JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 3, September 2025, Pp. 2874-2884

DOLPHIN DETECTION USING AN ENHANCED LIGHTWEIGHT YOLO ARCHITECTURE

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

1. Department of Informatics, Faculty of Engineering, Universitas Sam Ratulangi Manado, Indonesia
2. Department of Informatics, Faculty of Engineering, Universitas Sam Ratulangi Manado, Indonesia
3. Department of Informatics, Faculty of Engineering, Universitas Sam Ratulangi Manado, Indonesia
4. Department of Informatics, Faculty of Engineering, Universitas Sam Ratulangi Manado, Indonesia
5. Department of Informatics, Faculty of Engineering, Universitas Sam Ratulangi Manado, Indonesia
6. Department of Informatics, Faculty of Engineering, Universitas Sam Ratulangi Manado, Indonesia

Article Info

Keywords: Deep Learning; Dolphin Detection; Lightweight Architecture; Underwater Object Detection; YOLOv8

Article history:
Received 16 September 2025
Revised 25 September 2025
Accepted 4 October 2025
Available online 6 October 2025

DOI :
https://doi.org/10.29100/jipi.v10i3.9169

* Corresponding author.
Corresponding Author
E-mail address:
febriyantiludja026@student.unsrat.ac.id

ABSTRACT

Dolphin detection plays an important role in marine ecosystem monitoring, species conservation, and behavioral analysis. However, visual identification in underwater environments faces challenges such as light refraction, water turbidity, and dynamic sea conditions. This study proposes a deep learning-based dolphin detection approach by modifying the YOLOv8 architecture to produce a lightweight yet accurate model. The modifications include reducing the number of channels in the backbone and neck, as well as simplifying the SPPF block, thereby reducing the model parameters from 3.01 million to 1.83 million and the computational complexity from 8.2 GFLOPs to 7.2 GFLOPs. A specialized dolphin dataset consisting of 5,493 labeled images, collected from underwater and surface conditions, was developed to train and evaluate the model. Experimental results show that the proposed model achieves 67.1% mAP@50 and 45.8% mAP@50–95, outperforming YOLOv8-Nano and other lightweight YOLO variants. Additionally, the model demonstrates better runtime efficiency, with a latency of 49.2 ms and 20.38 FPS, making it suitable for real-time implementation on resource-constrained devices. Overall, this research presents a more efficient and accurate dolphin detection solution, while also providing a specialized dataset that can support further research in the field of computer vision-based marine conservation.

I. INTRODUCTION

Underwater object detection plays an important role in marine science, supporting ecosystem monitoring, environmental exploration, and species conservation [1]. In the context of conservation, several dolphin species still have large populations. Common dolphins, in particular, are estimated to number more than six million individuals worldwide, reflecting the species' successful adaptation to various aquatic environments. A comprehensive study of marine mammals in the North Atlantic also emphasizes the importance of long-term monitoring and cross-regional population status assessments as the basis for sustainable conservation management [2]. However, this situation starkly contrasts with the Irrawaddy dolphin population in Southeast Asia, which is experiencing a critical decline. A 2020 long-term monitoring report on the Mekong River estimated that only about 89 individuals remain (95% CI: 78–102), with an average annual mortality rate of 2.14% and a population decline rate of 2.09% per year since 2007 [3]. This data confirms the urgency of the need for an efficient monitoring approach. Advances in computer vision and artificial intelligence (AI) have significantly improved detection accuracy, enabling more effective object identification and tracking. However, developing models that can operate reliably in a variety of underwater conditions, from deep-sea ecosystems to coastal and river areas, remains a significant challenge, requiring algorithms to ensure consistent detection performance [4]. One application of this technology is dolphin detection, to understand behavior, movement patterns, geographic distribution, and population dynamics. Underwater cameras and other visual monitoring tools enable direct observation in two main conditions, underwater and on the water surface. However, factors such as light refraction, water

turbidity, and unstable lighting can reduce image quality [5]. To overcome these challenges, computer vision systems must be trained on data sets that cover a variety of real-world environmental conditions.

In addition, dolphin monitoring is also important for detecting changes in population, behavior, and migration patterns, which are important indicators for maintaining the balance of the marine ecosystem [6]. Conventional methods that rely on human observation have limitations in terms of time, area, and safety, making them ineffective for long-term monitoring. Therefore, computer vision-based automated monitoring systems offer a more efficient and sustainable approach [7]. Supported by a Convolutional Neural Network (CNN) architecture, this system is capable of detecting dolphins in real time, even in complex visual conditions in the marine environment, thereby providing accurate data to support conservation efforts.

With advances in computer vision technology, object detection has become a key component in computer vision systems designed to identify and localize the presence of specific objects in images or videos. However, in underwater environments, object detection faces various challenges such as lighting conditions, water turbidity, and complex and dynamic backgrounds [8]. To overcome these problems, deep learning-based approaches have emerged as a solution due to their ability to extract features from image data.

The modern approach introduces Convolutional Neural Networks (CNN) as a feature extraction method that effectively distinguishes object features from the background [9]. Filter operations can recognize the characteristics of objects using trained weights. During the learning process, the network automatically updates the kernel weights by minimizing prediction errors during training. This method effectively highlights important information from objects, but requires significant computing power when using deep convolutional layers. Among CNN-based architectures, You Only Look Once (YOLO) is widely used due to its efficiency in real-time applications.

Previous YOLO-based research has generally focused on detecting land objects or environments with stable lighting conditions, while its application in marine environments faces different challenges. Prior versions of the YOLO model, including the original YOLOv8, tend to have difficulty distinguishing between marine objects with similar morphologies, such as dolphins and other species with similar dorsal fin shapes. In addition, detection performance tends to decline when images are affected by water turbidity, light reflections, and limited contrast. This gap is the basis for developing the YOLOv8-nano architecture, aiming to improve the ability to discriminate between marine species while maintaining computational effectiveness for real-time monitoring in the field.

The advantages of the CNN method prompted this study to implement a convolution-based architecture with a focus on efficiency. This study proposes a low-cost architecture that can be run on devices with limited resources. Efficiency is achieved through simplification of the network structure, particularly by reducing the number of channels in each layer. This allows for fast and lightweight dolphin feature extraction without sacrificing accuracy. YOLOv8 has demonstrated strong object detection performance in low-light conditions, dense object distribution, and suboptimal image quality[10]. However, further efficiency improvements are still needed for real-world applications. The main contributions of this research are summarized as follows.

1) An efficient deep learning model is proposed to develop a real-time dolphin detection system capable of rapid operation and deployment on low-power devices. The proposed architecture is a simplification of YOLOv8, which is focused on improving computational efficiency without sacrificing accuracy, thus enabling applicability to ocean monitoring robotic systems.
2) To support computer vision-based object detection systems, this study introduces a new dataset specifically designed for dolphin detection, annotated in a format compatible with YOLO. The dataset comprises images of dolphins captured under various conditions, both underwater and at the sea surface. Given the limited availability of public datasets for dolphin detection, this contribution is expected to serve as a valuable resource for advancing AI-based ocean monitoring systems.
3) Extensive evaluation of the proposed lightweight model shows superior performance reaching state-of-the-art performance in efficient object detection tasks, exceeding some mild variants of the YOLO family on the proposed dolphin dataset. This study also evaluates the model's computational efficiency, highlighting its suitability for rapid dolphin localization on resource-constrained devices.

II. RELATED WORKS

Several studies have shown that the effectiveness of detection systems can be improved by expanding the object detection approach to various visual entities that impact marine life. For example, fishing nets, fishing

boats, plastic waste, and other fishing gear can be included as additional classes in detection [11], [12], [13]. The application of this system enables the identification of marine animals and potential threats in their vicinity, thereby providing richer contextual information to support conservation [14].

Underwater object detection systems face significant challenges in the field of computer vision due to low visibility, poor lighting, and water turbidity, which often lead to detection errors [14], [16]. To address these issues, deep learning models such as Convolutional Neural Networks (CNNs) and You Only Look Once (YOLO) architectures have been widely used due to their ability to balance speed and accuracy [17]. However, these architectures are not always optimal in all underwater conditions, so various modifications have been proposed. For example, study [18] proposed YOLO-CTS by adding a Convolutional Block Attention Module (CBAM) and a Transformer module to YOLOv5s to improve feature extraction in complex environments. On the other hand, study [19] introduced YOLOX-U, an anchor-free detector optimized for small objects by considering light attenuation in color channels, outperforming YOLOv8-S in detecting sea cucumbers. Although significant progress has been made in detecting marine animals such as fish [18], sharks [15], and turtles [20], research focusing on dolphin detection is still limited. Most studies focus on acoustic detection, such as dolphin voice recognition or whistle detection [21], so the application of visual object detection methods is still very limited and requires further research.

Globally, the proposed architecture is inspired by YOLOv8 [10], which is designed to be lighter. The general structure of the detector architecture can be seen in Figure 1. This study modifies the nano version of YOLOv8 by increasing data processing speed without compromising detection performance. The focus of this research is to improve model efficiency by optimizing the number of channels in the feature map during convolution operations, with the aim of improving the accuracy of dolphin location predictions.

The implementation of dolphin detection systems plays an important role in supporting the conservation of protected marine species [21]. Computer vision and deep learning technologies have become popular methods for automatically recognizing marine objects in various types of data. One study applied this approach to recordings from unmanned aerial systems (UAS) to gain new insights into marine populations [22],[23]. Object detection models such as YOLO have demonstrated high performance in real-time detection tasks, even under challenging conditions such as low light and visual disturbances underwater [17],[15]. Since dolphins are difficult to observe directly due to their behavior and habitat, conventional monitoring requires high costs and field risks. Many dolphin species are currently classified as vulnerable, endangered, or data deficient in the International Union for Conservation of Nature (IUCN) Red List for Nature Conservation [24]. Therefore, global conservation efforts need to be supported by in-depth analysis enabled by modern detection technology. Adaptive and efficient automated monitoring systems are urgently needed, with the integration of AI and computer vision offering promising potential solutions.

III. RESEARCH METHODOLOGY

In this section, the proposed architecture is described in detail, focusing on modules designed to improve dolphin detection performance. In real-world conditions, dolphins are often found in the same habitats as other large marine animals such as sharks, orcas, and whale sharks. Their similarities lie in their streamlined and hydrodynamic body shapes, making them difficult to distinguish at a glance, especially in underwater recordings or images with limited lighting. However, dolphins have distinctive characteristics in their unique skin texture patterns, snout shapes, and swimming behaviors that set them apart from other species. Therefore, recognizing specific visual characteristics is key to ensuring the detection system can accurately identify dolphins in complex marine environments.

A. Backbone

This approach enables the resulting vision system to be implemented on low-cost devices, thereby supporting the development of underwater robots and marine observation systems for comprehensive monitoring of marine life. The system uses a backbone as the main extractor to capture essential features while reducing the dimension of the feature map, thereby reducing the computational cost. This process utilizes convolution layers that efficiently enhance feature representation. On the other hand, this design enhances multi-kernel weighting to accommodate richer information from the extracted features. This approach is a common characteristic of CNN architecture, where smaller spatial dimensions are typically integrated with a larger number of channels.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 3, September 2025, Pp. 2874-2884

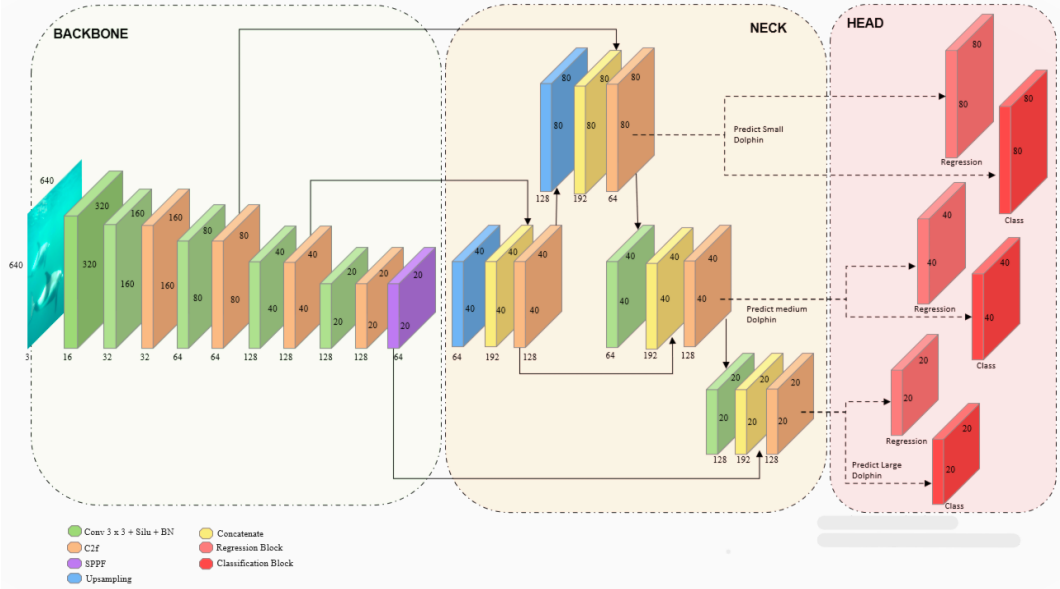

Figure. 1. Lightweight YOLOv8 Architecture for Dolphin Detection.

In this basic network, the YOLOv8 architecture integrates a fast convolution module known as Convolutional Two Faster (C2f). This block diagram is shown in Figure 2. The C2f module is an optimized version of the C2 module, which enhances computational efficiency while maintaining performance. This module also serves as a modification of the C3 module used in YOLOv5. In the initial stage, the input feature map is compressed using 1×1 convolutions with a stride of 1 to adjust the number of channels. The output is then split into two branches, with one branch passed directly, while the second branch passes through a series of n Bottleneck blocks. The two branches are then concatenated, and the resulting feature map is further compressed using 1×1 convolutions to mix information from both paths. This design allows the C2f module to retain the information flow from each Bottleneck block, enriching the feature representation and improving detection performance, especially for small objects. This module is integrated into the backbone and neck parts of YOLOv8, where it optimizing feature maps across all layers of the network. This design improves information flow and preserves critical features, especially for small objects, resulting in richer and more diverse feature representations.

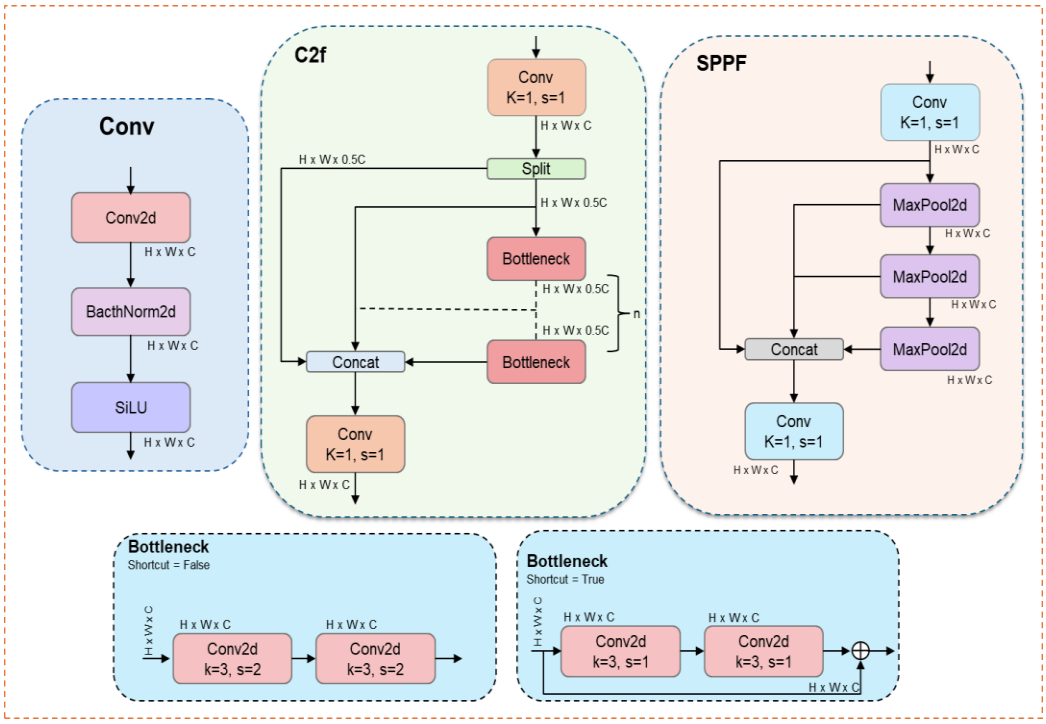

Figure. 2. Proposed Conv, C2f, SPPF, and Bottleneck blocks.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 3, September 2025, Pp. 2874-2884

In addition, spatial pyramid pooling-fast (SPPF) is applied after the last C2F module on the backbone to enrich the feature representation before entering the neck section. The block diagram of this process is illustrated in Figure 2. The process begins with a 1×1 convolution, which compresses the channels to adjust their number. The result is then processed through three consecutive 5×5 MaxPooling layers, where each layer preserves the spatial dimensions while gradually expanding the receptive field. This approach allows the model to capture both local and global information without significantly increasing the computational load. The output from each pooling stage is then concatenated and further processed using a 1×1 convolution to mix multi-scale information into the final feature map. This design enables the SPPF block to consistently extract features at various scales, thereby improving detection accuracy, especially for objects of varying sizes.

The SPPF block is positioned at the terminal stage of the backbone, preceding the neck section, where the channel dimensionality is compressed relative to that in the C2f module of the backbone. The channel reduction at the backbone-neck transition aims to reduce computational complexity and balance the feature load to be fused in the neck.

In the spine and neck, the number of outlets in several main layers is reduced, as shown in Table 1. The first modification occurs in the backbone, where the number of channels in the last Conv layer is reduced from 256 to 128, thereby reducing the parameters from 295,424 to 147,712, and in the next C2f layer, where the parameters are reduced from 460,288 to 115,456. Next, in the SPPF module, the number of channels was reduced by half from 256 to 64, which significantly reduced the parameters from 164,608 to 24,832. This reduction is important because the SPPF module transfers features to the neck section, and reducing the channels helps balance the feature load. In the neck section, the Upsample stage adjusts the number of channels to the previous output. Meanwhile, the Concat layer, which previously produced 384 channels, is reduced to 192, and the final C2f layer in the neck, which originally operated with 256 channels, is reduced to 128, with parameters reduced from 493,056 to 123,648. Overall, this channel reduction strategy resulted in a total of 3,011,043 parameters in the original model being reduced to 1,837,283 in the modified model, while also reducing computational complexity from 8.2 GFLOPs to 7.2 GFLOPs, resulting in a lighter and more efficient architecture.

TABLE I.
CONFIGURATIONS YOLOV8 ARCHITECTURE

Original			Modifikasi		
Layer	Output Shape	Parameter	Layer	Output Shape	Parameter
Input	640, 640, 3	0	Input	640, 640, 3	0
Conv	320, 320, 16	464	Conv	320, 320, 16	464
Conv	160, 160, 32	4672	Conv	160, 160, 32	4672
C2f	160, 160, 32	7360	C2f	160, 160, 32	7360
Conv	80, 80, 64	18560	Conv	80, 80, 64	18560
C2f	80, 80, 64	49664	C2f	80, 80, 64	49664
Conv	40, 40, 128	73984	Conv	40, 40, 128	73984
C2f	40, 40, 128	197632	C2f	40, 40, 128	197632
Conv	20, 20, 256	295424	Conv	20, 20, 128	147712
C2f	20, 20, 256	460288	C2f	20, 20, 128	115456
SPPF	20, 20, 256	164608	SPPF	20, 20, 64	24832
Upsample	40, 40, 256	0	Upsample	40, 40, 64	0
Concat	40, 40, 384	0	Concat	40, 40, 192	0
C2f	40, 40, 128	148224	C2f	40, 40, 128	123648
Upsample	80, 80, 128	0	Upsample	80, 80, 128	0
Concat	80, 80, 192	0	Concat	80, 80, 192	0
C2f	80, 80, 64	37248	C2f	80, 80, 64	37248
Conv	40, 40, 64	36992	Conv	40, 40, 64	36992
Concat	40, 40, 192	0	Concat	40, 40, 192	0
C2f	40, 40, 128	123648	C2f	40, 40, 128	123648
Conv	20, 20, 128	147712	Conv	20, 20, 128	147712
Concat	20, 20, 384	0	Concat	20, 20, 192	0
C2f	20, 20, 256	493056	C2f	20, 20, 128	123648
Detect	128	751507	Detect	128	604051
Total Params : 3, 011, 043			Total Params : 1, 837, 283		
Total GFLOPs : 8.2			Total GFLOPs : 7.2		

In the backbone and neck sections, the number of output channels in several main layers was reduced, as shown in Table 1. The first modification occurs in the backbone, where the number of channels in the last Conv layer is reduced from 256 to 128, thereby reducing the parameters from 295,424 to 147,712, and in the next C2f layer, where the parameters are reduced from 460,288 to 115,456. Next, in the SPPF module, the number of channels was reduced by half from 256 to 64, which significantly reduced the parameters from 164,608 to 24,832. This

reduction is important because the SPPF module transfers features to the neck section, and reducing the channels helps balance the feature load. In the neck section, the Upsample stage adjusts the number of channels to the previous output. Meanwhile, the Concat layer, which previously produced 384 channels, is reduced to 192, and the final C2f layer in the neck, which originally operated with 256 channels, is reduced to 128, with parameters reduced from 493,056 to 123,648. Overall, this channel reduction strategy resulted in a total of 3,011,043 parameters in the original model being reduced to 1,837,283 in the modified model, while also reducing computational complexity from 8.2 GFLOPs to 7.2 GFLOPs, resulting in a lighter and more efficient architecture.

Although a large number of channels can capture richer feature representations, not all extracted information is discriminative, some contains redundant information that actually increases computational costs. Reduced the number of channels in the backbone encourages the network to focus on extracting truly relevant features, resulting in more concise and efficient representations. This is particularly important because the neck acts as a multi-scale feature aggregator, when the number of channels is too high, information imbalance can hinder effective cross-scale aggregation. The proposed channel reduction strategy not only improves computational efficiency but also contributes to improved detection performance.

B. Neck

In YOLOv8, the Neck network determines the quality of features for object detection by integrating multi-level features extracted by the backbone network. Feature fusion enhances connections through up-sampling and down-sampling approaches to equalize feature map dimensions, thereby strengthening the connections between elements at each convolution stage. This module uses an optimized version of the Path Aggregation Network (PANet), which has been improved to enhance information flow between different feature levels. YOLOv8 implements a path aggregation network (PAN) that generates three feature levels using a bottom-up path aggregation strategy. It uses a C2F extractor to filter aggregated information. The process begins by taking features from several backbone stages, performing up-sampling and concatenation with resolution features through skip connections, followed by processing using an efficient C2f block. These features are then downsampled to combine information from different paths, processed again with C2f, and the results are passed to the Head for final prediction. The neck in YOLOv8 is lighter than previous generations because the C3 block is replaced with C2f, thereby improving inference speed without sacrificing accuracy.

C. Detection Layer and Loss

In YOLOv8, the detection layer generates predictions for bounding boxes and object classes. This work places the detection layer at the end of the network. The head network operates on several feature map scales, namely 80×80, 40×40, and 20×20. This architecture uses three detection layers with different assignments: large feature maps are used to detect small objects, small feature maps focus on large objects, and medium feature maps handle medium-sized objects. Each detection layer consists of two branch blocks formed by two 3×3 convolutional layers followed by one 1×1 convolutional layer, applied sequentially. For each scale, there are two branches: a regression branch that predicts the bounding box coordinates (x, y, w, h) and a classification branch that produces the probability of the detected object's class. This multi-scale approach enables YOLOv8 to detect objects of varying sizes more effectively.

In addition, the loss function in YOLOv8 consists of three main components, namely *Binary Cross-Entropy* (BCE) Loss, *Distribution Focal Loss* (DFL), and *Complete IoU* (CIoU) Loss. BCE Loss is used to measure class probability prediction errors. DFL is applied to bounding box regression by modeling the distribution of object boundary coordinates, thereby improving edge localization precision. Meanwhile, CIoU Loss evaluates the quality of predicted bounding boxes in regression tasks by considering overlap, distance, and aspect ratio. The integration of these three loss function components enables YOLOv8 to maintain a balance between detection accuracy and classification performance, ensuring the model's performance remains optimal across objects of varying scales.

IV. CONFIGURATION TRAINING AND TESTING

In this experiment, the training phase used the Kaggle platform, which provides an accessible and efficient environment for deep learning experiments. Training was performed on a Kaggle P100 GPU, which offers significant computing power to support complex model training. It was proposed to train for 300 epochs, which was identified as the optimal configuration based on the experimental results. Additionally, both training and evaluation used an input resolution of 640 × 640 pixels, consistent with the default YOLO settings. To optimize the

learning process on large datasets, *Stochastic Gradient Descent* (SGD) was used with a learning rate of 0.01 and a batch size of 32 to ensure stable convergence. During inference, model testing and evaluation were conducted on a local device using a CPU configuration. This approach was used to evaluate how well the model could be applied in a real-world implementation environment with limited computational resources.

V. PROPOSED DATASET

This dolphin dataset was specifically developed to detect dolphins in aquatic environments, as illustrated in Figure 3. The dataset consists of above-water and underwater images, with varying lighting conditions, ocean backgrounds, and detection challenges such as water reflections and dynamic movements. This dataset is divided into three non-overlapping subsets, namely training, validation, and testing, consisting of a total of 5,493 images.

The dataset was constructed by extracting frames from field videos and previous research sources [25]. The data initially collected was not annotated in a format compatible with the YOLO algorithm, so a re-annotation process was necessary to standardize it. The extracted images were carefully selected based on visual quality, object clarity, and water condition diversity.

This selection process ensures that the dataset provides a comprehensive representation of real-world scenarios, including variations in dolphin poses, lighting conditions, and movement dynamics within their natural habitat. The annotation was performed using the Boobs–YOLO Bounding Box Annotation Tool, with each image labeled using two primary attributes: class and bounding box. The class attribute specifies the object category, while the bounding box defines the object's location by enclosing its boundaries within a rectangle. The annotation process follows the YOLO format [class, x center ,y center, width, height], enabling the dataset to be directly utilized for training object detection models. During annotation, two key principles were strictly followed [26]. First is consistency; class definitions and bounding box placements must be consistent. Second is completeness and accurac, all object instances must be labeled comprehensively and accurately.

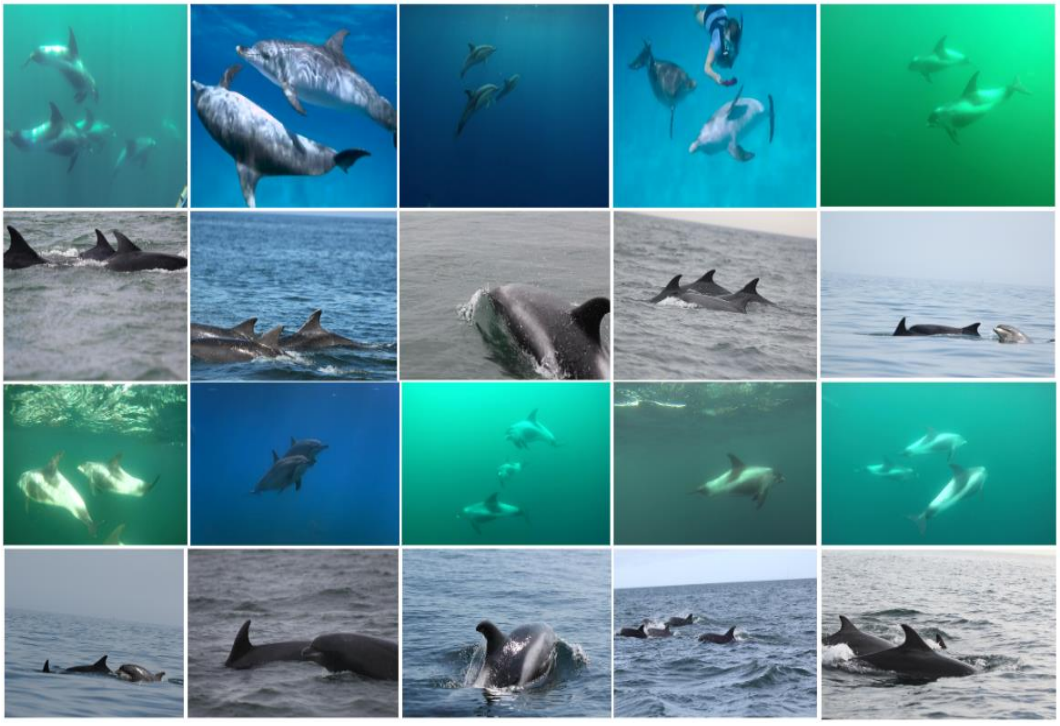

Figure. 3. Dolphin Dataset With Samples From Above and Below the Water Surface

Overall, this dataset consists of 5,493 images containing approximately 4,900 identified and labeled dolphin instances. Each instance represents the presence of an individual dolphin in an image, either alone or in a group. These instances vary in terms of visual conditions, including dolphins visible above the water surface, partially visible dolphins (e.g., fins or tails), and dolphins fully submerged underwater. Additionally, some instances depict dolphins in groups, which poses additional challenges in the detection process due to object overlap. This diversity provides a more realistic representation of dolphin behavior dynamics in their natural habitat and enriches visual feature variation, thereby enhancing the potential for object detection model generalization.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 3, September 2025, Pp. 2874-2884

In addition, the distribution of the dataset shows that all images were obtained during daylight hours, as the dataset was collected from internet sources with no record of pictures taken at night. Regarding water clarity, 333 images were taken in murky water, while the remaining samples correspond to clear water conditions. Depth values cannot be explicitly defined because the dataset was compiled exclusively from online sources, and no metadata is available regarding the vertical position or depth at which the images were obtained.

The dolphin dataset is divided into three main parts: 75% for training, 10% for validation, and 15% for testing, consisting of 4,122, 549, and 822 images, respectively. The training part is completely separate from the validation and testing parts, while the validation and testing parts are designed to include diverse and interrelated images, which collectively represent a variety of conditions. Data augmentation is applied exclusively to the training set, while the validation and testing sets remain unchanged to ensure objective evaluation. This division strategy produces a representative and balanced dataset, supporting effective model training and reliable performance assessment.

VI. RESULTS AND DISCUSSION

In this section, the performance of the proposed model is evaluated on a dolphin dataset and compared with lightweight YOLO-based object detection methods. The analysis focuses on two main aspects: Dataset Evaluation, which assesses detection accuracy and robustness in real underwater conditions, and Execution Time Efficiency, which analyzes computational cost and model suitability for real-time deployment. This evaluation highlights the effectiveness and practicality of the proposed architecture in addressing the challenges of underwater object detection.

A. Evaluation on Dataset

The proposed model was evaluated on a dolphin dataset covering various conditions, including variations in lighting, water turbidity, and dolphin movement. Detection accuracy was measured using standard evaluation metrics, namely average precision at an IoU threshold of 0.5 (mAP@50) and average IoU from 0.5 to 0.95 (mAP@50:95). This approach was compared to efficient detectors such as the lightweight YOLO family: YOLOv3-Tiny, YOLOv5-Nano, YOLOv6-Nano, YOLOv8-Nano, YOLOv9-Tiny, YOLOv10-Nano, YOLOv11-Nano, YOLOv12-Nano, and YOLOv12-Nano Turbo. As shown in Table 2, the YOLOv8-Nano model achieves a mAP@50 of 65.4% and a mAP@50:95 of 44.4%, outperforming YOLOv5-Nano (65.3% and 44.3%) and YOLOv3-Tiny (61.3% and 38.8%). Furthermore, the proposed modification to YOLOv8, namely YOLOv8+Best Channel, shows a significant improvement, with mAP@50 of 67.1% and mAP@50–95 of 45.8%.

TABLE II.
PERFORMANCE COMPARISON

Model	GFLOPS	Parameter	mAP 50%	mAP 50-95%
YOLOv12-Nano Turbo	6.0	2.51	61.3	41.6
YOLOv12-Nano	6.5	2.56	60.8	41.9
YOLO11-Nano	6.4	2.59	65.0	43.1
YOLOv10-Nano	8.4	2.70	59.6	39.9
YOLOv9-Tiny	7.8	2.00	61.2	41.7
YOLOv8-Nano	**8.2**	**3.01**	**65.4**	**44.4**
YOLOv8+Best Channel	**7.2**	**1.83**	**67.1**	**45.8**
YOLOv6-Nano	11.9	4.23	61.5	40.9
YOLOv5-Nano	7.2	2.50	65.3	44.3
YOLOv3-Tiny	19.0	12.1	61.3	38.8

These results are higher than those of YOLOv8-Nano. This improvement is mainly due to the channel reduction strategy applied to the backbone and neck, as well as to the SPPF block, which produces a more balanced feature distribution for multi-scale fusion. Thus, this model is not only lighter but also more accurate in detecting dolphins in real-time applications. When compared to YOLOv10-Nano, which achieved the highest mAP@50 of 69.6%, the proposed model shows lower performance in terms of mAP@50:95 (39.9%), indicating less consistent bounding box localization at stricter IoU thresholds. Other models, such as YOLOv11-Nano (mAP@50: 65.0%, mAP@50:95: 43.1%) and YOLOv12-Nano (mAP@50: 60.8%, mAP@50:95: 41.9%), have lower performance

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 3, September 2025, Pp. 2874-2884

than YOLOv8+Best Channel. This shows that simple yet effective modifications to the channel structure can provide an optimal balance between accuracy and computational efficiency. In fact, compared to YOLOv12 Nano-Turbo, which only requires 6.0 GFLOPs, YOLOv8+Best Channel remains superior in accuracy with a +5.8% increase in mAP@50 and a +3.7% increase in mAP@50:95, despite slightly higher computational costs. Overall, these quantitative results show that the channel reduction approach applied to YOLOv8 improves the model's generalization ability, reduces information redundancy, and produces an efficient yet accurate detector for real-time dolphin detection applications.

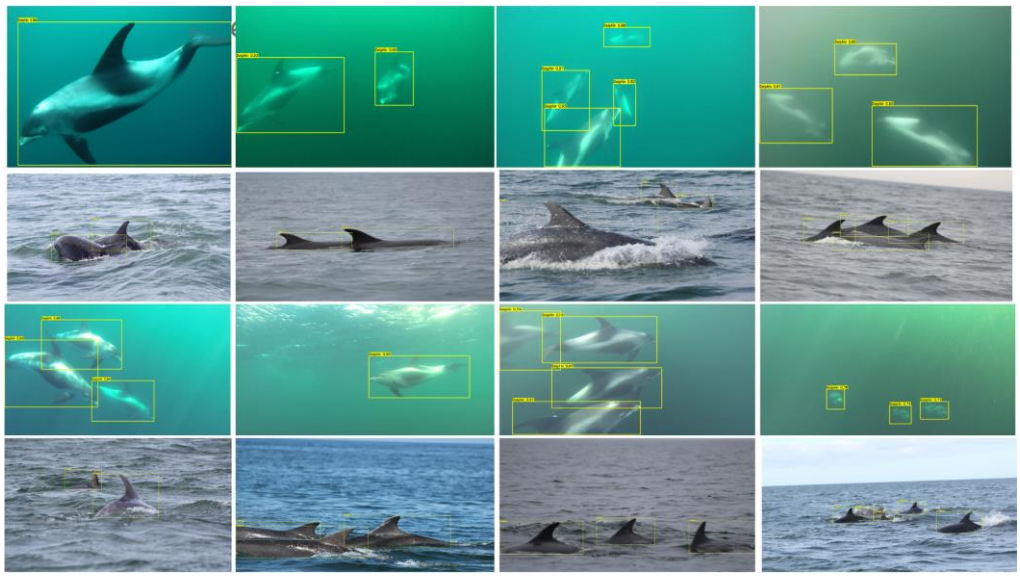

Fig 4. Dolphin detection results. The yellow box shows the predicted bounding box and confidence score.

The detection results can be seen in Figure 4. To further evaluate the dolphin detection performance, we analyzed the bounding box predictions on a test dataset that was completely independent from the training data. In some examples, dolphins were clearly detected even though only part of their bodies were visible above the water surface, or when the objects were at depths with low visibility. These results show that the model is capable of accurately recognizing the location of dolphins. Although the presence of multiple individuals in a single frame adds complexity to the detection process, the proposed system consistently distinguishes dolphins from the background at varying scales and positions.

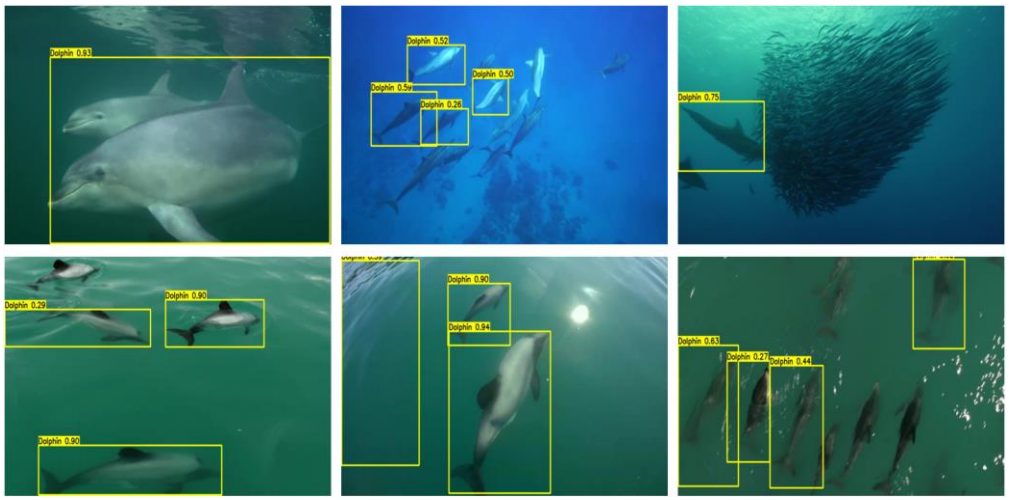

Figure. 5. Detection Error Analysis

Additionally, error analysis to identify model limitations, as illustrated in Figure 5. In some cases, the model had difficulty detecting dolphins that appeared partially, appeared in large groups, or were near schools of fish, which often resulted in false negatives, occlusions, or overlapping bounding boxes. When dolphins were near schools of fish, the similarity in visual patterns and viewing distance frequently caused the system to fail to distinguish the target, resulting in missed detections. False positives were additionally detected, where the model

incorrectly predicted the presence of dolphins in empty areas due to light reflections or water patterns resembling body shapes. These findings confirm that although the modified architecture improves overall precision, challenges in complex underwater conditions remain a limitation that needs to be addressed in future research.

To overcome these limitations, the proposed architecture is designed as a lightweight model with integrated attention modules, allowing the system to focus more effectively on discriminative features even in complex conditions. Although the network remains relatively lightweight, the synergy between efficient feature extractors and the potential of attention modules enables more prominent and selective information processing. These results demonstrate that accurate dolphin detection can still be achieved in real underwater environments, even when image quality is affected by optical disturbances, while also highlighting opportunities for further improvement through exploration of attention mechanisms.

B. Runtime Efficiency

Runtime efficiency is a critical factor in assessing the practical feasibility of object detection models, especially for real-time applications on resource-constrained devices. This metric reflects the balance between computational cost and inference speed, which is typically measured in GFLOPs, latency, and frames per second (FPS). As shown in Table 3, the proposed YOLOv8n+BestChannel model shows a significant improvement over YOLOv8n.

Table III.
LATENCY AND THROUGHPUT OF YOLOV8N

Model	GFLOPS	Latency/ms	FPS
YOLOv8n	8.2	55.9	17.92
YOLOv8+Best Channel	7.2	49.2	**20.38**

With a lower computational complexity of 7.2 GFLOPs compared to 8.2 GFLOPs in YOLOv8n, this model achieves lower latency of 49.2 ms compared to 55.9 ms and higher speed of 20.38 FPS compared to 17.92 FPS. These results show that the modifications made to YOLOv8n+BestChannel in channel adjustment not only improve detection accuracy but also optimize processing speed, making it more suitable for real-time applications while maintaining computational efficiency.

Architecturally, the increase in FPS accompanied by a decrease in GFLOPs and latency in YOLOv8n+BestChannel compared to the original YOLOv8n model can be explained by reducing the number of channels in the backbone and neck to improve computational efficiency. This process reduces the number of floating-point operations (GFLOPS), thereby reducing the computational load without eliminating essential feature representations. By minimizing redundancy in feature information, the data flow during inference becomes more efficient and faster with latency decreasing from 55.9 ms to 49.2 ms. The system processes more frames per second, with FPS increasing from 17.92 to 20.38. These results show that the modifications made to YOLOv8n+BestChannel in channel adjustment not only improve detection accuracy but also optimize processing speed, making this modified model more efficient for real-time deployment on devices with limited computing power resources.

VII. CONCLUSION

This study introduces a lightweight real-time detector for identifying dolphins in both underwater and surface environments. Dolphin habitat conservation encourages the use of low-cost underwater devices to visually monitor behavior. This requires an automatic vision system that can detect dolphins. The modified YOLOv8n+BestChannel architecture achieves this by reducing the number of channels in the backbone and neck. It also simplifies the SPPF block before the neck stage. As a result, the number of parameters is lowered from 3.01 million to 1.83 million, and computational complexity decreases from 8.2 GFLOPs to 7.2 GFLOPs. Experimental results show that the proposed model achieves a performance improvement of 67.1% mAP@50 and 45.8% mAP@50–95. It outperforms YOLOv8-Nano and other lightweight YOLO variants. Additionally, the model demonstrates better runtime efficiency, with a latency of 49.2 ms and a throughput of 20.38 FPS. This makes it more suitable for real-time applications on devices with limited resources.

Overall, this simple yet effective channel reduce strategy improves detection accuracy while reducing computational costs. This study not only proposes an efficient architecture for dolphin detection, but also introduces a new dolphin-specific dataset that can support further research and development. Going forward, future research can focus on integrating attention mechanisms to further improve model generalization in more complex underwater visual conditions.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the AIVISION team for their support, guidance, and expertise in computer vision and deep learning. Their provision of computational resources and continuous assistance has been invaluable in carrying out the experiments and in the preparation of this manuscript.

REFERENCES

[1] 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.
[2] WWF / FiA, The Monitoring of Irrawaddy Dolphin Population in the Mekong River: The Long-Term Population Monitoring based on Mark-Resight Models, Technical Report, Fisheries Administration (FiA) & WWF-Cambodia, 2020. [Online]. Available: https://wwfasia.awsassets.panda.org/downloads/the_2020_mekong_dolphin_population_report_final.pdf
[3] A. F. Chacón, H. Middel, and F. Wickson, Eds., Marine Mammals in the North Atlantic, vol. 12, NAMMCO Scientific Publications, 2022. doi: 10.7557/3.12.
[4] L. Liu *et al.*, 'Deep learning for generic object detection: A survey', *International journal of computer vision*, vol. 128, no. 2, pp. 261–318, 2020.
[5] F. Han, J. Yao, H. Zhu, and C. Wang, 'Marine organism detection and classification from underwater vision based on the deep CNN method', *Mathematical Problems in Engineering*, vol. 2020, no. 1, p. 3937580, 2020.
[6] C. Kim, B.-Y. Kim, and D.-G. Paeng, 'Monitoring of wild and rehabilitating dolphin interactions during rehabilitation period using surveillance technologies', *Scientific Reports*, vol. 15, no. 1, p. 26161, 2025.
[7] L. Cai, N. E. McGuire, R. Hanlon, T. A. Mooney, and Y. Girdhar, 'Semi-supervised visual tracking of marine animals using autonomous underwater vehicles', *International Journal of Computer Vision*, vol. 131, no. 6, pp. 1406–1427, 2023.
[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] H. Leo, F. Arnia, and K. Munadi, 'Fine tuning CNN pre-trained model based on thermal imaging for obesity early detection', *J Rekayasa Elektrika*, vol. 18, pp. 53–60, 2022.
[10] Z. Bao, 'The UAV target detection algorithm based on improved YOLO V8', in Proceedings of the International Conference on Image Processing, Machine Learning and Pattern Recognition, 2024, pp. 264–269.
[11] Z. Zhang, F. Gui, X. Qu, and D. Feng, 'Netting damage detection for marine aquaculture facilities based on improved mask r-cnn', *Journal of Marine Science and Engineering*, vol. 10, no. 7, p. 996, 2022.
[12] L. Ezzeddini *et al.*, 'Analysis of the performance of Faster R-CNN and YOLOv8 in detecting fishing vessels and fishes in real time', *PeerJ Computer Science*, vol. 10, p. e2033, 2024.
[13] T. O. Fossum, Ø. Sture, P. Norgren-Aamot, I. M. Hansen, B. C. Kvisvik, and A. C. Knag, 'Underwater autonomous mapping and characterization of marine debris in urban water bodies', *arXiv preprint arXiv:2208. 00802*, 2022.
[14] 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.
[15] 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.
[16] 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.
[17] 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.
[18] 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.
[19] 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.
[20] 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.
[21] R. De Marco, F. Di Nardo, A. Rongoni, L. Screpanti, and D. Scaradozzi, 'Real-Time Dolphin Whistle Detection on Raspberry Pi Zero 2 W with a TFLite Convolutional Neural Network', *Robotics*, vol. 14, no. 5, p. 67, 2025.
[22] A. P. Colefax, A. J. Walsh, C. R. Purcell, and P. Butcher, 'Utility of spectral filtering to improve the reliability of marine fauna detections from drone-Based monitoring', *Sensors*, vol. 23, no. 22, p. 9193, 2023.
[23] E. Bigal *et al.*, 'Reduction of species identification errors in surveys of marine wildlife abundance utilising unoccupied aerial vehicles (UAVs)', *Remote Sensing*, vol. 14, no. 16, p. 4118, 2022.
[24] M. Mingozzi, F. Salvioli, and F. Serafino, 'X-Band Radar for Cetacean Detection (Focus on Tursiops truncatus) and Preliminary Analysis of Their Behavior', *Remote Sensing*, vol. 12, no. 3, p. 388, 2020.
[25] C. Trotter *et al.*, 'NDD20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation', *arXiv preprint arXiv:2005. 13359*, 2020.
[26] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, 'The pascal visual object classes (voc) challenge', *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.