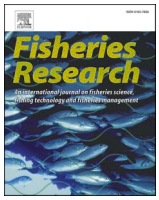
Fisheries Research 280 (2024) 107141

Contents lists available at ScienceDirect

Fisheries Research

journal homepage: www.elsevier.com/locate/fishres



Fishing event detection and species classification using computer vision and artificial intelligence for electronic monitoring

Muhammad Saqib [a,*], Muhammad Rizwan Khokher [a], Xin Yuan [a], Bo Yan [a], Douglas Bearham [b], Carlie Devine [b], Candice Untiedt [b], Toni Cannard [b], Kylie Maguire [b], Geoffrey N. Tuck [b], L. Rich Little [b], Dadong Wang [a]

[a] CSIRO Data61, Marsfield, NSW, Australia
[b] Environment, CSIRO, Hobart, TAS, Australia

ARTICLE INFO

Handled by Jie Cao

Keywords:
Fishing event detection
Fish species classification
Computer vision and deep learning
Electronic monitoring
Fisheries management

ABSTRACT

Fisheries regulations require detailed catch reporting on commercial fishing vessels. Vital components for the sustainable management of fish stocks include a robust estimate of the number of fish caught and the species composition. Catch recording is often done manually by human observers on fishing vessels. Human observers are costly, and consistent data streams can be subject to observer availability and the weather. On-vessel cameras (electronic monitoring, EM) are a growing alternative to human observers. However, on-land human auditors are required to review hundreds of hours of videos recorded during fishing trips that can last for weeks. In this paper, a framework is presented to automatically detect fish in EM videos, count the total fishing events, and classify the fish species. For this purpose, a deep learning and computer vision-based model is developed to efficiently detect fish and fishers onboard a vessel. Secondly, a vision-based tracking pipeline tracks the detected fish and counts the total fishing events in the videos. Thirdly, the extracted fishing events are classified through a deep learning-based fish species classifier, to provide the distribution of different fish species caught for a fishing trip. For our experiments, the datasets were prepared using the electronic monitoring data of multiple fishing trips of a fishing vessel. The videos were recorded on Australian longline vessels targeting tunas and billfish. For the fish detection task, video frames were extracted and labelled manually to provide a digital ground-truth. For the fish species classification task, hundreds of fish images of multiple species were cropped to provide a training dataset for the fish classifier. For the fish counting task, manual counts for the fishing events of individual fish species were generated for the test fishing trips. The developed fish and fisher detector achieves a mean Average Precision of 87.0 % for fish and 94.0 % for fishers on test video frames. The fishing event detection pipeline achieves an Average Precision of 81.0 % and an Average Recall of 74.5 % on test videos. The fish species classifier achieves an Accuracy (Top-1) of 91.11 % for the classification of cropped fish images and 89.05 % for the classification of extracted fishing events from the videos. Experimental results show that our proposed computer vision and artificial intelligence-based solution for video analysis has great potential to automate the auditing process from electronic monitoring footage and contribute to the sustainable management of fish stocks.

1. Introduction

The sustainability of harvested fish stocks and associated industries is of prime importance for fishery regulators. Licensing and quota management have been adopted by management authorities to address the challenges of optimising industry profitability, reducing over-fishing, and ensuring ecosystem impacts are acceptable. Observer programs assist these objectives by collecting independent data on target and non-target catches. However, on-vessel human observer programs are expensive and as a consequence, only a fraction of vessel trips may be covered (or none at all) (Benoiˆt and Allard, 2009; Depestele et al., 2011; Poos et al., 2013). The use of electronic monitoring (EM) camera systems is becoming increasingly common for the recording of catch, bycatch, and other fishing activities, encompassing tasks such as catch composition counts, catch handling procedures, and fishing method compliance. Monitoring such activities is of critical significance for

* Corresponding author.
E-mail address: muhammad.saqib@csiro.au (M. Saqib).

https://doi.org/10.1016/j.fishres.2024.107141
Received 29 January 2024; Received in revised form 20 June 2024; Accepted 7 August 2024
Available online 4 September 2024

operational adherence to licensing permits, stock management, and assessing the broader ecosystem impact of fishing van Helmond et al. (2020). However, the process of manually reviewing and analysing EM data can be time-consuming, costly, prone to errors, and challenging when dealing with large amounts of data captured over prolonged periods. Consequently, in Australia's Commonwealth-managed fisheries, a minimum of 10 % of video data is reviewed by experts Qiao et al. (2021), even though footage may exist of all trips.

Recent advancements in deep learning and computer vision-based methods have shown great potential for monitoring applications in fishing (Khokher et al., 2022; Qiao et al., 2021; Probst, 2020). For example, Salman et al. (2020) developed a method that used the combined outputs from Gaussian mixture models and an optical flow algorithm as inputs for a Convolutional Neural Network (CNN).

Similarly, underwater video analysis can be carried out using a Region-based Convolutional Network (R-CNN). Miranda and Romero Miranda and Romero (2017) proposed a fish detection-based approach in a constrained environment with a fixed background. French et al. (2020) proposed a segmentation-based system for identifying fish species on trawler conveyer belts using Mask R-CNN (an image segmentation network). Additionally, Palmer et al. (2022) used Mask R-CNN with a statistical model to automatically estimate the number and mean fork length of dolphinfish from landings on the vessel. The system deployed uses images for fish detection and statistical models to correct for biases introduced by undetected fish. Furthermore, in a feasibility study, Monkman *et al.* Monkman et al. (2019) explored the use of machine vision to automate the identification and size estimation of fish using Region-based Convolutional Neural Networks.

This paper explores deep learning techniques, specifically YOLOX model Ge et al. (2021), to detect fishing events and track activities in video footage captured by EM systems on longline commercial fishing vessels. To address the issue of imbalanced data for certain fish species, we approached fishing event detection as a two-class detection problem, focusing solely on annotating fish and fishers during the detection stage. Additionally, this approach ensures that the model can easily adapt to new fish species by training the classification part of the pipeline to accommodate any future changes, thereby enhancing its generalizability. The ultimate goal is to identify and store only those video segments containing "catch events", which would reduce data storage requirements and increase the percentage of video footage containing activities of interest to fisheries managers. This will markedly improve the efficiency of data analysis. Additionally, this study examines the use of ConvNext (a Convolutional Neural Network architecture) for fish species classification. By leveraging these AI technologies, this research has the potential to enhance the utilization of EM data for fisheries management and improve profitability and sustainability in the fishing industry Xie et al. (2017).

1.1. Related work

Ensuring that the impacts on related bycatch species and habitats are minimal has led to increased attention to regulations that promote sustainable fishing practices. Implementing EM as a means of monitoring fishing activities and compliance with regulations began with a pilot trial in British Columbia, where 50 vessels were equipped with EM systems, and a 36,000 trap limit was implemented fleet-wide van Helmond et al. (2020). Previously, in 2002, Alaskan longline fisheries began using EM to test for compliance with regulations on seabird bycatch (Ames et al., 2005; McElderry, 2004). Furthermore, a larger EM program was introduced in British Columbia, Canada, involving 200 vessels to monitor compliance. Similarly, New Zealand employed EM to monitor seabird interactions in a gillnet fishery McElderry et al. (2007). In Australia, EM trials were initiated in 2015 to monitor catch-in-line fisheries Emery et al. (2019). Despite EM's potential for cost-efficient monitoring of catch and compliance, its widespread deployment for this purpose has yet to be fully realized, due to the need for human video footage to be reviewed. However, with the advancements in deep learning and computer vision, there is now a unique opportunity to utilize onboard video recordings to analyse catch and bycatch, to promote sustainable fishing practices (Tseng and Kuo, 2020; Wu et al., 2023). Another recent study used deep learning models such as YOLOv4 and Mask RCNN to detect and classify fish in high-quality visual and acoustic data; however, this study did not use data captured onboard vessels for event detection Kandimalla et al. (2022).

The authors of Vilas et al. (2020) developed an electronic device known as an observer that includes a camera and computer module for automatic fish detection and identification onboard the vessel. The device is installed on a conveyor belt before species are sorted. Additionally, the device is equipped with onboard sensors and magnets to detect the movement of the conveyor belt. The camera module uses image processing to detect fish species and estimate their length and weight without the involvement of fishers. The image analysis data is combined with vessel meta-data such as location and velocity and transmitted to a shore-based data centre. Another study Marini et al. (2018) developed a video-based automated procedure using genetic programming for image analysis to effectively track and estimate the numbers of fish from underwater video cameras without separating fish from different classes. The system processed 20,000 images acquired in real-world coastal settings to capture the temporal dynamics of fish abundance. The automatic counting results were highly correlated with manual counts for different fish species. Several recent studies have focused on video-based hierarchical species classification that can predict coarse-level groups of fish species and fine-level species at the same time. The architecture allows the coarse-level prediction to be the final output if the fine-level confidence score is too low and improves accuracy on tail-class species when training data follows a long-tail (imbalanced) distribution Mei et al. (2021b). Furthermore, in another similar study, the Hierarchical Class Incremental Learning (HCIL) approach is designed to provide both coarse-level and fine-level species predictions concurrently, with the added advantage of the system gradually incorporating an increasing number of training classes for fish over time Mei et al. (2022a). Other relevant studies focused on estimating the 3D position and size of the fish from a single camera under occluded scenarios. Unlike other approaches that require costly data or multiple camera angles, the proposed technique is based on a single image Mei et al. (2021a). In similar other work, a template-based method is used to infer 3D shapes from a single-view image and apply the reconstructed mesh to a downstream task, i.e., the absolute length of fish, in an unsupervised manner Mei et al. (2022b). An unsupervised domain adaptation architecture is proposed, where a teacher network is used to enhance pseudo-label accuracy using progressive mixup augmentation for intermediate sample generation between source and target domains. The intermediate samples are used for fish classification under various domains Zheng et al. (2023). Previous studies have primarily focused on counting and characterizing catch and bycatch within a controlled environment.

Our study introduces several advancements that distinguish it from previous related work Qiao et al. (2021), enhancing the effectiveness and applicability of deep learning models in real-time applications. The YOLOX model used in our study has been optimized for real-time applications. We have focused on reducing computational complexity while maintaining high accuracy, making our model particularly well-suited for deployment in scenarios where processing speed is critical. Besides, we have incorporated a robust tracking model tuned to recover lost tracks and create more stable and reliable detections before classification. Unlike the previous studies that might have used more

conventional convolutional networks, our work utilizes the ConvNeXt model to classify the detected bounding boxes. This choice was motivated by ConvNeXt's improved efficiency and accuracy in handling complex image features. Finally, our methodology involves a rigorous evaluation method where the model is trained on data from three trips and tested on the fourth trip of the same vessel. This four-fold cross--validation approach ensures that our findings are robust and generalizable across different trips and conditions, providing a more reliable assessment of the model's performance than single-trip studies. The summary of different techniques and their evaluation on different datasets is provided in Table 1. However, our proposed system aims to detect, identify, and track fish within an unconstrained environment. This poses significant challenges, such as frequent occlusions, motion blur, changing illuminations (day and night), water drops on the camera lens, poor lighting conditions, variations in weather conditions, and complex and cluttered backgrounds.

1.2. Contributions

The following are the main contributions of this paper:

1. A robust fish and fisher detector is developed based on Darknet and YOLOX deep learning architectures Ge et al. (2021). The developed detector can handle complex lighting and weather variations, e.g., day and night, exposure to the sun, and sunny, cloudy, and rainy weather conditions.
2. A tracking pipeline is developed based on the Kalman filter and Hungarian algorithm to efficiently track the detected fish and extract the fishing events with the start and end times in videos. The tracking pipeline ensures no duplicate counting of the same fish, which can happen when images are used instead of video sequences.
3. A multi-class fish species classifier is developed based on ConvNeXt deep learning architecture Liu et al. (2022) to classify fish into different species for the extracted fishing events.

2. Materials and methods

In this section, we propose a robust framework for the detection, tracking, and classification of fish. Our solution's overall architecture is composed of three main modules. First, the *detection module* is responsible for identifying and detecting the fish in video frames in complex environments. Second, the *tracking module* ensures that previously-detected fish continue to be tracked after they are moved from the water to the sea door of a fishing vessel until they are landed in the fish processing area. Third, the *classification module* is responsible for classifying the extracted fish events and categorising the fish into different species. The design and functionality of these modules are further described in the following subsections.

2.1. Fish detection

Most current state-of-the-art object detectors, such as YOLOX Ge et al. (2021), take inspiration from the YOLOv3 Redmon and Farhadi (2018) architecture. The YOLOv3 algorithm is similar to YOLOv1 but uses a stronger feature extractor backbone called Darknet-53, also used in YOLOX model. An input image is passed through Darknet-53 for feature extraction, and then a specialized architecture, known as the head, is used to make predictions. The architecture also incorporates a Feature Pyramid Network (FPN) to extract features from images at different scales and aspect ratios.

Furthermore, the YOLOX model proposed a decoupled head design that significantly improves the original YOLOv3 architecture. Instead of a single output, the YOLOX model has three different outputs from each head. The YOLOv3 uses anchor boxes for object detection and predefined bounding box shapes that the model uses to predict the offset from these anchors. In contrast, YOLOX uses the Fully Convolutional One-stage (FCOS) Tian et al. (2019) object detection technique to split the image into grids based on three scales. The model assigns predictions to each intersection point on the grid, called anchor points and acts as an offset to move the projections.

The YOLOX also uses a dynamic label assignment approach called Simplified Optimal Transport Assignment strategy (SimOTA) to eliminate bad predictions by separating the predictions that match the ground truth as positive and those that match the background as negative during training. There are three outputs of the YOLOX model, each with its loss function to optimize it.

2.1.1. Class optimization loss

The first loss function in our proposed framework is focused on class optimization. The YOLOX model's output feature map has the shape of

Table 1
A summary of different computer vision and artificial intelligence approaches for fish detection and tracking along with datasets used, camera setups, and No. of species classified.

Method	Year	Detection and Segmentation	Tracking	Dataset and camera setup	No. of species
Vishnu *et al.* Kandimalla et al. (2022)	2022	YOLOv3 and Mask-RCNN	Norfair Alori et al. (2023)	DIDSON McCann et al. (2018) high resolution visual acoustic data RoI overlapped by optical camera and visual sonar camera	8
Palmer et al. (2022)	2022	Mask-RCNN for segmentation	–	Dataset curated for dolphins for length and weight estimation	–
Khokher et al. (2022)	2022	ResNext with Cascaded RCNN	IoU based tracking	Private dataset obtained from commercial vessels	5
Vilas et al. (2020)	2020	Background removal/morphological operations for segmentation, and color, shape, and texture for identification	–	Dataset curated from camera installed on conveyor belt and RoI defined on conveyor belt	18
Salman et al. (2020)	2020	GMM, Optical flow and Raw image as input to ResNet−152 for detection	–	Fish4Knowledge with Complex Scenes underwater dataset Fisher et al. (2016) and LifeCLEF 2015 Fish Joly et al. (2015) dataset - publicly available	15
Qiao et al. (2021)	2020	YOLOv5 network for detection and classification	Hungarian algorithm for tracking	Private dataset obtained from commercial vessels for class of fish and fisher	2
French et al. (2020)	2019	ResNet−50 with Mask RCNN for fish detection and classification	–	Private dataset-video obtained from commercial vessels for annotation on conveyor belt	6
Monkman et al. (2019)	2019	ResNet−101 with RCNN for fish detection for fish length measurements	–	Dataset curated from various public datasets	–

$H \times W \times C$, where H, W, and C represent height, width, and the number of classes, respectively. Each element in the C vector represents the model's confidence in predicting a specific class. During the optimization process, the YOLOX model employs a one-hot encoded representation for each ground-truth bounding box. This approach can predict a distribution of all classes rather than just a single class. The Binary Cross Entropy (BCE) loss function is employed to optimize the class predictions when applied to both the predictions and ground truth. Negative predictions are not utilized in this loss function.

2.1.2. Regression loss

The YOLOX model employs the Intersection-over-Union (IoU) loss function to compare the predicted bounding box with the ground-truth box. The IoU metric ranges from 0 to 1, where a higher value indicates a better match between the predicted and ground-truth boxes.

$$0 \leq \text{IoU} \leq 1 \tag{1}$$

The goal is to maximize the IoU metric to ensure the predicted box matches the ground-truth as closely as possible. To optimize the model by minimizing the IoU loss, the sum over all positive predictions is calculated and minimized.

2.1.3. Objectness loss

The loss term for objectness optimization aims to determine the probability of an object existing within the bounding box, with a score of 1 indicating that the model is certain of the presence of an object. Similar to the class loss optimization, the objectness score uses the Binary Cross Entropy (BCE) loss function to optimize. The YOLOX model employs SimOTA to assign positive labels to the ground truth, calculate the IoU and objectness score, and incorporate them into the BCE loss function for optimization. However, negative predictions that do not match the ground truth are optimized differently by assigning them an IoU value before being input into the BCE loss function.

The final loss is the sum of all loss terms for positive labels, as shown below:

$$l_{total} = \frac{l_{cls}}{N_{pos}} + weight_{reg} \times \frac{l_{reg}}{N_{pos}} + \frac{l_{obj}}{N_{pos}} \tag{2}$$

Where l_{total}, N_{pos}, l_{cls}, l_{reg}, l_{obj} represent the total loss, the number of positive predictions, classification loss, regression loss, and objectness loss, respectively. The term $weight_{reg}$ is a trade-off parameter that emphasizes the importance of the regression loss over the other loss terms.

2.2. Fish tracking and catch event detection

Fish tracking enables the framework to match previous detections with current ones to improve the performance and speed of the detection and classification processes. Since we are concerned with detecting the presence of fish within the Region Of Interest (ROI), we only focus on tracking the fish within the ROI. A block diagram of the tracking pipeline is shown in Fig. 1.

The tracking step begins with calculating the centre point and the size of each detection and then checking whether these detections are within the ROI or not. After that, these detections are matched with existing trajectories concerning the distances between their centre points. Meanwhile, trajectories that cannot be matched to the latest detections are removed since they either represent noisy detections or objects stationary for a long time. When matching is complete, each moving object's trajectory is independently tracked: If the direction of an object changes too frequently, it is removed from the matching process as noise with an adequate probability of occurring. As it is more important not to miss moving objects than to eliminate noise, we set a threshold to ensure that radically different noise sources are eliminated: the trajectory of an object is removed if it changes direction at least twice in three consecutive frames on average. The Kalman filter Kalman (1960) is employed to track moving objects after excluding abnormal detections. The Kalman filter is a useful tracking tool when objects disappear or temporarily stop. The framework keeps track of existing moving objects for ten seconds and labels them passive if they do not appear again.

Based on frame-by-frame detection results, we need to associate detection with the same object, i.e., tracking. We first use a Kalman Filter to predict new detections in consecutive frames and associate those predictions to the track's location in each frame. Then, we assign detections to tracks in the process of tracking multi-objects using James Munkers's variant of the Hungarian assignment algorithm Kuhn (1955), to ensure the matching score is maximized. Finally, we obtain the tracking results (i.e., the trajectories) for both fish and fishers. The detailed tracking process of fish and fishers is summarized in Algorithm 1. We define a fishing event as fish and fishers detected in the same frame. Meanwhile, the targeted objects (fish and fishers) are detected from consecutive video frames and last for a certain length (we only consider the fishing events that occurred on the Cutting Deck camera view, each with at least a fish and a fisher). Based on the obtained trajectories using the Hungarian assignment algorithm, we further define that a fishing detection event happens when the trajectory length of the

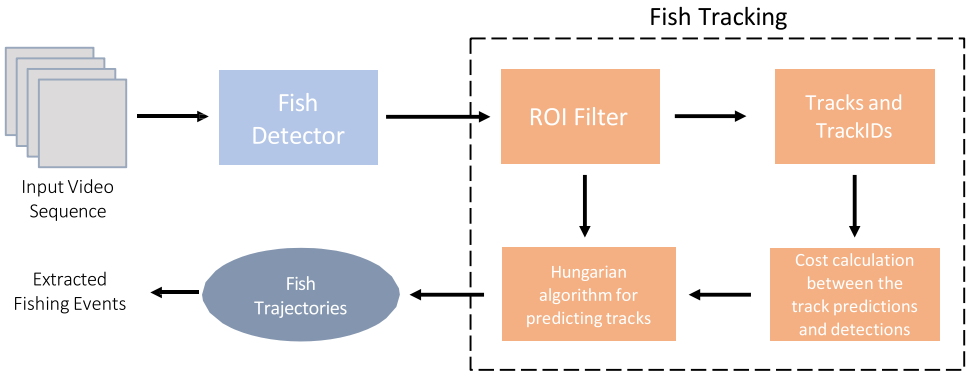


Fig. 1. A block diagram of the tracking pipeline.

fish going through an ROI on the cutting deck is larger than a specified threshold. Here, the threshold is the number of consecutive frames, and its value depends on the distance between the sea door and the fish processing area of a fishing vessel.

Input: Videos with T frames.
Output: Fish trajectories and *count*.
Initialize the detections, trajectories, and counts for the first frame, i.e., $t = 0$, denoted by $DET(0)$, $Traj(0)$, and $count = 0$, $Traj\_live\_flag = 0$, and a number of unmatched frames $len\_unm = 0$.
for $t = 1, \cdots, T$ **do**
  Detect locations of the fishes and fishers by the YOLOX model for the t-th frame, denoted by $DET(t)$.
  if $length(DET(t)) = 0$ **then**
    if $Traj\_live\_flag = 0$ **then**
      $len\_unm = len\_unm + 1$.
      if $len\_unm > FORW\_THRE$ **then**
        $Traj\_live\_flag = 0$.
      else
        $Traj\_live\_flag = 1$.
      end if
      Update $Traj(t)$ by copying the previously detections $DET(t-1)$.
    end if
  else if $length(Traj(t-1)) > 0$ **then**
    Calculate the cost matrix using the cosine distance between the previous trajectory $Traj(t-1)$ and the current detection $DET(t)$, denoted by $COST(t)$.
    if $COST(t) \neq 0$ **then**
      Obtain the current association results of matching cascade using the Hungarian algorithm based on the cost matrix, i.e., $match(t)$;
      Update the current and previously unmatched trajectory indexes, $unmatch(t)$ and $unmatch(t-1)$.
    end if
  else
    Initialize the trajectory $Traj(t)$.
  end if
  for idx in $match(t)$ **do**
    Update the trajectory $Traj(t)$ by appending the current matching detections.
    $length(Traj(t)) = length(Traj(t-1)) + 1$.
  end for
  for idx in $match(t-1)$ **do**
    Update the previously unmatched detections by copying the previous trajectory $Traj(t-1)$.
    $length(Traj(t)) = length(Traj(t-1)) + 1$.
    if $len\_unm > FORW\_THRE$ **then**
      $Traj\_live\_flag = 0$.
    else
      $Traj\_live\_flag = 1$.
      $len\_unm = len\_unm + 1$.
      Update the trajectory $Traj(t)$ by appending the previously unmatched detections.
    end if
  end for
  for idx in $unmatch(t)$ **do**
    Update the trajectory $Traj(t)$ by appending the currently unmatched detections.
    $count = count + 1$.
  end for
end for

2.3. Fish species classification

In this section, the fish species classification model is presented. From the event detection segment of the overall pipeline, the detected fish images for a fishing event or trajectory are automatically cropped from the video frames. The cropped images are then used to classify them into different fish species. The training and inference modules for fish classification are shown in Fig. 2. During the training process, the input (training) fish images are augmented or pre-processed first. The image pre-processing includes resizing, flipping, rotation, blur, noise, affine transformation etc. The purpose of the image augmentation step is to increase the number of training images to capture more variation that may be present in the test images. A deep learning-based image classifier is trained to learn the image features and classify images into different fish species. During inference/testing, the test images are resized and classified through the trained image classifier.

The fish species classifier is built using a deep learning framework called ConvNeXt Liu et al. (2022). It takes advantage of both ResNet and Transformer CNNs. It gradually 'modernizes' a standard ResNet architecture towards the design of a vision-transformer deep learning architecture and discovers several key components that contribute to the performance difference. The outcome of this exploration is a family of pure ConvNet models dubbed ConvNeXt. Constructed entirely from standard ConvNet modules, ConvNeXts compete favourably with transformers regarding accuracy and scalability. For more details, refer to Liu et al. (2022).

For a fishing trip, there exist many videos containing fishing events with trajectories of fish as they are processed. Each trajectory can spread over frames ranging from 10 s to 100 s. The bounding-boxes for each trajectory are automatically cropped from the videos and classified. The following filters are then applied to the classification results of each trajectory.

i. In a trajectory, two types of bounding-boxes are present: those coming from the fish detector and those resulting from the tracker predictions. Sometimes the tracker predicted bounding-boxes are empty with no fish. The first filter discards such bounding-boxes.
ii. The classification score is used as a threshold to discard classified bounding-boxes with low confidence. The classification score threshold is set to 0.5, which retains the bounding-boxes with confidence 50 % or above.
iii. The fish class appearing most in the remaining trajectory images is selected as the final class for that trajectory.
iv. If most of the trajectory is classified as *unknown*, which can happen if the video quality is poor due to blur, fog, water drops on the camera lens, poor lighting etc., the second-highest occurrence of a fish class is selected as the final class. If the whole trajectory is classified as *unknown*, then the final class remains *unknown*.

2.4. Dataset

In this study, EM videos from four trips of a commercial fishing vessel were analyzed. The annotations, including images extracted and labelled with the species name, were completed for all the trips. The problem was formulated as event detection with two objects in the event, fish and fisher in the same frame. Therefore, the detected objects were tagged as either fish or fisher. The following section will provide more details of the annotations.

2.4.1. Dataset for fish detection

Many commercial fishing vessels take extended trips that can last for weeks. For this study, a dataset was created from videos recorded by a camera focused on a fishing vessel's deck area and sea door. These videos and corresponding ground-truth annotations provided by domain experts were made available by the Australian Fisheries Management

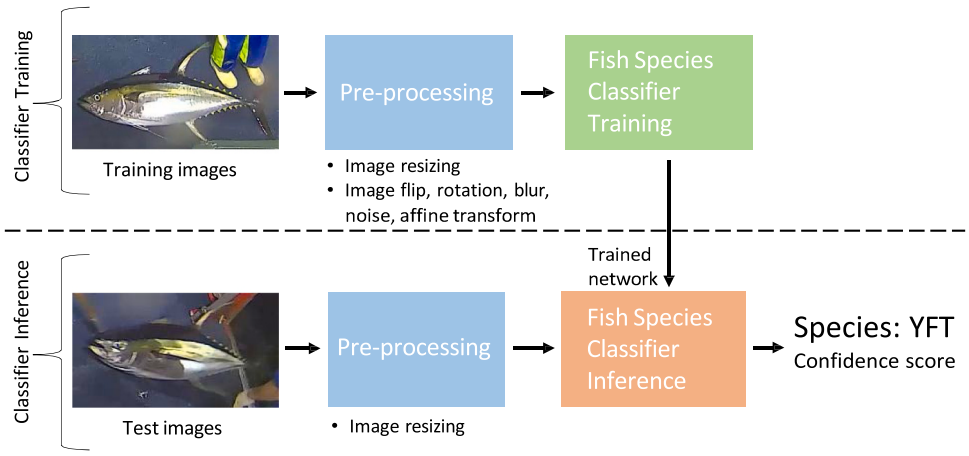


Fig. 2. The training and inference pipeline for fish species classification (YFT: Yellowfin tuna).

Authority (AFMA). EM trials began in the Australian Eastern Tuna and Billfish Fishery (ETBF) in 2010 and in 2015 EM was introduced to replace human observers. The video segments are in MP4 format with a resolution of 1280 × 720 pixels and a frame rate of 10fps. Most video segments have a duration of either 30 or 60 min. However, many of these video segments do not contain any catch events. Therefore, as the first step in the video selection process, only videos with catch events were selected. In the second step, 30–40 s of video clips were extracted based on timestamp information for all the species caught on the vessel.

We utilized the Computer Vision Annotations Tool (CVAT) CVAT.ai, Corporation (2022) for annotations in the Pascal-VOC Everingham et al. (2010) and COCO format Lin et al. (2014). A total of 5198 images were annotated from four trips of a commercial fishing vessel, with 20,019 annotations for fish and 19,274 for fishers, respectively, as shown in Table 2. The annotations in a video frame can be quite dense in some cases, with as many as 45 annotations per frame, depicting the cluttered scenario of many fish stacked on the deck. The objects in the dataset are divided into small, medium, and large based on the area covered by the object bounding boxes. Small objects cover an area of fewer than 32 × 32 pixels, medium objects cover an area of less than 96 × 96 pixels and greater than 32 × 32 pixels, and large objects cover an area of more than 96 × 96 pixels. The ground-truth analysis of our dataset shows that most of the objects characterized are either large or medium based on the COCO analysis as shown in Fig. 3.

The fish and fishers are annotated in the video segment of every catch event. The ground-truth analysis of our dataset identified that the dataset is highly imbalanced for some target species. We formulated and changed the dataset into two categories to avoid the detector being biased towards the majority class's species. The main reason for this decision is to classify correctly, not miss any catch event, and balance the dataset for the two-class classification problem (fish and fisher).

The distribution of video frames across different weather conditions in both the training and testing splits is comprehensively detailed in Table 3. For the training split, sunny or clear sky conditions comprise 35.0 % of the frames during the day and 15.0 % at night. Rainy conditions account for 17.5 % during the day and 7.5 % at night, while cloudy conditions represent 19.5 % during the day and 5.5 % at night. The testing split, designed to evaluate the model's performance under various conditions, includes 24.0 % sunny frames during the day and 16.0 % clear sky frames at night, 18.0 % rainy frames during the day and 12.0 % at night, and 20.0 % cloudy frames during the day and 10.0 % at night. This balanced distribution across different weather conditions and times of day ensures that the model is both trained and tested on a diverse set of environmental scenarios, enhancing its robustness and generalizability.

Table 2
Details of the dataset used for the fish detection task, including the number of images labelled, total annotations of fish and fishers, minimum and maximum annotations per image, and the number of images used for training and testing.

Trip #	Total frames	Annotations per class		Annotations per image		Dataset splits	
		Fish	Fishers	Min	Max	Training	Testing
1	403	394	403	2	9	4795	403
2	2114	3068	8167	2	9	3084	2114
3	1157	6730	4572	3	45	4041	1157
4	1524	9827	6312	1	28	3674	1524

2.4.2. Dataset for fish species classification

The fish detection dataset described in the previous section was refined for species classification. The fish instances were labelled using bounding boxes. Those bounding boxes were automatically cropped from the video frames and organised for the following fish species: Albacore (ALB), Bigeye Tuna (BET), Escolar (ESC), Mahi Mahi (MAH), Ray's Bream (RBM), Southern Bluefin Tuna (SBT), Shark (SHK), Skipjack Tuna (SJT), Striped Marlin (SMN), Swordfish (SWD), Unknown (UKN) and Yellowfin Tuna (YFT). From the fish species distributions in the prepared dataset, there is a high imbalance in the frequencies of certain classes. The species like *Blue Shark* and *Hammerhead Shark* were merged into one category as SHK. Sample cropped images of the above fish species are given in Table 4.

More than 30,000 cropped images were extracted for the fishing trips. The images were checked manually to fix the following issues. Firstly, the annotations were done in consecutive video frames, resulting in many redundant images. An image should not be repeated in the training dataset to avoid class biases. Secondly, due to human error, some images were annotated incorrectly during labelling; such images should not be used. Thirdly, due to noise, water droplets on the camera lens, and fog, many images were not identifiable by a human observer. Such images were also removed. Lastly, as the event detection is performed within a predefined ROI, the images outside the ROI, where the fish were stacked together or bagged, were removed.

After applying the above filters, we reduced the number of usable images to between 28 and 520 for each species, given in Table 4. To fix the problem of class imbalance and increase the number of samples for the underrepresented classes the following image augmentations, i.e., horizontal flip, vertical flip, rotation 90˚, rotation 270˚, Gaussian noise, motion blur and elastic transformation, were applied so that each class has an equal number of images (same as YFT in our case, i.e., 520). The above image augmentations can be visualised in Fig. 4. After manually selecting images and performing image augmentations, all the images were unique, and the number of instances was the same for each fish species, i.e., 520. Although we could do it through the data loader, we

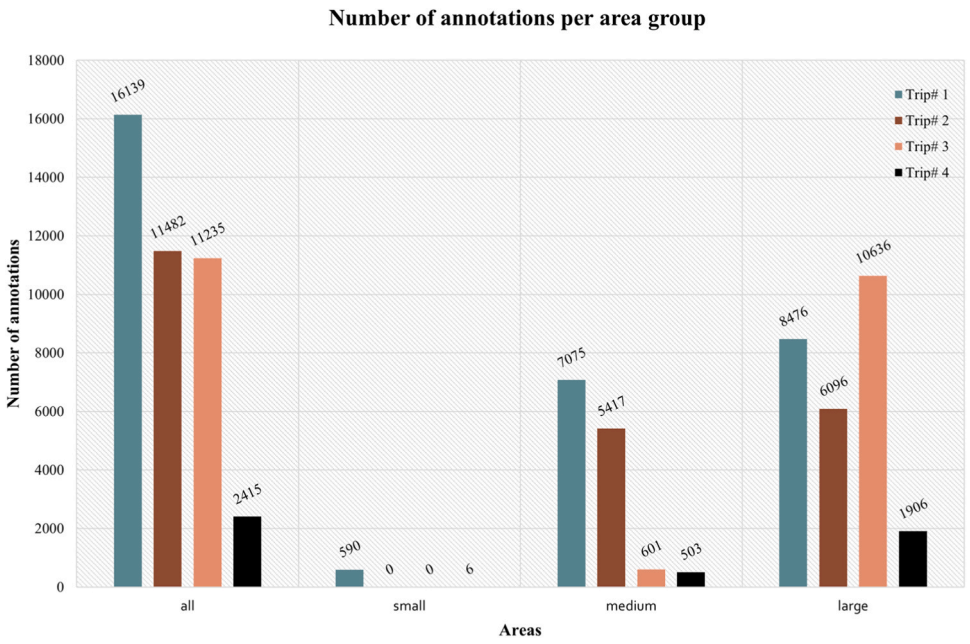


Fig. 3. The graph shows a ground-truth analysis conducted on bounding box areas. Each bar in the graph represents the count of annotations for four specific trips (Trips # 1, 2, 3, and 4). The annotations are further categorized into three distinct groups, namely small, medium, and large, based on the respective bounding box areas.

Table 3
Distribution of video frames across different weather conditions and times of day in the training and testing splits, demonstrating a balanced approach to ensure model robustness and generalizability.

Condition	Training Split (%)		Testing Split (%)	
	Day	Night	Day	Night
Clear/Sunny	35.0 %	15.0 %	24.0 %	16.0 %
Rainy	17.5 %	7.5 %	18.0 %	12.0 %
Cloudy	19.5 %	5.5 %	20.0 %	10.0 %

chose to do this manually so that there is no ambiguity left in the dataset quality.

2.5. Experimental setup

Several experiments were conducted to fine-tune the different modules in the proposed framework. The Fishing Event Detection and Classification modules are validated separately to facilitate analysis and improvement. The framework is optimized by selecting the best performance on each module based on the experimental results.

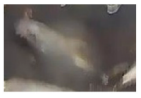
2.5.1. Fishing event detection

We use 140 video segments, in.MP4 format, with a resolution of 1280 × 720 pixels and a frame rate of 10fps, provided by the AFMA. All these segments are captured by a camera covering the Deck area of the commercial fishing vessel over four trips. Each segment lasts 30 or 60 min and is reviewed and annotated by an EM analyst. The annotation contained details from each fish captured, including date, time, location, and species. We prepare a dataset that uses videos from three trips for training and the videos from the remaining trip for testing. To achieve a stable result, the leave-one-out cross-validation (LOOCV) method is adopted. We split the videos from three trips for training and videos from the remaining trip for testing and repeated this four times.

2.5.2. Fish species classification

The dataset was divided into training (80 %) and testing (20 %) sets randomly for each class. The image classifier was implemented using a

Table 4
Number of selected images for different fish species.

Species	Sample image	No. of images	Species	Sample image	No. of images
ALB		297	BET		66
ESC		98	MAH		110
RBM		35	SBT		236
SHK		28	SJT		104
SMN		56	SWD		98
UKN		360	YFT		520

toolbox called -'MMclassification' from the OpenMMLab MMClassification, Contributors (2020). The toolbox uses PyTorch, MMCV, Python, and many other libraries to implement deep learning-based classification models. The ConvNeXt-based fish classifier was trained on our image dataset for 300 epochs. The ConvNeXt architecture makes use of layer normalization instead of batch normalization, GeLU activation function instead of ReLU, and the 'AdamW' optimizer from the

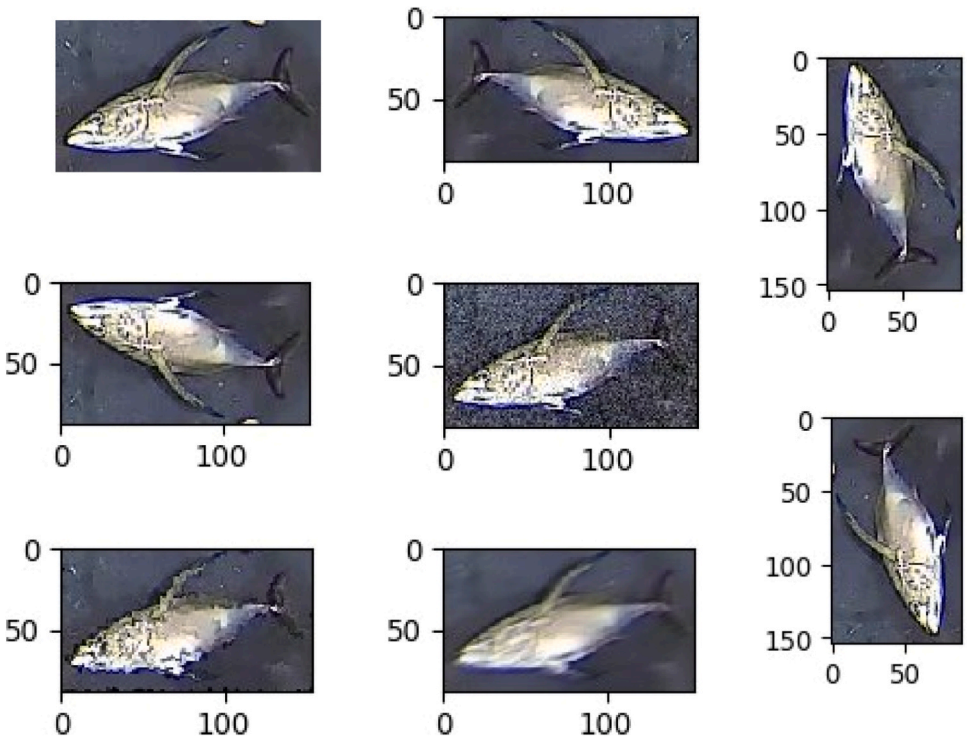


Fig. 4. Distinct types of image augmentations applied to make the number of images the same for each fish species.

vision-transformers, which makes it quite robust in extracting deep features from images. The input images were resized to a fixed size of 128 × 128 pixels. The batch size of 16 was used for a single GPU for training. The training took nearly two hours, whereas the testing of 20 % images took around 3 s

2.6. Evaluation metrics

2.6.1. Fishing event detection

For the fishing event detection, metrics like Precision and Recall are used to evaluate the performance of our algorithms, we define Precision and Recall, p and r, as follows:

$$p = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

$$r = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4}$$

where TP denotes the *true positive*, which is the number of the correctly detected fishing events within the ROI; FP denotes the *false positive*, which is the number of the incorrectly detected fishing events within the ROI; and FN denotes the *false negative* which is the number of the missing fishing events not being detected within the ROI. Precision provides a percentage that shows how accurately the detection network detects the targets, and Recall provides a percentage that shows how many actual targets are detected out of the total number of targets.

2.6.2. Fish species classification

Image classification is widely performed across many vision-based applications. The same evaluation metrics as those used in most image classification applications were adapted to evaluate our fish classifier. The evaluation measures, like Precision, Recall, Accuracy, F1-score, and confusion matrix, were used for the evaluation of the classifier. Precision and Recall are given in Eq. (3) and Eq. (4), respectively. F1-score is calculated as:

$$\text{F1 - score} = 2 \times \left(\frac{p \times r}{p + r}\right) \tag{5}$$

where p and r represent Precision and Recall, respectively.

Accuracy gives a percentage of how many images were classified correctly out of the total number of images. Finally, a confusion matrix is used to evaluate the performance of individual classes in terms of Accuracy, e.g., whether the Accuracy is affected by other classes due to inter or intra-class similarities or not.

3. Results and discussion

In this section, we present the details about how the inflorescence detector is trained, followed by the analysis of both visual and quantitative results, including average precision, recall, and F1-score. The tracking and counting results for the test dataset are also discussed. The overall and panel-wise inflorescence counts are presented and discussed. In addition, the counting results are quantitatively evaluated using MAE, RMSE, and R^2 measures when compared to the ground-truth. Based on the automatic counting results for the test videos, an estimate of the yield is provided and compared with the actual yield after harvest.

3.1. Fish detection results

To evaluate the performance of our detector, we employed the COCO evaluator metrics Lin et al. (2014). The COCO evaluator utilizes a range of IoU thresholds, with ten threshold values, to calculate Average Precision and Recall values. Additionally, the COCO evaluator calculates the Average Precision across different scales, including small, medium, and large objects.

Table 5
The experimental settings for all the detection experiments.

Model	Batch size	Weight decay	Momentum	No. of epochs
YOLOX	4	0.9	$5e^{-4}$	300

To train our model, we utilized transfer learning from pre-trained weights on a COCO dataset and trained the YOLOX model for 300 epochs with the experimental settings outlined in Table 5. We employed a k-fold cross-validation strategy with k=3, training the model on three trips of vessel data and validating on the fourth trip.

Our results demonstrate that the model performs well when evaluated on Trips # 1, 3, and 4 in Fig. 5. However, a drop in Recall value is observed for Trip # 2, likely attributed to poor data quality and many occlusions. Table 6 presents a comprehensive analysis of the mean Average Precision for the detection of fish and fishers at an IoU of 0.5 for all trips. Additionally, the table includes the mean Average Precision based on the size of bounding boxes. In some trips, most objects are larger, and small objects are missing, as indicated by a value of 0.00 in the Table 6. The analysis shows that the model performs well for large objects.

The analysis of errors in the object detector for four different trips is depicted in Fig. 5(a), (b), (c), and (d). A comprehensive examination of the detection performance has been conducted, utilizing the metrics C75, C50, Loc, Sim, Oth, BG, and FN. C75 and C50 represent the results at the IoU of 0.75 and 0.50, respectively. The localization errors (Loc), false positives of super categories (Sim), category confusions (Oth), all false positives (BG), and all false negatives (FN) reveal areas for potential improvement towards achieving a flawless model. As demonstrated in Fig. 5(b), localization errors, false positives, and false negatives hinder the model's performance. Conversely, in Fig. 5(a), (c), and (d), the model's performance is commendable and only minor adjustments are necessary to rectify the false negatives depicted by the orange portions of the graphs.

3.2. Fishing event detection results

In the first experiment, we evaluate the performance of the YOLOX-based event detector and YOLOv5-based event detector in terms of Average Precision and Recall under cross-validation, as shown in

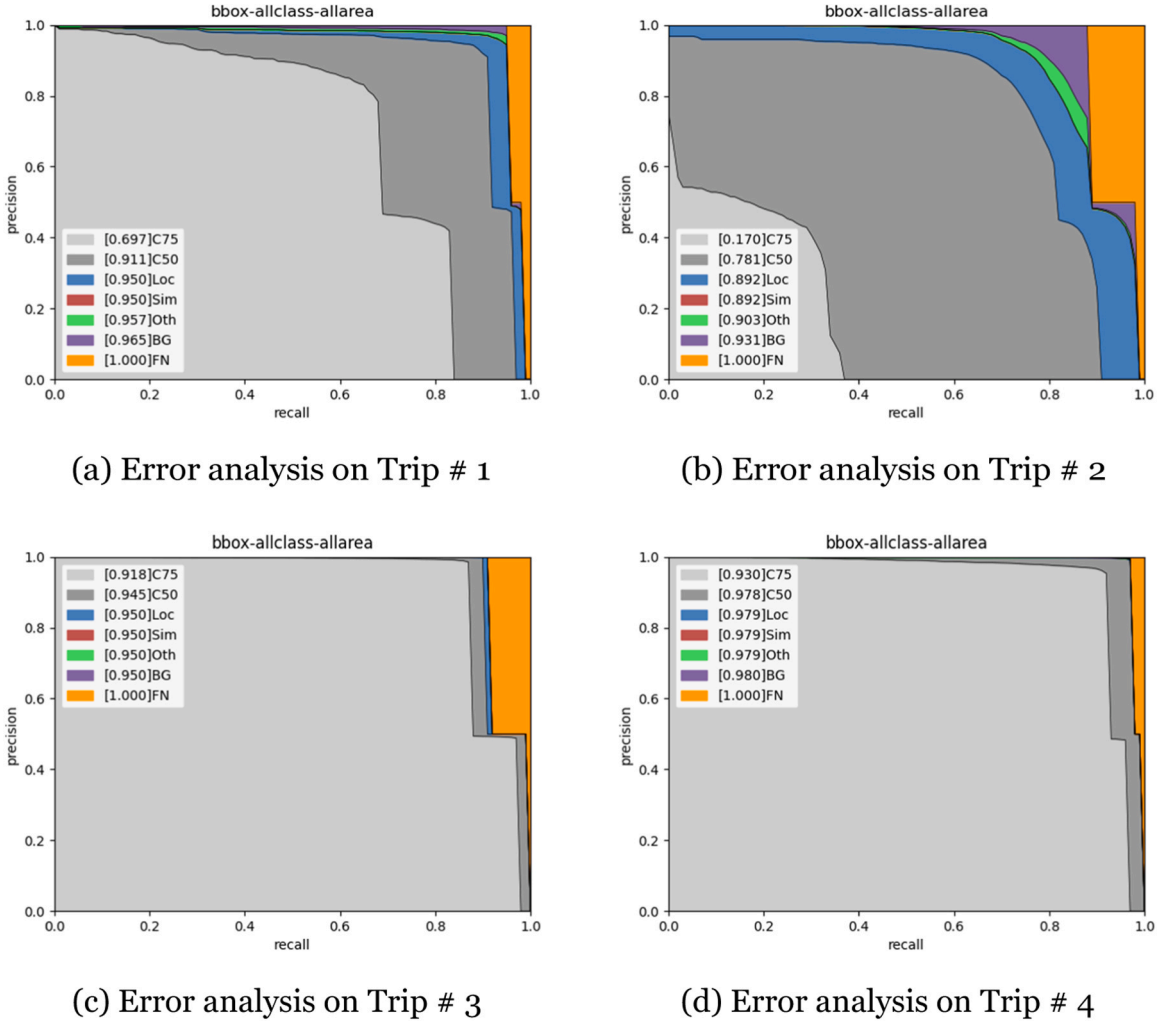


Fig. 5. Error analysis during inference for four different fishing trips.

Table 6
Fish detection results for different fishing trips with Average Precision (AP) at different intersections over Union (IOU) thresholds and object sizes (pixel square).

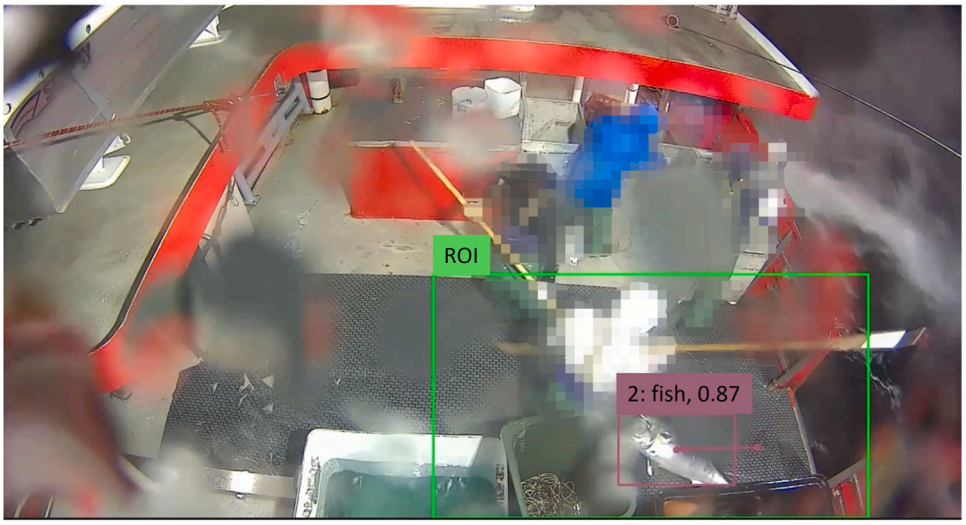
Trip #	mean Average Precision (Fish) IoU=0.5				mean Average Precision (Fisher) IoU=0.5			
	All area	Small area	Medium area	Large area	All area	Small area	Medium area	Large area
1	0.87	0.60	0.78	0.93	0.95	1.00	0.83	0.96
2	0.74	0.00	0.59	0.75	0.83	0.00	0.50	0.84
3	0.90	0.00	0.88	0.97	0.99	0.00	1.00	0.99
4	0.97	0.95	0.96	0.99	0.99	0.00	0.99	0.99

Table 7
Evaluation of YOLOX vs. YOLOv5 for fishing event detection using Average Precision (AP) and Average Recall (AR).

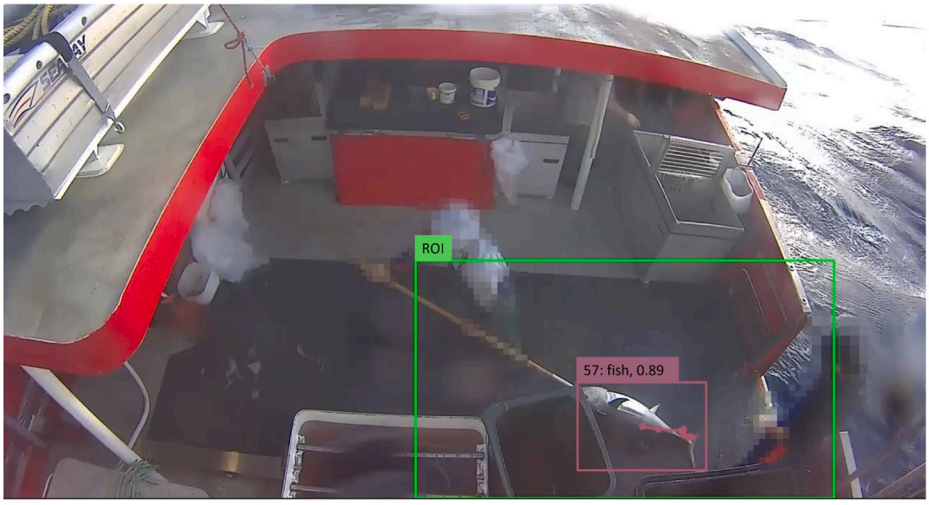
Trip #	YOLOX for detection		YOLOv5 for detection	
	AP	AR	AP	AR
1	0.92	0.90	0.82	0.82
2	0.83	0.82	0.79	0.84
3	0.66	0.61	0.56	0.54
4	0.83	0.65	0.72	0.66
Average	0.81	0.75	0.72	0.72

Table 7. We observe that our proposed YOLOX-based fishing event detector has higher Average Precision and Recall (i.e., 81 % and 74.5 %) than that based on the YOLOv5 (i.e., 72.25 % and 71.5 %). This is because YOLOX adopts a decoupled head and achieves higher accuracy, while a classic coupled head is used in the YOLOv5. In Table 7, Trip # 1 has a much higher Average Precision and Recall than the other three trips, which is likely due to its better video qualities relative to the other three trips. Since the testing videos are noisy or blurry for the other three trips, the fishing event detection results are not as good as Trip # 1. Nevertheless, the Average Precision and Recall are 81 % and 74.5 % over the four trips using the YOLOX-based fishing event detector, which means a more than 12.1 % and 4.2 % improvement in Average Precision and Recall over the YOLOv5-based event detector, as shown in Table 7.

In the second experiment, the fish detection results are presented in Figs. 6 and 7 for visual analysis. The images show fish detection results on four different fishing trips. In each image, the green rectangle box is the region of interest (ROI), the red bounding box identifies the fish, the red trajectory (i.e., a red line with dot points) denotes the fish movement trajectory, and the blue trajectory denotes the movement trajectory of a fisher. We consider only fish trajectories within the ROI when detecting and counting fishing events to reduce the number of repeated counts. These figures show that the proposed algorithms perform well for all the considered trips, especially when the quality of the videos is good (Trip

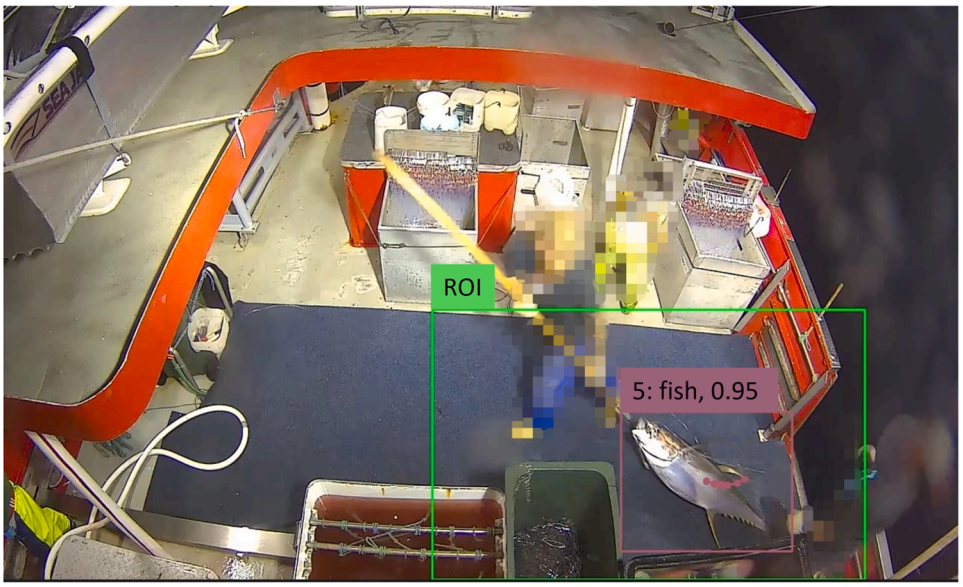


(a) Trip # 1

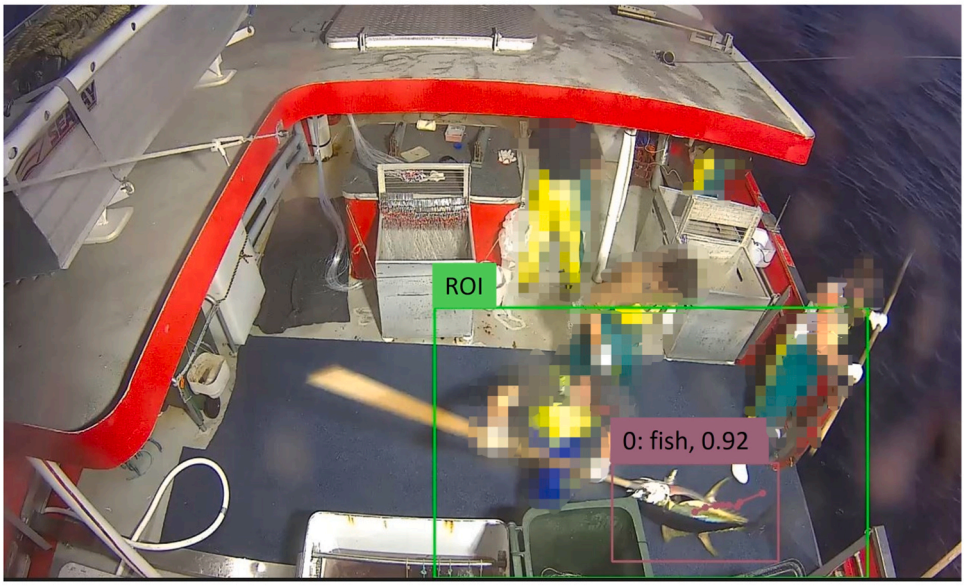


(b) Trip # 2

Fig. 6. Fish detection results on video frames of the first two fishing trips. The green rectangles represent the ROI, and the red bounding-boxes represent the detected/tracked fish with confidence values in the format (track-id: object category, confidence value). The red dots inside the bounding boxes represent the trajectories.



(a) Trip # 3



(b) Trip # 4

Fig. 7. Fish detection results on video frames of the third and fourth fishing trips. The green rectangles represent the ROI, and the red bounding-boxes represent the detected/tracked fish with confidence values in the format (track-id: object category, confidence value). The red dots inside the bounding boxes represent the trajectories.

1).

In the above two experiments, we also investigated issues such as false positives and missing fishing detections due to missed fish detection or falsely detected fish and fishers. False object detections (low Precision of fish detection) may cause false positives (ground truth: 0, predicted: 1) on fishing event detection, which is mainly from the algorithms wrongly recognizing background or fish as targets in the cutting deck camera, as illustrated in Fig. 8(a) and (b). These resulted from several factors, including:

1. *Video quality:* If the video quality is poor, the frames extracted from the video for fish detection and tracking are also in poor quality, reducing the detection and tracking accuracy. Even with good video quality, the selected frames may be blurry or occluded, which may lead to false positives or missed detections.
2. *Occlusion:*An occlusion happens when one or more key attributes used in the fish recognition or tracking are unavailable while the fish is still present at the scene. Occlusions can occur for various reasons, including inter-object occlusion, background scene occlusion, and blurry or noisy image frames due to the poor quality of the videos. The occlusion may result in either false positives or missed detections.
3. *Fish size:*The duration of a fishing event varies with the fish size. It is much easier and faster to move a fish of a smaller size. Consequently, it can be challenging to select an optimal threshold of trajectory length (or the number of consecutive video frames) to identify fishing events, as the threshold may change with the size of the fish.

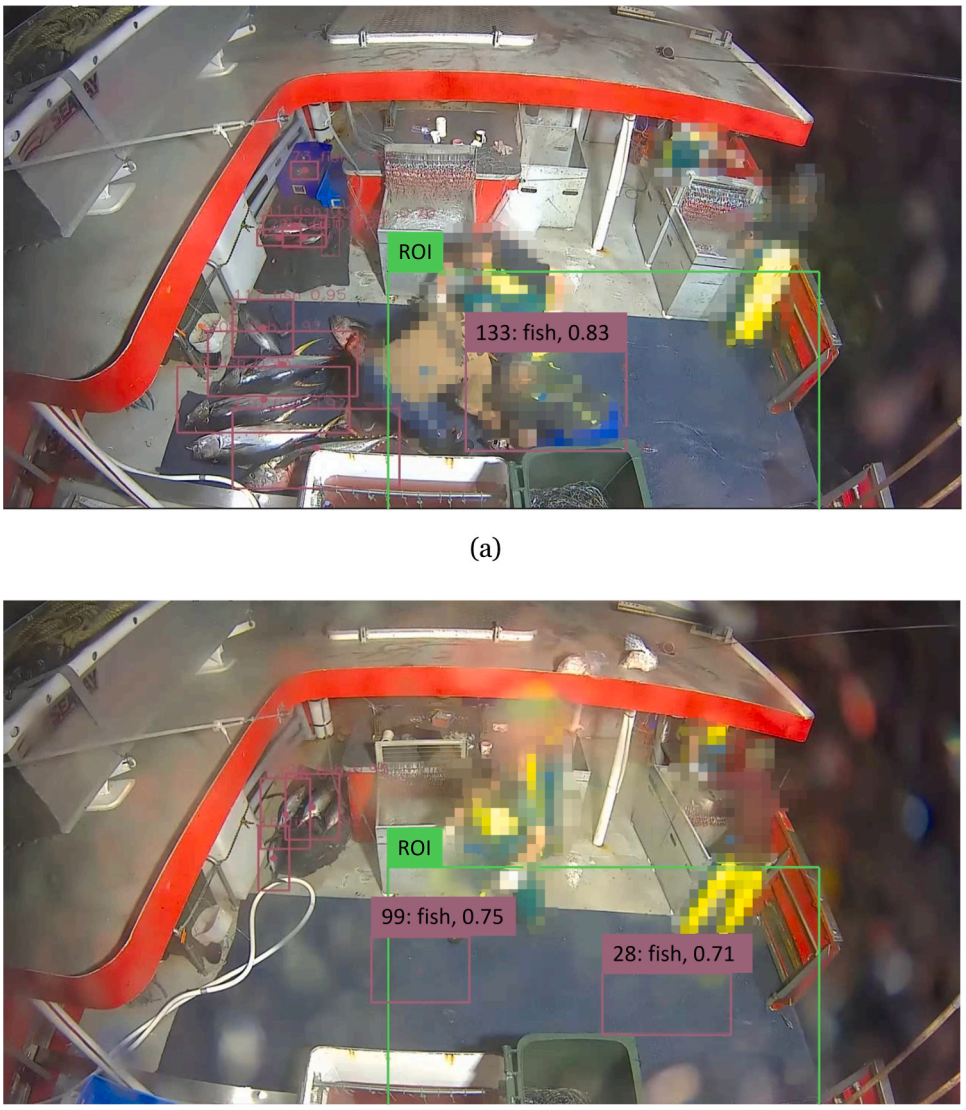


Fig. 8. Examples of false-positive detections. (a) A fisher is detected as a fish due to inter-class similarities. (b) Cutting deck with false-positive fish detections due to bad video quality.

Table 8
Evaluation metrics for the fish species classification task on the test set.

Accuracy (Top-1)	Accuracy (Top-5)	Precision	Recall	F1-score
91.11 %	99.76 %	91.30 %	91.11 %	90.58 %

3.3. Fish species classification results

In the first experiment, the performance of the fish species classifier is evaluated using different quantitative measures (refer to Section 2.6) calculated for the validation dataset. The classifier took each image as input and generated a label with a confidence score. The label indicated the fish's class. The evaluation metrics calculated for the validation dataset are given in Table 8. The proposed classifier achieved an Accuracy (Top-1) of 91.11 % which is also represented by Recall, Accuracy (Top-5) of 99.76 % (which is high because there are only 12 classes), Precision of 91.30 %, and an F1-score of 90.58 %. Here, Top-1 represents that the top prediction matches the ground-truth, whereas Top-5 represents that one of the top five predictions matches the ground-truth. These show that the classifier has excellent potential in classifying different fish species.

In the second experiment, the performance of the classifier is analysed for individual fish species in terms of classification accuracy. For this purpose, confusion matrices were calculated to determine where the inter-class similarities affected the accuracy. Figs. 9 and 10 show the confusion matrices obtained for each fish species. Nine out of twelve classes achieved more than 95 % classification accuracy, with four having 100 % accuracy. Three classes (ALB, UKN, and YFT) have been confused with other classes because of inter-class similarities present (similar shape or image representation). YFT achieved 85.4 % classification accuracy which is reasonable, however, it was occasionally classified incorrectly as ALB and other species of tuna. The classification accuracy for ALB was about 65 % which was often incorrectly classified as YFT and SBT, two species that are quite similar in appearance and shape. UKN with 57.3 % accuracy was confused with almost all other classes because the UKN class was created by taking unrecognisable images of all classes. The UKN class was built from the fish images that humans could not identify, however, the classifier was still able to classify them into different fish species and vice-versa.

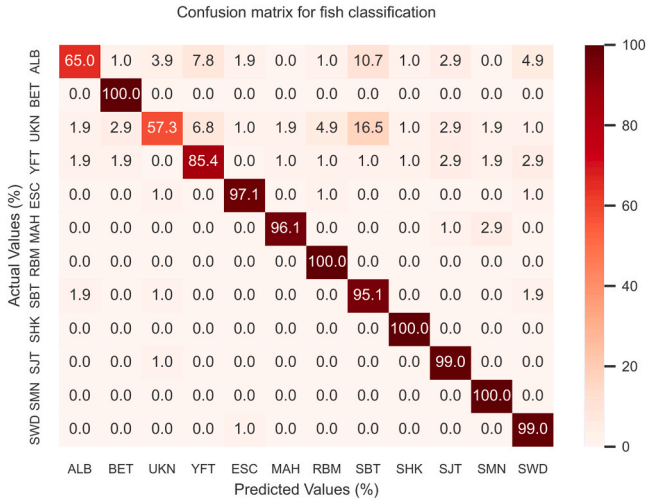


Fig. 9. Confusion matrix for the fish species classification as actual vs. predicted values in percentage (%).

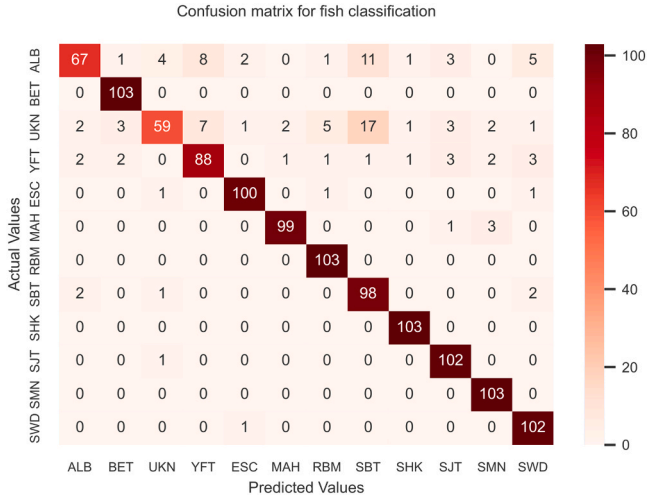
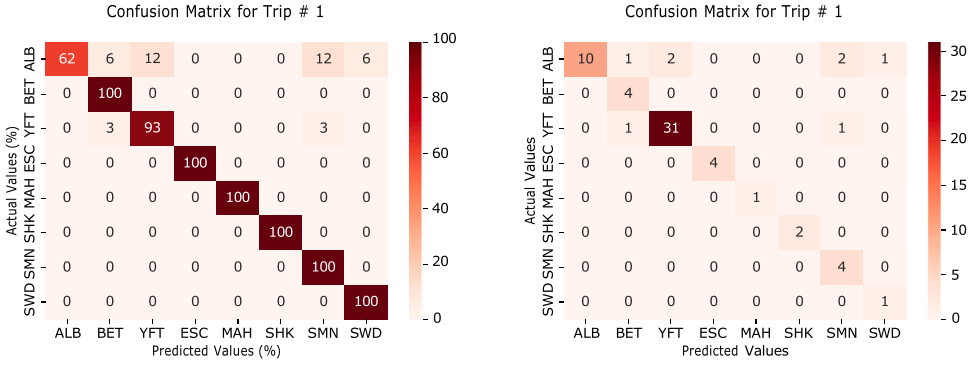


Fig. 10. Confusion matrix for the fish species classification as actual vs. predicted values in the number of images.

In the third experiment, the trained fish classifier was tested on the videos acquired from the commercial fishing vessel. Four fishing trips were evaluated for the task of fishing event classification. In the experiments, the fishing events were given as input in terms of trajectories in the videos. The trajectories were cropped out of the videos, and each cropped fish image from the trajectories was classified using the trained fish species classifier. Confusion matrices were calculated for each of the four fishing trips to analyse the classification accuracy of individual fish species that appear in the trips.

i. In the first fishing trip (Fig. 11), around 75 % of the fish caught were YFT (50 %) and ALB (25 %). Other species caught were BET, ESC, MAH, SHK, SMN, and SWD, representing the remaining 25 %. These six classes achieved 100 % accuracy each. For YFT, 31 out of 33 were classified correctly with 93 % accuracy. Only the ALB class has a lower accuracy than others i.e., 62 %. This is likely because ALB is sometimes incorrectly classified as SMN or SWD due to poor video quality; and YFT or BET, due to inter-class similarities. The mean accuracy of fish species classification achieved for this trip was 94.3 %. The classes RBM, SBT, SJT, and UKN were not identified in this trip.
ii. In the second fishing trip (Fig. 12), the majority of catch events belonged to YFT (58 %) and SJT (36 %). The remaining 6 % of catch events belonged to MAH, SMN, and SWD, with these three species classified with 100 % accuracy. Although these three classes had 100 %, there were relatively few catch events so their high accuracy may not be statistically significant. SJT was classified with 96 % accuracy (25 out of 26 events correctly classified), while YFT, achieved 80 % accuracy (33 out of 41 events correctly classified); a few events were classified incorrectly as SJT because the YFT and SJT in this trip were of similar size. In addition, the water droplets on the camera lens likely added noise to the processing and affected accuracy. The mean accuracy of fish species classification achieved for this trip was 95.2 %. The classes ALB, BET, ESC, SBT, SHK and UKN were not identified in this trip.
iii. In the third fishing trip (Fig. 13), similar to Trip # 1, YFT and ALB have the highest numbers of catch events, i.e., 54 % and 24 %, respectively. The catch events of RBM were nearly 10 %, and the remaining 12 % of catch events belonged to BET, ESC, and SMN. SMN and ESC have only a few events with 100 % and 66 % classification accuracy, respectively. BET and RBM (with slightly more catch events than SMN and ESC) were classified with 80 % and 100 % accuracy, respectively. The highest number of catch events were for YFT, and 39 out of 41 were classified correctly (95 % accuracy), which is promising. The classifier achieved 66 % accuracy for ALB classification, with a few incorrect classifications likely due to inter-class similarities with YFT and other classes due to noisy/blurry images. There were also issues like trajectories jumping from one fish to another, and therefore containing more than one species in a trajectory. This is because multiple fish were stacked together within the ROI. In addition,



(a) (b)

Fig. 11. Confusion matrices for Trip # 1 as actual vs. predicted values: (a) percentage % and (b) the number of catch events.

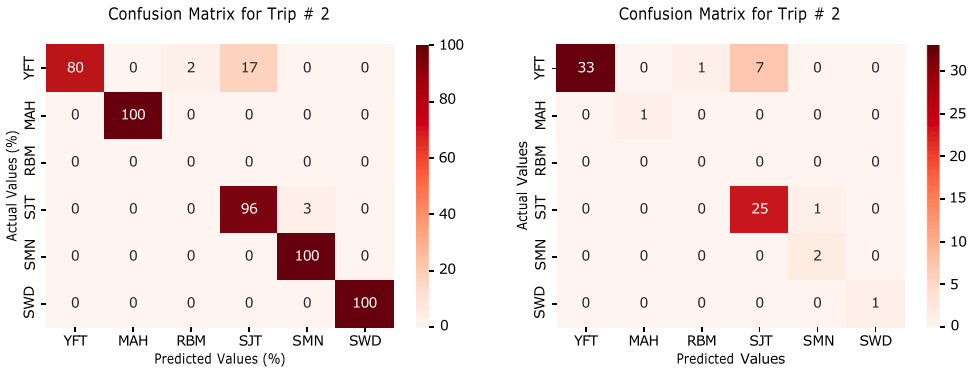


Fig. 12. Confusion matrices for Trip # 2 as actual vs. predicted values: (a) percentage % and (b) the number of catch events.

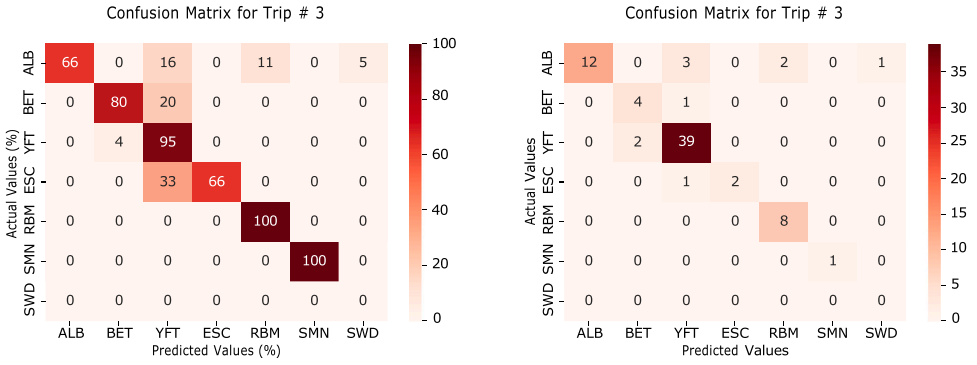


Fig. 13. Confusion matrices for Trip # 3 as actual vs. predicted values: (a) percentage % and (b) the number of catch events.

there were a few trajectories of bagged fish and sometimes multiple trajectories for the same fish. The mean accuracy of fish species classification achieved for this trip was 84.5 %. The classes MAH, SBT, SHK, SJT and UKN were not identified in this trip.

iv. In the fourth fishing trip (Fig. 14), most of the catch events belonged to SBT (i.e., 199 out of 244; 82 %). The second highest number of catch events belonged to ALB which was 16 %. The remaining 2 % of the catch events were for YFT and SWD. SBT was classified with 94 % accuracy, and the classifier achieved 67 % accuracy for ALB with some incorrect classifications as SBT, due to noisy and blurry images and when fish were stacked together. The stacking of fish causes multiple trajectories and multiple fish in a single trajectory. The image quality of some events was poor, resulting in trajectories being classified as UKN. In this case, the second-best class is selected as the final class. If the whole of the trajectory is classified as UKN, the final class remains UKN. The mean accuracy of fish species classification achieved for this trip was 82.2 %. The classes BET, ESC, MAH, RBM, SHK and SMN were not identified in this trip.

The fish species classification results presented for the four fishing trips here show the great potential of computer vision and deep learning approaches to augment human review of EM videos. The overall fishing event classification accuracy for the four fishing trips was 89.05 % with high and consistent accuracy for most of the fish species.

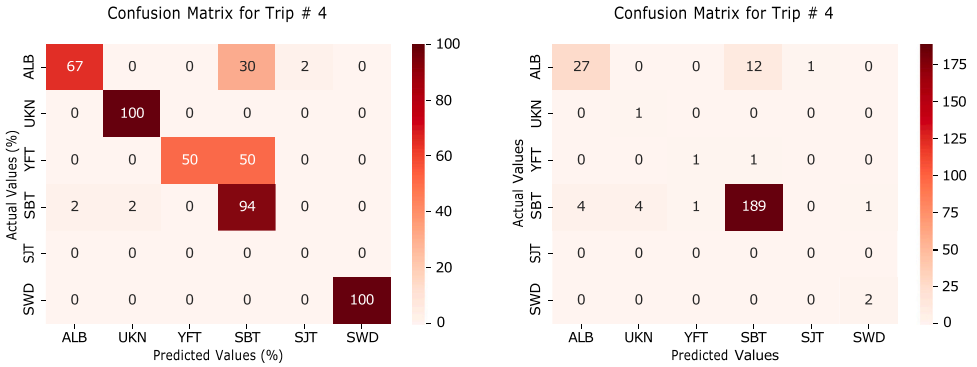


Fig. 14. Confusion matrices for Trip # 4 as actual vs. predicted values: (a) percentage % and (b) the number of catch events.

4. Conclusions

In this work, a fishing event detection and species classification framework is presented for automated analysis of EM videos of fish species captured from a commercial fishing vessel. The contributions are four-fold: first, a comprehensive imagery dataset was curated from the electronic monitoring videos with thousands of fish and fishers annotated by marine science experts to provide a digital ground-truth for the detection and classification tasks. Second, a robust fish and fisher detector was developed based on the Darknet and YOLOX deep learning architectures. The detector was able to handle complex lighting and weather variations, e.g., day and night, exposure to the sun, and sunny, cloudy, and rainy weather. Experimental results show that the developed fish and fisher detector achieved a mean average precision of 87.0 % for fish and 94.0 % for fishers. Third, to track the detected fish and extract the fishing events from videos, a tracking pipeline based on the Kalman filter and the Hungarian algorithm was developed. The tracking was performed within a region of interest and multiple parameters were tuned to efficiently track the fish trajectories and provide the start and end times of fishing events. There were challenges such as fast-moving fish, multiple or stacked fish, and false positives. The pipeline achieved an Average Precision of 81.0 % and an Average Recall of 74.5 % for the task of fishing event detection on the test videos. Fourth and final, a multi-class fish species classifier was developed based on ConvNeXt deep learning architecture. We performed multiple image augmentations to balance the dataset, as rare fish species had only a handful of images. The fish species classifier achieves a fish species image classification Accuracy (Top-1) of 91.11 % and an F1-score of 90.58 % on the validation fish image dataset. For the classification of fishing events extracted by the tracking pipeline, there were issues where trajectories were too short due to fish being obstructed by fishing equipment and infrastructure, water droplets on the camera lens affecting the image quality, and trajectory identity switching due to multiple or stacked fish. These issues made the catch event classification more difficult. Multiple filters were deployed to handle such issues and an overall fishing event classification accuracy of 89.05 % was achieved on extracted catch/fishing events from video of multiple fishing trips.

The study has faced many limitations. Firstly, many videos collected during the trips were not of good quality. Severe weather conditions are a prevalent issue that can deteriorate the quality of video data and require the implementation of video enhancement modules before applying our detection algorithms. Such enhancements are crucial for mitigating the effects of rain, fog, and other environmental factors that obscure visibility. Secondly, to use the video data to develop the machine learning models, we need to spend a significant amount of labour and time annotating the images and videos. This is a critical stage where we can not afford mistakes because only good annotated data would result in good models. Thirdly, the study faced challenges related to severe occlusions, particularly instances where the fisherman might obscure the fish on deck. This, coupled with the suboptimal camera positioning that offers a limited view of the deck, complicates the task of accurate object detection. Fourthly, the high speed of carrying the fish across the deck often results in blurry imagery during their trajectory, further complicating the detection process. We suggested our industry partners deploy a fishing mat where the fishers put down a fish for a few seconds before processing. This would give us an edge where ML and AI models can work better. Lastly, the present model is a single-vessel model. In our future studies, we aim to develop a unified model that is robust and generalizable across multiple vessels. This approach involves training a single model on a diversified dataset that captures various scenarios, including different weather conditions, vessel layouts, and operational procedures. By ensuring a comprehensive distribution in the training dataset from multiple vessels, we aim to enhance the model's adaptability without retraining it for each vessel.

We have shown that computer vision and artificial intelligence have great potential to automate the analysis of electronic monitoring video for fisheries management. The substantial amount of manual work to view and record catch can be reduced through automation and efficient models can be developed to provide on-vessel analysis of the fishing activities. Currently, we are developing a working product as a web portal for consumers to upload their fishing vessel videos and generate reports with fish detection results, fish counts per species, and fishing event timings. This will help human reviewers audit hundreds of hours of videos quickly and with less effort, thereby reducing management costs.

Funding

This research was supported by funding from the Commonwealth Scientific and Industrial Research Organisation (CSIRO).

CRediT authorship contribution statement

Muhammad Saqib: Writing – review & editing, Writing – original draft, Software, Methodology, Data curation. **Muhammad Rizwan Khokher:** Writing – original draft, Visualization, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Xin Yuan:** Writing – original draft, Visualization, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. Bo Yan: Software. **Douglas Bearham:** Writing – review & editing, Data curation. **Carlie Devine:** Writing – review & editing, Data curation, Validation, Formal analysis. **Candice Untiedt:** Writing – review & editing, Data curation, Validation, Formal analysis. **Toni Cannard:** Writing – review & editing, Data curation. **Kylie Maguire:** Writing – review & editing, Data curation. **Geoffrey N. Tuck:** Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **L. Rich Little:** Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Dadong Wang:** Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Muhammad Saqib reports financial support was provided by Commonwealth Scientific and Industrial Research Organisation. Muhammad saqib reports equipment, drugs, or supplies was provided by Australian Fisheries Management Authority. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The data that has been used is confidential.

Acknowledgements

The authors are grateful for the assistance of the Australian Fisheries Management Authority in the provision of footage and Tamre Sarhan (AFMA) and David Ellis (Tuna Australia) in reviewing earlier versions of the manuscript. We acknowledge and thank the fishing vessel and the crew for video data collection.

References

Alori, J., Descoins, A., Javier, L., F., KotaYuhara, Fernández, D., Castro, A., Fatih, D., Linares, R.C., Kurucz, F., Ríos, B., shafu.eth, Nar, K., Huh, D., Moises, 2023.tryolabs/norfair: v2.2.0.10.5281/zenodo.7504727.

Ames, R.T., Williams, G.H., Fitzgerald, S.M., 2005.Using digital video monitoring systems in fisheries: application for monitoring compliance of seabird avoidance

devices and seabird mortality in pacific halibut longline fisheries.⟨https://permanent.fdlp.gov/lps119710/NOAA-TM-AFSC-152.pdf⟩.
Benoîˆt, H.P., Allard, J., 2009. Can the data from at-sea observer surveys be used to make general inferences about catch composition and discards? Can. J. Fish. Aquat. Sci. 66, 2025–2039. https://doi.org/10.1139/F09-116.
CVAT.ai, Corporation, 2022.Computer vision annotation tool (cvat).10.5281/zenodo.10527725.
Depestele, J., Vandemaele, S., Vanhee, W., Polet, H., Torreele, E., Leirs, H., Vincx, M., 2011. Quantifying causes of discard variability: an indispensable assistance to discard estimation and a paramount need for policy measures. ICES J. Mar. Sci. 68, 1719–1725. https://doi.org/10.1093/icesjms/fsr030.
Emery, T.J., Noriega, R., Williams, A.J., Larcombe, J., 2019. Changes in logbook reporting by commercial fishers following the implementation of electronic monitoring in australian commonwealth fisheries. Mar. Policy 104, 135–145. https://doi.org/10.1016/j.marpol.2019.01.018.
Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88, 303–338. https://doi.org/10.1007/s11263-009-0275-4.
Fisher, R.B., Chen-Burger, Y.H., Giordano, D., Hardman, L., Lin, F.P., 2016. Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data. volume 104. Springer International Publishing. https://doi.org/10.1007/978-3-319-30208-9.
French, G., Mackiewicz, M., Fisher, M., Holah, H., Kilburn, R., Campbell, N., Needle, C., 2020. Deep neural networks for analysis of fisheries surveillance video and automated monitoring of fish discards. ICES J. Mar. Sci. 77, 1340–1353. https://doi.org/10.1093/icesjms/fsz149.
Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J., 2021.Yolox: Exceeding yolo series in 2021.arXiv preprint arXiv:2107.08430 10.48550/arXiv.2107.08430.
Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.P., Planqué, R., Rauber, A., Palazzo, S., Fisher, B., Müller, H., 2015. LifeCLEF 2015: Multimedia Life Species Identification Challenges. Springer International Publishing, pp. 462–483. https://doi.org/10.1007/978-3-319-24027-5_46 volume 9283.
Kalman, R.E., 1960.A new approach to linear filtering and prediction problems.
Kandimalla, V., Richard, M., Smith, F., Quirion, J., Torgo, L., Whidden, C., 2022. Automated detection, classification and counting of fish in fish passages with deep learning. Front. Mar. Sci. 8, 2049. https://doi.org/10.3389/fmars.2021.823173.
Khokher, M.R., Little, L.R., Tuck, G.N., Smith, D.V., Qiao, M., Devine, C., O'Neill, H., Pogonoski, J.J., Arangio, R., Wang, D., 2022. Early lessons in deploying cameras and artificial intelligence technology for fisheries catch monitoring: where machine learning meets commercial fishing. Can. J. Fish. Aquat. Sci. 79, 257–266. https://doi.org/10.1139/cjfas-2020-0446.
Kuhn, H.W., 1955. The hungarian method for the assignment problem. Nav. Res. Logist. Q. 2, 83–97. https://doi.org/10.1002/nav.3800020109.
Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft COCO: Common Objects Context 740–755. https://doi.org/10.1007/978-3-319-10602-1_48.
Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S., 2022. A convnet for the 2020s, 11966-11976.10.1109/CVPR52688.2022.01167.
Marini, S., Fanelli, E., Sbragaglia, V., Azzurro, E., Fernandez, J.D.R., Aguzzi, J., 2018. Tracking fish abundance by underwater image recognition. Sci. Rep. 8, 13748. https://doi.org/10.1038/s41598-018-32089-8.
McCann, E., Li, L., Pangle, K., Johnson, N., Eickholt, J., 2018. An underwater observation dataset for fish classification and fishery assessment. Sci. Data 5, 180190. https://doi.org/10.1038/sdata.2018.190.
McElderry, H.I., 2004.Electronic monitoring of seabird interactions with trawl third-wire cables on trawl vessels: a pilot study.⟨https://repository.library.noaa.gov/view/noaa/22858⟩.
McElderry, H., McCullough, D., Schrader, J., Illingworth, J., 2007.Pilot study to test the effectiveness of electronic monitoring in Canterbury fisheries.volume 264. Science & Technical Publishing Department of Conservation.⟨https://www.doc.govt.nz/documents/science-and-technical/drds264.pdf⟩.
Mei, J., Hwang, J.N., Romain, S., Rose, C., Moore, B., Magrane, K., 2021a.Absolute 3d pose estimation and length measurement of severely deformed fish from monocular videos in longline fishing, In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE.2175-2179.
Mei, J., Hwang, J.N., Romain, S., Rose, C., Moore, B., Magrane, K., 2021b. Video-based hierarchical species classification for longline fishing monitoring. International Conference on Pattern Recognition. Springer, pp. 422–433.
Mei, J., Romain, S., Rose, C., Magrane, K., Hwang, J.N., 2022a. Hcil: Hierarchical class incremental learning for longline fishing visual monitoring. 2022 IEEE International Conference on Image Processing (ICIP), IEEE, pp. 3662–3666.
Mei, J., Yu, J., Romain, S., Rose, C., Magrane, K., LeeSon, G., Hwang, J.N., 2022b. Unsupervised severely deformed mesh reconstruction (dmr) from a single-view image for longline fishing. 2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW). IEEE, pp. 1–6.
Miranda, J.M., Romero, M., 2017. A prototype to measure rainbow trout's length using image processing. Aquac. Eng. 76, 41–49. https://doi.org/10.1016/j.aquaeng.2017.01.003.
MMClassification, Contributors, 2020.Openmmlab's image classification toolbox and benchmark.⟨https://github.com/open-mmlab/mmclassification⟩.
Monkman, G.G., Hyder, K., Kaiser, M.J., Vidal, F.P., 2019. Using machine vision to estimate fish length from images using regional convolutional neural networks. Methods Ecol. Evol. 10, 2045–2056. https://doi.org/10.1111/2041-210X.13282.
Palmer, M., Álvarez Ellacuría, A., Moltó, V., Catalán, I.A., 2022. Automatic, operational, high-resolution monitoring of fish length and catch numbers from landings using deep learning. Fish. Res. 246, 106166 https://doi.org/10.1016/j.fishres.2021.106166.
Poos, J., Aarts, G., Vandemaele, S., Willems, W., Bolle, L., van Helmond, A., 2013. Estimating spatial and temporal variability of juvenile north sea plaice from opportunistic data. J. Sea Res. 75, 118–128. https://doi.org/10.1016/j.seares.2012.05.014.
Probst, W.N., 2020. How emerging data technologies can increase trust and transparency in fisheries. ICES J. Mar. Sci. 77, 1286–1294. https://doi.org/10.1093/icesjms/fsz036.
Qiao, M., Wang, D., Tuck, G.N., Little, L.R., Punt, A.E., Gerner, M., 2021. Deep learning methods applied to electronic monitoring data: automated catch event detection for longline fishing. ICES J. Mar. Sci. 78, 25–35. https://doi.org/10.1093/icesjms/fsaa158.
Redmon, J., Farhadi, A., 2018.Yolov3: An incremental improvement.arXiv preprint arXiv:1804.02767 10.48550/arXiv.1804.02767.
Salman, A., Siddiqui, S.A., Shafait, F., Mian, A., Shortis, M.R., Khurshid, K., Ulges, A., Schwanecke, U., 2020. Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. ICES J. Mar. Sci. 77, 1295–1307. https://doi.org/10.1093/icesjms/fsz025.
Tian, Z., Shen, C., Chen, H., He, T., 2019. Fully convolutional one-stage object detection, IEEE.9626-9635.10.1109/ICCV.2019.00972.
Tseng, C.H., Kuo, Y.F., 2020. Detecting and counting harvested fish and identifying fish types in electronic monitoring system videos using deep convolutional neural networks. ICES J. Mar. Sci. 77, 1367–1378.
van Helmond, A.T., Mortensen, L.O., Plet-Hansen, K.S., Ulrich, C., Needle, C.L., Oesterwind, D., Kindt-Larsen, L., Catchpole, T., Mangi, S., Zimmermann, C., Olesen, H.J., Bailey, N., Bergsson, H., Dalskov, J., Elson, J., Hosken, M., Peterson, L., McElderry, H., Ruiz, J., Pierre, J.P., Dykstra, C., Poos, J.J., 2020. Electronic monitoring in fisheries: lessons from global experiences and future opportunities. Fish Fish 21, 162–189. https://doi.org/10.1111/faf.12425.
Vilas, C., Antelo, L., Martin-Rodriguez, F., Morales, X., Perez-Martin, R., Alonso, A., Valeiras, J., Abad, E., Quinzan, M., Barral-Martinez, M., 2020. Use of computer vision onboard fishing vessels to quantify catches: The iobserver. Mar. Policy 116, 103714. https://doi.org/10.1016/j.marpol.2019.103714.
Wu, B., Liu, C., Jiang, F., Li, J., Yang, Z., 2023. Dynamic identification and automatic counting of the number of passing fish species based on the improved deepsort algorithm. Front. Environ. Sci. 11, 1059217.
Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks, IEEE.5987-5995.10.1109/CVPR.2017.634.
Zheng, A., Mei, J., Wallace, F., Rose, C., Hussein, R., Hwang, J.N., 2023.Progressive mixup augmented teacher-student learning for unsupervised domain adaptation, In: 2023 IEEE International Conference on Image Processing (ICIP), IEEE.3030-3034.