JIKO (Jurnal Informatika dan Komputer) Accredited KEMDIKTISAINTEK, No.0173/C3/DT.05.00/2025
Vol. 9, No. 1, April 2026, pp. 59-67 p-ISSN: 2614-8897
DOI: 10.33387/jiko.v9i1.11256 e-ISSN: 2656-1948

REAL-TIME DOLPHIN DETECTION IN AQUATIC ENVIRONMENTS USING YOLO11-NANO

Febriyanti Ludja[1], Florensce Sumarauw[2], Robby Moody Lintong[3], Steven R. Sentinuwo[4], Alwin M. Sambul[5], Muhamad Dwisnanto Putro[6]

[1,2,3,4] Master Program of Informatics, Faculty of Engineering, Sam Ratulangi University, Manado, 95115, Indonesia
Email: [*1]febriyantiludja026@student.unsrat.ac.id, [2]florenscesumarauw026@student.unsrat.ac.id, [3]robbymoodylintong026@student.unsrat.ac.id, [4]steven@unsrat.ac.id, [5]asambul@unsrat.ac.id, [6]dwisnantoputro@unsrat.ac.id

(Received: 3 January 2026, Revised: 13 January 2026, Accepted: 30 January 2026)

Abstract

Dolphin monitoring plays a crucial role in maintaining the balance of marine ecosystems and supporting the ecotourism sector. However, in practice, automated dolphin monitoring still faces significant challenges, particularly when deployed in real-time applications within dynamic underwater environments. Previous research on computer vision-based dolphin detection generally uses models with high computational complexity. This condition has resulted in increased resource requirements and long inference times, making it difficult to apply to underwater device-based monitoring systems with limited computing power. Therefore, it is necessary to develop more efficient detection models and algorithms so that the system can operate reliably under real-world monitoring scenarios in resource-limited environments. Moreover, the adoption of the latest-generation lightweight detection architectures in aquatic scenarios remains limited. To address these challenges, this study proposes the application of YOLOv11-Nano as a lightweight detection architecture designed for low-latency dolphin monitoring on resource-constrained devices. The proposed model is optimized to strike a balance between inference speed and detection accuracy, enabling competitive performance under challenging underwater conditions. Experimental results show that YOLOv11-Nano achieves a computational complexity of 6.4 GFLOPs with 2.59 million parameters, while attaining 65.0% mAP@50, 43.1% mAP@50:95, and an inference speed of 18.34 FPS. These results show that YOLOv11-Nano is capable of delivering stable and efficient performance with relatively low computational requirements and high inference speed, demonstrating strong potential for application in real-time monitoring systems based on devices with limited resources to support automatic dolphin detection as part of marine ecosystem conservation efforts.

Keywords: *Deep Learning, Dolphin Detection, Real-time, YOLO11-Nano*

**Corresponding Author: Muhamad Dwisnanto Putro*

1. INTRODUCTION

Underwater object detection is a challenging problem in computer vision due to factors such as water turbidity, light variation, surface reflections, and particles that reduce image quality [1], [2]. These conditions make object identification much more difficult than detection in terrestrial environments. Therefore, detection methods that can adapt to dynamic water conditions are essential for achieving accurate and efficient results. Recently, underwater optical image-based object detection methods have become popular as an efficient approach [3]. However, this task remains very challenging due to the complex underwater environment and lighting conditions [4].

Among underwater species, dolphins play an important ecological role and serve as indicators of marine ecosystem health. Accurate detection of these mammals supports population monitoring, migration tracking, and conservation management, contributing to a deeper understanding of marine biodiversity [5]. However, findings from long-term monitoring studies indicate a sustained decline in dolphin populations in recent years [6], [7]. On the other hand, the behavioural characteristics of these mammals, such as their rapid movement, frequent surfacing and diving,

and varied body orientation in different lighting conditions, as well as the degradation of underwater image quality and complex backgrounds, make detection challenging [8]. Traditional observation methods, such as manual visual surveys and acoustic monitoring, are time-consuming and often unreliable. These limitations have driven the development of automated vision-based systems. The integration of deep learning into these systems enables real-time dolphin detection, even in complex aquatic environments. In particular, lightweight architectures such as YOLO11-Nano offer real-time detection with high accuracy and efficiency, making them suitable for integration on mobile platforms with limited resources. Studies [9] show that YOLOv11-Nano has a high degree of suitability for drone-based applications, particularly in supporting real-time processing and inference efficiency. These characteristics make YOLO11-Nano a promising approach for automated marine monitoring and support the acceleration of marine ecosystem conservation efforts [10].

Traditional observation methods, such as manual visual surveys and acoustic monitoring, are time-consuming and often unreliable. These challenges have motivated the advancement of automated systems based on computer vision. The integration of deep learning into these systems enables real-time detection of dolphins, even in complex aquatic environments. In particular, lightweight architectures such as YOLO11-Nano offer real-time detection with high accuracy and efficiency, even in complex underwater conditions, thereby supporting automated marine monitoring and accelerating conservation efforts [11].

Deep learning is rapidly advancing the field of underwater object detection by improving detection performance. In computer vision, object detection refers to the task of identifying and locating objects in images or videos [8]. In addition to classification, this method provides spatial location using bounding boxes. However, underwater object detection remains difficult because of factors including illumination variations, water turbidity, and cluttered backgrounds [12]. Deep learning-based methods help overcome these challenges by automatically extracting discriminative features from image data, improving detection accuracy in aquatic environments [13].

Convolutional Neural Networks (CNNs) are now a cornerstone of modern object detection, primarily due to *their* ability to extract representative features from images [3]. Several studies *have demonstrated* that CNNs are *highly effective for* anomaly detection [14]. This concept applies directly to dolphin detection: against the complex sea background, the presence of a dolphin *is, in essence, an anomaly*. Therefore, the CNN approach can improve the accuracy and reliability of dolphin detection systems in aquatic environments. One algorithm from CNN development is You Only Look Once (YOLO), which integrates classification and localization in one step, enabling real-time detection and efficiency [15]. This system continues to evolve by offering improvements in accuracy, processing speed, and the ability to detect small objects, making it highly relevant for detecting objects such as dolphins.

This architecture offers lightweight variants such as YOLO11-Nano, balancing accuracy and speed with low computational complexity for resource-constrained devices [16], [9]. Nevertheless, underwater object detection systems continue to face significant challenges within the computer vision domain as a result of low visibility, poor lighting conditions, and water turbidity, which often lead to detection errors [17], [18]. To address these issues, various deep learning models based on CNNs and YOLO architectures have been widely adopted because of their ability to balance detection speed and accuracy [19]. However, conventional YOLO architectures do not always exhibit stable performance under diverse and dynamically complex underwater environmental conditions. To overcome these limitations, several studies have proposed architectural modifications to enhance feature extraction capabilities, including the integration of Convolutional Block Attention Modules (CBAM) and Transformer modules into YOLOv5s, as reported in [20]. Although these approaches have been shown to improve detection accuracy, the introduction of additional components generally increases computational complexity while reducing inference efficiency, thereby limiting their applicability in real-time systems operating on resource-constrained devices. In the context of dolphin detection using deep learning approaches, research is still relatively rare and has not been widely developed. Early studies focused more on acoustic signal processing than on exploiting visual information. The study in [21] utilized a CNN to improve the reliability of dolphin whistle detection in complex and noisy acoustic environments. Furthermore, study [22] developed a ResNet15-based CNN model to recognize dolphin vocalizations, but this approach was not designed to support image-based detection and object localization.

Previous studies [23] attempted to overcome these limitations by applying the YOLOv8-nano architecture and providing a specific dataset for dolphin detection. However, the model showed inconsistent performance, particularly in detecting small and partially occluded dolphins in dynamic underwater environments with varying lighting and turbidity. In addition, the architecture relied on the design principles of previous generations of YOLO, which may limit the efficiency of feature representation in complex aquatic environments. These limitations prompted the exploration of newer generations of YOLO architecture. Specifically, to date, there have been no research reports examining the application of YOLO11-Nano as the latest detection architecture for dolphin identification in dynamic underwater environments. Furthermore, YOLO11-Nano involves less computational complexity and fewer parameters, making the proposed model more efficient.

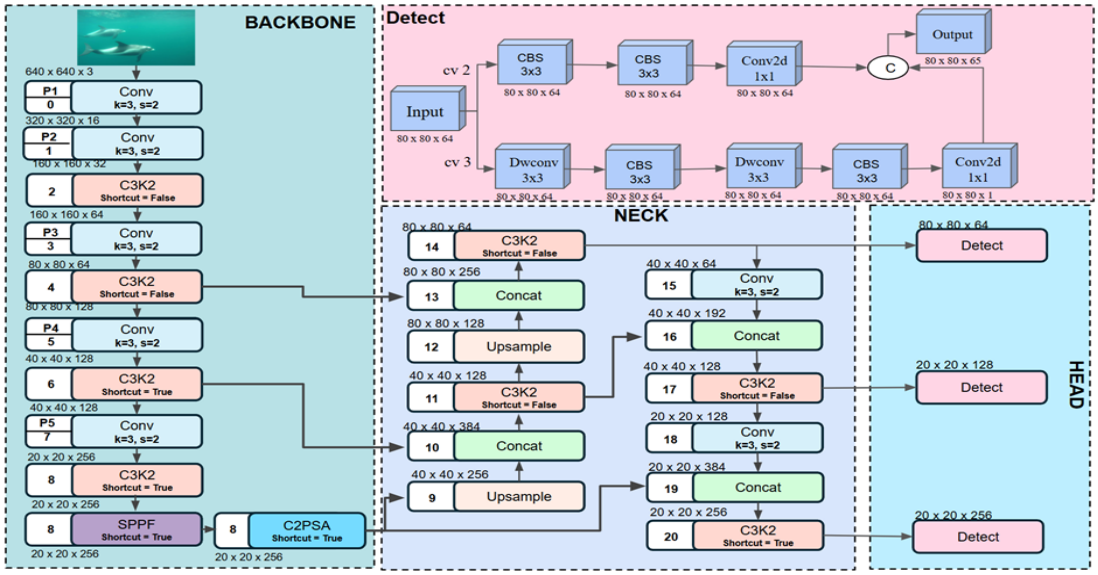

Figure 1. Architecture of YOLO11-Nano [30]

This study contributes by evaluating the application of YOLO11-Nano to identify dolphins in underwater environments. This approach has not been previously used or analyzed in computer vision-based. dolphin detection studies. Therefore, this study clearly presents a different contribution compared to previous studies. In addition to this novelty, the results of this study also show that YOLO11-Nano has better computational efficiency, as indicated by a lower GFLOPs value compared to the approach in previous studies. This low GFLOPs value reflects a smaller number of computational operations, resulting in lower total floating-point operations (FLOPs) in each network layer. With fewer total computational operations per layer, the overall computational complexity of the model can be reduced. This low computational complexity allows for faster inference processes to be performed more efficiently under limited computational resources without compromising detection performance

2. RESEARCH METHOD

This study proposes using YOLO11-Nano as a solution for real-time dolphin detection in complex aquatic environments, due to its high computational efficiency while maintaining competitive detection accuracy, making it ideal for implementation on devices with limited resources. This architecture is a lightweight YOLO family variant designed to balance inference speed and detection accuracy. Figure 1 presents the overall architecture, which is structured into three main components, namely the backbone, neck, and head.

2.1 Backbone

The backbone section in the YOLO11-Nano architecture serves as the primary component in the feature extraction process, responsible for learning the visual representation of input images. This backbone comprises multiple fundamental blocks arranged sequentially to achieve a balance between computational efficiency and feature representation capabilities. Each block is designed to progressively enrich visual information, resulting in increasingly discriminative feature maps. The main blocks that form the backbone include:

2.1.1 C3K2

The C3K2 module, as illustrated in Figure 2, begins with a 1 × 1 convolution that serves to adjust the channel dimensions, where some channels are used for feature extraction, and the rest are retained as identity connections to improve computational efficiency. This module supports two operational configurations: the C3K configuration, which consists of three convolutional layers with flexible kernel sizes, and the C2f configuration, which uses two 3 × 3 convolutional layers. The output from each C3K branch or bottleneck is then combined with identity features and further processed through 1 × 1 convolution to strengthen inter-channel interactions. This concise and flexible design enables C3K2 to produce rich and robust feature representations, making it suitable for computer vision tasks in underwater environments.

2.1.2 SPPF

Following the final C3K2 module, the backbone is extended with a Spatial Pyramid Pooling–Fast (SPPF) module to strengthen feature extraction, as presented in Figure 3. The process begins with a 1 × 1 convolution that serves to compress the channel dimensions so that the number of channels can be adjusted. Next, the resulting features are processed through three consecutive 5 × 5 MaxPooling layers, where each stage preserves the spatial dimensions while gradually expanding the receptive field. This approach allows the model to capture local and global information effectively without significantly increasing the computational load. The output from each pooling stage is then combined and reprocessed using 1 × 1 convolution to integrate multi-scale features into the final feature map. This design enables the SPPF module to consistently extract feature

representations at various scales, thereby contributing to improved detection accuracy without adding significant computational complexity.

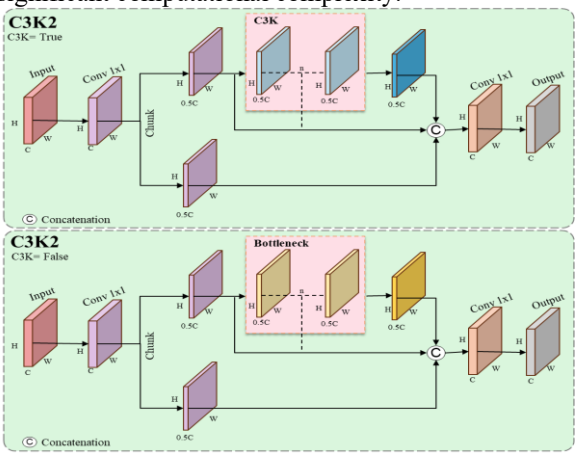

Figure 2. C3K2 module. The true configuration employs C3K blocks, whereas the False configuration utilizes bottleneck blocks [30]

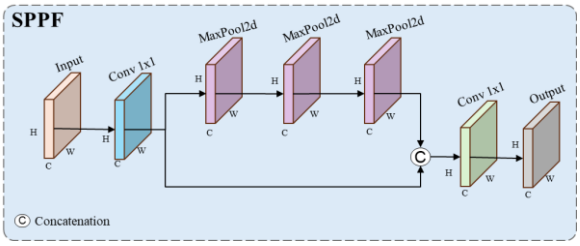

Figure 3. Design of the SPPF Module [30]

2.1.3 C2PSA

In the final stage of the backbone, YOLO11-Nano integrates the C2PSA module after the SPPF layer to enhance feature representation quality, as shown in Figure 4. This module begins with a 1 × 1 convolution to adjust the channel dimensions, followed by feature partitioning into multiple parallel branches. One branch is processed through a Partial Self-Attention (PSA) block [24] to model global relationships between features through a self-attention mechanism, and the other branch retains the original features to maintain representation efficiency. The outputs from all branches are then concatenated and processed through a 1 × 1 convolution at the final stage to integrate inter-channel information, resulting in a more discriminative feature map before being passed on to the next stage in the network.

2.2 Neck

The Neck section in the YOLO11-Nano architecture integrates feature information from various scales extracted by the backbone. This process involves up-sampling and down-sampling operations to align the dimensions of the feature map, thereby strengthening the connections between elements at different convolution stages. This module combines a PAN-inspired aggregation mechanism [25] and a multi-scale feature pyramid representation [26] to generate three hierarchical feature levels via bottom-up and top-down aggregation paths. The model integrates C3K2 blocks to capture deeper contextual information while combining features from multiple receptive fields, thereby improving multiscale feature representation without increasing computational

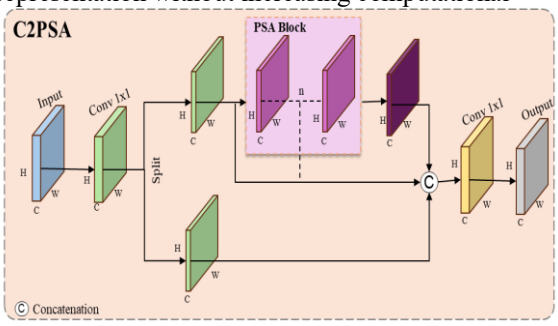

Figure 4. Cross-Stage Partial Parallel Split Attention (C2PSA) Module [30]

complexity, and CBS, a combination of convolution, batch normalization, and SiLU activation function, to enhance the model's capacity for recognizing complex patterns. The aggregated information is combined with features extracted from the backbone, strengthening the combined representation before it is forwarded to the detection head.

2.3 Head

The detection system in YOLO11-Nano generates prediction outputs in the Head section, which serves as the final stage for producing object detection results by leveraging feature representations extracted by the backbone and neck. In a single inference step, this component performs two main tasks simultaneously: category prediction and spatial localization. The detection head processes multi-scale feature representations at resolutions of 80×80, 40×40, and 20×20, enabling reliable detection of objects across different size ranges. This architecture uses three detection layers, each designed for a specific scale: large feature maps detect small objects, small feature maps handle large objects, and medium feature maps target medium-sized objects. Each detection layer consists of two parallel branches formed by successive 3×3 convolution layers followed by 1×1 convolution layers. The regression branch predicts the bounding box coordinates (x, y, w, h), while the classification branch estimates the probability of the detected object class.

This multi-scale detection strategy enables YOLO11-Nano to achieve high adaptability and robustness when detecting objects at different spatial resolutions. The optimization process utilizes a combination of three loss functions: Complete Intersection over Union (CIoU) Loss to maximize localization accuracy by considering the overlapping area, center distance, and aspect ratio [27]; Distribution Focal Loss (DFL) to improve bounding box regression by modeling localization targets as probability distributions [28], and Binary Cross-Entropy (BCE) Loss to optimize object prediction and class probability estimation by minimizing the divergence between predicted labels and actual labels. The integration of these loss functions ensures a balance between localization precision and

classification accuracy, enabling YOLO11-Nano to maintain optimal detection performance on objects of varying scales and complexities.

2.4 Computational Complexity

The computational complexity of the proposed model is measured using Giga Floating-Point Operations (GFLOPs), which represent the total number of floating-point operations required for a single forward pass. Following previous studies [29], the GFLOPs value is calculated as shown in Equation (1):

$$\text{GFLOPs} = 10^{-9}\,\text{X}\sum_{l=1}^{L} fl \quad (1)$$

Where fl indicates the number of floating-point operations executed at layer l, while L denotes the total count of layers involved in the network's computational process. The factor 10^{-9} is employed to convert the total number of floating-point operations into giga operations (GFLOPs).

For convolutional layers, which dominate the computational cost in YOLO-based architectures, the number of floating-point operations at layer l is calculated according to Equation (2):

$$fl = H \times W \times C_{out} \times (K_h \times K_w \times C_{in}) \quad (2)$$

Where $H$and Wrefer to the spatial size of the resulting feature map, C_{in} and C_{out} represent the numbers of input and output channels, and K_h and K_w correspond to the kernel dimensions. This formulation reflects the primary computational operations required to generate the complete output feature map for a convolutional layer.

2.5 Implementation Setup

To evaluate the effectiveness of the proposed method, experiments were conducted to achieve optimal performance while maintaining a balance between computational efficiency and model accuracy. The system configuration and dataset were carefully designed to align with the research objectives and to support the model training process optimally. Further implementation details will be explained below.

2.5.1 Training and Testing Configuration

In this experiment, the training and testing of the YOLO11-Nano model were conducted using fixed parameter configurations, as summarised in Table 1. The selection of training and testing parameters was based on settings commonly used in the YOLO family and previous object detection studies, to ensure stable convergence, fair comparison, and repeatability of results, rather than extensive hyperparameter optimisation.

Training was performed on the Kaggle platform with an NVIDIA Tesla P100 GPU, which provides high computing power for efficient model optimisation and prediction. An input resolution of

Table 1. Training and Testing Configuration

Parameters	Setup
Platfrom/device	Kaggle
GPU	P100
Image Size	640 x 640 pixels
Epochs	300
Batch Size	32
Optimizer	Stochastic Gradient Descent (SGD)
Learning Rate	0,01

640 × 640 pixels was chosen to achieve a balance between detail accuracy and computational efficiency. The model was trained for 300 epochs with a batch size of 32, ensuring stable convergence while effectively utilising GPU memory. Optimisation was performed using the Stochastic Gradient Descent (SGD) algorithm due to its robustness and consistent convergence in object detection tasks. A learning rate of 0.01 was applied to improve convergence and mitigate overfitting, thereby improving generalisation performance.

For the inference stage, model testing and evaluation were performed on a local device using a CPU configuration. This approach was chosen to test the extent to which the model can be applied in a real-world implementation environment with limited computational resources.

2.5.2 Dataset

This study utilised a dolphin dataset obtained from previous research [23]. The dataset was constructed by extracting image frames from field videos and previously published research sources [30], then thoroughly re-annotated to convert the data into a bounding box format compatible with the YOLO algorithm.

This dataset comprises 5,493 images, with 4,122 images for training, 822 samples for testing, and 549 images for validation, as shown in Table 2. Data augmentation was applied exclusively to the training set to improve model generalisation, while the validation and testing sets remained unchanged to ensure objective performance evaluation. This data separation strategy produces a representative and balanced dataset, supporting reliable model training and performance assessment. Each sample in the dataset is labelled with a bounding box to identify and determine the location of the dolphins.

Table 2. Dataset Configuration

Parameters	Setup
Training Data	4.122 images
Validation Data	822 images
Testing Data	549 images

This dataset comprises images above and below water, with varying lighting conditions, ocean backgrounds, and challenging detection scenarios such as water reflections and dynamic dolphin movements, as shown in Figure 5. In total, the dataset contains 5,493 images with approximately 4,900 labelled

Figure 5. The dolphin dataset captured under two conditions, above water and underwater

dolphin instances. Each instance represents the presence of a dolphin in the image, either as a single subject or as part of a group.

3. RESULTS AND DISCUSSION

The performance of the proposed YOLO11-Nano model was evaluated based on detection accuracy, computational efficiency, robustness to various aquatic environmental conditions, and execution time efficiency. The evaluation results show that YOLO11-Nano has relatively low, consistent, and stable computational efficiency, enabling it to maintain an optimal balance between inference speed and computational complexity. These characteristics support real-time applications on devices with limited resources in complex aquatic environments.

3. 1 Evaluation on Dataset

The detection performance of the proposed model was assessed using widely adopted evaluation metrics, including the average precision metric measured at an Intersection over Union (IoU) threshold of 0.5, referred to as mAP@0.5, as well as the mean of average precision scores computed across IoU thresholds from 0.5 to 0.95, denoted as mAP@0.5:0.95. As shown in Table 3, the YOLO11-Nano model achieved an mAP50 value of 65.0% and an mAP50:95 of 43.1%, with 2.59 million parameters and a computational complexity of 6.4 GFLOPs. This performance is compared to other lightweight detectors in the YOLO family, namely YOLOv12-Nano, which achieved an mAP50 of 60.8%, an mAP50:95 of 41.9%, 2.56 million parameters, and a complexity of 6.5 GFLOPs, and YOLOv10-Nano, which achieved an mAP50 of 59.6%, an mAP50:95 of 39.9%, 2.70 million parameters, and a complexity of 8.4 GFLOPs. Meanwhile, the YOLOv8-Nano model achieved mAP50 of 65.4% and mAP50:95 of 44.4%, but required a larger number of parameters (3.01 million) and higher computational complexity (8.2 GFLOPs) compared to YOLO11-Nano. Furthermore, the YOLOv8-Nano Best Channel variant showed the highest detection accuracy among all models compared, with mAP50 of 67.1% and mAP50:95 of 45.8%, using 1.83 million parameters and a computational complexity of 7.2 GFLOPs. This performance improvement demonstrates that the channel optimization strategy can significantly improve feature representation quality and detection performance. However, despite achieving the highest accuracy, this model still has greater computational complexity, so the total floating-point operations (FLOPs) in the network will be greater than YOLO11-Nano. Overall, these results show that YOLO11-Nano offers relatively low computational efficiency that is more consistent and stable for implementation on devices with limited resources in real-time detection applications.

Table 3. Evaluation Performance on Dataset

Model	GFLOPs	Parameter	mAP 50%	mAP 50-95%
YOLOv12-Nano	6.5	2.56	60.8	41.9
YOLOv10-Nano	8.4	2.70	59.6	39.9
YOLOv8-Nano [23]	8.2	3.01	65.4	44.4
YOLOv8-CR [23]	7.2	1.83	67.1	45.8
YOLO11-Nano	**6.4**	**2.59**	**65.0**	**43.1**

Figure 6 shows the results of dolphin detection using the YOLO11-Nano model in above- and below-water conditions. In general, the model is able to detect dolphins accurately, even when the object is only partially visible, is at a depth with low visibility, or is affected by surface light reflections. This demonstrates the model's ability to generalize and deal with variations in complex aquatic environmental conditions. However, some detection errors were still found. In some images, dolphins were detected more than once (multiple bounding boxes), while in other cases, some individuals were only partially detected or not detected at all. These conditions indicate that although the model has achieved a high level of precision, performance improvements are still needed so that detection accuracy can be further improved and detection errors can be minimized in future.

To further assess the classification performance of the proposed YOLO11-Nano model, the normalized confusion matrix is presented in Figure 7. It should be emphasized that the detection task in this study involves only one main object class, namely dolphins, which are distinguished from the background. Therefore, this confusion matrix represents object detection results, not conventional multi-class classification problems. Consequently, the visual structure of the confusion matrix appears relatively simple and less balanced when compared to multi-class scenarios.

The test results show that the model has a reliable ability to distinguish dolphin objects from the background class. In the dolphin category, the model produced a true prediction value of 0.56, indicating that dolphin features can be recognised well in the test dataset. The remaining value of 0.44 represents dolphin objects that were misclassified as background. This condition commonly occurs in underwater

environments due to the small size of objects, partial occlusion, low visual contrast, and blurring effects due to movement.

Figure 6. Dolphin detection results using the YOLO11-Nano model

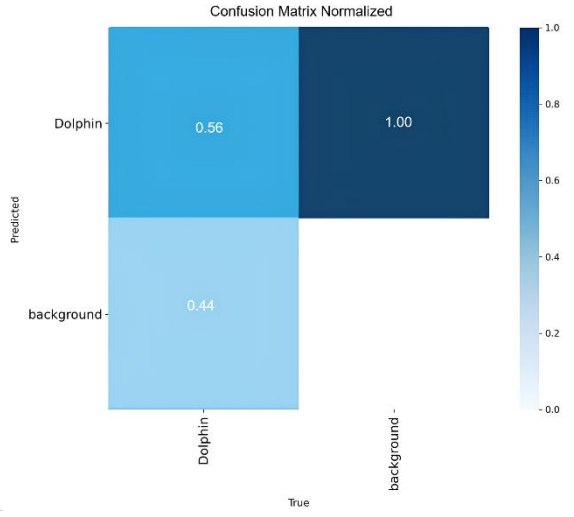

Figure 7. Normalized confusion matrix of the proposed YOLO11-Nano model on the test dataset.

Meanwhile, the background class was identified with a value of 1.00, indicating that the model was consistently able to recognise non-dolphin areas without classification errors. These results suggest the model's strong ability to suppress detection errors in background areas, which is particularly important given the high visual complexity and dynamic nature of underwater environments. Although the per-class analysis is limited as it only involves one foreground class, the confusion matrix still provides a clear picture of the main challenge in underwater dolphin detection, which lies in enhancing sensitivity without introducing additional false detections in non-target regions. Overall, these results demonstrate that YOLO11-Nano can effectively extract distinguishing characteristics between target objects and backgrounds, thereby supporting stable detection performance in underwater scenarios.

3. 2 Runtime Efficiency

The evaluation of execution time efficiency focuses on the inference performance and computational complexity of several official lightweight YOLO models. As shown in Table 4, the YOLO11-Nano model recorded the fastest inference speed of 18.34 FPS with a computational complexity of 6.4 GFLOPs. This combination results in a higher performance-to-computational ratio compared to other models, indicating more optimal utilization of computational resources. Specifically, YOLOv12-Nano achieves 16.96 FPS with 6.5 GFLOPs, while YOLOv10-Nano achieves 17.63 FPS with a significantly higher computational complexity of 8.4 GFLOPs. Despite producing competitive inference performance, the increased computational complexity of YOLOv10-Nano results in lower inference efficiency. Meanwhile, YOLOv8-Nano produces 17.92 FPS with a complexity of 8.2 GFLOPs, indicating that this model requires greater computational costs to achieve a level of inference performance comparable to YOLO11-Nano. This confirms that the standard YOLOv8 approach is still inefficient, making it less than optimal for resource-constrained device scenarios. In contrast, YOLO11-Nano shows the most balanced trade-off between inference speed and computational complexity, resulting in a more computationally efficient approach that is well suited to time-sensitive object detection implementation on edge devices, where low GFLOPs complexity directly contributes to reduced computing costs and increased inference efficiency.

Quantitatively, YOLO11-Nano demonstrates clear efficiency advantages over the official lightweight YOLO model. Compared to YOLOv12-Nano, YOLO11-Nano achieves a higher inference

speed with a difference of 1.38 FPS, namely 18.34 FPS compared to 16.96 FPS, and has a computational complexity of 0.1 GFLOPs, respectively 6.4 GFLOPs and 6.5 GFLOPs. Furthermore, compared with YOLOv10-Nano, the proposed model demonstrates a more significant performance improvement, achieving an inference speed increase of 0.71 FPS, with values of 18.34 FPS and 17.63 FPS, while reducing computational complexity by 2.0 GFLOPs, from 8.4 GFLOPs down to 6.4 GFLOPs. Furthermore, when compared to YOLOv8-Nano, YOLO11-Nano produces a relatively comparable inference speed with a difference of 0.42 FPS, namely 18.34 FPS compared to 17.92 FPS. However, this model still demonstrates superiority in terms of computational efficiency with a reduction in complexity of 1.8 GFLOPs, from 8.2 GFLOPs to 6.4 GFLOPs.

This reduction in GFLOPs provides significant advantages, especially for edge device applications. Models with lower computational complexity not only reduce power consumption and inference latency, but also hardware requirements. Thus, models with low computational complexity, such as YOLO11-Nano, not only improve inference efficiency but also expand the potential for object detection system applications in resource-constrained environments.

Table 4. Runtime Efficiency Comparison of Official YOLO-Nano Models

Model	GFLOPs	FPS
YOLOv12-Nano	6.5	16.96
YOLOv10-Nano	8.4	17.63
YOLOv8-Nano	8.2	17.92
YOLO11-Nano	**6.4**	**18.34**

The increase in YOLO11-Nano inference speed is due to its optimized architectural design, specifically the efficient combination of C3K2 and C2PSA modules that improve feature extraction while reducing redundant calculations. This balance between speed and accuracy ensures that the model not only excels in detection precision but also meets the practical needs for real-time dolphin monitoring in aquatic environments.

4. CONCLUSION

This study demonstrates the effectiveness of the YOLO11-Nano architecture to perform real-time dolphin detection in complex aquatic environments. This model achieves a mean Average Precision mAP50 of 65.0% and mAP50:95 of 43.1%, with 2.59 million parameters and a computational complexity of 6.4 GFLOPs, demonstrating its capability to maintain an effective trade-off between detection accuracy and computational efficiency. Compared to other variants such as YOLOv10-Nano and YOLOv12-Nano, the proposed model outperforms in terms of both accuracy and inference speed, attaining 18.34 FPS, confirming its potential for real-time application on devices with limited resources. As a major contribution, this study presents a comprehensive evaluation of the YOLO11-Nano architecture for underwater dolphin detection, highlighting its effectiveness as a lightweight yet accurate detection framework.

Furthermore, experimental results show that YOLO11-Nano is capable of overcoming various underwater challenges, including low visibility conditions, lighting variations, and surface light reflections. However, some detection inaccuracies were still found, such as overlapping bounding boxes or partial detection. Therefore, further research could focus on improving the attention mechanism to increase precision. Overall, this research provides valuable insights into lightweight deep learning models for marine species monitoring, while establishing a solid foundation for automated and efficient aquatic ecosystem monitoring systems.

Acknowledgment

The authors would like to acknowledge the **AIVISION** team for their valuable contributions and support throughout this study. Their expertise and insights in the field related to computer vision and deep learning, supported by access to computational facilities and technical assistance, played an important role in supporting the experimental process and the preparation of this manuscript.

5. REFERENCE

[1] A. Homoud, S. Das, and S. Townley, 'Challenges in underwater object detection and video segmentation using deep learning', in *2024 First International Conference for Women in Computing (InCoWoCo)*, 2024, pp. 1–6..

[2] Y. Zhang and X. Zhang, 'YOLO-AES A Lightweight Model for Real-Time Underwater Object Detection', in *2024 5th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, 2024, pp. 611–615.

[3] M. Elmezain, L. S. Saoud, A. Sultan, M. Heshmat, L. Seneviratne, and I. Hussain, 'Advancing underwater vision: a survey of deep learning models for underwater object recognition and tracking', *IEEE Access*, 2025.

[4] L. Chen *et al.*, 'Underwater Optical Object Detection in the Era of Artificial Intelligence: Current, Challenge, and Future', *ACM Computing Surveys*, 2025.

[5] M. D. Putro, Y. Mose, A. C. Andaria, J. Litouw, V. C. Poekoel, and X. Najoan, 'Streamlining Deep Learning Network for Real-time Sea Turtle Detection', *Jurnal Rekayasa Elektrika Vol*, vol. 20, no. 3, pp. 116–124, 2024.

[6] S. B. Kurniawan et al., "Tackling marine pollution in the blue economy: Synergies between wastewater treatment technologies and governmental policies," Marine Pollution Bulletin, vol. 212, p. 117431, Jan. 2025, doi: 10.1016/j.marpolbul.2024.117431.

[7] A. March, M. Bennett, M. Germishuizen, T. Evans, and P. Failler, "The status of Blue Economy development in Africa," Marine Policy, vol. 165, p. 106205, Jul. 2024, doi:

10.1016/j.marpol.2024.106205.

[8] H. Zhang, Q. Zhang, P. A. Nguyen, V. C. S. Lee, and A. Chan, 'Chinese white dolphin detection in the wild', in *Proceedings of the 3rd ACM International Conference on Multimedia in Asia*, 2021, pp. 1–5.

[9] M. F. Riftiarrasyid, F. L. Gaol, H. Soeparno, and Y. Arifin, 'Suitability of Latest Version of YOLOv11 in Drone Development Studies', in 2024 7th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2024, pp. 504–509.

[10] S. Cheng, Y. Han, Z. Wang, S. Liu, B. Yang, and J. Li, 'An underwater object recognition system based on improved yolov11', *Electronics*, vol. 14, no. 1, p. 201, 2025.

[11] R. Maglietta, R. Carlucci, C. Fanizza, and G. Dimauro, 'Machine learning and image processing methods for cetacean photo identification: a systematic review', *IEEE Access*, vol. 10, pp. 80195–80207, 2022.

[12] W. Ouyang and Y. Wei, 'An anchor-free detector with channel-based prior and bottom-enhancement for underwater object detection', *IEEE Sensors Journal*, vol. 23, no. 20, pp. 24800–24811, 2023.

[13] M. D. Putro, A. Sutrisno, I. S. Manembu, I. Y. Chun, and T.-H. Oh, 'STAR: Sea Turtle Basic Activity Recognizer Network Via Efficient Transformer', *IEEE Access*, 2025.

[14] M. Jain and A. Shah, 'Anomaly Detection Using Convolutional Neural Networks (CNN)', *ESP International Journal of Advancements in Computational Technology* (ESP-IJACT), vol. 2, no. 3, pp. 12–22, 2024.

[15] T. Diwan, G. Anirudh, and J. V. Tembhurne, 'Object detection using YOLO: challenges, architectural successors, datasets and applications', *multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, 2023.

[16] R. Khanam and M. Hussain, 'Yolov11: An overview of the key architectural enhancements', *arXiv preprint arXiv*:2410. 17725, 2024.

[17] V. Mane, S. Patwardhan, P. Pethkar, and R. Patil, 'Underwater object tracking and classification of marine animals', in 2024 *International Conference on Inventive Computation Technologies* (ICICT), 2024, pp. 1054–1058.

[18] P. Vijayalakshmi, M. Seetharaman, and E. Praveen, 'Underwater Image Enhancement and Object Recognition Using CNN Algorithm', in 2024 5th *IEEE Global Conference for Advancement in Technology* (GCAT), 2024, pp. 1–5.

[19] [20] C. Prathima, C. Silpa, A. Charitha, G. Harshitha, C. S. Charan, and G. R. Sailendra, 'Detecting and Recognizing Marine Animals Using Advanced Deep Learning Models', in 2024 *International Conference on Expert Clouds and Applications* (ICOECA), 2024, pp. 950–955.

[20] X. Xu, J. Hu, J. Yang, Y. Ran, and Z. Tan, 'A fish detection and tracking method based on improved inter-frame difference and YOLO-CTS', *IEEE Transactions on Instrumentation and Measurement*, 2024.

[21] Nardo, 'Convolutional Neural Networks for enhancing detection of Dolphin whistles in a dense acoustic environment', *IEEE Access*, 2024.

[22] G. Frainer et al., 'Automatic detection and taxonomic identification of dolphin vocalisations using convolutional neural networks for passive acoustic monitoring', *Ecological Informatics*, vol. 78, p. 102291, 2023.

[23] F. Ludja, R. M. Lintong, F. Sumarauw, A. M. Sambul, S. R. Sentinuwo, and M. D. Putro, 'DOLPHIN DETECTION USING AN ENHANCED LIGHTWEIGHT YOLO ARCHITECTURE', JIPI (*Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*), vol. 10, no. 3, pp. 2874–2884, 2025.

[24] H. Zhang, K. Zu, J. Lu, Y. Zou, and D. Meng, 'EPSANet: An efficient pyramid squeeze attention block on convolutional neural network', in *Proceedings of the asian conference on computer vision*, 2022, pp. 1161–1177.

[25] Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.

[26] Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

[27] Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Adv. Neural Inf. Process. Syst. 2020, 33, 21002–21012.

[28] Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. Proc. AAAI Conf. *Artif. Intell*. 2020, 34, 12993–13000. [CrossRef]

[29] T. Wang, H. Wang, W. Wang, K. Zhang, B. Ye, and H. Dong, 'F3M: A Frequency-Domain Feature Fusion Module for Robust Underwater Object Detection', *Journal of Marine Science and Engineering*, vol. 14, no. 1, p. 20, 2025.

[30] C. Trotter et al., 'NDD20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation', *arXiv preprint* arXiv:2005. 13359, 2020.

[31] H. Wang and J. Zhao, 'Research on Defect Detection on Steel Rails Based on Improved YOLO11n Algorithm', *Applied Sciences*, vol. 16, no. 2, p. 842, 2026.