Monitoring Fishing Activity on the Edge

MOBILIZING AI AND EDGE TECHNOLOGIES TO ADVANCE NEAR REAL-TIME ELECTRONIC MONITORING FOOTAGE REVIEW

PREPARED BY

tryo·labs

PREPARED FOR

The Nature Conservancy

Acknowledgements

The Tryolabs team would like to thank our volunteer Advisory Committee, Sanjay Nichani, Rakshit Agrawal, and Jeff Bier, for their thoughtful advice and technical guidance throughout this project. Their insights and support were instrumental in shaping the final AI-powered solution.

We also sincerely appreciate the collaboration and commitment of our project partners: Bureau Veritas, Deckhand, Thalos, Instituto Costarricense de Pesca y Acuicultura (INCOPESCA), and Cámara Nacional de Industria de Palangre (CNIP).

Finally, we are especially thankful to the vessel owners, captains, and crew who generously contributed their time, knowledge, and effort in the field. Their participation was essential to the success of this work.

This work was made possible thanks to The Nature Conservancy's collaboration with the Patrick J. McGovern Foundation and other supporters.

About Tryolabs

Tryolabs is an AI and data partner that helps organizations turn data and AI ambition into production-ready systems. With 15+ years of experience, we work with companies and institutions across industries to design and deploy custom machine learning solutions that are robust, scalable, and built to perform in the real world. Our expertise spans computer vision, MLOps, and edge AI, with a track record of shipping systems that operate reliably in complex, resource-constrained environments

Suggested citation

The Nature Conservancy and Tryolabs have made this work publicly available to support broad adoption of AI-enabled electronic monitoring across the tuna longline sector—advancing efforts toward more transparent and sustainable fisheries management. Please cite this work using the following reference:

Tryolabs, The Nature Conservancy. (2026). *Mobilizing AI and edge technologies to advance near real-time electronic monitoring footage review* (Report prepared for The Nature Conservancy)

Foreword

The health of our oceans—and the communities and economies that depend on them—hinges on our ability to sustainably manage industrial fisheries. Yet for decades, fisheries managers have lacked timely, reliable, actionable data to keep pace with the complexity of modern fishing operations. This initiative marks a transformative leap forward, demonstrating how artificial intelligence (AI) can be harnessed not just to streamline data collection on catch activity, but to fundamentally reshape how we manage industrial fishing. **By deploying an AI-powered system capable of analyzing electronic monitoring (EM) footage directly onboard longline vessels, this initiative brings near real-time visibility to one of the most opaque corners in the seafood supply chain.**

The implications for conservation are profound. With AI-predicted catch counts that rival expert human review and risk profiles delivered within hours of fishing activity, managers are no longer forced to rely on delayed, incomplete, or inaccurate self-reported catch data. Instead, they can act swiftly to enforce management objectives and deter illegal, unreported, and unregulated fishing. Beyond compliance, these rapid insights open the door to more dynamic management strategies—such as adjusting effort limits based on catch trends, identifying high-risk fishing zones for targeted monitoring or spatial planning, and coordinating with market actors to align supply chain decisions with sustainability goals. **In short, the system equips decision-makers with the timely, granular intelligence needed to move from data-poor management to proactive stewardship.**

The system's ability to automate and accelerate catch reporting means decisions can be made with confidence and speed. It's built to work alongside human experts, not replace them. EM reviewers remain in the loop to validate AI predictions, ensuring that final catch assessments are grounded in professional judgment. By handling the initial, time-intensive review of EM footage for catch events, the system allows managers to quickly verify AI outputs and redirect their attention to higher-value activities—such as enforcing regulations, refining management strategies and implementing on-the-water improvements. **By combining machine efficiency with expert oversight, the system delivers both scalability and accountability, reinforcing trust in the data that underpin fisheries management.**

The AI-powered system is a transferable and publicly available solution that all stakeholders—including fisheries managers, AI providers, and EM service providers—can adopt and iterate on. Its flexible, modular architecture and reliance on open-source tools enable it to be reconfigured for different target species and monitoring goals. All components, from the models to the reporting logic, were designed with reuse and scalability in mind to lower barriers to adoption and enable equitable access across regions and sectors.

By making the technology freely available, The Nature Conservancy and our partners aim to empower global longline fisheries operations to harness AI-enabled electronic monitoring for smarter, more sustainable management practices. To this end, our colleagues at Tryolabs have rigorously documented the development process to enable stakeholders to quickly—and freely—adopt the solution stack. To learn more about how the AI-powered system can be paired with traceability hardware to generate valuable data on seafood product verification, risk, and quality, see the report titled "*First Mile Transparency & Traceability on the Edge.*"

In an era of accelerating environmental change and growing demand for sustainable seafood, this AI-powered system demonstrates meaningful advancement in fisheries monitoring. It's a promising step toward a future where sustainable, global industrial fishing is not just an aspiration but a standard we achieve together.

Vienna Saccomanno,
Senior Scientist
Large-Scale Fisheries Program, The Nature Conservancy

Executive summary

This project developed and deployed an AI-powered system that analyzed electronic monitoring (EM) footage of longline fishing activity directly onboard vessels, delivering near real-time insights that significantly enhanced fisheries transparency. By combining edge computing, computer vision, and automated reporting, this system replaced time-intensive manual review of EM footage with same-day, actionable intelligence. Fishery managers can now receive species-level catch counts and fishing risk profiles within hours of each hauling event, enabling timely verification of the AI predictions, and informed decisions on planning, pricing, and supply chain logistics—all before a vessel returns to port.

The project delivered four key contributions:

- **A high-quality, curated dataset** of over one million images drawn from 29 hauling events on partner vessels (roughly 1,100 individual catch events), with species-level annotations provided by expert EM footage reviewers. The dataset was carefully prepared and processed to support the training of computer vision models and reflected a wide range of fishing activity across multiple species.
- **A performant AI-powered system** that detected, identified, and counted catch in near-real-time using a compact edge device. The system performed with remarkable accuracy on the most operationally valuable task: producing an accurate count of retained catch, with a miss rate of only 6%, making it reliable to support real-world decision-making.
- **An automated reporting and monitoring system** that compiled AI outputs into structured Daily Reports, including catch summaries, fishing risk profiles, visual evidence, and geolocated fishing data. Daily Reports were securely transmitted from the vessel via satellite internet within hours of fishing; additional real-time dashboards provided ongoing visibility into both fishing operations and AI performance.
- **A transferable, adaptable, and publicly available solution** that other stakeholders, including fisheries managers, AI providers, and EM service providers, can adopt and iterate on. Its flexible, modular architecture and reliance on open-source tools enable it to be reconfigured for different EM data infrastructures, target species, and monitoring goals. All components, from the dataset and models to the reporting logic, were designed with reuse and scalability in mind.

Table of contents

1. Introduction

Industrial fishing occurs on more than half of the ocean's surface, which is over three times the area covered by all land-based agriculture. It supplies billions of people with critical protein and supports millions of livelihoods worldwide. Yet today, over one-third of global fish stocks are being fished at unsustainable levels (Food and Agriculture Organization of the United Nations, 2020). In most industrial fisheries, fishers are required to report their own catch using logbooks, to ensure that quotas and regulations are met. However, without independent monitoring, this self-reporting is unverifiable, creating conditions where illegal, unreported, and unregulated (IUU) fishing can persist unchecked. IUU fishing is estimated to cause global economic losses between $10–20 billion annually (Agnew et al., 2009), while threatening marine biodiversity and sustainable fishing efforts.

To combat this lack of on-the-water monitoring, the Large-Scale Fisheries Program at The Nature Conservancy (TNC) has focused on scaling the use of electronic monitoring (EM) globally. EM involves equipping vessels with cameras, gear sensors, and GPS to continuously record and transmit information about fishing activity, giving fisheries managers and other stakeholders the means to independently verify logbook data. Despite EM's promise to improve fisheries transparency, adoption has been slowed by the high cost and logistical burden of reviewing EM footage. In many cases, hard drives must be physically retrieved after a vessel returns to port from fishing and sent to data review centers, where EM footage is manually analyzed, often months after the fact and long after the seafood has entered supply chains. Catch data verified in near real-time can support effective compliance and conservation actions.

To dismantle these major barriers to scaling EM, TNC partnered with Tryolabs to co-develop an AI-powered system capable of reviewing EM footage directly onboard vessels. The goal of our partnership was to develop a repeatable, edge-based EM footage review system that provided near real-time, verified information on the sustainability of a vessel's catch before products entered global supply chains. The system automated the detection, identification, and counting of catch using edge computing and computer vision, and rapidly compiled predictions for final human verification. This design enabled same-day reporting and near real-time monitoring without reliance on post-trip hard drive recovery or delayed human analysis.

The system was deployed on a semi-industrial longline fishing vessel in the Eastern Tropical Pacific that partnered with TNC for a pilot test of the AI-powered system. The vessel typically conducted two- to three-week fishing trips targeting tuna species, followed by one-week port stays. This operational rhythm allowed the Tryolabs team to collect EM footage under real-world conditions, annotate and validate the data, iteratively retrain the AI models, and deploy updated versions of the AI-powered system between trips.

The system integrated both hardware and software components in a robust, field-ready architecture (Figure 1). EM cameras installed on deck continuously recorded fishing operations, while a compact edge device—a NVIDIA Jetson AGX Orin—processed EM footage on the vessel in near real-time. The edge device ran a fully containerized AI-powered system, capable of detecting fish, identifying their species, tracking their movement across the vessel's deck, and counting both retained and discarded catch. Predictions were stored in a local database along with vessel telemetry and captain-entered electronic logbook (eLog) data, and compiled into structured Daily Reports. These reports included summary statistics, fishing risk profiles, and EM-based evidence clips for every AI catch prediction. Data were transmitted back to shore using Starlink satellite internet, and near real-time system performance was monitored through Amazon Web Services (AWS) using CloudWatch and Grafana dashboards.

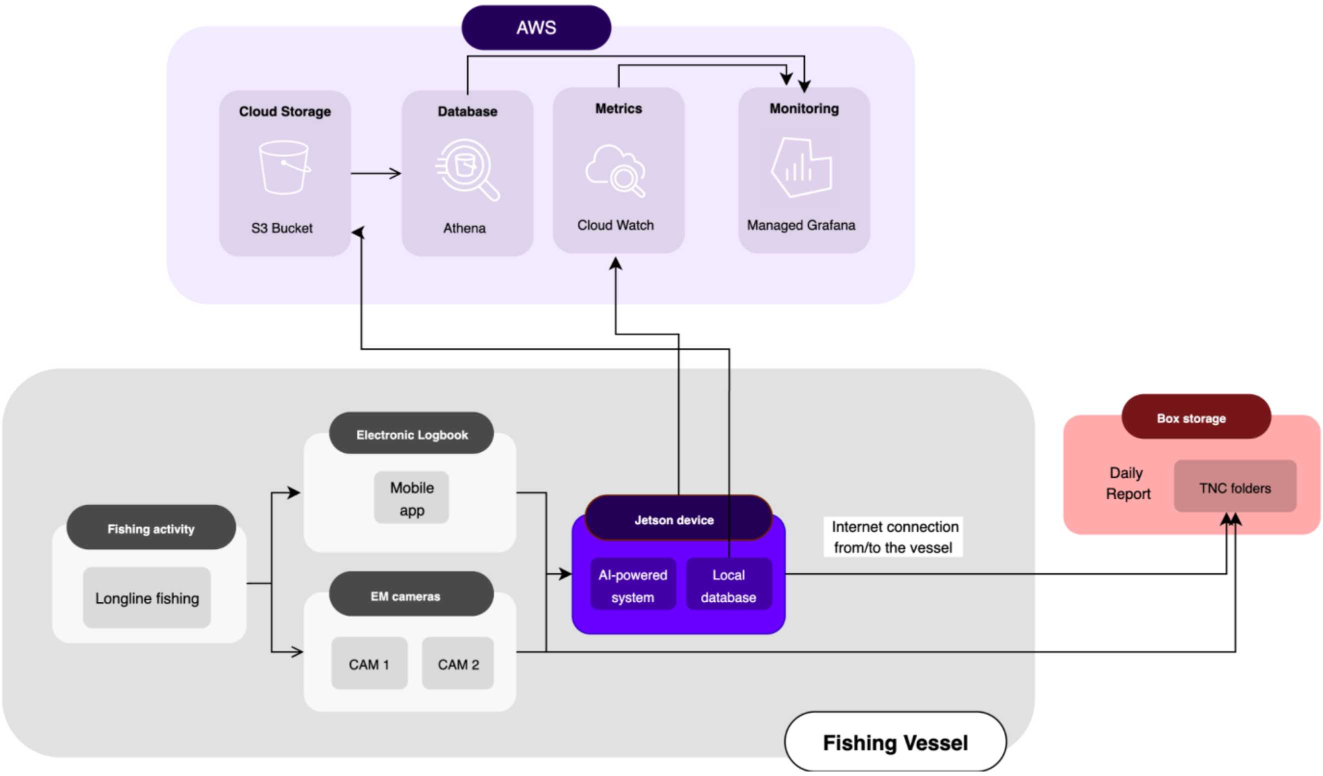


Figure 1. Simplified AI-powered system architecture (see Figure 12 for a more detailed system architecture diagram).

The remainder of this report follows the full machine learning operations lifecycle (Figure 2), which guided development of the AI-powered system from early research through real-world deployment:

- **Data** – how EM footage was collected, annotated, and transformed into a model-ready dataset
- **Model** – how the AI-powered system was trained, tested, and evaluated for performance
- **Development** – the software practices, tools, and quality assurance methods used to maintain reliability
- **Operations** – how the AI-powered system was deployed to vessels, monitored in production, and connected to reporting workflows
- **Feedback and iteration** – how field results were used to plan and design improvements to enhance accuracy, usability, and transparency.

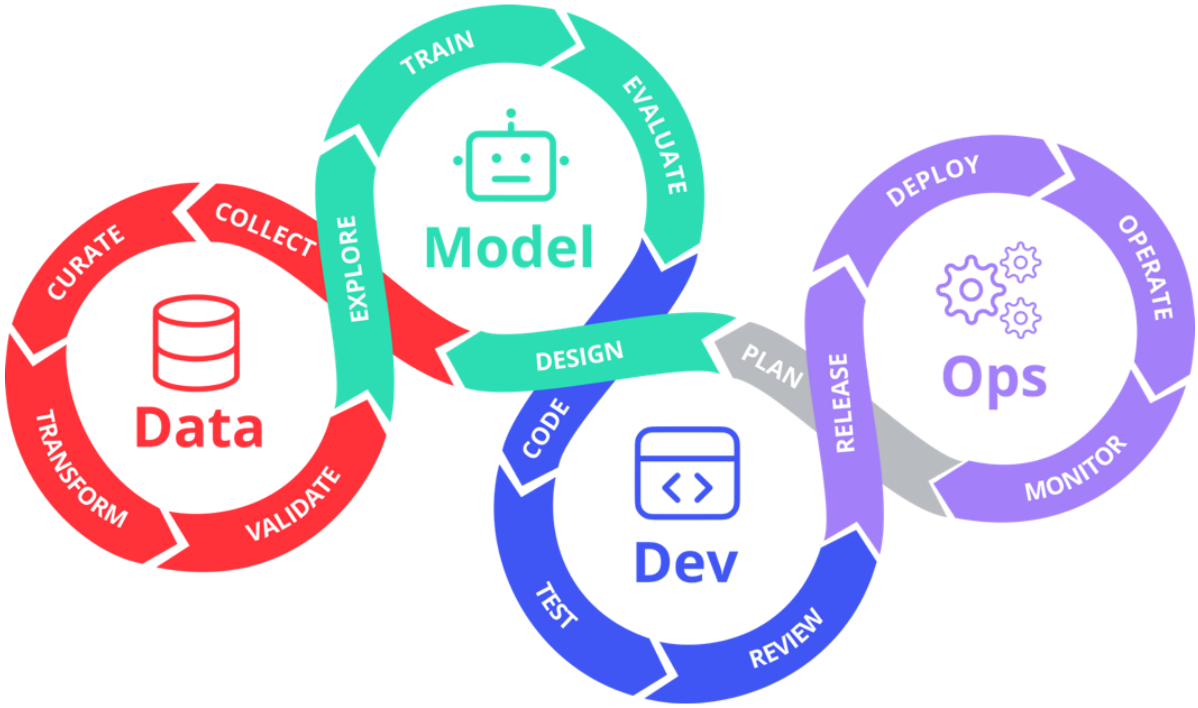


Figure 2. Machine Learning Operations (MLOps) lifecycle used to develop the AI-powered system. The diagram outlines the end-to-end process: from data collection and model training to deployment, monitoring, and iteration based on field performance.

Additional analyses, development experiments, and reference materials are included in the appendices:

- **Appendix A** provides a glossary of acronyms and terminology used throughout the report.
- **Appendix B** documents experimental features explored during development, including those that informed design trade-offs but were not included in production.
- **Appendix C** includes two supporting analyses: one on how training data volume affects model performance and cost, and another comparing AI catch counts to expert EM footage reviews and captain-submitted eLogs.
- **Appendix D** offers a high-level implementation guide for deploying the AI-powered system on new vessels.

These appendices offer technical depth for readers interested in implementation details, system limitations, and considerations for future improvements.

2. Data

The AI-powered system relied on three key data sources: EM footage from onboard EM cameras, species-level annotations created by expert reviewers for training data, and eLogs submitted daily by vessel captains. These data streams were enabled through close collaboration with project partners: Thalos provided and installed the EM hardware, Bureau Veritas (BV) carried out expert review of the EM footage, and Deckhand developed the tablet-based application used to submit eLogs.

This section focuses on the dataset used to train the fish detection model, which was created by aligning EM footage with expert-reviewed catch records and adding detailed visual annotations around individual catch. While eLogs supported live risk profile development and Daily Reports, they were not used for model training (see 5.3.1 Daily Report section for details).

2.1 Collection

Over the course of 17 fishing trips, EM cameras recorded 191 hauling events, capturing approximately 5,000 individual catch events. This full dataset was available to the project, but due to the time and cost of manual annotation, only a subset could be processed. In total, 29 hauling events, representing approximately 1,100 catch events, were fully annotated and validated to create the final dataset used for model training and evaluation. Due to confidentiality agreements, the training dataset will remain private.

At the start of the project, only three annotated hauling events (about 70 catch events) were available for initial training. As more EM footage was collected during fishing trips and annotated throughout the 10-month project, the dataset expanded in parallel with model development. This progressive growth allowed the model to improve incrementally, as shown in Figure 3, demonstrating that effective AI deployment could begin with a modest amount of data and improve as more is gathered.

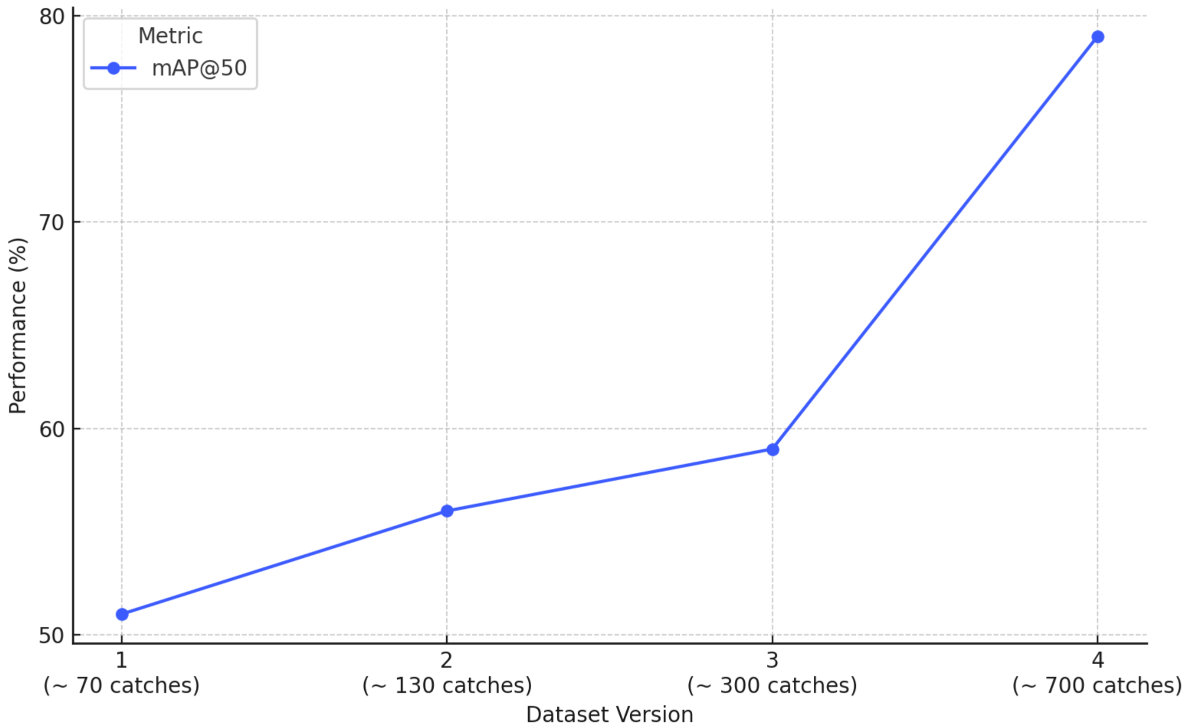


Figure 3. Model performance (mAP@0.5) improved steadily as more labeled data were added.

2.2 Curation

Expert reviewers from BV conducted 100% review of the EM footage and produced detailed Excel records for each catch, identifying species, catch timestamps, catch type (e.g., target or bycatch), fate (retained or discarded), and condition (alive, dead, etc.). However, these records did not include spatial annotations—there were no bounding boxes indicating the precise location of each catch within the EM footage frames.

To use the EM footage for model training, every catch had to be manually annotated with bounding boxes across the relevant EM footage frames. Given the manual effort required, a prioritization strategy was applied to select hauling events for annotation based on species frequency, ecological importance (e.g., conservation-priority species), and diversity (e.g., rare or unusual catch). This ensured efficient use of resources while capturing a wide range of species.

2.3 Annotation

Before annotation began, all EM footage went through a preprocessing pipeline (Figure 4). Originally recorded in five-minute clips, the EM footage was stitched into continuous video streams, normalized to 12 frames per second (FPS), and systematically renamed to include metadata such as vessel ID, trip number, and segment type. These steps ensured consistency and traceability throughout the annotation process.

To accelerate labeling and align with expert-reviewed catch data, each catch entry in the BV Excel records was used to create a pre-seeded annotation. This annotation was a bounding box placed near the expected time of appearance in the video. Annotators used these initial cues to locate and track each catch across multiple frames using CVAT (CVAT.ai, 2025), an open-source video annotation tool. This pre-seeding approach significantly reduced manual effort while ensuring that all tracks were grounded in verified catch events.

In more complex scenes, particularly where multiple landed fish appeared in piles on the deck, annotating individual catch became difficult due to occlusion and visual clutter. To address this, a dedicated object detection class called "FISHPILE" was introduced, allowing annotators to label these clusters as single objects. This solution preserved important visual context for training while reducing annotation time and cognitive load.

Annotation quality was ensured through a two-phase workflow supported by detailed written and video guidelines authored by Tryolabs. Tryolabs staff handled the first round of annotations internally to refine the labeling standards. Once finalized, the remaining annotations were completed by CVAT labeling services under a custom non-disclosure agreement (NDA) to ensure data privacy and protect the anonymity of participating vessel owners and crew.

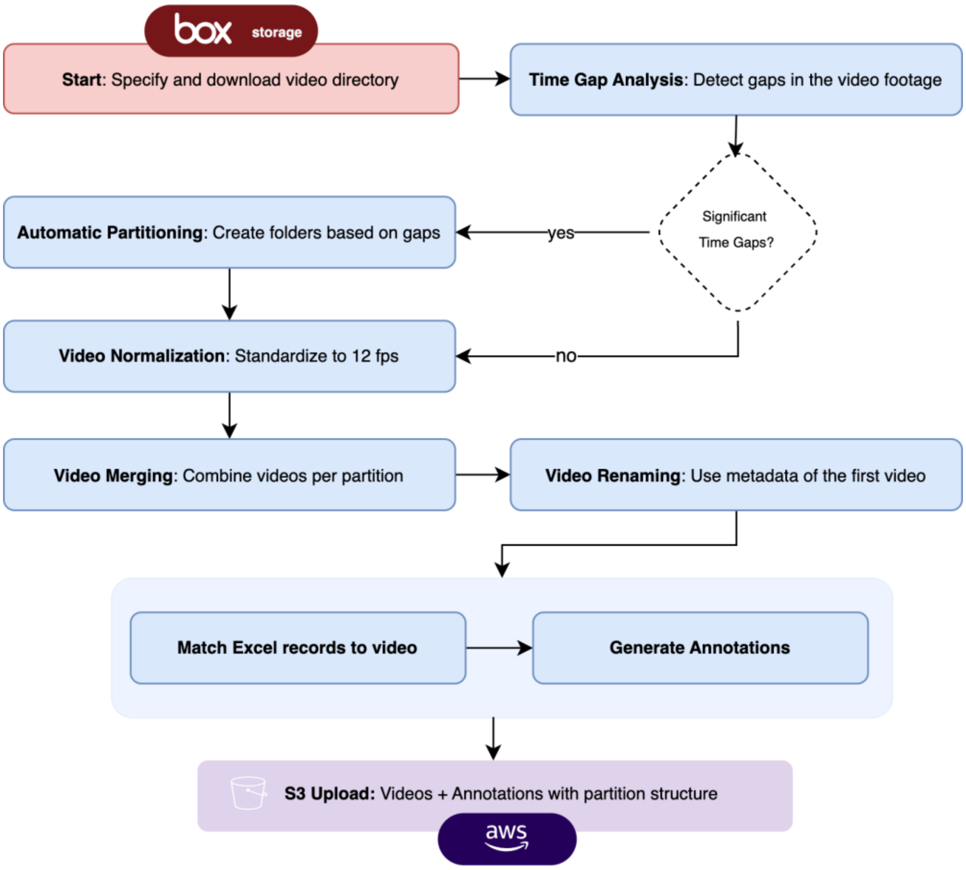


Figure 4. Data transformation workflow from raw EM footage and Excel records to pre-seeded EM footage for annotation.

2.4 Validation

Annotation validation was conducted continuously during the annotation phase. This included regular spot checks, where randomly selected tasks were reviewed to confirm that catches were correctly tracked across frames. Common issues, such as missed catches, inaccurate bounding boxes, or incorrect labels, were addressed through updated guidance to the annotation team.

After annotation, the dataset was cleaned to remove any invalid or inconsistent tracks. In total, 29 hauling events (roughly 1,100 catch events) were fully annotated and validated for use in model training and evaluation.

2.5 Final dataset

The final dataset used to train and evaluate the fish detection model included over one million images (1,025,010 frames) derived from the annotated EM footage. Key target species represented included Yellowfin Tuna (YFT), Striped Marlin (MLS), Common Dolphinfish (DOL), and Indo-Pacific Sailfish (SFA). As expected in real-world catch data, some species occurred much more frequently than others. This imbalance posed challenges for detecting rare species, which was addressed through targeted model training strategies (see 3.1.1 Handling class imbalance).

To prepare the dataset for training and evaluation, a train/test split was created, dividing the labeled data into two separate sets: one used to train the AI model (the training set) and another used to evaluate its performance on unseen data (the test set). The split was done by object track (i.e., per individual catch), rather than by frame to ensure no single catch appeared in more than one partition.

The initial split included:

- 951,767 training images
- 73,243 test images

The test set was drawn from six full days of EM footage that had not been used during training. These test days captured a range of conditions, including different lighting, species composition, and discard behaviors, to reflect real world variation.

Since catches appeared across multiple consecutive frames, the raw dataset contained many near-duplicate images. To reduce redundancy without losing visual diversity, perceptual hashing was applied to the training and validation sets. This process removed visually similar frames, producing a more compact dataset with:

- 308,085 training images
- 57,973 validation images
- 73,243 test images (unchanged to preserve evaluation consistency).

The species composition of target catch of the final dataset is detailed in Table 1, showing the number of tracks per target species across each partition.

Table 1. Target species composition of the final dataset (see A.2 FAO codes for species descriptions)

Species FAO code	Train tracks	Val tracks	Test tracks
YFT	62	15	10
MLS	42	10	10
DOL	66	16	8
SFA	60	15	12
BUM	15	3	1
SWO	13	3	6
BIL	14	3	1
WAH	2	1	0
TUN	3	0	0
TUS	1	1	0
BIP	1	1	0

The overall dataset preparation workflow is illustrated in Figure 5, outlining the full pipeline from raw EM footage through annotation, splitting, and deduplication.

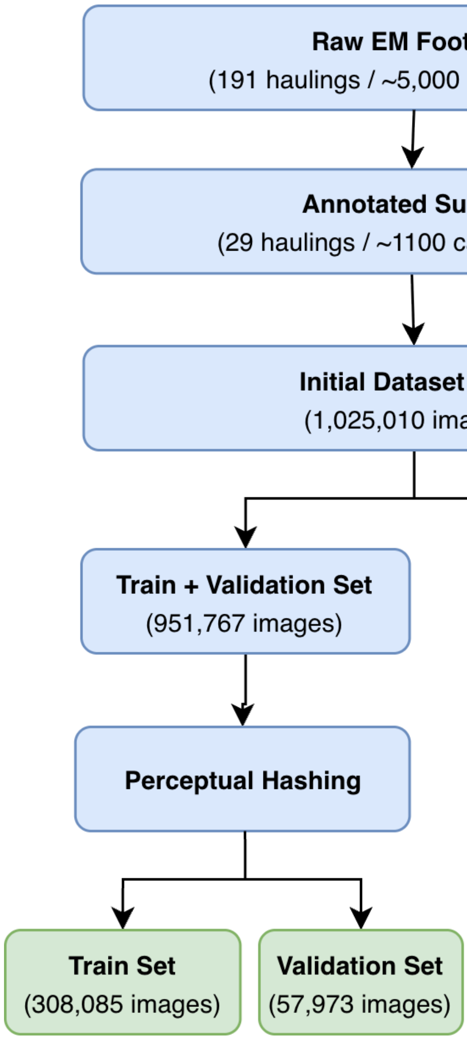


Figure 5. Dataset preparation workflow. From 191 hauling events, 29 were annotated and used to train the model. Data were split into train/validation/test sets, with perceptual hashing used to remove redundancy from the training and validation sets. The test set was unchanged.

To improve the model's ability to generalize, data augmentation was applied during training. This included color adjustments (hue, saturation, and brightness), geometric transformations (flips, rotations, and crops), and Mosaic augmentation, which combined multiple images into a single training frame. These transformations helped the model learn to detect catch under varied lighting, motion, and visibility conditions.

3. Model

This section outlines the AI-powered system deployed on an edge device to analyze EM footage directly onboard the vessel. The system consisted of three core components:

- **A fish detector,** which identified catch in video frames and classified catch events by species.
- **A fish tracker**, which maintained consistent catch identities across video frames.
- **A fish counter**, which classified catch events as Vessel Retained, Vessel Discard, or Water Discard, and generated total catch counts (see 3.3 Fish counter for event definitions).

3.1 Fish detector

The fish detector was based on the YOLOv11-M architecture (Jocher & Qiu, 2024) and was trained to identify individual fish in each frame of the EM footage and classify them by species.

Training began with pre-trained weights from the Fishnet.AI dataset (The Nature Conservancy, 2022) and was conducted on NVIDIA RTX 3090 GPUs hosted by Tryolabs. The model was then trained for 300 epochs on 640×640 images, using the Adam optimizer with cosine learning rate decay and dropout (rate 0.228). The loss function combined components for bounding box regression, class prediction, and focal loss. Hyperparameters were tuned using Optuna, and all training metrics and artifacts were tracked using MLflow for reproducibility.

3.1.1 Handling class imbalance

To prevent the model from overfitting to common species, a custom batch sampling strategy was implemented. Each training batch consisted of 32 images and was constructed to ensure (i) balanced sampling across rare, normal, and common species, and (ii) the inclusion of negative images with no detections to reduce false positives (Table 2).

Table 2. Custom batch sampling strategy used during training to balance species representation.

Species Group	Sampled classes per batch	Sampled images per class	Total sampled images	% of Batch
Rare	9	1	9	28%
Normal	5	3	15	47%
Common	1	6	6	19%
Negative (no detections)	–	–	2	6%
Total	–	–	32	100%

3.1.2 Fish detector performance

The fish detection model was evaluated on the test set using standard object detection metrics, including precision, recall, and mean Average Precision at IoU thresholds of 0.5 (mAP@0.5) and 0.5:0.95 (mAP@0.5:0.95). Table 3 summarizes the model's per-species performance across seven target catch classes present in this fishery.

Table 3. Fish detector performance on target species.

Class	Images	Instances	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Weighted Avg	35,681	43,752	78%	69%	73%	59%
YFT	15,843	23,914	80%	73%	74%	67%
MLS	7,371	7,371	77%	73%	73%	64%
SFA	5,318	5,318	71%	72%	75%	49%
SWO	4,511	4,511	88%	38%	63%	34%
DOL	2,160	2,160	76%	68%	75%	41%
BUM	328	328	27%	77%	70%	63%
BIL	150	150	0%	0%	1%	0%

The model achieved strong overall performance: 78% precision, 69% recall, 73% mAP@0.5, and 59% mAP@0.5:0.95. These values were weighted by the number of examples per species, so species with more instances in the dataset have a greater influence on the overall scores. Performance was highest for frequent species like Yellowfin Tuna (YFT), Striped Marlin (MLS) and Indo-Pacific Sailfish (SFA), while species with fewer than 500 images each, like Billfish (BIL) and Blue Marlin (BUM), showed reduced accuracy due to limited representation in the training data.

Figure 6 provides a deeper view of classification behavior through a normalized confusion matrix. Each row represents the ground-truth label, and each column shows the AI-predicted label. The matrix captures how frequently each species was confused with others.

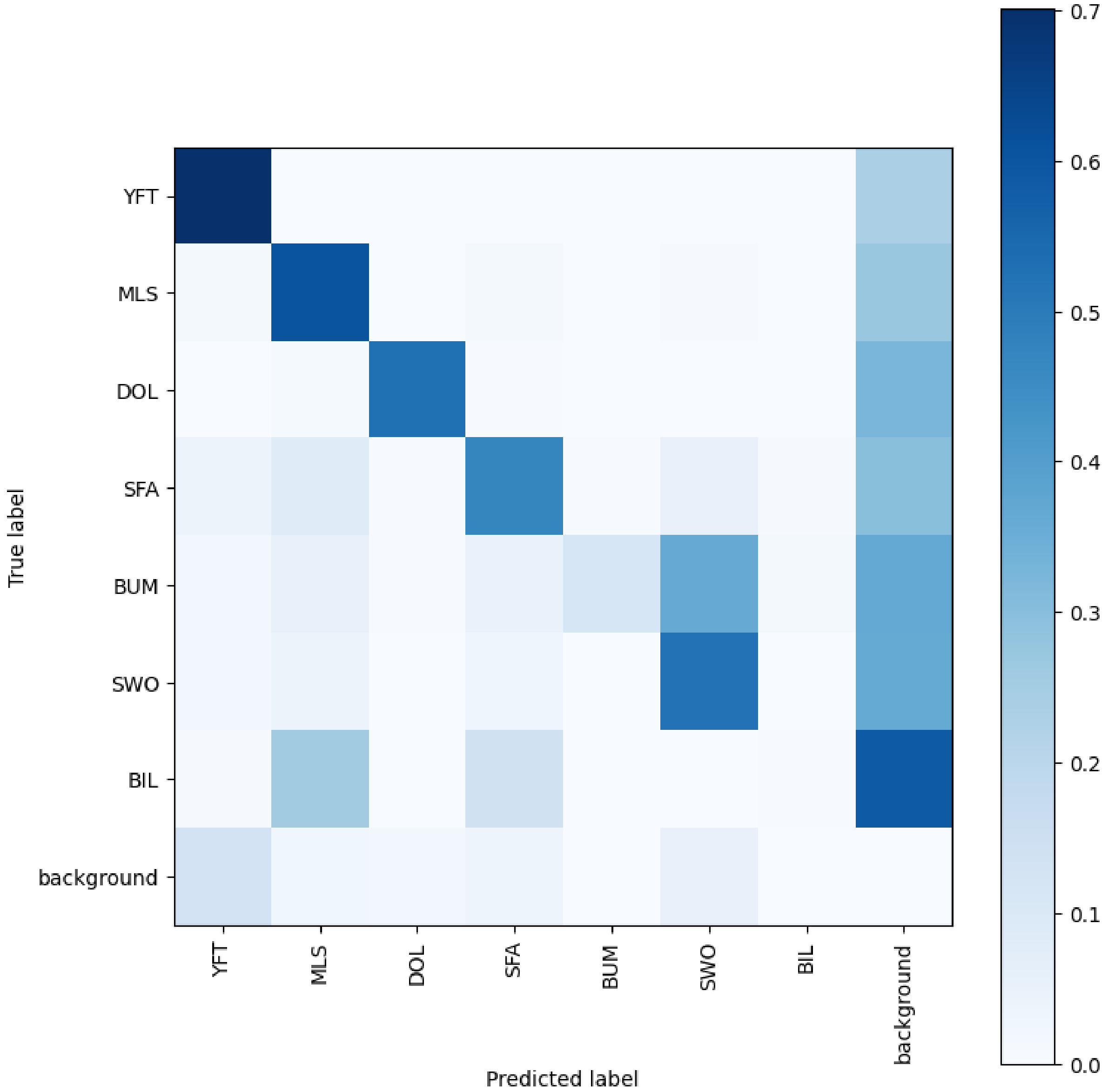


Figure 6. Confusion matrix for fish species classification. Each row represents the actual species label, and each column represents the model's predicted label. Strong diagonal values reflect correct classification, while off-diagonal patterns highlight common confusions.

The confusion matrix confirms robust classification of frequent species such as YFT, MLS, DOL, SFA, and SWO, all of which show strong diagonal dominance. Some confusion occurred with the background object class, which represents regions where no fish are present. Values in the background row indicate false positives, where the model predicted a fish in an empty area. In contrast, values in the background column reflect false negatives, where the model failed to detect a fish, effectively treating it as background. This type of error is particularly common for rare species such as BUM and BIL, which were more likely to be missed due to their limited representation in the training data.

Figure 7 shows the Precision-Recall (PR) curve for the model across all object classes.

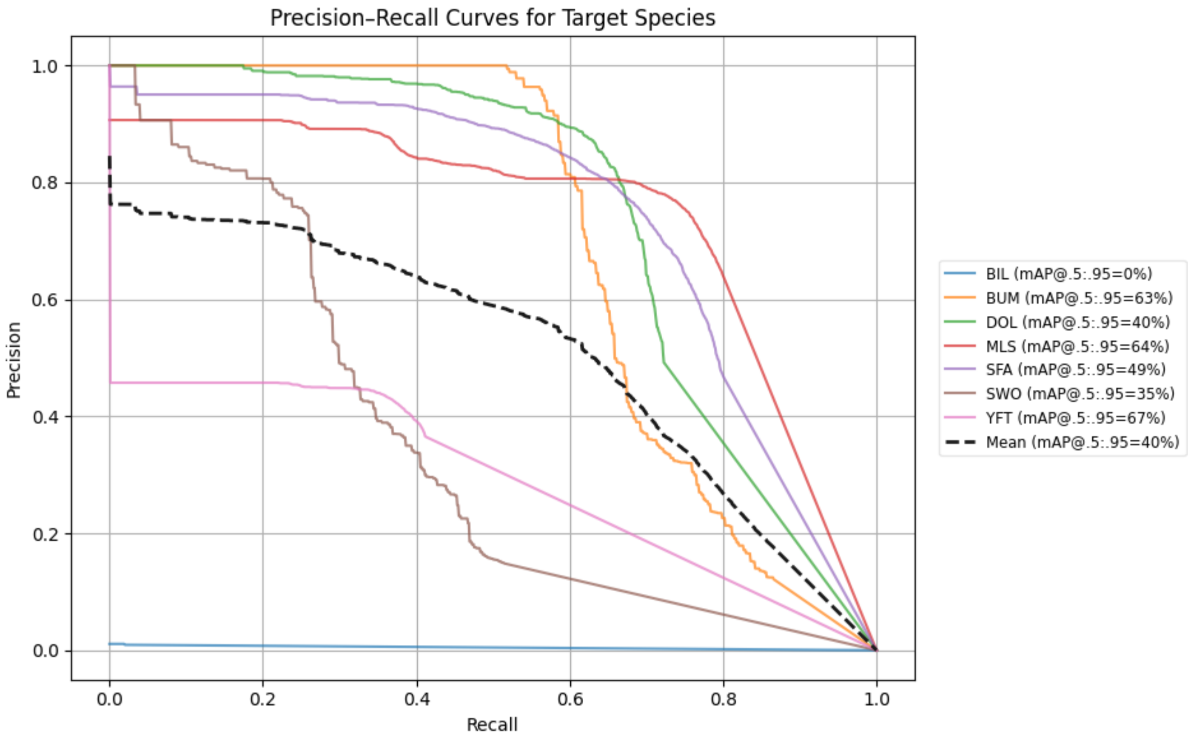


Figure 7. Precision-Recall (PR) curve across all target object classes. The black dashed curve shows the average performance, with an overall mAP@0.5:0.95 of 40%. Despite extreme class imbalance, the model demonstrates strong detection performance across frequent categories.

While aggregated mAP@0.5:0.95 was 40%, this was heavily influenced by underrepresented classes with minimal training data. Frequent species maintained a strong balance between precision and recall, even under varying IoU thresholds, reaffirming the model's operational reliability.

Although fine-grained species classification is valuable, the fish detector's primary operational goal was to reliably identify and count catch events with high confidence. Accurate event detection is essential for key monitoring objectives such as catch verification, risk assessment, and compliance oversight.

3.2 Fish tracker

The fish tracker used the BoT-SORT algorithm to assign a unique ID to each catch and maintain that identity across frames. This was essential for understanding movement over time and preventing double counting. Parameters were manually tuned to balance stability and computational efficiency, minimizing identity switches during occlusions or motion blur.

3.3 Fish counter

The fish counter was designed to handle two key scenarios to ultimately provide predictions on the total catch count that was retained, as well as discarded. The first was when catch was brought onboard the vessel and then either retained ("Vessel Retained") or discarded ("Vessel Discard"). The second scenario was when catch was released and discarded directly from the sea without ever being brought onboard ("Water Discard").

3.3.1 Vessel retained and discarded catch

For onboard events, a rule-based counter, based on Ultralytics' Object Counter (Jocher & Qiu, 2024), applied domain-specific logic to determine when catch was retained or discarded.

A virtual line was drawn along the deck edge of the vessel (Figure 8), and each track was analyzed based on how the centroid of the predicted box crossed this virtual line. If a track moved from the water to the deck and stayed on the deck, it was counted as a Vessel Retained catch. If a track moved from the water to the deck and then back to the water, it was counted as a Vessel Discard event.

To increase reliability, a short delay was introduced before confirming a count, allowing clearer frames to be captured once the catch was fully out of the water. Very short tracks (<12 frames) were filtered out to avoid miscounts from light reflections or foam, and once a track was complete, the most frequently predicted species was assigned to ensure a stable classification.

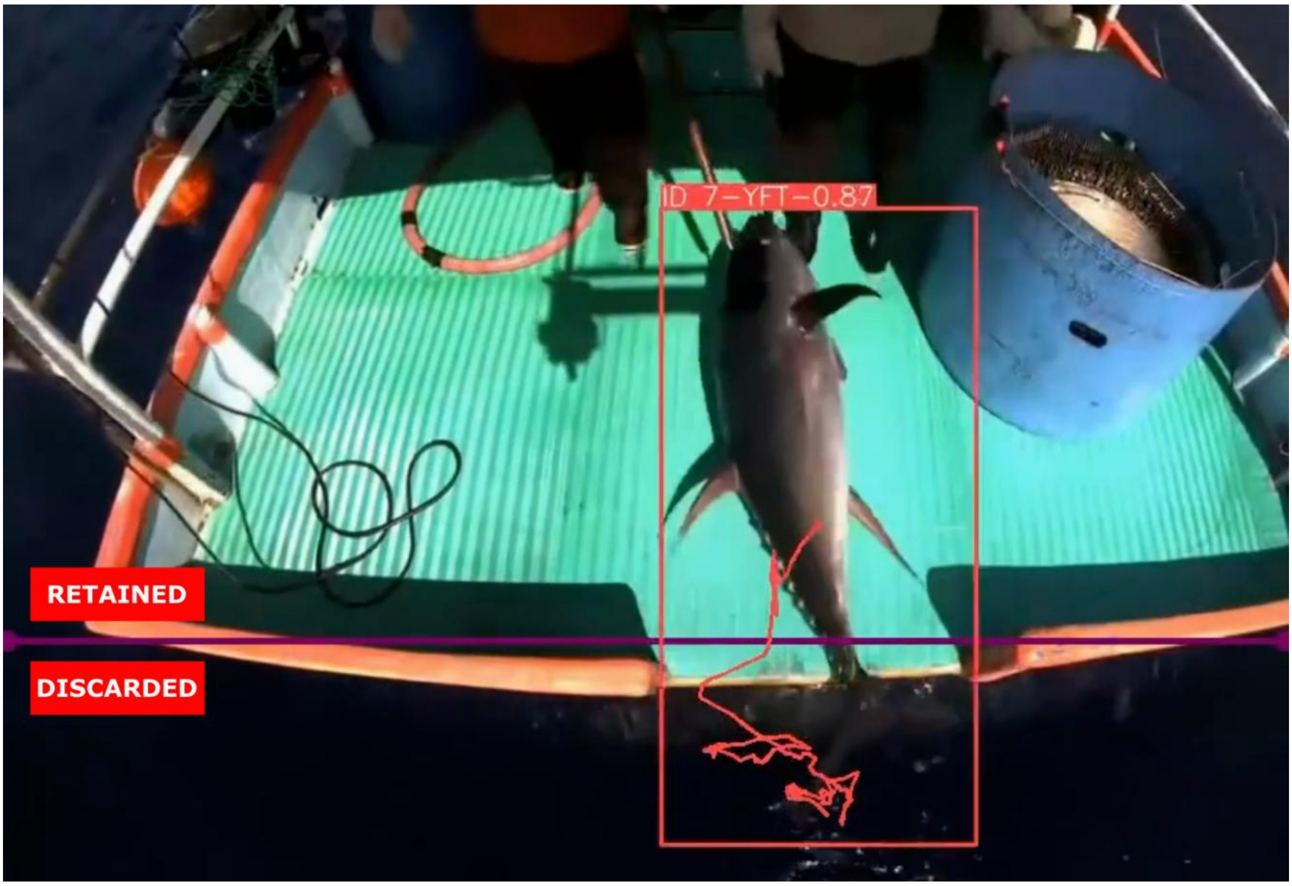


Figure 8. Example output from the AI-powered system showing fish detection, species classification, and tracking. A detected fish is shown in a red bounding box, labeled with a unique track ID, FAO species code (e.g., YFT for yellowfin tuna), and model confidence score. The red trajectory line shows the fish's movement across consecutive video frames. The purple virtual line represents a boundary on the deck used to classify events: tracks that cross onto the deck and remain onboard are labeled Retained, while tracks that cross and then return to the water are labeled Discarded.

3.3.2 Catch discarded in the water

Some species, non-commercial species, were not brought onboard the vessel. These discard events are visually and procedurally different: catch might only appear briefly at the water's surface, often obscured by foam or partially submerged, and never cross the virtual line.

Because these events were difficult to identify in real-time, Water Discard predictions were handled as a retrospective process. At the end of each day, a dedicated script processed all detection and tracking data to flag likely Water Discard events using species-specific heuristics. These rules identified consistent, water-level AI model predictions that lacked any corresponding Vessel Retained or Vessel Discard events nearby in time, helping to distinguish true discards from false positives or interrupted tracks.

3.3.3 Fish counter performance

The fish counter was evaluated by comparing each AI-predicted catch event against ground-truth annotations from BV. To be considered correct, predictions had to match the event type (retained or discarded), and align closely in time with the corresponding ground-truth crossing frame.

Performance was assessed using detection metrics derived from the confusion matrix: true positives (TP), false positives (FP), and false negatives (FN). Two error metrics were used to quantify model performance: Miss Rate, defined as FN / GT, representing the proportion of ground-truth events that were not detected by the model, and False Discovery Rate (FDR), defined as FP / Pred, representing the proportion of AI predictions that were incorrect. A ±10% miss rate was established as the deployment threshold for operational use.

Performance metrics of the Fish Counter

			Count								
Icon	Class	Event	GT	Pred	TP	FP	FN	Precision	Recall	Miss Rate	FDR
	CATCH	RETAINED	135	130	127	3	8	98%	94%	6%	2%
	CATCH	VESSEL DISCARD	5	7	3	4	2	43%	60%	40%	57%
	CATCH	WATER DISCARD	31	34	24	10	7	71%	77%	23%	29%

Model: Final Model | Test set: Final test dataset (6 haulings) | Icons: Tryolabs – M&D Team

Figure 9. Fish counter performance by event type. Comparison of AI predictions with BV ground-truth for Retained catch, Vessel discard, and Water discard events. Counts include ground-truth (GT), predictions (Pred), true positives (TP), false positives (FP), and false negatives (FN). Performance metrics include Precision (TP / Pred) and Recall (TP / GT). Error metrics include Miss Rate (FN / GT) and False Discovery Rate (FP / Pred).

Figure 9 summarizes fish counter performance by event type. The model achieved strong and dependable performance in the system's most critical task: detecting Vessel Retained catch. This category showed a low miss rate (6%) and very low false discovery rate (2%), indicating that most retained catch events were successfully detected while incorrect AI alerts were rare. These results fall comfortably within the acceptable range for operational use.

In contrast, detecting discard events—both Vessel Discards and Water Discards—was more challenging. Vessel Discards showed a 40% miss rate and 57% false discovery rate, indicating that many discard events were missed and that more than half of the AI detections did not correspond to true events. Water Discards performed somewhat better, with a 23% miss rate and 29% false discovery rate, but still showed noticeably lower reliability compared to retained catch detection. It should also be noted that the number of discard events in the test set was relatively small, particularly for Vessel Discards; therefore, the reported error rates should be interpreted with caution, as a small number of additional detections or misses could substantially change the calculated metrics.

Performance metrics of the Fish Counter on target species												
			Count									
Icon	Class	Event	GT	Pred	TP	FP	FN	Precision	Recall	Miss Rate	FDR
	YFT	RETAINED	11	16	11	5	0	69%	100%	0%	31%
	DOL	RETAINED	10	11	10	1	0	91%	100%	0%	9%
	SFA	RETAINED	17	22	14	8	3	64%	82%	18%	36%
	SWO	RETAINED	15	4	3	1	12	75%	20%	80%	25%
	MLS	RETAINED	15	14	9	5	6	64%	60%	40%	36%
	BUM	RETAINED	3	3	1	2	2	33%	33%	67%	67%
	BIL	RETAINED	1	1	0	1	1	0%	0%	100%	100%

Model: Final Model | Test set: Final test dataset (6 haulings) | Icons: Tryolabs – M&D Team

Figure 10. Fish counter performance for retained catch at the target-species level. Comparison of AI predictions with BV ground-truth across retained target species in the test set. Counts include ground-truth (GT), predictions (Pred), true positives (TP), false positives (FP), and false negatives (FN). Performance metrics include Precision (TP / Pred), Recall (TP / GT), Miss Rate (FN / GT), and False Discovery Rate (FP / Pred).

A more fine-grained evaluation at the target-species level shows that retained-catch detection was strong for several of the most operationally important species (Figure 10). Yellowfin Tuna (YFT) and Common Dolphinfish (DOL) both achieved 100% recall, indicating that all ground-truth retained events for these species were accurately detected by the fish counter. Performance was more variable across the remaining target species, with moderate results for Indo-Pacific Sailfish (SFA) and Striped Marlin (MLS), and substantially weaker performance for Swordfish (SWO), Blue Marlin (BUM), and Billfish (BIL), especially in recall. These lower scores likely reflect inter-species confusion within the billfish group, whose similar visual characteristics make them difficult to distinguish reliably in EM footage, especially under occlusion, motion blur, and limited representation in the training data.

In practice, many of the remaining errors could still be addressed through operational review. Some false positives and species-level misclassifications could be readily reviewed by a human reviewer during Daily Report validation (see 5.3.1 Daily Report), and others could be further reduced in future work using rule-based post-processing, such as excluding AI predictions made outside typical fishing hours, at high vessel speeds, or with low model confidence.

Importantly, the AI-powered system also detected several valid catch events of species that were not included in the original expert-reviewed annotations due to human error. Across the six test hauling events (~90 hours of EM footage), seven such human omissions were identified, highlighting the potential of AI-assisted review to complement and strengthen human workflows. While expert review remains the gold standard, these findings underscore the benefits of combining automated analysis with targeted human validation.

4. Software development

This section outlines the key practices, tools, and workflows used to ensure robust, maintainable, and high-quality software throughout the project lifecycle.

4.1 Codebase

The source code is publicly available under [TNC's GitHub repository.](#) The codebase used a structured Git workflow (Figure 11) to ensure a clear separation between active development and stable releases. The *main* branch served as the production-ready codebase deployed to the vessel's edge device. New features were developed in isolated branches created from *develop*, which acted as the integration branch. Once reviewed and validated, *feature* branches were merged into *develop*, and when a release was ready, develop was merged into *main*.

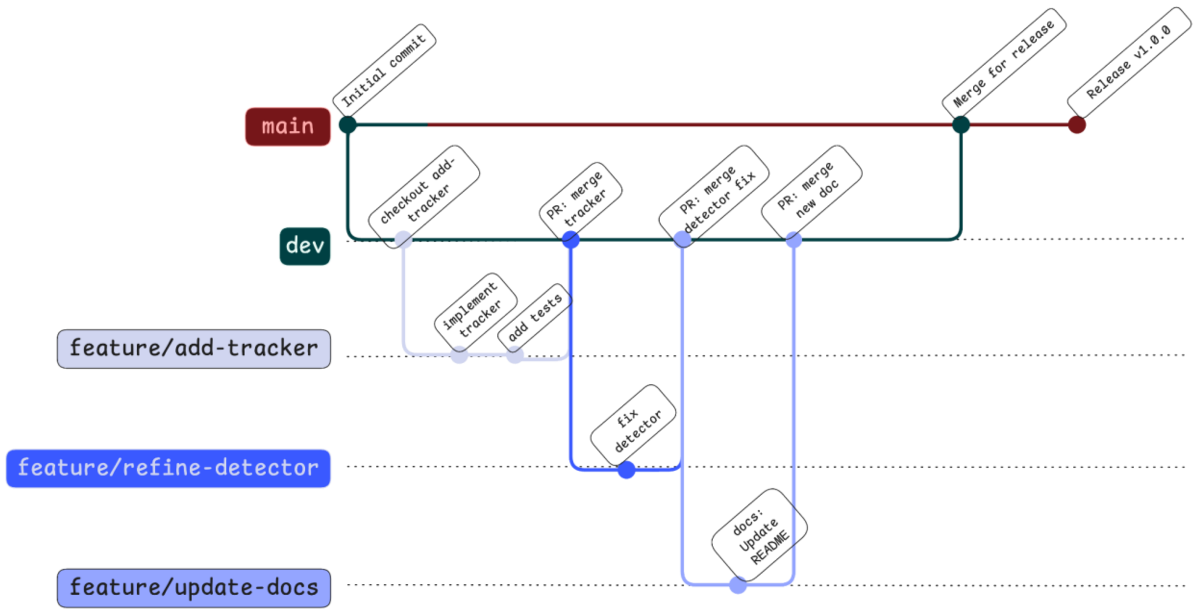


Figure 11. The code lifecycle from development to a stable release.

The package manager uv was used to ensure efficient dependency management and reproducible environments, while Ruff was used to enforce code style consistency and perform fast, reliable static code analysis across the codebase.

4.2 Testing

Code quality was safeguarded through a comprehensive automated testing framework, triggered via pre-commit hooks and GitHub Actions as part of Continuous Integration (CI). This process automatically ran checks on every proposed change before merging, including:

- Static analysis with Ruff to enforce consistent style and detect early issues
- Secret scanning to prevent accidental exposure of sensitive credentials
- Execution of a unit test suite covering core components such as metric calculations, data aggregation, and tracking utilities

In addition to automated CI checks, candidate releases were tested in a dedicated staging environment that mirrored the vessel hardware. An edge device on Tryolabs' premises was configured identically to the deployed AI-powered system. All code was run end-to-end under real-world constraints to ensure performance and stability before deployment.

4.3 Reviewing

In addition to automated checks, all new code contributions required a mandatory peer code review. At least two other developers reviewed each submission to ensure logical correctness, adherence to architectural patterns, and overall code quality before merging it into the development branch.

Leveraging this template-driven approach streamlined the development process, reduced setup time, and reinforced best practices throughout the project lifecycle.

5. Operations

The AI-powered system was designed to operate reliably in real-world fishing conditions by combining onboard processing with cloud-based infrastructure. This hybrid architecture allowed most tasks, such as analyzing EM footage and generating reports, to happen directly on the vessel, while also enabling remote management and monitoring through the cloud.

The architecture consisted of two main environments:

- **Edge environment (onboard the vessel):** Ran the AI-powered system locally to process EM footage, detect, classify, and count catch, and stored results in a local database, even if internet connectivity was temporarily lost.
- **Cloud environment (hosted on AWS):** Managed software updates, collected system health metrics, and provided access to structured data and reports for onshore monitoring teams.

This architecture supported operational resilience: the AI-powered system continued to function independently during connectivity outages, and automatically synchronized data to the cloud once the connection was restored. The full workflow is illustrated in Figure 12.

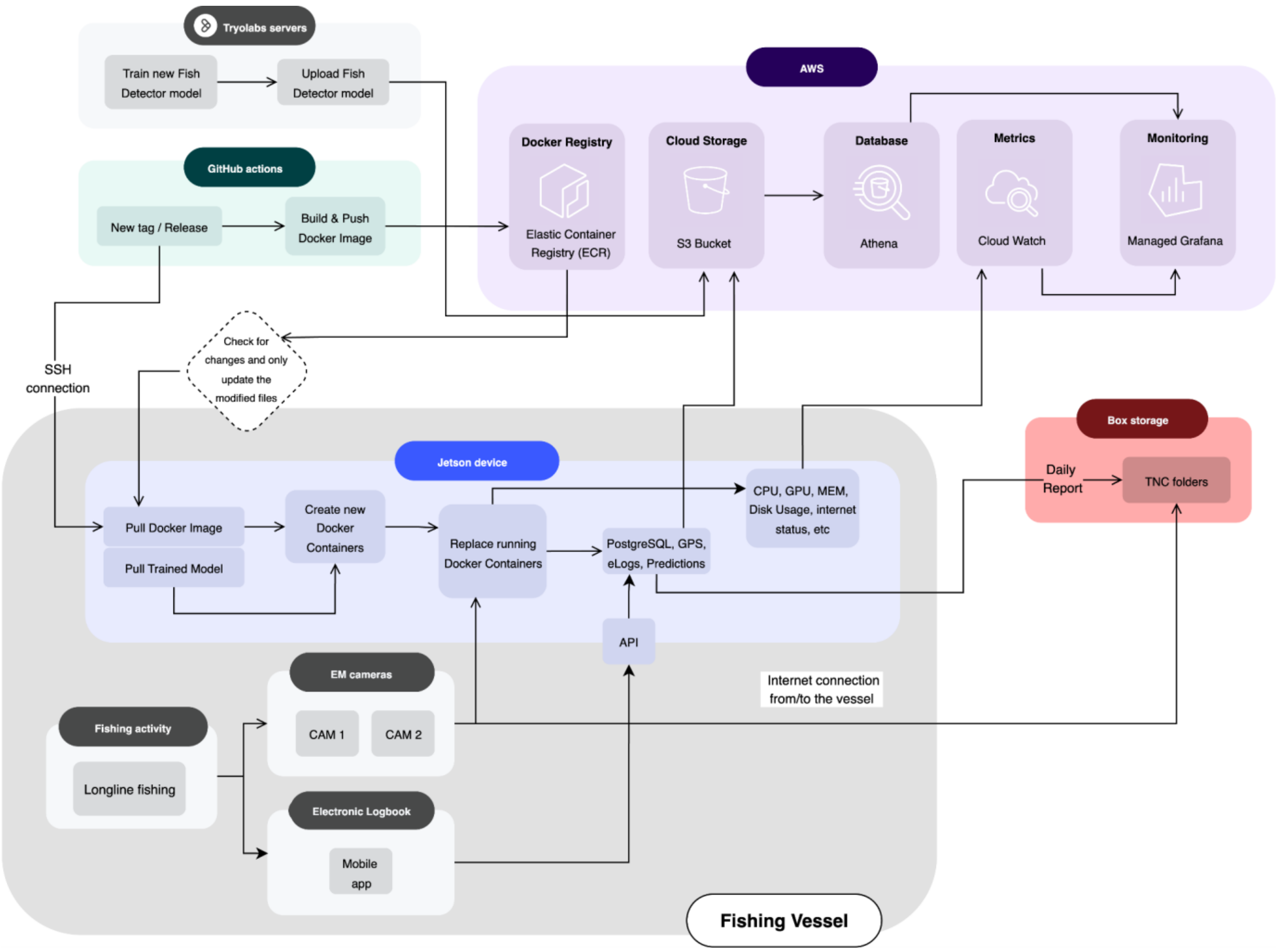


Figure 12. System architecture for AI-powered fisheries monitoring. The diagram shows the full workflow, from onboard components, EM cameras, tablet-based eLogs, edge device, and Starlink connectivity, to cloud infrastructure for model deployment, data storage, and remote monitoring. Key external services included GitHub, Tryolabs servers, Box, and AWS.

5.1 Hardware on vessel

The onboard hardware stack consisted of EM cameras to capture fishing activity, an edge device to run the AI-powered system, and a tablet used by the captain to submit eLogs. The edge device's software environment was fully containerized with Docker, ensuring consistent deployments and simplifying updates.

All data generated onboard the vessel, including AI model predictions, eLog entries, and vessel telemetry, was stored locally in a PostgreSQL database to maintain data integrity even during connectivity loss. A Starlink terminal provided satellite internet, enabling continuous synchronization with the cloud. System metrics were streamed to AWS CloudWatch for remote performance monitoring. At the end of each day, the AI system automatically compiled a structured Daily Report (see 5.3.1 Daily Reports) and uploaded it to a designated and secure cloud storage folder for timely review by onshore monitoring teams.

5.2 Releases and deployments

A Continuous Integration/Continuous Deployment (CI/CD) pipeline, managed through GitHub Actions, automated the delivery of updated software components to the edge device. This system provided a standardized procedure for deploying the fish detector, fish tracker, and fish counter with minimal manual intervention.

The deployment workflow was initiated upon the creation of a new code release. The process involved uploading the most recent trained fish detector model to an AWS S3 bucket and concurrently building a Docker image that contained the latest application source code. This image was then pushed to the AWS Elastic Container Registry (ECR). The pipeline then established a remote connection to the edge device, pulled the updated assets, and replaced the existing Docker container with a new one instantiated from the updated image.

5.3 Monitoring & evaluation

5.3.1 Daily Report

The Daily Report (Figure 13) was generated on the vessel's edge device and sent ashore at the end of each 24-hour fishing period. It consolidated the AI model's predictions into structured, actionable data, eliminating the delays of traditional EM footage review workflows that often require offloading hard drives and manually reviewing EM footage post-trip.

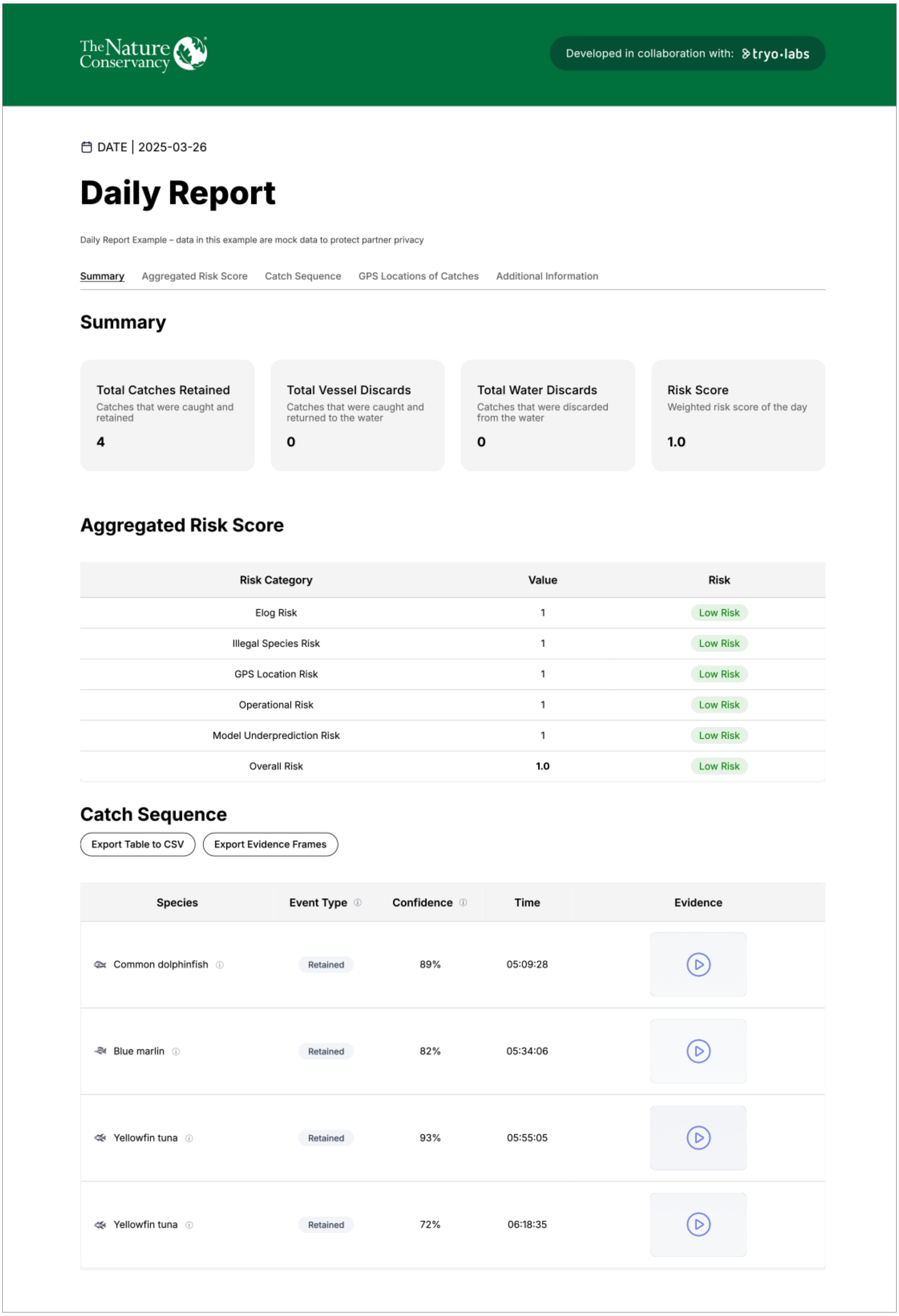
Each Daily Report began with a summary of the day's fishing activity, including the total number of Vessel Retained, Vessel Discard, and Water Discard catch. It also included a risk score designed to help monitoring teams prioritize secondary human review of fishing activity. The score ranged from 1 (no risk) to 3 (high risk) and was calculated by weighting five different components, summarized in Table 4.

Table 4. Risk score components used in Daily Reports

Risk Component	Description	Primary Concern
ELog Risk	Discrepancy between AI-predicted catch counts and captain's eLog records	Potential underreporting of catch by vessels
Illegal Species Risk	Detection of illegal species in catch	Retention of ETP or illegal bycatch
GPS Risk	Geolocation of fishing activity	Fishing occurring in marine protected areas and other sensitive habitats
Model Underprediction Risk	Signs the AI may have missed catch events	Reduced reliability of AI-based monitoring
Operational Risk	Technical system issues such as missing GPS or video data	Loss of monitoring coverage that could enable non-compliance

Each component was scored daily, then combined into a weighted average where the first three components (ELog, Illegal Species, GPS) carried greater weight because they reflected compliance-related concerns. The last two components (Model Underprediction, Operational) were given lower weight so that technical issues alone did not unduly raise a vessel's risk profile.

Beyond summary metrics, the Daily Report included a detailed catch report for every prediction event. Each entry recorded the predicted species, event type (retained or discarded), model confidence, timestamp, and a representative clip from the moment the catch crossed the virtual counting line. These evidence clips provided valuable visual context, transparency, and facilitated human validation of AI model performance. The report also included a map of the vessel's fishing locations for the day, with geolocated catch shown relative to regulatory boundaries such as marine protected areas (MPAs) and exclusive economic zones (EEZs), allowing for quick geographic interpretation of events.



The Nature Conservancy

Developed in collaboration with: tryo·labs

DATE | 2025-03-26

Daily Report

Daily Report Example – data in this example are mock data to protect partner privacy

Summary | Aggregated Risk Score | Catch Sequence | GPS Locations of Catches | Additional Information

Summary

Total Catches Retained
Catches that were caught and retained
4

Total Vessel Discards
Catches that were caught and returned to the water
0

Total Water Discards
Catches that were discarded from the water
0

Risk Score
Weighted risk score of the day
1.0

Aggregated Risk Score

Risk Category	Value	Risk
Elog Risk	1	Low Risk
Illegal Species Risk	1	Low Risk
GPS Location Risk	1	Low Risk
Operational Risk	1	Low Risk
Model Underprediction Risk	1	Low Risk
Overall Risk	**1.0**	Low Risk

Catch Sequence

Export Table to CSV | Export Evidence Frames

Species	Event Type	Confidence	Time	Evidence
Common dolphinfish	Retained	89%	05:09:28	
Blue marlin	Retained	82%	05:34:06	
Yellowfin tuna	Retained	93%	05:55:05	
Yellowfin tuna	Retained	72%	06:18:35	

Figure 13. Example of a Daily Report generated by the AI-powered system using mock data. All reports include: (1) a summary of Vessel Retained, Vessel Discard, and Water Discard catch counts; (2) a risk scoring table that flags potentially non-compliant activity; and (3) a detailed sequence of predicted catch events, each accompanied by species label, event type, timestamp, model confidence, and an evidence clip captured at the moment of counting.

5.3.2 Grafana dashboards

While the Daily Report provided end-of-day intelligence, additional monitoring and evaluation capabilities were developed using AWS Managed Grafana to offer near real-time visibility. These dashboards allowed both technical and operational teams to monitor system health and fishing activity as it happened, without having to wait for Daily Report delivery.

Data from each vessel, including AI predictions, Daily Reports, edge device health metrics, and EM camera consistency checks, was uploaded to S3 and exposed through AWS Athena, enabling Grafana to query and visualize it on a scheduled refresh cycle. This setup supported multiple dashboards tailored to different user needs.

Operational dashboard

The operational dashboard (Figure 14) enabled the technical team to monitor the health and status of the onboard edge device. It displayed key metrics such as internet connectivity, GPU and CPU usage, disk status, and EM camera angle consistency. This dashboard was particularly useful for verifying that the edge device was online, and operating within safe resource limits. It also helped track any periods of downtime, identifying when and for how long the system was offline and unable to process EM footage, highlighting areas of the EM footage where further investigation by human review may be necessary.

Figure 14. Operational dashboard on Grafana.

Real-Time Catch

The real-time catch dashboard (Figure 15) visualized catch events just minutes after they occurred, giving monitoring teams timely awareness of vessel activity. Users could observe catch progression, species distribution, and individual event details as the trip unfolded. This dashboard was especially valuable for surfacing potential IUU activity in near real-time, enabling faster detection and management action.

Figure 15. Real-time catch dashboard on Grafana.

Daily Reports

The Daily Report dashboard (Figure 16) presented the same core information as the Daily Reports—such as catch counts, species breakdowns, and risk scores—but with the added benefit of vessel-specific historical comparison. Users could explore trends across time, compare catch composition across hauling events or trips, and monitor how risk scores evolved over time. This supported longer-term analysis and helped monitoring teams identify behavioral patterns, recurring issues, or potential areas for enforcement or intervention.

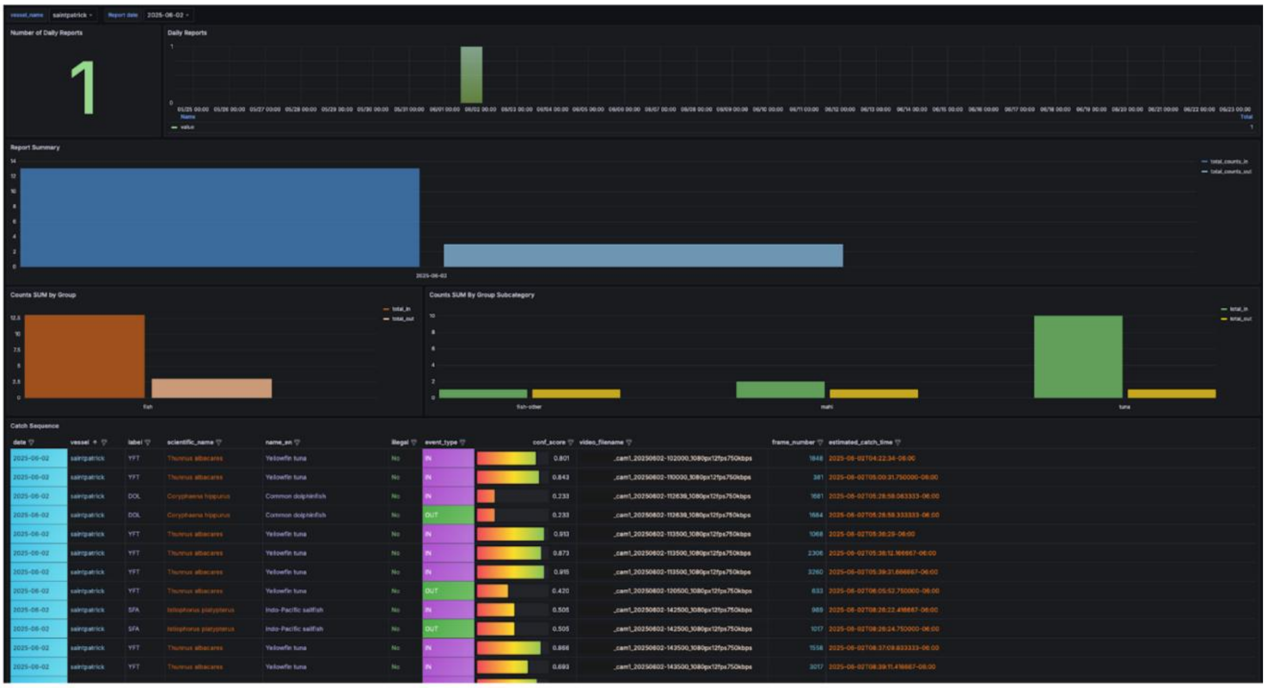


Figure 16. Daily Report dashboard on Grafana.

5.3.3 Camera position monitoring

A dedicated camera position monitoring tool ran on the vessel to detect if the EM camera position was moved and ensure consistent EM footage quality. Because the model relied on a stable view of the vessel's outline to detect Vessel Retained and Vessel Discard events, even minor shifts in camera angle could degrade performance. The monitoring system checked the position of the deck edge every 20 minutes using a combination of edge detection and template matching. It recorded the Intersection over Union (IoU) and confidence score, sending these values to CloudWatch for display within the Grafana operational dashboard. This automated check ensured that any issues with camera placement were caught early and could be addressed before affecting model accuracy.

6. Feedback and iteration

Following initial deployment, the AI-powered system was continuously evaluated aboard active fishing vessels to identify areas for performance improvement. These real-world operations surfaced practical challenges that could not be fully anticipated during development. In response, targeted refinements were made to improve the system's robustness, accuracy, and usability across three key dimensions: species identification, false positive reduction, and reliable reporting. In parallel, a Proof of Concept (PoC) was developed to explore future functionality around catch size estimation.

6.1 Species classification improvements

Early deployments revealed that the object tracker often followed catch for too long, sometimes well after catch had been brought onboard, processed, and placed into a pile of fish on the deck. As a result, the AI-powered system occasionally assigned incorrect labels based on distorted or overlapping visual input from the latter part of the track. To address this, the tracker was modified to finalize tracks after a fixed number of frames, representing the typical duration needed to capture a clear view of the catch. This change reduced misclassification caused by label drift over time and ensured that the AI model predictions were based on the most relevant visual information.

Alongside these changes, the counting logic was revised to improve species identification at the moment a catch was counted. When a track first appeared—typically while the catch was still in the water and partially obscured by foam—early AI model predictions tended to be low-confidence and error-prone. As catches were hauled onboard, visibility improved, and the AI model's predictions became more accurate. To take advantage of this, the AI-powered system deferred the registration of Vessel Retained and Vessel Discard events for a short interval after the virtual line crossing. This delay allowed the model to accumulate a sequence of clearer AI model predictions, and the species label was then determined using the most frequent prediction across that portion of the track.

6.2 False positive reduction

During field operations, the model frequently misidentified background objects, such as hoses, buoys, and crew boots, as catch. To address this, the training dataset was augmented with 14,524 negative images (about 5% of the training set) containing no catch but including common false-positive triggers (Figure 17). An additional 2,520 negative samples were added to the validation set. These additions taught the model to better distinguish catch from background clutter, resulting in a 25% relative improvement in overall mAP@0.5:0.95 compared to the same model trained without these negative images.

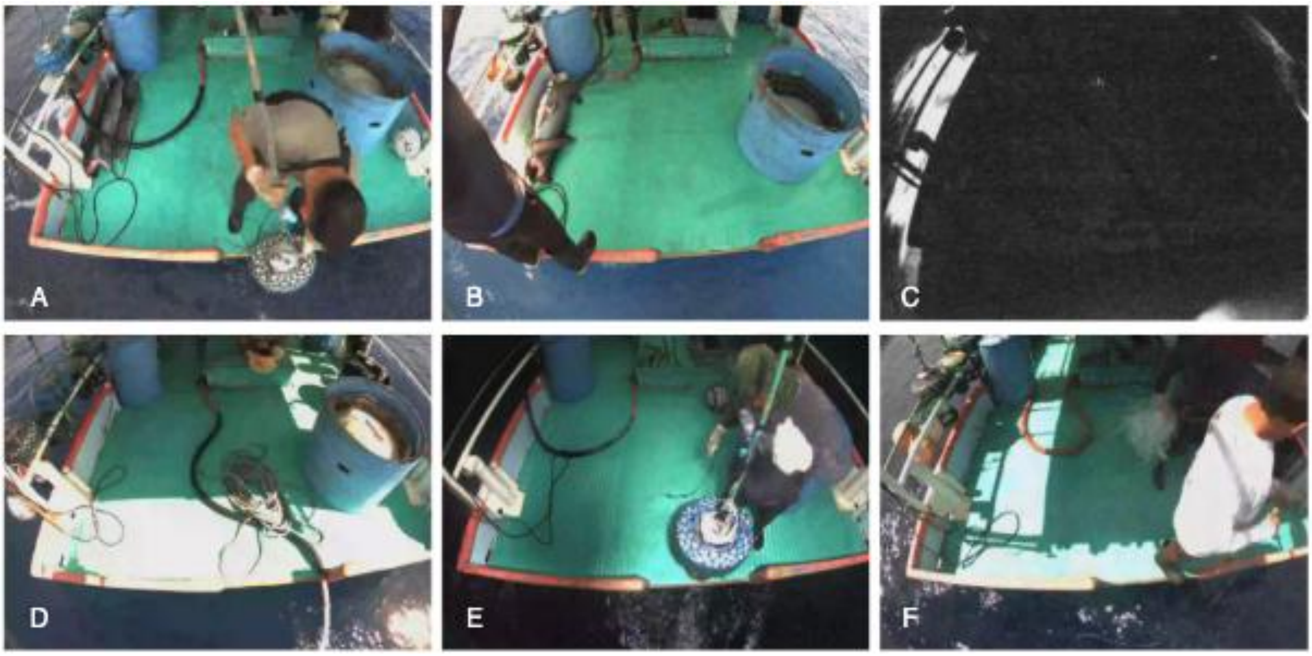


Figure 17. Example images used to reduce false positives by introducing common background elements such as gear, boots, and deck objects into the training process. The AI-powered system initially misclassified these non-fish objects as follows: (A) Dolphinfish (DOL), (B) Pelagic Stingray (PLS), (C) Yellowfin Tuna (YFT), (D) Indo-Pacific Sailfish (SFA), (E) Dolphinfish (DOL), and (F) Pelagic Stingray (PLS).

6.3 Reliable and timely Daily Reports

Several improvements were made to ensure consistent Daily Report generation and timely delivery. Initially, network interruptions onboard vessels sometimes caused the system to fail when generating or uploading reports. To address this, a dedicated monitoring container was introduced. This service continuously checked whether the previous day's report had been successfully generated. If not, it automatically triggered its creation and upload. This ensured temporary failures did not cause gaps in reporting and were resolved without manual intervention.

The timing of the report itself also proved to be a critical design decision. Analysis of partner vessel activity revealed that most hauling occurs during the night and early morning, with fishing typically concluding by early afternoon. Based on this insight, the report generation time was set to 5:00 PM local time, aligning the report's cut-off with the typical end of the fishing day. This ensured that reports reflected a full day's activity while remaining timely.

6.4 Catch size estimation PoC

A PoC was developed to evaluate the feasibility of estimating catch size directly from EM footage. For some regulated species, the size of the catch determines the legality of retention, and integrating length estimation could enable real-time detection of illegal retention.

The PoC implementation included the following components:

- **Lens calibration:** Intrinsic camera parameters were estimated using chessboard patterns to correct radial and tangential distortion. These images were obtained in collaboration with the vessel's crew, aiming to capture as many chessboard pattern angles as possible (Figure 18). This step was essential for accurately mapping image pixels to real-world coordinates.

- **Perspective correction:** Using known reference points and deck measurements provided by vessel crews, a homography transformation was applied to correct for camera tilt and positioning. This step ensured consistent scale across the entire frame and enabled length calculations from bounding boxes.

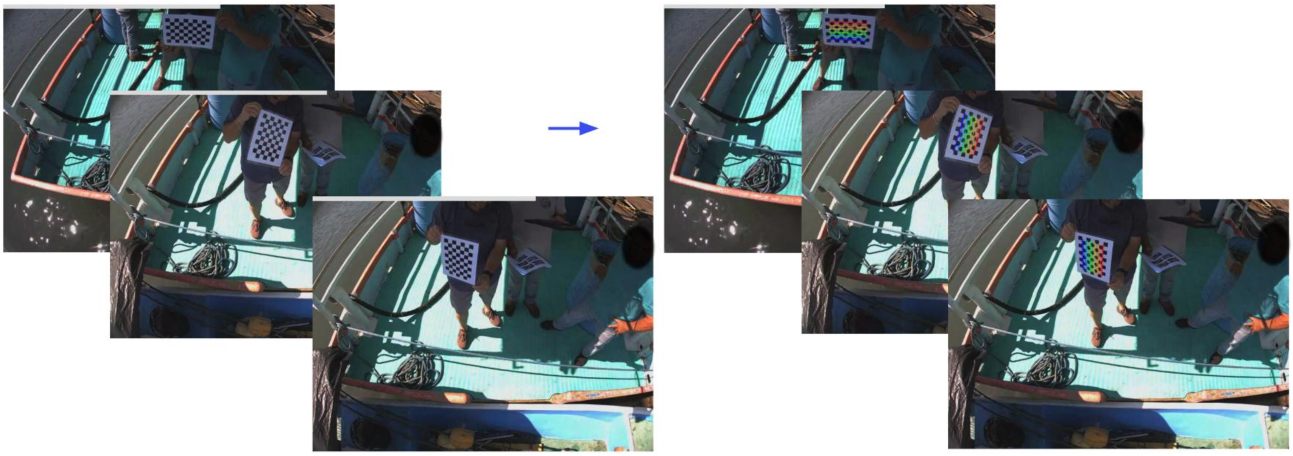


Figure 18. Sample chessboard calibration images used to correct camera distortion and prepare for size estimation.

The PoC for catch length estimation demonstrated highly encouraging results, achieving an average error of just 0.3 ± 10 cm across test samples. This level of precision, obtained using only EM footage and simple onboard calibration, showed that reliable size measurements were feasible using existing EM infrastructure.

Deploying this in production could be an important step forward: by combining species classification with length estimation, the system could move beyond simple catch counts to provide more detailed information on both the type and size of each individual catch. This added granularity could have meaningful applications: for conservation, it could support enforcement of size-based regulations more efficiently; for industry, it could offer early insights into the size and composition of catch, helping improve planning, pricing, and supply chain logistics before vessels return to port.

7. Conclusions

This project demonstrated a practical and adaptable approach to using edge-based AI for fisheries monitoring, showing that accurate, timely, and scalable independent monitoring is possible even in resource-constrained environments. Developed as a modular and reusable platform, the AI-powered system processed EM footage directly onboard vessels, reducing months-long review delays to same-day reporting and enabling near real-time visibility into fishing activity.

A major outcome was the integration of tools that made monitoring workflows more efficient and transparent. The automated Daily Report compiled AI-predicted catch events, flagged potentially non-compliant activity, and included evidence clips to support rapid human verification. Alongside this, live dashboards built with Grafana gave operational and technical teams immediate insight into vessel activity and system performance, allowing issues to be identified and addressed without delay.

The system achieved excellent accuracy in counting retained catch, with a miss rate well within operational thresholds—affirming its reliability for real-time monitoring. While performance on core tasks was strong, challenges remain in secondary tasks such as species-level classification of rare species and in the detection of discard events. Still, initial field tests revealed cases where the AI model identified catch missed by expert human reviewers, highlighting its potential not only as an automation tool, but as a complementary layer of quality assurance in EM workflows.

Ultimately, this project showed how edge-based AI could help shift fisheries electronic monitoring from a delayed, manual process to one that supports real-time decision-making. By making the system publicly available, TNC and Tryolabs offer a replicable model that can support broader adoption across the tuna longline sector—advancing efforts toward more transparent, responsive, and sustainable fisheries management.

8. References

Agnew, D. J., Pearce, J., Pramod, G., Peatman, T., Watson, R., Beddington, J. R., *et al.* (2009). Estimating the worldwide extent of illegal fishing. *PLoS ONE, 4*(2), e4570. https://doi.org/10.1371/journal.pone.0004570

CVAT.ai. (2025). *CVAT: Computer Vision Annotation Tool* [Computer software]. GitHub. https://github.com/cvat-ai/cvat

Food and Agriculture Organization of the United Nations. (2020). *The state of world fisheries and aquaculture 2020: Sustainability in action.* https://doi.org/10.4060/ca9229en

Jocher, G., & Qiu, J. (2024). *Ultralytics YOLO11* (Version 11.0.0) [Computer software]. GitHub. https://github.com/ultralytics/ultralytics

The Nature Conservancy. (2022). *Fishnet Open Images Database* (Version 1.0.0) [Dataset]. https://www.fishnet.ai/description

Appendix A: acronyms & glossary

A.1 Acronyms

Acronym	Definition
AI	Artificial Intelligence
AWS	Amazon Web Services
BV	Bureau Veritas
CI/CD	Continuous Integration / Continuous Deployment
CPU	Central Processing Unit
CVAT	Computer Vision Annotation Tool
EEZ	Exclusive Economic Zone
EM	Electronic Monitoring
ETP	Endangered, Threatened, and Protected
FAO	Food and Agriculture Organization
FPS	Frames per Second
GPU	Graphics Processing Unit
IoU	Intersection over Union
IUU	Illegal, Unreported, and Unregulated
KPI	Key Performance Indicator
mAP	Mean Average Precision
MLOps	Machine Learning Operations
MPA	Marine Protected Area
NDA	Non-Disclosure Agreement
RAM	Random Access Memory
SSH	Secure Shell
TNC	The Nature Conservancy
TOPS	Tera Operations Per Second
YOLO	You Only Look Once

A.2 FAO codes

The AI-powered system generated species-level predictions using FAO (Food and Agriculture Organization) codes. These are three-letter codes assigned by the FAO to identify specific fish species and other marine organisms.

Acronym	Scientific name	Common name
BIL	Istiophoridae	Marlins, sailfishes, etc. nei
BIP	*Sarda orientalis*	Striped bonito
BRO	*Carcharhinus brachyurus*	Copper shark
BSH	*Prionace glauca*	Blue shark
BTH	*Alopias superciliosus*	Bigeye thresher
BUM	*Makaira nigricans*	Blue marlin
DOL	*Coryphaena hippurus*	Common dolphinfish
FAL	*Carcharhinus falciformis*	Silky shark
GES	*Gempylus serpens*	Snake mackerel
LEC	*Lepidocybium flavobrunneum*	Escolar
MLS	*Kajikia audax*	Striped marlin
MRW	*Masturus lanceolatus*	Sharptail mola
PLS	*Pteroplatytrygon violacea*	Pelagic stingray
PSK	*Pseudocarcharias kamoharai*	Crocodile shark
PTH	*Alopias pelagicus*	Pelagic thresher
REM	*Remora spp.*	Shark suckers
RMV	*Mobula spp.*	Mobula nei
RSK	Carcharhinidae	Requiem sharks nei
SFA	*Istiophorus platypterus*	Indo-Pacific sailfish
SKH	*Selachimorpha*	Pleurotremata, Various sharks nei
SPN	*Sphyrna spp.*	Hammerhead sharks nei
SPL	*Sphyrna lewini*	Scalloped hammerhead
SPY	Sphyrnidae	Hammerhead sharks, etc. nei
SPZ	*Sphyrna zygaena*	Smooth hammerhead
SWO	*Xiphias gladius*	Swordfish
THR	*Alopias spp.*	Thresher sharks nei

TTX	Testudinata	Turtles and shelled relatives
TUG	*Chelonia mydas*	Green sea turtle
TUN	Thunnini	Tunas nei
TUS	*Thunnus spp.*	True tunas nei
WAH	*Acanthocybium solandri*	Wahoo
WHH	*Tetrapturus spp.*	Spearfishes nei
YFT	*Thunnus albacares*	Yellowfin tuna

A.3 Glossary

Term	Definition
Amazon Athena	AWS service for running SQL queries directly on data stored in S3, without managing servers.
Amazon Cloudwatch	AWS service for monitoring, collecting, and analyzing real-time metrics, logs, and events from AWS resources and applications.
Amazon S3 Bucket	Scalable object storage used to store raw footage, annotations, and trained model weights.
Batch size	The number of training examples processed in a single iteration before the model's parameters are updated.
Bureau Veritas	Third-party EM footage analysts (i.e., fish taxonomy experts) who reviewed full video sets and produced species-level catch datasets.
BoT-SORT	Multi-object tracking algorithm that assigns unique IDs to detected fish across frames based on motion cues.
CI/CD	Development workflow that automates testing, integration, and deployment to ensure rapid and reliable code delivery.
CVAT	Open-source tool for annotating video and image data, used to create labeled training datasets.
Edge Computing	Processing data locally on a device (e.g., onboard a vessel) instead of sending it to the cloud.
Epoch	One complete pass through the entire training dataset during the training of a machine learning model.
Fish Counter	Model component that operates on top of the Fish Detector and Fish Tracker, using a virtual counting line to estimate retained and discarded catch based on track trajectories.
Fish Detector	Model component based on YOLOv11-M architecture, trained to detect and classify fish species in video frames.
F1-Score	Harmonic mean of precision and recall, used to evaluate model performance by balancing false positives and false negatives.
False Negative	The model incorrectly predicted the negative class when it should have been positive (e.g., the model didn't detect a catch).

False Positive	The model incorrectly predicted the positive class (e.g., predicts a catch when there was no catch observed).
Grafana	Open-source platform for building dashboards and visualizing system metrics in real-time.
IoU	Metric used in object detection to quantify overlap between predicted and ground-truth bounding boxes.
Jetson	NVIDIA's edge AI hardware platform for running real-time AI model predictions on low-power embedded devices.
MLflow	Open-source platform for managing the machine learning lifecycle, including experiment tracking, model versioning, and reproducibility.
NVIDIA	Technology company providing GPUs and AI platforms, including Jetson, used for onboard AI model predictions.
Perceptual Hashing	Technique that generates hash values based on an image's visual content, used to identify duplicate or near-duplicate frames in video streams.
Precision	Measures the accuracy of positive predictions (catches). It answers the question: "Of all the instances the model predicted as positive, how many were actually positive?".
Recall	Measures the model's ability to find all the actual positive instances. It answers the question: "Of all the instances that were actually positive (catches), how many did the model correctly identify?".
Re-ID (Re-identification)	Process of matching the same object across different frames or camera views using visual appearance features.
Retained Catch	Catch brought onboard and kept.
True Positive	The model correctly predicted the positive class (e.g, the model predicted a catch event).
Ultralytics	Developer and maintainer of YOLO-based object detection models, used in this project.
Vessel Discards	Catch brought onboard and then discarded.
Water Discards	Catch released from the water without being brought onboard.
YOLOv11-M / YOLOv11-L / YOLOv12	Variants of the YOLO object detection architecture.

Appendix B: development experiments

Throughout the project, several experimental features were tested to improve the accuracy or robustness of the AI-powered system. These trials helped refine the final system design, even though they were not carried forward into production due to performance trade-offs, deployment constraints, or added complexity. They are included here for transparency and to inform future work.

B.1 Tracking enhancements: re-identification

One experimental focus was improving fish tracking during brief occlusions in the EM footage. The team tested the new re-identification (Re-ID) module added to Ultralytics' BoT-SORT tracker. The goal was to preserve the tracking ID of a fish even when temporarily obscured, such as by splashes, crew members' hands, or overlapping motion.

Initial tuning involved adjusting proximity thresholds, appearance similarity, and track age parameters. While Re-ID occasionally succeeded in recovering interrupted tracks, it often introduced ID recycling, where a new catch was mistakenly assigned the ID of a previous one. These errors disrupted downstream counting and introduced uncertainty in catch event classification. As the overall benefit was limited and the risk to count integrity was high, this approach was not adopted.

B.2 Detector backbone variants

Another set of experiments explored larger detector architectures. The baseline YOLOv11-M model was replaced with:

- **YOLOv11-L**, a larger version of the same model family
- **YOLOv12**, a newer architecture that promised improved generalization

YOLOv11-L showed a modest improvement in performance but significantly increased prediction time, surpassing the near real-time processing window needed to handle daily footage on the edge device. YOLOv12, despite being a more recently developed model, delivered lower performance than YOLOv11-M and was even more computationally demanding.

Due to the lack of a meaningful performance/speed trade-off, neither model was incorporated into the production system, and YOLOv11-M remained the optimal choice for edge deployment.

B.3 Training strategy modifications

Several training variations were also tested to fine-tune performance of the YOLOv11-M baseline:

- **Frozen-backbone warmup:** The backbone was initially frozen to retain pre-trained features and gradually unfrozen across epochs. This approach underperformed compared to the standard end-to-end training regime and was dropped.
- **Increased resolution:** A higher input resolution was tested to improve fine-grained feature detection. While it produced a small boost in performance, it also increased prediction times, rendering it impractical for edge device deployment.
- **Rectangular training:** This method preserved the original aspect ratio of images rather than forcing them into square input shapes. While it better reflected the real-world layout of EM footage, it introduced inconsistent learning signals and yielded no measurable improvement in performance.

Appendix C: supporting analysis

This appendix presents two analyses that complement the main performance metrics described in the report. The first investigates how different volumes of annotated training data impacted model performance and labeling cost. The second compares AI-generated catch predictions with human-generated records, including expert-reviewed EM footage annotations and self-reported eLogs. Together, these analyses offer practical insights into data requirements, system limitations, and pathways for future optimization.

C.1 Impact of training data volume on model performance and labeling cost

This analysis explored how the amount of annotated training data, measured in hauling and catch events, affected the performance of the fish detection model, and how this performance scaled in relation to annotation costs. The goal is to provide practical guidance for teams planning to annotate EM footage to train their own AI-powered systems.

The experiment used a partially annotated dataset available at mid-project, before completion of the final training set. Five subsets were created by incrementally adding annotated hauling events: 2, 5, 6, 10, and 20 haulings, corresponding to 76, 156, 213, 324, and 554 annotated catch events, respectively. Each subset included all the data from smaller ones to simulate how annotation efforts might progress over time. The hauling events were selected to include a balanced representation of the most common species in the fishery.

The model was retrained from scratch on each subset using consistent training parameters, and evaluated on a fixed test set. Annotation costs were estimated based on commercial rates from CVAT labeling services.

Results, summarized in Table C.1, show that model performance improved as more data were added, particularly at the 5- and 10-hauling levels (around 150–300 catches). However, the improvement was not linear. Adding hauling events beyond this range resulted in smaller, more incremental gains, while annotation costs continued to rise. For instance, increasing the training data from 10 to 20 hauling events produced only modest performance improvements.

This pattern reflected diminishing returns, where additional data still improved results, but at a lower efficiency. In practical terms, this suggested that a relatively small, well-curated set of around ten hauling events (or ~300 catches) might be sufficient to train a model with strong performance.

This baseline could be particularly useful for smaller projects or new deployments, where annotation budgets are limited and full dataset labeling might not be feasible. However, the exact amount of annotated data required to reach strong performance might vary by fishery. Factors such as species diversity, visual conditions, and the quality and representativeness of the training set could all influence model learning. The values reported here served as a reference point based on the specific conditions and species composition of this project.

Table C.1. Data volume effect on model performance and labeling cost.

Data volume	Hauling events	Catches	Delta Precision	Delta Recall	Delta mAP@0.5	Delta mAP@0.5:0.95	Labeling cost
10%	2	76	+10.3%	+12.8%	+9.8%	+7.0%	$217
25%	5	156	+13.9%	+12.5%	+13.9%	+8.3%	$445
50%	6	213	+0.4%	+3.7%	+0.9%	+1.1%	$607
75%	10	324	+14.9%	+5.1%	+12.3%	+10.9%	$923
100%	20	554	+4.4%	+2.4%	+2.8%	+5.1%	$1,579

C.2 AI model prediction comparison against BV reports and eLogs

To assess real-world performance, AI-predicted catch counts were compared to two reference sources: expert-reviewed EM footage annotations from Bureau Veritas (used as ground-truth), and eLogs submitted by vessel captains. The comparison was conducted over five days of fishing activity.

As shown in Table C.2, predictions made directly on the vessel tended to undercount the total catch compared to the BV-reviewed annotations. The main cause was incomplete EM footage processing. On smaller vessels, power to the edge device is routinely shut off in the evening to conserve energy, often before all daily video has been analyzed. In the test cases, this led to 21%-45% of footage going unprocessed.

To confirm this explanation, the same EM footage was later reprocessed offline using the complete set of video files. This eliminated most of the discrepancy, demonstrating that the prediction gap was primarily due to limited processing time on the vessel, not issues with the model itself.

To reduce the risk of unprocessed EM footage in the future, several improvements are possible. The processing pipeline can be optimized by refining orchestration logic, reducing input/output bottlenecks, and simplifying rule-based logic in the fish counter. Currently, the largest bottleneck is model prediction time, which accounts for about 66% of the total processing duration. Using a faster model architecture or replacing the BoT-SORT tracker with a more efficient algorithm could reduce processing time and increase daily video coverage. These upgrades are especially relevant for smaller, less powerful vessels, where compute time is limited by energy availability.

Table C.2. Comparison of AI predictions, eLogs, and BV-verified ground-truth.

Date	eLogs	BV Report	Predictions on the vessel	Percentage of unprocessed EM footage	Predictions on 100% EM footage
2025-02-15	29	45	18	45%	33
2025-02-17	43	44	21	21%	28
2025-03-15	27	28	16	37%	20
2025-03-17	7	16	9	26%	17
2025-03-20	10	11	5	26%	13

Appendix D: how to apply the AI-powered system to your fishery

This project and its outcomes were designed to be reproducible and adaptable to other longline fisheries. By following the steps outlined below—together with the publicly available codebase and fish detector model weights—you can set up an AI-powered EM review workflow tailored to your own vessels, species, and regulations.

All source code, model weights, and deployment instructions will be progressively released as components are documented and adapted for broader use in TNC's GitHub repository.

Adopting the AI-powered system to your fishery can be broken into four main phases: (D.1) Gather EM data, (D.2) Annotate & prepare your dataset, (D.3) Train or fine-tune the fish detector model, and (D.4) deploy the AI-powered system on your vessel.

D.1 Gather EM data

To begin, you will need an EM system installed on your vessels. This typically involves working with an EM service provider to set up:

- **Video cameras** positioned to capture the entire catch-handling sequence, from the moment a fish is alongside the vessel in the water to when it is placed on the deck. The ideal setup (Figure D.1) uses a top-down, dual-view configuration covering both the waterline and the deck. This ensures every catch event is fully visible and provides the AI system with complete visual context for accurate detection and classification.
- **GPS tracking** to geolocate fishing events.
- **Onboard storage** sufficient to retain EM footage until it can be reviewed or uploaded.

If your fleet already uses EM, you can start collecting footage immediately. The goal is to progressively build a representative dataset of fishing operations for training and refining your fish detector model.

D.2 Annotate & prepare your dataset

AI models learn from labeled examples. To teach the model to recognize the species in your fishery, you will need to annotate your EM footage.

As demonstrated in this pilot, this can be done by combining expert-level fish identification with annotation support from a labeling service or people from your own team, using a free, open-source platform like CVAT.

The process generally involves:

- **Selecting representative footage** — include a variety of lighting conditions, catch compositions, and fishing scenarios.
- **Annotating each frame** — draw bounding boxes around each fish in the video frames and assign the correct species.

You don't need a huge dataset to begin. In this pilot project, strong performance was achieved with just ten hauling events (~300 individual catches) early in development, then improved as more data were added over time.

D.3 Train or fine-tune the fish detector

Rather than training an AI model from scratch, you can begin with the baseline fish detector provided in the TNC GitHub repository. This pre-trained model has already learned to detect fish in EM footage. Using a process called transfer learning, you can adapt it to your fishery by retraining it with your annotated dataset so that it recognizes your specific species and fishing conditions.

This step typically benefits from the support of a technology partner experienced in AI model training. They can ensure that the fine-tuned model is optimized for your fishery's unique vessel setup, camera angles, and species composition, helping you achieve strong performance more quickly and reliably.

D.4 Deploy the system on your vessel

Once your fine-tuned model is ready, the next step is to install and run the full AI-powered monitoring workflow on your vessel. The GitHub repository includes the guidelines and code needed to:

- **Run the model on an onboard edge device** (e.g., a NVIDIA Jetson or other edge device) to detect, classify fish species, and count catch in near-real-time as they cross a virtual counting line in the EM footage.
- **Generate Daily Reports** (see 5.3.1 Daily Report), which summarize fishing activity, risk profiles, and geolocated catch data, along with supporting evidence clips.
- **Transmit Daily Reports ashore automatically** at the end of each fishing day, enabling monitoring teams to review events without the delays of traditional EM data retrieval.

While the repository is well-documented, most fisheries will benefit from partnering with technical experts, either through their EM provider or a dedicated technology partner, particularly for:

- Installing and configuring the edge device on the vessel, including integration with EM cameras, GPS, and onboard networks.
- Adapting the codebase to match the fishery's specific camera angles, and reporting needs.
- Monitoring system performance and troubleshooting operational issues in the field.

PREPARED BY

tryo·labs

PREPARED FOR

The Nature Conservancy