Towards automated bycatch monitoring: optimizing and evaluating multi-object tracking of salmon in pollock trawls

Moses Lurbur*†[1], Katherine C. Wilson[2], and Noëlle Yochum[3]

[1]Pacific States Marine Fisheries Commission, under contract to Alaska Fisheries Science Center, National Marine Fisheries Service, NOAA, 7600 Sand Point Way Northeast, Seattle, WA 98115 USA
[2]Alaska Fisheries Science Center, National Marine Fisheries Service, NOAA, 7600 Sand Point Way Northeast, Seattle, WA 98115 USA
[3]Trident Seafoods, 5303 Shilshole Ave NW, Seattle, WA 98107 USA

February 26, 2026

*Corresponding author: mlurbur@cs.washington.edu
†Present address: University of Washington, Seattle, WA 3800 E Stevens Way, Seattle, WA 98195 USA

Abstract

Automated tracking of marine life during fishing operations can provide quantitative data on animal movement, improving our understanding of animal behavior, and informing sustainable fishing gear design and use. While many fields use multi-object tracking (MOT) to automatically track objects in videos, its application remains limited in commercial fishing due to the challenges of collecting and automatically annotating footage of dynamic marine environments.

We present the first application of MOT in a commercial Alaska walleye pollock (*Gadus chalcogrammus*) trawl, evaluating the feasibility of accurately tracking bycaught Pacific salmon (*Oncorhynchus* spp.). We provide a detailed comparison of five sizes of YOLO12 detection models and four tracking algorithms: BoT-SORT, ByteTrack, Intersection over Union, and Centroid. Due to limited computational resources available on many commercial fishing vessels, we also evaluate tracker performance at low frame rates.

Our findings demonstrate the feasibility of MOT in the commercial trawl environment, provide actionable conclusions about low-frame-rate tracking, outline effective tracker optimization techniques, and identify remaining challenges to improving MOT accuracy in this domain. The most accurate detection model (YOLO12x) and tracker (BoT-SORT) achieved a higher order tracking accuracy (HOTA) score of 59.5 on our challenging dataset. Evaluating tracker performance across frame rates revealed that simpler trackers outperformed more advanced Kalman-filter based trackers at low frame rates (10 and 7.5 FPS), indicating that frame rate should be considered during tracker selection. Object detection errors were the largest source of tracking error and should be the primary focus for improving tracking accuracy.

To facilitate future MOT applications and research in this domain, our code, model weights, and dataset are publicly available.

Keywords

Multi-object tracking, object detection, machine learning, computer vision, fisheries bycatch

1 Introduction

Fishing gear design and fisheries management practices play an important role in reducing discard in commercial fisheries (Cox et al., 2007; Tarantino et al., 2024; Worm et al., 2009). The discard rate from all U.S. fisheries fell by half between 2010 and 2015, from 22.3% to 10.5%, due in part to gear modifications and management changes (Savoca et al., 2020). Discard reduction efforts continue to be an important tool for improving ecosystem health, fishery sustainability, and the vitality of fishing communities (Benaka et al., 2019; Savoca et al., 2020). While discard rates in U.S. fisheries have declined in the last 20 years, trawl fishing still accounts for 72% of U.S. fisheries discards (Savoca et al., 2020). Developing more selective trawl fishing gear remains an important step toward further reducing discard rates.

The morphology and behavior of target and non-target species can inform the development of more selective fishing gear that limit discard and bycatch while still effectively retaining target species. Cameras placed *in situ* during fishing can provide important information about fish behavior, as well as catch rates, selectivity, and the presence of protected species (Abangan et al., 2024; DeCelles et al., 2017; Wilson et al., 2025; Yochum et al., 2021).

Video footage collected *in situ* during fishing allows for the analysis of actual fish behavior without the confounding variables and additional complexity of laboratory simulations. But the nature of commercial fishing presents significant challenges during and after data collection. Footage collected during fishing is often expensive and difficult to obtain, and video quality suffers from poor visibility, limited lighting, and high fish densities in some fisheries. Additionally, analyzing videos to extract data about behavior, gear interactions, and catch counts is a time consuming and often manual process that can lead to subjective bias (Bryan et al., 2024; Wilson et al., 2025; Yochum et al., 2021).

Developing methods to automate the analysis of footage captured during fishing would provide large volumes of quantitative data about fish movement with minimal human annotation and could inform the design of more selective fishing gear and bycatch reduction devices (”excluders”). Machine learning methods, such as object detection, have already been evaluated for detecting protected species in the Alaska walleye pollock (*Gadus chalcogrammus*; ”pollock”) fishery (Wilson et al., 2025).

The Alaska pollock fishery is the largest fishery in North America, worth more than 500 million dollars in 2022 (Fisheries, 2025). The prohibited species catch limits for Pacific salmon (*Oncorhynchus* spp.) are a significant concern for the fishery, which can shut down if the limits are reached. Fishermen use excluders in trawls to allow for the escapement of bycaught salmon (Ianelli et al., 2021). Currently, scientists and fishermen use recorded and live-feed videos collected inside pollock trawls to monitor fishing progress and evaluate salmon excluder performance (Bryan et al., 2024). The poorly lit, crowded, and highly dynamic environment in which the videos are collected make the detection and tracking of species uniquely difficult, even for human annotators (Wilson et al., 2025).

While object detection can automate the identification of species, it is not well suited for collecting behavioral information, which requires the ability to track individuals as they interact with fishing gear. In the pollock fishery, automating the tracking of salmon would enable scientists to efficiently collect detailed data about salmon behavior. This data could inform gear design to reduce salmon bycatch based on their behavioral differences with target species and potentially supporting the development of more advanced ”active” excluders (e.g., Rose and Barbee (2022)). Integrating tracking into existing camera systems would also provide similar benefits to other fisheries in the US and around the world, improving data collection and informing sustainable fishing practices.

Collecting movement-based behavioral information and accurate fish counts requires multi-object tracking (MOT), which relies on tracking algorithms and object detection models to track objects in videos. MOT is a mature field of computer vision and used commercially in autonomous driving, video surveillance, human behavior analysis, and virtual reality (Luo et al., 2021). MOT has also been applied to a range of scenarios in the marine environment. Katija et al. (2021) and Cai et al. (2023) integrate MOT with underwater robotic platforms to enable the efficient and semi-autonomous collection of large video datasets of marine species. Beyond marine research, MOT has demonstrated potential as a time-saving tool for practical fisheries applications, including automating the monitoring of fish health and behavior in aquaculture facilities (Cui et al., 2025) and automatically counting salmon in fish passages (Atlas et al., 2023). Regulatory agencies have also explored the use of MOT to automate electronic monitoring on commercial fishing boats (Khokher et al., 2022). MOT has also been used to identify meaningful insights about fish behavior even when track accuracy is low (Abangan et al., 2024). Across all of these applications, the diversity of the marine environment, crowded scenes, and limited datasets remain significant challenges.

Existing open-source fish tracking datasets FISHTRAC (Mandel et al., 2023) and BrackishMOT (Pedersen et al., 2023) both lack footage inside commercial fishing gear. FishTrack23 (Dawkins et al., 2024) does contain footage inside a pelagic trawl prototype, but the trawl footage accounts for only 1% of the dataset and does not contain a diversity of species or fishing conditions representative of the commercial fishing environment. To the best of our knowledge, there is no publicly available work examining the performance of MOT on footage recorded inside commercial pollock trawls.

Other popular marine fish datasets, such as FathomNet (Katija et al., 2022) and DeepFish (Saleh et al., 2020) do not include track-level annotations. This makes them useful for object detection, but not suitable for evaluating and optimizing tracking algorithms.

As part of our prior work, we built a diverse dataset of salmon and pollock track-level annotations collected in a commercial pollock trawl net in Alaska and analyzed the ability of object detection models to identify salmon and pollock (Wilson et al., 2025). After observing strong detection performance on our dataset, we saw the potential to move beyond detection and begin tracking individual fish.

In this study, we apply and evaluate MOT methods in the Alaskan walleye pollock fishery for the first time. We evaluated the feasibility of accurately tracking salmon in a commercial pollock trawl using our trawl dataset, open-source computer vision libraries, standardized evaluation metrics, and systematic, automated optimization techniques. Because of the potential to deploy edge computing devices with limited computational power in commercial fishing gear, we included a range of object detection model sizes (larger models generally achieve higher accuracy but require more computational power (Zhang et al., 2023)) as well as simple and advanced state-of-the-art tracking algorithms in our analysis. To further test the limits of tracking in resource-constrained conditions, we evaluated tracker performance on downsampled videos of 15, 10, and 7.5 frames per second (FPS) in addition to the dataset's original 30 FPS.

This work furthers our understanding of the capabilities and limitations of MOT technology for quantifying behavior of animals in the trawl environment and brings us a step closer to being able to use MOT data to build more selective and sustainable fishing gear. We also make the following contributions to the MOT field: (1) we provide a detailed analysis of five object detection models and four tracking algorithms on a new and challenging dataset; (2) we assess tracking accuracy at low frame rates required in resource-constrained applications such as edge computing; (3) we outline methods for automated tracker parameter optimization that eliminate the subjectivity of manual tuning and enable efficient adaptation to new frame rates and domains.

Our code and training data are available via GitHub (github.com/noaa-afsc-mace/salmon_tracking).

2 Methods

2.1 Dataset

For model training and tracker analysis, we used the salmon and pollock trawl dataset from Wilson et al. (2025). We partitioned the videos into datasets for object detection model training and tracker optimization and evaluation. Videos were annotated with bounding boxes, classes ("salmon" and "pollock"), and track IDs to distinguish between individual fish across frames. For complete documentation of the pollock trawl dataset, please refer to Wilson et al. (2025).

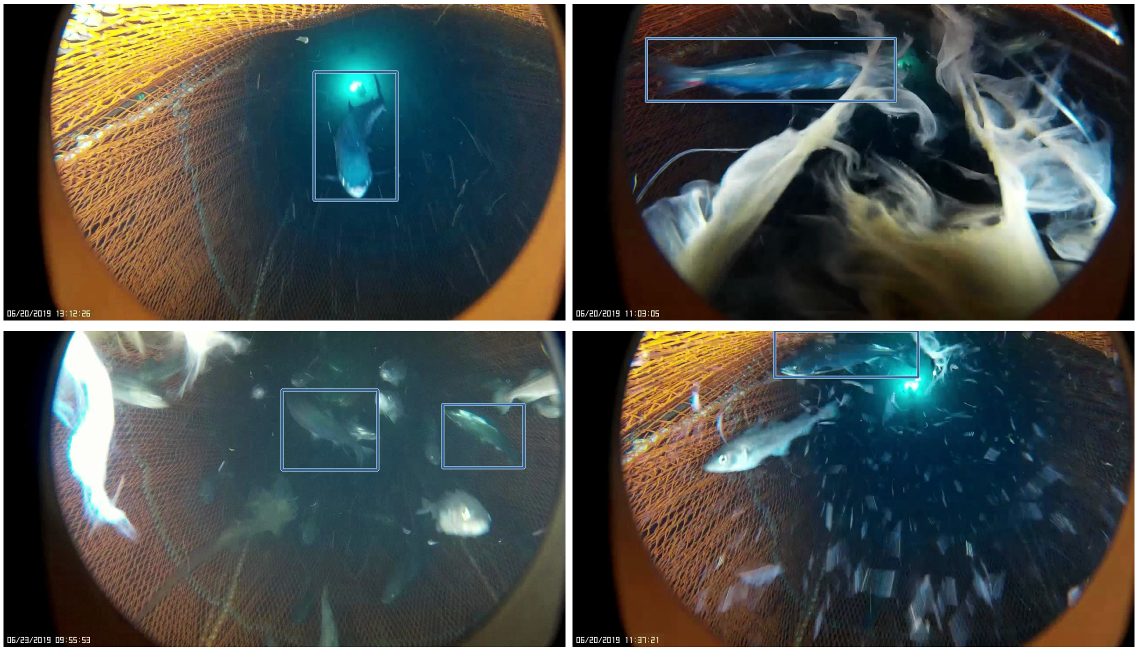

Figure 1: Example images from the training dataset. Salmon annotations are shown in blue, pollock annotations are omitted for clarity. Clockwise from top left: single salmon with good visibility, salmon partially occluded by jellyfish, salmon and pollock with krill, two salmon among pollock.

The full dataset includes 184 video clips, 168 of which contain salmon. Yochum et al. (2021) recorded the videos inside a commercial pollock trawl equipped with an experimental salmon excluder in Alaska's eastern Bering Sea during the summer pollock commercial fishing season in 2019 and 2020. Video was collected using the Sexton Corporation's trawl camera systems over multiple tows. All footage was recorded at 30 frames per second, with a resolution of 1280 × 720 pixels in 2019 and 1920 × 1080 pixels in 2020 (Wilson et al., 2025). To capture variability in fishing conditions, we randomly selected salmon video clips from data collected over four fishing tows in 2019 and three tows in 2020. We also included a small selection of clips without any fish present to ensure a representative dataset for evaluation of object detection models.

Four human annotators labeled the footage with bounding boxes, classes, and track IDs (process described in detail in (Wilson et al., 2025)). A single person reviewed all annotations to ensure consistency and accuracy. Annotators labeled all salmon and pollock tracks from the time the fish could first be identified entering the camera's field of view until the fish could no longer be identified or the fish became indistinguishable from the background. Annotations include partial occlusions. During full occlusions, annotators began a new fish track only when they could not confidently re-identify the same fish.

We randomly partitioned the 184 clips in the data set into two groups, one for model training and one for testing, with roughly 80% of the annotations being used for object detection model training (Table 1). For example images, see Figure 1.

Table 1: Object detection dataset properties. Training data were used to fine-tune object detection models. Test data were used for model evaluation.

Dataset split	Salmon tracks	Salmon annotations	Pollock annotations	Frames	Video clips
Train	168 (77%)	9,213 (80%)	56,784 (77%)	13,569 (80%)	142 (77%)
Test	51 (23%)	2,362 (20%)	16,624 (23%)	3,429 (20%)	42 (23%)
Total	219	11,575	73,408	16,998	184

For the tuning and evaluation of salmon tracking, we discarded video clips not containing salmon tracks. The resulting tracking dataset has identical test and train sets as the object detection dataset, except for the removed non-salmon videos (Table 2). All annotations were converted to the MOT format (Dendorfer et al., 2019), the standard annotation format for tracking evaluation.

To analyze tracking performance at lower frame rates, we downsampled videos and annotations, keeping every other, second, and third frame to simulate tracking at 15, 10, and 7.5 frames per second respectively. We discarded annotated salmon tracks that spanned fewer than two frames after downsampling. For downsampled tracking dataset details, see Table 8, Table 9, and Table 10.

Table 2: Tracking dataset properties. Training data were used to optimize tracker parameters. Testing data were used for tracker evaluation. The test and train sets are identical to the object detection dataset, except clips without salmon tracks were removed.

Dataset split	Salmon tracks	Frames	Video clips
Train	168 (77%)	12,759 (79%)	128 (76%)
Test	51 (23%)	3,309 (21%)	40 (24%)
Total	219	16,068	168

2.2 Object detection

To analyze how object detection model performance impacted tracking performance, we fine-tuned five pre-trained object detection models in the You Only Look Once (YOLO) model family (Redmon et al., 2016).

We used five pre-trained versions of the YOLO12 model architecture, using the pre-trained model weights given by Ultralytics for fine-tuning (Table 3) (Jocher et al., 2025). We chose these models because they are accurate, open-source, widely used, and well supported in the Ultralytics package. Our previous work also showed that YOLO11n performed well at detection on this dataset (Wilson et al., 2025). Training a wide selection of models allowed us to compare the performance of all YOLO12 model sizes.

Table 3: YOLO model properties. Image size indicates the width and height of the input images to the model. Parameters are trainable variables in a model. Larger models have more parameters. Floating point operations (FLOPs) describe how many operations are required to run a single instance of a model. Faster models require fewer FLOPs.

Model	Image size (pixels)	Parameters (M)	FLOPs (B)
YOLO12n	640	2.6	6.5
YOLO12s		9.3	21.4
YOLO12m		20.2	67.5
YOLO12l		26.4	88.9
YOLO12x		59.1	199.0

We used fine-tuning, a form of transfer learning that adapts a pre-trained model to a specific task, to train the models to detect salmon and pollock using our object detection dataset. We trained all models using Ultralytics' default hyper-parameters. To prevent overfitting, we stopped model training after validation metrics did not improve for 100 epochs. We trained and evaluated models using an n1-highmem-32 Google Cloud virtual machine with four NVIDIA T4 GPUs. After training, we saved the best model weights for evaluation and use with trackers.

For object detection model evaluation, we calculated Average Precision (AP), mean Average Precision (mAP), mAP50, precision, and recall using Ultralytics' built-in evaluation methods. Average Precision is calculated by averaging the area under the precision-recall curve at intersection over union (IoU) thresholds of 0.5 to 0.95 with a step size of 0.05. IoU measures the overlap between a predicted bounding box and the ground truth annotation, with values ranging from 0 (no overlap) to 1 (perfect overlap). Mean Average Precision is the mean AP for all classes (salmon and pollock); mAP50 is similar to mAP but calculated at a single IoU threshold of 0.5. Precision is the fraction of all model detections that are correct. Recall is the fraction of all true instances that the model detects. All metrics were calculated using a model confidence score threshold of 0.001.

2.3 Tracking

2.3.1 Tracker Implementation

Many of the most recent MOT algorithms rely on the association framework of the foundational simple online and realtime (SORT) tracker (Bewley et al., 2016). The basic tracking steps of SORT are:

1. *Detect:* An object detection algorithm, such as the You Only Look Once (YOLO) model (Redmon et al., 2016), predicts the location of objects in a single frame of a video and passes them to the tracker.

2. *Filter:* The tracker filters detections based on their confidence score; generally trackers discard low confidence detections.

3. *Predict location:* The tracker predicts the current location of existing tracks based on previous frames, usually using a Kalman filter (Welch and Bishop, 1995).

4. *Compute similarity:* The tracker computes a similarity matrix between the predicted location of existing tracks and new detections using a metric such as intersection over union (IoU) or image feature vectors.

5. *Match tracks:* Using the similarity matrix, the tracker matches new detections with existing tracks using an assignment algorithm, such as the Hungarian Algorithm (Kuhn, 1955)

6. *Delete unmatched tracks:* The tracker deletes tracks that have remained unmatched for too many frames.

7. *Create new tracks*: The tracker creates new tracks based on any remaining unmatched detections.

We evaluated the salmon tracking performance of two state-of-the-art SORT-based trackers, BoT-SORT (Aharon et al., 2022) and ByteTrack (Zhang et al., 2022), as well as our own implementation of two simpler trackers that use IoU and the Euclidean distance between detection centroids to associate tracks (hereafter referred to as the IoU and Centroid trackers). We also dramatically sped up the process of evaluation and parameter optimization by adding features to the Ultralytics package that allow saved detections to be reused.

ByteTrack's primary difference from the SORT algorithm is its use of low confidence detections. ByteTrack divides detections into high and low confidence detections. Detections above the high confidence threshold are matched with existing tracks first. The remaining detections with confidence scores lower than the high confidence threshold and greater than a minimum confidence threshold are matched with the remaining tracks. After matching, new tracks are created from the remaining detections with a confidence score above the new track threshold. BoT-SORT is an adaptation of ByteTrack, with improvements to the Kalman filter location predictions, the addition of camera motion compensation, and the option to fuse IoU and image features for re-identification (Re-ID) to improve track matching. We do not use Re-ID because it was not supported in Ultralytics' implementations of ByteTrack and BoT-SORT at the time of our analysis. BoT-SORT used the sparse optical flow method for global motion compensation.

The IoU and Centroid trackers simplify the tracking process by omitting the use of Kalman filters and location prediction. They filter out low confidence detections and match detections and tracks based on IoU or the Euclidean distance between the centers of detections and the bounding boxes of existing tracks. We contributed our implementations of the simpler and faster IoU and Centroid trackers to our fork of the Ultralytics package.

Generating detections is the most computationally expensive and slowest step in the tracking process. Adding the ability to use saved detections meant we only generated detections once for a given video and detection model, speeding up the evaluation and optimization process significantly. For all detections, a confidence score threshold of 0.0 was used to ensure that a tracker's minimum detection confidence threshold parameter was never lower than that of the detection model used to generate the saved detections.

2.3.2 Tracker evaluation

We used the TrackEval package (Luiten, 2025) for tracker evaluation and the higher order tracking accuracy (HOTA) metric (Luiten et al., 2021) as our primary metric. We also manually reviewed visualized tracking results and performed regression analyses to better understand how object detection model accuracy, the number of fish per frame, and track length impact tracking accuracy. All 12 combinations of tracking algorithms (IoU, Centroid, ByteTrack, BoT-SORT) and object detectors (YOLO12n, YOLO12m, YOLO12x) were evaluated on the entire test set of the tracker evaluation dataset (Table 2). For downsampled frame rates, all trackers used YOLO12x detections.

Several MOT benchmark sites use TrackEval as their official evaluation code and it supports many tracking metrics (Luiten, 2025). We updated the evaluation class names in the TrackEval repository to support our custom object class: ”salmon”.

We chose HOTA as our primary tracking metric because it quantifies all aspects of tracking performance and is widely used in MOT literature. HOTA scores range from 0 to 100, with a score of 100 meaning that a tracker perfectly detects, aligns bounding boxes, and links detections to the correct objects across a video.

The HOTA metric is composed of sub-metrics that quantify different aspects of tracking performance. In addition to the overall HOTA score, we included five HOTA sub-metrics: detection accuracy, localization accuracy, association accuracy, association recall, and association precision. Detection accuracy measures how well all annotated objects are detected; localization accuracy measures how well each predicted object aligns with its corresponding annotation bounding box; and association accuracy measures how well the tracker links detections over time to the same identities. We also report the sub-components of association accuracy: association recall and precision. Association recall measures how well trackers can avoid splitting the same object into multiple shorter tracks. A low association recall is caused by a tracker splitting a single track into multiple predicted tracks. Association precision measures how well a tracker can avoid merging multiple objects together into a single track. A low association precision is caused by a tracker extending a track over multiple different objects.

To facilitate comparison with other studies, we also report two other common tracking metrics: multiple object tracking accuracy (MOTA) (Bernardin and Stiefelhagen, 2008) and IDF1 (Ristani et al., 2016).

In addition to a quantitative review of tracking performance using HOTA, we also performed a manual review of tracker performance by visualizing track annotations, predicted tracks, and track IDs on video clips. We reviewed tracker performance for every clip in the test dataset for each tracker paired with YOLO12x. For each video, we recorded the presence of detection errors, localization errors, split tracks, and merged tracks.

We used ordinary least squares (OLS) regression to analyze relationships between tracker performance and the variables mAP, average fish per frame, and average length of salmon track. To get mAP scores for each clip in the test dataset, we ran separate model evaluations for every clip. We also used our annotations to determine the average number of salmon and pollock per frame for each clip. We used the regressions to find a line of best fit and the R-squared correlation value. *P*-values were calculated based on a *t*-distribution with $\alpha = 0.05$.

2.3.3 Optimization

We used the Bayesian optimization package SMAC3 (Lindauer et al., 2022) to search for the optimal tracker parameters. Trackers were optimized on the training set of the tracking dataset. SMAC3 uses a Bayesian optimization technique to determine the optimal input parameters for a ”black-box” function (a tracking algorithm in our case) by repeatedly evaluating the function with different input parameters. We chose SMAC3 because it is an efficient, effective, and well maintained tool for hyper-parameter optimization (Madrigal et al., 2019).

Manually selecting parameters is subjective and could result in biased or sub-optimal performance. Automated optimization removes any potential human bias introduced by manual parameter tuning and ensures all trackers are optimized identically. It also allows the efficient optimization of a tracker for a particular detection model and video frame rate.

Table 4: All optimized tracking parameters. Cells are left blank if the parameter is not used by the tracker.

Parameter	Function	BoT-SORT	ByteTrack	IoU	Centroid
Match threshold	Used for matching new detections with existing tracks. It is the maximum allowable "cost" or distance for linear assignment of tracks and detections.	X	X	X	X
Track threshold	The minimum allowable detection confidence.			X	X
Track buffer	The maximum amount of time (frames in our case) that a track can exist without being matched with a new detection.	X	X	X	X
New track threshold	The minimum allowable detection confidence required to create a new track if the detection is not matched with an existing track.	X	X		
Track high threshold	The minimum allowable detection confidence for "high confidence" detections used during the first matching phase of ByteTrack and BoT-SORT.	X	X		
Track low threshold	The minimum allowable detection confidence for "low confidence" detections used during the second matching phase of ByteTrack and BoT-SORT.	X	X		
Fuse score	Indicates whether to use confidence scores and IoU distances during matching.	X	X		

All of our trackers have several parameters that significantly influence their performance and optimal parameters can vary depending on the object detection model and tracking scenario (see Table 4 for a complete list of optimized parameters). Proximity and appearance thresholds are only used for Re-ID and were omitted from optimization. We did not consider other global motion compensation algorithms for BoT-SORT as part of optimization.

We used the SMAC3 optimizer to search for optimal tracker parameters for all four trackers with each of these three detection models: YOLO12n, YOLO12m, and YOLO12x. When optimizing trackers for lower frame rates, we used YOLO12x detections. For each tracker and model, we ran 75 search iterations over the entire training dataset using a deterministic black-box optimization scenario. We chose the number of search iterations qualitatively to balance optimization time and the observed convergence of parameters. The optimizer determined the best parameters using the HOTA score for the entire training dataset.

We used bounds to limit the parameter search within logical ranges for all optimized parameters (Table 11). During optimization for BoT-SORT and ByteTrack, we used scaling factors between zero and one (inclusive) to calculate the new track threshold and track high threshold. Using scaling factors ensured that the track low threshold is always the lowest confidence threshold, and that the track high and new track thresholds are between zero and one (inclusive). We calculated confidence thresholds based on the scaling factor and track low threshold using Equation 1.

$$\text{newThreshold} = \text{trackLowThreshold} + (1 - \text{trackLowThreshold}) \cdot \text{newThresholdScalingFactor} \quad (1)$$

3 Results

The best tracker and detection model combination, BoT-SORT and YOLO12x, achieved an average HOTA score of 59.5 on our test dataset using SMAC3 optimized tracker parameters (see Figure 2 for visualized tracking results).

Object detection model accuracy significantly influenced tracking accuracy; BoT-SORT's HOTA scores surpassed 80 for clips with mAPs above 80, but fell below 40 when mAP was low (Figure 5). The largest and most accurate detection model, YOLO12x, improved tracking accuracy over the smallest model, YOLO12n, by an average of 4.7 points across all trackers (Figure 3). While BoT-SORT had the highest HOTA score on our dataset at the original 30 FPS, the simpler IoU and Centroid trackers scored higher on the downsampled 15, 10, and 7.5 FPS clips (Table 17).

Using SMAC3 to automatically select optimized tracker parameters for every combination of tracker, detection model, and frame rate allowed us to efficiently and fairly compare tracker performance. SMAC3 optimized parameters performed slightly worse than Ultralytics' default parameters for BoT-SORT and ByteTrack at 30 and 15 FPS, but performed better than defaults at 10 and 7.5 FPS (Table 18).

The 219 salmon tracks in the dataset ranged in length from 0.7 to 35.6 seconds, with an average length of 1.8 seconds (Figure 7).

Unless otherwise noted, all discussion of tracker performance is based on the SMAC3 optimized trackers using YOLO12x detections.

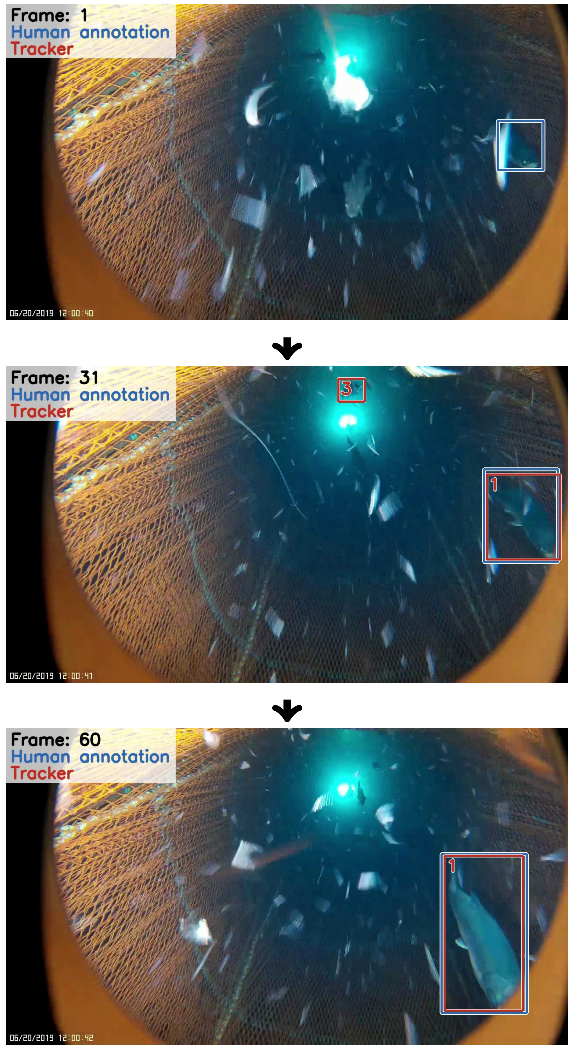

Figure 2: Visualized salmon tracking results for BoT-SORT with YOLO12x on a test dataset clip showing a salmon moving aft in a trawl net. Predicted tracks are plotted in red with their unique IDs. Human annotations are shown in blue. Frame one shows a false-negative track (missed salmon detection) and frame 31 shows a false-positive track (pollock incorrectly detected as a salmon) in the upper half of the frame. See the video for visualized results.

3.1 Object detection

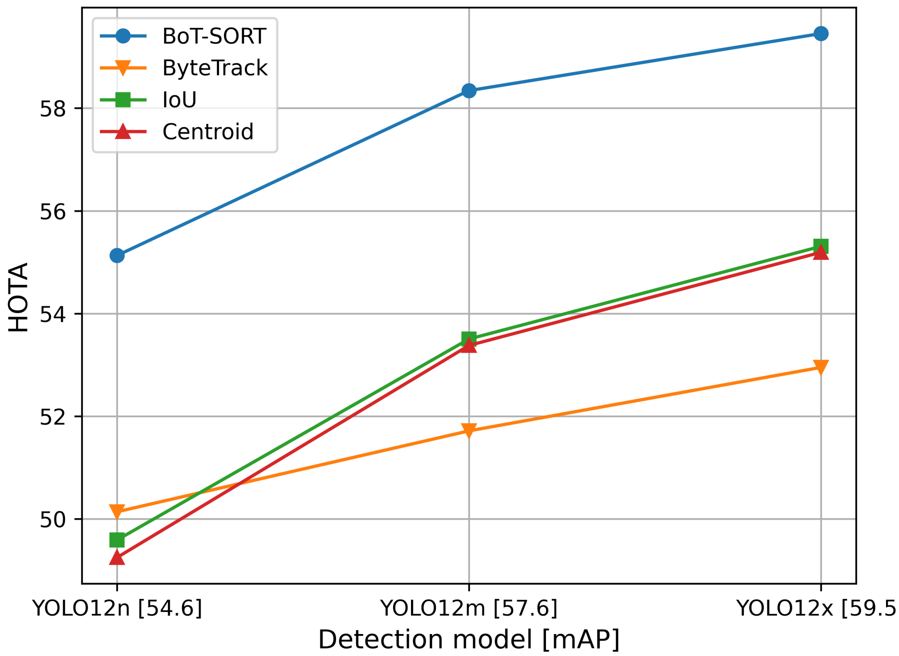

Figure 3: Tracker HOTA score versus detection model and mAP. YOLO models are ordered by size and mAP score on our dataset. Detection accuracy was a key factor influencing tracking performance, with more accurate models leading to better tracking results.

The YOLO12x model received the highest mAP score of 59.5. The smallest model in our evaluation, YOLO12n, had the lowest mAP score of 54.6. See Table 5 for complete detection model results. All trackers achieved their highest HOTA scores with YOLO12x (Figure 3).

Table 5: Object detection model results with mAP, AP, mAP50, precision (P), and recall (R) are shown. Best scores, based on the unrounded values, are in bold. All metrics are calculated using a model confidence score threshold of 0.001.

Model	mAP	mAP50	P	R	AP salmon	AP pollock	P salmon	P pollock	R salmon	R pollock
YOLO12n	54.6	84.0	82.3	75.5	56.3	53.0	78.9	**85.8**	74.3	76.7
YOLO12s	55.9	85.0	83.1	77.6	58.2	53.6	80.7	85.4	77.1	78.1
YOLO12m	57.6	85.7	83.5	79.4	60.3	54.9	82.8	84.3	77.2	81.5
YOLO12l	59.0	**87.2**	**84.8**	80.3	62.7	55.4	**84.4**	85.3	79.2	81.4
YOLO12x	**59.5**	87.2	83.8	**81.1**	**63.1**	**56.0**	83.3	84.4	**79.4**	**82.9**

3.2 Tracking

BoT-SORT had the highest HOTA score of 59.5, followed by IoU tracker (55.3), Centroid tracker (55.2), and ByteTrack (53.0) using SMAC3 optimized tracker parameters. See Table 6 for complete tracking results.

The manual tracking evaluation showed that BoT-SORT and ByteTrack were more robust to intermittent missed detections and created fewer fragmented tracks than IoU and Centroid trackers (video link), but were also more prone to combine false-positive and true-positive salmon detections into a single track (video link). Missed and false-positive detections contributed to most instances of localization and association error.

HOTA metrics corroborated the manual tracking observations. BoT-SORT's robustness to missed detections led to higher association recall than IoU: 70.0 and 54.5 respectively. BoT-SORT's tendency to combine tracks of different fish led to lower association precision than IoU: 75.3 and 89.7 respectively (Figure 4).

IoU and Centroid trackers had nearly identical localization accuracy scores, 87.4 for both with YOLO12x, because neither change the original detection bounding box during tracking and both use very similar track threshold parameters. BoT-SORT had much higher localization accuracy than ByteTrack: 85.7 and 80.3 respectively. BoT-SORT and ByteTrack both use Kalman filter-based bounding box prediction during tracking, but BoT-SORT uses a more accurate Kalman filter (Aharon et al., 2022).

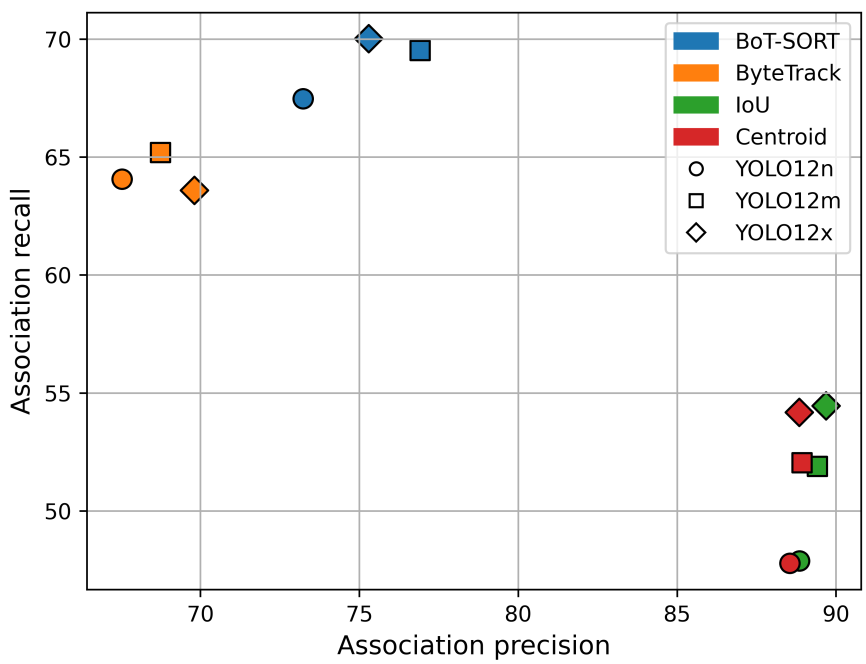

Figure 4: Association precision versus association recall of trackers. Association recall measures how well trackers avoid splitting the same object into multiple shorter tracks. Association precision measures how well tracks avoid merging multiple objects together into a single track.

Table 6: Salmon tracking results. HOTA, MOTA, IDF1, association accuracy (AssA), association recall (AssRe), association precision (AssPr), detection accuracy (DetA), and localization accuracy (LocA), and number of predicted track IDs are listed. The optimal value of metrics, high ↑ or low ↓, are indicated by arrows. Best scores, based on the unrounded values, are in bold. Ground truth annotations contain 51 IDs.

Tracker	HOTA ↑	MOTA ↑	IDF1 ↑	AssA ↑	AssRe ↑	AssPr ↑	DetA ↑	LocA ↑	IDs
IoU + YOLO12n	49.6	56.0	59.4	46.4	47.9	88.9	53.2	86.4	174
IoU + YOLO12m	53.5	61.8	64.0	50.2	51.9	89.4	57.3	87.2	164
IoU + YOLO12x	55.3	62.4	65.3	52.6	54.5	**89.7**	58.4	**87.4**	163
Centroid + YOLO12n	49.2	54.3	58.8	46.3	47.8	88.5	52.7	86.4	188
Centroid + YOLO12m	53.4	61.1	64.3	50.2	52.0	88.9	56.9	87.2	163
Centroid + YOLO12x	55.2	62.1	64.7	52.3	54.2	88.8	**58.5**	87.4	160
ByteTrack + YOLO12n	50.1	47.0	67.3	54.7	64.1	67.5	46.2	79.1	55
ByteTrack + YOLO12m	51.7	50.2	69.3	55.6	65.2	68.7	48.3	79.9	53
ByteTrack + YOLO12x	53.0	57.5	72.5	55.8	63.6	69.8	50.4	80.3	50
BoT-SORT + YOLO12n	55.1	58.2	71.6	57.8	67.5	73.2	52.7	84.9	52
BoT-SORT + YOLO12m	58.3	60.0	**75.6**	**62.0**	69.5	76.9	55.0	85.4	53
BoT-SORT + YOLO12x	**59.5**	**64.2**	74.5	61.4	**70.0**	75.3	57.7	85.7	51

We found a statistically significant positive relationship between the HOTA score of all trackers and the detection model's mAP for video clips; $p = 0.00$, $R^2 = 0.46$ for BoT-SORT (Figure 5). Detection model accuracy explained more variability in HOTA than the average number of fish per frame (salmon and pollock annotations) and was also significant; $p = 0.02$, $R^2 = 0.13$ for BoT-SORT. However, the relationship between fish per frame and HOTA was not significant for IoU and Centroid trackers (Figure 9). We did not find a significant relationship between HOTA and average salmon track length for any trackers.

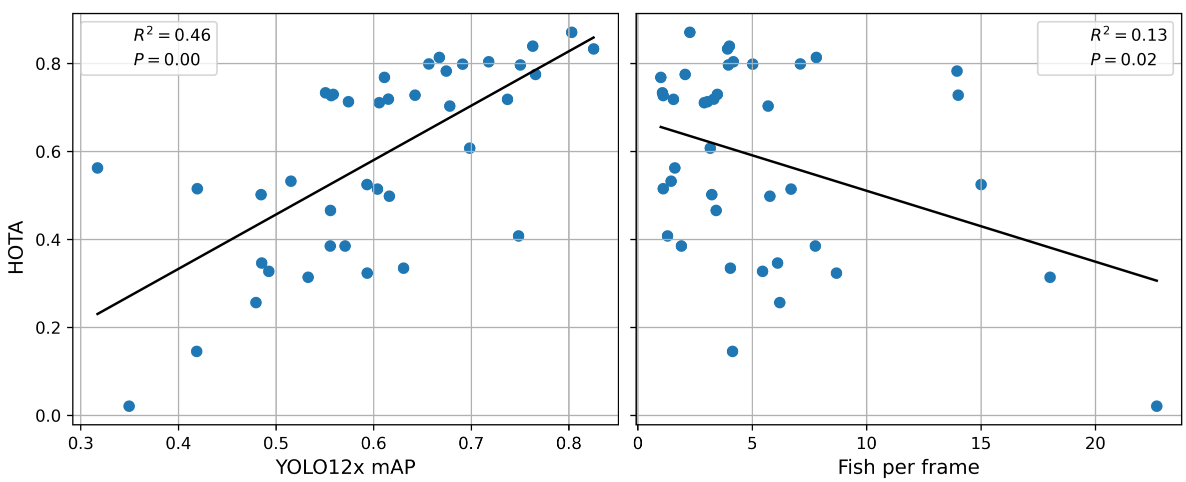

Figure 5: HOTA scores for BoT-SORT with YOLO12x detections versus mAP and average fish per frame (salmon and pollock annotations). Each point represents a clip in the test dataset. The line of best fit and R-squared were found using an ordinary least squares regression. P-values are based on a t-distribution with $\alpha = 0.05$. For graphs of all four trackers, see Figure 8 and Figure 9.

For all lower frame rates, the simpler IoU and Centroid trackers were the most accurate (Figure 6). At 10 FPS, IoU tracker had a HOTA of 58.6, compared to BoT-SORT's HOTA of 53.2. The association accuracy of IoU and Centroid trackers increased relative to BoT-SORT as the frame rate decreased. Detection and localization accuracy did not change significantly at lower frame rates for IoU and Centroid trackers, while they decreased for BoT-SORT. For full results, see Table 17.

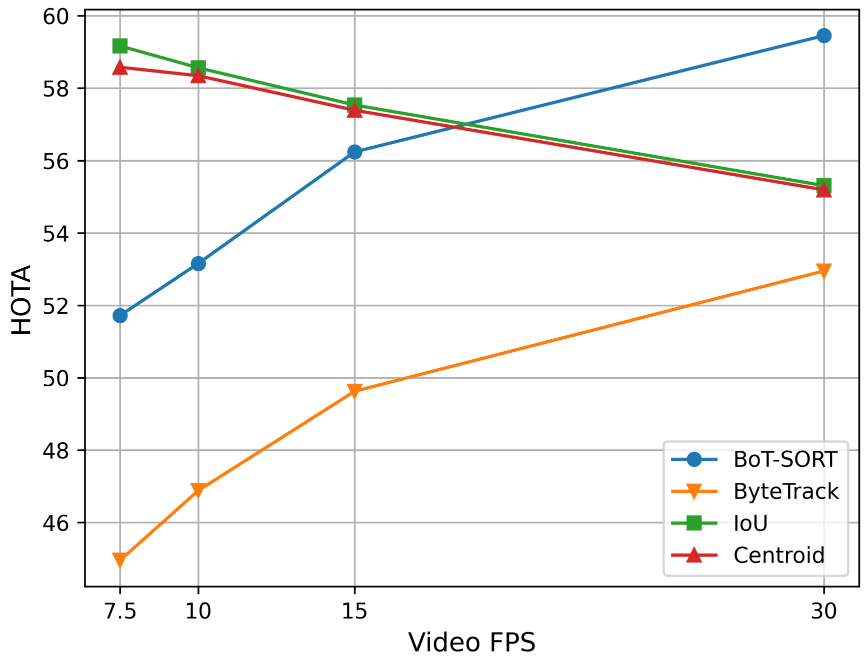

Figure 6: HOTA versus frame rate of tracking. All trackers used YOLO12x detections. HOTA scores should be compared at a single frame rate because the testing datasets differ across frame rates.

3.3 Optimization

Using object detection training data for tracker optimization using SMAC3 consistently identified strong combinations of tracking parameters that achieved near-optimal HOTA scores on our test dataset, while allowing us to efficiently and fairly compare tracker performance across detection models and frame rates.

Our optimized parameters performed worse than Ultralytics' default parameters for BoT-SORT and ByteTrack at 30 and 15 FPS, but better than defaults at 10 and 7.5 FPS (Table 7). BoT-SORT's HOTA score was 0.9 points lower with optimized parameters at 30 FPS and ByteTrack's HOTA score was identical with default and optimized parameters. HOTA scores were 2.7 and 3.5 points higher for BoT-SORT and ByteTrack respectively using optimized parameters at 7.5 FPS.

When using optimized parameters, BoT-SORT and ByteTrack were more prone to merging multiple salmon tracks into a single track, leading to fewer track IDs and lower association precision scores than with default parameters at 30 FPS (Table 12). The number of track IDs more than doubled for BoT-SORT and ByteTrack using default parameters. Association precision scores were 12.2 and 10.2 points lower for BoT-SORT and ByteTrack respectively with optimized parameters. This difference is explained by the optimized values for track buffer and match threshold, which were much higher than the defaults for BoT-SORT and ByteTrack with YOLO12x (Table 13, Table 14).

When optimized on the training dataset at 30 FPS, BoT-SORT received a HOTA of 59.5, only 2.2 points lower than when optimizing *and* evaluating BoT-SORT on the test dataset (Table 13). ByteTrack, IoU and Centroid trackers saw similar near-optimal results with optimized parameters (Table 14, Table 15, Table 16).

Table 7: Comparison of default and SMAC parameters for BoT-SORT and ByteTrack across different frame rates. The optimal value of metrics, high ↑ or low ↓, are indicated by arrows. Best scores, based on the unrounded values, are in bold. Lower frame rates were simulated by downsampling videos and annotations. All trackers used YOLO12x for detections. The number of IDs in ground truth annotations (GT IDs) are given for each frame rate. For detailed results see Table 18.

Frame rate	Tracker	Parameter	HOTA ↑	MOTA ↑	IDF1 ↑	IDs	GT IDs
30.0	BoT-SORT	Default	**60.4**	**65.3**	73.4	111	51
30.0	BoT-SORT	SMAC	59.5	64.2	**74.5**	51	51
30.0	ByteTrack	Default	53.0	57.9	69.4	125	51
30.0	ByteTrack	SMAC	53.0	57.5	72.5	50	51
15.0	BoT-SORT	Default	57.1	62.2	71.4	91	50
15.0	BoT-SORT	SMAC	56.2	59.2	72.6	47	50
15.0	ByteTrack	Default	50.3	51.5	67.9	89	50
15.0	ByteTrack	SMAC	49.6	47.6	66.9	47	50
10.0	BoT-SORT	Default	53.1	57.4	69.3	74	50
10.0	BoT-SORT	SMAC	53.2	53.8	70.5	46	50
10.0	ByteTrack	Default	45.2	47.6	62.5	81	50
10.0	ByteTrack	SMAC	46.9	40.0	63.6	45	50
7.5	BoT-SORT	Default	49.0	52.9	65.1	72	48
7.5	BoT-SORT	SMAC	51.7	50.9	69.9	45	48
7.5	ByteTrack	Default	41.5	40.3	55.8	77	48
7.5	ByteTrack	SMAC	45.0	36.8	61.9	42	48

4 Discussion

Developing effective techniques for automatically processing footage recorded inside commercial fishing gear is an important step towards building more sustainable fisheries. Multi-object tracking (MOT) has the potential to efficiently provide detailed information about fish behavior and gear interactions. Detailed data about fish behavior will enable the design of more selective fishing gear that excludes restricted species based on their behavioral differences from target species.

Despite these potential benefits, MOT has yet to be as widely applied in commercial fishing as it has been in fields like human behavior recognition and autonomous driving. The commercial fishing environment presents unique challenges for MOT that these domains lack. Poor image quality due to limited lighting and visibility underwater, coupled with the high cost of collecting video at sea, leads to relatively small and challenging datasets for object detection (Khokher et al., 2022). High fish densities in high-volume trawl fisheries create crowded scenes where identifying and tracking individuals is difficult even for human reviewers, as observed in our dataset.

In our work, we explored the capabilities and limitations of MOT in the commercial pollock trawl environment and found that the feasibility of tracking bycaught salmon largely depends on the accuracy of object detection. Tracking accuracy (HOTA) exceeded 80% when object detection accuracy (mAP) surpassed 80%, but fell as detection accuracy declined. Evaluating trackers at lower frame rates revealed that the most accurate tracker at higher frames rates (BoT-SORT) was outperformed by the simpler IoU and Centroid trackers at low frame rates. Our tracker optimization methods consistently identified near-optimal parameters, allowing us to fairly compare tracker performance across a range of frame rates and detection models in a unique fishing scenario.

Our results show that MOT is feasible in the commercial fishing domain but high fish densities, poor visibility, and occlusions still cause tracking errors. To improve the accuracy of tracking, we need larger and higher quality datasets, more accurate object detection models, and trackers that are robust to missing and false-positive detections. While our analysis was limited to four common tracking algorithms, our methods and findings for tracker performance and optimization in challenging scenarios and low-frame rates are broadly applicable to MOT in fisheries and beyond.

4.1 Data

Large, diverse, and representative datasets are the foundation of accurate deep learning models, and accurate deep learning models are the foundation of accurate tracking. Developing high-quality, domain-specific datasets remains the most important hurdle in the path towards more efficient and automated image-based data collection in the fisheries domain. In our study, an even larger and more diverse dataset would likely have improved the accuracy and the generalizability of our detection models and trackers.

In addition to larger datasets, characteristics of image data can influence detection performance. Issues common in commercial trawl fishing scenarios, such as turbidity, high fish density, and uneven lighting, make object detection particularly challenging (Wilson et al., 2025). The videos in our dataset reflect these challenges and also feature less than ideal camera angles, generally showing an anterior view of fish passing through the net (Figure 1). Adjusting the camera angle to capture a dorsal view could provide a stronger contrast between the silhouette of passing fish and the background, improving detection and tracking performance.

Annotation quality also plays a critical role in model development and tracker analysis. In our previous work, we reviewed our dataset to ensure consistency and accuracy across annotations, but ambiguity about when salmon are clearly visible can lead to discrepancies in the true start and end times of tracks. During tracker evaluation, we observed instances where trackers correctly tracked salmon beyond our annotations. As with all deep learning research, awareness of dataset strengths and limitations is important.

4.2 Object detection

Across all five model sizes, the largest model (YOLO12x) achieved the highest mAP score of 59.5, 4.9 points higher than our smallest model (YOLO12n). Detection accuracy remains the main bottleneck for tracking, as shown in our regression analysis (Figure 5). Missed and false positive detections propagate into detection, localization, and association errors during tracking. Improving dataset quality and refining camera placement and lighting would likely improve both detection and tracking accuracy.

MOT is prone to more kinds of errors, requires additional validation methods, and is generally a more complicated task than object detection. As such, it is important to first consider whether a task can be solved with object detection. Tracking enables the collection of more detailed temporal data, but it is challenging to implement and unnecessary for many common object detection-based tasks in the fisheries domain. Presence detection, species identification, and size estimation can be accomplished with object detection and generally do not require MOT.

4.3 Tracking

BoT-SORT consistently received the highest overall HOTA scores in our evaluations at 30 FPS (59.5 with YOLO12x), and the IoU and Centroid trackers performed well during low frame rate tracking.

While our results communicate tracking accuracy for a range of YOLO detection model sizes and tracking algorithms, they cannot tell us if a tracker and object detection model are accurate enough for a given application and frame rate. Like many machine learning tools, the acceptable accuracy threshold of a tracker varies. In the case of fish counting, tracking needs to be very accurate because segmented or missed tracks can significantly change totals. In other applications, like behavior identification, the accuracy threshold might be lower. For example, Abangan et al. (2024) achieved "reliable behavior quantification" about bait interactions of black seabream (*Spondyliosoma cantharus*) using a tracker that had a HOTA score of only 24.0 on their dataset.

Comparing tracker performance across domains and datasets is challenging, but we can offer some additional context about tracker performance on other datasets. BoT-SORT achieved HOTA scores of 64.6 and 62.6 on the widely used benchmark pedestrian tracking datasets MOT17 and MOT20 (Aharon et al., 2022; Dendorfer et al., 2020; Milan et al., 2016). BoT-SORT achieved a top score of 59.5 on our dataset, indicating a remaining performance gap between the well-studied task of pedestrian tracking and the novel task of tracking salmon in trawl nets.

The IoU and Centroid trackers achieved HOTA scores of 58.6 and 58.3 respectively on our test dataset when evaluated at 10 FPS, showing that low frame rate tracking is possible in commercial trawl nets. This finding is particularly relevant in the fisheries domain, where computational resources are limited. Our results suggest that at low frame rates, using simpler trackers that require less computational resources will maximize both efficiency and accuracy. Note that we cannot fairly compare HOTA scores across frame rates. At lower frame rates a tracker uses fewer detections and is evaluated against a smaller subset of annotations.

Further research is needed to explore the relationships between fish speed, tracking performance, and frame rate. Frame rate, the speed of objects tracked, and the granularity of tracking data all influence tracking performance. Ultimately, reliable tracking requires objects of interest to be present in a video for a sufficient period of time. The optimal frame rate for efficient and accurate tracking of animals in the commercial fishing domain will vary based on factors such as swimming speed, vessel speed, and current. For example, capturing salmon behavior in pollock trawls may require a higher frame rate than crab behavior in stationary pots.

We prioritized the use of well-supported, open-source detection and tracking libraries to improve the accessibility and usefulness of our work. However, relying on Ultralytics limited the detection models and trackers available for analysis. We did not use re-identification (Re-ID) for BoT-SORT and ByteTrack because the feature was not implemented in the Ultralytics package at the time of our analysis. Using Re-ID could potentially improve tracking performance in the commercial fishing domain and warrants further study.

Given the prevalence of object detection errors observed in our study, building trackers that are robust to missed and false detections is an important area for future work. BoT-SORT and ByteTrack use Kalman filters to predict the location and size of bounding boxes, potentially reducing noise and making them more resilient to missed and false detections. While BoT-SORT achieved the top tracking accuracy at 30 frames per second, both Kalman filter-based trackers performed worse than the simpler IoU and Centroid trackers at low frames rates. At lower frame rates, Kalman filters have less data to estimate track trajectories and objects move more between each frame. We used Ultralytics' default parameters for Kalman filters during tracking. Adjusting the Kalman filter parameters could improve performance at lower frames rates and warrants further study.

Additional work is needed to develop trackers that are robust in scenarios with sparse and unreliable data, and to evaluate whether Re-ID and optimized Kalman filter parameters can improve tracking performance in challenging commercial fishing environments.

4.4 Optimization

Our parameter optimization methods consistently identified strong combinations of tracking parameters that achieved near-optimal HOTA scores on our test dataset. Using systematic, automated optimization methods like ours provides several important benefits over manual parameter selection. Our method saves time, removes the bias of manual testing, and makes it easier to optimize tracking performance for specific domains.

Our optimized parameters performed slightly worse than Ultralytics' default parameters for BoT-SORT and ByteTrack at higher frame rates (30 and 15 FPS), but much better than defaults at lower frame rates (10 and 7.5 FPS).

Match threshold and track buffer were the parameters with the most striking differences between the optimized and default parameters for BoT-SORT and ByteTrack. Both were much higher in our optimized parameters at 30 FPS. A high match threshold allows tracks with low or no IoU to be matched. A high track buffer allows new detections to be matched with tracks that have been ”lost” or unmatched for long periods of time. These parameters made our optimized BoT-SORT and ByteTrack trackers more prone to merging multiple salmon tracks into a single track. This reduced track IDs, but also caused tracking errors.

The performance of our optimized parameters shows that our optimization method is effective at adapting to non-standard domains (low FPS videos) and achieves similar performance to parameters optimized across much larger datasets for applications like pedestrian and vehicle tracking (Zhang et al., 2022). Using domain knowledge to impose tighter bounds on parameter values during optimization and using larger datasets for optimization would likely improve optimization results.

Despite its benefits, systematic tracker optimization is an easily overlooked or infeasible step in MOT. Directly applying MOT to new data is a much simpler task than optimizing or evaluating tracking performance on a new dataset. A tracker only needs an object detection model. Optimizing and evaluating a tracker's performance requires a video-based dataset with annotated tracks, optimization methods, and customized evaluation packages.

Access to sufficient training data is a common constraint in all areas of machine learning. In an effort to optimize tracking performance and maximize the amount of data available for detection model training and tracker evaluation, we used object detection model training data for tracker optimization. While our results showed that this method still achieved near-optimal tracking results, with BoT-SORT receiving a HOTA of 59.5 using YOLO12x, only 2.2 points lower than when optimizing *and* evaluating BoT-SORT on the test dataset, this technique warrants further study. Model accuracy and detection confidence influence tracker performance and the accuracy and confidence scores of detection models were higher on the optimization dataset because the models had seen the data during training.

4.5 Future directions

In our work, we show the viability and potential for using MOT to track and collect accurate movement and behavior data from commercial trawl fishing footage. Our findings provide valuable contributions to the MOT field, showing that at low frame rates, automated tracker optimization and simple tracking algorithms can significantly improve tracking accuracy over default tracker parameters and Kalman-filter based trackers.

The large volumes of quantitative data about fish movement that MOT automatically collects with minimal human annotation can be applied to inform the design of more selective fishing gear and enable real-time monitoring during fishing. But developing accurate tracking methods for the commercial fishing domain is only an initial step towards potential applications. More work is needed to develop standardized and efficient methods for creating and using MOT-generated data, including more accurate tracking algorithms, more marine datasets, and accessible tools for using MOT data in fisheries research.

5 Acknowledgments

Special thanks to Sandy Parker-Stetter, Rick Towler, and Lukas DeFilippo for their valuable feedback on the manuscript, and to the AFSC/RACE/MACE team for their dedication to scientific discovery.

The data used in this work was collected with support from the 2019 and 2020 FV *Pacific Explorer* captain and crew, David Bryan, Matthew Callahan, Connor Fox, Katherine Hellen-Schneider, and Deborah Sharpe, and we are grateful for their help.

The scientific results and conclusions, as well as any views or opinions expressed herein, are those of the authors and do not necessarily reflect those of NOAA or the U.S. Department of Commerce. Reference to trade names does not imply endorsement by the National Marine Fisheries Service, NOAA.

6 Author contributions

Moses Lurbur: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, Visualization. Katherine C. Wilson: Conceptualization, Validation, Resources, Data Curation, Writing - Review & Editing, Supervision, Funding Acquisition, Project administration. Noëlle Yochum: Conceptualization, Resources, Data Curation, Writing - Review & Editing, Supervision, Funding Acquisition

7 Data availability statement

Our code, model weights, and dataset are publicly available and can be found at github.com/noaa-afsc-mace/salmon_tracking. Additional information about our dataset can also be found on our NOAA InPort data entry.

8 Funding

This work was supported by the National Cooperative Research Program (2019, 2020, 2021), which provided funding for data collection and part of ML's work on the project. The NOAA High Performance Computing and Communications Incubator program (2023) also provided funding for part of ML's work on the project. NOAA's Alaska Fisheries Science Center (AFSC) supported KW's work on the project. AFSC and Trident Seafoods supported NY's work on the project.

9 Conflict of interest statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

10 CO2 emission estimate of computational experiments

Experiments were conducted using Google Cloud Platform. Approximately 100 hours of computation was performed using NVIDIA T4 GPUs. Total emissions are estimated to be 2.1 kgCO_2eq of which 100% was directly offset by the cloud provider.

Estimations were conducted using the machine learning emissions calculator presented in Lacoste et al. (2019).

11 LLM usage disclosure

An LLM did not write this paper. All ideas, writing, and data are those of the authors. Code-assist tools were used to accelerate development and ensure clean, accurate, and well tested code for our experiments.

12 Appendix

Table 8: 15.0 FPS tracking dataset properties. Training data were used to optimize tracker parameters. Testing data were used for tracker evaluation. The test and train sets are identical to the object detection dataset, except clips without salmon tracks were removed.

Dataset split	Salmon tracks	Frames	Video clips
Train	165 (77%)	6,397 (79%)	128 (76%)
Test	50 (23%)	1,665 (21%)	40 (24%)
Total	215	8,062	168

Table 9: 10.0 FPS tracking dataset properties. Training data were used to optimize tracker parameters. Testing data were used for tracker evaluation. The test and train sets are identical to the object detection dataset, except clips without salmon tracks were removed.

Dataset split	Salmon tracks	Frames	Video clips
Train	165 (77%)	4,280 (79%)	128 (76%)
Test	50 (23%)	1,114 (21%)	40 (24%)
Total	215	5,394	168

Table 10: 7.5 FPS tracking dataset properties. Training data were used to optimize tracker parameters. Testing data were used for tracker evaluation. The test and train sets are identical to the object detection dataset, except clips without salmon tracks were removed.

Dataset split	Salmon tracks	Frames	Video clips
Train	163 (77%)	3,201 (79%)	127 (76%)
Test	48 (23%)	838 (21%)	40 (24%)
Total	211	4,039	167

Table 11: Search bounds used for tracker parameter optimization. Cells are left blank if the parameter is not used by the tracker.

Tracker	Match threshold	Track threshold	Track buffer	New track scaling factor	Track high scaling factor	Track low threshold
BoT-SORT	(0.0001, 10.0)	—	(1, 60)	(0.0, 1.0)	(0.0, 1.0)	(0.0001, 1.0)
ByteTrack	(0.0001, 10.0)	—	(1, 60)	(0.0, 1.0)	(0.0, 1.0)	(0.0001, 1.0)
IoU	(0.0001, 1.0)	(0.0001, 1.0)	(1, 60)	—	—	—
Centroid	(0.0001, 10.0)	(0.0001, 1.0)	(1, 60)	—	—	—

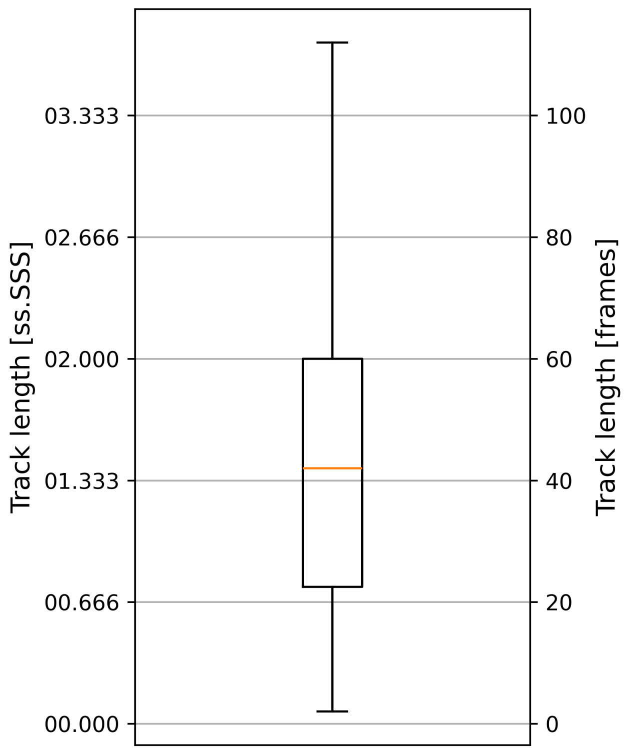

Figure 7: Box plot showing the length of all tracks in our dataset. The median track length is 1.4 seconds, or 42 frames. The average track length is 1.8 seconds, or 52.9 frames.

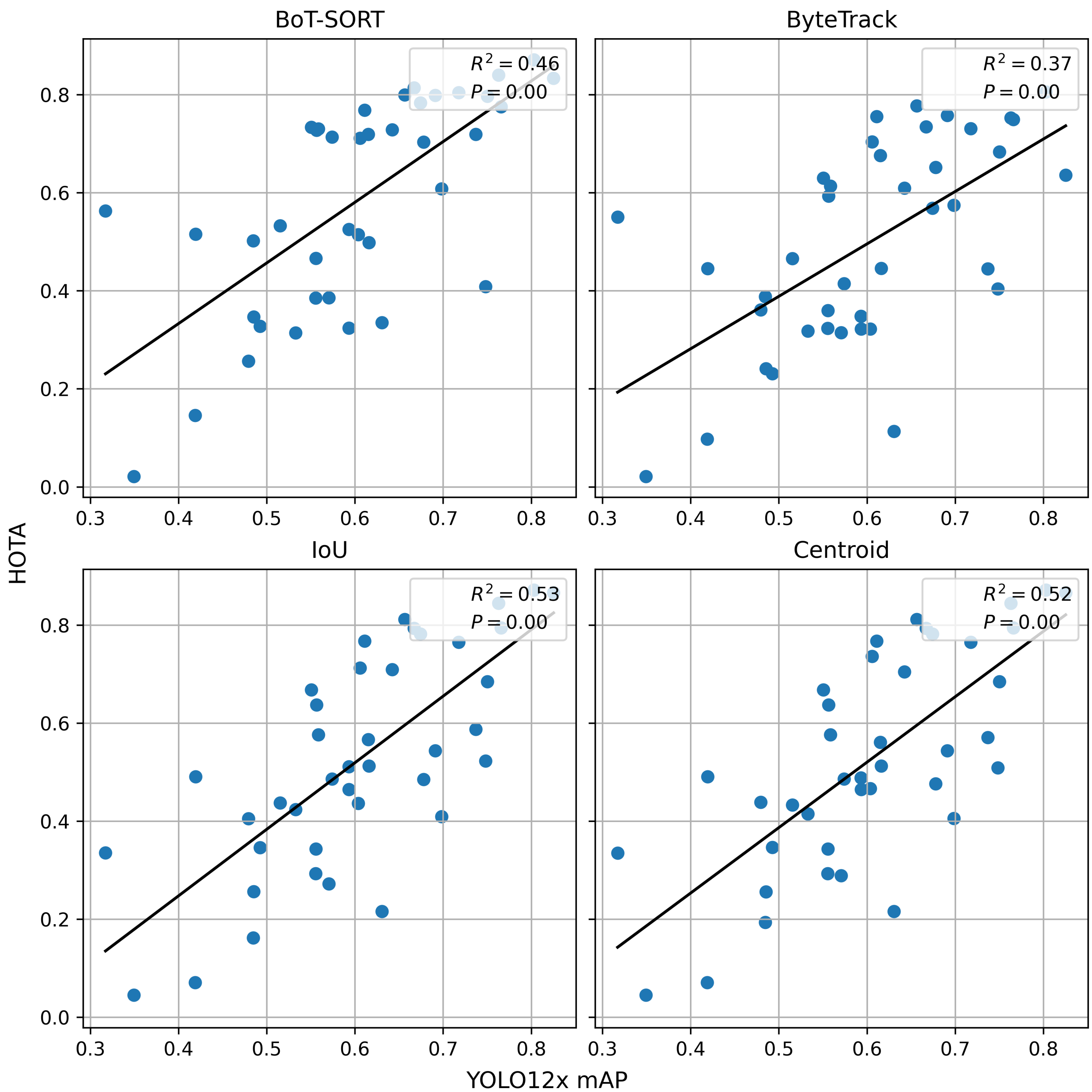

Figure 8: HOTA scores versus mAP for each tracker. All trackers used YOLO12x model detections. Each point represents a clip in the tracker evaluation dataset. The line of best fit and R-squared were found using an ordinary least squares regression. P-values are based on a t-distribution with $\alpha = 0.05$.

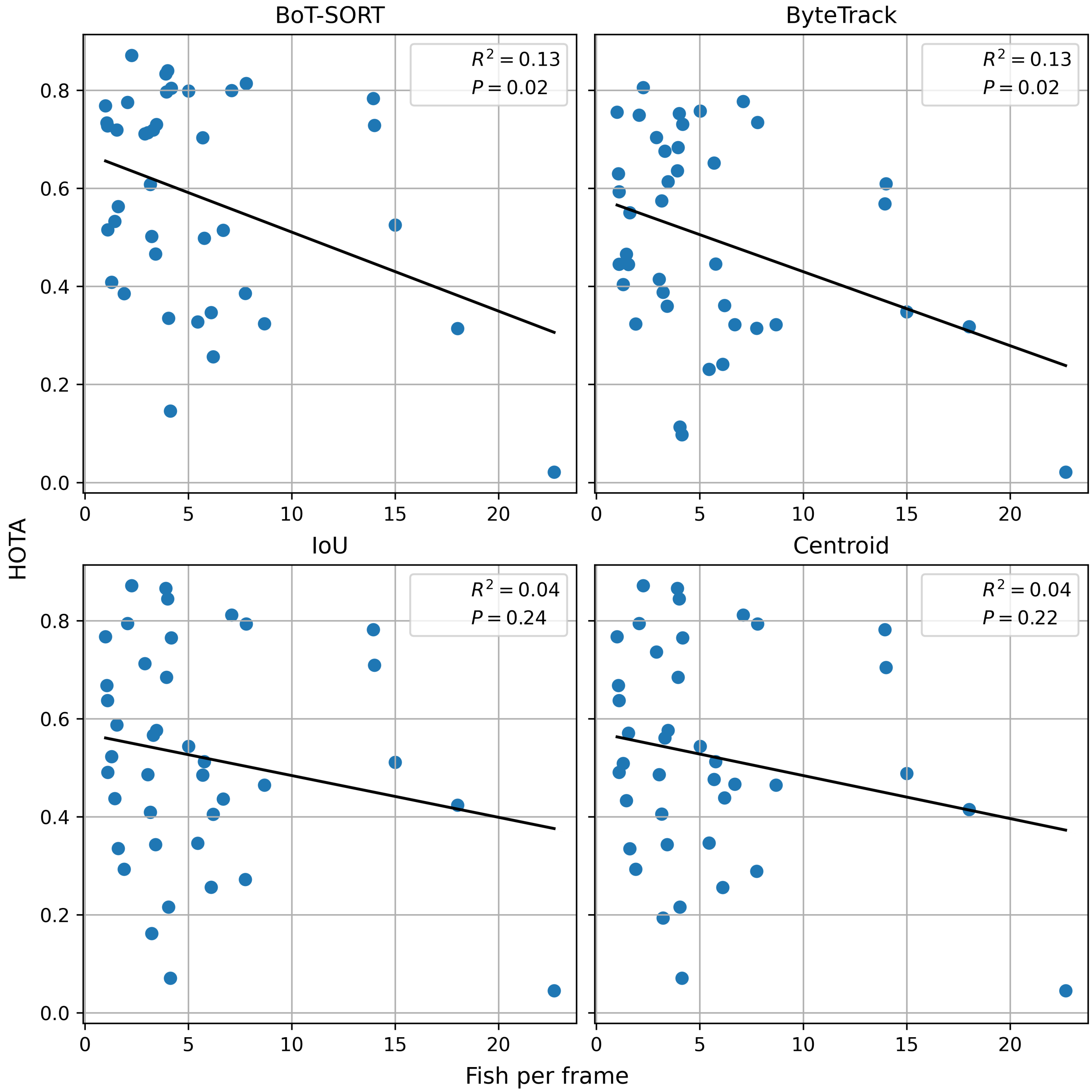

Figure 9: HOTA scores versus the average number of fish (salmon and pollock annotations) per frame. All trackers used YOLO12x model detections. Each point represents a clip in the tracker evaluation dataset. The line of best fit and R-squared were found using an ordinary least squares regression. P-values are based on a t-distribution with $\alpha = 0.05$.

Table 12: Complete tracking results for BoT-SORT and ByteTrack using default parameters and SMAC optimized parameters. HOTA, MOTA, IDF1, association accuracy (AssA), association recall (AssRe), association precision (AssPr), detection accuracy (DetA), and localization accuracy (LocA), and number of predicted track IDs are listed. The optimal value of metrics, high ↑ or low ↓, are indicated by arrows. Best scores, based on the unrounded values, are in bold. Ground truth annotations contain 51 IDs.

Tracker	HOTA ↑	MOTA ↑	IDF1 ↑	AssA ↑	AssRe ↑	AssPr ↑	DetA ↑	LocA ↑	IDs
ByteTrack (default) + YOLO12n	49.5	47.9	65.2	53.4	56.2	78.7	46.1	79.5	133
ByteTrack (default) + YOLO12m	52.5	56.1	69.1	55.1	58.0	79.9	50.1	80.2	133
ByteTrack (default) + YOLO12x	53.0	57.9	69.4	55.1	57.9	80.0	51.2	80.4	125
ByteTrack (SMAC) + YOLO12n	50.1	47.0	67.3	54.7	64.1	67.5	46.2	79.1	55
ByteTrack (SMAC) + YOLO12m	51.7	50.2	69.3	55.6	65.2	68.7	48.3	79.9	53
ByteTrack (SMAC) + YOLO12x	53.0	57.5	72.5	55.8	63.6	69.8	50.4	80.3	50
BoT-SORT (default) + YOLO12n	55.5	56.7	68.1	57.9	60.1	86.2	53.3	85.4	126
BoT-SORT (default) + YOLO12m	59.0	64.2	72.2	60.2	62.4	87.2	57.8	85.9	122
BoT-SORT (default) + YOLO12x	**60.4**	**65.3**	73.4	61.9	64.2	**87.5**	**59.0**	**86.3**	111
BoT-SORT (SMAC) + YOLO12n	55.1	58.2	71.6	57.8	67.5	73.2	52.7	84.9	52
BoT-SORT (SMAC) + YOLO12m	58.3	60.0	**75.6**	**62.0**	69.5	76.9	55.0	85.4	53
BoT-SORT (SMAC) + YOLO12x	59.5	64.2	74.5	61.4	**70.0**	75.3	57.7	85.7	51

Table 13: HOTA scores and default and optimized tracker parameters for the BoT-SORT tracker. Confidence and matching thresholds are rounded for readability. The last row shows the "best case" parameters and HOTA score found by optimizing *and* evaluating the tracker directly on the evaluation dataset. The best HOTA score, based on the unrounded HOTA value, is in bold.

Tracker	Parameter source	HOTA ↑	Fuse score	Match threshold	New track threshold	Track buffer	Track high threshold	Track low threshold
BoT-SORT + YOLO12n	Default	55.5	True	0.80	0.25	30	0.25	0.10
BoT-SORT + YOLO12m	Default	59.0	True	0.80	0.25	30	0.25	0.10
BoT-SORT + YOLO12x	Default	60.4	True	0.80	0.25	30	0.25	0.10
BoT-SORT + YOLO12n	SMAC	55.1	False	3.66	0.28	60	0.50	0.28
BoT-SORT + YOLO12m	SMAC	58.3	False	8.38	0.63	60	0.36	0.36
BoT-SORT + YOLO12x	SMAC	59.5	False	5.40	0.51	60	0.51	0.20
BoT-SORT + YOLO12x (optimized on test)	SMAC	**61.7**	False	4.05	0.48	13	0.07	0.07

Table 14: HOTA scores and default and optimized tracker parameters for the ByteTrack tracker. Confidence and matching thresholds are rounded for readability. The last row shows the ”best case” parameters and HOTA score found by optimizing *and* evaluating the tracker directly on the evaluation dataset. The best HOTA score, based on the unrounded HOTA value, is in bold.

Tracker	Parameter source	HOTA ↑	Fuse score	Match threshold	New track threshold	Track buffer	Track high threshold	Track low threshold
ByteTrack + YOLO12n	Default	49.5	True	0.80	0.25	30	0.25	0.10
ByteTrack + YOLO12m	Default	52.5	True	0.80	0.25	30	0.25	0.10
ByteTrack + YOLO12x	Default	53.0	True	0.80	0.25	30	0.25	0.10
ByteTrack + YOLO12n	SMAC	50.1	True	5.95	0.57	31	0.45	0.05
ByteTrack + YOLO12m	SMAC	51.7	False	9.07	0.55	46	0.44	0.16
ByteTrack + YOLO12x	SMAC	53.0	False	10.00	0.65	48	0.54	0.36
ByteTrack + YOLO12x (optimized on test)	SMAC	**55.4**	False	2.76	0.56	10	0.13	0.00

Table 15: HOTA scores and tracker parameters from our optimization method for the IoU tracker. Confidence and matching thresholds are rounded for readability. The last row shows the ”best case” parameters and HOTA score found by optimizing *and* evaluating the tracker directly on the evaluation dataset. The best HOTA score, based on the unrounded HOTA value, is in bold.

Tracker	HOTA ↑	Match threshold	Track buffer	Track threshold
IoU + YOLO12n	49.6	0.81	16	0.39
IoU + YOLO12m	53.5	1.00	60	0.39
IoU + YOLO12x	55.3	1.00	36	0.35
IoU + YOLO12x (optimized on test)	**55.7**	0.89	11	0.23

Table 16: HOTA scores and tracker parameters from our optimization method for the Centroid tracker. Confidence and matching thresholds are rounded for readability. The last row shows the ”best case” parameters and HOTA score found by optimizing *and* evaluating the tracker directly on the evaluation dataset. The best HOTA score, based on the unrounded HOTA value, is in bold.

Tracker	HOTA ↑	Match threshold	Track buffer	Track threshold
Centroid + YOLO12n	49.2	6.73	24	0.36
Centroid + YOLO12m	53.4	1.70	14	0.40
Centroid + YOLO12x	55.2	6.71	60	0.32
Centroid + YOLO12x (optimized on test)	**56.0**	2.04	1	0.22

Table 17: Salmon tracking results for different frame rates. HOTA, MOTA, IDF1, association accuracy (AssA), association recall (AssRe), association precision (AssPr), detection accuracy (DetA), and localization accuracy (LocA), number of predicted track IDs, and number of IDs in ground truth annotations (GT IDs) are listed. The optimal value of metrics, high ↑ or low ↓, are indicated by arrows. Best scores, based on the unrounded values, are in bold. Lower frame rates were simulated by downsampling videos and annotations. All trackers used YOLO12x for detections.

Frame rate	Tracker	HOTA ↑	MOTA ↑	IDF1 ↑	AssA ↑	AssRe ↑	AssPr ↑	DetA ↑	LocA ↑	IDs	GT IDs
30.0	IoU	55.3	62.4	65.3	52.6	54.5	89.7	58.4	87.4	163	51
30.0	Centroid	55.2	62.1	64.7	52.3	54.2	88.8	58.5	87.4	160	51
30.0	ByteTrack	53.0	57.5	72.5	55.8	63.6	69.8	50.4	80.3	50	51
30.0	BoT-SORT	**59.5**	**64.2**	**74.5**	**61.4**	**70.0**	75.3	57.7	85.7	51	51
15.0	IoU	57.5	62.9	68.8	56.5	58.5	89.6	58.7	87.4	114	50
15.0	Centroid	57.4	62.4	68.6	56.4	58.7	88.9	58.6	87.3	109	50
15.0	ByteTrack	49.6	47.6	66.9	53.6	63.0	67.1	46.2	77.8	47	50
15.0	BoT-SORT	56.2	59.2	72.6	59.6	68.7	74.0	53.2	83.3	47	50
10.0	IoU	58.6	63.0	70.3	58.3	60.4	**89.8**	**59.1**	**87.5**	92	50
10.0	Centroid	58.3	62.0	69.9	58.2	60.6	88.5	58.7	87.4	89	50
10.0	ByteTrack	46.9	40.0	63.6	51.4	60.7	65.5	43.1	76.6	45	50
10.0	BoT-SORT	53.2	53.8	70.5	56.9	64.8	72.4	49.9	81.9	46	50
7.5	IoU	59.2	62.7	71.1	59.9	62.0	89.7	58.6	87.4	83	48
7.5	Centroid	58.6	60.7	70.5	59.8	62.2	88.3	57.6	87.3	79	48
7.5	ByteTrack	45.0	36.8	61.9	50.3	60.2	64.0	40.5	75.9	42	48
7.5	BoT-SORT	51.7	50.9	69.9	56.2	64.1	72.1	47.8	81.3	45	48

Table 18: Comparison of default and SMAC parameters for BoT-SORT and ByteTrack across different frame rates. The optimal value of metrics, high ↑ or low ↓, are indicated by arrows. Best scores, based on the unrounded values, are in bold. Lower frame rates were simulated by downsampling videos and annotations. All trackers used YOLO12x for detections. The number of IDs in ground truth annotations (GT IDs) are given for each frame rate.

Frame rate	Tracker	Parameter	HOTA ↑	MOTA ↑	IDF1 ↑	AssA ↑	AssRe ↑	AssPr ↑	DetA ↑	LocA ↑	IDs	GT IDs
30.0	BoT-SORT	Default	**60.4**	**65.3**	73.4	**61.9**	64.2	**87.5**	**59.0**	**86.3**	111	51
30.0	BoT-SORT	SMAC	59.5	64.2	**74.5**	61.4	**70.0**	75.3	57.7	85.7	51	51
30.0	ByteTrack	Default	53.0	57.9	69.4	55.1	57.9	80.0	51.2	80.4	125	51
30.0	ByteTrack	SMAC	53.0	57.5	72.5	55.8	63.6	69.8	50.4	80.3	50	51
15.0	BoT-SORT	Default	57.1	62.2	71.4	59.0	61.1	86.5	55.5	84.8	91	50
15.0	BoT-SORT	SMAC	56.2	59.2	72.6	59.6	68.7	74.0	53.2	83.3	47	50
15.0	ByteTrack	Default	50.3	51.5	67.9	54.2	57.3	78.1	47.0	78.7	89	50
15.0	ByteTrack	SMAC	49.6	47.6	66.9	53.6	63.0	67.1	46.2	77.8	47	50
10.0	BoT-SORT	Default	53.1	57.4	69.3	55.8	57.8	85.6	50.8	83.8	74	50
10.0	BoT-SORT	SMAC	53.2	53.8	70.5	56.9	64.8	72.4	49.9	81.9	46	50
10.0	ByteTrack	Default	45.2	47.6	62.5	47.2	50.0	78.8	43.7	78.3	81	50
10.0	ByteTrack	SMAC	46.9	40.0	63.6	51.4	60.7	65.5	43.1	76.6	45	50
7.5	BoT-SORT	Default	49.0	52.9	65.1	51.5	53.4	86.3	46.9	83.5	72	48
7.5	BoT-SORT	SMAC	51.7	50.9	69.9	56.2	64.1	72.1	47.8	81.3	45	48
7.5	ByteTrack	Default	41.5	40.3	55.8	44.5	47.1	81.1	39.0	78.2	77	48
7.5	ByteTrack	SMAC	45.0	36.8	61.9	50.3	60.2	64.0	40.5	75.9	42	48

References

Alexa Sugpatan Abangan, Kilian Bürgi, Sonia Méhault, Morgan Deroiné, Dorothée Kopp, and Robin Faillettaz. Assessment of sustainable baits for passive fishing gears through automatic fish behavior recognition. *Sci Rep*, 14(1):13110, June 2024. ISSN 2045-2322. doi:10.1038/s41598-024-63929-5.

Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv*, June 2022. doi:10.48550/arXiv.2206.14651.

William I. Atlas, Sami Ma, Yi Ching Chou, Katrina Connors, Daniel Scurfield, Brandon Nam, Xiaoqiang Ma, Mark Cleveland, Janvier Doire, Jonathan W. Moore, Ryan Shea, and Jiangchuan Liu. Wild salmon enumeration and monitoring using deep learning empowered detection and tracking. *Front. Mar. Sci.*, 10, September 2023. ISSN 2296-7745. doi:10.3389/fmars.2023.1200408.

Lee R. Benaka, Daryl Bullock, Aimee L. Hoover, and Noelle A. Olsen. U.S. national bycatch report first edition update 3. *NOAA Technical Memorandum*, 2019.

Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008. ISSN 1687-5176, 1687-5281. doi:10.1155/2008/246309.

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, September 2016. doi:10.1109/ICIP.2016.7533003.

David R. Bryan, Noëlle Yochum, and Katherine C. Wilson. Evaluating Pacific salmon swimming behavior in the aft end of a pelagic trawl to inform bycatch reduction device design and use. *Marine and Coastal Fisheries*, 16:e10306, August 2024. doi:10.1002/mcf2.10306.

Levi Cai, Nathan E. McGuire, Roger Hanlon, T. Aran Mooney, and Yogesh Girdhar. Semi-supervised visual tracking of marine animals using autonomous underwater vehicles. *Int J Comput Vis*, 131(6):1406–1427, June 2023. ISSN 0920-5691, 1573-1405. doi:10.1007/s11263-023-01762-5.

T. M. Cox, R. L. Lewison, R. Žydelis, L. B. Crowder, C. Safina, and A. J. Read. Comparing effectiveness of experimental and implemented bycatch reduction measures: The ideal and the real. *Conservation Biology*, 21(5):1155–1164, 2007. ISSN 1523-1739. doi:10.1111/j.1523-1739.2007.00772.x.

Meng Cui, Xubo Liu, Haohe Liu, Jinzheng Zhao, Daoliang Li, and Wenwu Wang. Fish tracking, counting, and behaviour analysis in digital aquaculture: a comprehensive survey. *Reviews in Aquaculture*, 17(1): e13001, 2025. doi:10.1111/raq.13001.

Matthew Dawkins, Jack Prior, Bryon Lewis, Robin Faillettaz, Thompson Banez, Mary Salvi, Audrey Rollo, Julien Simon, Matthew Campbell, Matthew Lucero, Aashish Chaudhary, Benjamin Richards, and Anthony Hoogs. FishTrack23: An ensemble underwater dataset for multi-object tracking. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7152–7161, January 2024. doi:10.1109/WACV57701.2024.00701.

Gregory R. DeCelles, Emily F. Keiley, Travis M. Lowery, Nicholas M. Calabrese, and Kevin D. E. Stokesbury. Development of a video trawl survey system for New England groundfish. *Transactions of the American Fisheries Society*, 146:462–477, May 2017. doi:10.1080/00028487.2017.1282888.

Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixe. CVPR19 tracking and detection challenge: How crowded can it get? *arXiv*, June 2019. doi:10.48550/arXiv.1906.04567.

Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv*, March 2020. doi:10.48550/arXiv.2003.09003.

NOAA Fisheries. Alaska Pollock | NOAA Fisheries. https://www.fisheries.noaa.gov/species/alaska-pollock, 2025. (accessed 6 March 2025).

James Ianelli, Ben Fissel, Sarah Stienessen, Taina Honkalehto, Elizabeth Siddon, and Caitlin Allen-Akselrud. Chapter 1: Assessment of the Walleye Pollock Stock in the Eastern Bering Sea. *NPFMC SAFE Report*, December 2021.

Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO (Version 8.0) [software]. https://github.com/ultralytics/ultralytics, 2025.

Kakani Katija, Paul L. D. Roberts, Joost Daniels, Alexandra Lapides, Kevin Barnard, Mike Risi, Ben Y Ranaan, Benjamin G Woodward, and Jonathan Takahashi. Visual tracking of deepwater animals using machine learning-controlled robotic underwater vehicles. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 859–868, January 2021. doi:10.1109/WACV48630.2021.00090.

Kakani Katija, Eric Orenstein, Brian Schlining, Lonny Lundsten, Kevin Barnard, Giovanna Sainz, Oceane Boulais, Megan Cromwell, Erin Butler, Benjamin Woodward, and Katherine L. C. Bell. FathomNet: A global image database for enabling artificial intelligence in the ocean. *Sci Rep*, 12(1):15914, September 2022. ISSN 2045-2322. doi:10.1038/s41598-022-19939-2.

M. Rizwan Khokher, L. Richard Little, Geoffrey N. Tuck, Daniel V. Smith, Maoying Qiao, Carlie Devine, Helen O'Neill, John J. Pogonoski, Rhys Arangio, and Dadong Wang. Early lessons in deploying cameras and artificial intelligence technology for fisheries catch monitoring: Where machine learning meets commercial fishing. *Canadian Journal of Fisheries and Aquatic Sciences*, 79:257–266, February 2022. doi:10.1139/cjfas-2020-0446.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 (1-2):83–97, 1955. ISSN 1931-9193. doi:10.1002/nav.3800020109.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022. ISSN 1533-7928. doi:10.48550/arXiv.2109.09831.

Jonathan Luiten. TrackEval [software]. https://github.com/JonathonLuiten/TrackEval, March 2025.

Jonathon Luiten, Aljoša Ošep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int J Comput Vis*, 129(2): 548–578, February 2021. ISSN 1573-1405. doi:10.1007/s11263-020-01375-2.

Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448, April 2021. ISSN 0004-3702. doi:10.1016/j.artint.2020.103448.

Francisco Madrigal, Camille Maurice, and Frédéric Lerasle. Hyper-parameter optimization tools comparison for multiple object tracking applications. *Machine Vision and Applications*, 30(2):269–289, March 2019. ISSN 1432-1769. doi:10.1007/s00138-018-0984-1.

Travis Mandel, Mark Jimenez, Emily Risley, Taishi Nammoto, Rebekka Williams, Max Panoff, Meynard Ballesteros, and Bobbie Suarez. Detection confidence driven multi-object tracking to recover reliable tracks from unreliable detections. *Pattern Recognition*, 135:109107, March 2023. ISSN 0031-3203. doi:10.1016/j.patcog.2022.109107.

Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. *arXiv*, March 2016. doi:10.48550/arXiv.1603.00831.

Malte Pedersen, Daniel Lehotský, Ivan Nikolov, and Thomas B. Moeslund. BrackishMOT: The brackish multi-object tracking dataset. In Rikke Gade, Michael Felsberg, and Joni-Kristian Kämäräinen, editors, *Image Analysis*, pages 17–33, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-31435-3. doi:10.1007/978-3-031-31435-3_2.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *arXiv*, May 2016. doi:10.48550/arXiv.1506.02640.

Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 17–35, Cham, 2016. Springer International Publishing. ISBN 978-3-319-48881-3. doi:10.1007/978-3-319-48881-3_2.

Craig S. Rose and David Barbee. Developing and testing a novel active-selection (ActSel) bycatch reduction device to quickly alternate trawls between capture and release configurations with real-time triggering. *Fisheries Research*, 254:106380, October 2022. doi:10.1016/j.fishres.2022.106380.

Alzayat Saleh, Issam H. Laradji, Dmitry A. Konovalov, Michael Bradley, David Vazquez, and Marcus Sheaves. A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. *Sci Rep*, 10 (1):14671, September 2020. ISSN 2045-2322. doi:10.1038/s41598-020-71639-x.

Matthew S. Savoca, Stephanie Brodie, Heather Welch, Aimee Hoover, Lee R. Benaka, Steven J. Bograd, and Elliott L. Hazen. Comprehensive bycatch assessment in US fisheries for prioritizing management. *Nat Sustain*, 3(6):472–480, June 2020. ISSN 2398-9629. doi:10.1038/s41893-020-0506-9.

Giulio Tarantino, Gregorio Motta, Paolo D'Ambrosio, Serena Felline, Valerio Sbragaglia, Stanislao Bevilacqua, Perla Tedesco, Giuseppe Scordella, and Antonio Terlizzi. Increasing trammel mesh size reduces biomass removal, mitigates discards and increases economic revenue in artisanal fisheries. *Frontiers in Marine Science*, 11:1267381, 2024. doi:10.3389/fmars.2024.1267381.

Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. Technical Report, University of North Carolina at Chapel Hill, USA, October 1995.

Katherine C. Wilson, Moses Lurbur, and Noëlle Yochum. Automated video processing to support commercial fishing sustainability and innovation in the walleye pollock (*Gadus chalcogrammus*) fishery in Alaska. *ICES Journal of Marine Science*, 82(9), 2025. doi:10.1093/icesjms/fsaf168.

Boris Worm, Ray Hilborn, Julia K. Baum, Trevor A. Branch, Jeremy S. Collie, Christopher Costello, Michael J. Fogarty, Elizabeth A. Fulton, Jeffrey A. Hutchings, Simon Jennings, Olaf P. Jensen, Heike K. Lotze, Pamela M. Mace, Tim R. McClanahan, Cóilín Minto, Stephen R. Palumbi, Ana M. Parma, Daniel Ricard, Andrew A. Rosenberg, Reg Watson, and Dirk Zeller. Rebuilding global fisheries. *Science*, 325: 578, July 2009. ISSN 0036-8075. doi:10.1126/science.1173146.

Noëlle Yochum, Michael Stone, Karsten Breddermann, Barry A. Berejikian, John R. Gauvin, and David J. Irvine. Evaluating the role of bycatch reduction device design and fish behavior on Pacific salmon (*Oncorhynchus* spp.) escapement rates from a pelagic trawl. *Fisheries Research*, 236:105830, April 2021. ISSN 0165-7836. doi:10.1016/j.fishres.2020.105830.

Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. *Dive into deep learning*. Cambridge University Press, 2023.

Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, volume 13682, pages 1–21. Springer Nature Switzerland, Cham, 2022. ISBN 978-3-031-20046-5 978-3-031-20047-2. doi:10.1007/978-3-031-20047-2_1.