Improving Object Detection with Semi-Supervised Learning from Temporal Information

Viljar Holm Elvevoll[1], Kim Tallaksen Halvorsen[2], and Ketil Malde[2]
[1] Department of Informatics, University of Bergen, Norway
[2] Institute of Marine Research, Bergen, Norway
e-mail: viljarhe00@hotmail.no, {kim.halvorsen | ketil.malde}@hi.no

***Abstract*—Effective fisheries management and conservation depends on accurate and sustainable monitoring of marine biodiversity. Traditionally, fish population assessments have relied on manual annotation and invasive techniques that are labor-intensive and often detrimental to marine ecosystems they aim to preserve. Use of cameras in combination with automated object detection can make ecosystem observation both less invasive and less costly, but sufficient accuracy in species detection and classification can be a challenge, especially for low abundance species. Here we present a semi-supervised learning approach that leverages extensive unlabeled underwater video data to significantly enhance object detection performance for fish species. By integrating the YOLOv8 object detector with multi-object tracking algorithms ByteTrack and DeepSORT, a novel methodology is proposed to generate high-quality pseudolabels from temporal sequences. We show that iterative training that incorporates such pseudolabels consistently improved model precision and recall, with the best-performing approach (ByteTrack with an extrapolated heuristic) demonstrating average precision of 90%, recall of 70%, mAP50 of 74%, and mAP50-95 of 59%. Notably, scores improved substantially over the baseline supervised model on all metrics. These results underscore the potential of temporally informed pseudolabeling in enhancing fish detection accuracy and robustness, reducing reliance on manual annotations, and ultimately supporting more effective monitoring of fish communities.**

***Keywords*-Image classification; machine learning; species recognition; semi-supervised learning**

I. Introduction

Traditionally, methods for monitoring marine ecosystems include trawling, netting, and manual visual surveys by divers, which are labor intensive, costly, and often disruptive to habitats or producing bycatch [1]. Less invasive methods using underwater cameras, such as Baited Remote Underwater Video (BRUV) and Remote Underwater Video (RUV), are often an attractive alternative [2][3]. Most surveys employing these tools rely on manually scrutiny of video, which is costly in terms of time and labor. There is therefore a considerable incentive for leveraging object detection and classification models for semi or fully automated workflows [4][5]. However, training such models requires large amounts of high-quality, labeled data, which is costly to produce.

To address this limitation, we here investigate semi-supervised learning [6], an automated method to iteratively generate training data sets using predictions from preliminary models that are considered sufficiently reliable. To differentiate between labels determined by an expert annotator and those output from a model, the latter are referred to as *pseudolabels*. In contrast to earlier work, we selected the pseudolabeled data to use based on temporal information (*i.e.*, tracking) rather than more commonly used confidence scores. This is particularly advantageous in this setting, since an abundance of temporally contiguous video or image data can be produced, but expert annotation is time consuming and requires skilled curators.

The rest of the paper is structured as follows. In Section II, we describe the data set and the method for generating pseudolabels, as well as the training regime. In Section III, we present the results, and select the best performing method to investigate further. In Section IV, we discuss the results and their implications, and propose an explanation for the observations, before we conclude in Section V.

II. Methods

For this study, we used a data set consisting of 1248 images from a combination of sources (RUVs, photos by divers) under different conditions and with variable resolutions [7], (see Figure 1 for an example). The annotation by experts from the Institute of Marine Research include 10 categories (Figure 2): corkwing wrasse (*Symphodus melops*; male and female), two-spotted goby (*Pomatoschistus flavescens*), goldsinny (*Ctenolabrus rupestris*), rock cook (*Centrolabrus exoletus*), cuckoo wrasse (*Labrus mixtus*) male and female), pollack (*Pollachius pollachius*), ballan wrasse (*Labrus bergylta*), and *unknown* fish that could not be labeled to the species level due to low visibility or by being too distant from the camera. As shown in Figure 3, the number of objects (fish) per image varied, with a single fish being the most common, up to a maximum of 26 fish in one image.

In addition, six unannotated videos from similar habitats [7] were used as sources of pseudolabeled frames for the semi-supervised training. Each video has a resolution of 3840 x 2160 pixels and a frame rate of 30 frames per second. The average duration is approximately 8 minutes and 28 seconds with a minimum length of 6 minutes and 35 seconds and a maximum length of 8 minutes and 51 seconds. The videos were recorded by submerging a GoPro Hero9 Black camera near an underwater nest of a male corkwing wrasse to capture the surrounding environment. As some species were not present or very scarce in the unlabeled data, the semi-supervised method is only trained on five of the classes: male and female corkwing, two-spotted goby, goldsinny, and ballan. The videos may also record fish from other species which are not present in the training data labels. These factors are

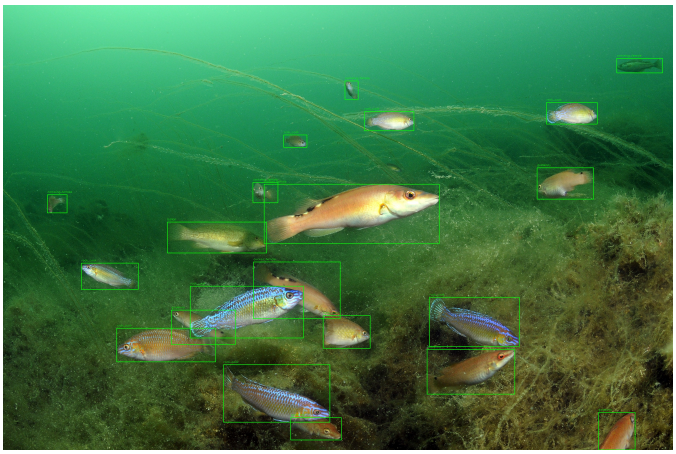


Figure 1. An example from the data set showing several annotated fish of various species (Photo by Erling Svendsen, used with permission).

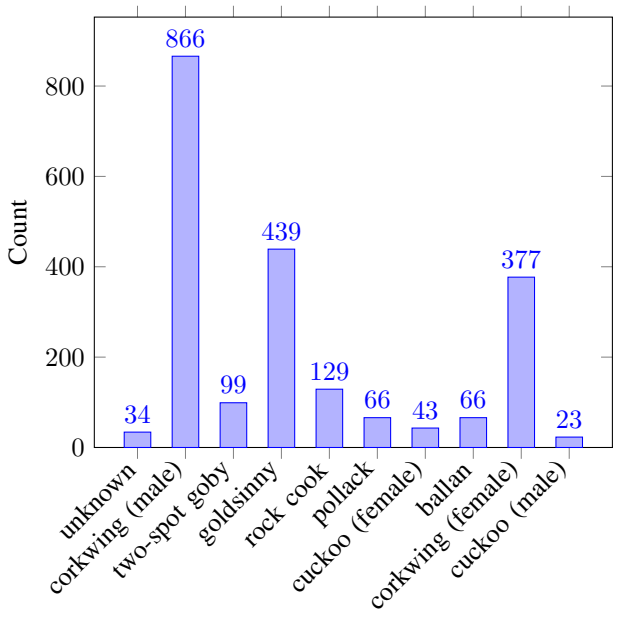


Figure 2. The distribution of the classes of annotated objects in the data set.

difficult to control when using real world data, but are likely to negatively affect the training process and consequently, the performance of the final classifier.

The object detection model used was YOLOv8 (Ultralytics, 2023), a state-of-the-art object detector from the family of single-pass models [8]. We used two different multi-object tracking algorithms, ByteTrack [9] and DeepSORT [10]. DeepSORT (Deep Simple Online and Realtime Tracking) is based on SORT [11]. Like SORT, DeepSORT uses a Kalman filter that predicts the next location of an object based on previous observations, but extends it with a deep learning based embedding method that matches the visual appearance of detected objects.

ByteTrack is a conceptually simpler method that links all detections (including low confidence ones) based on the overlap (intersection over union, or IoU) between detection bounding boxes. Although it is a less complex approach, it often performs well in independent benchmarks [12]. Both DeepSORT and ByteTrack have been adapted and developed to track object detector output for underwater images. For instance, the detection comparison module in DeepSORT has been modified to give better performance on fish [13], and the ByteTracks use of intersection-over-union has been adapted to fish morphology [14].

Each of the object trackers is combined with the two different strategies for pseudolabel generation illustrated in Figure 4 to construct four versions of an iterative pipeline, illustrated in Figure 5:

1. Base Model Training: An initial YOLOv8 model (YOLOv8x, the largest variant) was trained using a smaller, manually annotated dataset [7]. This provides the baseline for performance evaluation.

2. Pseudolabel Generation using Temporal Information: The trained model is then used to process unlabeled underwater video recordings and extract pseudolabeled data as illustrated in Figures 6 and 7. By integrating Multi-Object Tracking (MOT) algorithms (ByteTrack and DeepSORT), frames with objects that are missed or given low score by the detector can still be identified with high confidence. Two heuristics were investigated for selecting frames to generate pseudolabels:

A. Interpolated Intermediate Labels: This heuristic infers an object's presence in intermediate frames if it is detected by the model in preceding and subsequent frames, and the MOT algorithm assigns the same track ID. This method yields fewer but potentially highly accurate pseudolabels.

B. Extrapolated Labels: This more inclusive approach requires an object to be natively detected at least three times consecutively. All subsequent detections of that object by the MOT algorithm are included, forming an unbroken chain, even if the native model fails to detect it in every frame. This significantly increases the volume of pseudolabeled data.

To operationalize the two heuristics, we run two parallel inference streams on the same video frames. The first is a detector-only stream where YOLOv8 predicts each frame independently, and the second is a tracking stream where the detector outputs are linked over time by an MOT algorithm. Pseudo-label candidacy is then decided by comparing these two streams: the heuristics only accept samples when the tracker provides temporal evidence for an object in frames where the detector is inconsistent or missing. When multiple fish are present simultaneously, decisions are made per track ID, and ambiguous cases (e.g., potential ID switches or multiple competing associations) are skipped to avoid propagating uncertain assignments into the training data.

For the *interpolated intermediate labels* heuristic, we look for short temporal gaps where a track is present, but the detector misses it. Specifically, if the detector natively detects an object in a preceding frame t_1 and a subsequent frame t_3, and the tracker maintains the same track ID through an intermediate frame t_2, we infer that the object is also present in t_2 and add the tracker-provided box as a pseudo-label for that frame. During implementation we allow the "gap" to span

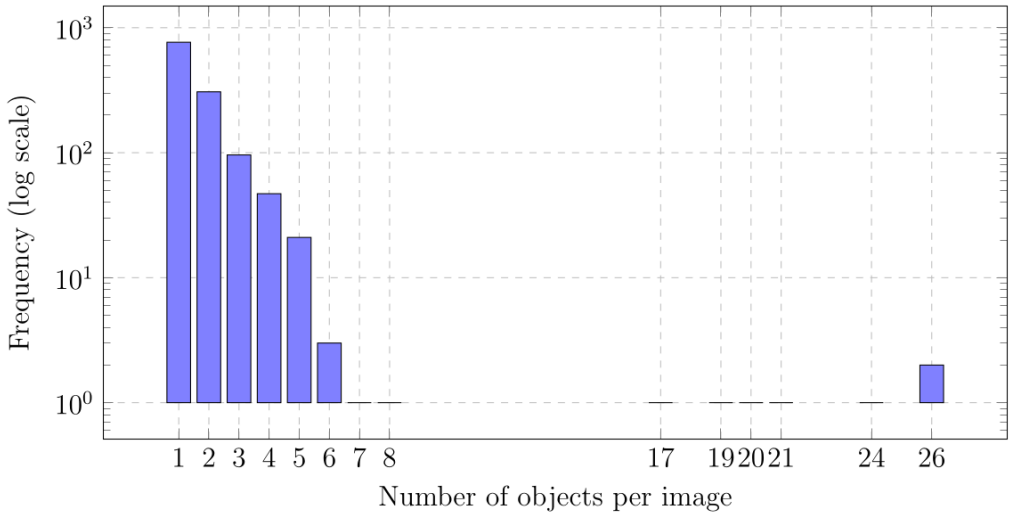


Figure 3. Distribution of the number of annotated fish objects per image in the data set.

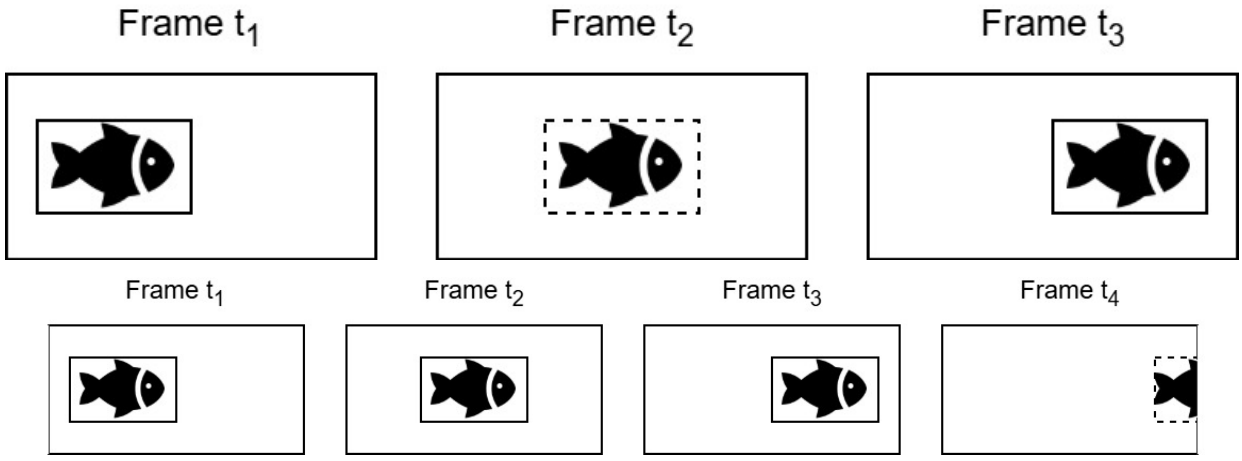


Figure 4. Two methods for pseudolabel generation. For interpolated labels (top row), a detection missed by the classifier is inferred from detections in the preceding and subsequent frames, targeting detector weaknesses on visually challenging frames. For extrapolated labels (bottom row), detections are inferred from a series of prior detections, which can add difficult partial views to the training data and improve robustness to occlusions.

up to a few consecutive frames to capture brief misses due to pose changes, turbidity, or partial occlusion, while still keeping the pseudo-labels tightly anchored to native detections.

For the *extrapolated labels* heuristic, we instead start from a reliable "seed" where the detector natively detects the same object at least three frames in a row, which also enables estimating a velocity for the track. From that seed onward, we include subsequent frames as pseudo-labels as long as (i) the tracker maintains an unbroken chain for the same ID and (ii) the detector does *not* natively detect the object in those frames. This produces substantially more pseudo-labeled training data, but it also increases the risk that tracker noise or a single incorrect native detection can propagate into many frames. The velocity requirement acts as a simple sanity check that helps exclude degenerate cases (e.g., static false positives) that would otherwise be easy to "track" over time.

This example illustrates the intuition behind the temporal selection criteria: the middle frame is visually challenging (unusual pose and angle), and the detector fails to produce a native detection. However, because the tracker maintains a consistent identity through the sequence, the middle frame becomes a high-value pseudo-label candidate that specifically targets a known weakness of the detector rather than reinforcing already-confident detections.

This second example highlights a characteristic outcome of the extrapolated heuristic: later pseudo-labeled frames may contain only a partial view of the fish (e.g., the tail), which would be difficult to label manually and is often missed by the detector. Including such cases increases data diversity and can improve robustness to partial occlusions, but it also emphasizes the importance of preventing tracker drift and ID switches from entering the training set.

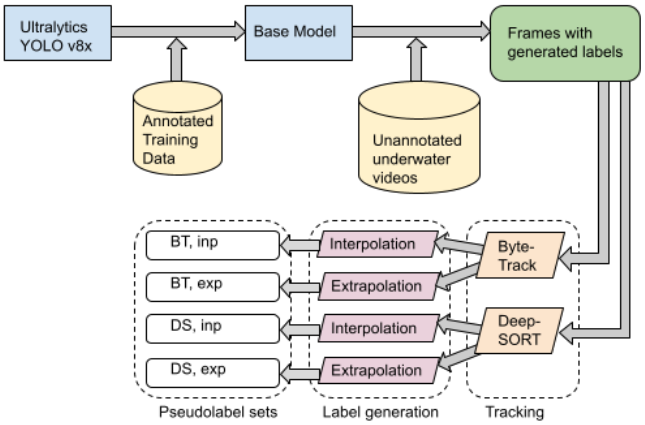


Figure 5. The process for generating the base model and the four pseudolabeled data sets (step 1 and step 2), using automated labeling of video data, tracking, and label interpolation and extrapolation.

The two heuristics also differ markedly in how much new training data they produce. In the first iteration of our pipeline, interpolated intermediate labels yielded on the order of a few thousand pseudo-labeled frames per tracker, while extrapolated labels yielded an order of magnitude more. This difference is important operationally: extrapolation increases diversity and coverage (including difficult partial views), but it also increases storage and compute requirements and can amplify any systematic tracker error. In other words, the heuristics represent a direct trade-off between expected label integrity and volume of additional training data.

3. Iterative Retraining: The pseudolabels generated are combined with the original labeled dataset, and the model is retrained. It is crucial to retain the original data to prevent catastrophic forgetting of classes not present in the video recordings. The learning rate of the AdamW optimizer is reset at the start of each new training phase to facilitate rapid adjustment and prevent trapping in suboptimal local minima. This iterative process (pseudolabel generation followed by retraining) is repeated for multiple cycles to progressively improve the model.

A practical detail in this iterative setting is *when* to stop each training phase. In our experiments, we used a fixed limit of 50 epochs per phase, but this number should be viewed as a sensitive heuristic rather than a universal default. Empirically, we observed that the base model's improvements often started to slow down around the same region, and saving a model state earlier preserved "plasticity" for the subsequent phase where new pseudo-labels are introduced. Standard early stopping is not directly applicable here: the goal is not to stop *at* convergence, but to stop *before* the model settles into a local minimum, while gradients are still sufficiently dynamic to adapt to the distribution shift introduced by new pseudo-labels. This is also why pseudo-labels are regenerated using the *current* model state at each iteration, and why we reset the optimizer learning rate between phases. Without a reset, we observed slower and more sporadic adaptation during relearning, sometimes plateauing prematurely. With a reset, it is common to see a brief dip in performance immediately after introducing new pseudo-labels, followed by recovery and improvement as the model re-optimizes on the updated training distribution.

TABLE I. PERFORMANCE METRICS AFTER 100 EPOCHS OF TRAINING.

Model	Prec	Recall	mAP50	mAP50-95
DS, inp	0.74988	0.56890	0.61023	0.47017
DS, exp	0.66251	0.56209	0.57969	0.45028
BT, inp	0.70776	0.53875	0.58142	0.44878
BT, exp	0.88792	0.62665	0.69120	0.52972

Note that for all cases, the validation and test set were the same, taken from the initial, annotated data set. In other words, pseudolabeled data were not used for evaluating model performance.

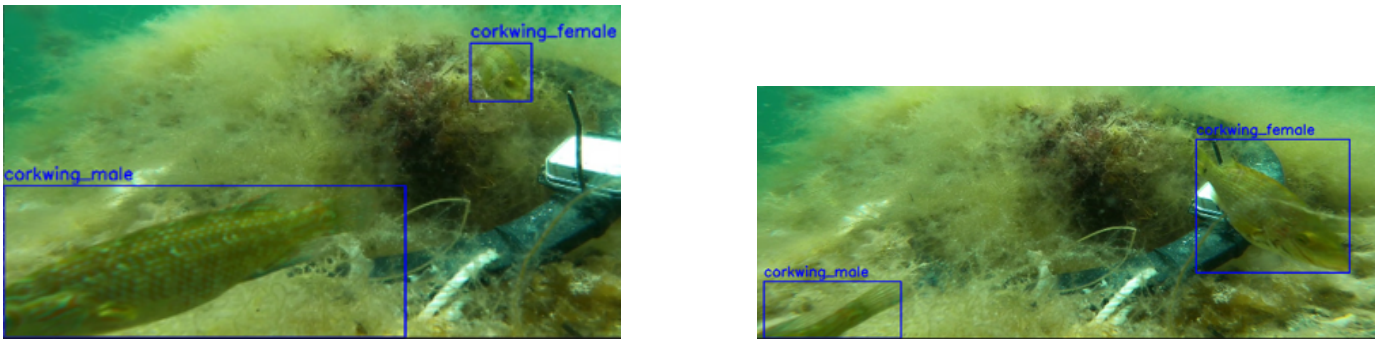
III. RESULTS

The study rigorously evaluated four configurations: ByteTrack with interpolated intermediate labels, DeepSORT with interpolated intermediate labels, ByteTrack with extrapolated labels, and DeepSORT with extrapolated labels. Each configuration was run for 100 epochs (50 supervised, followed by 50 semi-supervised with pseudolabels) which yielded the results in Table I. We see that all models perform adequately, and while ByteTrack with extrapolated labels consistently outperformed all other models, the ranking of the other approaches varied across metrics. Generally, DeepSORT with interpolation was the second-best performer, followed by DeepSORT with extrapolation, and ByteTrack with interpolation exhibiting the lowest performance for all metrics.

In order to explore the limits of semi-supervised training, the baseline and ByteTrack with extrapolation models were trained for 250 epochs. ByteTrack with extrapolation were chosed for comparison, since this comparison showed the best performance. In Figure 8, we can see how the different components of the loss rapidly decrease both for training (top row) and validation (bottom row) data, while the four different performance measures increase correspondingly. We also observe five distinct jumps in the graphs, these are caused by introduction of new data and resetting of the learning rate for each iteration, which cause an initial worsening of scores before the model gradually converges again. The loss curves provide additional insight into why the semi-supervised pipeline improves validation performance even though training becomes more "disrupted". The supervised baseline shows smooth convergence on the training data, as expected from repeatedly optimizing on a fixed dataset. In contrast, the semi-supervised approach experiences transient spikes in training loss at each iteration boundary due to (i) replacing pseudo-labels with new ones derived from the current model state and (ii) resetting the learning rate. Notably, the magnitude of these spikes tends to diminish over iterations, which suggests that successive pseudo-label sets become more consistent with one another as the model and tracker gradually align.

(a) Frame t_1 (model detection) (b) Frame t_2 (tracker only) (c) Interpolated label

Figure 6. Interpolated pseudolabel generation in practice. Note that due to ByteTrack's use of very low confidence predictions for interpolation, the inferred bounding box does not correspond exactly to the average of the two detections.



(a) Frame t_1 (model detection) (b) Extrapolated label

Figure 7. Extrapolated pseudolabel generation in practice. The visible part of the corkwing male exiting the frame is to small to be predicted, but is inferred from its presence in the previous frame.

Interestingly, the validation loss tells a different story: despite the validation set remaining unchanged and never receiving pseudo-labels, the semi-supervised validation loss trends lower over the full training schedule, while the supervised baseline tends to plateau earlier. This pattern is consistent with the semi-supervised approach improving generalization by adding difficult-but-informative training cases (frames the detector would otherwise miss), rather than merely strengthening performance where it is already confident.

Compared to the baseline model without using pseudolabels, we see from Figure 9 that semi-supervised training benefits from continued training for all metrics, and that this is reflected in validation loss shown in Figure 10. The performance statistics on the test set after 250 epochs are summarized in Table II. We see that using semi-supervised training with ByteTrack and the extrapolated pseudolabeling scheme results in substantial improvements for all metrics.

Per class improvements are shown in Figure 11. As expected, classes present in the semi-supervised training data (shown in solid colors) see substantial improvements on all metrics. Classes not present (shown with faded colors) see slight degradation in precision, and mAP, but surprisingly recall improves also for these classes.

Looking more closely at the evaluation metrics, the largest gains are typically observed in recall rather than precision. This is not unexpected: precision is often already high when the model is confident about species identity, whereas recall reflects how often the model can successfully detect harder instances. Since our pseudo-label selection explicitly targets frames where the detector fails but temporal continuity indicates the object is present, the additional training signal is biased toward "hard detections". This helps explain why recall improves substantially for pseudo-labeled classes, while precision changes are smaller and sometimes noisier.

We also observe that the semi-supervised curves can appear more sporadic than the supervised baseline on certain metrics. A plausible explanation is that the learning-rate reset and the introduction of new pseudo-label distributions can temporarily perturb the decision boundary, and metrics like mAP can be more sensitive to such perturbations than the loss. Despite this instability, the overall metric trajectories consistently favor the semi-supervised approach in our extensive training setting.

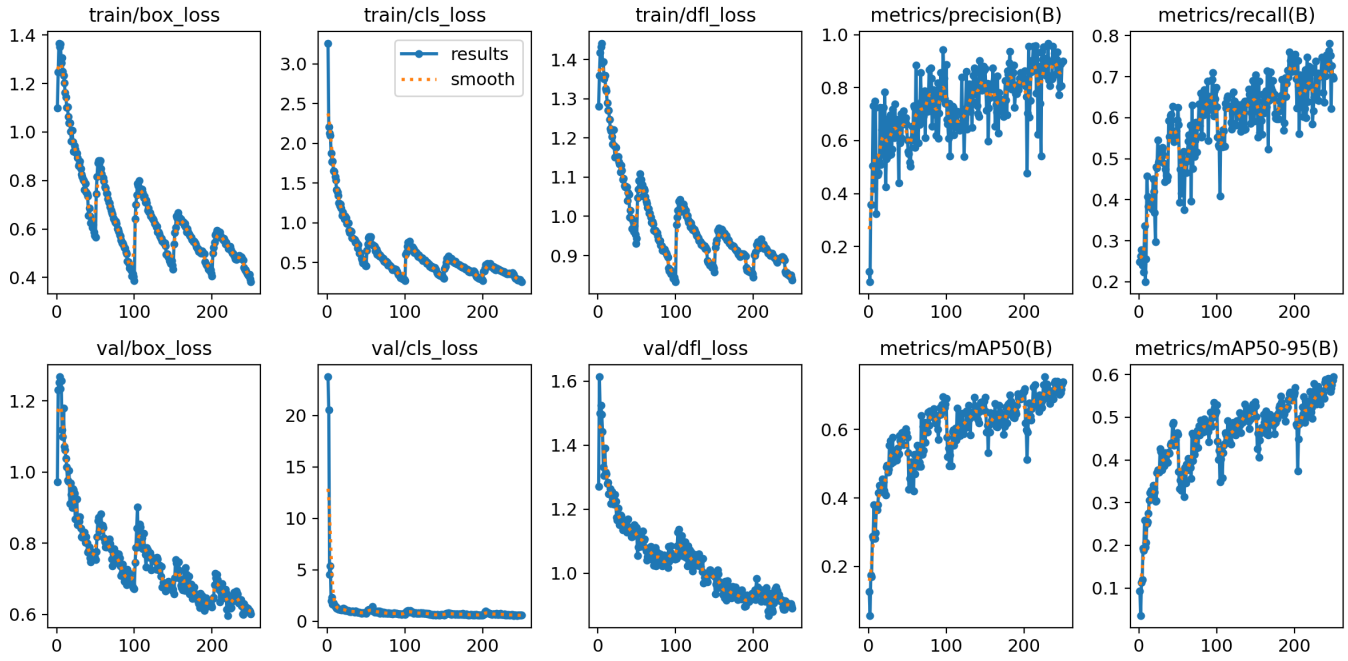


Figure 8. Training from ByteTrack tracks with extrapolated labels for 250 epochs (one iterations of supervised followed by four iterations of semi-supervised training).

TABLE II. PERFORMANCE OF BASELINE AND BYTETRACK WITH EXTRAPOLATION MODELS AFTER EXTENSIVE (250 EPOCHS) TRAINING.

Model	Prec	Recall	mAP50	mAP50-95
Base	0.75652	0.52468	0.57954	0.48811
BT, exp	0.90011	0.69644	0.73863	0.59468

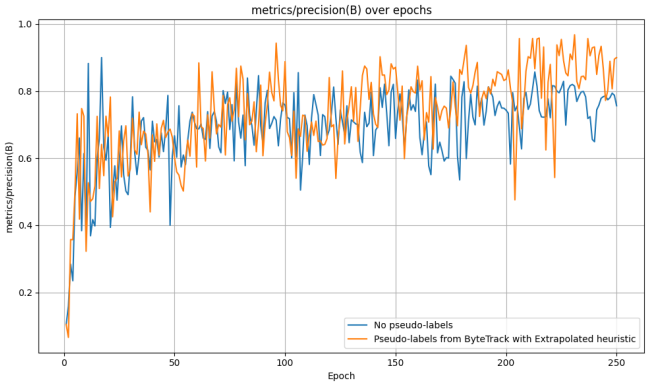
IV. DISCUSSION

It is well known that the performance of classifiers depends heavily on the amount of training data available. Especially powerful deep learning models with a large number of parameters can achieve great accuracy, but requires large amounts of high quality annotated data [15]. Semi-supervised training is seen as one way to obtain large labeled data sets at low cost, but it is important to ensure that the quality of the pseudolabeled data is sufficiently high. In addition, to optimize learning, the classifier needs to be provided with hard cases [16][17]. Using pseudolabels where the classifier has high confidence can ensure that the new annotations are trustworthy, but it also selects cases where the classifier is already strong and provide limited learning. Such an approach often leads to increased performance for the cases where the classifier is already performant (e.g., abundant species), at the detriment of poorly performing cases (rare species) [18][19]. Our use of tracking mitigates this problem. As each track is an ensemble of detections, reliability is increased. By selecting inferred observations that are supported by low confidence or missing predictions, we ensure that the new training data consists of hard cases, where the classifier has the greatest potential for improvement.

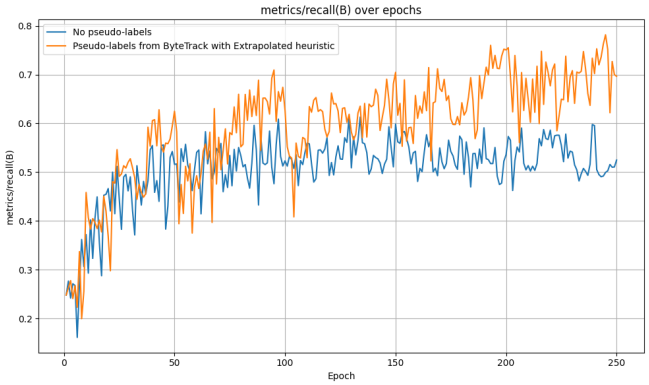
Here, we found that the most effective approach was ByteTrack combined with the extrapolated heuristic. This configuration consistently outperformed all other tested methods, as well as the baseline supervised model. The results demonstrate that leveraging temporal information through pseudolabeling significantly enhances fish detection accuracy and consistency, but also that the choice of pseudolabeling strategy and tracking algorithm is important. The substantial improvements in precision, recall, and mAP for the pseudolabeled classes, coupled with minimal negative impact on other classes, validate the effectiveness of this approach in mitigating annotation scarcity.

Pseudo-labels were only generated for five of the annotated classes in our labeled data, although the other species may still be present in the unlabeled videos. These species are excluded because they are rare, or perhaps observed at long distances or occluded, or due to other reasons, difficult for the current detector to recognize reliably. Occurrences of fish that are excluded by the process will be treated either as background or as incorrectly labeled fish by the subsequent training. Thus this introduces ambiguity: the detector and tracker may produce bounding boxes for objects that are outside the pseudo-label domain, but these are intentionally discarded to avoid injecting mislabeled samples into the training set. This constraint simplifies quality control, but it also means that the semi-supervised signal is intentionally restricted to a subset of the ecological variability present in the videos.

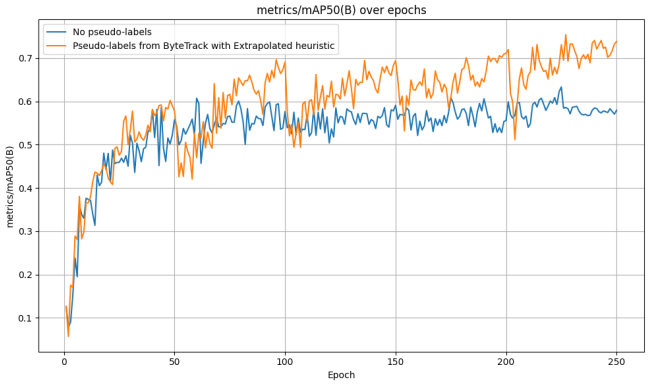
A crucial insight from this study is the progressive mitigation of initial model biases through iterative pseudolabeling. For instance, a systematic error where parts of the monitoring equipment were misclassified as "corkwing male" in early iterations (Figure 12) was effectively corrected and eliminated



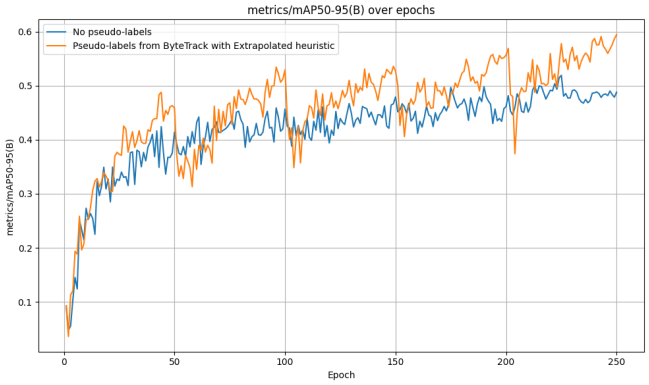
(a) Precision comparison



(b) Recall comparison

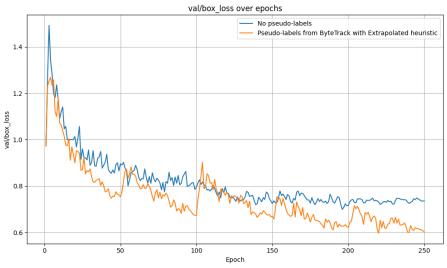


(c) mAP@50 comparison

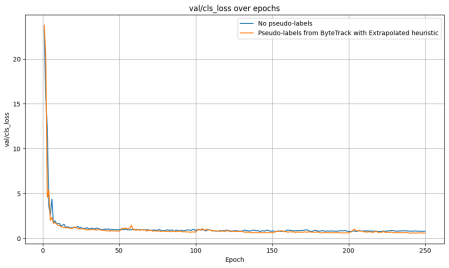


(d) mAP@50–95 comparison

Figure 9. Performance comparison across models.



(a) Val box loss



(b) Val class loss

(c) Val DFL loss

Figure 10. Validation losses during training.

in later iterations using ByteTrack with extrapolated labels.counting. Nevertheless, the iterative procedure can still correct This highlights the self-correcting nature of the temporalsuch biases when the tracker–heuristic combination does not semi-supervised framework, guiding the model towards morerepeatedly reinforce the error. In our experiments, this type of accurate predictions over time. A related observation is thatfalse positive disappeared in the final iteration for ByteTrack some systematic errors appear across multiple pipeline variantswith extrapolated labels, indicating that the evolving pseudo-early on, suggesting they originate from biases in the initial su-label sets increasingly contained counterexamples where the pervised model rather than from any single tracker or heuristic.structure was *not* labeled as a fish. This suggests an addi-In our case, a recurring false positive was linked to station-tional benefit of iterative temporal pseudo-labeling: beyond ary structures in the environment that frequently co-occurredimproving recall, it can also "wash out" certain dataset-specific with true instances of a target class in the labeled dataset.spurious correlations over time. At the same time, extrapo-Quantifying the full extent of such errors is difficult becauselation can be more susceptible to propagating false positives the erroneous boxes vary in location, shape, and confidence,(since a single error may be tracked for many frames), and we and the pseudo-labeled dataset can be too large for manualobserved indications that some tracker–heuristic combinations

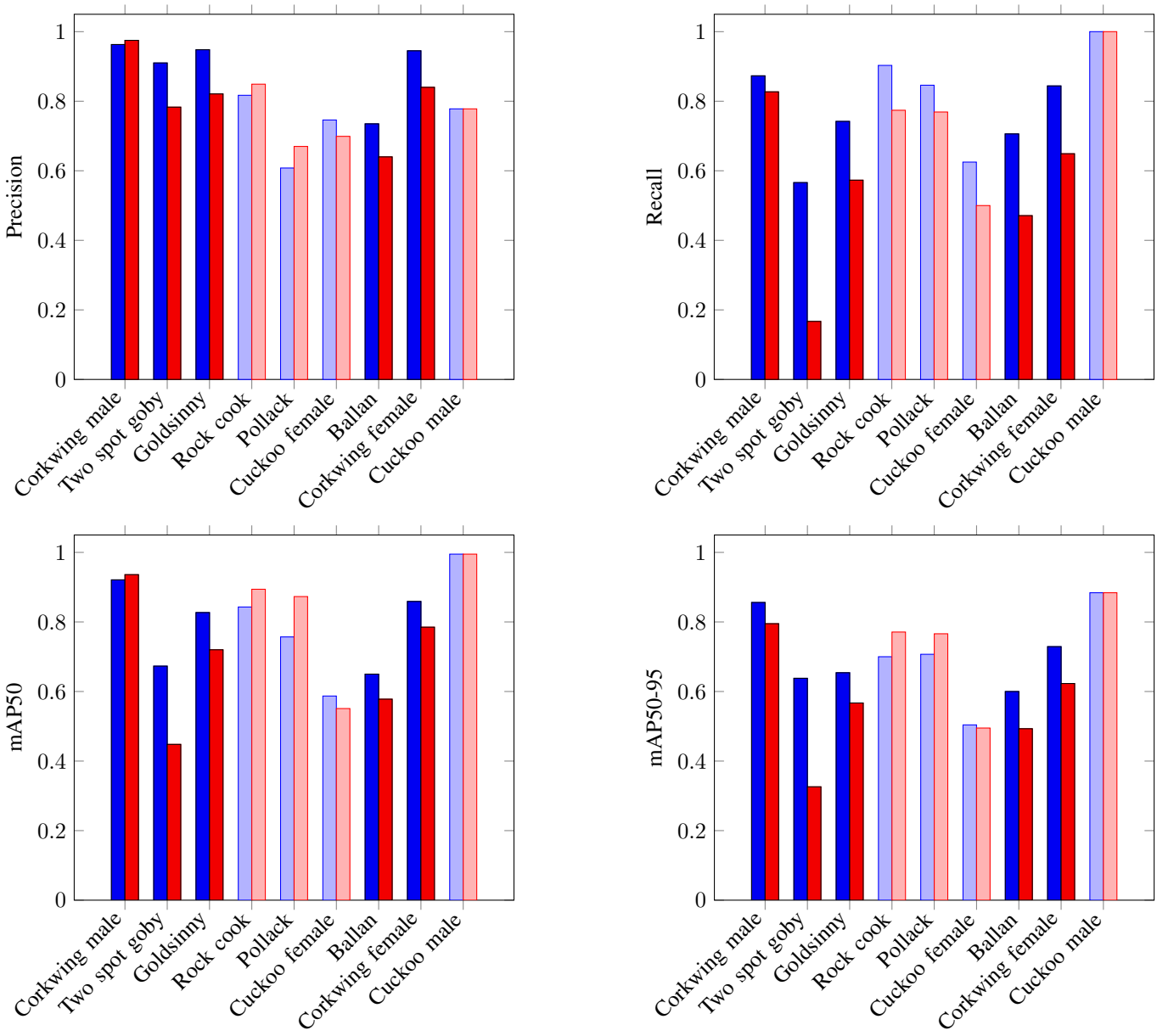


Figure 11. Precision (top left), recall (top right), mAP50 (bottom left) and mAP50-95 (bottom right) for baseline (red) and semi-supervised (blue) models. Classes not present in the pseudolabeled data shown with faded color.

may impede this bias correction compared to others.

The choice of MOT algorithm also proved critical. ByteTrack consistently outperformed DeepSORT when using extrapolated labels, but DeepSORT was better able to exploit interpolated labels. We suspect the discrepancy is caused by the use of Kalman filters in DeepSORT, which can interpolate predictions even when the object is lost by the detection model. While beneficial in predictable scenarios, this can lead to inaccurate pseudolabels for fish due to their often erratic movements, which can make DeepSORT predictions inaccurate. This phenomenon is reminiscent of the "off-policy" learning situation referred to as the "Deadly Triad" in Reinforcement Learning, which can destabilize convergence. ByteTrack, by contrast, relies solely on the detector's predictions, ensuring a stronger alignment between pseudolabels and the model's current capabilities, thus avoiding such instability. ByteTrack's reliance of overlap between detection boxes could be problematic in many settings, but for our data, a relatively high frame rate compared to fish movement speed ensures that detections of the same object from frame to frame would usually overlap.

An important nuance is that "better tracking" in a qualitative sense does not necessarily translate to "better pseudo-labels" for training. During exploratory experiments, DeepSORT often appeared visually stronger at maintaining identities, which

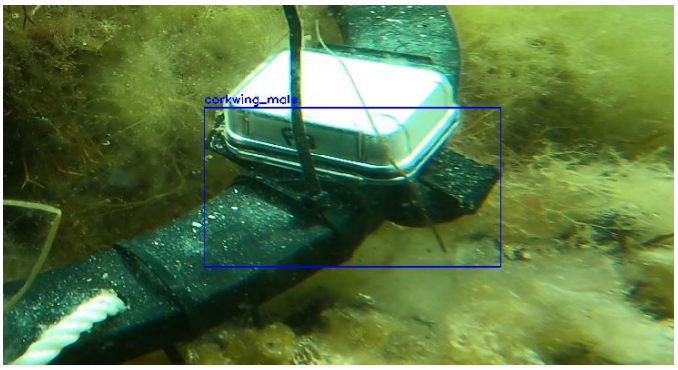


Figure 12. Equipment incorrectly predicted as "corkwing male" by the initial classifier.

made it a natural candidate for pseudo-label generation. However, when combined with the extrapolated heuristic, DeepSORT-generated pseudo-labels produced weaker downstream detector performance. A likely reason is that DeepSORT can continue producing track predictions through its motion model even when the detector has lost the object, effectively creating labels that are increasingly decoupled from what the detector currently considers plausible. This decoupling is especially problematic for fish, whose motion can be highly non-linear and strongly three-dimensional: a fish may rapidly change direction, rotate, or move toward/away from the camera such that a 2D motion model becomes a poor proxy for true dynamics. In such cases, Kalman-filter-driven interpolation can yield bounding boxes that do not match the visual evidence in the frame, and the resulting pseudo-labels can push the detector toward off-distribution targets. This aligns with the intuition behind the "Deadly Triad" analogy: when the training target (pseudo-label) becomes too disconnected from the model's own predictive distribution, iterative learning can become unstable. ByteTrack avoids this specific failure mode because its associations are anchored directly in detector-produced boxes, making pseudo-labels more "on-policy" with respect to the detector.

Semi-supervised learning has been used effectively in many different settings, but selecting pseudolabeled data to train on can be difficult. Using classifier confidence is an option [20], but tends in our experience to improve the classifier where it is already strong. Using augmentation [21] or taking class balance into account [22] may help to mitigate this, but by extracting pseudolabels from temporal information removes (or at least reduces) the dependence on the classifier itself from the selection process. Although temporal pseudolabeling methods have been attempted before (*e.g.*, [23]), our approach distinguishes itself by targeting the model's weaknesses rather than reinforcing its strengths. By relying on MOT algorithms to generate labels specifically where the base model fails to detect objects, it directly addresses gaps in detection capability.

V. Conclusion and Future Work

We have successfully demonstrated the significant potential of semi-supervised learning leveraging temporal information for enhancing object detection in marine life monitoring. By integrating YOLOv8 with ByteTrack, a robust methodology was developed to generate high-quality pseudolabels from unlabeled video data, substantially improving model performance for fish species. This approach reduces the reliance on costly and labor-intensive manual annotations, paving the way for more automation and scalable monitoring of the marine fauna. The insights gained regarding iterative bias mitigation and the critical role of MOT algorithm selection provide valuable directions for future research and practical deployment in real-world marine conservation efforts. We note that although we here target applications to marine ecosystems monitoring, we note that the techniques are general and equally applicable to the many other fields using object detection in video data.

Several practical limitations must be addressed for large-scale deployment. First, the current pipeline stores pseudo-labeled frames by extracting images from the video and writing separate label files, which is convenient but storage-intensive. A more scalable approach would treat pseudo-labels as references into the original video (video ID and frame index) alongside bounding boxes, extracting frames only on-demand during training. This would reduce storage overhead substantially and remove a clear bottleneck when scaling to many hours of footage. Second, the approach is currently reliant on a fixed per-iteration training budget (50 epochs) that is sensitive to dataset characteristics. A dynamic early stopping criterion tailored to the goal of stopping *before* convergence while learning remains sufficiently flexible could reduce compute cost and improve reproducibility across new datasets. More broadly, scaling the method will likely require more systematic exploration of per-tracker hyperparameters and heuristics, since applying the same training schedule uniformly across tracker–heuristic combinations may prevent certain variants from reaching their best performance. Finally, there are several promising directions to make the pseudo-label process more robust and operationally useful. One is to incorporate more videos and vary which videos contribute pseudo-labels across iterations (for example, starting with clear-water sequences and gradually introducing more challenging conditions). Another is to explore how and when the iterative process saturates: if the detector and tracker eventually agree on most frames, pseudo-label novelty may vanish, and continued training could yield diminishing returns or even regress toward the baseline distribution. Beyond object detection alone, improved detection consistency can also benefit downstream tasks that rely on clean trajectories, where tracking errors can otherwise propagate into higher-level behavioral or event classification.

Acknowledgment

An earlier version of this work was presented at IARIA COCE [1], and is based on results from the Master's degree thesis of VHE [24], which includes a more detailed exploration of the methods and data. The image data was collected and annotated as part of the CoastVision project, RCN grant

number 325862. We thank Tonje Knutsen Sørdalen and Torkel Larsen for contributing to the annotation work.

REFERENCES

[1] V. H. Elvevoll, K. T. Halvorsen, and K. Malde, "Semi-supervised object detection for marine monitoring using temporal information," in *COCE 2025: The Second International Conference on Technologies for Marine and Coastal Ecosystems*, Barcelona, Spain: IARIA, Oct. 2025, pp. 27–31, ISBN: 978-1-68558-329-3.

[2] A. W. Bicknell, B. J. Godley, E. V. Sheehan, S. C. Votier, and M. J. Witt, "Camera technology for monitoring marine biodiversity and human impact," *Frontiers in Ecology and the Environment*, vol. 14, no. 8, pp. 424–432, 2016. DOI: https://doi.org/10.1002/fee.1322. eprint: https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.1002/fee.1322. [Online]. Available: https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1002/fee.1322.

[3] S. K. Whitmarsh, P. G. Fairweather, and C. Huveneers, "What is Big BRUVver up to? methods and uses of baited underwater video," *Reviews in Fish Biology and Fisheries*, vol. 27, no. 1, pp. 53–73, 2017, ISSN: 1573-5184. DOI: 10.1007/s11160-016-9450-1. [Online]. Available: https://doi.org/10.1007/s11160-016-9450-1.

[4] H. Liu, X. Ma, Y. Yu, L. Wang, and L. Hao, "Application of deep learning-based object detection techniques in fish aquaculture: A review," *Journal of Marine Science and Engineering*, vol. 11, no. 4, p. 867, 2023.

[5] P. Rubbens et al., "Machine learning in marine ecology: An overview of techniques and applications," *ICES Journal of Marine Science*, vol. 80, no. 7, pp. 1829–1853, 2023. DOI: 10.1093/icesjms/fsad100.

[6] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[7] J. D. Poling et al., *Fjordfish*, version 1.1, Zenodo, Oct. 2025. DOI: 10.5281/zenodo.17950781. [Online]. Available: https://doi.org/10.5281/zenodo.17950781.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, 2016. arXiv: 1506.02640 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1506.02640.

[13] S. Liu, J. Zhang, H. Zheng, C. Qian, and S. Liu, "An improved deepsort-based model for multi-target tracking of underwater fish," *Journal of Marine Science and Engineering*, vol. 13, no. 7, 2025.

[14] W. Li et al., "When trackers date fish: A benchmark and framework for underwater multiple fish tracking," *arXiv preprint arXiv:2507.06400*, 2025.

[9] Y. Zhang et al., *Bytetrack: Multi-object tracking by associating every detection box*, 2022. arXiv: 2110.06864 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2110.06864.

[10] N. Wojke, A. Bewley, and D. Paulus, *Simple online and realtime tracking with a deep association metric*, 2017. arXiv: 1703.07402 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1703.07402.

[11] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, pp. 3464–3468.

[12] M. Abouelyazid, "Comparative evaluation of sort, deepsort, and bytetrack for multiple object tracking in highway videos," *International Journal of Sustainable Infrastructure for Cities and Societies*, vol. 8, no. 11, pp. 42–52, 2023.

[15] Y. Zhou, F. Tu, K. Sha, J. Ding, and H. Chen, "A survey on data quality dimensions and tools for machine learning," *arXiv preprint arXiv:2406.19614*, 2024.

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[17] R. Baldock, H. Maennel, and B. Neyshabur, "Deep learning through the lens of example difficulty," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 876–10 889, 2021.

[18] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang, "Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 857–10 866.

[19] L.-Z. Guo and Y.-F. Li, "Class-imbalanced semi-supervised learning with adaptive thresholding," in *International Conference on Machine Learning*, PMLR, 2022, pp. 8082–8094.

[20] K. Sohn et al., "A simple semi-supervised learning framework for object detection," *arXiv preprint arXiv:2005.04757*, 2020.

[21] J. Jeong, S. Lee, J. Kim, and N. Kwak, "Consistency-based semi-supervised learning for object detection," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[22] M. Xu et al., "End-to-end semi-supervised object detection with soft teacher," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3060–3069.

[23] R. J. Veiga et al., "Autonomous temporal pseudo-labeling for fish detection," *Applied Sciences*, vol. 12, no. 12, p. 5910, 2022.

[24] V. H. Elvevoll, "Semi-supervised object detection using temporal information," M.S. thesis, Department of Informatics, The University of Bergen, 2025.