An Efficient Deep Learning Model for Whale Shark Detection

Imanuel Kutika
Master Program of Informatics,
Postgraduate Program
Sam Ratulangi University
Manado, Indonesia
imanuelkutika026@student.unsrat.ac.id

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

Vicky Nolant Setyanto Lahimade
Master Program of Informatics,
Postgraduate Program
Sam Ratulangi University
Manado, Indonesia
vickylahimade026@student.unsrat.ac.id

***Abstract* — Marine conservation depends on efficient whale shark (Rhincodon typus) monitoring, yet traditional observation techniques are still expensive, intrusive, and challenging to scale. This paper suggests a lightweight deep learning-based detection system that uses the YOLOv10 architecture for real-time underwater monitoring in order to overcome these drawbacks. A carefully selected dataset that was taken in different lighting and visibility circumstances was used to train the algorithm. With just 2.7 million parameters and 8.4 GFLOPs, experimental results show that the suggested method achieves a mAP@50 of 97.2% and mAP@50–95 of 85.5%, demonstrating a solid balance between detection accuracy and computational economy. These results validate YOLOv10's potential for real-time, resource-efficient whale shark detection in marine monitoring applications.**

***Keywords*— Whale Shark, Object Detection, Convolutional Neural Network, YOLOv10, Deep Learning**

I. INTRODUCTION

The health of marine ecosystems depends on whale sharks (Rhincodon typus), the largest known species of fish [1]. As filter feeders, they play a key role in regulating plankton populations, contributing to the balance of marine food webs, and influencing nutrient cycling within the oceans [2]. Despite their importance, whale sharks are classified as endangered, facing numerous threats such as habitat degradation, overfishing, and climate change [3]. Effective monitoring of whale shark populations and understanding their behaviour are critical for informing conservation strategies and ensuring the long-term survival of the species. Traditional methods of tracking these animals, such as manual surveys or boat-based observations and GPS tagging are not only resource-intensive and costly but also limited in their scalability, effectiveness while protecting the whale sharks in remote or expansive marine environments [4]. As such, there is a growing need for more efficient, scalable, and less invasive techniques to monitor whale sharks, with the aim of supporting sustainable conservation efforts while minimizing human impact on these vulnerable creatures.

The detection of marine species can now be automated thanks to recent developments in deep learning-based computer vision, offering more effective, scalable, and affordable solutions. The You Only Look Once (YOLO) family of object recognition models has garnered significant interest among these technologies because of its ability to provide accurate and fast real-time identification [5]. A lightweight version of the YOLO architecture, YOLOv10, maintains a convincing compromise between computational economy and performance, making it well-suited for edge-devices applications like marine species monitoring, which have been developed on multiple related studies [6–10].

This study presents a YOLOv10-based whale shark detection system tailored to the challenges of underwater imaging, such as variable visibility, lighting, and marine clutter. The model, which was trained on a carefully selected undersea dataset, achieves great detection performance and computational efficiency, allowing for large-scale, real-time inferences [11]. The system demonstrates strong potential for non-invasive, continuous monitoring of whale shark populations, supporting marine conservation through:

- Develop and validate an efficient, real-time whale shark detection system based on YOLOv10.
- Evaluate the usage of this model on resource-constrained environments.
- Improve conservation efforts by providing researchers with a reliable tool for large-scale monitoring, helping to ensure better protection for this endangered species and promoting the preservation of marine biodiversity.

II. METHODS

A lightweight deep learning system designed for real-time object recognition, especially in vehicle detection applications, is the YOLOv10n architecture. It identifies and classifies multiple vehicle types while maintaining efficiency. The framework comprises three core modules which show in Figure 1: the Backbone, which extracts essential visual features; the Neck, which fuses multi-scale feature maps for effective representation; and the Head, which performs classification and confidence estimation. By utilizing these layers, YOLOv10n achieves efficient feature extraction and robust accuracy.

Received: 13 March 2026, Revision: 17 April 2026, Accepted: 24 April 2026
International Journal of Informatics, Data, and Emerging Applications, Vol 1, No 1, January – June, 2026, pp. 24-28
available at: https://ejournal.unsrat.ac.id/v3/index.php/idea/

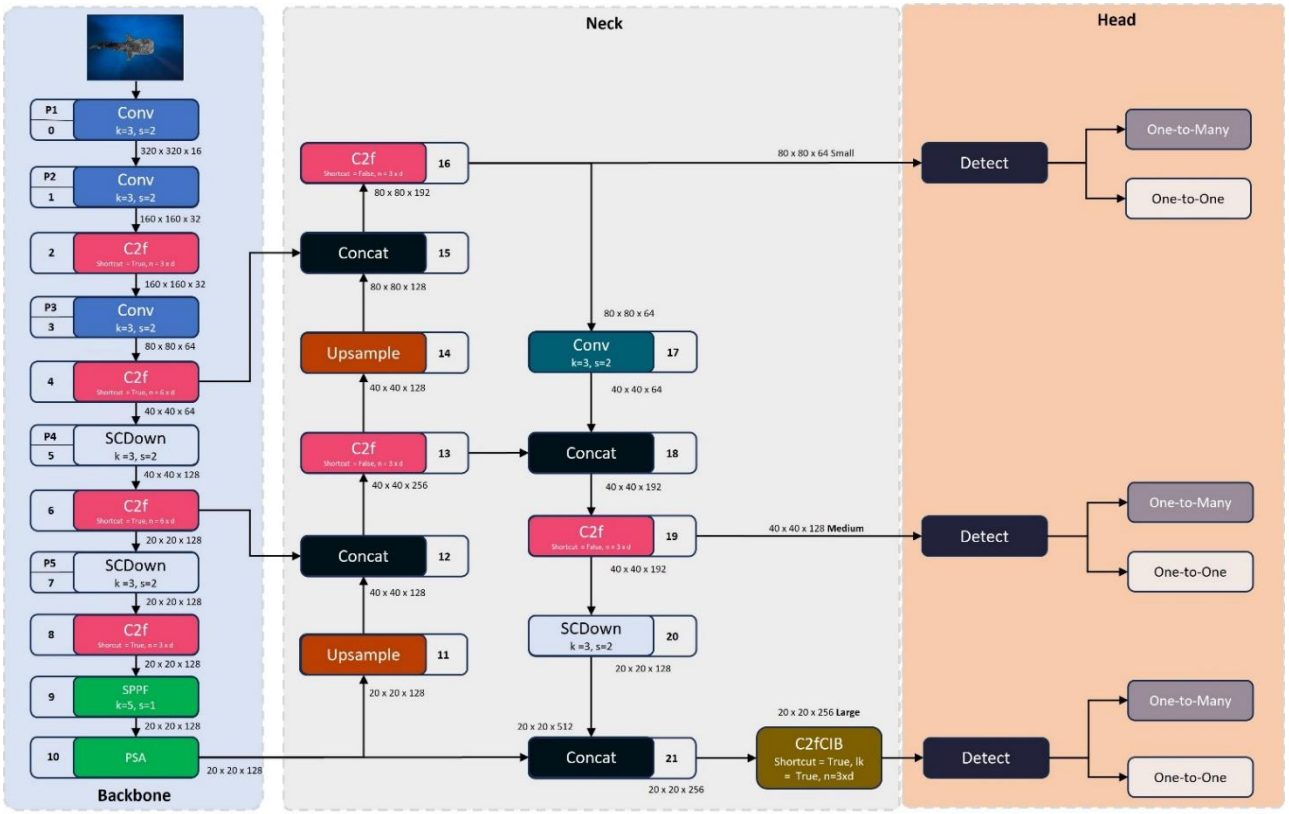

Fig 1. YOLOv10n architecture showing the backbone, neck, and detection head components.

A. Backbone

The convolutional (Conv) layer performs various tasks at the beginning of the backbone feature extraction process, including convolution, batch normalisation, and the SiLU activation function. Batch normalization is applied to accelerate and stabilize the training process by reducing internal covariate shifts. This technique can also enhance the learning rate and, in certain cases, reduce the dependence on dropout regularization [12]. The SiLU activation function, also referred to as the Swish function, effectively addresses the issue of neurons becoming inactive at zero outputs. Acting as a smooth gating mechanism for negative inputs, SiLU behaves similarly to a linear function but introduces a gradual, non-zero transition that improves model learning dynamics [13].

The next layer in the architecture is the C2f block, which derives its name from the Cross Stage Partial (CSP) bottleneck structure with two convolutional operations [14]. A 1×1 convolution is the first step in the procedure, and then the feature map is divided based on an expansion factor. The outputs from each branch are then concatenated and passed through another 1×1 convolution to fuse the extracted features for subsequent processing in the next layer. Some feature paths are propagated continuously between the Conv and C2f layers, while others bypass certain operations to improve computational efficiency. The C2f structure facilitates better gradient flow during training and helps stabilize optimization in deeper networks. Additionally, it allows for richer representations at a reduced computational cost while maintaining crucial feature information. Before moving on to the next step, the merged features are blended using a 1×1 convolution at the conclusion of the C2f block.

The Spatial Pyramid Pooling Fast (SPPF) layer is a crucial component for extending the model's spatial context and improving its ability to recognise objects at different scales. uses window sizes of 5, 9, and 13 to do a series of two-dimensional max pooling operations after starting with a 1x1 convolution. Contextual data is aggregated at various spatial resolutions using these pooling layers. The initial 1x1 convolution output is then concatenated with the feature maps that are obtained from each pooling stage. This architecture significantly expands the receptive field and enhances the feature representation while maintaining computational economy by utilising sequential max pooling processes instead of additional convolutional layers.

In order to enhance the representation of global contextual data while maintaining computational efficiency, the architecture includes a Partial Self-Attention (PSA) block following the SPPF layer. The PSA block includes a Feed-Forward Network (FFN), a unidirectional network structure responsible for processing multiscale characteristics [15]. In addition to applying nonlinear transformations, the FFN can serve as the last phase in tasks involving prediction or classification. The feature maps are separated into discrete branches following the initial processing. A deeper combination of spatial and contextual information is made possible by concatenating the final feature maps with those generated by the FFN branch. These features are then fused using a 1x1 convolution, producing the output that is then sent to the following processing block.

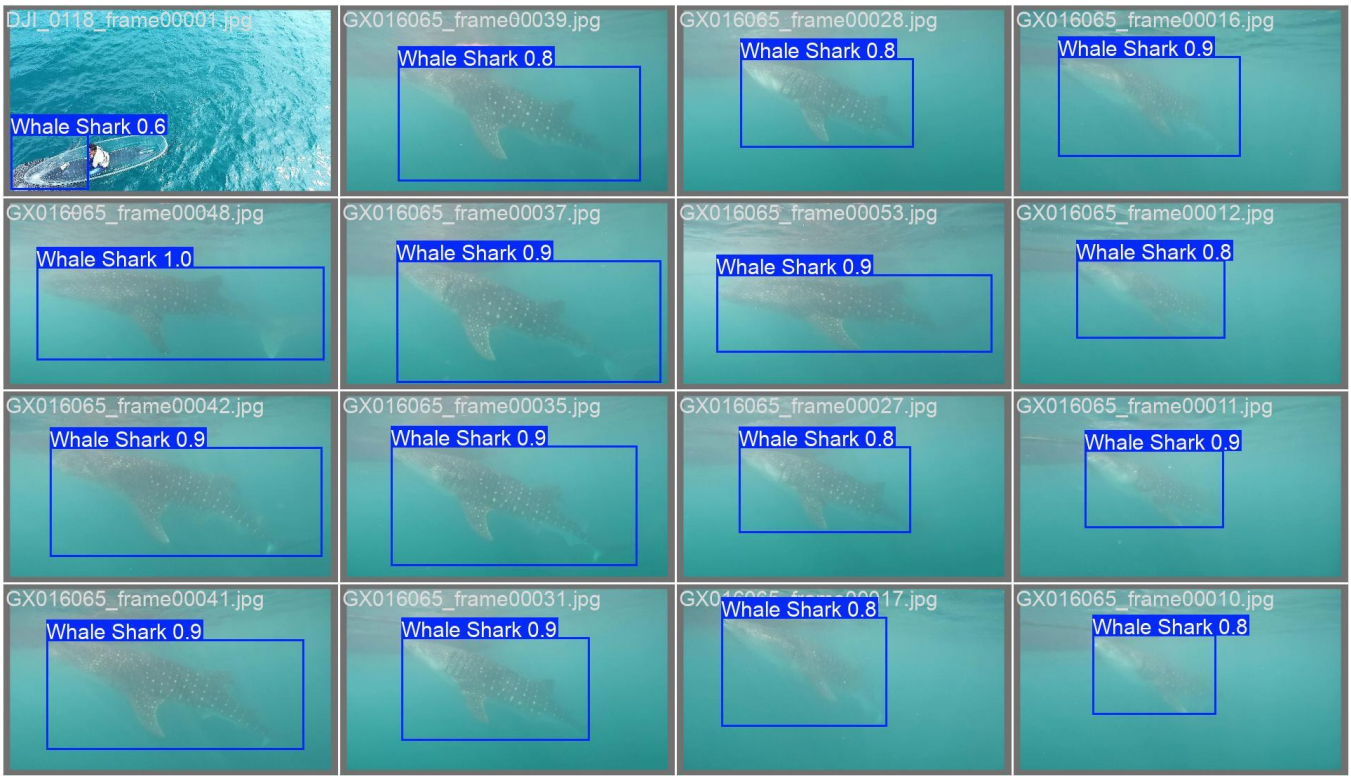

Fig. 2 Detection Result on Whale Shark

B. Neck

An integral part of the object identification system is the neck module, which combines and enhances the multi-scale data that the backbone retrieves. It commonly employs a Path Aggregation Network (PAN) to facilitate feature fusion across multiple spatial resolutions, thereby improving detection performance under various conditions [16]. Within this structure, C2f blocks are utilized to enhance feature representation and promote efficient information flow between layers. In order to compensate for the loss of detail that results from downsampling, an Upsample block is also included to restore spatial resolution by increasing the feature maps. A Concat block connects these elements, enhancing feature diversity and aligning feature dimensions. When combined, these processes enhance the architecture's ability to recognise things of different sizes.

C. Head

The head component is in charge of classifying and localising objects, producing class probabilities, and regressing bounding box coordinates for every object that is identified. YOLOv10 provides a major advance over previous systems by substituting the conventional Non-Maximum Suppression (NMS), which can be computationally taxing when handling a large number of detected bounding boxes [17]. Rather, YOLOv10 uses a new approach called Consistent Dual Assignments. During training, the One-to-Many head creates numerous predictions to deliver dense supervisory signals, and the One-to-One head, which uses a label assignment method to guarantee that a single prediction corresponds to each ground-truth instance, are the two complementary parts of this mechanism. This dual-head approach greatly increases detection accuracy and overall model robustness in addition to improving training process efficiency.

The head computes the difference between predictions and ground-truth annotations using multiple loss functions. Among these, the Complete Intersection over Union (CIoU) loss improves bounding box regression by taking aspect ratio consistency, overlap area, and box center distance into consideration [18]. As a result, localisation becomes more precise and reliable. Furthermore, YOLOv10 includes the Distribution Focal Loss (DFL), which aligns the projected probability distribution with a target determined from the ground-truth box to reformulate bounding box regression as a distributional prediction issue [20]. This technique makes it possible for the model to produce a single, ideal forecast for each observed object and enhances the head's ability to produce accurate bounding boxes.

D. Implementation Setup

The arrangement shown in Table 2 was used in this investigation to train and evaluate the YOLOv10 model. To guarantee effective training and inference, a PC platform equipped with an accelerator NVIDIA RTX 3060 GPU was utilised. To balance computing performance and detail, the input image size was adjusted to 640 × 640 pixels. With a batch size of 16, the model was trained over 150 epochs. The Stochastic Gradient Descent (SGD) method was used for optimisation since it is robust in object detection tasks [19]. Table 1 displays the whole training arrangement. To enhance generalisation and speed up convergence, a learning rate of 0.01 was used.

TABLE 1. TRAINING AND TESTING CONFIGURATION

Parameters	Setup
Platform/device	AMD Ryzen 5 - 8 Core
GPU	Nvidia Colourful RTX 3060
Image Size	640 x 640 pixels
Epochs	150
Batch Size	16
Optimizer	SGD
Learning Rate	0,01

TABLE 2. EVALUATION MODEL PERFORMANCE

Model	*GFLOPS*	*Parameter*	*mAP 50%*	*mAP 50-95%*
YOLOv10n	8.4	2.71	97.2	85.5

III. RESULT AND DISCUSSION

A. Evaluation Model Performance

The mean Average Precision (mAP) metric was used to assess the suggested YOLOv10-based whale shark identification method, as indicated in Table 2. Specifically, the average precision calculated at an Intersection over Union (IoU) threshold of 0.50 is called mAP@50, indicating basic detection accuracy, whereas the average precision calculated across IoU thresholds from 0.50 to 0.95 is known as mAP@50-95, offering a more thorough evaluation of localisation performance. The evaluation demonstrated that the YOLOv10 model successfully balances high accuracy and computational efficiency, making it perfect for real-time, resource-efficient marine monitoring applications.

With a mAP@50 of 97.2% and a mAP@50-95 of 85.5%, the YOLOv10 model showed exceptional detection capabilities even in challenging underwater conditions, such as varying lighting and visibility. These findings demonstrate the model's ability to locate and recognise whale sharks in a variety of marine habitats, significantly increasing the effectiveness of automated monitoring systems for conservation efforts.

With just 2,707,430 parameters and 8.4 GFLOPs, the model's small architecture guarantees outstanding performance and computational efficiency. The model can be implemented on devices with limited resources, like mobile platforms or edge devices, thanks to its minimal parameter count, without sacrificing real-time inference capabilities. This effectiveness is especially crucial for large-scale deployments when ongoing whale shark population monitoring requires prompt, on-site detection.

The model's robustness is further supported by the normalised confusion matrix (Figure 3), which displays strong true positive rates in a variety of detection settings. The model demonstrated consistent accuracy in detecting whale sharks, even in complex underwater environments where challenges such as light attenuation and background noise are common.

B. Efficiency Analysis

The capacity of the detector to perform well in practical applications, especially when used on devices with limited resources, is referred to as model efficiency. A key element of this efficiency is the trade-off between detection performance and processing cost. The YOLOv10-based whale shark identification algorithm used in this work was created to be highly accurate and computationally efficient,

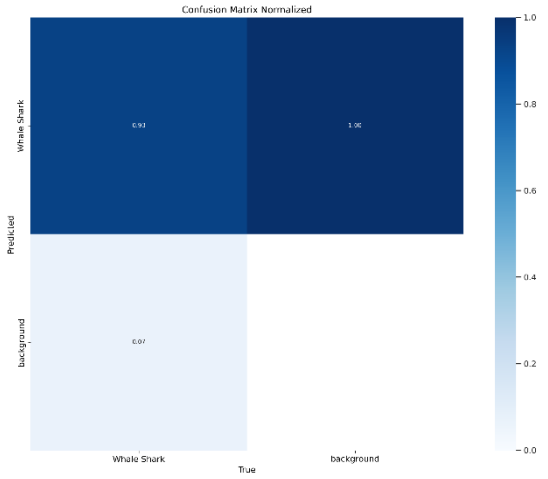

Fig. 3 Confusion Matrix Normalized

created to be highly accurate and computationally efficient, making it appropriate for large-scale, real-time marine monitoring.

Key metrics like the number of parameters, inference speed, and computational cost (GFLOPs) were calculated to assess the YOLOv10 model's computational efficiency. With just 2,707,430 parameters and 8.4 GFLOPs, the model is lightweight and efficient for deployment on edge devices or systems with modest computing resources. This compact architecture ensures that the model can be deployed in marine monitoring systems where processing power and storage capacity are often constrained.

An Intel Core i7 CPU and a Raspberry Pi 4 Model B with a 1.5 GHz quad-core Cortex-A72 processor and 4 GB RAM were the two platforms chosen to evaluate the model's real-time performance. On the Intel Core i7 platform, the model achieved 14.13 FPS with a latency of 0.07 seconds per inference, demonstrating its capability for rapid whale shark detection in time-sensitive applications.

On the Raspberry Pi 4 platform, representing a resource-constrained edge device, the model maintained 7.27 FPS with a latency of 0.14 seconds per inference. Despite its lower performance in comparison to the Intel i7 system, the results demonstrate that YOLOv10 is still useful and practicable for real-time deployment on low-cost embedded devices in marine surveillance scenarios.

IV. CONCLUSION

This study validates the YOLOv10-based model's efficacy in real-time whale shark detection under challenging marine settings. With only 2.7 million parameters and 8.4 GFLOPs, the suggested technique obtains a mAP@50 of 97.2% and mAP@50–95 of 85.5%, exhibiting a remarkable balance between detection accuracy and computational economy. Additionally, the model achieves 14.13 FPS with a latency of 0.07 seconds, suggesting that it may be deployed on edge devices with limited resources. Although minor detection inaccuracies remain, the overall results highlight the model’s robustness in distinguishing whale sharks from surrounding marine objects. These findings underscore the potential of lightweight deep learning architectures for

scalable and efficient marine wildlife monitoring, while future improvements may focus on enhancing feature discrimination to further reduce false positives.

REFERENCES

[1] H. M. Guzman, C. M. Collatos, and C. G. Gomez, "Movement, behavior, and habitat use of whale sharks (Rhincodon typus) in the tropical eastern Pacific Ocean," Frontiers in Marine Science, vol. 9, Art. no. 793248, Jun. 2022.

[2] C. A. Rohner and C. E. Prebble, "Whale shark foraging, feeding, and diet," in Whale Sharks: Biology, Ecology, and Conservation, pp. 153–180, 2021.

[3] D. Rowat, F. C. Womersley, B. M. Norman, and S. J. Pierce, "Global threats to whale sharks," in Whale Shark Biology, Ecology, and Conservation, pp. 239–265, 2021.

[4] D. Daye, R. De La Parra, J. Vaudo, J. Harvey, G. Harvey, M. Shivji, and B. Wetherbee, "Tracking four years in the life of a female whale shark shows consistent migrations in the Gulf of Mexico and Caribbean," Marine and Freshwater Research, vol. 75, no. 10, 2024.

[5] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," Procedia Computer Science, vol. 199, pp. 1066–1073, 2022.

[6] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, and J. Han, "YOLOv10: Real-time end-to-end object detection," Advances in Neural Information Processing Systems, vol. 37, pp. 107984–108011, 2024.

[7] W. Pan, J. Chen, B. Lv, and L. Peng, "Lightweight marine biodetection model based on improved YOLOv10," Alexandria Engineering Journal, vol. 119, pp. 379–390, 2025.

[8] J. Wuntu, M. D. Putro, and R. Syahputra, "Real-time fish detection in Indonesian marine ecosystems using lightweight YOLOv10-nano architecture," arXiv preprint arXiv:2509.17406, 2025.

[9] R. Mai and J. Wang, "UM-YOLOv10: Underwater object detection algorithm for marine environment based on YOLOv10 model," Fishes, vol. 10, no. 4, p. 173, 2025.

[10] Z. Hu and Q. Chen, "MOA-YOLO: An accurate, real-time and lightweight YOLOv10-based algorithm for deep-sea fish detection," IEEE Sensors Journal, 2025.

[11] M. Kholiavchenko, "Comprehensive deep learning pipeline for whale shark recognition," M.S. thesis, Rensselaer Polytechnic Institute, 2022.

[12] R. O. Ogundokun, R. Maskeliunas, S. Misra, and R. Damaševičius, "Improved CNN based on batch normalization and Adam optimizer," in Proc. Int. Conf. Computational Science and Its Applications (ICCSA), Cham, Switzerland: Springer Int. Publ., Jul. 2022, pp. 593–604.

[13] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," Neural Networks, vol. 107, pp. 3–11, 2018.

[14] Y. He, J. Hu, M. Zeng, Y. Qian, and R. Zhang, "DCGC-YOLO: The efficient dual-channel bottleneck structure YOLO detection algorithm for fire detection," IEEE Access, vol. 12, pp. 65254–65265, 2024, doi: 10.1109/ACCESS.2024.3385856.

[15] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, et al., "ResMLP: Feedforward networks for image classification with data-efficient training," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 4, pp. 5314–5321, 2022.

[16] H. Yu, X. Li, Y. Feng, and S. Han, "Multiple attentional path aggregation network for marine object detection," Applied Intelligence, vol. 53, no. 2, pp. 2434–2451, 2023.

[17] D. H. Jeon, T. S. Kim, and J. S. Kim, "A method for reducing false negative rate in non-maximum suppression of YOLO using bounding box density," Journal of Multimedia Information System, vol. 10, no. 4, pp. 293–300, 2023.

[18] S. Du, B. Zhang, P. Zhang, and P. Xiang, "An improved bounding box regression loss function based on CIOU loss for multi-scale object detection," in Proc. IEEE 2nd Int. Conf. Pattern Recognition and Machine Learning (PRML), Chengdu, China, 2021, pp. 92–98, doi: 10.1109/PRML52754.2021.9520717.

[19] X. Li, C. Lv, W. Wang, G. Li, L. Yang, and J. Yang, "Generalized focal loss: Towards efficient representation learning for dense object detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3139–3153, Mar. 2023, doi: 10.1109/TPAMI.2022.3180392.

[20] Y. Tian, Y. Zhang, and H. Zhang, "Recent advances in stochastic gradient descent in deep learning," Mathematics, vol. 11, no. 3, p. 682, 2023.