remote sensing

MDPI

Article

Edge-Optimized Lightweight YOLO for Real-Time SAR Object Detection

Caiguang Zhang [1,*], Ruofeng Yu [2], Shuwen Wang [1], Fatong Zhang [1], Shaojia Ge [1], Shuangshuang Li [1] and Xuezhou Zhao [1]

[1] Shanghai Radio Equipment Research Institute, Shanghai 201109, China; 13817508033@139.com (S.W.); zft38768@163.com (F.Z.); geshaojia@foxmail.com (S.G.); 15821261925@163.com (S.L.); zhao_1048@163.com (X.Z.)
[2] College of Electronic Science, National University of Defense Technology, Changsha 410073, China; yuruofeng17@nudt.edu.cn
* Correspondence: zhangcaiguang17@nudt.edu.cn

Abstract

Synthetic Aperture Radar image object detection holds significant application value in both military and civilian domains. However, existing deep learning-based methods suffer from excessive model parameters and high computational costs, making them impractical for real-time deployment on edge computing platforms. To address these challenges, this paper proposes a lightweight SAR object detection method optimized for edge devices. First, we design an efficient backbone network based on inverted residual blocks and the information bottleneck principle, achieving an optimal balance between feature extraction capability and computational resource consumption. Then, a Fast Feature Pyramid Network is constructed to enable efficient multi-scale feature fusion. Finally, we propose a decoupled network-in-network Head, which significantly reduces the computational overhead while maintaining detection accuracy. Experimental results demonstrate that the proposed method achieves comparable detection performance to state-of-the-art YOLO variants while drastically reducing computational complexity (4.4 GFLOP) and parameter count (1.9 M). On edge platforms (Jetson TX2 and Huawei Atlas DK 310), the model achieves real-time inference speeds of 34.2 FPS and 30.7 FPS, respectively, proving its suitability for resource-constrained, real-time SAR object detection scenarios.

Keywords: lightweight object detection; SAR; edge device; YOLO; real-time processing

check for updates

Academic Editor: Dusan Gleich

Received: 12 May 2025
Revised: 9 June 2025
Accepted: 17 June 2025
Published: 24 June 2025

Citation: Zhang, C.; Yu, R.; Wang, S.; Zhang, F.; Ge, S.; Li, S.; Zhao, X. Edge-Optimized Lightweight YOLO for Real-Time SAR Object Detection. *Remote Sens.* **2025**, *17*, 2168. https://doi.org/10.3390/rs17132168

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Synthetic Aperture Radar (SAR), as an important means of remote sensing, is widely used in many fields such as military reconnaissance, ocean monitoring, disaster assessment, etc., because of its ability to work all-weather and all-day without being limited by weather conditions. With the rapid development of SAR imaging technology of drones, satellites and other platforms, a huge amount of SAR image data is constantly generated, and how to efficiently and accurately extract useful information from these data has become an urgent problem to be solved. Object detection in SAR images is one of the key technologies that significantly enhances the practical value of SAR systems. In recent years, deep learning-based object detection methods have provided new approaches to SAR image interpretation. The existing research primarily follows two paradigms: CNN-based detection methods [1–3] and Transformer-based methods [4–6].

Recent advancements in CNN-based SAR object detection showcase substantial progress driven by innovations addressing core challenges, with research systematically evolving along several interconnected directions. To mitigate the critical issue of domain shifts between SAR and optical data or across different SAR sensors, cross-domain adaptation and knowledge transfer have emerged as pivotal strategies: Zhou et al. [7] leverage the latent similarities between optical and SAR images for domain-adaptive few-shot ship detection, while Luo et al. [8] propose a semi-supervised framework fusing optical and limited SAR data to counteract shifts at image, instance, and feature levels. Huang et al. [9] further reduce reliance on annotated SAR data through pixel-instance information transfer, and Chen et al. [10] achieve unsupervised detection via cross-modal distribution alignment and pseudo-label refinement. Concurrently, overcoming the challenges posed by multi-scale objects and small targets has spurred significant innovation in feature representation. Sun et al. [11] introduce a multi-scale dynamic feature fusion network enhanced by MSLK-Blocks, DFF-Blocks, and GPD loss for superior elliptical ship detection, and Zhang et al. [12] employ a dynamic feature discrimination module (DFDM) and center-aware calibration module (CACM) with bidirectional aggregation to counter cross-sensor degradation. Liu et al. [13] develop a Subpixel Feature Pyramid Network (SFPN) preserving fine details. Furthermore, strategies to amplify discriminative features prominently utilize attention mechanisms. Ma et al. [14] integrate bidirectional coordinate attention and multi-resolution fusion into CenterNet for small ship detection, and Chen et al. [15] apply Incentive Attention Feature Fusion (IAFF) to align scattering characteristics with semantics. In response to data scarcity and open-world complexities, few-shot, open-set, and semi-supervised methodologies are advancing. Zhao et al. [16] utilize context-aware Gaussian flow representations to alleviate foreground–background imbalance, and Xiao et al. [17] enhance open-set generalization via global context modeling. Semi-supervised approaches, such as Chen et al.'s [18] teacher–student framework (SMDC-SSOD) with cross-scale consistency and Yang et al.'s [19] fuzzy-evaluation guided adaptive data selection effectively minimize annotation burdens. These interrelated advances collectively demonstrate developments in the field in terms of robust domain adaptation, complex multi-scale modeling, and efficient learning paradigms, driving higher accuracy in SAR target detection in complex scenarios.

Transformer-based methods achieve remarkable breakthroughs in SAR object detection, with innovations systematically progressing across three dimensions—model architecture, noise robustness, and semantic optimization—collectively enhancing accuracy for objects in complex scenarios. In architectural design, hybrid approaches dominate to leverage complementary strengths. Yang et al. [20] combine CNNs' local feature extraction with Swin Transformer's global modeling for multi-scale ship detection, and Li et al. [21] propose GL-DETR, adopting global-to-local processing to boost small-object accuracy. Zhou et al. [22] enhance YOLOv7 with Swin Transformer through PS-FPN modules and mix-attention strategies, while multi-scale frameworks by Chen et al. [23] and Feng et al. [24] significantly improve cross-scale fusion and orientation adaptability through window-based Swin Transformers and rotation-aware DETR modules. To address the critical challenge of speckle noise and data scarcity, advanced denoising and adaptation strategies have emerged. Liu et al. [25] develop MD-DETR with multi-level denoising to suppress noise interference, and Chen et al. [23] integrate embedded denoising modules for low-SNR robustness. Cross-domain gaps are mitigated by semi-supervised frameworks leveraging optical-to-SAR transfer [8] and domain-adaptive Transformers with pseudo-label refinement [26]. For handling geometric variations and semantic ambiguity, innovative attention mechanisms provide solutions. Lin et al. [27] incorporate deformable attention modules to address shape-scale diversity in ships, while Chen et al. [28] propose anchor-free SAD-Det

using adaptive Transformer features to minimize background interference in oriented detection. Future directions emphasize leveraging large-scale pre-trained models [29] and datasets (e.g., SARDet-100K [30]) to enhance generalization and scalability. Collectively, Transformer-based SAR detection demonstrates transformative potential through synergistic architectural innovation, noise-robust modeling, and geometry-aware optimization, establishing new paradigms for handling real-world complexity while pointing toward efficient, generalizable solutions.

Although modern object detection algorithms are being designed with increasing sophistication and substantially higher accuracy, their GPU-oriented architectures typically involve excessive parameterization (e.g., >100 M trainable weights) and prohibitive computational complexity (often exceeding 200 GB Floating-point Operations (GFLOPs) per inference), rendering them impractical for resource-constrained platforms. This limitation becomes particularly critical in SAR image analysis scenarios such as military reconnaissance and disaster assessment, where latency-sensitive object detection must be performed on embedded systems with strict power budgets (<10 W), including drones, nanosatellites, and portable devices. The inference latency of traditional deep models on edge TPUs is greater than 500 milliseconds, which cannot meet the real-time processing requirements. Therefore, it is urgent to research and develop lightweight SAR object detection algorithms, which can not only improve processing speed and energy efficiency but also better adapt to the needs of embedded platforms. The question of how to achieve a lightweight model while ensuring detection performance has become one of the key challenges in the field of SAR image object detection.

Recent lightweight studies focus on optimizing existing frameworks [31–33]. Gao et al. [34] design a lightweight network for multi-band SAR vehicle detection to improve accuracy in complex scenarios, and Wang et al. [35] propose SAFN, combining attention-guided fusion and lightweight modules (LGA-FasterNet and DA-BFNet) to reduce parameters while enhancing ship detection. Huo et al. [36] introduce GSE-ships, integrating GhostMSENet and ECIoU loss to lower computational burden by 7.97% while boosting accuracy, and Guan et al. [37] enhances YOLOv8 with a Shuffle Re-parameterization module and hybrid attention to improve small SAR ship detection. Knowledge distillation techniques leverage to bridge performance gaps between large and lightweight models. Chen et al. [38] propose DPKD for precise localization knowledge transfer, reducing the performance gap in SAR detection, and Zhang et al. [39] incorporate attention selection modules and lightweight multi-scale pyramids into RTMDet, enhancing nearshore ship detection. Attention mechanisms, such as the Multi-Scale Coordinate Attention in MSSD-Net [40] and Shuffle Attention in SHIP-YOLO [41], are critical in balancing feature extraction and computational costs. FPGA-based implementations and edge-device adaptations accelerate real-time SAR detection. Huang et al. [42] deploy lightweight SAR ship detection on FPGAs, significantly reducing processing time, and Fang et al. [43] propose a hybrid FPGA-YOLOv5s architecture for low-power, high-precision small object detection. Addressing scale and orientation variability, Meng et al. [44] propose LSR-Det with adaptive feature pyramids and lightweight rotation heads for efficient multi-scale ship detection, and Tian et al. [45] achieve 98.8% mAP on SSDD using YOLO-MSD, which integrates DPK-Net and BSAM modules for multi-scale adaptability. Zhu et al. [46] introduce DSENet for weak-supervised arbitrary-oriented detection, eliminating reliance on rotated bounding box annotations. Integrated applications demonstrate practical viability. Yasir et al.'s [47] YOLOv8n-based tracker with knowledge distillation and C-BIoU tracking, and Luo et al.'s [41] SHIP-YOLO with ghost convolutions and Wise-IoU, enhance real-time maritime monitoring. Rahman et al. [48] optimize lightweight CNNs for SAR classification in resource-limited environments, while Zhang et al. [49] deploy APDNet for edge-based

person detection with minimal accuracy loss. These advancements collectively address the trade-off between detection accuracy and computational efficiency, enabling practical deployment in satellite, UAV, and edge-device scenarios while maintaining robustness against SAR-specific challenges such as noise, scale variation, and complex backgrounds.

However, these approaches exhibit notable limitations in real-time SAR image interpretation: (1) The lightweight architecture design (e.g., channel compression and depth reduction) degrades feature map resolution and weakens cross-channel information interaction, leading to insufficient feature representation in complex scenarios (e.g., dense objects and multi-scale interference). (2) Existing methods primarily optimize inference speed for GPU platforms, whereas their deployment efficiency on edge devices remains suboptimal due to limited computational parallelism and scarce on-chip memory resources.

Therefore, it is necessary to design a lightweight model that balances real-time performance and object detection accuracy while being easy to deploy on edge platforms. This paper addresses the SAR image object detection problem from a more balanced perspective, significantly reducing the number of parameters and improving detection accuracy. The main contributions of this paper are summarized as follows:

1. The efficient backbone feature extraction network leverages inverted bottlenecks and information bottlenecks, achieving an optimal balance between feature extraction capability and computational resource consumption. This design provides enhanced flexibility for real-time SAR image-processing tasks.

2. The core idea of the proposed fast pyramid architecture network is to facilitate rapid multi-scale feature sharing, aiming to accelerate the detection process by reducing the demand for computational resources. By optimizing the information flow and minimizing intermediate computations, P4 is adopted as the primary fusion node to efficiently aggregate complementary information from both low-level and high-level layers, enhancing the feature representation capability in complex backgrounds.

3. We introduce a decoupled network-in-network detection head, specifically designed to provide fast and lightweight computations required for classification and regression tasks. This design enhances the model's expressive power while effectively alleviating computational burdens, ensuring robust detection performance even in resource-constrained environments.

2. Materials and Method

2.1. Dataset

SARDet-100K [30] is a large-scale multi-category Synthetic Aperture Radar object detection dataset jointly created by Nankai University and the National University of Defense Technology, designed to match the widely adopted general object detection benchmark COCO. This dataset integrates and standardizes 10 existing SAR detection datasets from institutions in China, Europe, and the United States, including SAR-AIRcraft [50] for aircraft detection, HRSID [51] for ship recognition, and SSDD [52] for maritime surveillance. After rigorous quality checks and annotation validation, the final dataset comprises 116,598 images containing 245,653 annotated instances, comparable in scale to the COCO dataset. High-resolution images in this dataset are uniformly cropped into 512 × 512 patches with a 200-pixel overlap to avoid target truncation. The dataset is divided into training (94,493 images), validation (10,492 images), and test (11,613 images) sets in an 8:1:1 ratio.

The dataset covers six key object categories with the following distribution: aircraft (21% of total instances), ships (47%), vehicles (5%), bridges (14%), tanks (11%), and harbors (2%). These instances exhibit significant diversity in scale and imaging conditions, as they originate from multiple SAR platforms, including satellite systems, airborne radar, and ground-based radar systems. The spatial resolution ranges from 0.1 m for high-

resolution close-range observations to 25 m for low-resolution wide-area surveillance (e.g., the smallest ship is only 9 pixels wide). The dataset supports multiple polarization modes such as HH, HV, VH, and VV, posing substantial challenges for detection algorithms in handling target appearance variations, complex background clutter, and the inherent speckle noise characteristics of SAR imagery.

2.2. Method

2.2.1. Overview of Proposed Architecture

YOLOv8 [53] advances the YOLOv5 [32] architecture with key improvements throughout its design, offering five scalable variants to accommodate different computational constraints and performance needs. The architecture follows a streamlined four-stage pipeline: First, the input stage enhances robustness through Mosaic data augmentation, which stitches multiple training images to improve generalization. The backbone network builds upon Cross Stage Partial Darknet (CSPDarknet) with BatchNorm-SiLU convolutions for efficient feature extraction, incorporates YOLOv7-inspired gradient flow optimizations, and integrates an Spatial Pyramid Pooling-Fast (SPPF) module for multi-scale feature fusion. For hierarchical feature aggregation, the neck employs a bidirectional feature pyramid network (BiFPN) that combines top–down and bottom–up pathways from Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) structures, enabling richer semantic fusion. Finally, YOLOv8 adopts anchor-free and detection head decoupling design to separate the classification task from the regression task, which reduces the interference between the two tasks and improves the model's adaptability to complex scenes.

As shown in Figure 1, our proposed method is based on the YOLOv8 architecture and involves three aspects of lightweight optimization, aiming to optimize edge deployment performance while maintaining detection accuracy. First, the backbone network processes the 640 × 640 × 3 input image through an optimized STEM layer, which combines standard convolution and pointwise convolution to reduce the spatial dimensions to 160 × 160 × 16 with minimal channel expansion. This is followed by efficient inverted bottleneck layers and downsampling for feature extraction. Second, the neck architecture significantly reduces the computational overhead by simplifying the PANet design. It uses P4 as the primary fusion node to aggregate multi-scale features and replaces standard convolutions with depthwise separable convolutions. This enables computationally efficient bidirectional (top–down and bottom–up) multi-scale feature fusion. Third, the detection head adopts a decoupled design and incorporates a network-in-network structure, constructing parallel branches for classification and localization. This achieves a substantial reduction in computational cost while preserving detection accuracy. Collectively, these architectural improvements deliver a significant reduction in computational cost and model parameters while maintaining highly competitive detection accuracy, making the proposed method particularly suitable for resource-constrained edge devices.

2.2.2. Efficient Backbone Feature Extractor

A series of previous studies have shown that optimizing the number of channels can effectively reduce the computational requirements, especially for large-size spatial feature maps. The proposed method designs the backbone network based on the inverted residual blocks in MobileNetV2, reducing computational costs while maintaining the model's feature representation capability.

The STEM component in modern object detectors serves as the foundational processing stage that handles raw image inputs (typically 3-channel RGB data). Its primary purpose is twofold: (1) rapidly downscale spatial dimensions to reduce the computational overhead, and (2) expand the channel capacity for richer feature representation. Our

proposed method employs a carefully balanced stem structure that combines pointwise and standard convolutions with constrained channel counts. This design progressively reduces spatial dimensions while maintaining computational leanness as shown in Table 1. Its features include (1) using depthwise separable convolutions (DWConv) for spatial processing, (2) maintaining consistent channel width (16 channels) throughout, and (3) employing SiLU activation for nonlinearity. This configuration achieves a 4× spatial reduction (640→160) while keeping computations to 0.32 GFLOP, comparable to the most efficient YOLO variants.

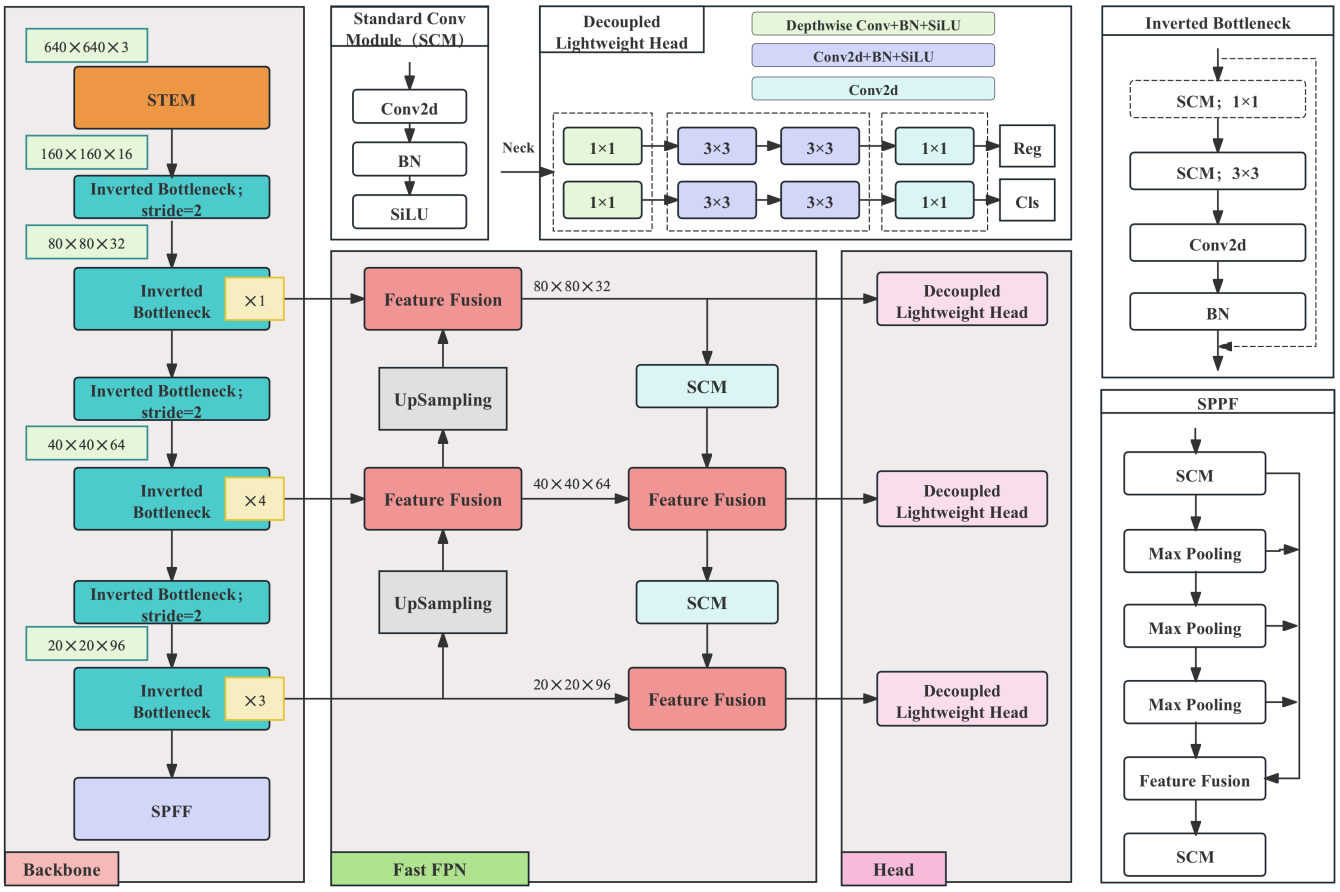


Figure 1. Network structure of the proposed method. The gray background indicates the network structure, and the transparent background indicates the key components. The entire network consists of three core parts: a redesigned backbone utilizing inverted bottleneck layers, a more efficient feature fusion structure, and a lightweight decoupled detection head.

Table 1. Stem architecture specifications for proposed method.

Block Composition	Output Channels	Kernel Size	Stride	Output Dimension
Conv + BN + SiLU	16	3 × 3	2	320 × 320
Conv + BN + SiLU	16	1 × 1	1	320 × 320
DWConv + BN + SiLU	16	3 × 3	2	160 × 160
Conv + BN	16	1 × 1	1	160 × 160

Meanwhile, the proposed method introduces dynamic improvements based on the inverted bottleneck structure, incorporating a channel optimization mechanism and conditional computation paths to significantly enhance computational efficiency while preserving lightweight advantages. Traditional inverted bottlenecks employ an "expand–compress" strategy: first expanding the channel dimension (e.g., with an expansion ratio of 6×) via pointwise convolution, then extracting spatial features through depthwise convolution, and finally

compressing back to the target channels. However, we observe that when the input channels (C_{in}) match the intermediate expanded channels (C_{mid}) (e.g., with an expansion ratio of 1 or when feature concatenation directly satisfies the channel requirement), the initial pointwise convolution can be dynamically skipped, proceeding directly to depthwise convolution and subsequent compression. This optimization reduces redundant computations, particularly beneficial for large-scale feature maps (e.g., shallow layers in detection tasks), achieving approximately 15% FLOP reduction in experiments. Additionally, the model enforces residual connections to stabilize gradient propagation, ensuring feature fusion via skip connections even when input and output channels are identical, thereby mitigating information loss. Compared to classical bottleneck structures (Figure 2a) and traditional inverted bottlenecks (Figure 2b), the proposed method achieves "computation-on-demand" through flexible conditional judgments. This design balances lightweight and accuracy in object detection tasks, offering a novel approach for real-time mobile detection.

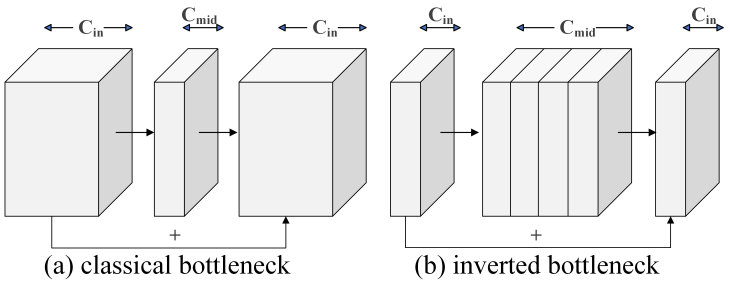


Figure 2. Classical bottleneck structure and inverted bottleneck structure.

Based on the information bottleneck theory [54], the proposed method addresses inter-layer information degradation in deep neural networks (DNNs) through dynamic channel expansion mechanisms and hierarchical information flow control. Traditional DNNs, due to their non-strict Markov chain property ($X \rightarrow \tilde{X} \rightarrow Y$), tend to lose critical features during layer-wise processing ($h_{i-1} \rightarrow h_i$), where $I(Y;h_i) \geq I(Y;h_{i+j})$ indicates the mutual information decay. To mitigate this, we establish a dual-constrained channel expansion strategy: (1) Global channel counts satisfy $Ch_i \in [Ch_1, Ch_n]$ with $Ch_n/Ch_1 \leq 6$, limiting cross-layer information exchange $I(h_1;h_n)$; (2) Differentiated expansion ratios for different layers: 3× for base layers, increased to 6× during cross-layer propagation ($P_i \rightarrow P_{i+1}$), while applying 9× expansion for high-semantic layers ($P_4 \rightarrow P_5$) to boost mutual information $I(Y;h_i)$. Compared to YOLOv3-v9's fixed-layer stacking, our approach reduces P_3 layer repetitions to 2, combining residual cross-layer connections to minimize $I(h_{i-1};h_i)$ loss while avoiding a dense connection overhead. This design integrates inverted bottlenecks with spatial compression via pointwise convolutions (total 6× expansion at stride = 2), achieving optimal balance between computational complexity and feature extraction capability.

2.2.3. Fast Feature Pyramid Network

The proposed method designs a more lightweight feature fusion method. As shown in Figure 3, the Fast Feature Pyramid Network (FFPN) structure simplifies the traditional PANet design by reducing convolutional layers and channel numbers while preserving hierarchical feature integration capabilities. It employs P4 as the primary fusion node, which efficiently aggregates complementary information from P3 (fine-grained details) and P5 (high-level semantics) through computationally optimized paths. The proposed method leverages the inherent compatibility of backbone output channels (e.g., $C_{out}P_4 + C_{in}P_3 = C_{mid}P_3$) to eliminate redundant pointwise convolution operations. Compared to the traditional PANet, the Fast Feature Pyramid Network maintains the detection performance for multi-scale

targets by optimizing the information flow and reducing the intermediate computation while greatly reducing the neck computation cost.

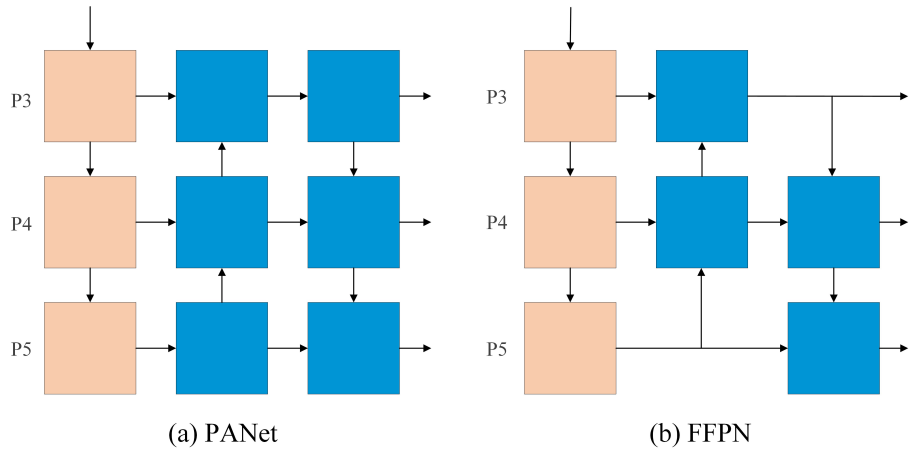


Figure 3. Fast Feature Pyramid Network structure.

Simultaneously, the proposed method employs depthwise separable convolutions to replace standard convolutions, dramatically reducing parameter counts. Specifically, the parameter count for standard convolution is

$$P_{\text{std}} = M \times N \times 3 \times 3 \tag{1}$$

whereas depthwise separable convolution decomposes it into (1) depthwise convolution (per-channel 3×3 convolution) with parameters

$$P_{\text{depth}} = M \times 3 \times 3 \tag{2}$$

and (2) pointwise convolution (1×1 convolution) with parameters

$$P_{\text{point}} = M \times N \tag{3}$$

The total parameter reduction ratio becomes

$$\frac{P_{\text{sep}}}{P_{\text{std}}} = \frac{1}{N} + \frac{1}{9} \tag{4}$$

For example, when the output channel dimension $N = 256$, the parameters can be reduced to approximately 11.1% of the standard convolution. This optimization is based on the low-rank decomposition principle of convolutional kernels, which maintains the spatial feature extraction capability while significantly reducing computational complexity and memory dependencies.

2.2.4. Decoupled Lightweight Head

Since YOLOv6 [55], the detection head has evolved into a more powerful dual branch (independent classification and object regression branches). Although this design demonstrates significant efficiency, its computational cost nearly doubles due to the requirement of performing separate convolutional operations for classification and detection.

Based on the pixel-wise grid operation characteristic of YOLO, the proposed method introduces network-in-network employing pointwise convolutions as illustrated in Figure 1, "Head". This architecture adopts a dual-path pointwise convolution: the classification network and the regression network utilize independent pointwise operations. In object

detection tasks, pointwise convolutions essentially function as pixel-wise classifiers and regressors, with their network-in-network property being crucial. The core value of the network-in-network structure lies in its use of nested micro-networks (such as 1 × 1 convolutions) to achieve cross-channel feature fusion and nonlinear enhancement, significantly improving the model's representational capacity. Simultaneously, by leveraging channel compression techniques, it substantially reduces the computational costs and parameter counts. At the cost of a slight increase in structural complexity, it markedly optimizes the overall model efficiency, thereby achieving an effective balance between detection accuracy and computational resource consumption. This makes network-in-network a key design for lightweight, high-precision models. By introducing 1×1 convolutions after traditional convolutional layers (equivalent to performing fully connected operations along the spatial dimension) the model enhances nonlinear interactions between channels, strengthens nonlinear modeling capabilities within local receptive fields, and improves performance in both classification and detection tasks.

The decoupled head employs an independent dual-path pointwise convolution design: the classification branch and regression branch specialize in their respective tasks. It expands channel dimensions via pointwise convolutions to match the number of classes while maintaining the spatial resolution of feature map P_i, enabling each pixel to carry a predictive potential. Additionally, 1×1 convolutions reduce channel dimensionality, decreasing the computational overhead. Furthermore, depthwise convolutions are decomposed into two 3×3 convolutions, effectively lowering overall computational costs compared to a single 5×5 convolution. The proposed head achieves a 50% reduction in computational cost compared to the traditional YOLOv6 detection head without compromising detection accuracy. This design philosophy aligns closely with recent advancements in lightweight network architectures, demonstrating superior hardware compatibility for edge deployment scenarios.

2.2.5. Loss

The proposed method introduces significant improvements in both classification and localization loss components. The overall loss function combines these terms with task-balancing coefficients [53]:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{box}}(\mathcal{L}_{\text{CIoU}} + \mathcal{L}_{\text{DFL}}) \tag{5}$$

where λ_{cls} and λ_{box} are balancing hyper-parameters empirically set to 0.5 and 7.5, respectively.

The classification branch adopts Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}[y_{i,c}\log(p_{i,c}) + (1 - y_{i,c})\log(1 - p_{i,c})] \tag{6}$$

Here, N is the number of sampled anchors, C is the total categories, $y_{i,c}$ is the ground-truth label (0/1) for class c at anchor i, and $p_{i,c}$ is the predicted probability for class c at anchor i. The Task Alignment Learning (TAL) [56] dynamically adjusts positive sample selection:

$$t = s^{\alpha} \times u^{\beta} \tag{7}$$

where s is the classification score, u is the Intersection over Union (IoU) between the prediction and ground truth, and α, β are balancing exponents (default: $\alpha = 0.5$, $\beta = 6.0$).

The localization loss $\mathcal{L}_{\text{CIoU}}$ [57] combines two components:

$$\mathcal{L}_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v \tag{8}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{9}$$

where $\rho(\cdot)$ denotes the Euclidean distance, c is the minimal enclosing diagonal, and $\alpha = v/(1 - \text{IoU} + v)$. The Distribution Focal Loss (DFL) [53] is defined as

$$\mathcal{L}_{\text{DFL}} = -\sum_{j\in\{y_i, y_{i+1}\}} |y - j| \log(\mathcal{P}(j)) \tag{10}$$

Here, y is the normalized ground-truth coordinate ($y = l_{\text{true}}/s_{\text{downsample}}$), $\mathcal{P}(j)$ is the predicted probability at discrete position j, and $y_i = \lfloor y \rfloor$, $y_{i+1} = \lceil y \rceil$. The localization target conversion follows:

$$\mathcal{T}(y_i) = y_{i+1} - y, \quad \mathcal{T}(y_{i+1}) = y - y_i \tag{11}$$

TAL can dynamically select the top-k predictions using the t-score and applies multi-scale supervision with feature strides of 8, 16, and 32 pixels. The loss function unifies classification and objectness tasks into a single BCE formulation, improving training efficiency and gradient flow. The combination of DFL and CIoU enhances the bounding box precision, while TAL adapts to varying object scales and aspect ratios. The anchor-free design simplifies implementation without sacrificing recall, making the loss robust for challenging cases like occlusions and small objects.

3. Results

To ensure fair evaluation, all experiments in this study utilize the training set for model optimization and the validation set for performance assessment. All of the following experimental results are from the validation set.

3.1. Experimental Setup

3.1.1. Implementation Details

Our experimental framework employs carefully configured hardware to ensure reproducible results across different computing platforms. For model training, we use a workstation equipped with a NVIDIA RTX 4060Ti GPU (16GB VRAM), Intel i5-13490k CPU, and 32 GB RAM running Windows 10. To evaluate real-world deployment scenarios, we conduct comparative testing on two edge computing platforms: (1) The NVIDIA Jetson TX2 featuring a 256-core GPU (1.33 TFLOPS FP16), 6-core hybrid CPU (dual-core Denver 2 + quad-core Cortex-A57), and 4 GB LPDDR4 memory with 7.5 W power consumption; and (2) The Huawei Atlas DK 310 with its Ascend 310 AI processor (12 nm process, Da Vinci architecture) delivering 16 TOPS at INT8 and 8 TFLOPS at FP16 through dual AI Cores, complemented by 8GB LPDDR4X memory (51.2 GB/s bandwidth) and 16 TaishanV200M CPU cores (1.9 GHz). Both edge platforms demonstrate remarkable energy efficiency, with the Atlas DK 310 maintaining just 8 W typical power consumption while offering dedicated hardware video codecs for accelerated multimedia processing. This multi-platform approach enables comprehensive performance benchmarking across desktop and embedded environments.

The training protocol employs Stochastic Gradient Descent (SGD) optimization with a momentum of 0.9 and weight decay of 0.001. The learning rate is initialized at 0.01 and follows a cosine annealing schedule to dynamically adjust during training. We maintain a fixed batch size of 16 to balance memory constraints and convergence stability. All input images are resized to 640 $\times$ 640 pixels and processed with standard augmentation techniques including random flipping and mosaic augmentation. The models are trained for 300 epochs to ensure full convergence, with periodic validation checks to training progress.

3.1.2. Evaluation Metrics

Our comprehensive evaluation framework assesses model performance across three aspects. First, all the experiments in this paper adopt FLOP as the main evaluation metric of model computational efficiency. By counting the total number of multiply–add operations required by the neural network to complete the forward propagation, this metric provides a hardware-independent benchmark for the comparison of the computational complexity of different models, and can reflect the computational characteristics of the model more objectively. In addition, the detection accuracy is evaluated using two standard metrics: mAP50 (mean average precision at IoU threshold 0.5) and mAP50-95 (average precision across IoU thresholds from 0.5 to 0.95 in 0.05 increments). Finally, the deployment efficiency is measured through frames per second (FPSs) performance on the Jetson TX2 and Ascend 310 under FP16 precision, providing practical reference for real-world applications.

3.2. Ablation Studies

3.2.1. Experiments on Feature Extraction Backbones

The backbone network of the proposed method is shown in Table 2. Although the classical inverted residual architecture is less used in current object detection tasks, its design concept is still informative. The proposed method employs an inverted residual architecture with a spanning distance greater than 1 during the feature layer transformation, which optimizes the computational cost by performing the final channel expansion (with an expansion ratio of only 3 instead of the traditional 6) prior to the deep convolution. This improved spanning strategy greatly reduces the overall computational burden while maintaining accuracy.

Table 2. Proposed backbone network architecture.

Input	Operation	Extended Layer	Output	Stride
P0	Conv: 3 × 3	-	16	2
P1	Conv: 1 × 1	16	16	1
P1	Inverted Bottleneck: 3 × 3	16	16	2
P2	Inverted Bottleneck: 3 × 3	96	32	2
P3	Inverted Bottleneck: 3 × 3	96	32	1
P3	Inverted Bottleneck: 5 × 5	96	64	2
P4	Inverted Bottleneck: 5 × 5	192	64	1
P4	Inverted Bottleneck: 5 × 5	192	64	1
P4	Inverted Bottleneck: 5 × 5	192	64	1
P4	Inverted Bottleneck: 5 × 5	192	64	1
P4	Inverted Bottleneck: 5 × 5	576	96	2
P5	Inverted Bottleneck: 5 × 5	576	96	1
P5	Inverted Bottleneck: 5 × 5	576	96	1
P5	Inverted Bottleneck: 5 × 5	576	96	1
P5	Inverted Bottleneck: 5 × 5	576	96	1

We conduct comprehensive experiments to evaluate the impact of different convolutional kernel configurations in an inverted bottleneck, (1) entire network using 3 × 3 kernels, (2) 5 × 5 kernels starting from P3 level (80 × 80 feature maps), (3) 7 × 7 kernels starting from P3 level, and (4) 5 × 5 kernels starting from P4 level (reduced computation at P3). The results are shown in Table 3. We can see that larger kernels consistently improve detection accuracy, with the 7 × 7 configuration achieving the highest mAP (88.2%), surpassing 3 × 3 by 2.4% and 5 × 5 by 0.4%, demonstrating the effectiveness of larger receptive fields for feature extraction. However, while 7 × 7 kernels provide the best accuracy, they incur 21.9% higher computational cost compared to 5 × 5 kernels (3.9 vs. 3.2 GFLOP),

making the 5 × 5 configuration more practical with 87.9% mAP at moderate computation. Additionally, strategic kernel placement shows that starting 5 × 5 kernels from P4 instead of P3 reduces computation by 6.25% (3.0 vs. 3.2 GFLOP) while maintaining comparable accuracy (87.7% vs. 87.9%), leveraging the smaller feature map size at P4 (40 × 40 vs. P3's 80 × 80) to minimize the computational overhead. The 5 × 5 (P4) configuration emerges as the optimal choice, achieving near-state-of-the-art accuracy (87.7% mAP) with only 7.1% higher computation than the baseline 3 × 3 model (3.0 vs. 2.8 GFLOP).

Table 3. Comparison of kernel settings.

3 × 3	5 × 5	7 × 7	5 × 5 (P4)	mAP	FLOP (G)
✓				85.5	2.8
	✓			87.9	3.2
		✓		88.2	3.9
			✓	87.7	3.0

3.2.2. Comparison of Feature Fusion Methods

The comprehensive experimental evaluation demonstrates the effectiveness of our Fast Feature Pyramid Network (FFPN). As shown in Table 4, the proposed method achieves competitive detection accuracy (87.7 mAP) compared to PANet (87.8 mAP), while reducing computational costs by 34.8% compared to PANet and 16.7% compared to FPN. Compared with traditional methods, our approach offers a simple structure, and computational flow optimization significantly improves hardware utilization. By focusing on the P4 level and reducing inter-level dependencies, we achieve higher parallelism than conventional methods and eliminate feature splicing blocking for more efficient processing. While FPN maintains advantages in simplicity and ease of implementation, and PANet excels in multi-level feature fusion for higher accuracy, our method effectively balances these aspects—maintaining comparable accuracy to PANet while approaching the computational efficiency of FPN. The design effectively solves the problems of the high-layer dependency and low parallelism of traditional methods, improves hardware utilization by reducing inter-layer dependency and eliminating feature splicing blocking, and provides a more efficient lightweight feature fusion scheme for the real-time detection of edge devices.

Table 4. Comparison of feature fusion methods.

Method	mAP	FLOP (G)
FPN	85.6	1.8
PANet	87.8	2.3
FFPN	87.7	1.5

3.2.3. Comparison of Detection Heads

As shown in Table 5, the experimental results demonstrate that our head achieves comparable accuracy to YOLOv8's dual-branch head (87.7% vs. 87.6% mAP50) with 44.4% fewer computations (0.5 vs. 0.9 GFLOPs), showing that spatial refinement through depthwise convolutions provides sufficient context without branch duplication. Incorporating optimized depthwise convolutions improves the detection accuracy by 4.5 percentage points over the pure pointwise baseline (87.7% vs. 83.2%) while only adding 0.1 GFLOPs, as the 3 × 3 depthwise layers enhance spatial relationships efficiently. Experiments show that the decoupled detection header introduced into the network-in-network design can significantly optimize the computational effort while maintaining the original detection

accuracy compared to the traditional two-branch approach that handles classification and regression via independent streams.

Table 5. Performance comparison of detection heads.

Head Design	mAP50	FLOP (G)
Pointwise-Only	83.2	0.4
YOLOv8 Dual-Branch	87.6	0.9
Ours	87.7	0.5

3.3. Comparison with State-of-the-Art Methods

The proposed method demonstrates significant advantages across multiple dimensions as shown in Table 6. At 320 × 320 input resolution, our approach achieves 67.3 mAP50 and 42.8 mAP50-95, outperforming MobileNetv3 by 7.8 and 16.6, respectively, while maintaining comparable computational costs (1.1 G vs. 1.0 G FLOP). Notably, this performance gain is attained with 40.6 fewer parameters (1.9 M vs. 3.2 M), highlighting our architectural efficiency in low-resolution scenarios. When scaled to 640 × 640 resolution, our method establishes new state-of-the-art performance with 87.7 mAP50 and 54.0 mAP50-95, surpassing all YOLO variants including YOLOv10n (87.4/53.1). This is achieved with 34.3 fewer FLOP than YOLOv10n (4.4 G vs. 6.7 G) and 42.9 fewer parameters than YOLOv8n (1.9 M vs. 3.2 M). The consistent parameter count across resolutions (1.9 M) indicates effective architectural unification without resolution-specific adaptations. The 54.0 mAP50-95 represents a 3.2 improvement over YOLOv8n, demonstrating particular strength in strict localization accuracy. Compared to other methods besides YOLO, our method also has significant advantages in terms of lightweight and efficiency compared to EfficientDet-D0, RT-DETR-R18, and RT-DETRv2-R18: With only 1.9 million parameters (reductions of 51%/90%/90% compared to the three models, respectively) and 4.4 GFLOPs of computational power (just 7.3% of the RT-DETR series), it achieves an 87.7% mAP50 accuracy on a 640-input (exceeding EfficientDet-D0's 82.9%, and approaching the 90.2% of RT-DETRv2-R18), significantly outperforming the Transformer architecture, which relies on high computational power. These advancements position our method as a computationally efficient solution suitable for both edge devices and high-accuracy applications.

Table 6. Performance comparison with advance lightweight methods.

Method	Input Size	mAP50	mAP50-95	FLOP (G)	Params (M)
MobileNetv3	320	59.5	26.2	1.0	3.2
YOLOv5n	640	82.3	49.6	4.5	2.6
YOLOv8n	640	85.1	51.8	8.7	3.2
YOLOv9t	640	86.0	52.4	7.7	2.0
YOLOv10n	640	87.4	53.1	6.7	2.3
EfficientDet-D0	512	82.9	50.8	5.0	3.9
RT-DETR-R18	640	89.5	57.1	60	20
RT-DETRv2-R18	640	90.2	58.2	60	20
Ours	320	67.3	42.8	1.1	**1.9**
Ours	640	87.7	54.0	**4.4**	**1.9**

As illustrated in Figure 4, which presents the classification accuracy across different target categories, the columns represent the recall rates for each class, while the rows indicate the precision rates, with "background" denoting the non-target class. It can be observed that the **car** (recall: 0.98) and **harbor** (0.95) classes achieve near-perfect performance, indicating their distinct features and sufficient training. Although **ship** (0.92) and **aircraft**

(0.91) maintain high recall rates, they exhibit elevated false alarm rates of 35% and 29%, respectively. The **bridge** (0.71) and **tank** (0.66) classes demonstrate notably lower recall rates. Specifically, 29% of bridge instances are misclassified as background, suggesting that their slender structures are prone to confusion with background clutter, while 34% of tank misclassifications highlight insufficient feature distinction between oil tanks and natural environments.

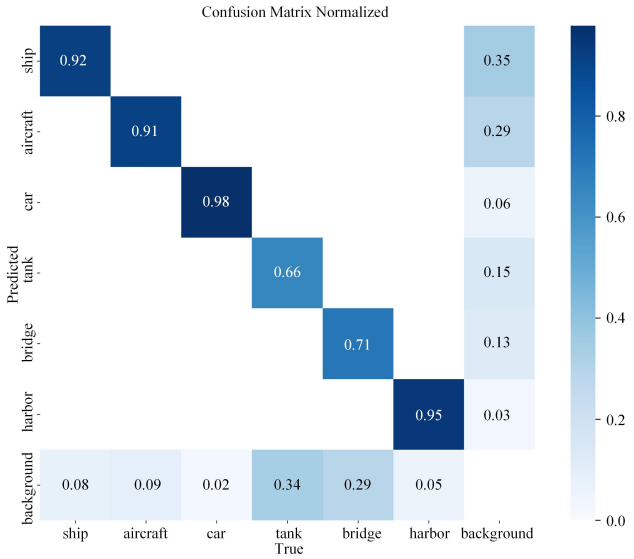


Figure 4. Confusion matrix for different detection categories.

Similarly, as shown in Figure 5 through precision–recall curve analysis, the model achieves exceptional performance on most classes (e.g., **car**, **harbor**, and **ship**), with average precision (AP) exceeding 0.94. Notably, the **car** class attains near-perfect detection capability with an AP of 0.979. However, significant shortcomings persist in recognizing complex targets such as **tank** (AP: 0.703) and **bridge** (AP: 0.754). Overall, the model exhibits robust comprehensive performance.

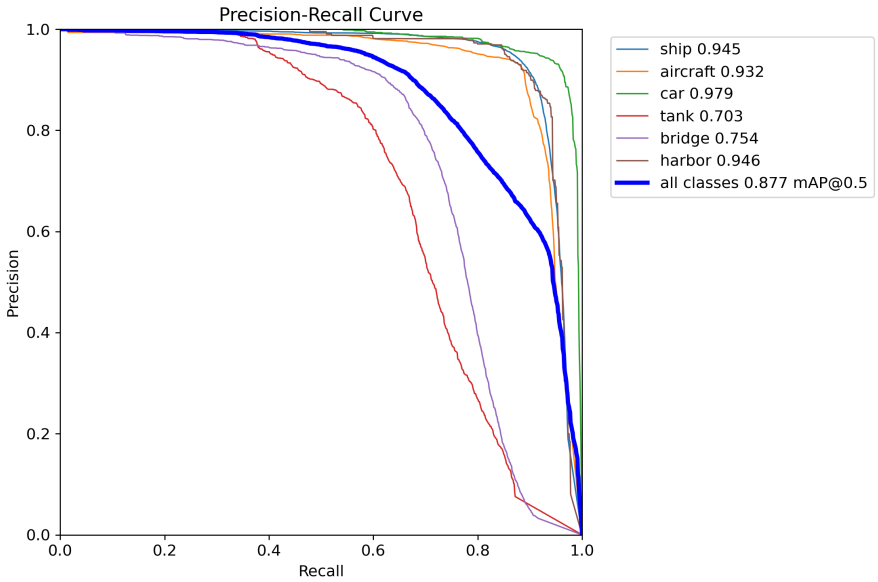


Figure 5. Precision–recall curve for different detection categories.

Figure 6 visualizes the detection results of YOLOv8n, the proposed method, and the ground truth, intuitively demonstrating the performance differences between different

methods in complex scenarios. The figure uses differently colored rectangular boxes (e.g., red for ships and pink for aircraft) to distinguish target categories, and incorrect cases are intuitively marked with green circles (false alarm) and red circles (missed alarm). YOLOv8n, as the baseline model, exhibits a high false positive rate and false negative issues, while the improved method demonstrates significant effectiveness in reducing these two types of errors, particularly showing greater robustness in detecting complex backgrounds (ships and harbors) and low-contrast aircraft targets.

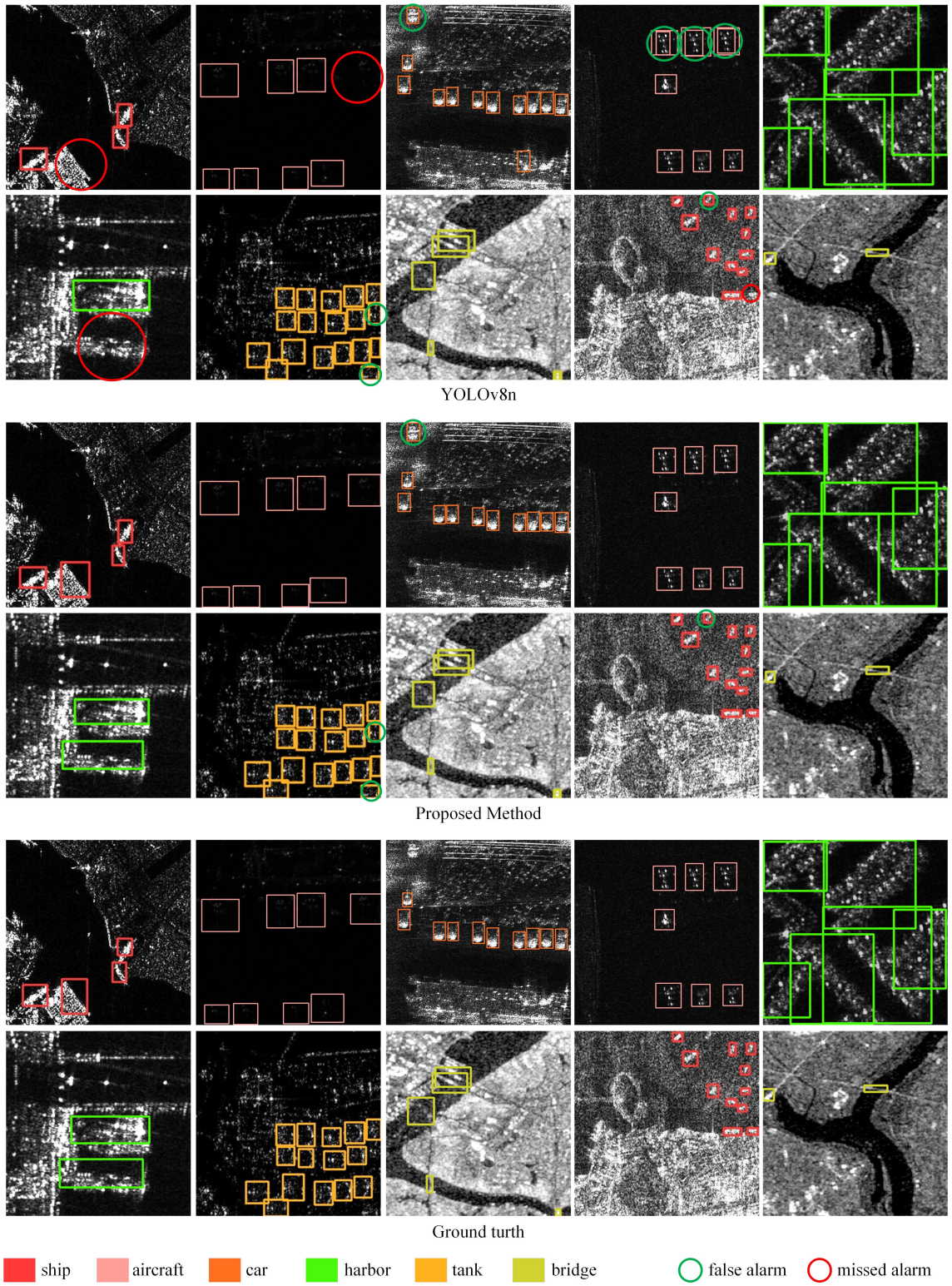


Figure 6. Visualization of detection results for some typical scenarios.

3.4. Performance Comparison on Edge Devices

As shown in Table 7, we compare the proposed method with other state-of-the-art YOLO methods in two edge-AI platforms.

The experimental results demonstrate that our model achieves the most efficient architecture design, with the lowest computational complexity (4.4 G FLOP) and smallest parameter count (1.9 M) among all compared methods. This represents a 2.2% reduction in FLOP compared to the previously most efficient YOLOv5n, while simultaneously achieving a 26.9% reduction in parameters. Notably, our method maintains competitive inference speeds despite its extreme efficiency. On the Jetson TX2 platform, our solution achieves 34.2 FPS, outperforming both YOLOv8n (32.8 FPS) and YOLOv9-Tiny (30.8 FPS), while being only marginally slower than YOLOv5n (38.1 FPS). Additionally, since YOLOv10n adopts a design without NMS post-processing, although the proposed method has much lower computational complexity and parameter count than YOLOv10n, its processing speed advantage is not significant. This performance is particularly remarkable considering our model's significantly reduced computational requirements. The results validate our design philosophy of optimizing for FLOP efficiency rather than purely chasing speed metrics.

The comparison reveals that our method successfully balances three aspects of edge deployment: computational efficiency, model compactness, and real-time performance. While inverted bottlenecks may limit parallelization potential as noted in the paper, our architecture demonstrates that careful design can still deliver practical inference speeds suitable for embedded applications. The 1.9 M parameter size makes our model especially attractive for memory-constrained edge devices, where model footprint directly impacts deployability. Furthermore, our approach shows consistent performance advantages across both computational and speed metrics when compared to more recent alternatives like YOLOv8n and YOLOv10n. This suggests that our architectural innovations provide meaningful improvements beyond simple model scaling or incremental updates to existing designs. The results show that the proposed method is more suitable for the task of SAR image object detection on edge devices.

Table 7. Performance comparison of lightweight methods on edge devices.

Method	FLOP (G)	Params (M)	FPS (TX2)	FPS (310)
YOLOv5n	4.5	2.6	38.1	32.5
YOLOv8n	8.7	3.2	32.8	28.3
YOLOv9-Tiny	7.7	2.0	30.8	25.6
YOLOv10n	6.7	2.3	34.5	30.1
Ours	**4.4**	**1.9**	**34.2**	**30.7**

4. Discussion

Analysis of Experimental Results. The proposed edge-optimized lightweight YOLO method achieved a detection accuracy of 87.7% mAP50 and 54.0% mAP50-95 on the SARDet-100K dataset, while requiring only 4.4 GFLOPs of computational load and 1.9 M parameters, significantly outperforming YOLOv10n (87.4% mAP50, 6.7 GFLOPs) and YOLOv8n (85.1% mAP50, 8.7 GFLOPs). The P4-driven Fast Feature Pyramid Network reduced the computational cost of the neck by 34.8% while maintaining multi-scale feature fusion capabilities. The decoupled detection head adopted a point convolution network structure, reducing the computational load by 44.4% compared to the traditional dual-branch design, with only a 0.1% drop in accuracy. Deployment tests on edge devices demonstrated real-time performance, achieving 34.2 FPS on Jetson TX2 and 30.7 FPS on Huawei Atlas DK 310. As shown in Figure 4, the model exhibited excellent detection performance for vehicles (recall rate 0.98) and ports (0.95) but showed noticeable deficiencies

in identifying bridges (0.71) and oil tanks (0.66), primarily due to their slender structures or high similarity to the background, leading to feature confusion.

Potential Limitations. The method performed weakly in detecting bridges and oil tanks, with AP scores of only 0.754 and 0.703, indicating limited feature extraction capability for targets with special structures. Although the depthwise separable convolution used in the inverted bottleneck layer significantly reduced the computational load, its channel-wise operation characteristic lowered parallel efficiency, resulting in slightly lower FPS compared to YOLOv5n (34.2 vs. 38.1 FPS). Despite the optimization of multi-scale fusion by FFPN, missed detections still occurred for extremely small targets . Current experiments were only validated on two edge devices, and further evaluation on more heterogeneous hardware (e.g., FPGA and DSP) is needed to assess generalizability.

Practical Application. The lightweight nature of the method (1.9 M parameters) makes it highly suitable for UAV-mounted SAR systems, enabling real-time ship/aircraft detection. Combined with the low-power (8 W) feature of Ascend chips, it can support real-time ship tracking on satellites. In the field of remote sensing reconnaissance, the model's real-time performance can meet the demand for rapid target identification in complex environments.

Future Research. The follow-up work will focus on two aspects: first, continuing to validate performance across different embedded devices while researching universal quantization and pruning techniques to further reduce computational load and improve detection speed; second, leveraging incremental learning and few-shot learning techniques to enhance the model's detection capability for imbalanced class distributions and unknown incremental categories, thereby improving the robustness of long-term deployment.

5. Conclusions

To meet the deployment requirements on edge devices, this paper proposes a lightweight SAR image object detection method. By designing an efficient backbone network based on inverted bottlenecks, the method significantly reduces the computational complexity while preserving the feature representation capability. The rapid Feature Pyramid Network achieves efficient multi-scale feature fusion through simplified hierarchical dependencies. Combined with a decoupled detection head design, it achieves an optimal balance between computational efficiency and detection accuracy for classification and regression tasks. Experimental results demonstrate that the proposed method achieves 87.7 mAP detection accuracy on the SARDet-100K dataset with only 1.9 M parameters and 4.4 GFLOP computational cost, showing superior accuracy-efficiency trade-offs compared to mainstream lightweight models. Deployment validation on edge platforms (Jetson TX2 and Huawei Atlas DK 310) shows the method's capability for real-time applications.

Author Contributions: Methodology, C.Z.; Software, R.Y., S.G. and S.L.; Validation, S.W., F.Z. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shanghai Science and Technology Program, grant number 24YF2743600.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
2. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

3. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9626–9635.
4. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
5. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable transformers for end-to-end object detection. In Proceedings of the 9th International Conference on Learning Representations, Virtual Event, Austria, 3–7 May 2021; pp. 3–7.
6. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.; Shum, H.Y. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In Proceedings of the 11th International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
7. Zhou, Z.; Zhao, L.; Ji, K.; Kuang, G. A domain adaptive few-shot SAR ship detection algorithm driven by the latent similarity between optical and SAR images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *65*, 5216318. [CrossRef]
8. Luo, C.; Zhang, Y.; Guo, J.; Hu, Y.; Zhou, G.; You, H.; Ning, X. SAR-CDSS: A semi-supervised cross-domain object detection from optical to SAR domain. *Remote Sens.* **2024**, *16*, 940. [CrossRef]
9. Huang, H.; Guo, J.; Lin, H.; Huang, Y.; Ding, X. Domain Adaptive Oriented Object Detection from Optical to SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *63*, 5200314. [CrossRef]
10. Xi, C.; Wang, Z.; Wang, W.; Xie, X.; Kang, J.; Fernandez-Beltran, R. Cromoda: Unsupervised oriented sar ship detection via cross-modality distribution alignment. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 11899–11914.
11. Sun, Z.; Leng, X.; Zhang, X.; Zhou, Z.; Xiong, B.; Ji, K.; Kuang, G. Arbitrary-Direction SAR Ship Detection Method for Multiscale Imbalance. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 1–21. [CrossRef]
12. Zhang, X.; Zhang, S.; Sun, Z.; Liu, C.; Sun, Y.; Ji, K.; Kuang, G. Cross-Sensor SAR Image Target Detection Based on Dynamic Feature Discrimination and Center-Aware Calibration. *IEEE Trans. Geosci. Remote Sens.* **2025**, *63*, 1–17. [CrossRef]
13. Liu, M.; Hou, B.; Ren, B.; Jiao, L.; Yang, Z.; Zhu, Z. Sub-pixel Feature Pyramid Network for Multi-Scale Ship Detection in Synthetic Aperture Radar Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 15583–15595. [CrossRef]
14. Ma, Y.; Guan, D.; Deng, Y.; Yuan, W.; Wei, M. 3SD-Net: SAR Small Ship Detection Neural Network. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5221613. [CrossRef]
15. Chen, J.; Wang, H.; Lu, H. Aircraft detection in SAR images via point features. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 1–5. [CrossRef]
16. Zhao, P.; Chen, J.; Wan, H.; Cao, Y.; Wang, S.; Zhang, Y.; Li, Y.; Huang, Z.; Wu, B. Few-Shot Object Detection for SAR Images via Context-Aware and Robust Gaussian Flow Representation. *Remote Sens.* **2025**, *17*, 391. [CrossRef]
17. Xiao, X.; Li, Z.; Mi, X.; Gu, D.; Wang, H. OSAD: Open-Set Aircraft Detection in SAR Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *18*, 3071–3086. [CrossRef]
18. Chen, M.; He, Y.; Wang, T.; Hu, Y.; Chen, J.; Pan, Z. Scale-Mixing Enhancement and Dual Consistency Guidance for End-to-End Semi-Supervised Ship Detection in SAR Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 15685–15701. [CrossRef]
19. Yang, Y.; Lang, P.; Yin, J.; He, Y.; Yang, J. Data matters: Rethinking the data distribution in semi-supervised oriented sar ship detection. *Remote Sens.* **2024**, *16*, 2551. [CrossRef]
20. Yang, Z.; Xia, X.; Liu, Y.; Wen, G.; Zhang, W.E.; Guo, L. LPST-Det: Local-perception-enhanced swin transformer for SAR ship detection. *Remote Sens.* **2024**, *16*, 483. [CrossRef]
21. Li, C.; Hei, Y.; Xi, L.; Li, W.; Xiao, Z. GL-DETR: Global-to-Local Transformers for Small Ship Detection in SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 4016805. [CrossRef]
22. Zhou, P.; Wang, P.; Cao, J.; Zhu, D.; Yin, Q.; Lv, J.; Chen, P.; Jie, Y.; Jiang, C. PSFNet: Efficient detection of SAR image based on petty-specialized feature aggregation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *17*, 190–205. [CrossRef]
23. Chen, Y.; Shen, Y.; Duan, C.; Wang, Z.; Mo, Z.; Liang, Y.; Zhang, Q. Robust and Efficient SAR Ship Detection: An Integrated Despecking and Detection Framework. *Remote Sens.* **2025**, *17*, 580. [CrossRef]
24. Feng, Y.; You, Y.; Tian, J.; Meng, G. OEGR-DETR: A novel detection transformer based on orientation enhancement and group relations for SAR object detection. *Remote Sens.* **2023**, *16*, 106. [CrossRef]
25. Liu, W.; Zhou, L. Multi-level Denoising for High Quality SAR Object Detection in Complex Scenes. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5226813. [CrossRef]
26. Zhao, S.; Luo, Y.; Zhang, T.; Guo, W.; Zhang, Z. A domain specific knowledge extraction transformer method for multisource satellite-borne SAR images ship detection. *ISPRS J. Photogramm. Remote Sens.* **2023**, *198*, 16–29. [CrossRef]
27. Lin, H.; Liu, J.; Li, X.; Wei, L.; Liu, Y.; Han, B.; Wu, Z. Dcea: Detr with concentrated deformable attention for end-to-end ship detection in sar images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 17292–17307. [CrossRef]
28. Chen, B.; Yu, C.; Zhao, S.; Song, H. An anchor-free method based on transformers and adaptive features for arbitrarily oriented ship detection in SAR images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *17*, 2012–2028. [CrossRef]
29. Liu, W.; Zhou, L.; Zhong, S.; Gong, S. Semantic Assistance in SAR Object Detection: A Mask-Guided Approach. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 19395–19407. [CrossRef]

30. Li, Y.; Li, X.; Li, W.; Hou, Q.; Liu, L.; Cheng, M.M.; Yang, J. Sardet-100k: Towards open-source benchmark and toolkit for large-scale sar object detection. *arXiv* **2024**, arXiv:2403.06534.
31. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
32. Jocher, G. Ultralytics YOLOv5. 2020. Available online: https://zenodo.org/records/7347926 (accessed on 22 November 2022).
33. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
34. Gao, D.; Wu, X.; Wen, Z.; Xu, Y.; Chen, Z. SVDD: SAR vehicle dataset construction and detection. *IEEE Access* **2025**, *13*, 18107–18122. [CrossRef]
35. Wang, C.; Cai, X.; Wu, F.; Cui, P.; Wu, Y.; Zhang, Y. Stepwise Attention-Guided Multiscale Fusion Network for Lightweight and High-Accurate SAR Ship Detection. *Remote Sens.* **2024**, *16*, 3137. [CrossRef]
36. Huo, L.; Li, H.; Wang, W.; Gao, X.; Wei, Y.; Chen, K. GSE-Ships: Ship Detection Using Optimized Lightweight Networks and Attention Mechanisms. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Urumqi, China, 18–20 October 2024; Springer: Singapore, 2024; pp. 516–530.
37. Guan, T.; Chang, S.; Wang, C.; Jia, X. SAR Small Ship Detection Based on Enhanced YOLO Network. *Remote Sens.* **2025**, *17*, 839. [CrossRef]
38. Chen, C.; Ding, H.; Duan, M. Discretization and decoupled knowledge distillation for arbitrary oriented object detection. *Digit. Signal Process.* **2024**, *150*, 104512. [CrossRef]
39. Zhang, Y.; Jia, Y.; Tang, Y. Accurate detection of arbitrary ship directions using SAR based on RTMDet. *Remote Sens. Lett.* **2025**, *16*, 156–169. [CrossRef]
40. Wang, X.; Xu, W.; Huang, P.; Tan, W. MSSD-Net: Multi-Scale SAR Ship Detection Network. *Remote Sens.* **2024**, *16*, 2233. [CrossRef]
41. Luo, Y.; Li, M.; Wen, G.; Tan, Y.; Shi, C. SHIP-YOLO: A Lightweight synthetic aperture radar ship detection model based on YOLOv8n algorithm. *IEEE Access* **2024**, *12*, 37030–37041. [CrossRef]
42. Huang, X.; Xu, K.; Chen, J.; Wang, A.; Chen, S.; Li, H. Real-Time Processing of Ship Detection with SAR Image Based on FPGA. In Proceedings of the IGARSS 2024—2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 7–12 July 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 8954–8957.
43. Fang, N.; Li, L.; Zhou, X.; Zhang, W.; Chen, F. An FPGA-Based Hybrid Overlapping Acceleration Architecture for Small-Target Remote Sensing Detection. *Remote Sens.* **2025**, *17*, 494. [CrossRef]
44. Zhang, S.; Pei, M.; Liu, X.; Zhao, X. Ship Detection in SAR Images Based on Oriented Bounding Box and Supervised Contrastive Learning. In Proceedings of the 2024 7th International Conference on Image and Graphics Processing, Beijing China, 19–21 January 2024; pp. 383–387.
45. Tian, S.; Jin, G.; Gao, J.; Tan, L.; Xue, Y.; Li, Y.; Liu, Y. Ship Detection in Synthetic Aperture Radar Images Based on BiLevel Spatial Attention and Deep Poly Kernel Network. *J. Mar. Sci. Eng.* **2024**, *12*, 1379. [CrossRef]
46. Zhu, S.; Wei, L.; Zheng, B.; Tao, X. Weakly supervised arbitrary-oriented ship detection network based on dynamic scale feature enhancement. *J. Electron. Imaging* **2025**, *34*, 013047. [CrossRef]
47. Yasir, M.; Liu, S.; Pirasteh, S.; Xu, M.; Sheng, H.; Wan, J.; de Figueiredo, F.A.; Aguilar, F.J.; Li, J. YOLOShipTracker: Tracking ships in SAR images using lightweight YOLOv8. *Int. J. Appl. Earth Obs. Geoinf.* **2024**, *134*, 104137. [CrossRef]
48. Rahman, N.; Khan, M.; Ullah, I.; Kim, D.H. A novel lightweight CNN for constrained IoT devices: Achieving high accuracy with parameter efficiency on the MSTAR dataset. *IEEE Access* **2024**, *12*, 160284–160298. [CrossRef]
49. Zhang, X.; Feng, Y.; Zhang, S.; Wang, N.; Lu, G.; Mei, S. Robust aerial person detection with lightweight distillation network for edge deployment. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5630616. [CrossRef]
50. Zhirui, W.; Yuzhuo, K.; Xuan, Z.; Yuelei, W.; Ting, Z.; Xian, S. SAR-AIRcraft-1.0: High-resolution SAR aircraft detection and recognition dataset. *J. Radars* **2023**, *12*, 906–922.
51. Wei, S.; Zeng, X.; Qu, Q.; Wang, M.; Su, H.; Shi, J. HRSID: A high-resolution SAR images dataset for ship detection and instance segmentation. *IEEE Access* **2020**, *8*, 120234–120254. [CrossRef]
52. Zhang, T.; Zhang, X.; Li, J.; Xu, X.; Wang, B.; Zhan, X.; Xu, Y.; Ke, X.; Zeng, T.; Su, H.; et al. SAR ship detection dataset (SSDD): Official release and comprehensive data analysis. *Remote Sens.* **2021**, *13*, 3690. [CrossRef]
53. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. 2023. Available online: https://yolov8.com (accessed on 10 January 2023).
54. Tishby, N.; Zaslavsky, N. Deep learning and the information bottleneck principle. In Proceedings of the 2015 IEEE Information Theory Workshop (ITW), Jerusalem, Israel, 26 April–1 May 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–5.
55. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.

56. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; IEEE Computer Society: Washington, DC, USA, 2021; pp. 3490–3499.
57. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.