Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 1, 67–79
DOI: 10.61822/amcs-2026-0006

REAL TIME OBJECT DETECTION FOR AUTONOMOUS AUVs USING AN ATTENTION–BASED FAST–RCNN FRAMEWORK

MOHCINE BOUDHANE [a,b,*], HAMZA TOULNI [c]

[a]LabSI & National School of Artificial Intelligence and Data Science Taroudant
Ibn Zohr University
BP 32 S, CP 80000, Agadir, Morocco
e-mail: m.boudhane@uiz.ac.ma

[b]Socio-Technical Systems Engineering Institute
Vidzeme University of Applied Sciences
Cēsu Street 4, LV-4201, Valmiera, Latvia

[c]Department of Computer Science
Rabat National School of Mines (ENSMR)
Hadj Ahmed Cherkaoui Avenue, BP 753, Agdal, Rabat, Morocco

This article considers the problem of fish monitoring in an underwater environment, where many problems might occur, including occlusion, pose changes, and complexity of the scene. Recognizing fish behavior is very important to develop various types of technologies able to provide more precise estimations and monitoring of fish populations in a long term. In this paper, we propose a novel method for underwater fish monitoring (shape modeling and pose estimation). Two main aspects of underwater image processing will be studied: classification and localisation. Additionally, we extract key point features from fish patterns. The fish position and motion are not sufficient features to avoid scene problems. Skeleton extraction could offer us a large range of additional information. It models an object as a set of points of a certain manifold. The 3-dimensional fish pose, along the track of its 3D motion, could depend on curve segments of the underlying manifold. Faster reccurent conventional neural networks (faster R-CNNs) will be used to extract the fish skeleton in different poses. Also, a 3-dimensional trajectory of multiple fish will be derived using a Kalman filter based on the previous feature matching process. The simulation is made for live fish in a fish tank. Experimental results show that our method outperforms relevant models in terms of precision, achieving a minimal accuracy of 94.2%.

Keywords: underwater monitoring, deep learning, object detection and identification, AUVs, optical images, underwater robotics, intelligent robots.

1. Introduction

Underwater object identification remains a challenging field due to environmental distortions such as low visibility, turbidity, and sparse labeled data (Jian *et al.*, 2024; Elmezain *et al.*, 2025; Gou *et al.*, 2025a; Gregory *et al.*, 2025; Boudhane *et al.*, 2019; Hasan *et al.*, 2024). Although many recent methods have yielded moderate success, they try to improve underwater exploration (Chen *et al.*, 2023; Boudhane *et al.*, 2018; Durlik *et al.*, 2025; Boudhane and Balcers, 2019; Guo *et al.*, 2020). However, despite all these efforts, its not sufficient to exploit underwater resources.

Gao *et al.* (2017) introduced a dual-branch CNN using semi-synthetic data. While this yields decent results under controlled laboratory conditions, its performance drops significantly in more variable open-water environments. Czapiewska *et al.* (2025) used acoustic-based approach using Doppler shift. Doppler shift determination methods dedicated to MBFSK modulation. Folkman *et al.* (2025) applied unsupervised domain adaptation to reduce the domain shift from terrestrial to underwater imagery. However, tuning

*Corresponding author

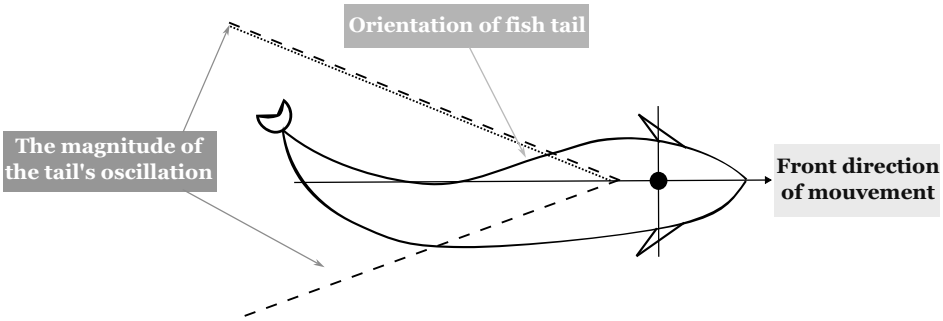

Fig. 1. Morphological criteria of the fish body.

constraints and sensitivity to style inconsistencies limits real-world applicability. Wang *et al.* (2021) adopted an attention-based YOLO for AUV deployment, which yields fast inference but consistently lower recall and precision under low light and turbidity conditions. Their method achieves at most an 85% mAP. Xiao *et al.* (2024) proposed multi-scale feature fusion to mitigate low-light effects. While this improves sensitivity, it introduces latency and still fails to recover degraded features embedded in high turbidity, resulting in lower IoU values than when using our method. Kapoor *et al.* (2023) proposed a domain-aware small-data architecture with efficient memory usage. Unfortunately, it generalizes poorly when encountering rare classes or complex backgrounds, showing worse confusion matrix performance than in the case of our proposed system. Gue *et al.* (2025a; 2025b) fused sonar and RGB via transformer modules to improve detection in turbid water. This multimodal approach slightly improves accuracy, although at the cost of significant calibration overhead and a large model size.

Cai and Zhang (2025) similarly fused many sensors, showing improved recall in turbid water, although at the cost of precise sensor synchronization, unlike in our purely vision-based approach. In addition, Boudhane and Toulni (2024) combined image enhancement preprocessing with a faster R-CNN. That slightly improved detection, but introduced high computational cost and slower inference than in the case of our optimized pipeline. Magdy *et al.* (2025) improved training pipelines for faster R-CNNs in low light using contrast adjustments. They slightly increased recall, but showed high false positive rates, especially in varying illumination gradients. He *et al.* (2022) worked on adaptive anchor refinement for irregular shapes. Their improvements in bounding box alignment are modest and remain inferior to ours in terms of IoU stability and mAP. Other recent techniques such as self-supervised learning (Mello *et al.*, 2022; Wu *et al.*, 2024), GAN-based turbidity simulation (Han *et al.*, 2023; Kaur *et al.*, 2025), program domain adaptation (Chen and Pei, 2022; Deng *et al.*, 2023), and edge-based boundary refinement (Zhou *et al.*, 2022; Priyadharsini and Sree Sharmila, 2019) offer additional gains. Zhang *et al.* (2023) present an approach that uses a backbone framework to assist the detection network in extracting important features at a high level and minimizing the impact of irrelevant information. In underwater environments, their method demonstrated good performance in detecting dense objects, as it reached arround 79.8%. Cai and Zhang (2025) introduce MAW-YOLOv11, a lightweight underwater object detection model designed to enhance detection accuracy in challenging underwater environments. The model employs a multi-scale edge information selection module, combined with downsampling structures, to improve feature extraction while reducing computational overhead. Experimental results reached about 82.4% as detection accuracy. Also, Wang *et al.* (2024) introduced a specialized underwater target detection method designed for freshwater fish recognition, incorporating the model for better feature extraction in order to improve efficiency. Although this approach shows promising results, its reliance on a custom dataset may limit its generalization to other underwater environments, with different fish species and conditions. Despite improvements, the model's complexity can create obstacles for real time applications, particularly in situations where resources are scarce.

2. System overview

We initially have a series of raw images. Since fish appear in several poses in the images (Fig. 1), it is important to use a follow up process (or a combination of channels) on which the structures of targeted fish appear as clearly as possible. The skeleton of a fish includes vertebrates, a backbone (the central ridge) and a skull. The central ridge runs from the head to the caudal ridge and is made up of vertebrae. The vertebrae are not very specialized and significantly similar to each other. Each carries, in the caudal region, a dorsal process and a ventral spine, the whole clearly marking the median plane of the body. These vertebrae have lateral developments which carry the ribs.

The ribs and the ridges are fibrous rods, more or less calcified and sharp, which are embedded in the muscular masses. The fish's body can be divided into two groups as shown in Fig. 1. On the one hand, the first half (oriented to the fish head) is generally a rigid part. On the other, since fish use the caudal tail for directions, the second half of the fish is usually non-rigid. Figure 2 demonstrates this repartition.

Odd-numbered fins, supported by rays, are characteristic organs of fish. The proportion, the position, and the shape of the fins is related to the shape of the body and there is a correlation with the way of swimming. The balance of the fish depends on the compensatory effects of these different organs. The characters of the fin shape enter a large part in the classification of fish. In fact, we can split the fish shape into two different groups:

- rigid part (half oriented on the head part),
- non-rigid part (second half used mostly for directions).

3. Proposed approach

Our model for object identification uses convolutional neural networks that focus on regions. In a single network, this model brings together proposal generation and detection steps. By including a region proposal network, object detection can be significantly improved in terms of speed and accuracy.

3.1. Convolutional feature extraction (first layer (CNN)). With an image as input I, we have

$$\text{Overall map} = F(I;\theta), \tag{1}$$

where F represents the CNN with weights θ.

3.2. Region proposal network (RPN). The RPN is used to generate region proposals (RPs). The RPN slides small regions. The convolutional characteristics map created by the CNN is used to drag a small network. The prediction of region proposals (bounding boxes) and objectivity scores is made at every location in the sliding window.

Anchor boxes. Anchors are predefined enclosing boxes of different scales and proportions, centered at each location in the sliding window:

- $W \times H$: size of the feature-map,
- k: number of anchors per position.

RPN layers.

1. *Intermediate convolutional layer.* A 3×3 convolutional layer is applied to the feature map:

$$f_{i,j} = \sigma(w^T \cdot x_{i,j} + b), \tag{2}$$

where

- $x_{i,j}$ is the feature vector at location (i, j),
- b and w are, respectively, the bias and weights of the 3×3 convolution,
- σ is a non-linear activation function (ReLU).

2. *Bounding box regression and objectness classification*. The previous step generates two steps in this phase:

- *Classification layer*: Outputs $2k$ scores (object vs. background) for each anchor,

$$p_{i,j} = \text{softmax}(w_{\text{cls}}^T \cdot f_{i,j} + b_{\text{cls}}), \tag{3}$$

where $p_{i,j}$ is the probability of objectness for each anchor.

- *Regression layer*: Outputs $4k$ coordinates for each anchor (dx, dy, dw, dh),

$$t_{i,j} = w_{\text{reg}}^T \cdot f_{i,j} + b_{\text{reg}} \tag{4}$$

where $t_{i,j}$ represents the bounding box regression parameters.

Loss function for the RPN. The loss function combines classification and regression losses (defined by Peng *et al.*, 2018):

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i [p_i^* L_{reg}(t_i, t_i^*)], \tag{5}$$

where

- p_i is the predicted probability of anchor i,
- p_i^* is the ground truth label (1 if a positive anchor, 0 if negative),
- t_i are predicted coordinates,
- t_i^* are ground truth coordinates,
- L_{cls} is classification loss (log loss),
- L_{reg} is regression loss (smooth L1 loss),
- λ is a balancing parameter,
- N_{cls} and N_{reg} are normalization factors.

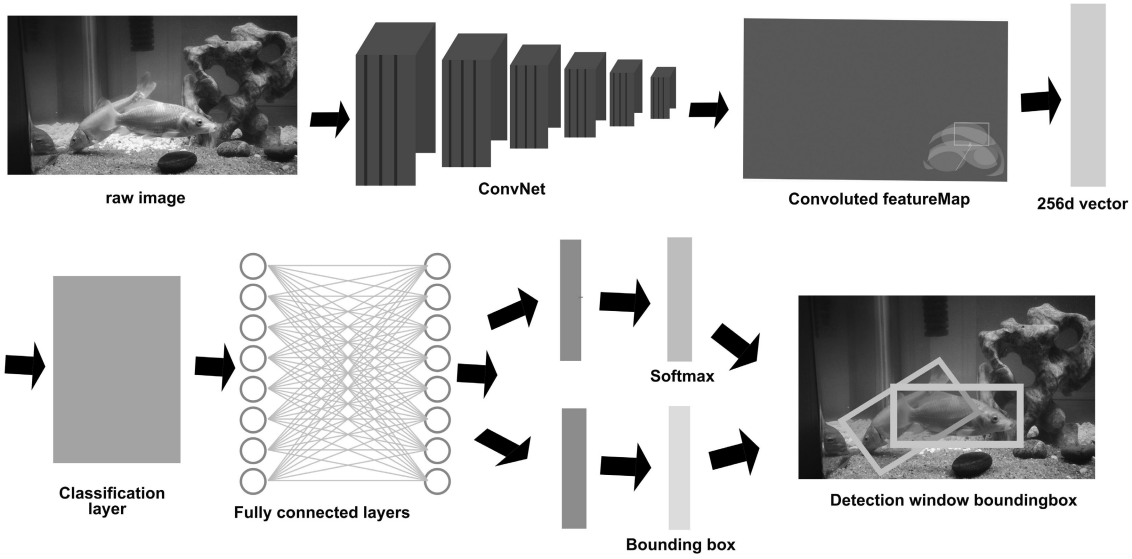

Fig. 2. Architecture of the proposed approach.

3.3. Region of interest pooling. Given a proposal $R = (x, y, w, h)$,

- divide R into a grid of size $H_{\text{pool}} \times W_{\text{pool}}$,
- apply max pooling to each grid cell to produce a fixed-size output (e.g., 7×7).

3.4. Classification and bounding box regression. The pooled feature maps are fed into fully connected layers for final classification and bounding box refinement.

Fully connected layers.

- *Classification layer*: Outputs C scores (one for each class),

$$\text{score} = \text{softmax}(W_{\text{cls}} \cdot \text{RoI\_pool\_output} + b_{\text{cls}}). \quad (6)$$

- *Bounding box regression layer*: Outputs $4C$ coordinates (dx, dy, dw, dh for each class),

$$\text{bbox} = W_{\text{reg}} \cdot \text{RoI\_pool\_output} + b_{\text{reg}}. \quad (7)$$

Loss function for the detection network. Similarly to the RPN, the loss function combines classification and regression losses (defined by Peng *et al.*, 2018):

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i [p_i^* L_{reg}(t_i, t_i^*)]. \quad (8)$$

3.5. Training and inference. During training, the detection network with the RPN are trained jointly. The overall loss is the sum of their losses. On the other hand, during inference, the RPN generates proposals, which are then refined and classified by the detection network to produce the final detections. Figure 3 presents the whole classification process of the proposed approach (Algorithm 1).

3.6. Trajectory estimation using a Kalman filter. The goal is to associate detected object positions across multiple frames to form complete trajectories. The Kalman filter helps by predicting the object's state and correcting these predictions with actual measurements (detections), making it highly suitable for trajectory linking in object tracking. This algorithm proceeds through a series of prediction and update steps to estimate the position and velocity of objects over time. Figure 3 introduces the tracking process. The algorithm begins by initializing the state vector, which includes the initial location and speed in the x and y directions, and the initial covariance matrix representing the initial uncertainty of the state estimates. For each frame, the algorithm determines the object's next state by employing the state transition matrix $\mathbf{F}$. It also updates the covariance matrix to reflect the increased uncertainty over time due to process noise, represented by the matrix $\mathbf{Q}$. These prediction and update steps are repeated for each frame, continuously refining the state estimates and linking detections into trajectories. Algorithm 2 illustrates a pseudo code for the trajectory estimation process that follows nine steps:

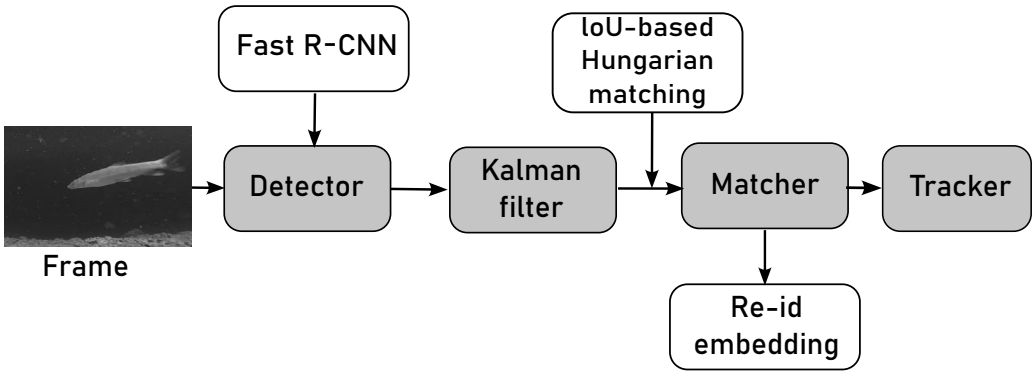

Fig. 3. Tracking process used in this approach.

Algorithm 1. Proposed identification algorithm.

REQUIRE:
    Extracted frame I from image sequences
Loop:
    Extract features using pre-trained CNN on I
    Generate region proposals using RPN
forall region proposals p_i
    Apply RoI pooling to obtain fixed-size feature vector $f(p_i)$
    Pass $f(p_i)$ through fully connected layers
    Get probability estimates for classes and more precise boundary coordinates
End Loop
Return Class labels with associated bounding boxes for objects that have been spotted

1. *State prediction* x_k:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \end{bmatrix}, \tag{9}$$

where k is the state time, x_k and y_k are the locations, and $\dot{x}_k$ and $\dot{y}_k$ reflect the speed in the directions x and y.

2. *State transition* $\mathbf{F}_k$:

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

where Δt is the time step.

3. *Prediction covariance* Pc: The error covariance prediction is

$$\mathbf{Pc}_{k|k-1} = \mathbf{F}_k \mathbf{Pc}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k, \tag{11}$$

where

- $\mathbf{Q}_k$ is the covariance matrix that processes noise,
- $\mathbf{Pc}_{k|k-1}$ is the covariance matrix of the predicted error.

4. *Measurement prediction*: The predicted measurement is

$$\mathbf{z}_{k|k-1} = \mathbf{H}_k \mathbf{x}_{k|k-1}, \tag{12}$$

where $\mathbf{H}_k$ is the observation matrix. For position measurements only, we have

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{13}$$

5. *Measurement update (correction)*: The measurement residual (innovation) is

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{z}_{k|k-1}, \tag{14}$$

where $\mathbf{z}_k$ is the actual measurement vector.

6. *Innovation covariance*: The innovation covariance is

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k, \tag{15}$$

where $\mathbf{R}_k$ is the measurement noise covariance matrix.

7. *Kalman gain*: The Kalman gain is computed as

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}. \tag{16}$$

8. *State update*: The state estimate is updated with the measurement

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k. \tag{17}$$

9. *Error covariance update*: The error covariance is updated as

$$\mathbf{Pc}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{Pc}_{k|k-1}. \tag{18}$$

Algorithm 2. Tracking and trajectories linking.

Input:
  fish orientation: list of detections for each frame
  initial_state: initial state vector $[x, \dot{x}, y, \dot{y}]$
  initial_covariance: initial state covariance matrix
  $\mathbf{F}$: state-transition matrix
  $\mathbf{Q}$: covariance matrix (noise processing)
  $\mathbf{H}$: observation matrix
  $\mathbf{R}$: covariance matrix for noise estimation
  Δt: time step
Output:
  trajectories: linked trajectories of detected objects
Procedure:
Initialize state and covariance
  state ← initial_state
  covariance ← initial_covariance
Initialize empty list for trajectories
  trajectories ← []
for each frame in detections **do**
  Predict:
    state ← $\mathbf{F} \times$ state
    covariance ← $\mathbf{F} \times$ covariance $\times \mathbf{F}^T + \mathbf{Q}$
  Get detections for the current frame
    current_detections ← detections[frame]
  if current_detections is not empty **then**
    for each detection in current_detections **do**
      Calculate measurement residual (innovation)
        residual ← detection $- \mathbf{H} \times$ state
      Calculate the innovation covariance
        innovation_covariance ← $\mathbf{H} \times$ covariance $\times \mathbf{H}^T + \mathbf{R}$
      Calculate Kalman gain
        kalman_gain ← covariance $\times \mathbf{H}^T \times$ inverse(innovation_covariance)
      Update state with measurement
        state ← state + kalman_gain $\times$ residual
      Update error covariance
        covariance ← ($\mathbf{I}-$ kalman_gain $\times \mathbf{H}$) $\times$ covariance
      Append updated state to trajectories list
        trajectories.append(state)
    end for
  else
    Append predicted state to trajectories list
      trajectories.append(state)
  end if
end for
Return trajectories

10. *Data association via IoU and Hungarian matching*: To associate new detections B_t with predicted tracks $\hat{B}_{t|t-1}$, a cost matrix $C \in \mathbb{R}^{M \times N}$ is built based on the intersection-over-union (IoU) metric:

$$\text{IoU}(b_i, b_j) = \frac{\text{area}(b_i \cap b_j)}{\text{area}(b_i \cup b_j)}. \tag{19}$$

The cost for each pair is

$$C(i, j) = 1 - \text{IoU}(b_i, b_j). \tag{20}$$

The optimal matching π is found using the Hungarian algorithm:

$$\pi^* = \arg\min_{\pi} \sum_i C(i, \pi(i)). \tag{21}$$

11. *Re-identification embedding for appearance modeling*: To enhance robustness under occlusion, we integrate an appearance embedding vector $\phi(b_t^i)$ extracted from a CNN-based Re-ID module:

$$\phi : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^d. \tag{22}$$

Similarity between two embeddings is computed via cosine similarity:

$$\text{sim}(\phi_i, \phi_j) = \frac{\phi_i \cdot \phi_j}{\|\phi_i\| \|\phi_j\|}. \tag{23}$$

If the IoU-based match fails, appearance similarity is used to recover the lost identity across frames.

4. Results and the experiment

4.1. Dataset. To ensure the objective of this work, we placed different fish species in a 70 cm × 120 cm × 39 cm container. The matrix includes seven marine classes: *Goldfish*, *Jellyfish*, *Stingray*, *Parrotfish*, *Shark*, *Jacks*, *Surgeonfishes*, but certain other species were also used in the test. A depth of 60 cm was filled with almost 275 liters of water. Two optical sensors were placed, respectively, into two positions on top and side. Two Barlus Underwater IP Cameras were used for recording with a resolution of 2592×1944. Three months of recording day and night were used to monitor 15 fish. Figure 4 shows the system montage of this experiment. In practice, the obtained data are decomposed into two groups: side view and top view. The extracted data are captured simultaneously. Each couple of frames is labeled by the name, position and timestamp. Figure 4 shows a sample of the data used in this experiment.

To effectively deploy the proposed algorithm for underwater object detection in a controlled environment, the system must integrate robust hardware components, optimized software configurations, and a well-prepared

aquatic setup. The core of the system is a waterpro underwater camera (IP68 rated), capable of capturi high-resolution video (1080p) and suitable for low-lig conditions. Cameras with wide-angle, low-distortic lenses are preferred for wider scene coverage. The cameras will be mounted at fixed locations in t environment or integrated into a mobile robotic platfor such as an autonomous underwater vehicle (AUV depending on the intended monitoring application.

For real-time inference, the computing platform essential. A viable option is to deploy GPU-base embedded systems such as NVIDIA Jetson in waterpro enclosures. These systems support deep learnir inference with relatively low power consumption. In scenarios requiring higher computing power, image data can be transmitted via Ethernet cables to a surface workstation equipped with a powerful GPU (RTX 4080). Cooling and sealing mechanisms are integrated to ensure operational safety and longevity of the hardware components. On the other sides lighting is another key consideration. Underwater environments suffer from reduced visibility and uneven lighting, so high-intensity, waterproof LED arrays are installed around the monitored area. These lights must provide uniform illumination and be adjustable in intensity. The use of pulse-width modulation (PWM) and software control allows lighting conditions to be tailored to the needs of the detection task. Particular attention must be paid to minimizing reflections and backscatter by properly aiming the lights. The system is powered by DC sources connected to the surface or by high-capacity internal batteries such as LiPo ones. Communication between the underwater system and the surface can be ensured via wired Ethernet for high bandwidth and acoustic modems for autonomous operation. The integration of the perception system into the AUV control architecture was achieved using a ROS-based communication framework. Detection and tracking outputs were published as geometry messages at 10 Hz, enabling real-time interaction with navigation and mission planning nodes. This modular design allows future upgrades to lighter detection networks or fusion with sonar data without altering the control stack.

On the software side, the proposed model is implemented with the TensorFlow library and is fine-tuned on underwater datasets to adapt to the visual distortions specific to aquatic environments. Classes of interest may include fish species, coral types, artificial markers, or other underwater objects. Preprocessing steps such as resizing, normalization, and possibly color correction must be applied to the raw images before feeding them into the model. Once inference is complete, bounding boxes and class labels are rendered on the images, stored locally, and transmitted to a surface station for visualization.

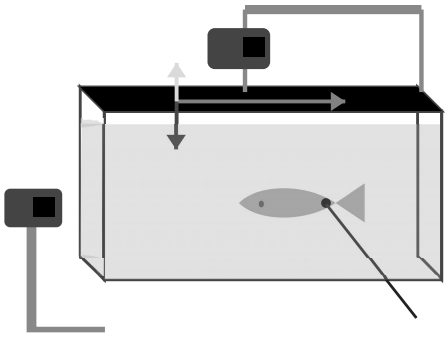

Fig. 4. Two optical sensors (top and side cameras) and a fish tank are used for the setup of the system.

4.2. Comparison of classification metrics. What would be optimal is to have a minimum number of markers to obtain, and this to simplify the size of the data to be processed and to be wary of problems of exchanging markers during capture. As mentioned above, four markers were chosen to define the model of the human trajectory. The vertical Y axis has a positive up-and-down orientation, the X axis has a forward-and-down direction, and the Z axis has a right-and-right orientation. The position of the cameras is optimized so that two cameras always place the markers in the detected objects.

In the experiment, we use a range of performance metrics to comprehensively evaluate the effectiveness of our model. Precision quantifies the proportion of true positive predictions out of all positive predictions made by the model, providing insight into the model's accuracy when it predicts a positive class. Recall, on the other hand, measures the ability of the model to correctly identify all relevant instances, reflecting its sensitivity to positive cases. To balance both precision and recall, we use the F1 score, which is the harmonic mean of these two metrics and offers a single value that highlights the trade-off between them.

For tasks like segmentation, we employ intersection over union (IoU), which evaluates the overlap between the predicted and ground truth regions, providing a clear measure of how well the model identifies relevant areas. To assess its overall performance in classification tasks, we utilize the area under the curve (AUC), which reflects the model's ability to discriminate between positive and negative classes across varying thresholds. Finally, we track the number of parameters to evaluate the model's complexity and efficiency, helping to understand its computational requirements and potential for deployment in resource-constrained environments. In summary, to evaluate our results, we use the following metrics:

Fig. 5. Real system setting: some fish used in the experiment (left), real system montage (right).

Accuracy:

$$Precision = \frac{TP}{TP + FP}, \tag{24}$$

Recall:

$$Recall = \frac{TP}{TP + FN}, \tag{25}$$

F1 score:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{26}$$

Mean absolute error:

$$\sum_{i=1}^{D} |x_i - y_i|, \tag{27}$$

Root mean square error:

$$\sum_{i=1}^{D} (x_i - y_i)^2. \tag{28}$$

4.3. Numerical results (classification and tracking). The proposed method was evaluated in a controlled aquatic environment containing various target objects such as several fish classes. Table 1 illustrates the diferent classes used in the experiment. The system was tested under varying conditions, light intensity, and object motion to assess its robustness and reliability. The main performance indicators used in this evaluation included mean accuracy (mAP), precision, F1 score, recall, intersection over union (IoU), and detection latency (inference time per frame).

The confusion matrix presented above evaluates the performance of the proposed method in a binary classification task distinguishing target objects (fish) from the "background" classes. This matrix synthesizes the model's predictions with respect to the ground truth and provides a clear view of the correct classifications and errors made during the evaluation. In this matrix, the model correctly predicted 5,945 instances as objects when they were in fact that. This high number of true positives reflects the system's strong ability to detect and classify underwater targets, which is essential for applications such as marine life monitoring, pipeline inspection, and debris detection.

Table 1. Dataset split by class.

Super class	Training	Validation	Testing
Goldfish	1061	359	249
Jellyfish	585	155	154
Parrotfish	530	104	82
Jacks	575	134	85
Shark	559	107	38
Surgeonfishes	478	77	61
Stingray	436	83	55
Others	200	83	55

In addition, the model correctly classified 96 background regions as non-object areas, confirming its ability to filter out irrelevant or empty parts of the scene. However, the model made some errors. There were 104 false positives, meaning that background elements were incorrectly identified as objects. This type of error can stem from environmental factors such as marine snow, light reflection, or underwater noise patterns that visually resemble real objects. On the other hand, there were 55 false negatives, where real objects were not detected by the model. These missed detections could be due to partial occlusion, blurring due to water turbidity, or unusual poses deviating from the training examples. From these values, we can derive several performance indicators that more precisely quantify the models effectiveness. The overall accuracy reaches approximately 97.4%, indicating that almost all predictions were correct. The precision, which focuses on the number of predicted objects that were real objects, is around 98.28%, showing that the model generates very few false alerts. The recall is even higher, at 98.68%, meaning the model captures almost all real objects. The overall precision is approximately 98.68%, which highlights the model's reliability and consistency.

Table 2. Detailed comparison of object detection performance for different models in underwater controlled environments.

Metric	Proposed method	Zhang *et al.*, 2023 (YOLOv5)	Cai *et al.*, 2025 (SSD)	Wang *et.al*, 2024 (RetinaNet)
Mean average precision (mAP)	**91.4%**	81.2%	76.4%	83.1%
Precision	**94.2%**	86.7%	80.5%	88.1%
Recall	**89.6%**	82.1%	74.3%	85.5%
F1-score	**91.9%**	84.4%	77.2%	86.7%
Intersection over union (IoU)	**0.78**	0.69	0.62	0.73
AUC (ROC)	**0.96**	0.90	0.85	0.88
Number of parameters (millions)	134 M	7.5 M	24 M	56 M

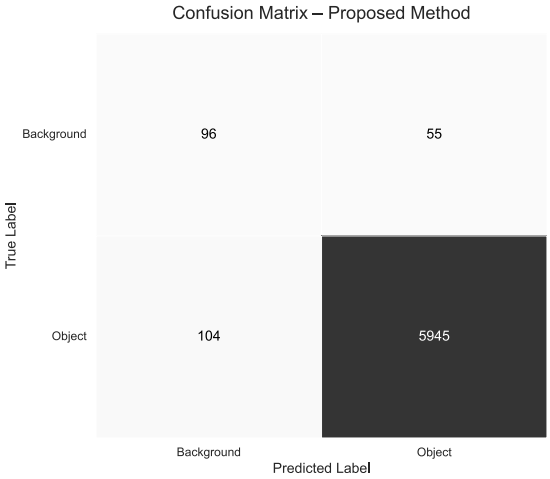

Fig. 6. Confusion matrix of the proposed approach.

4.3. Visual results. For comparison, several benchmark object detection algorithms were implemented and evaluated under identical experimental conditions. YOLOv5s, presented by Zhang *et al.* (2023), known for its speed, achieved a higher inference rate (17.5 FPS), but its detection accuracy was lower, with an mAP of 81.2% and an IoU of 0.69. Cai and Zhang (2025) achieved an mAP of 76.4% with faster processing (12.3 FPS) for the single shot multiBox detector (SSD), but it was particularly impacted in low light or high turbidity. RetinaNet by Wang *et al.* (2024), which incorporates a focal loss mechanism to correct class imbalance, achieved an mAP of 83.1% with an average IoU of 0.73, which is superior to the SSD and YOLO in more complex scenes but still inferior to the proposed method.

Due to its two-stage architecture, our approach was able to achieve superior performance by producing high-quality region proposals through the RPN, followed by fine-grained classification and bounding box refinement. In underwater environments with blurred object contours and noisy backgrounds, this leads to more accurate localization and less chance of false positives. In addition, the proposed model on domain-specific underwater datasets significantly improved classification accuracy. Data augmentation strategies such as simulated blurring, color jittering, and noise injection also contributed to its generalization in variable underwater conditions. In terms of robustness, the proposed model showed a minimal drop in detection accuracy (only about 5%) under low-visibility conditions, while YOLOv5 recorded a drop of about 13% and the SSD of nearly 20%. Furthermore, qualitative results show that the proposed approach provided more stable bounding boxes with better temporal consistency between video sequences. In terms of precision, localization quality, and robustness under underwater constraints, our method is better than other realtime detection models. Although it operates at a lower frame rate than single-stage detectors, it offers a highly reliable solution for applications where detection accuracy is critical, such as marine research, inspection and underwater robotics. Figure 8 shows a comparative study of the proposed approach vs. the other cited methods. The presented ROC curve provides a comparative evaluation of the proposed faster R-CNN-based underwater object detection method against other state-of-the-art approaches, including YOLOv5, the SSD, and RetinaNet. Our method achieves the highest performance with an AUC of approximately 99.28%, demonstrating an exceptional ability to distinguish underwater objects with minimal false positives. The curve remains consistently superior to those of other models, especially in the medium and low false positive rate ranges, which is critical in underwater scenarios, where noise and clutter are common.

4.4. Limitations and future directions. Despite the encouraging performance of the proposed fast R-CNN framework for underwater object tracking, several limitations must be acknowledged. The most significant challenge arises from the model's sensitivity to severe turbidity, which frequently occurs in coastal and estuarine environments. Under such conditions, light scattering and absorption significantly degrade image contrast, color fidelity, and texture cues that are essential for convolutional feature extraction. As a result, the detector

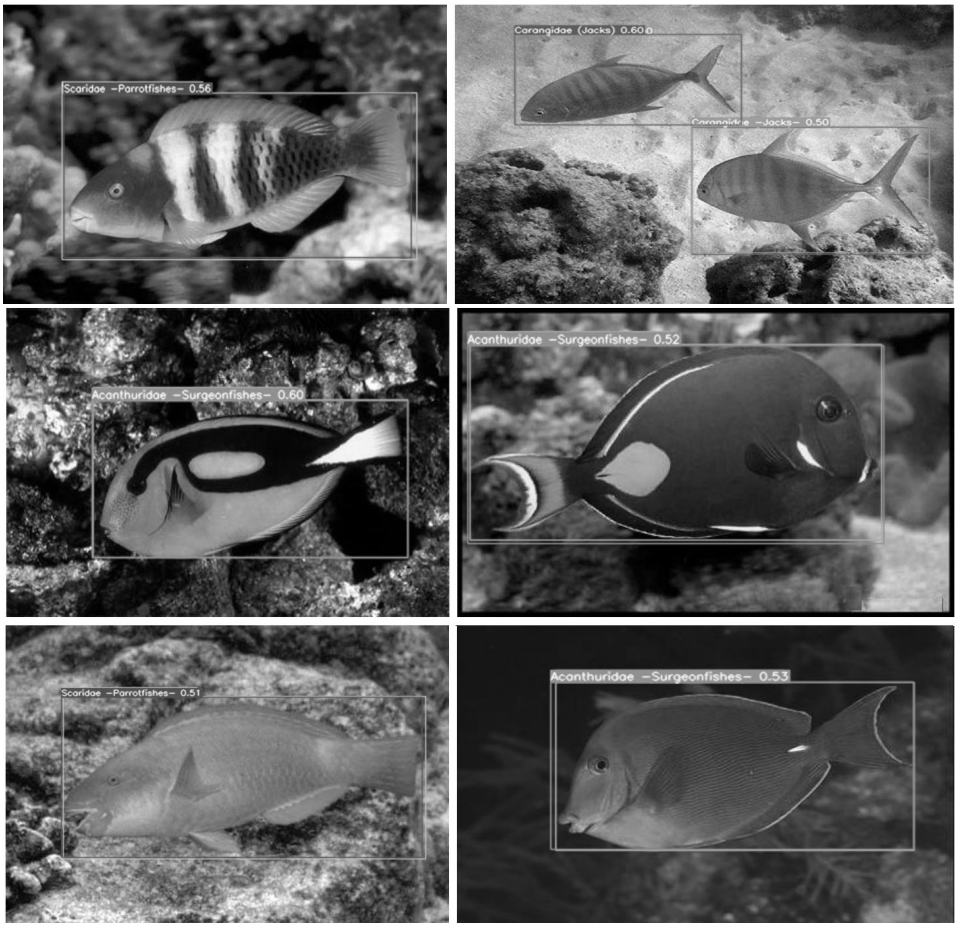

Fig. 7. Experimental results of the proposed approach.

occasionally fails to maintain consistent bounding box localization and confidence scores, leading to an increase in false negatives and tracking interruptions. Although the inclusion of preprocessing steps such as adaptive histogram equalization and white balancing improved robustness by approximately 8% in MOTA (multi-object tracking accuracy), extreme turbidity remains a bottleneck for purely vision-based approaches. Another limitation concerns the generalization capability of the model across different underwater domains. The dataset used is geographically limited. Consequently, the trained model may exhibit domain bias when exposed to previously unseen optical characteristics or fauna. Future work will focus on expanding the dataset to include deeper-water and high-salinity environments, as well as augmenting it with synthetic data generated through physics-based underwater rendering engines. From an algorithmic perspective, the reliance on the fast R-CNN entails a relatively high computational footprint compared to more recent transformer or anchor-free architectures.

This constraint limits scalability for ultra-low power AUV platforms and long endurance missions. Future research will therefore investigate the integration of lightweight backbones (e.g., MobileNet-V3 or YOLOv8-nano) and the application of knowledge distillation to preserve accuracy while reducing energy consumption. Additionally, multi-sensor fusion combining optical and forward-looking sonar modalities will be explored to mitigate visual degradation and maintain tracking continuity under extreme loss of visibility.

5. Conclusion

A robust framework for detecting underwater fish was proposed in this work based on the faster R-CNN architecture and suitable for deployment in controlled aquatic environments. The model was trained and evaluated on 15 object classes, achieving high performance on key metrics, with the overall accuracy reaching 99.28%. Visualizations such as confusion matrices, ROC curves, and per-class histograms demonstrated the system's ability to accurately identify underwater fish features while minimizing false detections. Comparative analysis with state-of-the-art

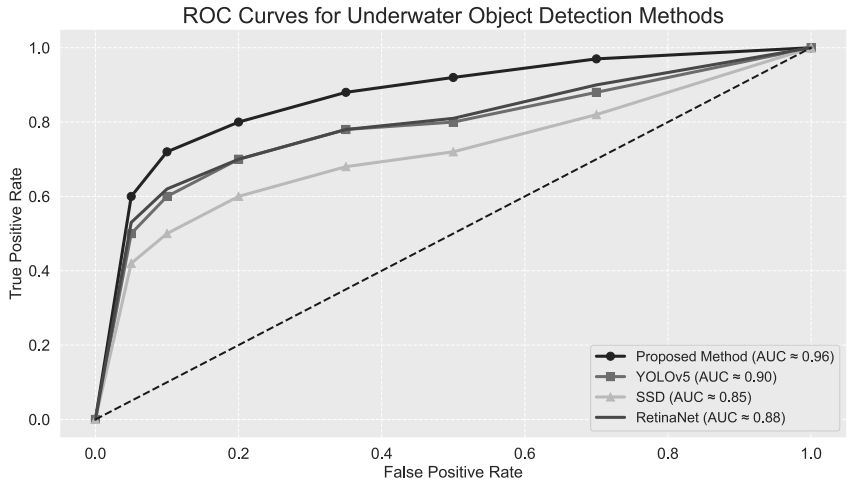

Fig. 8. ROC curve.

methods confirmed the superiority of the proposed approach, particularly in terms of IoU, F1 score, and accuracy. Applications like AUV exploration, marine life monitoring, and infrastructure inspection can depend on the system's reliability. Its success in controlled environments indicates strong potential for deployment in real-time embedded systems, with future work focused on generalization to open-water environments, sensor fusion with sonar data, and integration with low-power computing platforms.

Acknowledgment

This research has been supported by a grant from the European Regional Development Fund project *The Study of Computer Vision Algorithms for Underwater Fish Inspection* (no. 1.1.1.2/VIAA/2/18/348) within Activity 1.1.1.2: "Post-doctoral ResearchAid".

References

Boudhane, M. and Toulni, H. (2024). An adaptive fast-RCNN method for fish monitoring: From an artificial environment to the ocean, *in* M.B. Ahmed *et al.* (Eds.), *Information Systems and Technological Advances for Sustainable Development*, Springer, Berlin/Heidelberg, pp. 301–309, DOI: 10.1007/978-3-031-75329-9_33.

Boudhane, M., Balcers, O. and Nsiri, B. (2019). Underwater exploration issues: deep study on optical underwater vision for an effective traditional fishing, *ICWIP 2019: Proceedings of the 2019 2nd International Conference on Watermarking and Image Processing, Marseille, France*, pp. 32–35, DOI: 10.1145/3369973.3369981.

Boudhane, M. and Balcers, O. (2019). Underwater image enhancement method using color channel regularization and histogram distribution for underwater vehicles AUVs and ROVs, *International Journal of Circuits* **13**(1): 571–578.

Boudhane, M., Nsiri, B. and Belhoussine, T.D. (2018). Underwater optical fish classification system by means of robust feature decomposition and analysis using multiple neural networks, *International Journal of Advanced Computer Science and Applications* **9**(12): pp 621–630, DOI: 10.14569/IJACSA.2018.091286.

Cai, S., Zhou, X., Cai, W., Wei, L. and Mo, Y. (2025). Lightweight underwater object detection method based on multi-scale edge information selection, *Scientific Reports* **15**(1): 27681, DOI: 10.1038/s41598-025-13566-3.

Cai, W. and Zhang, M. (2025). Multi-modality object detection with sonar and underwater camera via object-shadow feature generation and saliency information, *Expert Systems with Applications* **287**(C): 128021, DOI: 10.1016/j.eswa.2025.128021.

Chen, G., Mao, Z., Wang, K., and Shen, J. (2023). HTDet: A hybrid transformer-based approach for underwater small object detection, *Remote Sensing* **15**(4): 1076, DOI: 10.3390/rs15041076.

Chen, Y.-W. and Pei, S.-C. (2022). Domain adaptation for underwater image enhancement via content and style separation, *IEEE Access* **10**(1): 1-1, DOI: 10.1109/ACCESS.2022.3201555.

Czapiewska, A., Łuksza, A., Schmidt, J.H., Studański, R., Wojewódka, Ł. and Żak, A. (2025). Doppler shift determination methods dedicated to MBFSK modulation, *International Journal of Applied Mathematics and Computer Science* **35**(3): 467–477, DOI: 10.61822/amcs-2025-0033.

Deng X., Liu T., He S., Xiao X., Li, P. and Gu, Y. (2023). An underwater image enhancement model for domain adaptation, *Frontiers in Marine Science* **10**(1): 1138013, DOI: 10.3389/fmars.2023.1138013.

Durlik, I., Miller, T., Kostecka, E., Kozlovska, P., and Slaczka, W. (2025). Enhancing safety in autonomous maritime transportation systems with real-time AI agents, *Applied Sciences* **15**(9): 4986, DOI: 10.3390/app15094986.

Elmezain, M., Saad Saoud, L., Sultan, A., Heshmat, M., Seneviratne, L. and Hussain, I. (2025). Advancing underwater vision: A survey of deep learning models for underwater object recognition and tracking, *IEEE Access* **13**(1): 17830–17867, DOI: 10.1109/ACCESS.2025.3534098

Folkman, L., Pitt, K.A. and Stantic, B. (2025). A data-centric framework for combating domain shift in underwater object detection with image enhancement, *Applied Intelligence* **55**(4): 272, DOI: 10.1007/s10489-024-06224-0.

Gao, F., Huang, T., Wang, J., Sun, J., Hussain, A., and Yang, E. (2017). Dual-branch deep convolution neural network for polarimetric SAR image classification, *Applied Sciences* **7**(5): 447, DOI: 10.3390/app7050447.

Guo, L., Liu, X., Ye, D., He, X., Xia, J. and Song, W. (2025a). Underwater object detection algorithm integrating image enhancement and deformable convolution, *Ecological Informatics* **89**: 103185, DOI: 10.1016/j.ecoinf.2025.103185.

Guo, F., Ren, P. and Luo, C. (2025b). UTNet: Event-RGB multimodal fusion model for underwater transparent organism detection, *Intelligent Marine Technology and Systems* **3**(18): 1953–2948, DOI: 10.1007/s44295-025-00065-4.

Guo, P., Zeng, D., Tian, Y., Liu, S., Liu, H. and Li, D. (2020). Multi-scale enhancement fusion for underwater sea cucumber images based on human visual system modelling, *Computers and Electronics in Agriculture* **175**(1): 105608, DOI: 10.1016/j.compag.2020.105608.

Gregory, J., Miehls, S.M., Eickholt, J.L., and Zielinski, D.P. (2025). A real-time fish detection system for partially dewatered fish to support selective fish passage, *Sensors* **25**(4): 1022, DOI: 10.3390/s25041022.

Hasan, K., Ahmad S., Liaf A.F., Karimi M., Ahmed T., Shawon M.A. and Mekhil, S. (2024). Oceanic challenges to technological solutions: A review of autonomous underwater vehicle path technologies in biomimicry, control, navigation, and sensing, *IEEE Access* **12**(1): 46202–46231, DOI: 10.1109/ACCESS.2024.3380458

Han, J., Zhou, J., Wang, L., Wang, Y., and Ding, Z. (2023). FE-GAN: Fast and efficient underwater image enhancement model based on conditional GAN, *Electronics* **12**(5): 1227, DOI: 10.3390/electronics12051227.

He, B., Zhang, Q., Tong, M., and He, C. (2022). An anchor-free method based on adaptive feature encoding and Gaussian-guided sampling optimization for ship detection in SAR imagery, *Remote Sensing* **14**(7): 1738, DOI: 10.3390/rs14071738.

Jian, M., Yang, N., Tao, C., Zhi, H. and Sluo, H. (2024). Underwater object detection and datasets: A survey, *Intelligent Marine Technology and Systems* **2**(9): 1–12, DOI: 10.1007/s44295-024-00023-6.

Kapoor, M., Baghel, R., Badri, N.,Š, Jakhetiya, V., Bansal, A., Jammu, J. and Kashmir, I. (2023). Domain adversarial learning towards underwater image enhancement, *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Paris, France*, pp. 2233–2243, DOI: 10.1109/ICCVW60793.2023.00238.

Kaur, A., Rani, S. and Shabaz, M. (2025). Underwater image dehazing using a hybrid GAN with bottleneck attention and improved Retinex-based optimization, *Scientific Reports* **15**(1): 26132, DOI: 10.1038/s41598-025-11815-z.

Magdy, A., Moustafa, M.S., Ebied, H.M. and Tolba, M.F. (2025). Lightweight faster R-CNN for object detection in optical remote sensing images, *Scientific Reports* **15**(1): 16163, DOI: 10.1038/s41598-025-99242-y.

Mello, C.D., Moreira, B.M., Dias de Oliveira Evald, P.J., Lilles Drews, P.J. and da Costa Botelho, S.S. (2022). Underwater enhancement based on a self-learning strategy and attention mechanism for high-intensity regions, *Computers and Graphics* **107**(1): 264–276, DOI: 10.1016/j.cag.2022.08.003.

Peng, X., Yuelei, X., Hong, T., Shiping, M., Shuai, L. and Chao, L. (2018). Fast airplane detection based on multi-layer feature fusion of fully convolutional networks, *Acta Optica Sinica* **38**(3): 315003, DOI: 10.3788/AOS201838.0315003

Priyadharsini, R. and Sree Sharmila, T. (2019). Object detection in underwater acoustic images using edge based segmentation method, *Procedia Computer Science* **165**(1): 759-765, DOI: 10.1016/j.procs.2020.01.015.

Wang, Z., Ruan, Z., and Chen, C. (2024). DyFish-DETR: Underwater fish image recognition based on detection transformer, *Journal of Marine Science and Engineering* **12**(6): 864, DOI: 10.3390/jmse12060864.

Wang, R., Wang, Z., Xu, Z., Wang, C., Liu, Q. and Zhang, Y. (2021). A real-time object detector for autonomous vehicles based on YOLOv4, *Computational Intelligence and Neuroscience* **2021**(12), DOI: 10.1155/2021/9218137.

Wu, Z., Chen, X., Lu Y. and Yu, J. (2024). Self-supervised underwater image generation for underwater domain pre-training, *IEEE Transactions on Instrumentation and Measurement* **73**, Article no. 5012714, DOI: 10.1109/TIM.2024.3373105.

Zhang, J., Zhang, J., Zhou, K., Zhang, Y., Chen, H., and Yan, X. (2023). An improved YOLOv5-based underwater object-detection framework, *Sensors* **23**(7): 3693, DOI: 10.3390/s23073693.

Zhou, J., Wei, X., Shi, J., Chu, W. and Lin, Y. (2022). Underwater image enhancement via two-level wavelet decomposition maximum brightness color restoration and edge refinement histogram stretching, *Optics Express* **30**(10): 17290–17306.

Xiao, Z., Li, Z., Li, H., Li, M., Liu, X., and Kong, Y. (2024). Multi-scale feature fusion enhancement for underwater object detection, *Sensors* **24**(22): 7201, DOI: 10.3390/s24227201.

Mohcine Boudhane received his bachelor's degree in 2010, his MS degree from Hassan II University, Casablanca, Morocco, in 2012, and his PhD degree in computer application technology from University Hassan II, in collaboration with the University of Applied Sciences, Kiel, Germany, in 2017. He then received a postdoctoral grant at the Vidzeme University of Applied Sciences, Valmiera, Latvia (2019–2022). He is curently an assistant professor at the National School of Artificial Intelligence and Data Science in the Ibn Zohr University of Agadir, Morocco, and a researcher in the Vidzeme Uniersity of Applied Sciences. He is also working in the field of applied artificial intelligence using multi sensors. His research interests include computer vision, image processing, pattern recognition, machine learning, multisensor fusion, marine science and robotics.

Hamza Toulni received his BSc in mathematics and computer science in 2010 from the Faculty of Sciences Ain Chock, Hassan II University in Casablanca, Morocco. In 2012, he obtained his master's degree. In 2018, he received a PhD from Faculté des Sciences Ain Chock, Université Hassan II de Casablanca, Morocco. He is currently an assistant professor and researcher at the Rabat National School of Mines (ENSMR), Morocco, and a member of the LISTD Laboratory at the same school. His research interests include, but are not limited to, VANET communication and its applications, smart cities, IoT and artificial intelligence.

Received: 19 August 2025
Revised: 12 November 2025
Accepted: 21 November 2025