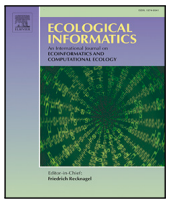
Ecological Informatics 86 (2025) 103013

Contents lists available at ScienceDirect

Ecological Informatics

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



DeepFins: Capturing dynamics in underwater videos for fish detection

Ahsan Jalal [a,b], Ahmad Salman [c,a,*], Ajmal Mian [d], Salman Ghafoor [a], Faisal Shafait [a,b]

[a] School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Sector H-12, 44000, Islamabad, Pakistan
[b] Deep Learning Laboratory, National Center of Artificial Intelligence (NCAI), Sector H-12, 44000, Islamabad, Pakistan
[c] School of Computing, Skyline University College, University City of Sharjah, 1797, Sharjah, United Arab Emirates
[d] School of Computer Science and Software Engineering, University of Western Australia, 35 Stirling Highway, Crawley, WA 6009, Perth, Australia

ARTICLE INFO

Keywords:
Fish detection
Classification
Relative fish abundance
Underwater videos
Deep neural networks
Clustering

ABSTRACT

The monitoring of fish in their natural habitat plays a crucial role in anticipating changes within marine ecosystems. Marine scientists have a preference for automated, unrestricted underwater video-based sampling due to its non-invasive nature and its ability to yield desired outcomes more rapidly compared to manual sampling. Generally, research on automated video-based detection using computer vision and machine learning has been confined to controlled environments. Additionally, these solutions encounter difficulties when applied in real-world settings characterized by substantial environmental variability, including issues like poor visibility in unregulated underwater videos, challenges in capturing fish-related visual characteristics, and background interference. In response, we propose a hybrid solution that merges YOLOv11, a popular deep learning based static object detector, with a custom designed lightweight motion-based segmentation model. This approach allows us to simultaneously capture fish dynamics and suppress background interference. The proposed model i.e., DeepFins attains 90.0% F1 Score for fish detection on the OzFish dataset (collected by the Australian Institute of Marine Science). To the best of our knowledge, these results are the most accurate yet, showing about 11% increase over the closest competitor in fish detection tasks on this demanding benchmark OzFish dataset. Moreover, DeepFins achieves an F1 Score of 83.7% on the Fish4Knowledge LifeCLEF 2015 dataset, marking an approximate 4% improvement over the baseline YOLOv11. This positions the proposed model as a highly practical solution for tasks like automated fish sampling and estimating their relative abundance.

1. Introduction

The automated estimation of fish populations through the analysis of underwater video footage holds significant importance for marine scientists as it enables them to gauge the relative abundance and biomass of different species across various marine ecosystems (Jennings and Kaiser, 1998; McLaren et al., 2015). This critical information is instrumental in safeguarding endangered species from the perils of over-fishing and environmental shifts (Radinger et al., 2019). Furthermore, determining the maximum count of a specific fish species aids marine scientists in establishing environments conducive to fostering greater fish biodiversity in specific regions. The advent of rapid data acquisition through underwater camera systems has made thousands of hours of video data available from diverse regions around the world. However, manually observing and sampling such vast volumes of data presents a labor-intensive and cost-prohibitive challenge for marine biologists and conservationists, despite the undeniable merits of its non-destructive nature. In contrast, the automatic sampling of fish using modern machine learning and computer vision tools is increasingly garnering attention from marine and fisheries communities as an essential requirement.

Fish detection is an essential precursor to their species classification, as each video frame may contain multiple fish belonging to various species. To address this challenge, numerous machine learning (ML) algorithms are available; however, they grapple with the considerable variability within species, non-rigid deformations, alterations in orientation, reduced visibility, and intricate lighting conditions.

In conventional studies, marine scientists rely on textural features to detect and classify fish, utilizing methods such as Principal Component Analysis (PCA), hierarchical decision trees with Support Vector Machines (SVM), and Gaussian Mixture Models (GMM). These techniques aim to automate fish sampling (Huang and Huang, 2014; Lawson et al., 2001; Palazzo and Murabito, 2014; Huang et al., 2015). However, given the intricate nature of the problem, recent research endeavours have shifted towards Deep Neural Networks (DNN) for tasks like fish detection, tracking, and classification.

* Corresponding author at: School of Computing, Skyline University College, University City of Sharjah, 1797, Sharjah, United Arab Emirates.
E-mail address: ahmad.salman@skylineuniversity.ac.ae (A. Salman).

https://doi.org/10.1016/j.ecoinf.2025.103013
Received 5 October 2023; Received in revised form 9 January 2025; Accepted 11 January 2025
Available online 25 January 2025

Salman et al. (2016) introduce fish classification based on Convolutional Neural Networks (CNN) applied to the LifeCLEF 2014 and 2015 datasets, which featured a more extensive class distribution. Similarly, Sung et al. (2017) employ YOLO, a renowned object detector (Redmon and Farhadi, 2018), to achieve a remarkable 93% detection accuracy in 839 fish samples. Xu and Matzner (2018) also leverage YOLO for fish detection across multiple datasets, achieving an impressive mean average precision score of 54%. In a different approach, Jäger et al. (2016) utilize a multi-class SVM on AlexNet CNN features for the LifeCLEF 2015 fish detection task, securing a 74% F1 Score. In their work, Sun et al. (2022) introduce an image enhancement strategy tailored for the task of coral reef fish detection within the LifeCLEF 2015 dataset. The presented method leverage saliency maps generated through a Siamese network to effectively reconstruct input images, markedly improving visual clarity despite challenges such as variations in luminosity, the presence of moving background objects, and image blurriness. Subsequently, Cascade-RCNN is applied to the refined dataset, attaining an F1 Score of 81.7%.

Taking a distinct route, Zhuang et al. (2017) propose pre- and post-processing techniques applied to deep learning to extract fish patches for detection. Choi (2015) base fish detection and classification on GoogleNet, achieving an F1 Score of 84% for 15 species on the same dataset. On the other hand Salman et al. (2019) introduce a combination of GMM features and pixel-wise posterior analysis for fish detection in complex backgrounds, attaining an average F1 Score of 84.28% in the LifeCLEF 2014 fish detection challenge. Furthermore, Jalal et al. (2020) present promising results on LifeCLEF 2015 by adopting a hybrid approach combining temporal and CNN features where GMM and optical flow features are used over raw sequential images and applied the Resnet-50 (He et al., 2016) fish classifier for fish identification. Moreover, YOLOv3 architecture is employed on raw images in parallel, ultimately achieving F1 Scores of 95.47% and 91.64% for fish detection and classification, respectively. Qu et al. (2024) presents ConvFishNet, a lightweight model designed for fish classification from underwater images. ConvFishNet reduces parameters by 80% compared to FishNet while maintaining high precision. It performs effectively on the WildFish and Fish4Knowledge datasets, and achieve precision of 88.4% on WildFish dataset demonstrating its suitability for challenging underwater environments.

Recent advancements in fish detection and behavior analysis have taken the advantage from dense optical flow and CNNs, improving classification accuracy to over 95% in specific datasets (Wang et al., 2021). Ubina et al. (2021) propose optical flow combined with 3D CNNs to achieve high accuracy in estimating fish feeding intensity. A novel semi-supervised learning method, introduced by Jahanbakht et al. (2023), has shown significant promise, using weakly-labeled data and an ensemble of deep neural networks to achieve a 93.6% F1 Score on the FishInTurbidWater dataset.

Accurate fish detection plays a vital role in various fish-related applications, enhancing both ecological research and aquaculture practices. It is fundamental for estimating fish biomass, providing insights into the total mass of fish in a region for sustainable fishery management. Automated fish counting systems monitor population density in aquaculture and natural habitats, while precise detection aids in assessing fish quality and identifying diseases early by analyzing physical anomalies. Furthermore, fish tracking enables the study of movement patterns, behavior, and habitat use, contributing to better management and conservation efforts. Additionally, fish detection supports length measurement for tracking growth and estimating age, and helps calculate relative species abundance, which is critical for biodiversity and ecological balance studies. Rani et al. (2024) propose an automated approach for fish biomass estimation using a modified version of YOLOv8, which incorporates five detection heads. They introduce a dataset called Aquatic WeightNet, consisting of approximately 5000 images of 30 fish species from the Genetically Improved Farmed Tilapia (GIFT). These images are captured in a controlled fish tank environment to ensure precise and consistent data. Their method achieves outstanding results, including 99.7% recall, 99.8% precision, and a 99.4% mean Average Precision (mAP) in fish detection task. Wang et al. (2025) present a comprehensive dataset comprising 256,680 fish instances and propose a lightweight instance segmentation approach based on YOLOv8. This method incorporates CSP bottleneck structures, dual convolutions, and deformable convolutional networks (DConv) to enhance model efficiency and accuracy. Their approach achieves an impressive 98.1% mAP on the fish counting task, demonstrating its effectiveness in handling complex aquatic environments and large-scale fish datasets. Zheng et al. (2024) propose fish recognition method based on YOLOv7 with object segmentation on Fish-Seg dataset comprising of 3000 fish frames from 5 ornamental and farmed fish species and able to achieve 94% F1 Score on fish detection. Similarly, Cao et al. (2024) optimize the YOLOv5 framework with a SimAM attention-based mechanism to enhance fish detection accuracy. They also introduce a custom dimension measurement method by integrating the YOLOv5-keypoint framework. Their dataset comprises 3849 images captured against a light-colored backdrop, focusing on a single fish species. This approach achieves a 97.81% mAP for fish detection and an average MSE of 0.104 for the fish measurement task. Fish diseases pose significant risks of contagion, often leading to considerable economic losses. In a study, Li et al. (2024) propose YOLO-FD, a multi-task learning network based on YOLOv8, designed to simultaneously detect fish instances and segment infected areas. They also introduce the Nocardiosis Fish Dataset, comprising 1072 images of 30 largemouth bass fish. Their approach achieves a fish disease detection mAP of 94.2%. On the other hand, Saqib et al. (2024) presents a YOLOX deep learning framework for automatic fish detection, event counting, and species classification using computer vision and AI in electronic monitoring (EM) videos. They propose a dataset comprising 20,019 fish annotations from 5198 images of 12 fish species. They achieve mAP of 81% and top-1 accuracy of 91.11% for fish detection and classification respectively. Additionally, unsupervised learning approaches for fish tracking and segmentation have been explored, demonstrating the effectiveness of integrating temporal and spatial features in underwater imagery analysis (Saleh et al., 2022).

Some studies have also delved into fish detection using the OzFish dataset. One study addressed the challenge of accurate data labeling under various underwater environmental conditions, employing a YOLOv4 model in a self-supervised setting on the CCMAR and OzFish datasets, resulting in an F1 Score of 74.68% (Veiga et al., 2022). Another study by Marrable et al. (2022) reports an F1 Score of 78.6% using YOLOv5. Muksit et al. (2022) unveil two robust fish detection models known as YOLO-Fish-1 and YOLO-Fish-2. These models are adaptations of the YOLOv3 framework, specifically tailored to address the challenges of detecting small fish instances and effectively identifying fish within dynamic environments. To evaluate these models, various experiments are conducted using the DeepFish and OzFish datasets. Remarkably, this approach achieved an impressive 73% F1 Score when tested on the OzFish dataset.

As widely recognized, the utilization of more intricate and deeper neural architectures necessitates a substantial volume of training data and may still be susceptible to overfitting, as acknowledged by LeCun et al. (2004). Additionally, despite their improved generalization capabilities and promising outcomes, these complex models demand considerable computational resources, rendering them less practical for real-time video processing applications. The primary focus in this study centers on enhancing contemporary state-of-the-art object detection methods like YOLO. This enhancement is achieved through a meticulously crafted algorithm designed to capture fish motion-related features while concurrently eliminating irrelevant noise and background interference. The evaluation draws upon the OzFish and Fish4Knowledge LifeCLEF 2015 datasets, both of which pose challenges stemming from the underwater environment's inherent variability, characterized by poor visibility and aquatic background ambiguity.

To summarize, the specific contributions in this study are as follows:

Fig. 1. Sample images to illustrate the high variation in terms of illumination, crowd, visibility, camouflage, and contrast in Ozfish dataset (first three rows) and LifeCLEF 2015 dataset (last two rows).

1. A hybrid solution is introduced that merges a modified YOLOv11-based spatial fish detector with a custom designed motion-based segmentation method.
2. The proposed approach employs the advantages of the proposed motion segmentation module, am algorithm that works on temporal data, while overcoming limitations related to detecting large moving background objects. This refinement yields enhanced segmentation performance, with reduced false alarms as well as reduced computation time.
3. A modified YOLOv11 branch is used to perform dual tasks: classifying fish versus non-fish objects within image patches detected by the motion segmentation module and conducting full frame-based fish detection.

The rest of the paper is structured as follows: Section 2 provides an overview of the datasets employed, namely the OzFish dataset and LifeCLEF 2015 dataset. It also presents the hybrid solution integrating YOLOv11 with a meticulously crafted motion-detection module, addressing fish detection in uncontrolled underwater environments. The model's performance is assessed in Section 3. Section 4 contains an in-depth exploration of the advantages of the proposed approach, comparing it with existing state-of-the-art. Finally, the paper concludes in Section 5, where future research directions are also discussed.

2. Materials and methods

This section commences by introducing the datasets employed and subsequently advances to present the proposed static and motion-based hybrid feature extraction method designed to enhance fish detection accuracy.

2.1. Datasets

Evaluations are conducted using two datasets, namely OzFish and LifeCLEF 2015, to assess the performance of the proposed algorithm. The LifeCLEF 2015 dataset is employed to conduct cross-model validation, with a model trained on OzFish data and then evaluated on the LifeCLEF 2015 dataset. Both datasets pose significant challenges related to dynamic backgrounds, moving background objects, blurriness, occlusions, jitter, and other factors. The dataset particulars are provided below.

2.1.1. OzFish dataset

The OzFish dataset, provided by the Australian Institute of Marine Sciences (AIMS), is employed as one of the validation datasets. This dataset encompasses more than 3000 videos recorded at various times, with a total of 1758 frames across the videos annotated for the fish detection task. These annotated frames collectively comprise approximately 45,000 labeled bounding boxes, with each frame potentially containing multiple such boxes. It is noteworthy that these annotated video frames are not consistently chronological and appear at arbitrary instances within the videos. The videos in the dataset are in 1080p resolution, in MP4 format, and are recorded at 24 frames per second. The dataset presents challenges related to luminosity variations, contrast, blurriness, fish camouflage, moving background elements like plants and corals, and water turbidity, among other factors as shown in Fig. 1.

To address the limitation of the original 1758 annotated frames for training the algorithm, annotations are added to 8185 additional frames, bringing the total dataset size to 9943 frames. These extra frames are selected from the pool of 3000 AIMS videos, which were not originally annotated. For the fish detection task, this augmented dataset is partitioned into training, validation, and test sets, with proportions

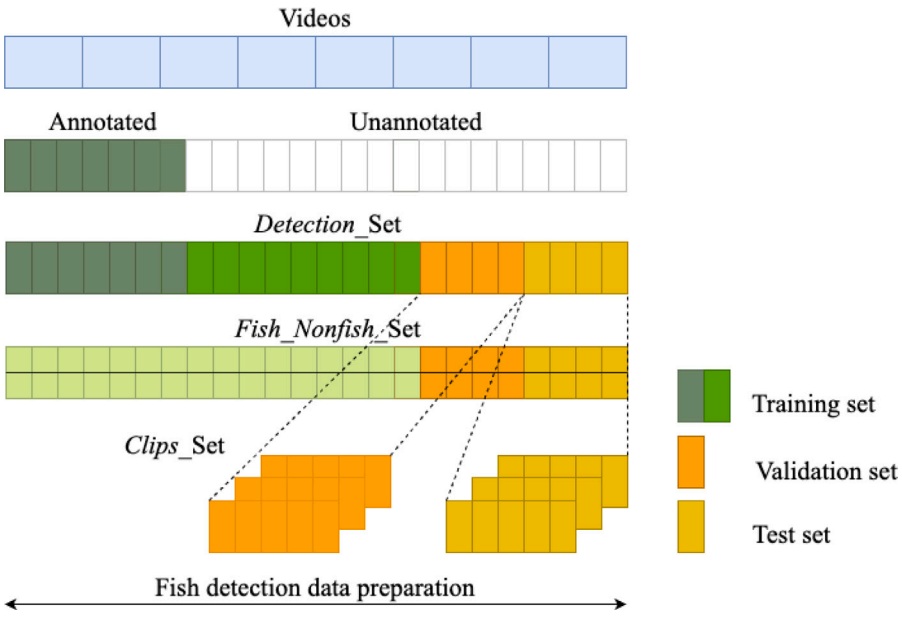


Fig. 2. Description of datasets and splitting strategy for fish detection. The same splitting strategy is used for the LifeCLEF 15 dataset except that no additional unannotated frames are used.

of 80%, 10%, and 10%, resulting in 7955, 994, and 994 frames, and 104,188, 8411, and 14,906 fish instances in each set, respectively. During the dataset splitting process, it is ensured that frames from the same video are included in only one of the three splits (training, validation, or test) for a fair evaluation. This extended dataset is referred to as the *Detection_Set* dataset as shown in Fig. 2.

In the pipeline, a binary *Fish-Nonfish* classifier is trained as an additional head of the YOLOv11 system (to be explained later). For this purpose, the *Detection_Set* training set with annotated frames is used, and up to 20 background instances (non-fish) are introduced in each frame. These background instances are generated with the constraint that they do not overlap with any bounding box covering fish in the same frame. The aspect ratio for each background box is randomly selected within a range of 0.25 to 3. Similarly, validation and test sets for the binary classifier are created using the initial validation and test splits, forming the *Fish_Nofish_Set* dataset.

Furthermore, short video clips of around 3 s, corresponding to various frames within the 3000 videos, are released by AIMS. Clips are selected exclusively from videos designated for the validation and test set, and this collection of short clips is referred to as the *Clips_Set* dataset. This dataset of clips is specifically used by the segmentation module, which is a non-trainable algorithm. Consequently, there is no need to create a dedicated training set for these short video clips. However, a few clips around the frames included in the validation sets are reserved for parameter fine-tuning.

2.1.2. LifeCLEF 2015 dataset

The LifeCLEF 2015 dataset comprises 93 annotated videos, each containing instances of 15 distinct fish species. This dataset is a subset of a more extensive collection of underwater videos known as Fish4Knowledge, as documented by Joly et al. (2015) in 2016. The Fish4Knowledge repository encompasses over 700,000 unconstrained underwater videos captured using stationary cameras. These videos are recorded over a span of five years, with the primary objective of monitoring the marine ecosystem within the coral reefs of Taiwan. Note that this region boasts one of the most diverse fish populations globally, featuring over 3000 different fish species. Fig. 1 provides a visual representation of some selected samples drawn from the dataset, offering a glimpse of its inherent diversity and characteristics.

The choice of datasets for any machine learning-based object detection system is crucial, as it directly impacts the system's accuracy, robustness, and ability to generalize. This is especially important for unconstrained underwater fish detection tasks, where the data must include a wide range of fish instances against diverse backgrounds. Such diversity helps the machine learning model distinguish fish of various shapes and colors from vibrant backgrounds, reducing the likelihood of misdetections due to camouflage. The LifeCLEF 2015 dataset effectively serves this purpose. Additionally, it is essential for the dataset to cover different luminosity conditions, including various levels of turbidity and water murkiness. This challenges supervised machine learning algorithms like YOLO to extract distinctive, fish-specific features from the underwater data. To our knowledge, the OzFish and LifeCLEF 2015 datasets are two publicly available large datasets that meet both of these requirements (see Fig. 1).

2.2. Methodology

Deep neural networks (DNN) continue to be the most effective choice for fish detection and species classification tasks in images or videos. Nevertheless, an augmentation of DNN capabilities can be achieved by integrating traditional computer vision algorithms. In light of recent advancements in spatio-temporal feature extraction, the proposed Algorithm 1 combines spatial and temporal feature extraction to enhance fish detection in underwater videos. see Fig. 3.

This approach effectively addresses the unique challenges posed by datasets such as OzFish and LifeCLEF 2015, which are characterized by low visibility, dynamic water movement, and the presence of non-fish objects. By integrating temporal motion analysis with state-of-the-art spatial detection models, the algorithm provides robust and precise fish detection capabilities.

The proposed algorithm begins by extracting individual frames from the input video, ensuring a consistent and standardized format for subsequent processing. Each frame undergoes preprocessing steps such as resolution adjustment, color normalization, and noise reduction to mitigate variations in video quality. These steps establish a uniform foundation for further analysis and enable the algorithm to perform reliably across diverse underwater environments.

To incorporate temporal dynamics, the algorithm employs a motion segmentation module that identifies regions of interest (ROIs) in each frame by segmenting moving blobs from the static background. Using a direction-based clustering technique applied to motion vectors,

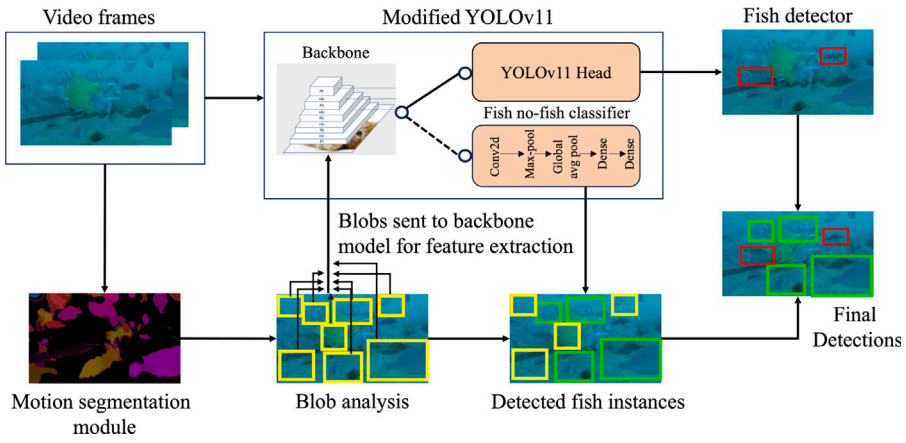


Fig. 3. The proposed hybrid fish detection system consists of two parallel streams. In the bottom stream, the segmentation module processes the video, resulting in moving fish candidates. This output is then fed into the YOLOv11 backbone, and the extracted features are subsequently input to the fish/non-fish classifier module. In the top stream, raw video frames are directly passed to the YOLOv11 backbone, followed by the YOLOv11 head module. The results from both streams are combined to enhance fish detection accuracy. In the visualization, false detections by the segmentation module are represented by yellow boxes, while legitimate moving fish candidates are shown in green, and red boxes indicate detections by YOLOv11 (best viewed in color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the module isolates dynamically relevant areas while discarding static background clutter. This segmentation process ensures that only moving objects – likely candidates for fish – are carried forward, reducing unnecessary computations and improving efficiency.

After motion segmentation, the proposed algorithm performs blob analysis to refine the candidate regions. Each segmented blob is examined for size, motion trajectory, and shape, with invalid blobs, such as noise or objects below a predefined size threshold, being discarded. This filtering step allows the algorithm to focus solely on meaningful objects, enhancing its accuracy and efficiency in detecting fish. In parallel, the algorithm employs YOLOv11, a state-of-the-art object detection framework, to extract spatial features from each preprocessed frame. YOLOv11 generates high-dimensional feature maps and detects bounding boxes representing potential objects. These spatial detections complement the temporal features derived from the motion segmentation module, providing a comprehensive representation of each frame's content.

The next step involves binary classification of blobs to differentiate between fish and non-fish objects. Each blob from the motion segmentation module is passed through the YOLOv11 classification head, which utilizes the extracted spatial features for fine-grained classification. Only blobs classified as fish are retained, significantly reducing false positives and enhancing the algorithm's precision. This classification step is crucial for filtering out non-fish objects, such as debris or marine vegetation, commonly present in underwater scenes.

Finally, the algorithm combines the outputs from the motion segmentation classifier and YOLOv11 detection branches to produce the final fish detections. For each frame, the fish blobs identified during binary classification are merged with YOLOv11's bounding box detections. The merging of fish instances from the YOLOv11 detector and the motion segmentation classifier employs a preferential approach. When fish blobs overlap between the two branches, the results from the YOLOv11 detector, which generally provides superior localization accuracy, are prioritized. In contrast, when no overlap exists, the fish blobs identified by both branches are retained as valid outputs, ensuring a comprehensive and accurate detection process.

By combining motion-based segmentation with advanced spatial detection techniques, this algorithm offers a robust solution for fish detection in underwater videos. It effectively addresses challenges such as cluttered backgrounds, dynamic water movement, and poor visibility. The synergy between spatial and temporal features enables high-precision detection, making the algorithm well-suited for applications in marine research and ecological monitoring.

Algorithm 1 DeepFins Algorithm for Fish Detection and Classification using Temporal and Spatial Features.

Require: V: Underwater video sequence, $\mathcal{Y}$: YOLOv11 model, C: Direction-based Clustering
Ensure: Set of fish detections $D = \{D_1, D_2, \ldots, D_N\}$
$F = \{F_1, F_2, \ldots, F_T\}$ from V.
for each frame $F_i \in F$ **do**
  $F_i \leftarrow Preprocessing(F_i)$
  $M_i \leftarrow Motion\ Segmentation(F_i)$
  $R_i \leftarrow C(M_i)$
  for each candidate region $r_j \in R_i$ **do**
    Calculate blob size s_j
    Discard r_j if $s_j < \tau_s$
    Extract features $\phi(r_j)$ from the corresponding region in F_i'.
    $c_j = \mathcal{Y}_c(\phi(r_j))$
    Retain r_j if $c_j =$ "fish"
  end for
  $B_i = \mathcal{Y}_d(F_i')$.
  $D_i = \emptyset$
  for each blob $r_j \in R_i$ and bounding box $b_k \in B_i$ **do**
    if $r_j \cap b_k = \emptyset$; $D_i \leftarrow D_i \cup \{r_j, b_k\}$
    if $r_j \cap b_k \neq \emptyset$; $D_i \leftarrow D_i \cup \{b_k\}$
  end for
end for
Output: Combine the merged detections from all frames to generate the final set of fish detections D.

2.2.1. Static image-based detection

In the static image-based detection system, YOLOv11 (Khanam and Hussain, 2024), an advanced real-time object detector that ranks among the current state-of-the-art solutions, is adopted. It is one of the latest and most stable iteration in the YOLO (You Only Look Once) family of object detection models, incorporates several advanced features to enhance its performance in scenarios with complex backgrounds, variable lighting conditions, and diverse object scales which are common characteristics in underwater settings. Moreover, this iteration of YOLO incorporates significant advancements in augmentation techniques during training, thereby enhancing its ability to harness the underlying architecture's learning potential. This feature is particularly advantageous for addressing specific problem in this study.

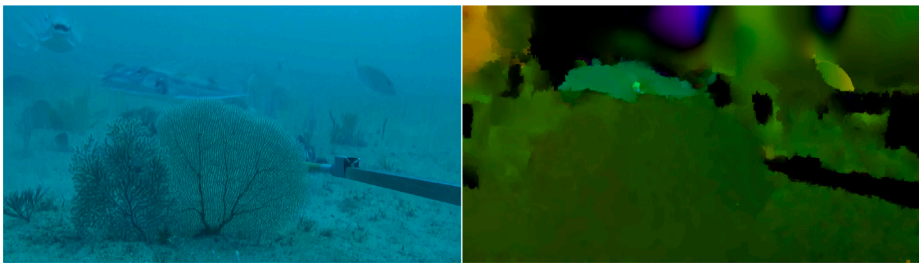


Fig. 4. The image on the left illustrates the presence of moving aquatic plants in the background, while the image on the right displays the intermediate results, which are identified fish candidates obtained from the proposed motion segmentation algorithm. It is evident that temporal-based feature detectors also identify moving background objects, resulting in an excessive number of false detections and subsequently compromising the precision rate. Image best viewed in color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

YOLOv11 is built upon a modified C3k2 (Cross Stage Partial with kernel size 2) block which is computationally more efficient implementation of the Cross-Stage Partial (CSP) Bottleneck. It employs two smaller convolutions instead of one large convolution, which contributes to faster processing while maintaining performance. This block efficiently extracts hierarchical features, crucial for recognizing fish with varying textures, scales, and shapes in underwater videos. This architecture enables the model to detect objects with high precision, even in visually noisy conditions, such as coral reefs or murky water. This block replaces the C2f block used in previous versions.

YOLOv11 uses an anchor-free detection head, simplifying the bounding box prediction process by directly regressing box coordinates and object probabilities. This removes the dependency on pre-defined anchor boxes and improves detection performance, especially for objects like fish that exhibit diverse sizes and orientations. The anchor-free design is advantageous for underwater detection, where fish can appear at random scales and in unpredictable locations within the frame.

The Spatial Pyramid Pooling - Fast (SPPF) module in YOLOv11 is a refined version of the traditional Spatial Pyramid Pooling (SPP), designed to accelerate multi-scale feature extraction while maintaining efficiency. It uses three consecutive max-pooling layers with a fixed kernel size to aggregate multi-scale spatial context. By stacking these pooling layers rather than relying on separate large kernel operations, SPPF reduces computational complexity and memory usage while effectively capturing a broad receptive field. This significantly improves fish detection in dynamic, complex underwater environments by enhancing the model's ability to capture spatial information at multiple scales. The aggregated features help the model better differentiate fish from challenging backgrounds like coral, seaweed, or debris, providing a wider context for more accurate detection. Its efficiency supports real-time applications, while the strong feature representation aids in recognizing fish-specific traits such as fins, scales, and body shapes, ultimately improving both detection accuracy and speed.

YOLOv11 integrates advanced modules to significantly improve detection accuracy, especially in challenging environments like underwater. One key feature is the C2PSA (Convolutional block with Parallel Spatial Attention), which enhances spatial attention by allowing the model to focus on important regions within the image. This is particularly valuable for detecting fish at different scales and positions, including smaller or partially occluded ones. By concentrating on key areas, this attention mechanism helps the model distinguish fish from complex and cluttered backgrounds like seaweed or coral.

YOLOv11 also enhances feature extraction and processing through its improved backbone and neck structures, including the C3k2 block. This enables the model to capture multi-scale features more effectively. These features are further refined by CBS (Convolution-BatchNorm-Silu) layers, which use the Sigmoid Linear Unit (SiLU) activation function for non-linearity. This setup stabilizes and normalizes data flow, improving the overall precision of object detection. For fish detection, these upgrades allow the model to more accurately identify fish by recognizing distinctive features such as fins, scales, and body shapes, even in the dynamic and complex underwater environment.

2.2.2. Motion-based detection

Researchers have employed various techniques in real-word applications to capture motion-based features, including methods like GMM and optical flow, to extract moving foreground objects from video data. These techniques can also contribute to the effectiveness of fish detection approaches, particularly in scenarios where static image-based segmentation struggles to accurately delineate fast-moving fish within dynamic backgrounds, resulting in frequent miss detections and reduced recall rates. However, these methods encounter challenges when confronted with moving background elements such as aquatic plants and algae, often erroneously identifying them as fish instances, as depicted in Fig. 4 on the left image. Such instances generate a surplus of candidates for fish detection, leading to increased computational load and elevated false alarm rates, ultimately diminishing the overall system precision. To tackle this issue, a novel temporal fish detection method is introduced which is designed to minimize false alarms caused by moving background elements.

For video based temporal detector, a segmentation algorithm is designed which captures motion pattern generated by moving objects in pairs of video frames and at the same time removes static objects including background comprising coral reef and aquatic plants. The mathematical model is presented in Appendix. However, the output of segmentation may still contain remnants of non-fish objects, such as noise resulting from dynamic light beams and the motion of aquatic plants. To eliminate these artifacts, the identified blobs in the segmented image are overlaid onto the RGB frame, and their corresponding regions are delineated with bounding boxes. These RGB images encompass all the motion candidates predicted by the segmentation module. The cropped regions within the bounding boxes are subsequently fed into the YOLOv11 binary classification branch, which distinguishes between fish and non-fish entities. This branch is implemented within a multi-objective optimization framework, positioned after the DarkNet feature extractor, as illustrated in Fig. 4. This addition allows the system to refine detections by leveraging deep feature representations extracted by the backbone, ensuring more accurate classification and filtering of irrelevant entities in challenging underwater environments. The fish candidates identified by this process are then integrated with the static fish candidates localized by the detector branch of YOLOv11. Fig. 5, illustrates some outcomes of the proposed segmentation module. The merging of fish instances from the YOLOv11 detector and classifier follows a preferential approach, where results from the detector branch, which typically exhibits better localization, are favored over those from the classifier branch, powered by the motion segmentation module, when there is an overlap between the fish blobs. Conversely, fish blobs from both branches are considered valid outputs when there is no overlap between them, as depicted in Fig. 3.

3. Results

In this section, the outcomes of the proposed fish detection architecture are presented. The model's performance is assessed using

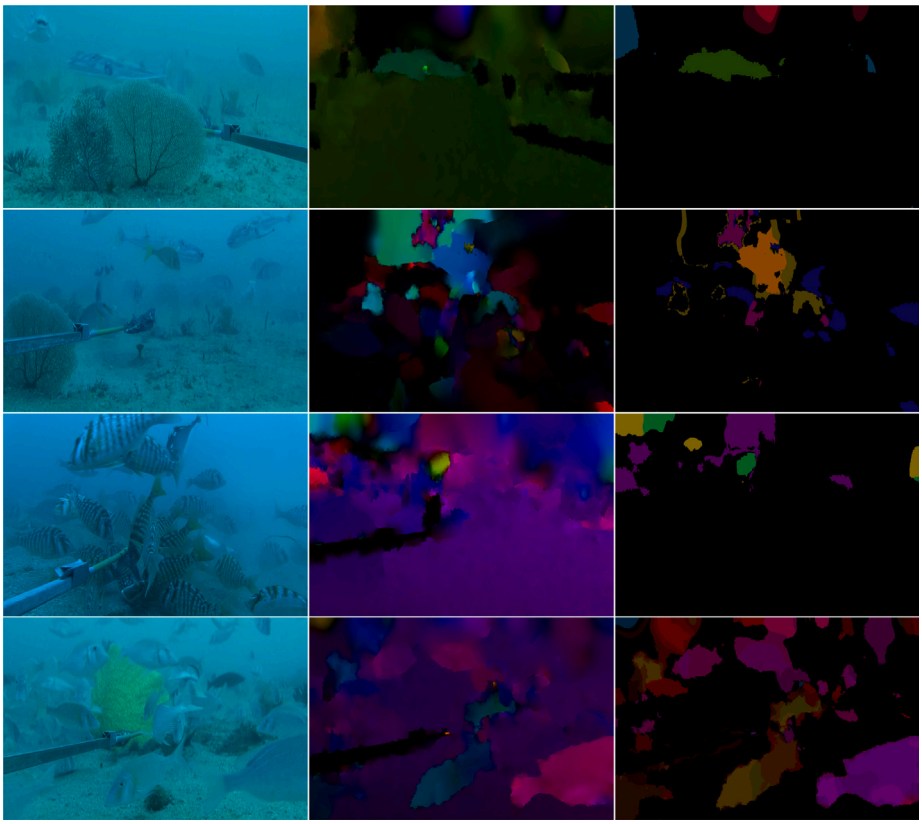


Fig. 5. Illustrative outcome of the proposed motion segmentation algorithm. First column are raw images. Second column shows motion depicting candidates while third column shows refined output after background subtraction for static and insignificantly moving objects (best viewed in color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

various metrics. Fish detection accuracy is measured using the F1 Score, as indicated in Eq. (1), which represents the harmonic mean of precision and recall. This score is particularly valuable when achieving a balanced trade-off between precision and recall is essential for overall performance.

$$\text{F1 Score} = \frac{2(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}. \quad (1)$$

3.1. Fish detection

It is recalled from Fig. 3 that the fish detection system is composed of two branches. In one of these branches, YOLOv11 is utilized for the detection of fish instances due to its exceptional accuracy and efficient computational performance when compared to competitors such as Single-Shot Detector (SSD) (Liu et al., 2016), Region Proposed Network (RPN) (Ren et al., 2015), and Feature Pyramid Network (FPN) (Lin et al., 2017). YOLOv11 is a pre-trained model on the extensive ImageNet dataset (Khanam and Hussain, 2024), serving as a versatile image feature extractor. ImageNet covers an extensive spectrum of one thousand object classes, incorporating a wide range of maritime categories like goldfish, ray fish, jellyfish, spiny lobster, crayfish, grey whale, starfish, anemone fish, garfish, lionfish, pufferfish, and sea, among others. Utilizing this pre-trained model significantly reduces the need for extensive fine-tuning on the fish dataset.

The YOLOv11 model is trained using raw images from the *Detection_Set* and fish/non-fish crops sourced from the *Fish_NonFish_Set*. The label file for YOLOv11 training contains information about the locations of fish candidates and their corresponding labels. For legitimate fish instances, the label is set to 1, accompanied by the coordinates of the corresponding bounding boxes. If there are no fish instances in an image, an empty vector is forwarded. During training, the detector branch of YOLOv11 is trained for regression tasks on coordinates using mean square error, while the classifier branch is trained on labels using cross-entropy loss. The training process is carefully designed with a set of hyperparameters and techniques to optimize the network's performance. The 300 epochs ensure sufficient training iterations for the model to learn complex patterns, while a batch size of 16 balances computational efficiency with gradient stability. The choice of an image size of 640 × 640 pixels provides a good trade-off between computational demand and the ability to capture fine details. Moreover, a learning rate of 0.01 is selected to control the pace of parameter updates, ensuring stable convergence, while an optimizer weight decay of 0.0005 prevents overfitting by regularizing the model weights. The warm-up momentum set to 0.8 facilitates a smoother start to training, avoiding abrupt weight updates in initial epochs.

Data augmentation techniques further enhance the model's robustness and generalization. HSV adjustments ([0.015, 0.7, 0.4]) simulate varying lighting conditions, while translation by 0.1 adds positional variance. Mosaic augmentation (factor of 1.0) combines multiple images, improving object detection in cluttered scenes, and horizontal flipping (probability of 0.5) introduces directional diversity. At the inference stage, a confidence threshold of 0.6 filters out low-probability detections, prioritizing precision. The consistent resolution of 640 × 640 pixels during training and inference ensures compatibility between the data pipeline and the model, maintaining the integrity of both original images and cropped regions. These hyperparameters collectively enable the model to achieve high accuracy and generalization across diverse scenarios.

The dataset is first split into 60% training, 20% validation, and 20% testing sets. The model is then trained until early stopping criteria is met on the validation set. The model achieved an F1 Scores of 75.2% and 69.4% on the validation and test set respectively. Subsequently, the training and validation sets are combined to form a larger training set,

Fig. 6. Image enhancement: First column are raw images. Second column shows enhanced images as a result of histogram equalization.

Table 1
Comparison of performances on fish detection task on OzFish test set. All systems are trained on OzFish training set.

Scheme	Precision	Recall	F1 Score
ATP (Veiga et al., 2022)	86.0	66.0	74.7
ASRD (Marrable et al., 2022)	89.8	69.9	78.6
YOLO-fish (Muksit et al., 2022)	83.0	64.0	73.0
YOLOv11 (Khanam and Hussain, 2024)	82.8	83.3	83.1
Deepfins (Our proposed)	88.1	**91.9**	**90.0**

and training continues from the point where early stopping was previously triggered.. The training process for YOLOv11 takes approximately 18 h. The larger amount of training data provided by the *Detection_Set* contributes to a more stable training curve for YOLOv11, resulting in a significant improvement in the F1 Score of 83.1% on the test set. These results pertain to frame-based detection.

The *Clips_Set* dataset is utilized to develop the segmentation module, which is then integrated with the YOLOv11 input to generate spatial and motion-based features for fish detection, as illustrated in Fig. 5. To enhance the data, we perform histogram equalization on the Y channel after converting the frame from RGB to YCrCb and then back to RGB color-space. Fig. 6 shows the image enhancement after applying this pre-processing steps.

The segment size, denoted as K, is assessed across a range from 3 to 30, with increments of 1, and optimal results are achieved when it is set to 24. To eliminate background patterns, we calculate histogram values for each cluster and remove patterns based on their count. We experiment with histogram value thresholds in the range of 50,000 to 200,000, with step increments of 5000, and find that a threshold value greater than $\alpha_{\text{bkg}} = 100{,}000$ effectively segments background noise. Subsequently, we employ blob analysis and test blob area thresholds ranging from 2500 to 100,000, with step increments of 2500, ultimately setting the threshold to $\alpha_{\text{blb}} = 10{,}000$. All blobs in the output frame with contour areas greater than α_{blb} are extracted, effectively eliminating small detections caused by noisy pixels and background noise. Fish blobs from the segmentation module are integrated with the YOLOv11 input alongside raw images, serving the purposes of fish/non-fish classification and localization. The inference time is approximately 8 ms per crop of size 50 × 50 pixels. All blobs of a single frame, generated by the motion segmentation module, are passed to YOLOv11's classification head with a batch size of 64. Consequently, the inference time to process all blobs for the given frame is approximately 10–12 ms. In other words, each frame in a video recorded at 30 FPS requires 33 ms and in a video recorded at 60 FPS requires 16 ms to get processed. Therefore, can easily be handled by DeepFins in real-time. It is worth mentioning that OzFish and LifeCLEF 2015 datasets are recorded at 24 FPS and 25 FPS respectively. This performance highlights the model's capability to deliver rapid classification results, enabling real-time applications such as underwater monitoring or ecological surveys. The low latency ensures that the classification step does not become a bottleneck in the overall fish detection pipeline, making the approach suitable for scenarios requiring high-speed and accurate decision-making in dynamic underwater environments. In contrast temporal neural network models like recurrent neural networks (RNNs) or long short-term memory networks (LSTMs) require considerable inference compute cost in general. This results in an F1 Score of 90.0% on the test set, representing approximately a 7% improvement over standalone YOLOv11 results, as detailed in Table 1.

In contrast, the architectures proposed by Veiga et al. (2022), Marrable et al. (2022), and Muksit et al. (2022) achieve considerably lower F1 Scores in the context of fish detection on the OzFish dataset. F1 Scores presented in Tables 1 and 2 clearly demonstrate the effectiveness of the proposed DeepFins method for underwater fish detection. Compared to existing methods, DeepFins significantly outperforms other models such as YOLOv11 and ATP. For instance, DeepFins achieves a F1 Score of 90.0% on the OzFish test set, which is notably higher than YOLOv11 (83.1%) and ATP (74.7%). This improvement in performance, especially in terms of precision and recall, highlights the proposed models' ability to maintain a balanced detection approach, reducing both false positives and false negatives. Furthermore, high F1 Score suggests better handling of complexities in underwater environments, where varying lighting conditions, dynamic backgrounds, and occlusions often challenge fish detection models (Zhuang et al., 2017; Salman et al., 2019).

To ensure a fair comparison and underscore the significance of the proposed approach, a performance breakdown for each individual component of the architectural design is presented, confirming the benefits of each module as displayed in Table 3. As illustrated in Fig. 7, the output of fish detection algorithm reveals that the red boxes generated by the segmentation module successfully identify swiftly moving fish against a complex background, a task in which the spatial features of YOLOv11 (represented by the green boxes) sometimes fall short. Consequently, this contributes to the detection of more fish instances, as substantiated in Table 3.

In practical terms, this enhanced accuracy offers significant advantages in ecological monitoring applications, where it is crucial to detect and classify fish species accurately. For instance, the improved detection capability enables more reliable monitoring of fish populations

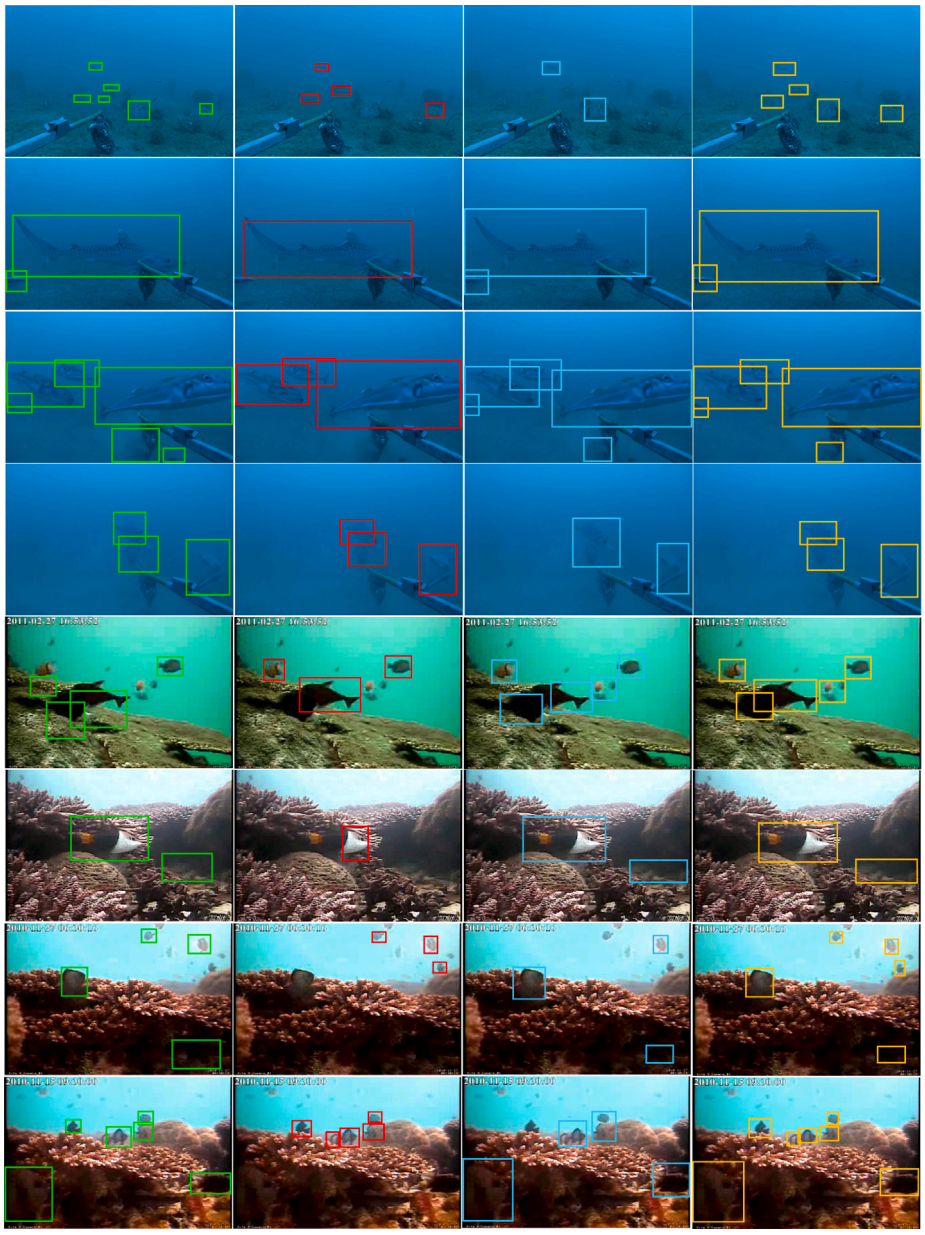


Fig. 7. Fish detection results from the proposed algorithm (best viewed in color). First column represent raw images and ground truth boxes. Second column are the fish detected by motion segmentation module. Third column are the outcomes of YOLOv11 while forth column are the results of the proposed hybrid system i.e., DeepFins. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Evaluation of model trained on Ozfish training set and tested on LifeCLEF 2015 test set and vice versa.

Training dataset	Test dataset	Scheme	Precision	Recall	F1 Score
Ozfish	LifeCLEF 2015	YOLOv11	77.4	73.2	75.2
		Deepfins	**79.1**	**81.4**	**80.2**
LifeCLEF 2015	Ozfish	YOLOv11	83.3	77.0	80.0
		Deepfins	**86.9**	**80.8**	**83.7**

Table 3
Ablation study on the proposed key components, evaluated with motion segmentation (MS), classifier head, YOLOv11 detector, and preferential merging strategy (PMS).

MS	Classifier head	YOLOv11	PMS	Precision	Recall	F1 Score
✓	×	×	×	31.0	3.2	5.2
✓	✓	×	×	42.6	4.2	7.3
×	×	✓	×	82.8	83.3	83.1
✓	✓	✓	✓	88.1	91.9	90.0

in natural ecosystems, allowing for better tracking of species diversity and health over time. With its robust performance across diverse datasets, such as OzFish and LifeCLEF 2015, DeepFins can be applied in large-scale underwater biodiversity assessments, facilitating long-term ecological research and conservation efforts Muksit et al. (2022). Furthermore, the ability to handle complex and dynamic underwater conditions makes DeepFins a valuable tool for real-time fish population monitoring, enabling marine researchers and environmental agencies to track ecosystem changes more effectively.

In a separate experiment, the proposed fish detection scheme, trained initially on the OzFish dataset, is evaluated on the LifeCLEF 2015 dataset to assess its generalization ability. Similarly, DeepFins is trained on the LifeCLEF 2015 dataset and then evaluated on the OzFish dataset, completing a round of cross-dataset validation. This approach

allows testing the model's robustness across different datasets, showing whether it can effectively detect fish species even when trained on a distinct set of images and ecological conditions. The performance of the proposed scheme in this context is shown in Table 2, highlighting its ability to adapt to unseen data.

Importantly, in both experiments, the parameter configurations for YOLOv11 and the motion segmentation module remain consistent. This consistency ensures that any performance differences observed between datasets stem from the datasets themselves, not from changes in the model's settings. The parameter values are optimized using validation sets from both the OzFish and LifeCLEF 2015 datasets, ensuring fair and unbiased performance evaluation. By using cross-validation, the model's ability to perform well on diverse datasets without overfitting to any specific data set is ensured, reinforcing the robustness of the proposed method for broader applications. This cross-dataset evaluation not only validates the proposed scheme's generalization ability but also demonstrates the model's adaptability to different conditions in ecological environments, making it a valuable tool for monitoring fish populations across various underwater habitats Veiga et al. (2022).

For computations and training deep networks, an Intel Core i5 processor with 32 GB RAM and an Nvidia 1080Ti GPU is used.

4. Discussion

4.1. A general overview

Recently, numerous computer vision and machine learning algorithms have emerged, showing promise in various fish-related tasks, addressing challenges such as automatic fish detection, species classification, tracking, and biomass estimation. These cutting-edge endeavours predominantly rely on advanced deep learning algorithms (Kandimalla et al., 2022; Knausgård et al., 2022). However, a research gap persists in addressing the impact of environmental factors on the performance of these systems. On one hand, state-of-the-art object detection algorithms can excel at localizing static fish based on their textural patterns, while, on the other hand, they may struggle to detect moving fish, particularly in cases of poor visibility in complex, uncontrolled underwater datasets.

In this study, we have achieved state-of-the-art performance for the fish detection task, particularly on the challenging OzFish dataset. The proposed approach involves a novel algorithm for temporal or motion-based feature extraction, which is seamlessly integrated with spatial features through the YOLOv11 deep architecture in the proposed hybrid machine vision pipeline (depicted in Fig. 4).

The temporal features, essential for fish detection, are obtained by applying motion segmentation and background subtraction simultaneously. This operation allows us to filter out irrelevant background pixels with lower motion magnitudes, while retaining other instances as potential fish candidates (as seen in Fig. 5). However, it is important to note that this background suppression approach may encounter challenges when dealing with the vigorous back-and-forth motion of aquatic plants and dynamic lighting. In such cases, they can create noises that mimic foreground objects, including fish in our context. To filter out such noises, we implement binary classification for distinguishing fish from non-fish objects within a multi-objective optimization framework using a modified YOLOv11 architecture. In addition to the detector branch, we introduce a classification branch after the DarkNet (Khanam and Hussain, 2024) feature extractor in the YOLOv11 backbone as shown in Fig. 4. This classification branch is created by combining a Conv2D layer, a max pooling layer, global average pooling, and two dense layers.

4.2. Justifying the proposed model: DeepFins

The standard configuration of YOLOv11 struggles with accurately detecting fast-moving fish, especially when these fish are located at a distance from the camera or lack clear texture or shape due to factors like water murkiness or camouflage. However, this limitation is effectively addressed by the proposed temporal pipeline, which excels in detecting moving fish, as shown in Fig. 8. The first column displays the original images, the second column presents the results of the temporal pipeline, and the third column showcases the output of the YOLOv11 pipeline. It is evident that the proposed motion segmentation module preserves the shape-related features of fish, whereas YOLOv11 often fails to capture similar information due to the fast-moving parts of the fish, such as caudal, anal, dorsal, and pelvic fins. As a result, the feature heatmaps generated by YOLOv11 tend to have distorted shapes, sometimes leading to misdetection or confusion with background objects like coral reefs and aquatic plants.

Fig. 9 illustrates a more challenging scenario involving videos with fast-moving fish. In this case, the motion segmentation module still attempts to capture fish instances, while YOLOv11 completely fails to detect the fish, instead identifying random background objects as legitimate fish. Therefore, in the proposed hybrid approach, the motion segmentation and YOLOv11 pipelines complement each other, significantly boosting the overall F1 Score of the algorithm. This synergy lays the foundation for potential applications, including assemblage and biomass estimation with tracking, which is planned for future work.

Nevertheless, there are instances where fish within the frame are not detected by the proposed algorithm due to challenges such as camouflage, a confusing background, and the static posture of fish. Fig. 10 shows some particularly challenging examples (hard negative samples) where fish instances are not detected by the model. The yellow boxes represent ground truth fish locations that are missed by the algorithm; in some cases, these are also instances inaccurately labeled as fish in the original ground truth files provided by AIMS, adding further complexity to the detection task.

In the empirical evaluation aimed at identifying the most suitable deep architecture, several popular backbone networks, including ResNet-18, ResNet-50 (He et al., 2016), ResNeXt (Xie et al., 2017), and MobileNetV3 (Howard et al., 2019), are trained. These architectures are coupled with two different object detection frameworks: SSD and Faster R-CNN, which utilizes RPN.

Despite exploring a wide range of architectures, it is found that these models are consistently outperformed by the YOLOv11 detector. YOLOv11 not only offers faster inference times but also excels in terms of detection accuracy when compared to the SSD and RPN-based models. Fig. 11 illustrates how the deep network, YOLOv11, extracts relevant fish-dependent features for detection. The initial layers capture high-level features such as fish edges and overall structural patterns, while the middle and top layers meticulously focus on features unique to the fish shape, including the snout, tail, and fins. The first and second rows of Fig. 11 display the standard and zoomed-in images, respectively, while the third row shows a fish-less background where the network's activations are random without any informative pattern. These static or very slow-moving fish are well detected by the YOLOv11 system. On the other hand, for relatively fast-moving fish, where the textural pattern is unclear due to motion blurriness in the images and cannot be effectively learned by a deep architecture, the proposed motion segmentation algorithm successfully captures these fish instances while suppressing the dynamic underwater background. Motion-based features not only help in detecting fish shapes but also enhance overall detection. Fig. 12 illustrates the proposed concept of combining motion-based and static features of fish instances to achieve final detection.

Given the complexity of the dataset and the image quality, as shown in Fig. 1, the detection and classification of fish faces numerous challenges. Various sources of variation, particularly dynamic backgrounds

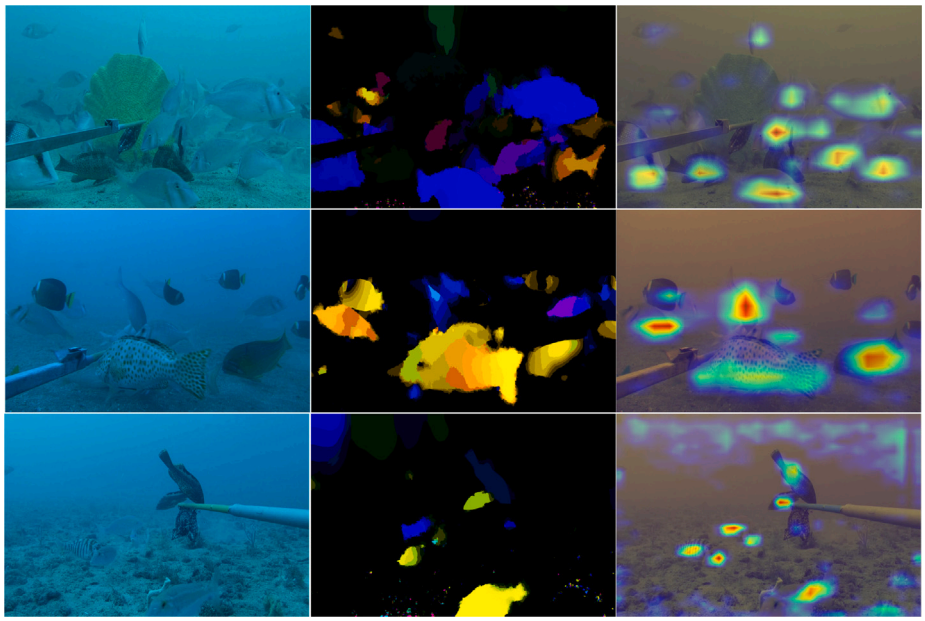


Fig. 8. Comparison of motion-segmentation and YOLOv11 feature extraction. The superior performance of the temporal branch (second column) is evident in accurately identifying fish shapes compared to the YOLOv11 branch (last column). The YOLOv11 branch often misses fish body parts that are moving rapidly or blend in with the dynamic background.

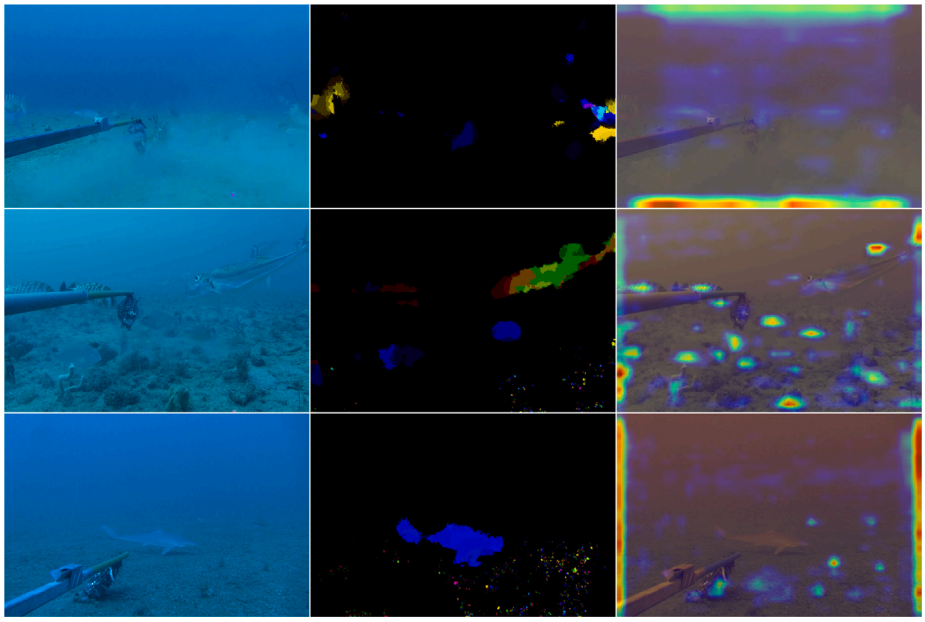


Fig. 9. Example images showing the temporal branch successfully identifying fast-moving fish (second column) that the YOLOv11 branch completely overlooks (last column). Combining both methods (hybrid approach) leads to a significant improvement in overall fish detection accuracy (F1 Score).

and unclear fish textures, compound the nonlinearity of the data (LeCun et al., 2004). Conventional machine learning and computer vision algorithms, which tend to be shallow and linear, are often struggled with addressing this issue effectively. Therefore, the ideal approach for this problem is involving the use of highly nonlinear deep neural networks, augmented with temporal information, to achieve the desired results.

The novelty of the approach is primarily centered around the specially designed architecture dedicated to fish detection, a vital task with broad applications in areas like fish assemblage and biomass estimation. These applications, in turn, have direct implications for the overall health of the ecosystem in the specific sampling region. In a separate experiment, the effectiveness of the DeepFins is demonstrated by testing it on the LifeCLEF 2015 dataset. Additionally, the complete model is trained on the LifeCLEF 2015 dataset and tested on the OzFish dataset. In both cases, the performance of the proposed system is promising and outperforms the baseline benchmark of YOLOv11.

Furthermore, the data augmentation methodology, involving the manual labeling of unannotated frames, yields favorable results. It is important to note that the lower performance observed in the algorithms mentioned by Marrable et al. (2022) and Muksit et al. (2022) may be attributed to the absence of data augmentation. Data augmentation is crucial for effectively training or fine-tuning a large network like YOLO. In contrast, Veiga et al. (2022) utilizes data augmentation, but even with similar techniques, their performance is unlikely to surpass that of vanilla YOLOv11 because they employed YOLOv4 and YOLOv5 architectures (Terven and Cordova-Esparza, 2023) without significant modifications.

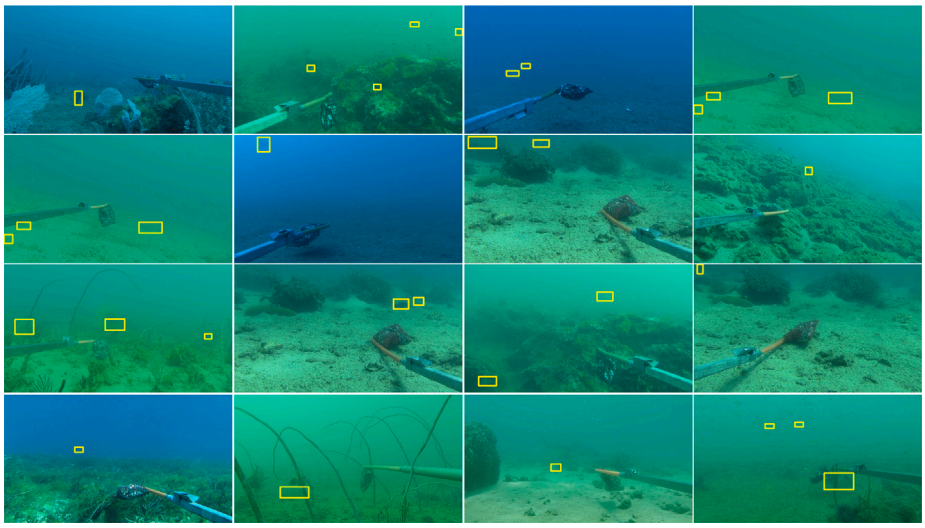


Fig. 10. Hard negative examples where the model is unable to find any fish in a given frame. Yellow boxes (best viewed in color) represent ground truths in respective frames. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

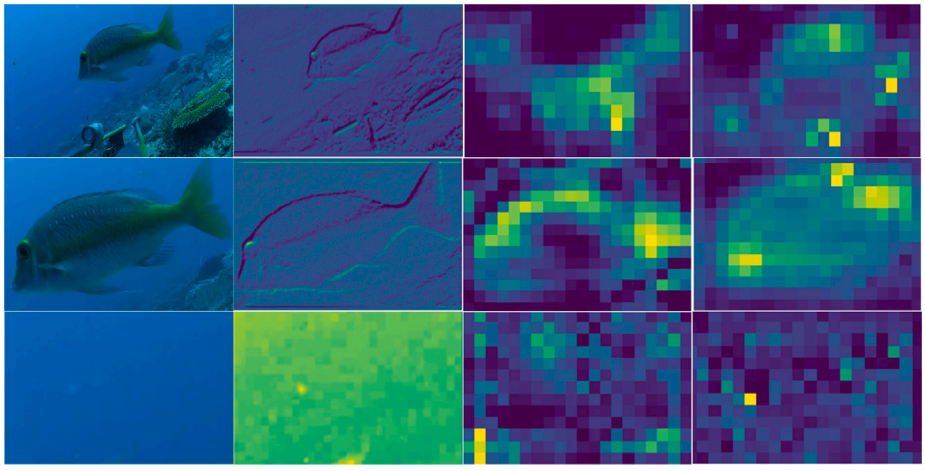


Fig. 11. Features extracted from the modified YOLOv11 model. The first column displays the RGB input images, while the second column exhibits the feature maps extracted from the initial layer of the Darknet backbone. In the third column, observe mid-level feature maps can be observed, and the last column showcases high-level feature maps obtained from the model's final layers.

4.3. Comparative analysis

There are several approached available in the literature that try to solve fish detection challenge wich varying experimental setup. For example, Jahanbakht et al. (2023) introduce the FishInTurbidWater dataset, which primarily focuses on turbid water conditions, while Rani et al. (2024) present the Aquatic WeightNet dataset, featuring 5000 images of 30 fish species from Genetically Improved Farmed Tilapia, captured in a controlled fish tank environment. Zheng et al. (2024) collect and annotate the Fish-Seg Dataset, comprising 1800 images from five species of farm-raised fish. Cao et al. (2024) use light-colored backdrop plate to capture 3849 images of 3–6 months old grass carp fish. The dataset is taken from the Guanqiao Breeding Base of the Institute of Hydrobiology, Chinese Academy of Sciences in an indoor environment. Similarly, Saqib et al. (2024) propose a dataset using electronic monitoring data for 12 fish species of Australian longline vessels. They annotate 20,019 fish instances from 5198 images. However, these datasets do not address the challenges associated with deep unconstrained underwater environments, such as the unbalanced number of samples of fish species in dataset which poses a daunting challenge for any deep learning model to train well, coral reefs, moving aquatic plants, algae, and varying illumination conditions. These factors significantly complicate fish detection by increasing false alarms and reducing overall accuracy. The OzFish dataset comes with all of these challenges, provides over 3000 videos featuring more than 500 fish species where more than 50 fish species have less than 10 samples, 200 species having less than 100 samples compared to the thousands for other species. Moreover, the data is recorded at diverse locations and times, offering a more comprehensive representation of unconstrained underwater scenarios.

Jahanbakht et al. (2023) employs an ensemble of well-established models, including EfficientNet and Vision Transformers, to validate their proposed dataset. Similarly Rani et al. (2024) focus on developing a fish biomass estimator for turbid environments, utilizing a YOLOv8-based fish detector combined with regression modules for biomass estimation. Zheng et al. (2024) introduce a novel approach to fish detection by integrating video-based object segmentation (VOS) with YOLOv7. They demonstrate that combining VOS with deep learning models effectively isolates individual fish from dynamic backgrounds, removing interfering factors and significantly enhancing fish recognition. Similarly, Cao et al. (2024) propose a custom dimension measurement method for fish using an optimized YOLOv5-keypoint framework with a multi-attention SimAM mechanism, achieving more accurate

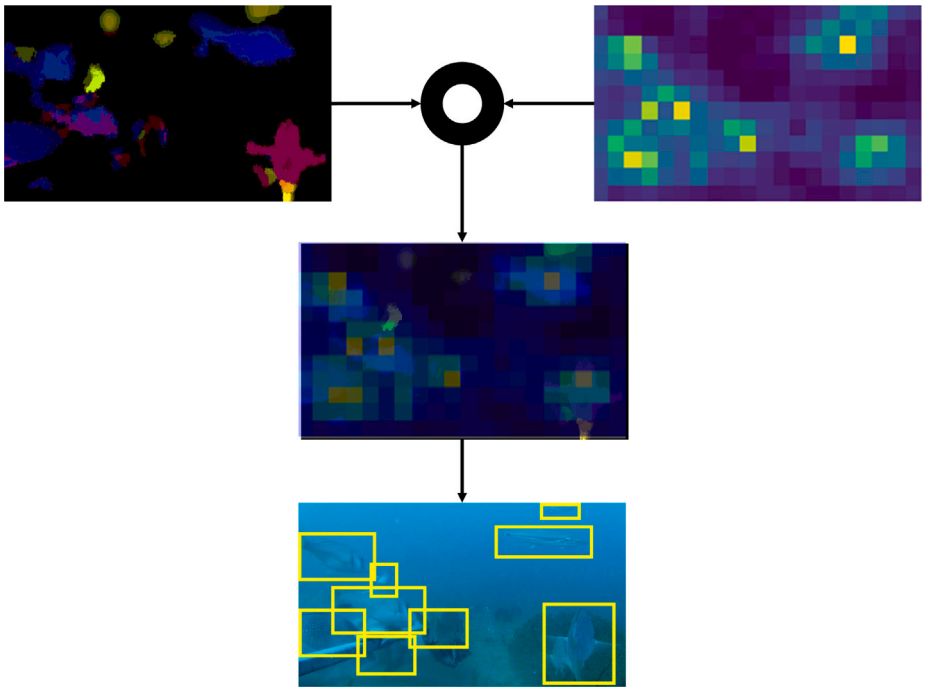


Fig. 12. Features from the temporal branch (top left) of the model are merged with the high-level feature maps of the YOLOv11 architecture (top right) and then fish blobs are selected based on the confidence score (best viewed in color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and faster fish detection, which improves measurement precision. Saqib et al. (2024) present a YOLOX-based deep learning framework for automatic fish detection, event counting, and species classification in electronic monitoring (EM) videos using computer vision and AI. However, the evaluations of these methods are confined to their respective datasets, leaving their generalization capabilities untested across diverse scenarios. Furthermore, these approaches focus solely on spatial features and do not incorporate temporal features, which are crucial in unconstrained underwater environments where fish move freely against dynamic backgrounds. In contrast, our approach includes a novel temporal detector combined with a modified YOLOv11 spatial feature extractor, which is validated on the OzFish dataset. Additionally, it is cross-validated on the LifeCLEF 2015 dataset, which exhibits visual differences compared to the OzFish dataset. Sun et al. (2022) applies an image enhancement technique to the LifeCLEF 2015 dataset to enhance fish visualization. They employ Cascade R-CNN for fish detection, resulting in a commendable F1 Score of 81%. However, it is essential to note that using image enhancement techniques alone introduces background noise, which, in turn, diminishes accuracy and consequently leads to a decrease in the overall F1 Score. On the other hand, the hybrid approach effectively identifies fast-moving fish, suppresses background noise, and eliminates outliers through a dedicated fish instance classifier. As a result, this method achieves an F1 Score that is more than 11% higher on the OzFish dataset. Notably, the cross-validation results on the LifeCLEF 2015 dataset demonstrate a 1% difference of F1 Score than the approach by Sun et al. (2022), underscoring the significance of motion detection and improved feature extraction in fish detection tasks.

Deep learning-based fish segmentation techniques for underwater videos have recently been embraced by researchers. For example, the Mask R-CNN model is employed by Garcia et al. (2020), while ResNet, YOLOACT, and Cascade R-CNN models are utilized by Alshdaifat et al. (2020) for fish segmentation. The datasets used in these studies are either collected in controlled environments or lack the challenges posed by issues such as illumination and contrast variations, blurriness, turbidity, camouflage, occlusion, jitter, and the presence of moving background objects. These challenges are notably present in the OzFish and LifeCLEF 2015 datasets. Their investigations reveal that the proposed methodologies produce false alarms and misclassifications, primarily attributable to dynamic backgrounds and overlapping fish instances. Systems employing such techniques are expected to significantly underperform when faced with the aforementioned challenging elements, resulting in a high number of errors and, consequently, reduced model precision.

The A-LCFCN (Affinity-based Fully Convolutional Neural Network) model is introduced by Laradji et al. (2021), which comprises two branches. One branch produces per-pixel scores, while the other generates an affinity matrix. This novel model is evaluated on the demanding DeepFish dataset. The results demonstrate an IOU (Intersection over Union) of 0.73 for foreground objects and an impressive 0.99 for background objects. However, the low IOU for foreground objects indicates that too many false alarms and misclassifications are generated during the segmentation of fish instances, resulting in a decrease in the overall accuracy of the system.

The research gap in the literature advocates a balanced spatio-temporal fish detection approach. The proposed approach benefits from the availability of effective and efficient deep CNN like YOLOv11 and especially crafted motion-segmentation module to mitigate the weaknesses of each other.

4.4. Ablation study

We perform a comprehensive ablation study on the OzFish dataset to analyze the contributions of individual components in our proposed approach. This study evaluates the impact of the motion segmentation module (MS), the classifier head, the YOLOv11 detector, and the preferential merging strategy (PMS) integrated into the Deep Fins framework. The configurations explored include using motion segmentation alone, which relies solely on the module without any classifier or spatial feature-based detector. Another configuration combines motion segmentation with a classifier head to distinguish fish from non-fish blobs, effectively reducing false alarms caused by moving aquatic plants and dynamic backgrounds. A further setup involves using only the YOLOv11 detector, which focuses on spatial feature extraction without

incorporating temporal features or merging strategies. Finally, the Deep Fins framework integrates all components, combining outputs from motion segmentation, the classifier head, and YOLOv11 detections through a preferential merging strategy.

The results, presented in Table 3, highlight the precision, recall, and F1 Score achieved by each configuration, illustrating the impact of individual components on detection performance. Using motion segmentation alone shows the lowest performance due to noise and the misclassification of non-fish blobs, particularly in dynamic backgrounds. Performance improves significantly when the classifier head is enabled, as it effectively filters out false alarms and irrelevant background blobs caused by contrast and illumination variations. Employing the YOLOv11 detector alone achieves high precision and competitive recall, but its performance is challenged in detecting fast-moving fish and in turbid water conditions with low visibility. The best results are achieved when all components are integrated in the Deep Fins framework. This comprehensive configuration demonstrates superior performance across all metrics, showcasing the effectiveness of combining motion and spatial features through a preferential merging strategy. This approach leverages the strength of motion segmentation to detect fast-moving fish in low-visibility regions while capitalizing on YOLOv11's precise fish detection capabilities in most scenarios. The preferential merging strategy prioritizes YOLOv11 detections in overlapping regions while retaining unique fish blobs from motion segmentation, reducing false positives and ensuring consistent fish detection.

This study confirms that the integration of spatial and temporal features significantly enhances underwater fish detection. While each individual component contributes to improved performance, their combined use in the Deep Fins framework consistently outperforms standalone methods, proving highly effective for addressing the unique challenges posed by underwater environments.

4.5. Challenges for further studies

Expanding the proposed fish detection method to include other fish-related tasks like fish classification and further automatic biomass estimation involves addressing several challenges that arise in underwater environments. One of the primary challenges is the variation in fish species, their size, scale, and orientation, all of which can impact the effectiveness of classification models. In underwater videos, the visibility of fish is often limited due to factors like occlusion, lighting, and water turbidity. These variations can make it difficult for traditional models to differentiate between species, especially when they share similar visual features Veiga et al. (2022) and Dai et al. (2024). Moreover, the presence of small, fast-moving fish or partially visible fish adds another layer of complexity. To overcome these issues, the integration of spatio-temporal features, as envisioned for the future work, can help capture both the spatial characteristics of fish and their movement over time, thus improving classification accuracy under challenging conditions.

A major barrier in fish classification is the lack of large, labeled datasets for rare or poorly represented fish species, which is often a limitation in ecological research. Collecting and labeling sufficient data is both time-consuming and expensive, which hinders the development of robust classification models. To tackle this, techniques such as semi-supervised learning or few-shot learning can be employed. These methods leverage a small amount of labeled data along with a larger pool of unlabeled data, enhancing the model's ability to classify species without extensive labeling efforts Radford et al. (2021). Additionally, self-supervised learning techniques, such as those used in DINOv2, can help the model learn useful features from the data itself, without relying heavily on annotated labels.

The dynamic and complex nature of underwater environments also poses a significant challenge for fish classification. Factors like fluctuating lighting conditions, water movement, and varying turbidity can drastically affect image quality and visibility. To address these issues, the future development of a multi-modal model that captures both spatial and temporal information is crucial. By processing both static image features and temporal motion patterns, this model can improve its ability to distinguish between species, even when visual cues are obscured by environmental conditions Marrable et al. (2022). This multi-modal approach would enable the model to adapt better to changes in the environment, thus enhancing its performance in diverse underwater habitats.

Scalability is another challenge when expanding to fish classification, particularly when dealing with a large number of species across different ecological conditions. One way to address this is through class-incremental learning, which allows models to progressively learn new classes without forgetting the previously learned ones. This method helps the model maintain its performance as new species are introduced, without requiring retraining from scratch Radford et al. (2021). By adopting this technique, the proposed fish detection and classification pipeline can remain versatile and adaptable, capable of handling an expanding set of species while still performing well in real-world scenarios.

5. Conclusion

A novel fish detection method incorporating spatio-temporal information is proposed to extract fish-specific features. Previous works generally suffer from performance limitations due to the unconstrained environmental variations in the videos and the inability to detect less visible moving fish. To address these challenges, motion-based segmentation of fish and background subtraction steps are combined to complement the static, texture-based fish detection in a modified manner. The challenges of fish classification can be addressed by combining existing technologies such as self-supervised learning, multi-modal models, and incremental learning. These methods can help overcome issues such as limited labeled data, complex underwater conditions, and the scalability of models to accommodate a growing number of fish species. With these advancements, the proposed approach can be refined to accurately monitor and classify fish species across diverse and dynamic aquatic environments. Additionally, work is planned to be extended to include fish species classification, allowing the entire pipeline to be used for fish biomass estimation, which is another important research avenue to explore using machine learning and computer vision.

CRediT authorship contribution statement

Ahsan Jalal: Methodology, Investigation. **Ahmad Salman:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis. **Ajmal Mian:** Writing – review & editing, Resources. **Salman Ghafoor:** Writing – review & editing, Formal analysis. **Faisal Shafait:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Professor Ajmal Mian (co-author) is the recipient of an Australian Research Council Future Fellowship Award (project number FT210100268) funded by the Australian Government. This grant supported the research work carried out in this article.

Appendix. Mathematical modelling of motion segmentation module

Starting with Horn–Schunck constraint (Wohn et al., 1983) over the consecutive video frames, the brightness and change of pixel location/motion over a short time interval is negligible, therefore, rate of spatial, temporal and motion change define brightness constraint i.e., for video frames with pixel intensity I, defined in 2D coordinates (i, j) in x a y direction and separated by time t,

$$\frac{\partial I(i,j,t)}{\partial x}u(i,j,t) + \frac{\partial I(i,j,t)}{\partial y}v(i,j,t) + \frac{\partial I(i,j,t)}{\partial t} = 0 \tag{A.1}$$

Similarly, another constraint over smooth motion implies that the average motion of pixels in 2D space over immediate neighborhood is not abrupt i.e.,

$$\frac{1}{2}\left[\left(u(i,j)-u(i+1,j)\right)^2 + \left(u(i,j)-u(i,j+1)\right)^2 + \left(v(i,j)-v(i+1,j)\right)^2 + \left(v(i,j)-v(i,j+1)\right)^2\right] = 0 \tag{A.2}$$

Using Eqs. (A.1) and (A.2), two loss functions are defined for optimization problem i.e., loss functions w.r.t brightness L_B as well as smoothness constraint L_S,

$$L_B = \left[\frac{\partial I(i,j,t)}{\partial x}u(i,j,t) + \frac{\partial I(i,j,t)}{\partial y}v(i,j,t) + \frac{\partial I(i,j,t)}{\partial t}\right]^2 \tag{A.3}$$

$$L_S = \frac{1}{2}\left[\left(u(i,j)-u(i+1,j)\right)^2 + \left(u(i,j)-u(i,j+1)\right)^2 + \left(v(i,j)-v(i+1,j)\right)^2 + \left(v(i,j)-v(i,j+1)\right)^2\right] \tag{A.4}$$

The overall loss function is given as

$$E = L_B + \lambda L_s \tag{A.5}$$

Eqs. (A.3) and (A.4) generate an optimization problem to solve $\min_{u,v}\sum\{L_B + \lambda L_s\}$ with motion vectors $u_{i,j}$ and $v_{i,j}$. Therefore, using gradient descent, the estimates of motion vectors $u(i, j)$ and $v(i, j)$ are obtained by solving $\frac{\partial E}{\partial u_{i,j}} = 0$ and $\frac{\partial E}{\partial v_{i,j}} = 0$

$$u_{i,j} = \bar{u}_{i,j} - \frac{I_x\bar{u}_{i,j} + I_y\bar{v}_{i,j} + I_t}{1 + \lambda(I_x^2 + I_y^2)}I_x, \tag{A.6}$$

$$v_{i,j} = \bar{v}_{i,j} - \frac{I_x\bar{u}_{i,j} + I_y\bar{v}_{i,j} + I_t}{1 + \lambda(I_x^2 + I_y^2)}I_y, \tag{A.7}$$

where $I_x = \frac{\partial I(i,j,t)}{\partial x}$, $I_y = \frac{\partial I(i,j,t)}{\partial y}$ and $I_t = \frac{\partial I(i,j,t)}{\partial t}$ as defined earlier. The average motion of pixel in 2D over the neighboring pixels is given by $\bar{u}_{i,j} = \frac{1}{2}\{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}\}$ and $\bar{v}_{i,j} = \frac{1}{2}\{v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1}\}$ respectively. The right-hand side of Eqs. (A.6) and (A.7) are known variables and can be calculated using consecutive video frames.

Now, the magnitude of the motion in 2D space for two consecutive video frames is defined as

$$M_{i,j} = \sqrt{(u_{i,j} + v_{i,j})^2}. \tag{A.8}$$

The segmentation of moving objects can be realized by dividing the magnitude image $M_{i,j}$ into K segments i.e., $m_{i,j}^k \in M_{i,j}$, $k = 1, 2, 3, \ldots, K$ and using a threshold (α_{bkg} and α_{blb}) bounded constraint i.e., for each segment k,

$$m_{i,j}^k = \max(m_{i,j}^k - \alpha_{\text{bkg}}, 0). \tag{A.9}$$

Now, a blob $B_k(i, j)$ is a moving fish candidate and is calculated as

$$B_k(i,j) = m_{i,j}^k \iff \left\|C_k - m_{i,j}^k\right\| < \left\|C_k - m_{i,j}^l\right\| \quad \forall k \neq l, \tag{A.10}$$

where C_k is the average motion magnitude in the segment $k = 1, 2, 3, \ldots, K$ and given as

$$C_k = \frac{1}{N}\left(\sum_{i=0}^{|i|}\sum_{j=0}^{|j|}\sqrt{(u_{i,j}^k)^2 + (v_{i,j}^k)^2}\right) = \frac{1}{N}\left(\sum_{i=0}^{|i|}\sum_{j=0}^{|j|}m_{i,j}^k\right). \tag{A.11}$$

where N is the total number of pixels in segment $m_{i,j}^k$ i.e., $N = |\, i \,||\, j \,|$. Finally,

$$[B_k(i,j)\,\text{exists}] = [|B_k(i,j)| > \alpha_{\text{blb}}]. \tag{A.12}$$

In Eqs. (A.11) and (A.12), the notation $|\cdot|$ signifies the count of elements. The parameter α_{bkg} denotes the minimum threshold for the most frequently occurring motion magnitudes within the background and can be determined by analyzing the histogram of $M_{i,j}$. Conversely, α_{blb} serves as a threshold for the minimum area of a segment or blob exhibiting rapid motion. Therefore, small magnitudes in $M_{i,j}$ typically correspond to static objects or the background, and they are generally smaller in magnitude than those of moving objects. Put differently, faster-moving entities such as fish tend to generate larger motion magnitudes compared to stationary or slow-moving objects.

Eq. (A.9) produces the kth image segment $m_{i,j}^k$, achieved through thresholding, which effectively eliminates static objects, including the background. The image segments $m_{i,j}^k$ are then further processed to identify distinct regions or blobs denoted as $B_k(i, j)$ in Eqs. (A.10) and (A.12).

Data availability

Data will be made available on request.

References

Alshdaifat, N.F.F., Talib, A.Z., Osman, M.A., 2020. Improved deep learning framework for fish segmentation in underwater videos. Ecol. Inform. 59, 101121.

Cao, D., Guo, C., Shi, M., Liu, Y., Fang, Y., Yang, H., Cheng, Y., Zhang, W., Wang, Y., Li, Y., et al., 2024. A method for custom measurement of fish dimensions using the improved YOLOv5-keypoint framework with multi-attention mechanisms. Water Biol. Secur. 3 (4), 100293.

Choi, S., 2015. Fish identification in underwater video with deep convolutional neural network: Snumedinfo at lifeclef fish task 2015.. In: CLEF (Working Notes). pp. 1–10.

Dai, K., Shao, J., Gong, B., Jing, L., Chen, Y., 2024. CLIP-FSSC: A transferable visual model for fish and shrimp species classification based on natural language supervision. Aquac. Eng. 107, 102460.

Garcia, R., Prados, R., Quintana, J., Tempelaar, A., Gracias, N., Rosen, S., Vågstøl, H., Løvall, K., 2020. Automatic segmentation of fish using deep learning with application to fish size measurement. ICES J. Mar. Sci. 77 (4), 1354–1366.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al., 2019. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1314–1324.

Huang, P.X., Boom, B.J., Fisher, R.B., 2015. Hierarchical classification with reject option for live fish recognition. Mach. Vis. Appl. 26 (1), 89–102.

Huang, X., Huang, P.X., 2014. Balance-guaranteed optimized tree with reject option for live fish recognition.

Jäger, J., Rodner, E., Denzler, J., Wolff, V., Fricke-Neuderth, K., 2016. Seaclef 2016: Object proposal classification for fish detection in underwater videos. In: CLEF (Working Notes). pp. 481–489.

Jahanbakht, M., Azghadi, M.R., Waltham, N.J., 2023. Semi-supervised and weakly-supervised deep neural networks and dataset for fish detection in turbid underwater videos. Ecol. Inform. 78, 102303.

Jalal, A., Salman, A., Mian, A., Shortis, M., Shafait, F., 2020. Fish detection and species classification in underwater environments using deep learning with temporal information. Ecol. Inform. 57, 101088.

Jennings, S., Kaiser, M.J., 1998. The effects of fishing on marine ecosystems. In: Advances in Marine Biology. Vol. 34, Elsevier, pp. 201–352.

Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.-P., Planqué, R., Rauber, A., Palazzo, S., Fisher, B., et al., 2015. LifeCLEF 2015: multimedia life species identification challenges. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 6th International Conference of the CLEF Association, CLEF'15, Toulouse, France, September 8-11, 2015, Proceedings 6. Springer, pp. 462–483.

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.

Khanam, R., Hussain, M., 2024. Yolov11: An overview of the key architectural enhancements. arXiv preprint arXiv:2410.17725.

Knausgård, K.M., Wiklund, A., Sørdalen, T.K., Halvorsen, K.T., Kleiven, A.R., Jiao, L., Goodwin, M., 2022. Temperate fish detection and classification: a deep learning based approach. Appl. Intell. 1–14.

Laradji, I.H., Saleh, A., Rodriguez, P., Nowrouzezahrai, D., Azghadi, M.R., Vazquez, D., 2021. Weakly supervised underwater fish segmentation using affinity LCFCN. Sci. Rep. 11 (1), 17379.

Lawson, G.L., Barange, M., Fréon, P., 2001. Species identification of pelagic fish schools on the South African continental shelf using acoustic descriptors and ancillary information. ICES J. Mar. Sci. 58 (1), 275–287.

LeCun, Y., Huang, F.J., Bottou, L., 2004. Learning methods for generic object recognition with invariance to pose and lighting. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.. Vol. 2, IEEE, pp. II–104.

Li, X., Zhao, S., Chen, C., Cui, H., Li, D., Zhao, R., 2024. YOLO-FD: An accurate fish disease detection method based on multi-task learning. Expert Syst. Appl. 258, 125085.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer, pp. 21–37.

Marrable, D., Barker, K., Tippaya, S., Wyatt, M., Bainbridge, S., Stowar, M., Larke, J., 2022. Accelerating species recognition and labelling of fish from underwater video with machine-assisted deep learning. Front. Mar. Sci. 9, 944582.

McLaren, B.W., Langlois, T.J., Harvey, E.S., Shortland-Jones, H., Stevens, R., 2015. A small no-take marine sanctuary provides consistent protection for small-bodied by-catch species, but not for large-bodied, high-risk species. J. Exp. Mar. Biol. Ecol. 471, 153–163.

Muksit, A.A., Hasan, F., Emon, H.B., Fahad, M., Haque, M.R., Anwary, A.R., Shatabda, S., 2022. YOLO-fish: A robust fish detection model to detect fish in realistic underwater environment. Ecol. Inform. 72.

Palazzo, S., Murabito, F., 2014. Fish species identification in real-life underwater images. In: Proceedings of the 3rd ACM International Workshop on Multimedia Analysis for Ecological Data. pp. 13–18.

Qu, H., Wang, G.-G., Li, Y., Qi, X., Zhang, M., 2024. ConvFishNet: An efficient backbone for fish classification from composited underwater images. Inform. Sci. 121078.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 2021. Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. PMLR, pp. 8748–8763.

Radinger, J., Britton, J.R., Carlson, S.M., Magurran, A.E., Alcaraz-Hernández, J.D., Almodóvar, A., Benejam, L., Fernández-Delgado, C., Nicola, G.G., Oliva-Paterna, F.J., et al., 2019. Effective monitoring of freshwater fish. Fish Fish. 20 (4), 729–747.

Rani, S.J., Ioannou, I., Swetha, R., Lakshmi, R.D., Vassiliou, V., 2024. A novel automated approach for fish biomass estimation in turbid environments through deep learning, object detection, and regression. Ecol. Inform. 102663.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. 28.

Saleh, A., Sheaves, M., Jerry, D., Azghadi, M.R., 2022. Unsupervised fish trajectory tracking and segmentation. arXiv preprint arXiv:2208.10662.

Salman, A., Jalal, A., Shafait, F., Mian, A., Shortis, M., Seager, J., Harvey, E., 2016. Fish species classification in unconstrained underwater environments based on deep learning. Limnol. Oceanogr.: Methods 14 (9), 570–585.

Salman, A., Maqbool, S., Khan, A.H., Jalal, A., Shafait, F., 2019. Real-time fish detection in complex backgrounds using probabilistic background modelling. Ecol. Inform. 51, 44–51.

Saqib, M., Khokher, M.R., Yuan, X., Yan, B., Bearham, D., Devine, C., Untiedt, C., Cannard, T., Maguire, K., Tuck, G.N., et al., 2024. Fishing event detection and species classification using computer vision and artificial intelligence for electronic monitoring. Fish. Res. 280, 107141.

Sun, H., Yue, J., Li, H., 2022. An image enhancement approach for coral reef fish detection in underwater videos. Ecol. Inform. 72, 101862.

Sung, M., Yu, S.-C., Girdhar, Y., 2017. Vision based real-time fish detection using convolutional neural network. In: OCEANS 2017-Aberdeen. IEEE, pp. 1–6.

Terven, J., Cordova-Esparza, D., 2023. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. arXiv preprint arXiv:2304.00501.

Ubina, N., Cheng, S.-C., Chang, C.-C., Chen, H.-Y., 2021. Evaluating fish feeding intensity in aquaculture with convolutional neural networks. Aquac. Eng. 94, 102178.

Veiga, R.J., Ochoa, I.E., Belackova, A., Bentes, L., Silva, J.P., Semião, J., Rodrigues, J.M., 2022. Autonomous temporal pseudo-labeling for fish detection. Appl. Sci. 12 (12), 5910.

Wang, G., Muhammad, A., Liu, C., Du, L., Li, D., 2021. Automatic recognition of fish behavior with a fusion of RGB and optical flow data based on deep learning. Animals 11 (10), 2774.

Wang, G., Yu, J., Xu, W., Muhammad, A., Li, D., 2025. Automated fish counting system based on instance segmentation in aquaculture. Expert Syst. Appl. 259, 125318.

Wohn, K., Davis, L.S., Thrift, P., 1983. Motion estimation based on multiple local constraints and nonlinear smoothing. Pattern Recognit. 16 (6), 563–570.

Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1492–1500.

Xu, W., Matzner, S., 2018. Underwater fish detection using deep learning for water power applications. In: 2018 International Conference on Computational Science and Computational Intelligence. CSCI, IEEE, pp. 313–318.

Zheng, T., Wu, J., Kong, H., Zhao, H., Qu, B., Liu, L., Yu, H., Zhou, C., 2024. A video object segmentation-based fish individual recognition method for underwater complex environments. Ecol. Inform. 82, 102689.

Zhuang, P., Xing, L., Liu, Y., Guo, S., Qiao, Y., 2017. Marine animal detection and recognition with advanced deep learning models. In: CLEF (Working Notes).