SINERGI Vol. 30, No. 3, October 2026: 943-960
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2026.3.020

WS-Det: an efficient whale shark detection via integrated feature modulation module

Stephan Adriansyah Hulukati[1]*, Muhamad Dwisnanto Putro[2], Irvan Abraham Salihi[1], Hebron Prasetya[2], Imanuel Kutika[2], Hoang Van Thanh[3]
[1]Informatics Engineering Study Program, Faculty of Computer Science, Ichsan University, Indonesia
[2]Magister Program of Informatics, Postgraduate Program, Sam Ratulangi University, Indonesia
[3]Faculty of Information Technology, Quang Binh University, Vietnam

Abstract
Whale sharks are vital plankton feeders that help maintain marine ecosystem stability and serve as key ecotourism assets supporting Gorontalo's blue economy. Reliable detection and identification systems are essential for monitoring their habitats and behaviors. Vision-based methods offer non-invasive observation without physical contact or electromagnetic interference. YOLOv12, a state-of-the-art object detector, achieves high localization accuracy with its lightweight "nano" variant. However, lightweight models often struggle with complex feature discrimination, which can be mitigated by integrating enhancement blocks to boost detection performance while maintaining computational efficiency. In this paper, we introduce a novel vision-based whale shark detection framework (WS-Det), which enhances YOLOv12-nano with MOGA attention modules. The proposed model incorporates lightweight convolutional operations to ensure efficient feature extraction while keeping computational cost low. The Integrated Feature Modulation (IFM) module improves backbone performance by selectively emphasizing spatially salient regions while interacting with multi-spatial-frequency information, thereby improving the overall robustness of feature learning. To support this study, we also present a dedicated whale shark dataset comprising underwater and aerial (drone-based) images for direct and practical monitoring. Experimental evaluations demonstrate that the WS-Det outperforms the baseline YOLOv12-nano and other efficient detectors, achieving higher detection precision. On the proposed dataset, the model achieves an mAP@0.5 of 0.979 and an mAP@0.5:0.95 of 0.868. It contains 2,191,459 parameters and requires 6.3 GFLOPs, indicating a low computational cost. Inference speed evaluations show the model runs at 17.19 FPS on a desktop PC and 6.64 FPS on a Raspberry Pi 5, confirming its suitability for deployment on resource-constrained devices.

Keywords:
Deep learning;
Enhancement;
Object detection;
Optimal model;
Whale shark;

Article History:
Received: September 27, 2025
Revised: May 1, 2026
Accepted: June 12, 2026
Published: October 2, 2026

Corresponding Author:
Stephan A. Hulukati
Informatics Engineering Study Program, Faculty of Computer Science, Ichsan University, Indonesia.
Email:
stephanhulukati17@gmail.com

INTRODUCTION

Whale sharks (Rhincodon typus) play a vital role in maintaining marine ecosystem balance and serve as a major ecotourism attraction in the waters of Gorontalo [1]. Conventional monitoring using electromagnetic tags risks behavioral disruption, population decline, and predator attraction [2]. To support conservation and tourism, automated vision-based systems using deep learning offer a non-invasive alternative for accurate detection and individual identification. However, underwater imaging is often affected by inconsistent lighting, varying depths, and water turbidity. These conditions scatter and absorb

light, reducing contrast and obscuring details. Overcoming these challenges ensures reliable and sustainable marine monitoring.

Several works have utilized vision-based algorithms to identify endemic marine species and support habitat conservation efforts, including the recognition of sea turtles [3][4], whale detection [5, 6, 7], shark monitoring [8, 9, 10], and other works on underwater object detection [11, 12, 13, 14, 15]. Developments in CNNs have driven recent advances in computer vision (Convolutional Neural Network)-based object detection, particularly YOLO (You Only Look Once), which offers high detection speed and competitive accuracy [16]. This architecture has evolved into more advanced YOLO architectures that incorporate global attention mechanisms to improve feature extraction and detection performance [17]. However, most YOLO-based models are designed primarily for natural image datasets and often degrade in underwater environments due to low illumination, image distortion, motion blur, and scattering effects [18][19]. Although recent approaches attempt to address these issues through attention mechanisms, these enhancements often add computational overhead and are not specifically optimized for underwater feature degradation. Therefore, more efficient, adaptive feature-modulation strategies are still needed to improve detection robustness in challenging marine environments.

Recent studies have used YOLO-based detectors to monitor sharks and whales. A modified YOLOv3-t model using satellite and drone imagery achieved a mAP of 0.74 for gray whale detection [6]. However, this approach is restricted to surface imaging and lacks a comprehensive real-time performance evaluation. To further improve detection performance, some studies have incorporated image enhancement techniques. In addition, YOLOv5 combined with Vision Transformer BiFormer achieved 88.1% accuracy in whale-related tasks [7], while another study integrated YOLOv5 with ESRGAN to enhance image quality for shark detection [10].

In contrast, earlier approaches relied on conventional CNN architectures. A study [8] proposed a white shark surveillance system using CNN models, while another employed VGG-UNet and VGG16 for shark recognition [9]. However, these methods suffer from high computational cost and limited inference speed. Although these works improve detection accuracy [6, 7, 8, 9, 10], their reliance on input-level enhancement adds computational overhead, which may limit robustness under varying underwater conditions.

Overall, existing approaches either suffer from high computational complexity, limited adaptability to underwater environments, or reliance on pre-processing techniques, highlighting the need for a more efficient and adaptive feature-level enhancement strategy for robust marine object detection. Table 1 summarizes recent state-of-the-art methods for marine animal detection and recognition. Despite prior advancements, several critical gaps remain:

- Existing models show limited robustness under underwater conditions such as low illumination, blur, and distortion.
- Many approaches introduce high computational cost, limiting real-time and edge deployment.
- Several methods rely on image enhancement techniques, increasing complexity and reducing generalization.
- Existing attention mechanisms are not specifically optimized for underwater feature degradation.
- Current approaches lack an effective balance between detection accuracy and computational efficiency.

In addition, various studies show that attention modules in CNNs improve underwater object detection under varying illumination [19], [20].

Table 1. State-of-the-Art positioning of representative related works

References	Methods	Results	Limitations
J. Liu et al. [8]	SharkNet	Accuracy of 93.6%	Limited to shark classification without localization
Nhat Anh Le et al. [9]	VGG-Unet	Accuracy of 81%	Traditional CNN architecture and is limited to shark recognition without localization
K. M. Green et al. [6]	Modified YOLOv3t	mAP of 74%	Based on early YOLO and limited to shark detection without evaluating inference speed.
Ruolan Zhang et al. [7]	Enhanced YOLOv5 with Vision Transformer BiFormer	Accuracy of 88.1%	Limited to whale classification tasks without detection or real-time evaluation
WS-Det	**Modified YOLOv12n with IFM module**	**mAP50:95 of 86.8%**	**Limited to real-world underwater deployment**

Transformer self-attention captures global context better than channel or spatial attention. However, it also generates high computational cost, limiting real-time use. To address this, Multi-Order Gated Aggregation (MOGA) [21] offers an efficient alternative to transformer self-attention. In this work, we further modify MOGA to enhance feature extraction. The proposed Integrated Feature Modulation (IFM) module builds on a modified MOGA design and enhances feature extraction by leveraging multi-scale receptive fields.

Unlike conventional attention modules such as SE, CBAM, ECA, and STAR, the IFM module captures global contextual information by explicitly modeling interactions across multiple feature scales. SE, CBAM, and ECA primarily focus on channel-wise and spatial attention with limited cross-scale interaction, while the STAR block emphasizes spatial feature connectivity without integrating multi-scale relationships. IFM enhances feature representation using deeper convolutional layers with multi-scale kernels of 7 × 7, 5 × 5, and 3 × 3. It also incorporates standard and bottleneck convolutions to aggregate local and global features effectively and improve feature extraction. This novel attention module introduces a new structure that combines multi-scale convolutional kernels with feature modulation mechanisms to enhance feature discrimination.

We incorporate these improvements into the proposed WS-Det framework, built on a modified YOLOv12-nano architecture [18], to balance detection accuracy and computational efficiency. Although researchers have widely studied object detection for marine species, a clear gap remains in using recent YOLO architectures for whale shark detection. Most existing studies rely on earlier YOLO versions or incorporate computationally expensive enhancement and attention mechanisms, which limit real-time applicability on resource-constrained devices. To address this gap, this study proposes the WS-Det framework for accurate, real-time whale shark monitoring [22], built on a modified YOLOv12-nano architecture [17]. Furthermore, achieving an effective model requires a robust whale shark dataset that includes both underwater and aerial drone perspectives to ensure comprehensive training and evaluation. In addition, labeled whale shark datasets from both underwater and surface environments remain limited. The need for reliable localization models further motivates developing a new labeled dataset. This dataset supports vision-based detection and monitoring systems.

In this work, we design a vision-based whale shark detection model using an improved YOLOv12-nano architecture for real-world applications. The proposed model incorporates an enhancement module that identifies key features and suppresses irrelevant information, improving efficiency and precision to support conservation objectives. The main contribution of this work is as follows:

- We propose a novel marine whale shark detection model, WS-Det, to localize whale shark presence based on a modified YOLOv12-nano architecture, achieving superior accuracy and computational efficiency.
- We introduce a novel whale shark detection dataset with precise bounding box annotations for a single object class, formatted for compatibility with the YOLO framework.
- The Fast Convolutional (FC) Block accelerates feature extraction by processing partial channels, balancing computational efficiency with improved detection accuracy.
- We introduce an Integrated Feature Modulation (IFM) module to enhance feature extraction performance while maintaining computational efficiency. This block captures multi-order convolutional relationships by integrating normalization and activation functions, enabling more effective and discriminative feature representations.
- We conduct a comprehensive performance analysis, runtime efficiency evaluation, and ablation study on the proposed architecture and compare it with recent lightweight and efficient detection models.

METHOD

In this study, we use a CNN as the base algorithm for feature extraction. This approach captures hierarchical information by applying learnable filters to local regions, allowing the model to identify spatial patterns and progressively learn higher-level semantic representations for object localization and classification. The CNN employs two-dimensional kernels to distinguish object features from the background. The overall process is as follows.

$$S(i,j) = (I * K)(i,j). \tag{1}$$

The CNN exploits local spatial information, which is effective because object components in an image are typically close together. The output feature map, denoted as $(S(i,j))$, is computed from the input image (I) by applying a weighted kernel (K) using a sliding window technique.

The WS-Det detector emphasizes feature-extractor performance to distinguish whale shark objects accurately. It integrates several compact and efficient modules, as illustrated in Figure 1. The overall architecture comprises three main components: the backbone, neck, and head layers, each responsible for progressively extracting, aggregating, and refining features for object detection. In addition, the WS-Det network incorporates a channel reduction strategy to minimize parameters and computational cost, improving efficiency while maintaining detection accuracy.

Backbone

The backbone serves as the core feature-extraction component, learning visual patterns from images through convolutional operations. Its architecture is optimized for high efficiency and fast inference, enhanced by an attention mechanism that captures global contextual information during feature extraction. The backbone integrates the C3K2 module, a faster version of the Cross Stage Partial (CSP) bottleneck that uses only two convolutional layers. This design enhances computational efficiency while preserving effective feature extraction.

Like the C2f structure, the C3K2 module uses residual connections within the bottleneck to enable smoother gradient flow and improve training stability. This design minimizes parameter count and computational cost while preserving high detection accuracy.

C3k2

The C3K2 block is a key component of the backbone architecture, designed to balance efficiency and flexibility in feature extraction. This block improves computational efficiency by utilizing only two main convolutional layers.

$$C3K2 = Conv_{1\times1}([Bt^n(V_1), V_2\,]). \quad (2)$$

The input features are first processed with a 1 × 1 convolution to adjust the channel dimensions, then split into two parallel branches, V1 and V_2, for subsequent processing. The first branch (V_1) applies a series of bottleneck (Bt^n) layers with a 3 × 3 kernel [23], while the second (V_2) serves as a bypass path. The outputs from both branches are concatenated ([.]) and projected back with a 1 × 1 convolution ($Conv_{1\times1}$), balancing accuracy, speed, and memory efficiency.

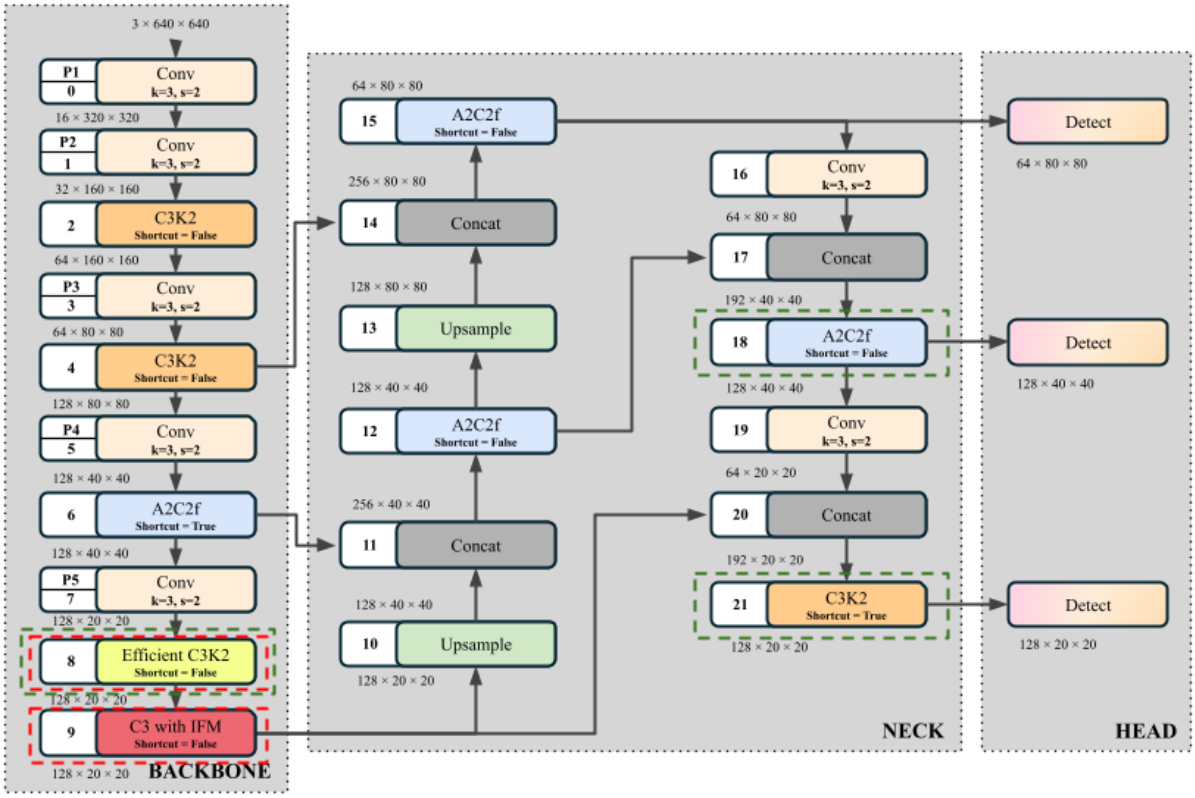

Figure 1. The proposed Whale Shark Detector is based on a modified YOLOv12 architecture integrated with enhancement modules to improve multi-scale feature extraction and detection accuracy. Red dashed boxes indicate the proposed modules, while green dashed boxes denote channel reduction layers derived from the original YOLOv12n architecture

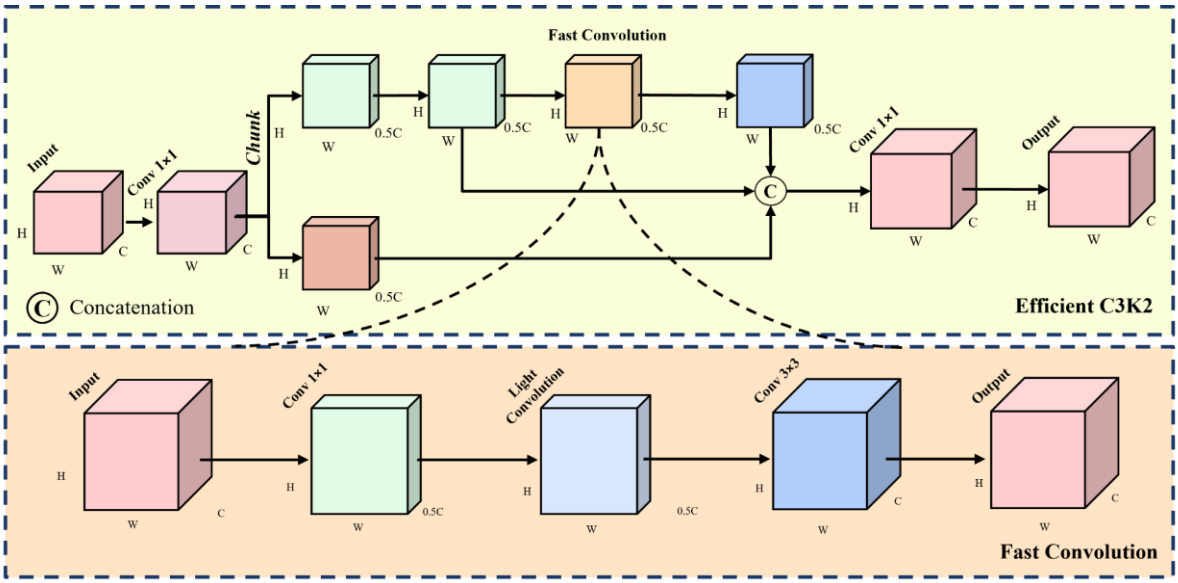

Figure 2. The proposed efficient module integrates the C3K2 module and a Fast Convolutional Block to reduce computational complexity while maintaining feature extraction performance

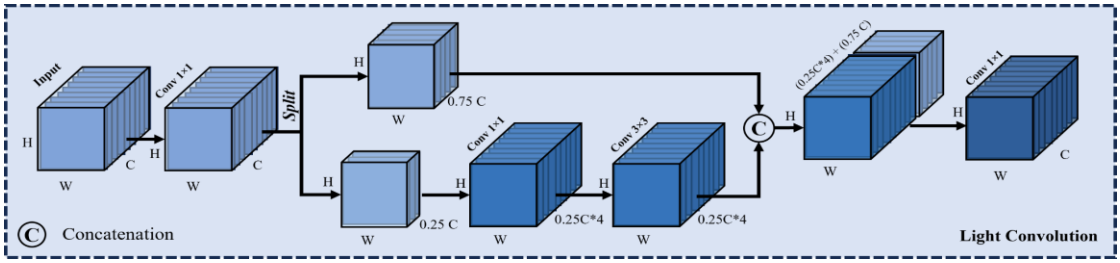

Figure 3. The Light Convolution module enhances efficiency by processing a reduced subset of features, thereby lowering computational complexity while maintaining performance

Similarly, the C2f block adopts a dual-branch structure, where one path undergoes bottleneck transformations and the other bypasses processing. This approach effectively reduces redundant computations and stabilizes the gradient flow during training. The C3k2 module offers two configurable options, allowing the bottleneck component to be replaced with a C3k module. Building upon this concept, the C3k2 block enhances the C2f structure by embedding either a C3k or a standard bottleneck module. This design improves spatial feature representation and robustness, particularly for complex object detection tasks on resource-constrained platforms.

A2C2f

The A2C2f block is an enhanced version of the C2f architecture that integrates Area Attention through either an ABlock or a C3k module. This approach can be applied selectively to balance spatial feature discrimination and computational efficiency.

$$A2C2f = Conv_{1\times1}([Ablock^{n}(z), z]). \quad (3)$$

The process begins with a 1×1 convolution to reduce the channel size by 50%. Subsequently, the reduced feature map (z) is processed by the two ABlocks, which partition it into multiple regions and compute inter-area relationships to highlight critical spatial dependencies [24]. The resulting features are then concatenated with the original input (z) and refined using a 1 × 1 convolution ($Conv_{1\times1}$) for more focused feature extraction.

Neck

This layer bridges the backbone and the head, aggregating multi-scale features [23]. The YOLOv12 neck structure retains the Path Aggregation Network (PAN) to unify information across multiple feature levels, providing richer spatial and semantic context. Furthermore, flexible upsampling and downsampling operations allow the network to adapt to varying input resolutions, ensuring stable performance across scales. This improved multi-scale integration

enables consistent detection of objects of different sizes.

Head

The head of YOLOv12 converts the feature maps produced by the neck into final predictions, including bounding box coordinates and classification scores [23]. It features an enhanced prediction path with a larger receptive field, allowing the model to capture contextual cues more effectively. Furthermore, adopting non-linear activation functions such as SiLU improves the network's ability to model complex feature patterns. To evaluate detection performance, the model employs CIoU, DFL, and binary cross-entropy losses [23]. The loss weights are set to 0.05 for bounding box regression, 0.5 for classification, and 1.0 for objectness.

Fast Convolutional Block

Extracting information often demands convolutional operations with substantial parameters and high computational cost. This requirement slows inference speed and restricts deployment on resource-limited edge devices. Moreover, conventional convolutions may introduce redundant features, further reducing model efficiency. In this work, we overcome these limitations with lightweight convolution techniques that reduce parameter count and computational complexity; Figure 3 shows the architecture. This convolution divides the channels into two parts and applies sequential convolution to only one segment. The proposed module is integrated into the C3K2 module to enhance network efficiency, as shown in Figure 2. This efficient C3K2 approach reduces neuron utilization while maintaining overall performance. The process is defined as follows:

$$(X_C^1, X_C^2) = X_C, \tag{4}$$

where

$$X_C = C_{1\times1}(x). \tag{5}$$

To maintain information flow across all feature maps, a 1 × 1 convolution $(C_{1\times1})$ is applied to the input feature X, followed by batch normalization and the SiLU activation function. This operation reconstructs the input feature information while enabling efficient sharing across all feature maps. Furthermore, a split operation divides the features (X_C) into two segments, X_C^1 and X_C^2. In this process, X_C^1 receives 25% of the features, while X_C^2 takes the remaining 75%. One partition is then processed using two sequential convolution operations with different kernel sizes, as described below.

$$P_C^1 = Conv_{3\times3}(Conv_{1\times1}(X_C^1)). \tag{6}$$

This operation enables larger receptive fields without significantly increasing computational cost. To enrich the X_C^1 feature representation, a pointwise 1 × 1 convolution $(Conv_{1\times1})$ expands the channels to four times the size of the original X_C^1 feature maps. Generally, a 1 × 1 convolution is applied in the bottleneck to enable information interaction across feature map channels.

However, it focuses solely on channel-wise interactions and fails to capture spatial context. To address this limitation, a larger 3 × 3 convolution kernel $(Conv_{3\times3})$ captures local neighborhood relationships and enhances spatial feature representation. This operation improves spatial feature extraction. Afterward, a concatenation process $(Concat)$ combines the processed P_C^1 features with the bypass branch X_C^2. The overall process is described as follows:

$$LC = C_{1\times1}(Concat[P_C^1, X_C^2]). \tag{7}$$

To mitigate feature mixing, a 1 × 1 convolution $(C_{1\times1})$ is applied with batch normalization and SiLU activation. This operation ensures the output channel count matches the input channel dimension, while batch normalization stabilizes the data distribution and prevents anomalies. The Fast Convolutional Block uses two convolutions at the beginning and end of the block to optimize the lightweight convolutional structure, using 1 × 1 and 3 × 3 filters, respectively. The initial 1 × 1 convolution adjusts the channel dimensionality, while the final 3 × 3 convolution enhances spatial feature interactions by effectively integrating channel information.

Integrated Feature Modulation Module

The attention module plays an important role in improving detection performance. Its ability to refine and enhance relevant information helps the model focus on important patterns. This process can be described as follows.

$$g(x) = f(x) * x. \tag{8}$$

In general, an attention module consists of two branches to process the input information. The function $(f(x))$ represents a mapping applied to the input feature to generate attention maps, which are used as scaling factors for the input (x) through a multiplication operation $(*)$. This

process refines the feature representation and produces an output feature map ($g(x)$).

High-level features provide crucial information for object recognition but often lack sufficient contextual relationships across spatial regions. In this work, we introduce the Integrated Feature Modulation (IFM) module to strengthen feature interaction and enhance contextual representation within the network. The proposed module differs from conventional attention mechanisms such as SE, CBAM, ECA, and self-attention, which primarily reweight feature maps. In contrast, the proposed IFM not only refines features but also acts as an additional feature extractor, capturing multi-frequency information to discriminate irrelevant components in high-level representations better. This dual functionality enables more effective feature modulation beyond simple attention weighting, clearly distinguishing it from existing methods.

As shown in Figure 4, the IFM module introduces a new structure with enhancement components designed to capture contextual variations crucial for recognizing distinctive patterns, particularly the unique skin textures of whale sharks. It employs different convolution types with diverse receptive fields to capture broad spatial information effectively. A learnable parameter refines features and emphasizes key information, and a bottleneck structure reduces computation while retaining essential details. Next, a Squeeze-and-Excitation (SE) mechanism optimizes channel selection by recalibrating feature importance. Since high-level features are prone to information degradation in deeper layers, a residual connection is integrated to maintain stability and ensure effective gradient flow. This IFM process can be described as follows:

$$IFM = XC_4(XC_3(XC_2(XC_1(x)))). \quad (9)$$

The IFM computes the variance across input feature maps (x) to recognize and emphasize the most informative and discriminative features at the first stage. This process begins with Batch Normalization ($BatchNorm$), which stabilizes the data distribution across feature maps within every batch. To calculate variance, Global Average Pooling (GAP) derives the mean values, transforming the tensor into a vector representation that captures global spatial relationships. The block can be described as follows:

$$XC_1 = x - GAP(BatchNorm(x)). \quad (10)$$

The mean feature maps are then subtracted element-wise from the original feature maps, allowing the module to highlight regions with higher variance and richer contextual information essential for accurate whale shark recognition. In addition, the subsequent process scales the feature maps using a learnable parameter (γ_1). This mechanism enables the model to learn and select feature maps that contain more relevant information. This process can be described as follows:

$$XC_2 = x + (\gamma_1 * XC_1). \quad (11)$$

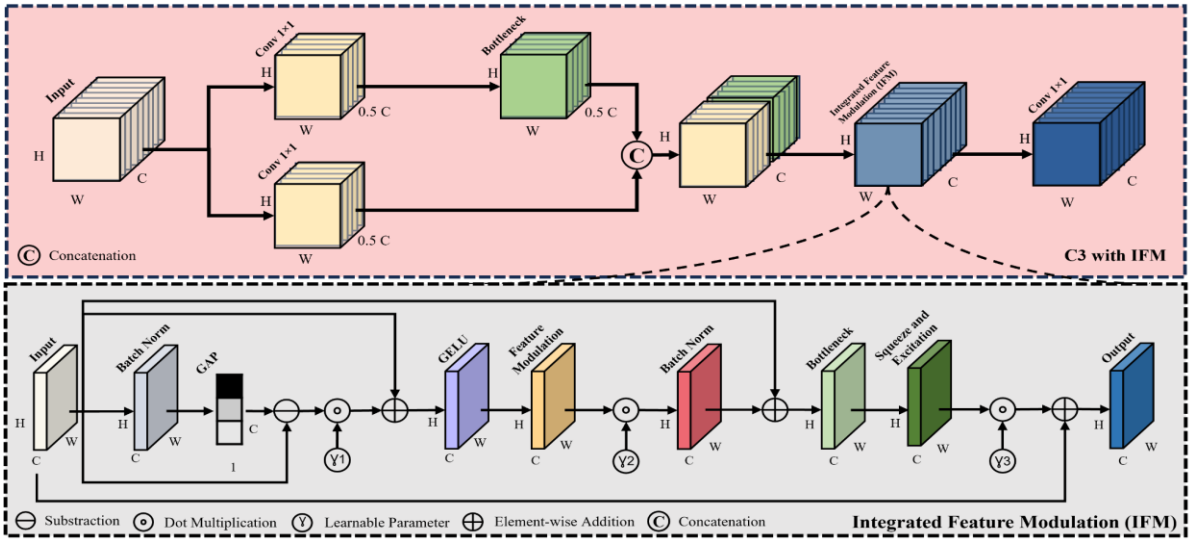

Figure 4. The Integrated Feature Modulation (IFM) module enhances model performance by combining efficient feature extraction with modulated features (bottom). The proposed module is integrated into the C3 block (top)

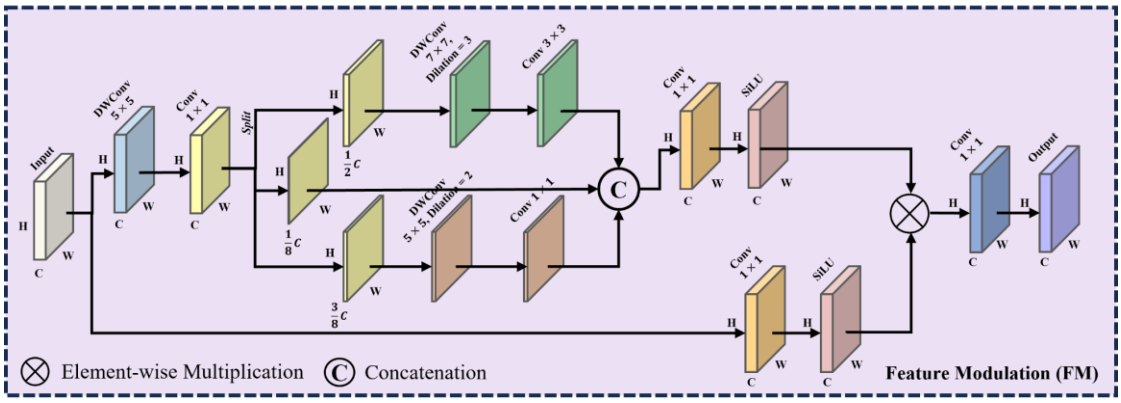

Figure 5. The Feature Modulation (FM) module enhances feature representation by employing multi-order depthwise convolutions, effectively capturing diverse spatial information with improved efficiency

The residual weighting operation is essential for the model to recognize object patterns. These methods can serve as a selective approach to the channel representation. Moreover, the feature maps incorporate the GELU activation. This activation enables non-linear representation and removes large negative values from the feature maps that indicate uninformative information. Furthermore, the Feature Modulation (FM) module with a learnable parameter (γ_2) is incorporated to capture richer and more diverse feature representations by modeling broader contextual information. This overall process can be described as follows:

$$XC_3 = x + (\gamma_2 * FM(GELU(XC_2))). \quad (12)$$

Moreover, attention further enhances critical information before it is passed to the neck stage. This process employs the SE module [19] to recalibrate channel responses and selectively emphasize the most informative components within the feature maps, as illustrated in Figure 6. This process can be defined as follows:

$$XC_4 = x + (\gamma_3 * SE(BT(BatchNorm(XC_3)))). \quad (13)$$

Furthermore, a bottleneck module (BT) with a 3 × 3 convolution kernel is incorporated to reconstruct feature representations and facilitate interaction among feature channels. This structure enables the SE module to more effectively identify channels containing the most relevant object information. Combined with Batch Normalization (BN), it emphasizes the most informative feature maps while suppressing less significant ones, thereby improving feature discrimination.

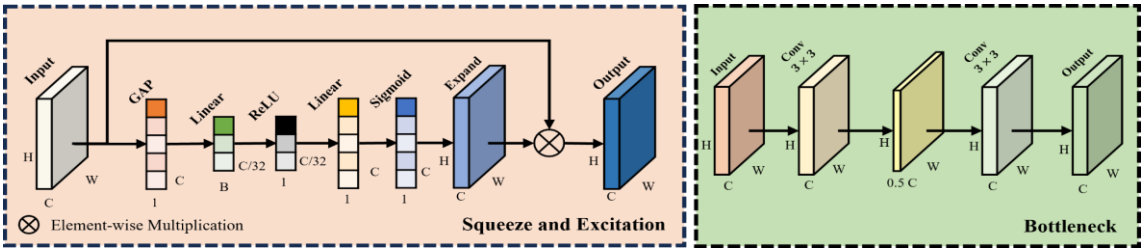

Figure 6. The Squeeze-and-Excitation (SE) module (left) and bottleneck module (right) enhance feature representation

Figure 7. The proposed dataset shows (a) drone images and (b) underwater scenes

Feature Modulation Module

The Feature Modulation (FM) component enhances information diversity in high-level features by integrating multiple convolutional mechanisms that extract features at different scales. Depthwise convolution processes the input feature (x), reducing computational cost and parameter usage while enabling efficient use of larger kernels. This module also splits the input to process three separate convolutional segments, each capturing complementary aspects of the feature maps. Large kernels further strengthen the model's ability to capture contextual relationships, helping it identify the most informative features. The overall process can be described as follows:

$$(CF_1, CF_2, CF_3) = CV_{1\times1}(DW_{5\times5}(x)). \quad (14)$$

Since large kernels require more parameters, we employ a split operation to reduce computational cost. In the first stage, the FM module applies a depthwise convolution with a 5 × 5 kernel ($DW_{5\times5}$) and a simple convolution ($CV_{1\times1}$) to enhance spatial and channel interaction. The resulting feature maps are then divided into three segments, where CF_1 contains 12.5 percent of the total channels, while CF_2 and CF_3 contain 37.5 and 50 percent, respectively. As shown in Figure 5, each segment is applied to a different convolutional process.

The CF_1 map serves as the direct identity value, where no operation is applied, while the CF_2 map is processed through a depthwise convolution with a 5 × 5 kernel and a dilation rate of two. This configuration allows the model to capture a wider receptive field and extract broader contextual information from the feature maps. Subsequently, a 1 × 1 convolution is applied to facilitate interaction and information exchange across channels. This process can be described as follows:

$$CP_2 = CV_{1\times1}(DW_{5\times5}(CF_2)). \quad (15)$$

$$BP = SiLU(CV_{1\times1}(x)). \quad (16)$$

To update the original information with the extracted processes, we apply a Hadamard operation. However, before the recalibration stage, the input feature map is filtered using a 1 × 1 convolution followed by a SiLU activation. Finally, at the end of the Feature Modulation process, pointwise convolution reconstructs the refined feature map.

Table 2. Training and Testing Configuration

Parameters	Setup
Device	AMD Ryzen 5 3500 6-Core
GPU	NVIDIA RTX 12 GB
Image Size	640 x 640 pixels
Epoch	150
Batch Size	16
Optimizer	Stochastic Gradient Descent (SGD)
Learning Rate	0.01
Momentum	0.937
Weight decay	0.0005
Learning Rate Scheduling	lr_0= 0.01, lr_f = 0.01

Table 3. Inference Configuration

Parameters	Setup
Operating System	Ubuntu
Compiler	Python 3.9.20
Network Construction Method	Pytorch 2.0
CPU	AMD Ryzen 5 3500

Implementation Setup

The training process plays an important role in achieving stable convergence and preventing overfitting. Based on Table 2, the model is trained for 150 epochs, which is the most effective point for stable convergence while preventing overtraining and performance decline. Furthermore, a batch size of 16 is selected to balance computational efficiency, training stability, and effective memory utilization.

The input resolution is set to the default YOLO configuration of 640 × 640 pixels to maintain sufficient spatial detail and enable accurate object localization throughout training. Additionally, the Stochastic Gradient Descent (SGD) optimizer is used with an initial learning rate of 0.01 to ensure stable, consistent parameter updates.

The learning rate plays a critical role in stabilizing optimization. In this study, the initial learning rate is set to lr_0 = 0.01 and lr_f = 0.01, while the final learning rate is defined as a fraction of the initial value. A cosine annealing scheduler can be activated, which provides a smoother decay of the learning rate:

$$lr(t) = lr_f + 0.5(lr_0 - lr_f)(1 + cos\,(\tfrac{t}{T}\pi)), \quad (17)$$

where t is the current training step, and T is the total number of training steps.

This configuration provides an efficient and well-regularized training process that promotes stable learning and improves the model's generalization performance. This training process runs on an AMD Ryzen 5 3500 six-core processor

with an NVIDIA RTX Colorful GPU accelerator equipped with 12 GB of memory.

This hardware configuration provides ample computational capacity to support intensive deep learning workloads. Integrating a multi-core CPU with a high-performance GPU accelerator enables parallel processing, reduces training time, and ensures stable performance during both training and testing. The inference setup reflects real-world deployment conditions, especially on low-end computing systems. As shown in Table 3, we evaluate the model on a CPU to assess performance without GPU acceleration. The system runs on a CPU device: an AMD Ryzen 5 3500 six-core processor running Ubuntu. The implementation is developed in Python 3.9.20 using the PyTorch 2.0 framework for network construction. The model operates on VGA-resolution inputs to ensure a fair comparison. This setup enables a practical evaluation of computational efficiency, inference speed, and suitability for deployment on devices with limited hardware resources.

During training, data augmentation is applied to the proposed dataset using geometric transformations, including scaling, translation, flipping, and perspective changes, to simulate spatial variability. In addition, photometric adjustments in the HSV color space handle illumination differences. Advanced strategies such as Mosaic augmentation combine multiple images into a single sample to improve small-object detection and contextual diversity. In addition, MixUp and Copy-Paste are applied to the mosaic images to further enrich the data distribution by blending images or inserting object instances into new backgrounds. Collectively, these augmentation techniques increase data variability, mitigate overfitting, and improve detection performance in complex real-world environments.

Proposed Dataset

The dataset serves as an important source for deep learning models to learn data patterns and characteristics. Whale sharks can be distinguished from their surroundings based on features such as body shape, texture, and color. They typically have elongated, moderately wide bodies with distinctive spots and skin patterns, with colors ranging from black and gray to dark blue. Gorontalo Bay in Botubarani Village was selected as the observation site because whale sharks frequently visit these waters, with five to seven individuals often observed. This enabled periodic data collection through camera recordings from aerial and underwater perspectives. Aerial videos were captured using a DJI Mini 4 Pro drone with a 4K HDR camera capable of recording up to 60 frames per second (FPS), while underwater videos were recorded using a GoPro HERO camera at Full HD resolution.

The footage was collected over three days, from morning to afternoon, under predominantly sunny and partly cloudy conditions. These varying weather conditions introduced natural lighting variations within the proposed dataset. Light reflections on the water surface introduced illumination distortions, thereby reducing the visibility and clarity of object features. Moreover, each data collection session involved different recording distances. The distance between the drone and the whale shark ranged from approximately 20 to 100 meters. Underwater footage was captured at closer ranges of 5 to 15 meters because limited visibility, caused by water quality and suspended particles, contributed to image blurriness. In total, we collected 1,090 videos, including 559 aerial and 531 underwater recordings. We converted the videos into image frames at fixed intervals to ensure consistent sampling and capture variability in whale shark appearance, position, and behavior. This produced 56,863 images: 28,884 drone images and 27,979 underwater images. Figure 7 shows examples of aerial and underwater images.

We also annotated all images to generate labels that include the object class and coordinate locations. We used a web-based annotation tool [24] to draw bounding boxes around each whale shark while excluding background regions as irrelevant information. The process involved 15 annotators assigning object locations, followed by cross-validation to ensure consistency. Each bounding box was reviewed to verify accurate alignment with the object boundaries. To facilitate data processing, the location and category information were saved in text files compatible with the YOLO format.

During data preparation, we divided all images and annotations into three subsets: training, validation, and testing. We performed the split manually to prevent frame overlap between the training and testing sets. In total, 56,863 images were used: 42,840 for training, 8,351 for testing, and 5,672 for validation. Each subset includes both surface and underwater scenes, enabling the model to recognize whale sharks under varying conditions. We did not apply data augmentation to the original proposed dataset. However, the training process followed the YOLO framework [23], which employs mosaic augmentation with color and brightness distortion as well as geometric transformations.

RESULTS

Model Analysis

The proposed modification improves performance by integrating several modules designed to enhance feature extraction and detection accuracy. We conduct an ablation study through multiple experiments to analyze each module's contribution. As shown in Table 4, we compare model performance using the mAP (mean Average Precision) metric, while evaluating efficiency based on the number of parameters and computational complexity (GFLOPs). All experiments are performed using an NVIDIA RTX 3060 Colorful GPU to ensure consistent training stability.

This ablation study systematically evaluates the progressive integration of the proposed modules and demonstrates how each module is incorporated into the baseline network. First, we apply channel reduction (CSR) to YOLOv12n across several selected layers. Second, we introduce an FC block at the 8th layer to replace the bottleneck component in the C3K2 module. Finally, we integrate an IFM block into the 9th layer and combine it with the C3 module. The baseline YOLOv12n achieves an mAP50:95 of 0.861. Additionally, the original network requires more computational resources, with 2.57 million parameters and 6.5 GFLOPs.

Next, the channel tuning approach (CRS) reduces the number of neurons in each convolution layer. This reduction decreases the total number of parameters and lowers the overall computational cost. As shown in Table 4, this method effectively reduces the number of parameters and computational cost to approximately 1.47 million and 5.7 GFLOPs, respectively.

Despite the reduction, the channel-tuning modification maintains comparable detection performance (0.862 mAP50:95), demonstrating its efficiency and robustness. Furthermore, integrating the Fast Convolutional (FC) module improves efficiency while maintaining detection accuracy. This modification reduces the number of parameters to 1.43 million, representing a 2.72% decrease compared to the channel tuning approach. In addition, the proposed model achieves the highest detection performance, with an mAP50:95 of 0.868, a 0.6% improvement over YOLOv12n.

Evaluation on Dataset

Evaluating model performance is crucial to highlighting the differences between the baseline model and the proposed network. As shown in Table 5, this comparison includes several YOLO variants to evaluate detection accuracy and parameter efficiency. We evaluated the model on the test and validation datasets to objectively assess the model's generalization capabilities. The baseline YOLOv12n model achieved a mAP@50 of 0.976 and a mAP@50:95 of 0.861, with a total of 2,568,243 parameters. In contrast, the proposed model achieves better detection performance, with mAP@50 of 0.979 and mAP@50:95 of 0.868, while reducing the total number of parameters to 2,191,459. This improvement shows that WS-Det provides better feature discrimination and generalization while maintaining a more efficient architecture than the baseline.

This evaluation assesses performance consistency between the validation and testing datasets using an Intersection over Union (IoU) threshold of 0.5. The YOLOv11n model achieves mAP@50 of 0.971 and mAP@50:95 of 0.857 with 2.59 million parameters, demonstrating a strong balance between accuracy and efficiency. The YOLOv10n model achieves mAP@50 of 0.972 and mAP@50:95 of 0.855 with 2.71 million parameters, showing slightly higher computational cost with minimal performance improvement.

Additionally, the YOLOv9t model achieves a mAP@50 of 0.974 and a mAP@50:95 of 0.865 with 2.00 million parameters, demonstrating efficient learning capabilities. Meanwhile, the YOLOv8n model recorded a mAP@50 of 0.972 and a mAP@50:95 of 0.861 with 3.01 million parameters, showing strong detection accuracy but a relatively higher parameter count.

Table 4. Ablation study comparing the performance and efficiency of the proposed model

Experiment	Model	mAP 50	mAP 50:95	Parameter	GFLOPS
1	YOLOv12n	0.976	0.861	2,568,243	6.5
2	YOLOv12n+CRS	0.976	0.862	1,466,339	5.7
3	YOLOv12n+CRS+FC	0.977	0.862	1,434,083	5.7
4	YOLOv12n+CRS+FC+IFM	0.979	0.868	2,191,459	6.3

Table 5. Model evaluation on the dataset compared with efficient architectures and recent attention modules

Model	Parameter	Validation		Testing	
		mAP 50	mAP 50:95	mAP 50	mAP 50:95
YOLOv11n	2,590,035	0.970	0.859	0.971	0.857
YOLOv10n	2,707,430	0.972	0.860	0.972	0.855
YOLOv9t	2,005,603	0.981	0.874	0.974	0.865
YOLOv8n	3,011,027	0.977	0.866	0.972	0.861
YOLOv5n	2,508,659	0.976	0.863	0.972	0.856
YOLOv3t	12,132,642	0.973	0.818	0.969	0.811
YOLOv12s	5,754,036	0.973	0.871	0.976	0.870
YOLOv12n	2,568,243	0.976	0.864	0.976	0.861
YOLOv12n+CRS+FC+CBAM	1,988,229	0.975	0.833	0.975	0.831
YOLOv12n+CRS+FC+SE	1,975,715	0.974	0.858	0.975	0.858
YOLOv12n+CRS+FC+ECA	1,971,622	0.975	0.833	0.974	0.832
YOLOv12n+CRS+FC+STAR	2,242,339	0.977	0.836	0.977	0.834
YOLOv12n+CRS+FC+InceptionNext	2,005,187	0.973	0.859	0.975	0.857
YOLOv12n+CRS+FC+BM	2,054,499	0.975	0.832	0.975	0.834
WS-Det	**2,191,459**	**0.978**	**0.871**	**0.979**	**0.868**

Table 6. Model evaluation on the whalesharktest2 dataset compared with efficient architectures and recent attention modules

Model	Parameter	Validation	
		mAP 50	mAP 50:95
YOLOv11n	2,590,035	0.995	0.900
YOLOv10n	2,707,430	0.995	0.891
YOLOv8n	3,011,027	0.995	0.889
YOLOv5n	2,508,659	0.995	0.863
YOLOv12n	2,568,243	0.995	0.900
YOLOv12n+CRS+FC+CBAM	1,988,229	0.995	0.893
YOLOv12n+CRS+FC+SE	1,975,715	0.995	0.915
YOLOv12n+CRS+FC+STAR	2,242,339	0.995	0.909
YOLOv12n+CRS+FC+BM	2,054,499	0.995	0.900
WS-Det	**2,191,459**	**0.995**	**0.919**

Furthermore, YOLOv5n and YOLOv3t perform worse than the baseline YOLOv12n, even though YOLOv3t uses more parameters. This indicates that older architectures are less efficient in balancing accuracy and computational complexity than the more optimized YOLOv12n framework. Additionally, YOLOv12s, the small variant, achieves an mAP50 of 0.976 and an mAP50:95 of 0.87 with 5.75 million parameters, indicating improved detection precision while maintaining moderate computational complexity. We integrated several attention modules into the YOLOv12n architecture to evaluate their impact on detection performance. When the Convolutional Block Attention Module (CBAM) was combined with CRS and FC within the baseline network, it achieved an mAP50 of 0.975 and an mAP50:95 of 0.831, utilizing 1.99 million parameters.

Similarly, the Squeeze-and-Excitation (SE) mechanism yielded comparable parameter efficiency, achieving an mAP50 of 0.975 and an mAP50:95 of 0.858. Furthermore, the Efficient Channel Attention (ECA) module recorded similar accuracy, reaching an mAP50 of 0.974 and an mAP50:95 of 0.832, with 1.97 million parameters. This evaluation also examines the performance of the CRS and FC support modules integrated into the Recent Attention Module. The STAR block [25], which employs a gated mechanism based on element-wise multiplication, achieves an mAP50 of 0.977 and an mAP50:95 of 0.834, utilizing a total of 2.25 million parameters.

Moreover, the InceptionNext module [26], which employs multi-scale feature aggregation integrated with CSR and FC within the baseline framework, achieves an mAP50 of 0.975 and an mAP50:95 of 0.857, utilizing 2 million parameters. In addition, the Bilinear Mapping (BM) module [27], which leverages a bottleneck Module for channel compression and feature refinement, attains an mAP50 of 0.975 and an mAP50:95 of 0.834, using 2.05 million parameters.

To evaluate the proposed model's generalizability, Table 6 presents performance comparisons on the whalesharktest2 datasets [28]. The results indicate that WS-Det achieves the highest mAP50:95 of 0.919, representing a 2.11% improvement over YOLOv12n. In addition, the SE module performs competitively, with an mAP50:95 of 0.915. However, WS-Det consistently delivers the best results. Table 7 also presents the consistency analysis on the whalesharktest2 dataset, using the mean and standard deviation from three independent training runs. The results indicate stable performance across all models, with identical mean values and zero standard deviation. Specifically, YOLOv12n+CRS achieves an mAP50:95 of 0.900, while YOLOv12n+CRS+FC records 0.888. The proposed model attains a higher mAP50:95 of 0.919 under the same conditions. These results confirm that all models train consistently, with no observable variance or outliers.

Efficiency Analysis

Deep learning algorithms typically require extensive computation and involve many neuron weights. This parameter is captured from several convolutional layers used for feature extraction, where each layer progressively learns to discriminate relevant information. Therefore, efficiency is critical to evaluating the proposed model's resource-saving performance.

Several studies [29, 30, 31, 32] have also emphasized this analysis to demonstrate the practical advantages of their models in real-world applications. This analysis measures and compares the proposed model's performance with other efficient architectures. It enables a comprehensive evaluation of each model's strengths and weaknesses, particularly in terms of multiplication operations and trained parameters.

Table 7. Statistical analysis with mean and standard deviation computed from three independent training runs

Model	Parameter	mAP 50		mAP 50:95	
		Mean	Std	Mean	Std
YOLOv12n	2,568,243	0.995	0	0.900	0
YOLOv12n+CRS	1,466,339	0.995	0	0.900	0
YOLOv12n+CRS+FC	1,434,083	0.995	0	0.888	0
WS-Det	**2,191,459**	**0.995**	**0**	**0.919**	**0**

Table 8. Comparison of inference speed among YOLO models based on parameters, GFLOPs, FPS, and latency on a CPU device

Models	FPS on PC Desktop	FPS on Raspi 5	Parameters	GFLOPs
YOLOv11n	18.74	8.87	2,590,035	6.4
YOLOv10n	14.13	7.27	2,707,430	8.4
YOLOv9t	15.83	7.38	2,005,603	7.8
YOLOv8n	15.58	7.41	3,011,027	8.2
YOLOv5n	15.68	7.72	2,508,659	7.2
YOLOv3t	7.61	5.13	12,132,642	19
YOLOv12s	7.34	3.4	5,754,036	19.3
YOLOv12n	14.98	6.3	2,568,243	6.5
YOLOv12n+CRS+FC+CBAM	16.86	6.23	1,988,229	6.1
YOLOv12n+CRS+FC+SE	17.79	6.66	1,975,715	6.1
YOLOv12n+CRS+FC+ECA	17.86	6.8	1,971,622	6.1
YOLOv12n+CRS+FC+STAR	15.53	6.58	2,242,339	6.2
YOLOv12n+CRS+FC+InceptionNext	16.59	6.93	2,005,187	6.1
YOLOv12n+CRS+FC+BM	16.47	6.74	2,054,499	6.2
WS-Det	**17.19**	**6.64**	**2,191,459**	**6.3**

Moreover, this analysis assesses the feasibility of deploying the detector in practical examples without compromising object localization and classification performance [33]. To evaluate model efficiency, we compared computational complexity, parameter count, latency, and processing speed. Computational complexity refers to the number of multiply-accumulate operations in the network, which generally increases as the network becomes deeper. The parameter count represents the total number of trainable weights, indicating the model's capacity to learn detailed features and produce accurate predictions [34].

For practical implementation, we also analyzed processing speed and latency on a resource-constrained device by measuring inference time, which reflects the proposed model's data-processing interval. The proposed model contains 2.2 million parameters and requires 6.3 GFLOPs, both lower than the original YOLOv12n. This represents a reduction of approximately 300K parameters and 0.2 GFLOPs compared to the baseline. Notably, these results indicate that the proposed model achieves greater computational efficiency and fewer trainable parameters than other lightweight YOLO variants. The efficiency experiments, presented in Table 8, were conducted on low-cost devices using high-resolution video input at 1920 × 1080 pixels. This approach involves maintaining consistency with the dataset configuration. The testing devices included a Raspberry Pi 5 equipped with a 64-bit quad-core Arm Cortex-A76 processor and a PC desktop powered by an Intel Core i7-12700 processor.

DISCUSSION

The proposed model achieves the best overall performance, with an mAP50 of 0.979 and an mAP50:95 of 0.868, supported by 2.19 million parameters. It improves mAP50:95 by about 1.0% over the baseline YOLOv12n while reducing the parameter count by nearly 15%. These results demonstrate that the enhancement module effectively strengthens the model's capability to capture discriminative whale shark features while maintaining robustness in complex marine environments. Overall, our model achieves an excellent balance between accuracy, efficiency, and computational complexity, making it highly suitable for real-time marine object detection applications. Furthermore, Figure 8 presents heatmap visualizations that show how effectively the proposed model captures and interprets important visual cues during the decision-making process. The heatmap visualizations are generated using EigenCAM [35], a gradient-free technique that applies principal component analysis (PCA) to highlight the most dominant features learned by the convolutional layers. The proposed network effectively captures distinctive whale shark patterns by focusing on the body region. This visualization further indicates that the model attends to key visual cues relevant to accurate detection and classification, demonstrating strong feature-representation capability. The highlighted regions in the heatmaps indicate that the model concentrates on the most informative areas of the whale shark, demonstrating its ability to identify discriminative features that contribute to accurate detection and classification.

In the ablation study, the performance improvement is attributed to integrating the integrated feature modulation module, which effectively emphasizes relevant information within high-level feature representations. Although this integration slightly increases the parameter count to 2,191,459 and the computational cost to 6.3 GFLOPs, the overall complexity remains lower than YOLOv12n, confirming WS-Det's superior efficiency and balanced performance. In addition, the experimental results show that the channel reduction strategy and integration of the Fast Convolutional module effectively reduce parameter count and computational load. Removing redundant layers in YOLOv12n eliminates unnecessary channel convolution operations, thereby improving efficiency. Conversely, including the enhancement module improves detector performance, albeit with a slight trade-off in efficiency.

This network efficiency stems from several strategies and modules applied to the proposed detector to reduce the growing dependence on operations and weights. First, the channel reduction strategy reduces the number of unnecessary convolution layers, lowering computation and trainable weights. Second, the proposed Fast Convolutional block maintains performance while applying lightweight convolution operations through partial convolution. These advantages allow the proposed model to achieve higher efficiency than its counterparts. This observation indicates that the proposed module requires fewer computational resources while maintaining accurate whale shark localization, achieving precision comparable to the original YOLOv12n model.

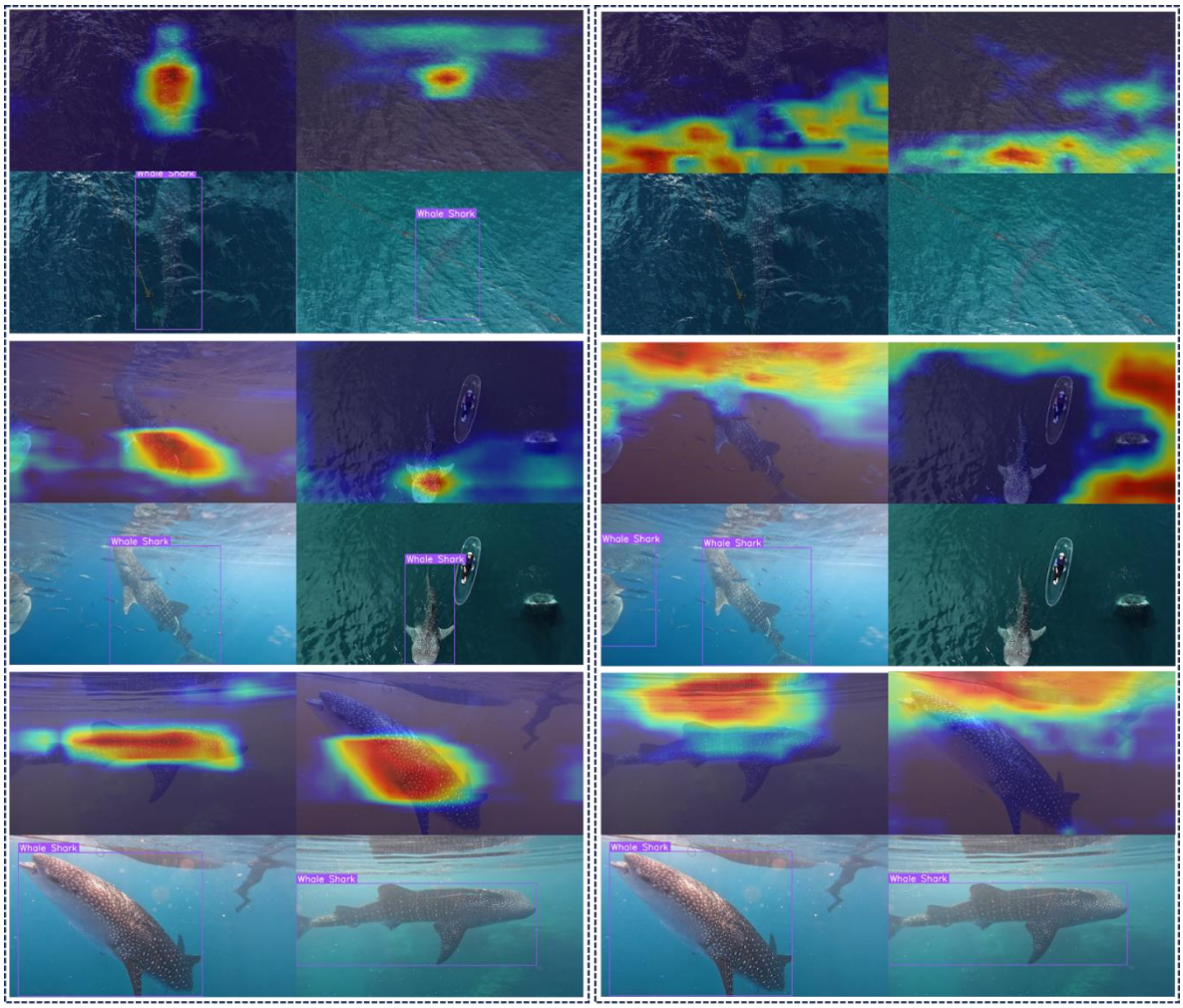

Figure 8. Heatmap visualization using EigenCAM, showing (a) the proposed model and (b) the YOLOv12-nano baseline

The efficiency analysis includes several experiments to evaluate the proposed model's inference speed. The results show that the proposed model achieves 17.19 FPS on the PC Desktop, outperforming YOLOv12n, YOLOv10n, and YOLOv8n. Although YOLOv11n runs slightly faster, it shows lower overall detection performance. Additionally, we convert the model to the OpenVINO format for deployment on the Raspberry Pi 5, enabling faster real-time inference on the Arm CPU. Under these conditions, the WS-Det achieves 6.64 FPS, showing competitive performance against other YOLO models. Furthermore, the channel reduction strategy improves inference speed, while the enhancement module causes only a slight decrease in processing speed.

CONCLUSION

This paper introduces an efficient whale shark detection system based on a modified YOLOv12n architecture enhanced with an Integrated Feature Modulation (IFM) module. The module improves detection robustness under diverse sea conditions, including both underwater and surface environments. The WS-Det model improves efficiency through a channel-reduction strategy, effectively reducing parameter and computational complexity. This optimization enables the integration of the IFM module, which further boosts detection precision. The proposed model achieves an mAP50:95 of 0.868, marking a 0.7% improvement over the original YOLOv12n. Despite its stronger performance, it remains lightweight, with only 2.2 million parameters and 6.3 GFLOPs, 0.2 GFLOPs lower than YOLOv12-nano.

The model was trained and evaluated using a custom dataset collected in the Gorontalo region, encompassing both surface and underwater imagery to capture the diverse visual appearances of whale sharks. The WS-Det achieves an inference speed of 17.19 FPS on CPU-based devices, demonstrating its effectiveness for real-time applications under resource-constrained conditions. Future research should focus on enhancing detector adaptability to varying illumination environments, including low-light and nighttime scenarios.

ACKNOWLEDGMENT

This research was supported by the Directorate of Research, Technology, and Community Service, Ministry of Education, Culture, Research, and Technology, Indonesia, through the 2025 Fundamental Research Scheme under Grant No.137/C3/DT.05.00/PL/2025 and No. 035/SK/C.02/LPPM-UNISAN/2025.

REFERENCES

[1] T. Setiadi, J. Wahyuni, and I. S. Lantu, “Stakeholder dynamics in whale shark conservation: A social-economic analysis of marine tourism in Botubarani Beach, Gorontalo, Indonesia,” *IOP Conference Series: Earth and Environmental Science*, vol. 1461, no. 1, p. 012047, Mar. 2025, doi: 10.1088/1755-1315/1461/1/012047.

[2] T. C. Maillard, F. Garzon, L. A. Hawkes, G. R. Tabor, and M. J. Witt, ‘Refining Electronic Tagging of Marine Animals: Computational Fluid Dynamics and Pelagic Sharks’, *Animals*, vol. 15, no. 20, 2025, doi: https://doi.org/10.3390/ani15202956.

[3] 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," in *IEEE Access*, vol. 13, pp. 171356-171370, 2025, doi: 10.1109/ACCESS.2025.3615067.

[4] M. D. Putro, D. G. S. Ruindungan, R. Syahputra, T. H. Oh, I. Y. Chun, and V. C. Poekoel, "An efficient and effective sea turtle detection using positioning enhancement module," in *Proc. 2024 International Workshop on Intelligent Systems (IWIS)*, Seoul, Korea, Aug. 18–20, 2024, doi: 10.1109/IWIS62722.2024.10706029.

[5] Y. Zheng, J. Yang, Y. Chen, Y. Wang, Y. Lu and G. Li, "Beluga Whale Detection from Satellite Imagery with Point Labels," *IGARSS 2025 - 2025 IEEE International Geoscience and Remote Sensing Symposium*, Brisbane, Australia, 2025, pp. 1298-1302, doi: 10.1109/IGARSS55030.2025.11243110.

[6] K. M. Green, M. K. Virdee, H. C. Cubaynes, A. I. Aviles-Rivero, P. T. Fretwell, P. C. Gray, D. W. Johnston, C. Schönlieb, L. G. Torres, and J. A. Jackson, "Gray whale detection in satellite imagery using deep learning," *Remote Sens. Ecol. Conserv*., vol. 9, no. 6, pp. 829–840, 2023, doi: 10.1002/rse2.352.

[7] R. Zhang, Y. Shi, L. Yang and Z. Zhou, "Classification of Whale Call Signals Based on the Improved YOLOv5 Model," *2023 6th International Conference on Information Communication and Signal Processing (ICICSP)*, Xi'an, China, 2023, pp. 429-433, doi: 10.1109/ICICSP59554.2023.10390548.

[8] J. Liu, F. Liu, S. Qian and Y. Zhao, "Shark Image Classification and Recognition Based on Convolutional Neural Networks," *2025 IEEE 2nd International Conference on Big Data Science and Engineering (ICBDSE)*, Kunming, China, 2025, pp. 1-5, doi: 10.1109/ICBDSE65491.2025.11219696.

[9] N. A. Le, J. Moon, C. G. Lowe, H. I. Kim, and S. I. Choi, "An automated framework based on deep learning for shark recognition," *J. Mar. Sci. Eng*., vol. 10, no. 7, p. 942, 2022, doi: 10.3390/jmse10070942.

[10] L. Kuhlane, D. Brown, and A. Boby, "Exploring the incremental improvements of YOLOv5 on tracking and identifying great white sharks in Cape Town," *Remote Sens. Ecol. Conserv*., vol. 9, no. 4, pp. 567–582, 2023, doi: 10.1002/rse2.412.

[11] S. Zhao, J. Zheng, S. Sun, and L. Zhang, "An improved YOLO algorithm for fast and accurate underwater object detection," *Symmetry (Basel)*, vol. 14, no. 8, p. 1669, 2022, doi: 10.3390/sym14081669.

[12] K. Liu, L. Peng, and S. Tang, "Underwater object detection using TC-YOLO with attention mechanisms," *Sensors*, vol. 23, no. 5, p. 2567, 2023, doi: 10.3390/s23052567.

[13] P. Xu, X. Luo, and J. Li, "Improved YOLOv8 underwater object detection combined with CBAM," in *Proc. 2024 International Symposium on Digital Home (ISDH)*, 2024, pp. 43–48, doi: 10.1109/ISDH64927.2024.00014.

[14] Q. Guo, Y. Wang, Y. Zhang, H. Qin, H. Qi, and Y. Jiang, "AWF-YOLO: Enhanced underwater object detection with adaptive weighted feature pyramid network," *Complex Eng. Syst*., vol. 3, p. 16, 2023, doi: 10.20517/ces.2023.19.

[15] J. Zhang, H. Chen, X. Yan, K. Zhou, J. Zhang, Y. Zhang, H. Jiang, and B. Shao, "An improved YOLOv5 underwater detector based on an attention mechanism and multi-branch reparameterization module,"

Electronics, vol. 12, p. 2597, 2023, doi: 10.3390/electronics12122597.

[16] V. Kumar Ancha, F. N. Sibai, V. Gonuguntla and R. Vaddi, "Utilizing YOLO Models for Real-World Scenarios: Assessing Novel Mixed Defect Detection Dataset in PCBs," in *IEEE Access*, vol. 12, pp. 100983-100990, 2024, doi: 10.1109/ACCESS.2024.3430329.

[17] S. Silva, F. H. A. Moraes Neto, P. H. Ferreira and D. B. Costa, "CNN-Based YOLOv12 for Damage Assessment in Residential Roofs," in *IEEE Access*, vol. 13, pp. 193311-193322, 2025, doi: 10.1109/ACCESS.2025.3629630.

[18] J. Huang, C. Fang, X. Zheng, and J. Liu, "YOLOv8-UC: An Improved YOLOv8-Based Underwater Object Detection Algorithm," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3496925.

[19] M. L. Martins, M. T. Coimbra and F. Renna, "Understanding Squeeze-and-Excitation Layers for Medical Image Segmentation," *2025 33rd European Signal Processing Conference (EUSIPCO)*, Palermo, Italy, 2025, pp. 1532-1536, doi: 10.23919/EUSIPCO63237.2025.11226703.

[20] 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, doi: https://doi.org/10.1007/978-3-031-26313-2.

[21] S. Li, Z. Wang, Z. Liu, C. Tan, H. Lin, D. Wu, et al., "Moganet: Multi-order gated aggregation network," in *International Conference on Learning Representations*, 2022, doi: 10.48550/arXiv.2211.03295.

[22] D. A. Andradi-Brown et al., "Diversity in marine protected area regulations: Protection approaches for locally appropriate marine management," *Frontiers in Marine Science*, vol. 10, p. 1099579, 2023, doi: 10.3389/fmars.2023.1099579.

[23] D. Guo *et al.*, 'SMA-YOLO: An enhanced architecture for the detection of corn diseases based on YOLOv12', *Smart Agricultural Technology*, vol. 12, p. 101502, 2025, doi: 10.1016/j.atech.2025.101502.

[24] L. T. Ramos and A. D. Sappa, "A Decade of You Only Look Once (YOLO) for Object Detection: A Review," in *IEEE Access*, vol. 13, pp. 192747-192794, 2025, doi: 10.1109/ACCESS.2025.3630988.

[25] X. Ma, X. Dai, Y. Bai, Y. Wang and Y. Fu, "Rewrite the Stars," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2024, pp. 5694-5703, doi: 10.1109/CVPR52733.2024.00544.

[26] W. Yu, P. Zhou, S. Yan and X. Wang, "InceptionNeXt: When Inception Meets ConvNeXt," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2024, pp. 5672-5683, doi: 10.1109/CVPR52733.2024.00542.

[27] Z. Peng et al., "Star with Bilinear Mapping," *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2025, pp. 25292-25302, doi: 10.1109/CVPR52734.2025.02355.

[28] ThesisTest1, "Whale Sharks Test Dataset," Roboflow Universe, 2024. [Online]. Available: https://universe.roboflow.com/thesistest1-kfdfb/whalesharkstest2/dataset/1.

[29] T. Kashyap and P. Rana, "Lightweight object detection model for False Smut detection in rice leaf with Eigen-cam Interpretability," *2025 2nd International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, Chennai, India, 2025, pp. 1-6, doi: 10.1109/RMKMATE64874.2025.11042591.

[30] M. D. Putro, D. -L. Nguyen and K. -H. Jo, "A Fast CPU Real-Time Facial Expression Detector Using Sequential Attention Network for Human–Robot Interaction," in *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7665-7674, Nov. 2022, doi: 10.1109/TII.2022.3145862.

[31] A. Suryanto, D. H. Wicaksono, A. Mulwinda, M. Harlanu, and M. N. Syah, "Car seatbelt monitoring system using real-time object detection algorithm under low-light and bright-light conditions," *SINERGI*, vol. 29, no. 3, pp. 771–778, 2025, doi: 10.22441/sinergi.2025.3.018.

[32] Moh. Khairudin et al., "Early detection of diabetes potential using cataract image processing approach," *SINERGI*, vol. 28, no. 1, pp. 55–62, 2024, doi: 10.22441/sinergi.2024.1.006

[33] P. Ge *et al.*, "SGA-YOLO: A Lightweight Real-Time Object Detection Network for UAV Infrared Images," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 12, pp. 22432-22446, Dec. 2025, doi: 10.1109/TITS.2025.3616525.

[34] X. Tang *et al.*, "YOLO-Ant: A Lightweight Detector via Depthwise Separable Convolutional and Large Kernel Design for

Antenna Interference Source Detection," in *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-18, 2024, Art no. 5016018, doi: 10.1109/TIM.2024.3379397.

[35] J. Choi, M. Rajendran and Y. Cheol Suh, "Explainable Rip Current Detection and Visualization with XAI EigenCAM," *2024 26th International Conference on Advanced Communications Technology (ICACT)*, Pyeong Chang, Korea, Republic of, 2024, pp. 1-6, doi: 10.23919/ICACT60172.2024.10471913.