International Journal of Informatics, Data, and Emerging Applications
Vol 1, No 2, August – December, 2026, pp. 54-59
ISSN: 3164-5089 (Print) | E-ISSN: 3164-6921

YOLO-FAST for Maritime Surveillance: A Lightweight Approach to Ship Detection

Marsel Marhaen Wungow[1], Dayen Manoppo[2], Ni Made Shavitri Mustikayani [3]
[1, 2, 3]Bachelor of Informatics Program, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia
Corresponding e-mail: marselwungow026@student.unsrat.ac.i)

***Abstract*—Ship detection in maritime waters plays a crucial role in ensuring transportation safety, monitoring traffic, and facilitating faster responses during naval accidents. Traditional methods are less effective because they depend heavily on human resources. To address this issue, automated detection systems are required to identify ships and their categories more accurately. Deep learning, particularly Convolutional Neural Networks (CNN), has achieved significant success in object detection, with YOLO (You Only Look Once) widely recognized as a fast and reliable real-time method. YOLO-FAST is one of the latest improvements, offering a good balance between model size and accuracy. For this research, the SeaShips dataset is used for training since it features various ship types and realistic sea conditions. Experimental results show that the proposed model achieves 94.6% mAP50 and 66.9% mAP50:95 while maintaining a lightweight structure with only 2,306,818 parameters and a computational cost of 6.4 GFLOPs. These findings highlight a strong balance between detection accuracy and computational efficiency, making YOLO-FAST suitable for deployment on low-cost devices to support scalable, real-time ship monitoring in maritime environments.**

***Keywords*— Ship, Object Detection, Convolutional Neural Network, YOLO-FAST, Deep Learning**

I. INTRODUCTION

The increasing frequency of vessel collisions in congested urban waterways and ports, driven by diverse ship types, high traffic density, and the rise of autonomous ships, highlights a critical need for advanced monitoring systems [2], [3]. While conventional methods such as AIS and radar exist, they have significant limitations, particularly for vessels that lack active transponders or are too small to be reliably detected [1].

Consequently, image-based detection has become a vital alternative for enhancing navigation safety and port efficiency [7]. However, these visual systems are severely challenged by factors within the complex maritime environment, like its wide variety of vessel scales, visual similarities, and extreme weather, making it complicated to balance model accuracy, speed, and efficiency [4], [5], [6]. This limitation is especially apparent in scenes with occluded or multi-scale ships, where models often prioritize larger vessels and fail to identify smaller ones, leading to missed detections and false positives [8]. Therefore, there is an urgent need to develop a more lightweight, adaptive, and accurate visual detection model designed for low-cost devices.

Modern ship detection uses deep learning, particularly Convolutional Neural Networks (CNNs), to automatically extract visual features from maritime images. This technology encompasses two main types: two-stage and one-stage detectors. Two-stage detectors, like the well-known Faster R-CNN, first identify potential object locations and then classify them. This method is very accurate but can be slow [10]. In contrast, one-stage detectors changed the game by doing everything at once. Models like YOLO [11] and SSD [12] predict an object's location and class in a single step, making them extremely fast and ideal for real-time use. For this research, the SeaShips dataset is used for training since it features various ship types and realistic sea conditions.

YOLO-FAST is a recent one-stage detector that achieves an effective balance between model size and accuracy, making it well-suited for tasks like port security and coastal monitoring [9]. To achieve this balance, it introduces FasterNet [13] as its primary feature extractor and incorporates the SimAM [14] attention module. This module enables the model to discriminate between important and unimportant features, enhancing its focus on relevant targets. The main motivation for using YOLO-FAST is to address the need for a lightweight ship detection model that can maintain reliable detection performance while reducing computational requirements. Its lightweight FasterNet backbone and SimAM attention mechanism make YOLO-FAST suitable for extracting relevant ship features with lower computational overhead, which is important for maritime surveillance applications on resource-constrained devices. This research contributes to the application of the YOLO-FAST model for detecting ships in complex maritime environments. The study focuses on the application of YOLO-FAST to evaluate its performance against varying weather conditions, diverse ship sizes, and complex backgrounds such as busy coastal areas.

II. METHODS

A. YOLO-FAST

The You Only Look Once (YOLO) series has risen to prominence in real-time object detection by providing a strong balance between accurate results and efficient performance. You Only Look Once-FAST (YOLO-FAST) is an object detection framework designed to achieve an optimal balance between inference speed and accuracy, making it ideal for real-time applications on edge devices.

Received: 11 August 2026, Revision: 26 September 2026, Accepted: 6 October 2026
available at: https://ejournal.unsrat.ac.id/v3/index.php/idea/

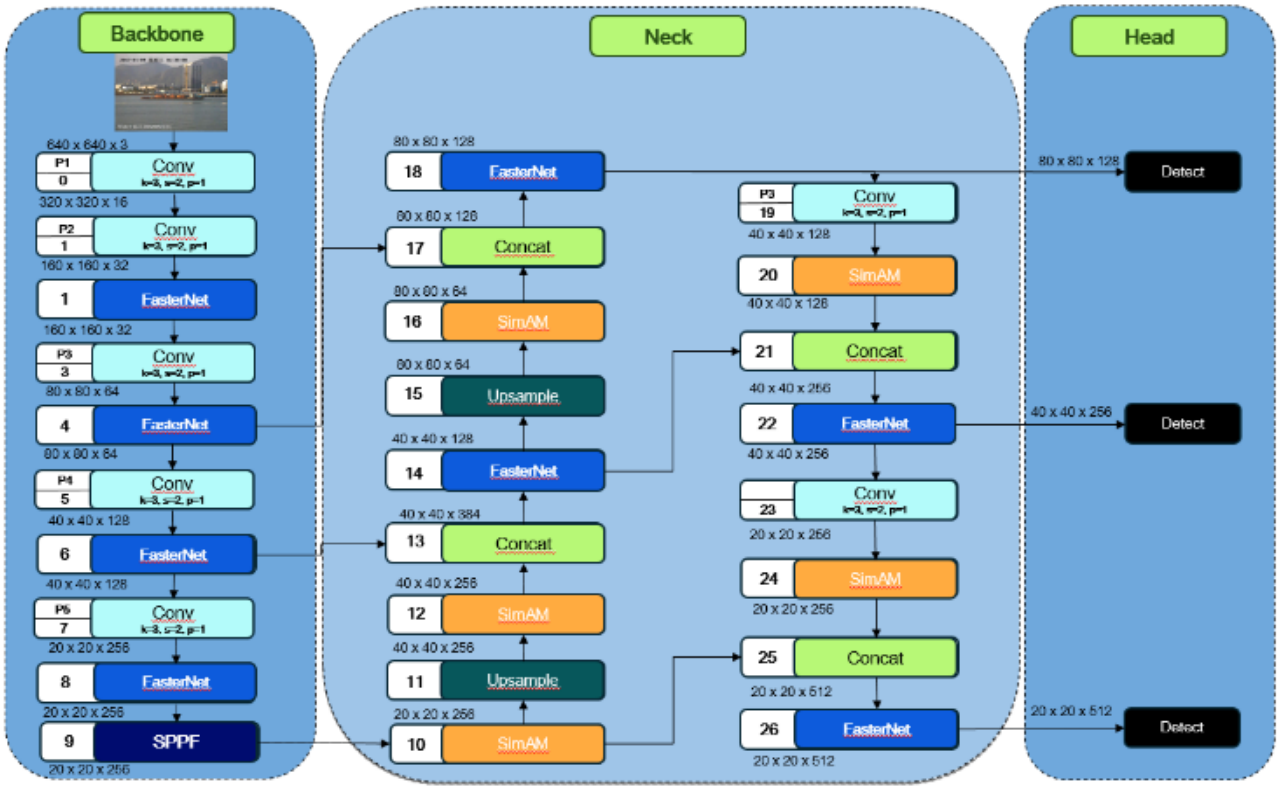

Fig. 1. The architecture of YOLO-FAST model, consisting of the Backbone for feature extraction, Neck for multi-scale feature fusion, and the Head for object detection

This architecture incorporates the efficient FasterNet backbone and SimAM attention mechanism to improve feature extraction without adding significant computational overhead. Unlike conventional YOLO architectures that often rely on heavier backbones, the YOLO-FAST approach prioritizes computational efficiency through the use of lightweight modules.

The backbone is responsible for extracting hierarchical features from the input image, the neck combines feature maps from different scales to enhance feature representation, and the detection head uses the fused features to generate the final class, bounding box, and confidence predictions. This design allows the model to process images in a single stage, combining region proposals and classification into a unified workflow for faster and more efficient inference. YOLO-FAST consists of three main components: backbone, neck, and detection head, as illustrated in Figure 1.

B. Backbone

In the YOLO-FAST architecture, FasterNet serves as the main feature extraction backbone that transforms the input image into hierarchical feature maps for subsequent processing by the neck. Its use of Partial Convolution reduces redundant computations while preserving spatial information, allowing the model to extract meaningful ship features with lower computational cost. Therefore, FasterNet is particularly suitable for YOLO-FAST because it supports efficient feature extraction while maintaining the lightweight design required for maritime ship detection on resource-constrained devices. This design significantly reduces the number of parameters and computational complexity (GFLOPs). It replaces the bottleneck with the faster block in C2f modules to reduce the number of operations while maintaining strong feature representation capabilities, which are essential for efficient and accurate object detection. The SPPF module, an enhanced version of spatial pyramid pooling, operates by applying pooling to feature maps at multiple scales. This process improves the model's robustness and its ability to adapt to targets of different sizes.

FasterNet is a highly efficient backbone designed to accelerate neural network processing. Its key innovation is the use of Partial Convolution (PConv), which selectively applies convolution only to a small portion of the input feature channels, while leaving the rest unchanged (identity mapping). This approach effectively reduces computational redundancy and memory access. An inverted residual block, located in the middle of the layer, follows the PConv layer. This block consists of two PWConv layers for channel expansion and a shortcut connection to reuse input features. Thus, FasterNet extracts rich spatial features at a much lower computational cost compared to standard convolution.

C. Neck

The neck plays a crucial role in connecting the backbone with the detection head by combining multi-scale feature representations. In this model, the neck utilizes a Path Aggregation Network [15] and Feature Pyramid Network [16], which generate three hierarchical merging paths to capture both bottom-up and top-down information. This process results in semantically rich feature maps for detecting objects of various sizes. The neck also includes the SimAM attention module, which further improves feature quality. In YOLO-FAST, SimAM helps the model focus on informative ship features by assigning higher attention to feature responses that are more distinctive from their surrounding features. This allows the network to emphasize relevant characteristics such as the shape, structure, and appearance of ships while reducing the influence of less informative background features.



Fig. 2. Detection Result On Seaships Dataset

SimAM (Simple, Parameter-Free Attention Module) is a lightweight attention mechanism that does not require additional convolution operations. Based on neuroscience principles, the most informative neurons typically exhibit firing patterns that are distinct from those of surrounding neurons. This process involves an energy function for each neuron based on the dissimilarity between the target neuron and the surrounding neurons in a feature map. Neurons with lower energy are considered to have more distinctive information and are assigned higher weighting. By deriving an analytical solution for this energy function, SimAM can efficiently compute 3D attention weights directly from the features themselves. Its ability to highlight informative features without adding parameters makes it an ideal choice for efficiency-focused architectures.

D. Head

The detection head is a critical part of the object detection network responsible for generating the final detection results, such as class labels, bounding boxes, and confidence scores. It employs a decoupled structure consisting of two distinct branches. This head consists of several parallel branches, where each branch has a 3x3 convolutional layer followed by a 1x1 convolutional layer for final feature extraction.

Three detection heads are operating at different feature map resolutions (e.g., 80x80, 40x40, and 20x20) to handle small, medium, and large object detection. The classification branch is responsible for predicting class probabilities for each anchor box, and the regression branch predicts the bounding box coordinates and confidence scores. The classification branch uses binary cross-entropy (BCE) loss, which is calculated from the actual and predicted class labels of each prior box to address class imbalance issues [20]. The regression branch employs a synthesis of Distribution Focal Loss (DFL) [17] and Complete Intersection over Union (CIoU) [18]. This enhances the model's sensitivity to difficult instances. As a result, it directs more attention to challenging predictions during training and reduces focus on easy-to-predict targets, which ultimately improves detection accuracy.

TABLE I. DATASET CONFIGURATIONS

Parameters	***Setup***
Training	1750 images
Validation	1750 images
Testing	3500 images

TABLE II. TRAINING AND TESTING CONFIGURATIONS

Parameters	***Setup***
Platform	Kaggle
GPU	P100
Image Size	640 × 640
Epochs	200
Batch Size	16
Optimizer	Stochastic Gradient Descent (SGD)
Learning Rate	0.01

E. Implementation Setup

- Dataset

The dataset used in this work is the Seaships dataset [19]. SeaShips is a ship detection dataset designed to support research in maritime object detection, with a particular focus on seagoing vessels. The images in this dataset originate from 10,080 real video segments captured by 156 surveillance cameras along the coast of Hengqin Island, China. Filtering out empty or redundant data yields the final dataset. This dataset consists of 31,455 precisely annotated, high-resolution images. It includes six main ship categories: ore carriers, bulk cargo carriers, general cargo ships, container ships, fishing boats, and passenger ships. Each image contains annotations for a bounding box and ship type.

TABLE III. PARISON OF YOLO VARIANTS SHOWING DETECTION ACCURACY AND EFFICIENCY TRADE-OFFS

Models	*Params*	*GFLOPs*	*mAP50*	*mAP50:95*
YOLOv5n	2.509.634	7.2	0.958	0.700
YOLOv6n	4.238.738	11.9	0.959	0.718
YOLOv8n	3.012.018	8.2	0.963	0.725
YOLOv10n	2.709.380	8.4	0.938	0.699
YOLO11n	2.591.010	6.4	0.960	0.705
YOLOv12n	2.539.266	6.0	0.959	0.699
YOLO-FAST	**2,306,818**	**6.4**	**0.946**	**0.669**

TABLE IV. INFERENCE PERFORMANCE COMPARISON OF YOLO VARIANTS, SHOWING MODEL SIZE, COMPUTATIONAL COST, AND REAL-TIME PROCESSING SPEED (FPS) ON RASPBERRY PI 5

Parameters	*Params*	*GFLOPs*	*FPS*
YOLOv5n	2.509.634	7.2	4.18
YOLOv6n	4.238.738	11.9	3.37
YOLOv8n	3.012.018	8.2	3.97
YOLOv10n	2.709.380	8.4	3.39
YOLO11n	2.591.010	6.4	4.04
YOLOv12n	2.539.266	6.0	3.50
YOLO-FAST	**2,306,818**	**6.4**	**4.25**

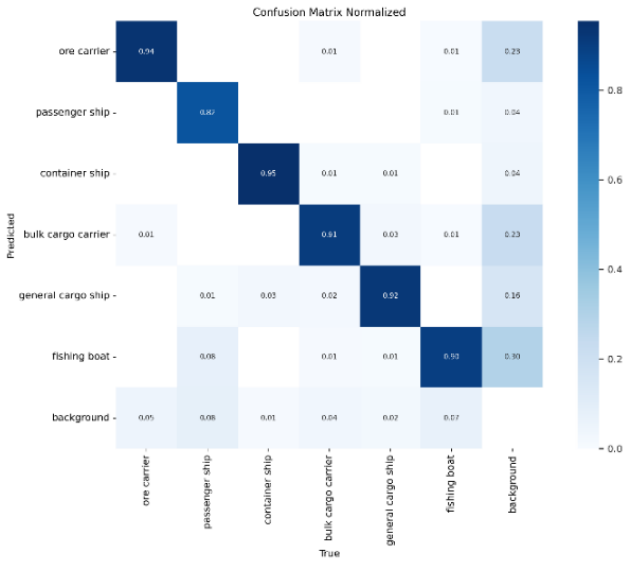

Fig. 3. Confusion Matrix on YOLO-FAST Model

The detailed configuration of the dataset is shown in Table 1. These six categories represent the different ship types annotated in the SeaShips dataset and provide diverse visual characteristics for evaluating the detection capability of YOLO-FAST. Their differences in shape, size, and appearance are relevant to maritime surveillance, where detecting and distinguishing various types of vessels is important for monitoring maritime traffic.

- Training and Testing Configuration

In this experiment, the training and testing of the YOLO-FAST model were performed using the configuration detailed in Table 2. The training was conducted on the Kaggle platform, utilizing an NVIDIA Tesla P100 GPU.

This hardware was selected for its high computing power, which is essential for efficient training and inference. An input image size of 640 × 640 pixels was set to balance spatial detail accuracy with computational efficiency. The model was trained for 200 epochs using a batch size of 16. This configuration was employed to ensure stable convergence and effective utilization of GPU memory. For optimization, establishes an excellent trade-off, prioritizing minimal computational cost and a small model footprint, making it an the Stochastic Gradient Descent (SGD) algorithm was selected, primarily due to its proven robustness and consistent convergence in object detection tasks. A learning rate of 0.01 was applied to enhance convergence, mitigate overfitting, and thereby improve the model's generalization performance.

III. RESULTS AND DISCUSSION

A. Evaluation on Dataset

The proposed model, YOLO-FAST, was evaluated using the mean Average Precision mAP metric, as detailed in Table 3. The evaluation focused on comparing its efficiency and performance against other lightweight detection systems from the YOLO family YOLOv5-Nano, YOLOv6-Nano, YOLOv8-Nano, YOLOv10-Nano, YOLO11-Nano, and YOLOv12-Nano. The quantitative results demonstrate that YOLO-FAST achieves its primary design goal of being the most lightweight and efficient model. It possesses only 2.31 million parameters, which is the lowest among all compared models, significantly fewer than competitors like YOLOv12-Nano 2.54M and YOLOv5-Nano 2.51M. In terms of performance, the proposed YOLO-FAST model achieved an mAP50:95 of 66.9% and an mAP50 of 94.6%. While the state-of-the-art YOLOv8-Nano achieves a higher accuracy 72.5% mAP50:95, it does so with a larger model 3.01 M Params. The proposed YOLO-FAST model, therefore, establishes an excellent trade-off, prioritizing minimal computational cost and a small model footprint, making it an ideal candidate for deployment on resource-constrained edge devices.

As shown in Figure 2, the qualitative detection results of the YOLO-FAST model on sample images from the Seaships dataset further confirm its capabilities. In general, the model is able to detect various ship types accurately, such as container ship 0.9 and bulk cargo carrier 0.7, even when the object is at a significant distance, partially obscured by environmental haze, or affected by water surface reflections. This demonstrates the model's ability to generalize and deal with variations in complex maritime surveillance conditions. However, some detection inconsistencies were still found. In some images, correctly identified ships like general cargo ship were detected with low confidence scores 0.4 and 0.3. These conditions indicate that although the model has achieved a high level of precision as supported by the mAP results, performance improvements are still needed so that detection confidence can be further improved, especially for challenging instances, and potential detection errors can be minimized in the future.

To further analyze its classification performance, the normalized confusion matrix is presented in Figure 3. The proposed YOLO-FAST model demonstrates a high true positive rate for ship-type predictions on the test dataset. It correctly identifies an ore carrier at 0.94, a passenger ship at 0.82, a container ship at 0.95, a bulk cargo carrier at 0.91, a general cargo ship at 0.92, and a fishing boat at 0.90. These results indicate that despite its low parameter count, the detection model can accurately distinguish the features of different ship types, enabling it to differentiate effectively between various large and small vessels. Among the six ship classes, the container ship achieved the highest detection percentage at 95%, followed by the ore carrier at 94%, general cargo ship at 92%, bulk cargo carrier at 91%, and fishing boat at 90%. The passenger ship recorded the lowest detection percentage at 82%. This variation indicates that the classification performance differs across the six ship categories, with the largest difference observed between the container ship and passenger ship classes.

B. Runtime Efficiency

Model efficiency refers to the detector's ability to be implemented in real-world applications. A key challenge is developing vision systems that can operate effectively on resource-constrained devices. The models were benchmarked on a Raspberry Pi 5 device to evaluate their practical performance 20. The proposed YOLO-FAST model demonstrates superior efficiency compared to other nano-scale models, including YOLOv5-Nano, YOLOv6-Nano, YOLOv8-Nano, YOLOv10-Nano, YOLO11-Nano, and YOLOv12-Nano, as detailed in Table 4. Furthermore, YOLO-FAST achieved the highest inference speed, reaching 4.25 FPS, which surpasses all other models in the comparison, including YOLOv5-Nano 4.18 FPS and YOLO11-Nano 4.04 FPS. In addition to its speed, the proposed model is the most lightweight, possessing only 2.31 million parameters.

This represents a significant parameter reduction of 7.9% compared to YOLOv5-Nano 2.51 M and 23.2% compared to YOLOv8-Nano 3.01 M. The model also maintains a low computational cost of 6.4 GFLOPs, which is 11.1% lighter than YOLOv5-Nano 7.2 GFLOPS and 21.9% lighter than YOLOv8-Nano 8.2 GFLOPS. This combination of the highest processing speed and the lowest parameter count demonstrates the success of the proposed architecture. These results indicate that the proposed YOLO-FAST model achieves an optimal balance of computational efficiency and high-speed performance, making it an ideal solution for real-time detection on edge devices.

IV. CONCLUSION

This study demonstrates the ability of the proposed YOLO-FAST architecture to perform real-time ship detection in complex maritime environments. This model achieves a mean Average Precision mAP50 of 94.6% and mAP50:95 of 66.9%, with only 2.31 million parameters and a computational complexity of 6.4 GFLOPs, demonstrating its ability to maintain an excellent balance between detection accuracy and computational efficiency. The proposed model demonstrates superior inference speed, reaching 4.25 FPS on a Raspberry Pi 5. This performance surpasses other variants, such as YOLOv5-Nano and the state-of-the-art YOLOv8-Nano, confirming its strong potential for real-time applications on resource-constrained devices.

Additionally, the experimental results show that YOLO-FAST effectively handles detection challenges by accurately distinguishing the features of different ship types. However, the model still produced some inaccuracies in detection. Therefore, further research can focus on enhancing the model's feature discrimination to improve precision. Overall, this research makes an essential contribution to the development of lightweight deep learning models for maritime monitoring. This work also provides a strong foundation for creating automated and efficient observation systems on edge devices.

REFERENCES

[1] L. Qian et al., "A new method of inland water ship trajectory prediction based on long short-term memory network optimized by genetic algorithm," Appl. Sci., vol. 12, no. 8, art. 4073, 2022.

[2] Namgung, H.; Kim, J.S. Collision risk inference system for maritime autonomous surface ships using COLREGs rules compliant collision avoidance. IEEE Access 2021, 9, 7823–7835.

[3] Vagale, A.; Oucheikh, R.; Bye, R.T.; Osen, O.L.; Fossen, T.I. Path planning and collision avoidance for autonomous surface vehicles I: A review. J. Mar. Sci. Technol. 2021, 26 , 1292–1306.

[4] M. Zhu et al., "YOLO-HPSD: A high-precision ship target detection model based on YOLOv10," PLOS ONE, vol. 20, no. 1, e0321863, Jan. 2025.

[5] L. Shen et al., "YOLO-LPSS: A Lightweight and Precise Detection Model for Small Sea Ships," J. Mar. Sci. Eng., vol. 13, no. 2, pp. 115–127, Feb. 2025.

[6] Y. Li and S. Wang, "EGM-YOLOv8: A Lightweight Ship Detection Model with Efficient Global–Local Feature Fusion and Attention," Proc. Int. Conf. Image Process., Oct. 2025, pp. 543–550.

[7] Namgung, H. Local route planning for collision avoidance of maritime autonomous surface ships in compliance with COLREGs rules. Sustainability 2021, 14, 198. [CrossRef]

[8] [Vagale, A.; Bye, R.T.; Oucheikh, R.; Osen, O.L.; Fossen, T.I. Path planning and collision avoidance for autonomous surface vehicles II: A comparative study of algorithms. J. Mar. Sci. Technol. 2021, 26, 1307–1323.

[9] Song, Z., Zhang, X., & Tan, P. (2025). YOLO-Fast: a lightweight object detection model for edge devices. Journal of Supercomputing, 81(5). https://doi.org/10.1007/s11227-025-07172-3

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems (NIPS), 2015.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in European Conference on Computer Vision (ECCV), 2016.

[13] J. Chen, et al., "Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[14] Y. Yang, et al., "SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks," in International Conference on Machine Learning (ICML), 2021.

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

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

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

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

[19] Shao, Z., Wu, W., Wang, Z., Du, W., & Li, C. (2018). SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. IEEE Transactions on Multimedia, 20(10), 2593–2604.

[20] A. U. Ruby, P. Theerthagiri, I. J. Jacob, and Y. Vamsidhar, "Binary Cross Entropy with Deep Learning Technique for Image Classification," International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 4, pp. 5393–5397, 2020. doi: 10.30534/ijatcse/2020/17