Are All Marine Species Created Equal? Performance Disparities in Underwater Object Detection

Supplementary Material

Overview

Here we provide additional implementation details for our systematic analysis (Section 1); extended results, mainly on the RUOD dataset [2]; and perform ablations for the architectures used in our systematic analysis. Section 2 supplements the results from the main paper, while Section 3 provides information on the performance of alternative architectures to support the generality of our results.

1. Additional Implementation Details

1.1. Localization

The model was loaded and trained with the default parameters, *i.e*. 30 epochs, with images resized to 640x640 pixels and data augmentations during training, including hue, saturation, brightness, translation, image scaling, horizontal flip and mosaic augmentation. The batch size was set to 8. All localization experiments were run on an NVIDIA H100 GPU.

1.2. Classification

For the classification experiments, all images are resized to 224x224 pixels and normalized using ImageNet means and standard deviation. We fine-tune this model for 30 epochs with a batch size of 32. The training images are augmented with random horizontal flip and random rotation within ± 15 degrees. We use standard cross-entropy loss and Adam as optimizer with an initial learning rate of 0.001 that we reduce by the factor 0.5 every time no validation improvement has been recorded for 3 epochs, stopping at the minimum learning rate of 0.00003. All classification experiments were conducted on an Apple MacBook Pro M4 with MPS.

2. Additional Results Analysis

We highlight in our paper that the main focus lies on the DUO dataset, as its extremely strong class-imbalance allows for effective dataset ablations and highly indicative results. The same experiments conducted on a modified RUOD dataset (RUOD-4C) confirm the main findings, though the effects are less pronounced. For this reason, only the major RUOD-4C results are reported in our paper and some additional visuals can be found in Section 2.1. Similarly, when we present the classification results in the main paper (Paper Sec. 5.4), we only report the inter-class dependencies regarding scallop performance, as these are considered the key insights. Section 2.2 provides proof of that significance compared to other classes.

2.1. RUOD-4C

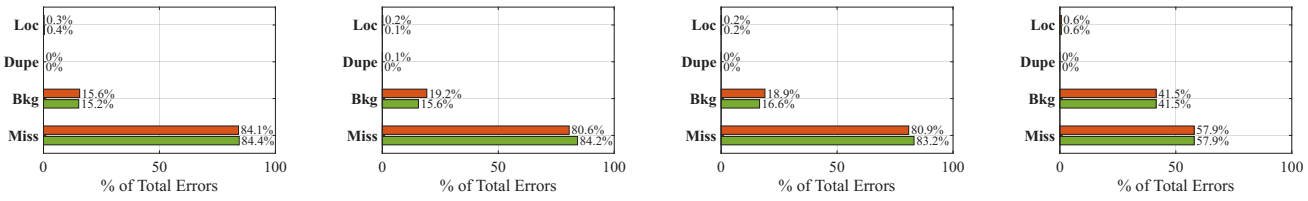
2.1.1 Additions to the Localization Study

The TIDE tool has contributed significantly to our localization study and its results on DUO were described in detail in Sec. 4.4 and Sec. 4.5 of the main paper. We apply TIDE to RUOD-4C (see Figure 1) and predominantly observe the error type distribution that we identified for the balanced DUO data, *i.e*. objects that blend into the background are completely overlooked and pose the biggest error risk. This pattern is expected to be evident in both, the original and balanced version of RUOD-4C, because its original class distribution is already nearly balanced (holothurian: 22%, echinus: 33%, scallop: 22%, starfish: 23%). Hence, these results support the claims made in our main paper.

Additionally, in light of the bounding box accuracy, this error profile with negligible localization errors is consistent with the findings on DUO and the IoU threshold comparison, indicating high box accuracy and minimal class differences in determining the exact object position.

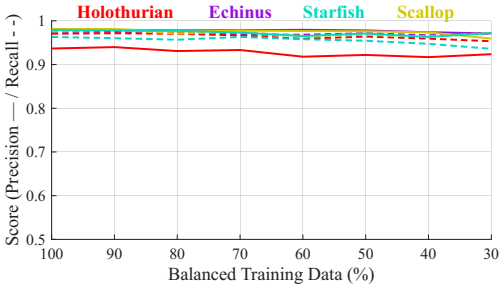
2.1.2 Additions to the Classification Study

One of the major insights from our classification study is the precision-recall tradeoff between balanced and imbalanced data distributions. However, as already explained in our paper, the RUOD-4C dataset does not notably show that because of its original fairly even distribution, leading to no significantly different subsets to compare. Nonetheless, we examine the balanced RUOD-4C data in more detail, as

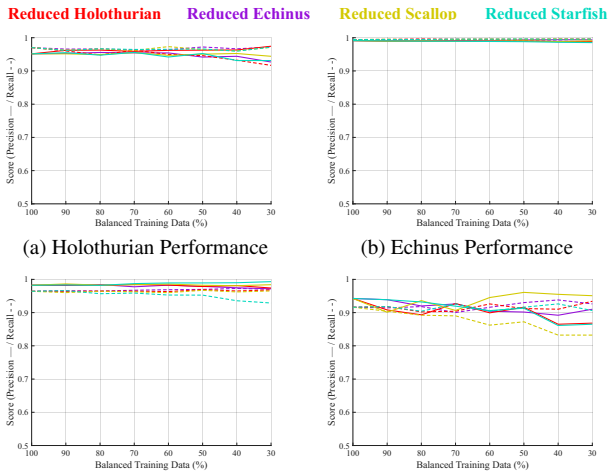


(a) Holothurian (b) Echinus (c) Starfish (d) Scallop

Supplementary Figure 1. Distribution of TIDE error types across all classes for both original (red) and balanced (green) RUOD-4C datasets. The error profiles do not change much from original to balanced. Missed ground truth objects are dominant type, followed by background error in every class.



Supplementary Figure 2. Per class classification performance on incrementally reduced balanced RUOD-4C subsets. Both, precision (solid lines) and recall (dashed lines) are generally high for all classes without major discrepancies. Precision values are slightly more variable than recall.



(a) Holothurian Performance (b) Echinus Performance

(c) Starfish Performance (d) Scallop Performance

Supplementary Figure 3. Classification performance change of every species when subject to class-specific data reductions in DUO. Echinus is extremely robust, starfish and holothurian are mainly influenced by reductions in their own training data, scallop is sensitive to data decrease in any class and notably shows the largest fluctuations.

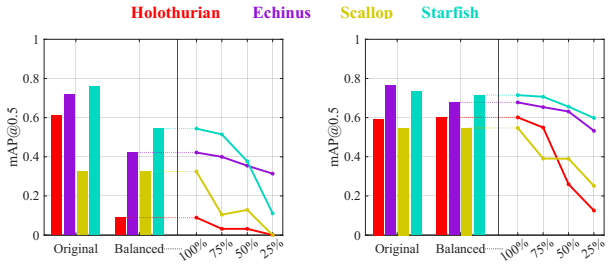
we have done for DUO in the main paper. Figure 2 shows the consistently high recall across all classes and a similar level for precision but with a larger range between classes. The values spread slightly more with less data in general, resulting in starfish still having better precision than recall, but for all other classes their respective recall is higher. This is coherent with our paper findings: extremely good recall is achieved with balanced data, whereas precision is more variable across classes with the same training data quantity.

2.2. Inter-Class Dependencies

When discussing the classification results in our main paper, we evaluate how sensitive scallops are to changes in data quantity per class (Paper Figure 7). We examine the increase or decrease of scallop precision and recall values from the original reference set to when one class is reduced to only 30% of its available training instances. This analysis is limited to the scallop class because we found that scallops are significantly more affected by data variations than the other classes, as directly visible in Figure 3. When monitoring performance over gradually decreasing data (one class at a time), the metrics generally changed most with the least data, as expected. However, there are fundamental differences as to what extent. The changes in echinus performance are minimal and fully negligible, whereas some effects on starfish and holothurian metrics can be observed, mostly linked to their own class-specific data. Scallops, on the other hand, rely on all the available data of any class, which is evident in very strong fluctuations and performance changes. This is the key insight of this sub-experiment that we aimed to present in our main paper using Figure 7.

3. Architecture Ablations

Our experimental framework presented in the main paper includes the systematic decomposition of the object detection pipeline into localization (Paper Sec. 4) and classification (Paper Sec. 5). While we report all representative results based on YOLO11 and ResNet-18 models, we here include complementary analyses with additional architectures. These experiments follow exactly the same framework as in the main paper and are intended to confirm the generality of our findings.



(a) DUO (b) RUOD-4C

Supplementary Figure 4. SSD localization performance per class compared across datasets. Bar charts indicate mAP@0.5 for the originally imbalanced and balanced data, line charts represent gradually reduced sets. Class differences are evident in all setups and SSD struggles with holothurians more than YOLO11. Still, the scallop class is mostly the worst-performing.

3.1. Extending the Localization Study with SSD

3.1.1 Implementation

We repeat our localization experiments with the popular Single Shot multibox Detector [3], specifically PyTorch's SSD300-VGG16, that is with input size 300x300 and VGG16 as backbone. Apart from the architectural change, we stay consistent with the settings *i.e.* 30 epochs, a batch size of 8, a confidence threshold of 0.25, a COCO-pretrained model state and a NVIDIA H100 GPU.

3.1.2 Results

Our main conclusions from the localization study as presented in our paper include that performance gaps remain evident under balanced conditions, pointing at inherent species characteristics and that foreground-background separation is the most challenging object detection step. In that regard, we show that once the presence of a target is detected, there are no significant differences between classes when determining the exact position afterwards anymore. These findings are supported by the results obtained using the SSD detector instead of YOLO11 in our experiments: Figure 4 and Table 1 compare various performance metrics between the different distributions and confirm persisting class-disparities. The SSD model achieves notably lower mAPs on every class in every dataset than we obtain with YOLO11 in our main paper, yet relative performance trends are similar. Although SSD is significantly challenged by holothurian mAP and echinus recall in the balanced DUO, the scallop class can still be identified as the most problematic one overall. It usually has the highest error rates and lowest recall and mAP.

The experiments using SSD also further verify that drawing accurate bounding boxes is not significantly easier or more difficult for certain classes, once the object has already been detected. Table 2 shows that class disparities become smaller when focusing on better-aligned boxes in mAP@0.5:0.95, consistent with the results in our main paper (Paper Table 3).

	Original Distribution (Imbalanced)			Balanced Sets		
	TPR	FDR	FNR	TPR	FDR	FNR
Ho	66.4 / 75.6	21.9 / 23.1	33.6 / 24.4	63.4 / 77.6	66.1 / 25.2	36.6 / 22.4
Ec	77.1 / 83.7	13.3 / 12.5	22.9 / 16.3	43.8 / 73.6	1.9 / 7.0	56.2 / 26.5
St	80.0 / 78.5	13.4 / 11.3	20.0 / 21.5	63.5 / 76.8	21.1 / 10.0	36.5 / 23.2
Sc	49.8 / 69.1	63.9 / 30.2	50.2 / 30.9	49.8 / 69.1	63.9 / 30.2	50.2 / 30.9

Table 1. SSD localization performance rates (True Positive Rate (TPR), False Discovery Rate (FDR), and False Negative Rate (FNR)) per class for DUO / RUOD-4C. Scallop performs better in SSD than in YOLO11 in terms of TPR/FNR but worse in FDR. It usually remains the worst-performing class.

	mAP@0.5		mAP@0.5:0.95	
Class	Value	Deviation	Value	Deviation
Holothurian	0.09 / 0.60	−0.45 / −0.11	0.03 / 0.30	−0.26 / −0.09
Echinus	0.42 / 0.68	−0.12 / −0.03	0.30 / 0.33	0.00 / −0.06
Starfish	0.54 / 0.71	0.00 / 0.00	0.30 / 0.39	0.00 / 0.00
Scallop	0.32 / 0.55	−0.22 / −0.16	0.19 / 0.29	−0.11 / −0.10

Table 2. DUO / RUOD-4C per-class mAP results of the balanced dataset and the deviation from the best-performing class using SSD model. The gaps decrease with increasing IoU threshold confirming the findings of our main paper.

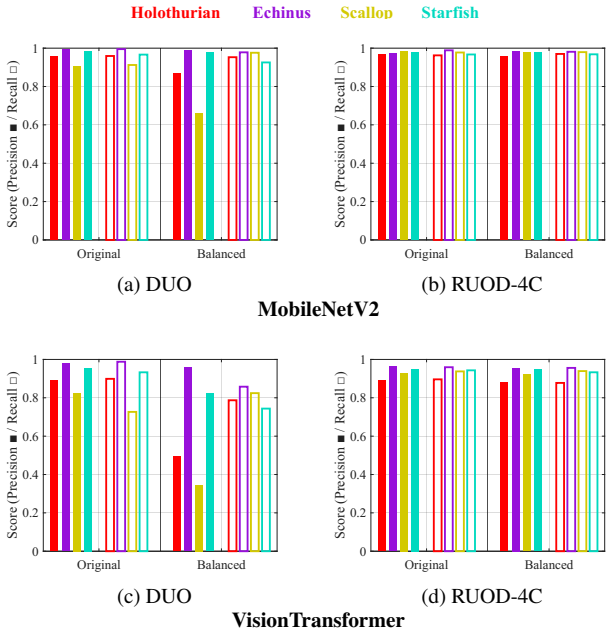
3.2. Extending the Classification Study with MobileNet and Vision Transformer

3.2.1 Implementation

To verify our classification results we conduct the same experiments as in Paper Sec. 5 again with the lightweight CNN architecture MobileNetV2 [4] and the transformer-based VisionTransformer-B/16 [1] as classifiers. Both are implemented in the exact same way as ResNet-18 in PyTorch, initialized with pretrained weights and then fine-tuned for 30 epochs, using a batch size of 32, input image size of 224x224, standard cross-entropy loss and the Adam optimizer.

3.2.2 Results

The main take-aways from our classification study presented in the paper include that performance is usually very high compared to the localization stage, but there is a notable tradeoff between recall and precision when comparing classification metrics between strongly imbalanced and balanced data versions. This is the case for scallops in DUO with all architectures: From imbalanced to balanced, we report a precision drop of 27.9% plus a recall gain of 4.6% using ResNet-18 in the main paper and we can see here in Figure 5 a precision decrease of 24.3% and recall increase by 6.4%



Supplementary Figure 5. Classification precision and recall for DUO (left column) and RUOD-4C (right column) across the additional architectures MobileNetV2 (top row) and ViT-B/16 (bottom row). While performance is steady on a high level for RUOD-4C, there are notable precision and recall changes with class-distribution in DUO. These patterns are consistent with ResNet-18 results reported in the main paper.

for MobileNetV2 as well as drastic -48% in precision and +9.8% in recall for VisionTransformer. Since the original and balanced RUOD-4C are very similar, there are no significant changes observed with other architectures, but this too stays consistent with the ResNet results.

In our paper, we also highlight the sensitivity of scallops as a minority and visually ambiguous class. Table 3 reports the results from the same experiment (reducing one class only) using MobileNetV2 and VisionTransformer. Both reveal the strong class-interdependence of scallops and underline their reliance on negative examples from other species. While holothurian and starfish also show some major performance changes with VisionTransformer, they stay mostly sensitive to their own training samples. All data reduction effects are pronounced significantly stronger in the ViT model, which might be related to transformer architectures generally requiring much larger datasets than CNNs for better performance [1].

Overall the results for the respective datasets remain consistent across alternative architectures and serve as further proof for our conclusions in the main paper.

	MobileNetV2				ViT-B/16			
Reduced:	Ec	Ho	Sc	St	Ec	Ho	Sc	St
Ec – P:	0.2%	-0.2%	-0.3%	-0.5%	0.7%	-2.9%	-0.8%	-1.3%
R:	-0.4%	0.0%	0.1%	0.2%	-3.3%	-0.5%	-0.5%	0.0%
Ho – P:	-1.3%	1.9%	-0.8%	-1.3%	-13.4%	1.3%	-7.8%	-6.7%
R:	0.9%	-3.6%	0.3%	0.8%	2.8%	-31.5%	-0.9%	1.1%
Sc – P:	-2.6%	-3.6%	1.6%	-5.0%	-9.6%	-19.2%	4.2%	-13.4%
R:	5.2%	3.0%	-11.9%	3.0%	-1.9%	-4.7%	-39.1%	2.2%
St – P:	-0.1%	-1.0%	0.2%	1.2%	-4.6%	-7.1%	-1.3%	2.6%
R:	0.2%	0.3%	-0.8%	-3.2%	-0.3%	-1.9%	-2.0%	-8.8%

Table 3. Relative change of classification performance metrics in DUO from 100% of available data to 30% per class. The reduced class is specified in the header (Ec = Echinus, Ho = Holothurian, Sc = Scallop, St = Starfish). We report positive (green) and negative (blue) changes in Precision (P) and Recall (R). The scallop class as a whole is the one most affected. Inter-class dependencies are generally more visible for VisionTransformer.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 13, 14

[2] Chenping Fu, Risheng Liu, Xin Fan, Puyang Chen, Hao Fu, Wanqi Yuan, Ming Zhu, and Zhongxuan Luo. Rethinking general underwater object detection: Datasets, challenges, and solutions. *Neurocomputing*, 517:243–256, 2023. 11

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 13

[4] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern recognition*, pages 4510–4520, 2018. 13