmentation methods are reported in Table 6. Cutout and Mixup did not improve model performance, while **CutMix(p=0.5)**, CutMix with 50% probability, resulted in increased mAP50, precision, and recall. This is consistent with the literature as CutMix generally outperforms Mixup and Cutout in various computer vision tasks (Yun et al., 2019). We observed that it is important to maintain a high proportion of the original images relative to the augmented samples. An excessive number of augmented images had a negative impact on the model, as Mixup performed better when applied to 20% of the training set than 50% and CutMix performed better when applied to 50% of the training set than 100%.

Our main dataset is highly imbalanced, as previously discussed. Certain classes, such as *tire* and *mask*, have thousands of instances while other classes have fewer than 100

Method	Precision	Recall	mAP50	mAP50-95
Baseline	0.796	0.711	0.776	0.506
Cutout(p=0.5)	0.792	0.662	0.741	0.485
Mixup(p=0.2, α=1.0)	0.766	0.716	0.758	0.495
Mixup(p=0.5, α=1.0)	0.817	0.668	0.733	0.489
Mixup(p=0.2, α=5.0)	0.746	0.697	0.755	0.492
CutMix(p=0.5)	0.818	0.720	**0.793**	0.507
CutMix(p=1.0)	0.741	0.681	0.751	0.486

Table 6. Validation set results of data augmentation methods

instances. To address the data imbalance, we applied **Balanced CutMix** by augmenting only the under-represented classes such that each class has at least 250 instances, resulting in 1440 additional images across the 7 classes with the fewest training instances: *sunglasses*, *rod*, *metal*, *can*, *electronics*, *misc*, and *plastic*. The YOLOv11s model fine-tuned on this extended training set with augmentation of targeted classes achieved a validation mAP50 score of 0.824, which is a notable gain over the baseline (0.776). Closer inspection of the validation set revealed significant improvements in the mAP50 scores of the augmented classes such as *rod*, *plastic*, and *misc*, contributing to an overall improvement.

5.8. Full Runs With Best Models

The best "proxy" results at 30 epochs for each tested method are reported in Table 7. Out of these, we trained the Baseline, CutMix(p=0.5) and Balanced CutMix for the full 108 epochs as described in §5.2. In addition, we trained Semi-Supervised for 108 epochs as it was the best method that leveraged an auxiliary dataset.

Model	Precision	Recall	mAP50	mAP50-95
Baseline	0.796	0.711	0.776	0.506
Semi-supervised	0.804	0.714	0.772	0.496
Sequential FT	0.784	0.650	0.736	0.472
Oversampling	0.781	0.753	0.791	0.516
Cutmix(p=0.5)	0.818	0.720	0.793	0.507
Balanced Cutmix	0.823	0.776	0.824	0.527

Table 7. Best validation set results for each method at 30 epochs

The validation and test results of the models trained for 108 epochs are reported in Table 8, and the baseline model's enhanced predictions on test images are shown in Figure 8. There was a drop from validation to test performance across all models, likely because the test set was smaller and contained more difficult instances.

Model	mAP50		mAP50-95	
	Val	Test	Val	Test
Baseline	0.834	**0.809**	0.550	0.499
Semi-supervised	0.816	0.796	0.539	0.505
CutMix(prob=0.5)	0.801	0.781	0.530	0.487
Balanced Cutmix	0.850	0.802	0.556	**0.506**

Table 8. Validation and test results for best models at 108 epochs

Overall, the Baseline and Balanced Cutmix achieved competitive metrics, outperforming the YOLOv8 results in (Rehman et al., 2025). The Baseline without any augmentation performed best in terms of test mAP50, while Balanced Cutmix achieved the best mAP50-95. This suggested that while the Baseline was more effective in coarse object detection, Balanced Cutmix excelled in precise boundary localisation. CutMix(p=0.5) underperformed relative to the Baseline and Balanced CutMix, indicating that in an imbalanced setting, data augmentation is more effective when applied exclusively to under-represented classes rather than across all classes. Semi-supervised did not improve results over the Baseline, demonstrating the model's sensitivity to the specific data distribution.

Fine-grained test set results at 108 epochs for each class are shown in Table 9 for the best two models, Baseline and Balanced CutMix. Both the Baseline and Balanced Cutmix excelled in *cellphone* and *sunglasses*, possibly due to cellphones having distinguishing physical features and glasses being present in the COCO dataset, as discussed in §5.6. Categories such as *metal* and *plastic* remained challenging for both models, as they contained a wide variety of objects. Contrary to our expectations, there was not a clear improvement for the under-represented classes in Balanced CutMix over the Baseline. In terms of mAP50, Balanced CutMix led to the largest improvement of 4.6% for *plastic* and the greatest decrease of 13.0% for *metal*. Looking at the stricter metric mAP50-95, Balanced CutMix improved results for most classes, notably a 8.0% increase for *electronics* and 6.7% increase for *mask*. These results demonstrated that targeted augmentation techniques can improve overall detection quality and localisation precision but with class-specific trade-offs that should inform model selection based on application requirements.

Class	mAP50		mAP50-95	
	Baseline	Balanced	Baseline	Balanced
All	**0.809**	0.802	0.499	**0.506**
mask	0.893	0.894	0.642	0.709
can	0.805	0.784	0.394	0.371
cellphone	0.995	0.995	0.861	0.864
electronics	0.892	0.895	0.588	0.668
gbottle	0.713	0.723	0.454	0.489
glove	0.827	0.823	0.671	0.704
metal	0.502	0.372	0.172	0.089
misc	0.824	0.799	0.399	0.394
net	0.942	0.946	0.676	0.677
pbag	0.964	0.968	0.846	0.862
pbottle	0.859	0.877	0.556	0.572
plastic	0.645	0.691	0.274	0.246
rod	0.512	0.520	0.201	0.208
sunglasses	0.995	0.995	0.398	0.398
tire	0.764	0.741	0.355	0.338

Table 9. Test mAP scores per class for Baseline and Balanced CutMix at 108 epochs. Under-represented classes are *italicised*.

5.9. Inference Speed Tests

Given the use case of automated marine debris identification and collection, AUVs require both high accuracy as well as fast processing time. Hence, as a supplementary task, we benchmarked and compared the inference speed

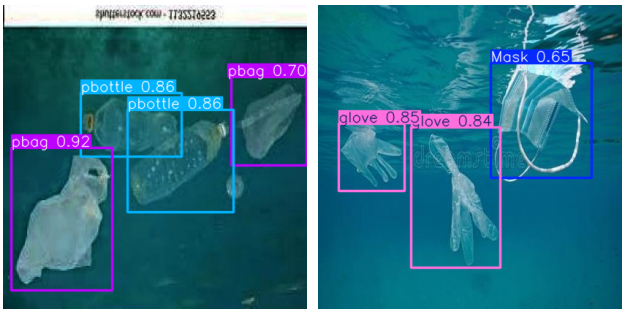


Figure 8. Predicitions on test images by YOLOv11s fine-tuned for 108 epochs

of the fine-tuned YOLOv8s, YOLOv11s, and YOLOv12s models on the test set containing 501 images of underwater garbage. We conducted the speed tests using the NVIDIA GeForce RTX 4060 Ti GPU and recorded the frames per second (FPS) and mAP50 for each model with batch sizes of [1, 4, 8, 16]. The processing times could vary across different GPUs, but the objective was to compare the relative speeds of the different models. Here we are only evaluating the *small* variants of these YOLO models, and not other variants such as *nano*, *medium*, or *large*.

The results of the inference speed tests are recorded in Table 10. Overall, YOLOv8s and YOLOv11s attained comparable inference speeds, while YOLOv12s was noticeably slower. Across all three models tested, we observed that using a larger batch size of 8 or 16 significantly improved inference speed without degradation in accuracy as measured by mAP50. As a result, we recommend using YOLOv11s with a high batch size in real-time objection detection applications for optimal speed and accuracy.

Batch	**YOLOv8s**		**YOLOv11s**		**YOLOv12s**	
	FPS	mAP50	FPS	mAP50	FPS	mAP50
1	57.60	0.785	72.86	0.807	59.03	0.751
4	131.98	0.785	125.57	0.807	98.11	0.751
8	128.78	0.784	125.08	0.807	100.75	0.751
16	127.89	0.784	127.15	0.807	100.54	0.751

Table 10. Inference speeds (FPS) of different YOLO versions

6. Conclusions

Underwater debris detection remains an largely understudied and challenging task as it requires predicting the class label and bounding box simultaneously. For a trash instance to be accurately detected, both the predicted class and bounding box need to be correct. The difficulty is aggravated by the lack of annotated images of ocean debris. Moreover, both accuracy and inference speed need to be taken into account for real-world applications such as AUVs that can automatically detect and collect marine debris.

Our work investigated the impact of semi-supervised learning, sequential fine-tuning, oversampling, and data augmentation for the underwater debris detection task, which has been largely underexplored. Our experiment results indicated that ocean debris detection is very sensitive to the data distribution.

Although semi-supervised learning was reported by (Sohn et al., 2020) to improve object detection performance, the authors had access to 118k labelled images and 123k unlabelled images from the COCO dataset for semi-supervised learning. Similarly, sequential fine-tuning often requires a large amount of data to train a domain-specific model. In the context of underwater imagery, there is limited data which makes these methods less effective. Furthermore, the auxiliary datasets in our experiments come from a different distribution from our main dataset, which made knowledge transfer more difficult. We suggest that semi-supervised learning and sequential fine-tuning can be re-evaluated if more underwater images of a similar distribution are available.

Data augmentation techniques, particularly CutMix, proved to be effective in improving precise boundary localisation and overall performance. Our hypothesis is that since CutMix generates new training samples by cutting and pasting patches from existing images, the augmented examples align closely to the original dataset. Consequently, the model could effectively learn patterns of these new images when combined with the original training set. Furthermore, our experiments indicated that for a highly imbalanced dataset, targeted augmentation of under-represented classes is particularly beneficial. This approach allowed the model to learn a more robust representation of each class, resulting in improved localisation precision and overall performance.

7. Future Work

Future research could explore other methods to address the scarcity of annotated underwater trash images. *GridMask* is a data augmentation method that randomly removes square patches in an image in a grid-like pattern (Chen et al., 2020). GridMask could be a promising method to apply to our dataset as it has been shown to perform well in multiple computer vision tasks. In addition, image generation tools could be used to generate synthetic images to extend the dataset while ensuring fair and ethical data usage. To mitigate the data imbalance issue, we could try undersampling the majority class in addition to oversampling the minority classes (Mohammed et al., 2020).

Most of our experiments have focused on the fine-tuning stage, but future work could explore using self-supervised learning as the pre-training step for the YOLO backbone. Self-supervised learning methods, such as contrastive learning (Xie et al., 2021) and masked prediction (Li et al., 2023), do not require annotated data and instead rely on unlabelled data for supervision. Finally, given sufficient computational resources, we could evaluate larger variants of YOLOv11, specifically *medium*, *large*, *extra-large*, which are expected to achieve higher precision and mAP50 compared to the *small* variants employed in this work (Khanam & Hussain, 2024).

References

Bajaj, Rahul, Garg, Suyash, Kulkarni, Nilima, and Raut, Rachana. Sea debris detection using deep learning: diving deep into the sea. In *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 1–6. IEEE, 2021.

Bhanumathi, M, Gugan, R, et al. Marine plastic detection using deep learning. In *Advances in Parallel Computing Algorithms, Tools and Paradigms*, pp. 406–413. IOS Press, 2022.

Bochkovskiy, Alexey, Wang, Chien-Yao, and Liao, Hong-Yuan Mark. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

Chen, Pengguang, Liu, Shu, Zhao, Hengshuang, Wang, Xingquan, and Jia, Jiaya. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020.

Córdova, Manuel, Pinto, Allan, Hellevik, Christina Carrozzo, Alialyát, Saleh Abdel Afou, Hameed, Ibrahim A., Pedrini, Helio, and Silva Torres, Ricardo da. Plastopol: A dataset for litter detection, 2022. URL https://doi.org/10.5281/zenodo.5829155.

DeVries, Terrance and Taylor, Graham W. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Eunomia Research and Consulting. Plastics in the marine environment, 2024. URL https://www.eunomia.co.uk/reports/plastics-in-the-marine-environment/.

Fulton, Michael, Hong, Jungseok, Islam, Md Jahidul, and Sattar, Junaed. Robotic detection of marine litter using deep visual detection models. In *2019 international conference on robotics and automation (ICRA)*, pp. 5752–5758. IEEE, 2019.

Hipolito, Japhet C, Alon, Alvin Sarraga, Amorado, Ryndel V, Fernando, Maricel Grace Z, and De Chavez, Poul Isaac C. Detection of underwater marine plastic debris using an augmented low sample size dataset for machine vision system: a deep transfer learning approach. In *2021 IEEE 19th Student Conference on Research and Development (SCOReD)*, pp. 82–86. IEEE, 2021.

Huang, Baoxiang, Chen, Ge, Zhang, Hongfeng, Hou, Guojia, and Radenkovic, Milena. Instant deep sea debris detection for maneuverable underwater machines to build sustainable ocean using deep neural network. *Science of the Total Environment*, 878:162826, 2023.

Huang, Hai, Zhou, Hao, Yang, Xu, Zhang, Lu, Qi, Lu, and Zang, Ai-Yun. Faster r-cnn for marine organisms detection and recognition using data augmentation. *Neurocomputing*, 337:372–384, 2019.

International Union for Conservation of Nature. The plastic pollution crisis, July 2022. URL https://www.iucn.org/story/202207/plastic-pollution-crisis.

Jia, Tianlong, Kapelan, Zoran, de Vries, Rinze, Vriend, Paul, Peereboom, Eric Copius, Okkerman, Inke, and Taormina, Riccardo. Deep learning for detecting macroplastic litter in water bodies: A review. *Water Research*, 231:119632, 2023. doi: 10.1016/j.watres.2023.119632.

Khanam, Rahima and Hussain, Muhammad. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

Li, Feng, Zhang, Hao, Xu, Huaizhe, Liu, Shilong, Zhang, Lei, Ni, Lionel M., and Shum, Heung-Yeung. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3041–3050, June 2023.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Merlino, Silvia, Paterni, Marco, Berton, Alessandro, and Massetti, Luciano. Unmanned aerial vehicles for debris survey in coastal areas: Long-term monitoring programme to study spatial and temporal accumulation of the dynamics of beached marine litter. *Remote Sensing*, 12(8):1260, 2020. doi: 10.3390/rs12081260.

Misra, Ishan and van der Maaten, Laurens. Self-supervised learning of pretext-invariant representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6706–6716, 2020. doi: 10.1109/CVPR42600.2020.00674.

Mohammed, Roweida, Rawashdeh, Jumanah, and Abdullah, Malak. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, pp. 243–248, 2020. doi: 10.1109/ICICS49469.2020.239556.

Musić, Josip, Kružić, Stanko, Stančić, Ivo, and Alexandrou, Floris. Detecting underwater sea litter using deep neural networks: an initial study. In *2020 5th International Conference on Smart and Sustainable Technologies (SpliTech)*, pp. 1–6. IEEE, 2020.

Noroozi, Mehdi and Favaro, Paolo. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Computer Vision – ECCV 2016*, volume 9910 of *Lecture Notes in Computer Science*, Cham, 2016. Springer. ISBN 978-3-319-46465-7. doi: 10.1007/978-3-319-46466-4_5.

Parker, Laura. Ocean trash: 5.25 trillion pieces and counting, but big questions remain, January 2015. URL https://education.nationalgeographic.org/resource/

ocean-trash-525-trillion-pieces-and-counting-big-questions-remain/.

Pilapitiya, P.G.C. Nayanathara Thathsarani and Rathnayake, Amila Sandaruwan. The world of plastic waste: A review. *Cleaner Materials*, 11:100220, March 2024. doi: 10.1016/j.clema.2024.100220.

Redmon, J. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Rehman, Faiza, Rehman, Mariam, Anjum, Maria, and Hussain, Afzaal. Optimized yolov8: An efficient underwater litter detection using deep learning. *Ain Shams Engineering Journal*, 16(1):103227, 2025.

Sánchez-Ferrer, Alejandro, Valero-Mas, Jose J., Gallego, Antonio Javier, and Calvo-Zaragoza, Jorge. An experimental study on marine debris location and recognition using object detection. *Pattern Recognition Letters*, 168:154–161, 2023. ISSN 0167-8655. doi: 10.1016/j.patrec.2022.12.019.

Shin, Hyeon-Cheol, Lee, Kwang-Il, and Lee, Chang-Eun. Data augmentation method of object detection for deep learning in maritime image. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 463–466. IEEE, 2020.

Singhal, Karan, Azizi, Shekoofeh, Tu, Tao, Mahdavi, S Sara, Wei, Jason, Chung, Hyung Won, Scales, Nathan, Tanwani, Ajay, Cole-Lewis, Heather, Pfohl, Stephen, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

Sohn, Kihyuk, Zhang, Zizhao, Li, Chun-Liang, Zhang, Han, Lee, Chen-Yu, and Pfister, Tomas. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.

Szegedy, Christian, Toshev, Alexander, and Erhan, Dumitru. Deep neural networks for object detection. *Advances in Neural Information Processing Systems*, 2013. URL http://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.

Teng, Cathy, Kylili, Kyriaki, and Hadjistassou, Constantinos. Deploying deep learning to estimate the abundance of marine debris from video footage. *Marine Pollution Bulletin*, 183:114049, 2022.

Terven, Juan, Córdova-Esparza, Diana-Margarita, and Romero-González, Julio-Alejandro. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, 2023.

Tian, Yunjie, Ye, Qixiang, and Doermann, David. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.

United States Environmental Protection Agency. Microplastics research, July 2024. URL https://www.epa.gov/water-research/microplastics-research.

Wu, Shijie, Irsoy, Ozan, Lu, Steven, Dabravolski, Vadim, Dredze, Mark, Gehrmann, Sebastian, Kambadur, Prabhanjan, Rosenberg, David, and Mann, Gideon. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

Xie, Enze, Ding, Jian, Wang, Wenhai, Zhan, Xiaohang, Xu, Hang, Sun, Peize, Li, Zhenguo, and Luo, Ping. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8392–8401, October 2021.

Xu, Mengde, Zhang, Zheng, Hu, Han, Wang, Jianfeng, Wang, Lijuan, Wei, Fangyun, Bai, Xiang, and Liu, Zicheng. End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3060–3069, 2021.

Xue, Bing, Huang, Baoxiang, Wei, Weibo, Chen, Ge, Li, Haitao, Zhao, Nan, and Zhang, Hongfeng. An efficient deep-sea debris detection method using deep neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:12348–12360, 2021.

Yun, Sangdoo, Han, Dongyoon, Oh, Seong Joon, Chun, Sanghyuk, Choe, Junsuk, and Yoo, Youngjoon. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Zhang, Hongyi, Cisse, Moustapha, Dauphin, Yann N, and Lopez-Paz, David. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhou, Wei, Zheng, Fujian, Yin, Gang, Pang, Yiran, and Yi, Jun. Yolotrashcan: a deep learning marine debris detection network. *IEEE Transactions on Instrumentation and Measurement*, 72:1–12, 2022.