

Article

YOLO-UAVShip: An Effective Method and Datasets for Multi-View Ship Detection in UAV Images

Yanguang Li ^{1,2}, Yichen Tian ^{1,2}, Chao Yuan ^{1,2,*}, Kun Yu ¹, Kai Yin ¹, Huiping Huang ^{1,2}, Guang Yang ¹, Fan Li ^{1,2}  and Zengguang Zhou ¹

¹ Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; liyouguang23@mails.ucas.ac.cn (Y.L.); tianyc@aircas.ac.cn (Y.T.); yukun@aircas.ac.cn (K.Y.); yinkai@aircas.ac.cn (K.Y.); huanghp@aircas.ac.cn (H.H.); yangguang201924@aircas.ac.cn (G.Y.); lifan23@mails.ucas.ac.cn (F.L.); zhouzg@aircas.ac.cn (Z.Z.)

² University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: yuanchao200973@aircas.ac.cn

Abstract

Maritime unmanned aerial vehicle (UAV) ship detection faces challenges including variations in ship pose and appearance under multiple viewpoints, occlusion and confusion in dense scenes, complex backgrounds, and the scarcity of ship datasets from UAV tilted perspectives. To overcome these obstacles, this study introduces a high-quality dataset named Marship-OB9, comprising 11,268 drone-captured images and 18,632 instances spanning nine typical ship categories. The dataset systematically reflects the characteristics of maritime scenes under diverse scales, viewpoints, and environmental conditions. Based upon this dataset, we propose a novel detection network named YOLO11-UAVShip. First, an oriented bounding box detection mechanism is incorporated to precisely fit ship contours and reduce background interference. Second, a newly designed CK_DCNv4 module, integrating deformable convolution v4 (DCNv4) and a C3k2 backbone structure, is developed to enhance geometric feature extraction under aerial oblique view. Additionally, for ships with large aspect ratios, SGKLD effectively addresses the localization challenges in dense environments, achieving robust position regression. Comprehensive experimental evaluation demonstrates that the proposed method yields a 2.1% improvement in mAP@0.5 and a 2.3% increase in recall relative to baseline models on the Marship-OB9 dataset. While maintaining real-time inference speed, our approach greatly enhances detection accuracy and robustness. This work provides a practical and deployable solution for intelligent ship detection in UAV imagery.

Keywords: UAV; ship detection; deep learning; multi-view; ship dataset



Academic Editors: Bingxin Liu, Jin Xu and Peng Liu

Received: 3 July 2025

Revised: 22 August 2025

Accepted: 3 September 2025

Published: 8 September 2025

Citation: Li, Y.; Tian, Y.; Yuan, C.; Yu, K.; Yin, K.; Huang, H.; Yang, G.; Li, F.; Zhou, Z. YOLO-UAVShip: An Effective Method and Datasets for Multi-View Ship Detection in UAV Images. *Remote Sens.* **2025**, *17*, 3119. <https://doi.org/10.3390/rs17173119>

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

As global maritime trade continues to expand, high-traffic zones such as port intersections and navigational chokepoints are experiencing unprecedented pressure in terms of vessel scheduling and traffic management. According to statistics from the United Nations Conference on Trade and Development (UNCTAD), the global commercial fleet surpassed 109,000 ships by 2024, placing increasing demands on maritime monitoring systems for real-time and intelligent surveillance [1]. In recent years, image-based ship detection techniques have been widely applied in maritime surveillance, where their accuracy and timeliness directly influence the performance of maritime accident early warnings, vessel trajectory management, and illegal activity detection. However, most existing studies

rely on imagery acquired from remote sensing satellites or fixed shore-based cameras. While these platforms have shown promising performance in specific scenarios, their inherent limitations—such as long revisit cycles, restricted spatial resolution, and fixed imaging perspectives—compromise their real-time responsiveness and flexibility in dynamic and complex maritime environments. In contrast, UAVs, with their mobility and flexible deployment, can capture high-resolution surface imagery from multiple viewpoints and altitudes, offering huge potential for maritime applications. However, existing ship detection algorithms are not well-adapted to the specific challenges of UAV platforms [2–4].

Specifically speaking, unlike general-purpose object detection tasks, ships captured in UAV imagery often exhibit large variations in orientation, pose, size, and even geometric appearance. Models trained on satellite or shore-based images struggle to generalize across these multi-view, multi-scale characteristics, leading to a significant drop in detection accuracy [5]. Moreover, the maritime environment is inherently complex: natural factors such as waves, lighting conditions, and sea fog frequently degrade image clarity, resulting in blurred ship edges and indistinct contours. In high-traffic zones, frequent vessel occlusion and overlap further obscure object boundaries, increasing the risk of missed or false detections and reducing the reliability and robustness of detection models in real-world deployments. From the data perspective, existing UAV datasets—such as the Stanford UAV Dataset [6] and the VisDrone Dataset [7]—primarily focus on urban or inland scenes, lacking dedicated datasets tailored to maritime conditions. Meanwhile, most publicly available ship detection datasets are collected from satellite, shore-based, or onboard camera platforms, which fail to adequately reflect the ship's rich diversity in UAV-based maritime surveillance. This lack of representative training data severely limits the development and performance of deep learning models in this domain. Collectively, these challenges highlight the imperative to develop specialized methodologies and datasets for reliable UAV-based ship detection.

To address the aforementioned challenges, this paper proposes an efficiency-oriented object detection methodology designed for accurate and efficient ship detection in high-traffic maritime zones utilizing UAV platforms. The principal contributions of this research are enumerated below:

1. Construction of a high-quality dataset: We introduce Marship-OBB9, a novel UAV ship detection dataset comprising 11,268 aerial images and 18,632 instances across nine representative ship categories.
2. Design of a robust rotated detection framework: Building upon the YOLO11, we propose the YOLO11-UAVShip model by integrating an oriented bounding box (OBB) detection mechanism, the CK_DCNv4 module, and the SGKLD loss function specifically designed for robust position regression of ships with large aspect ratios in dense environments. This framework considerably improves detection accuracy and robustness under UAV perspective.
3. Optimization of accuracy–efficiency balance: The proposed method achieves absolute improvements of 2.1% in mAP@0.5 and 2.3% in recall compared to the baseline model on the Marship OBB9 dataset, while maintaining negligible impact on inference speed.
4. Comprehensive ablation and comparison experiments: We perform comprehensive ablation studies to evaluate the efficacy of each proposed component and benchmark our methodology against existing horizontal and rotated object detection models. The results demonstrate that our framework offers a superior balance between accuracy, efficiency, and generalization ability.

The remainder of this paper is organized as follows. Section 2 reviews related work on ship datasets, rotated object detection, and ship detection. Section 3 presents the architecture of the proposed YOLO-UAVShip network and analyzes its underlying theoretical

mechanisms. Section 4 introduces the experimental dataset, evaluation metrics, ablation studies, and comparison experiments, followed by a comprehensive analysis and discussion of the results. Section 5 reveals the limitations of the model. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2. Related Work

2.1. Marine Ship Detection Datasets

High-quality datasets constitute the cornerstone for advancing deep learning techniques in maritime ship detection. Over the past few years, a variety of ship detection datasets have been released, encompassing different sensor platforms, imaging modalities, and application scenarios, as summarized in Table 1. However, high-quality datasets that can truly meet the requirements of UAV maritime scenarios remain extremely scarce. Specifically speaking, most existing datasets are captured from satellite, shipborne, or shore-based perspectives, lacking large-scale ship image collections with diverse viewpoints and scales tailored for UAV-based detection tasks [8].

Table 1. Maritime ship detection datasets.

Dataset	Ship Classes	Images	Instances	Annotation	Year	Source
Seaship [9]	6	31,455	40,077	HBB	2018	Camera
HRSID [10]	1	5604	16,951	HBB	2020	SAR
McShips [11]	13	14,709	26,529	HBB	2020	Search engine
ABOships [12]	9	9880	41,967	HBB	2021	Camera
FAIRIM [13]	9	2235	-	OBB	2021	Satellite imagery
SSDD [14]	1	1160	2456	HBB/OBB	2021	SAR
MVDD13 [15]	13	35,474	40,839	HBB	2024	USV
MODD-13 [16]	13	9097	-	HBB	2024	USV
MS2ship [8]	1	6470	13,697	HBB	2024	UAV
Marship-OBB9	9	11,268	18,632	HBB/OBB	2025	UAV

For instance, datasets such as Seaship, ABOships, and McShips collect images from shore-based cameras or online search engines. While these datasets contain relatively rich categories and sample volumes, they fail to represent the diverse aerial perspectives and pose variations inherent to UAV imagery. FAIRIM and SSDD, built on satellite images, offer wide-area coverage but are limited in resolution and fine-grained visual details and contain only nadir (vertical) views of ships. Recently introduced datasets like MVDD13 and MODD-13 focus on fine-grained ship classification for unmanned surface vehicle (USV) scenarios, incorporating various scales, lighting conditions, side views, ship parts, backgrounds, and occlusions—but they explicitly exclude top-down aerial imagery. MS2ship compiles UAV rescue images from the Seagull and SeaDronesSee datasets, providing diverse appearances and background conditions. However, it only includes a single ship class, and annotations are limited to horizontal bounding boxes (HBBs).

To deal with these limitations, we present Marship-OBB9, a large-scale, fine-grained ship detection dataset specifically designed for UAV ship identification. The dataset is constructed from UAV aerial imagery collected over different maritime regions and under various environmental conditions. It includes 11,268 images and 18,632 annotated instances across nine representative ship categories. Notably, Marship-OBB9 provides both HBB and OBB annotations. Compared with the existing datasets, Marship-OBB9 performs better in terms of view angle, scale, attitude, and category diversity, which can provide strong data support for rotating target detection, multi-scale feature extraction, and other research

areas, and help improve the intelligent perception capability of UAV platforms in complex maritime environments.

2.2. Rotated Object Detection

In deep learning-based object detection, HBBs are typically employed to localize objects. However, in UAV-based ship detection scenarios, this approach exhibits evident limitations. Due to the elongated shape and arbitrary orientation of ships in UAV-captured imagery, conventional HBBs often fail to accurately delineate ship contours, resulting in bounding boxes that encompass substantial non-target background regions. This redundancy significantly reduces localization precision. Moreover, in high-traffic zones, a large number of overlapping horizontal bounding boxes are easily suppressed or rejected in the post-processing stage due to the high Intersection over Union (IoU) value, which leads to the leakage and misdetection of ship targets. As illustrated in Figure 1, OBB introduces an additional angle parameter to construct bounding boxes, noticeably enhancing the ability to model object geometry compared to HBB. OBB enables finer alignment with object shapes and minimizes interference from the surrounding background. Representative OBB-based detection methods such as S^2 ANet [17], RetinaNet-obb [18], and ReDet [19] have been proposed. More recently, the emergence of transformer-based detectors [20] has also led to notable progress in OBB, such as ROI-transformer [21], AO2-DETR [22], and ARS-DETR [23].

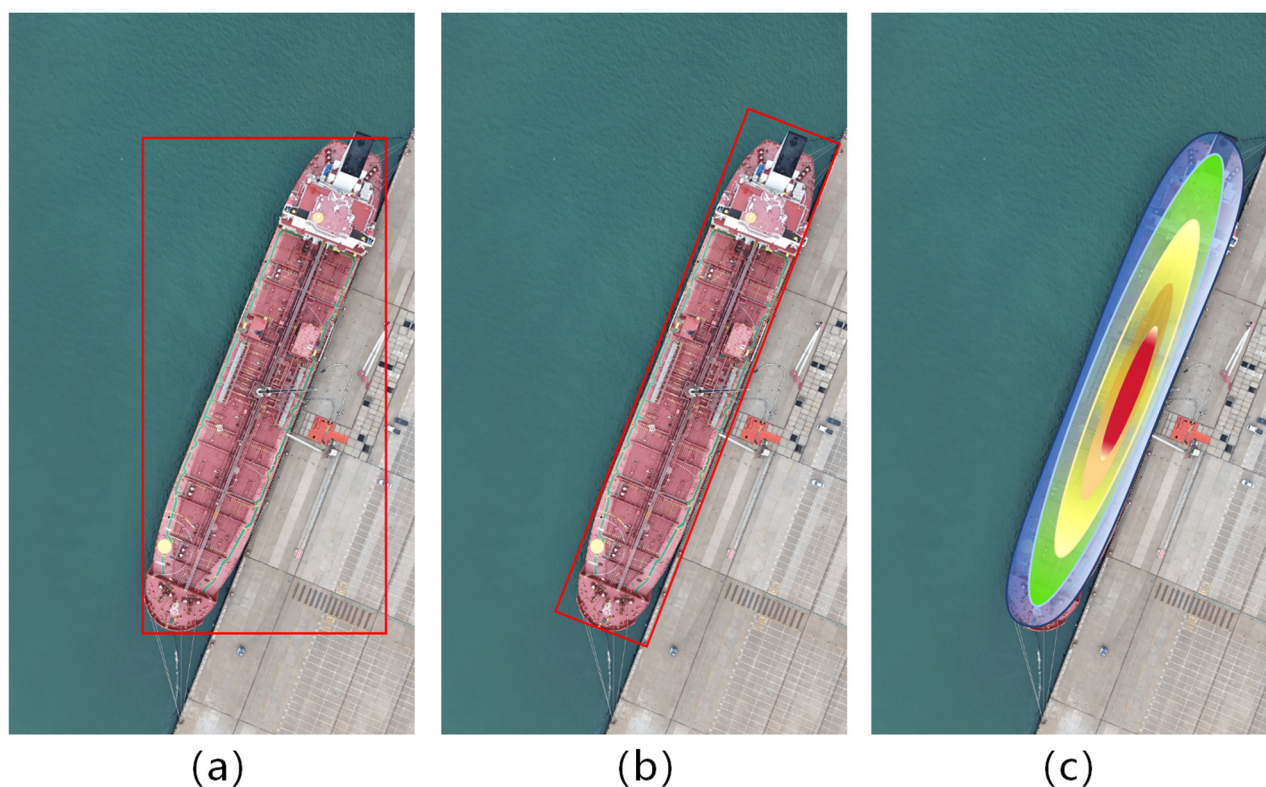


Figure 1. (a) Horizontal bounding box of the ship; (b) oriented bounding box of the ship; (c) Gaussian distribution of the oriented bounding box.

However, these rotated detection algorithms that introduce angular parameters face boundary discontinuities and angular discontinuities due to angular periodicity. This discontinuity in both boundary and angle space severely hampers model convergence and training stability. To overcome this challenge, the GWD [24] model converts each OBB to a Gaussian distribution and computes the localization loss using Wasserstein distance,

thereby alleviating boundary discontinuity issues. However, GWD lacks scale invariance, limiting its capability in accurately localizing small objects. Building upon GWD, KLD [25] replaces the Wasserstein distance with the Kullback–Leibler divergence and adjusts the angular weight according to the aspect ratio, thereby improving detection accuracy for high-aspect-ratio objects. While these probabilistic modeling methods based on 2D Gaussians have successfully tackled the boundary discontinuity associated with IoU-based rotation, they still struggle with objects that are nearly square in shape and thus less sensitive to angular variation, resulting in suboptimal detection performance in such cases.

2.3. Ship Detection

Most vessel identification methods are based on the existing general object detection architecture [26]. Traditional object detection techniques primarily rely on manually designed feature extraction methods (such as HOG [27] and SIFT [28]) to achieve object classification and position regression [29]. These methods often have huge limitations in practical applicability stemming from their suboptimal representation of target characteristics and significant computational complexity. With the rapid development of powerful parallel computing devices, such as graphics processing units (GPUs), deep learning technology has gained considerable attention. Convolutional neural networks (CNNs) [30,31] are the mainstream approach for object detection based on deep learning. Object detection algorithms based on CNN can be categorized into two-stage and one-stage methods. Two-stage algorithms, such as R-CNN [32], Faster R-CNN [33], and SPP-Net [34], offer higher detection accuracy but are relatively slower in inference speed. Conversely, one-stage object detection algorithms, such as the YOLO series [35–39] and SSD [40], have slightly lower detection accuracy but are faster in inference speed and more real-time, thus demonstrating greater advantages in practical applications.

To better apply general object detection algorithms to ship detection, many researchers have optimized and improved these algorithms. For example, Zwemer et al. [41] proposed a real-time vessel detection and tracking framework utilizing port surveillance camera feeds, in which the SSD detection model was trained by incorporating the dimensional and aspect ratio characteristics of ships. Wang et al. [42] incorporated a Comprehensive Feature Enhancement (CFE) module within the YOLOv3 architecture, which significantly improved the recognition capability for small and medium-sized vessels in optical satellite imagery. Tang et al. [43] developed an approach named H-YOLO that operates on high-resolution imagery, leveraging color contrast between vessels and background elements to preselect regions of interest and identify candidate areas. Li et al. [44] designed an image-based detection algorithm for unmanned surface vehicles (USVs), integrating DenseNet and YOLOv3 architectures to achieve multi-scale ship recognition and enhance detection robustness in real-world maritime settings. Guo et al. [45] integrated deformable convolution into their network to enhance infrared small ship detection on the maritime surface.

With the rapid advancement of UAV technology, its high mobility, low cost, and real-time imaging capabilities have demonstrated enormous advantages in maritime surveillance tasks. Some scholars have explored UAV ship detection methods [46]. Wang et al. [47] enhanced accuracy and speed in UAV videos by incorporating attention mechanisms and GSConv into the YOLOv7 model. Han et al. [48] designed a new model, SSMA-YOLO, which features a new SSC2f structure combined with a multi-dimensional attention mechanism to improve ship recognition in complex backgrounds. Feng et al. [2] addressed the scarcity of multi-angle and multi-scale ship datasets by proposing a data augmentation method using stable diffusion. Additionally, they improved the YOLOv8n-OBB model by integrating the BiFPN structure and EMA module, enhancing its ability to detect multi-scale ship instances. Cheng et al. [5] introduced a model named YOLOv5-ODConvNeXt, which

integrates ConvNeXt and ODConv modules into the backbone network to accelerate ship detection and adapt to the scale changes in ships due to UAV posture and altitude changes. Although existing studies have achieved certain progress in UAV-based ship detection, most methods have not sufficiently addressed the detection stability issues caused by variations in ship posture and appearance under multi-view conditions. To this end, this paper conducts research from both data construction and algorithm optimization perspectives, aiming to enhance the detection accuracy and robustness in UAV multi-view scenarios.

3. Proposed Network

The primary objective of this study is to achieve accurate and efficient ship detection in high-traffic zones using UAV platforms. Based on the proposed Marship-OBb9 dataset, we introduce an enhanced detection network built upon the YOLO11 [49] architecture. This section first provides an overview of the baseline YOLO11 model structure, followed by detailed descriptions of the architectural improvements and their intended purposes.

3.1. YOLO11

YOLO11 is a high-performance object detection architecture officially released by the Ultralytics team in September 2024. It inherits the end-to-end single-stage design paradigm of the YOLO family while further optimizing inference efficiency, making it capable of achieving real-time object detection even on resource-constrained devices.

The YOLO11 series includes five versions—YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x—differing in model size and computational requirements. Considering the need for rapid responsiveness in UAV-based maritime ship detection, we select YOLO11n, the lightest version, as our base model for further improvement.

As illustrated in Figure 2, the YOLO11 architecture is composed of three primary components: Backbone, Neck, and Head. The Backbone is responsible for extracting low-level features from input images. The Neck performs multi-scale feature aggregation and enhancement, and the Head outputs the final detection results. Structurally, YOLO11 introduces several enhancements, most notably the C3K2 module, an improved version of the traditional C3 module, which is a core feature extraction component introduced in YOLO11. It is primarily designed to enhance the model's multi-scale feature extraction capability in complex scenarios. Compared with the C2f module used in YOLOv8, the C3K2 module can dynamically select between using the C3K or the standard Bottleneck structure via the parameter *c3k*, thereby adapting to different feature hierarchy requirements. As shown in Figure 2, if *c3k* is set to True, the original bottleneck module is replaced with the C3K module.

3.2. YOLO11-OBb

To enhance the accuracy of ship target localization, this research incorporates an OBb detection mechanism. This approach adequately handles the diverse orientations and poses of ships frequently encountered in UAV images, allowing the model to more accurately align with ship contours and reduce background interference.

YOLO11-OBb is an extended model built upon the original YOLO11 framework, incorporating a rotated object detection head while maintaining the overall network architecture. Specifically, it shares the same Backbone and Neck structure as YOLO11, but introduces the following key modifications in the detection Head: Firstly, the output format is extended to include an additional angle parameter, enabling the prediction of object orientation in the form (x, y, w, h, θ) , where θ represents the rotation angle of the target. Secondly, the bounding box regression loss is replaced with ProbIoU [50], a loss function tailored for rotated object detection tasks.

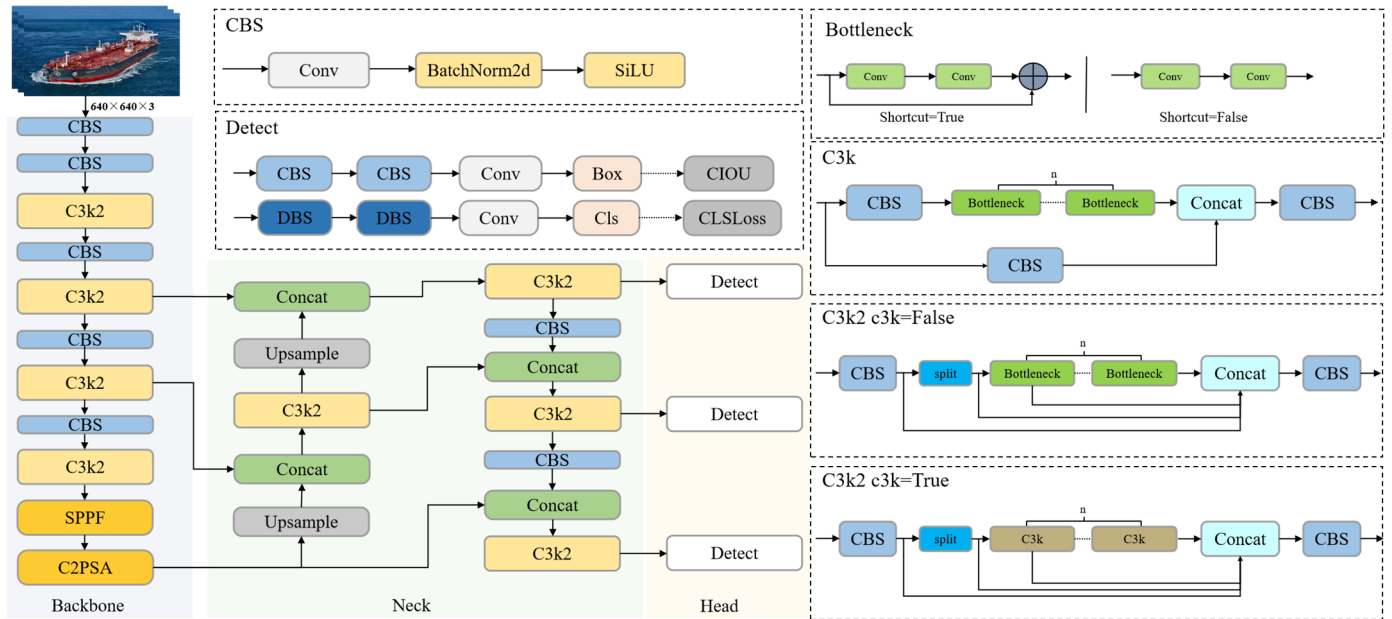


Figure 2. Network structure of YOLO11.

3.3. Improved YOLO-UAVShip Detection Algorithm

3.3.1. Overview

To enhance the accuracy and robustness of UAV ship type detection in high-traffic zones, we propose an improved detection algorithm named YOLO-UAVShip, which is built upon the YOLO11n-OBB framework.

The algorithmic enhancements mainly involve the CK_DCNv4 module and the SGKLD loss. Briefly, CK_DCNv4 improves the model's feature extraction capability to adapt to geometric deformations of ships in UAV images, while SGKLD enhances the model's ability to regress positions for large aspect-ratio objects, enabling stable rotation box regression. The overall architecture of the proposed YOLO-UAVShip model is illustrated in Figure 3.

3.3.2. Deformable Perception Module CK_DCNv4

For any given point p_0 on the input feature map, traditional convolution operations are generally limited to extracting features from a fixed sampling grid determined by the kernel size and stride. This rigid sampling mechanism struggles to adapt to the deformation nature of objects, and the output features are expressed as in Equation (1). In UAV imagery, ship targets are influenced by multiple factors, including drastic changes in viewing angles, variations in ship posture caused by waves, and structural differences among ship types. These factors collectively result in significant variability in the ship's appearance, while the complex maritime background can easily interfere with the target regions. Under such circumstances, traditional convolution often incorporates a large amount of background information during feature extraction, thereby weakening the focus on the ship's main contour features.

To address this issue, the Deformable Convolution Network (DCN) [51] innovatively introduces a dynamic offset mechanism. This mechanism overcomes the limitation of fixed sampling locations in traditional convolution, allowing feature sampling points to adaptively adjust according to the actual shape of the target. As a result, it significantly enhances feature representation and detection accuracy in UAV-based ship detection tasks. As illustrated in Figure 4, deformable convolution generates an additional learnable offset through a dedicated network (typically implemented via an extra convolution on the input

feature map), transforming the originally regular square receptive field into a flexible sampling pattern that closely aligns with the target's contour. The dynamic offsets at the sampling positions are automatically learned by the network, enabling precise capture of key structural features of the ship. Mathematically, the output feature values of DCN are computed as shown in Equation (2).

$$y(p_0) = \sum_{k=1}^K w_k \bullet x(p_0 + p_k) \quad (1)$$

$$y(p_0) = \sum_{k=1}^K w_k \bullet x(p_0 + p_k + \Delta p_k) \quad (2)$$

where $y(p_0)$ denotes the output feature value at location p_0 ; K represents the size of the convolution kernel; and w_k denotes the weight of the k position in the kernel. The term p_k corresponds to the offset of the k sampling point relative to the kernel center; $x(p_0 + p_k)$ indicates the feature value at location $(p_0 + p_k)$ in the input feature map; and Δp_k denotes the learnable offset added to p_k in deformable convolution.

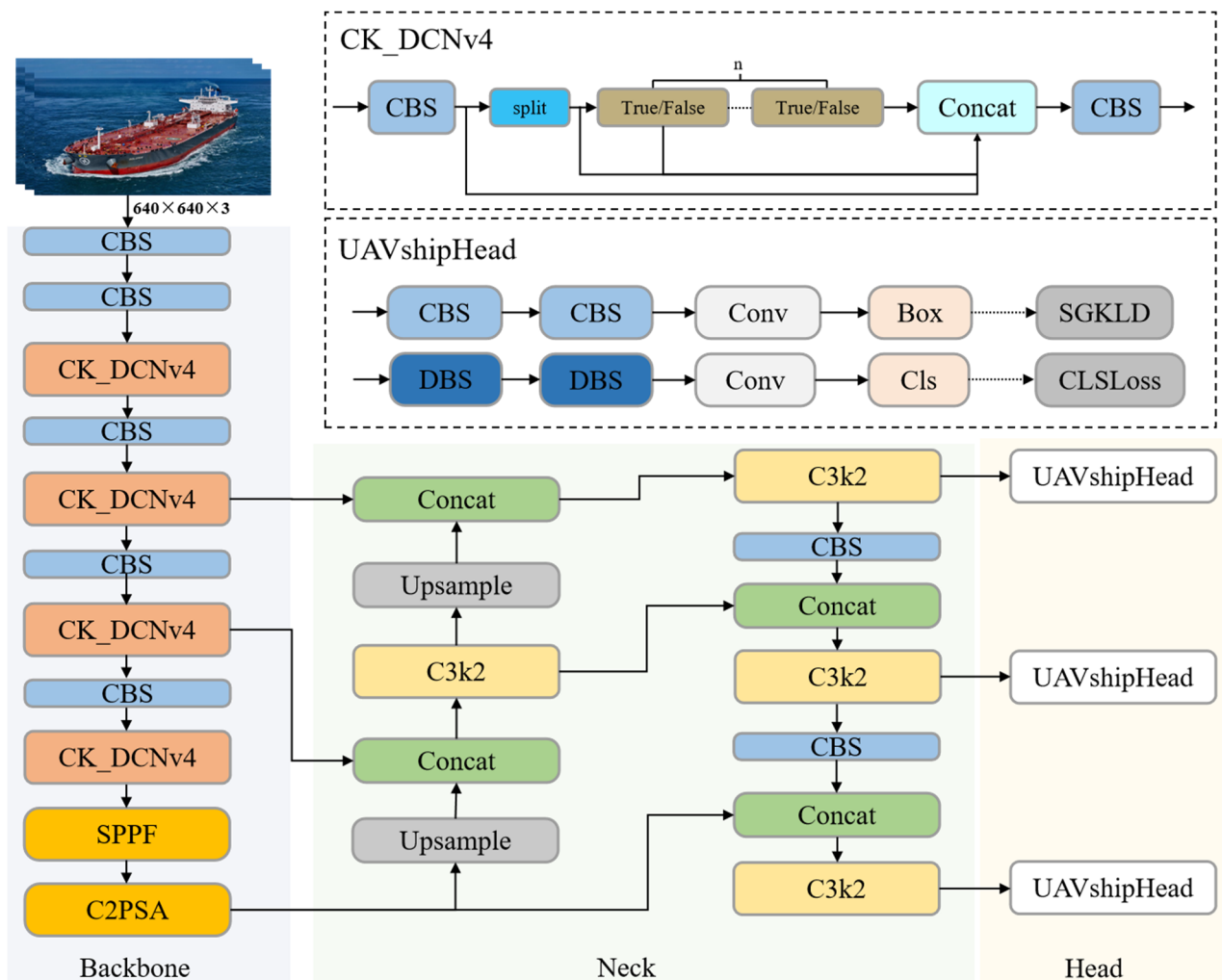


Figure 3. Network structure of YOLO-UAVship.

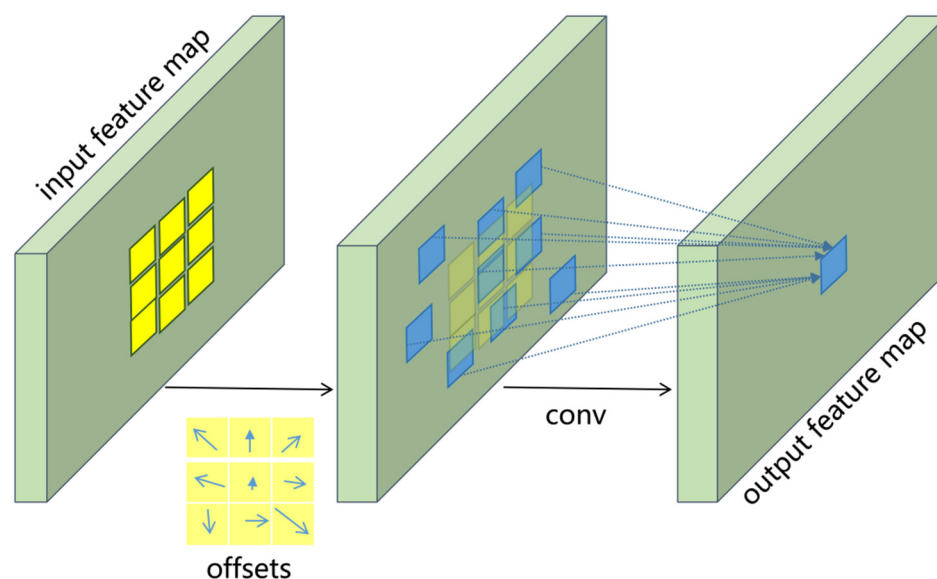


Figure 4. Illustration of 3×3 deformable convolutions. The fixed 3×3 yellow grid on the input feature map represents traditional convolution. Through the learnable offset mechanism, it is transformed into the flexible blue grid of the deformable convolution, which is then used to produce the output feature values.

DCNv4 is the latest version of DCN, featuring enhanced dynamic adaptability and expressive capability, enabling more precise capture of local non-rigid deformations, then making it suitable for scenarios with strong dynamic variations. Furthermore, as shown in Figure 5, DCNv4 optimizes memory access patterns through instruction-level kernel analysis, reducing redundant operations and significantly improving overall computational efficiency.

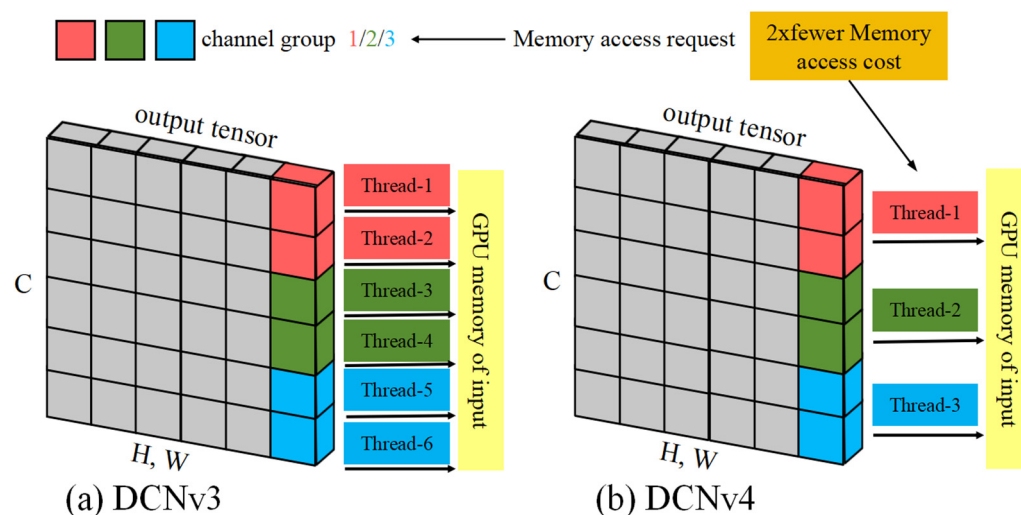


Figure 5. Comparison of DCNv3 and DCNv4 optimized memory access functions. H represents height, W represents width; and C represents the number of channels.

In this study, we improved the C3k2 structure in the YOLO11 backbone by replacing the conventional convolution in the bottleneck of the C3k2 module with DCNv4. The specific modifications for the CK_DCNv4 module are illustrated in Figure 6 below: DCNv4 replaces the standard convolution in the bottleneck of the C3k2 module, resulting in the Bottleneck_DCNv4 structure. This Bottleneck_DCNv4 is then used to replace the bottleneck in the C3k structure, forming the C3k_DCNv4 structure. Finally, C3k_DCNv4 replaces

the C3k in the original C3k2 module, yielding the CK_DCNv4 module. Similarly to C3k2, CK_DCNv4 can dynamically select between using the C3k_DCNv4 structure or the Bottleneck_DCNv4 structure via the parameter c3k.

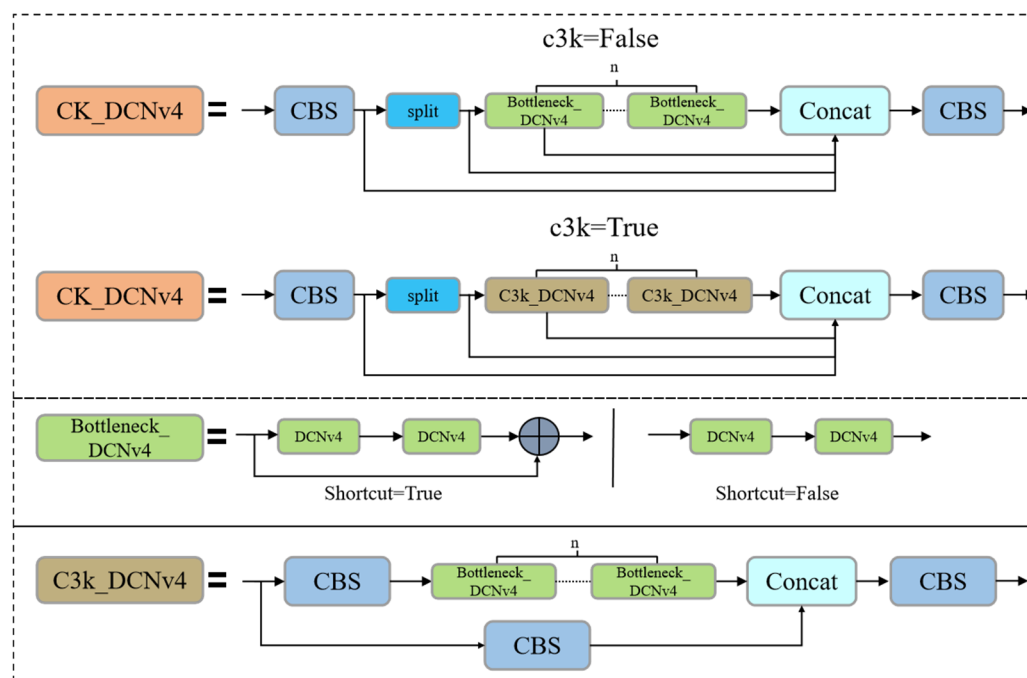


Figure 6. CK_DCNv4 Module Structure. The CK_DCNv4 module can dynamically select between using the C3k_DCNv4 structure or the Bottleneck_DCNv4 structure via the parameter c3k. Both the C3k_DCNv4 and Bottleneck_DCNv4 structures are obtained by replacing the standard convolution in the original modules with DCNv4.

3.3.3. Rotation-Robust Localization Loss Function

In the YOLO11 model, the ProbIoU loss function is used for the position regression of rotated bounding boxes, and its computation is as follows: Equations (3) and (4) represent the transformation of an oriented bounding box $B(c_x, c_y, w, h, \theta)$ into a Gaussian distribution $N(\mu, \Sigma)$. Here, c_x, c_y, w, h, θ denote the center coordinates, width, height, and rotation angle of the oriented bounding box. R denotes the rotation matrix, and Λ is the diagonal matrix of eigenvalues.

Then, the localization loss function is calculated based on the computation of the Hellinger distance.

$$\mu = (c_x, c_y)^\top \quad (3)$$

$$\begin{aligned} \frac{1}{\Sigma^2} &= R \Lambda R^\top = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \frac{w}{2} & 0 \\ 0 & \frac{h}{2} \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \\ &= \begin{pmatrix} \frac{w}{2} \cos^2 \theta + \frac{h}{2} \sin^2 \theta & \frac{w-h}{2} \cos \theta \sin \theta \\ \frac{w-h}{2} \cos \theta \sin \theta & \frac{w}{2} \sin^2 \theta + \frac{h}{2} \cos^2 \theta \end{pmatrix} \end{aligned} \quad (4)$$

However, ProbIoU has two limitations: on one hand, for targets with large aspect ratios, ProbIoU can produce excessively large gradients during training, causing instability; on the other hand, ProbIoU suffers from an ambiguity of orientation information when faced with square-like targets.

Due to the typically large aspect ratio of ships and their highly variable orientations in drone-view imagery, the ProbIoU loss function struggles to stably and accurately fit

their actual positions and directions during bounding box regression. To address this issue, this study introduces the SGKLD algorithm to improve the regression accuracy of rotated bounding boxes, thereby enhancing the overall ship detection performance.

SGKLD abandons the Gaussian distribution for representing OBB and instead employs a super-Gaussian distribution. The super-Gaussian distribution is an enhanced version of the Gaussian distribution modified using Lamé curves, with its implementation process illustrated in Figure 7.

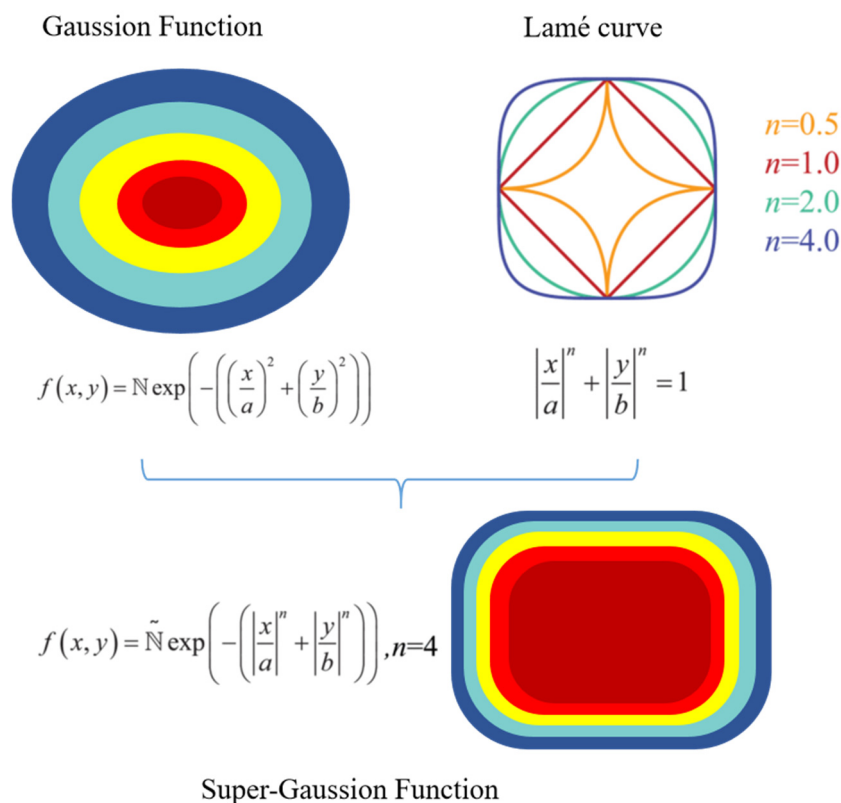


Figure 7. Formation process of the super-Gaussian distribution. $\mathbb{N}$ and $\tilde{\mathbb{N}}$ are normalization coefficients. The ellipse represents the equiprobability curve of the Gaussian distribution. By reconstructing the Gaussian function using Lamé curves, we obtain the super-Gaussian distribution, whose equiprobability curve is illustrated in the lower-right section.

The SGKLD loss dynamically adjusts the weight of angular parameters based on the aspect ratio of objects. For targets with larger aspect ratios, the model places greater emphasis on angle optimization. This mechanism is particularly crucial for high-precision ship detection, as even minor angular errors can lead to significant accuracy degradation for elongated objects like ships.

Unlike ProbIoU, SGKLD inherently avoids the orientation ambiguity problem. The key distinction lies in their equiprobability curves: SGKLD maintains an anisotropic closed curve regardless of target shape, while ProbIoU's elliptical curve becomes isotropic (degenerating to a perfect circle) when handling square-like targets. This circular symmetry in ProbIoU causes complete overlap between predicted and ground-truth distributions, making the model incapable of distinguishing their angular differences. As illustrated in Figure 8, SGKLD's super-Gaussian distribution preserves anisotropic characteristics even for square objects, enabling effective angular learning by maintaining distinguishable probability distributions between predictions and ground truths.

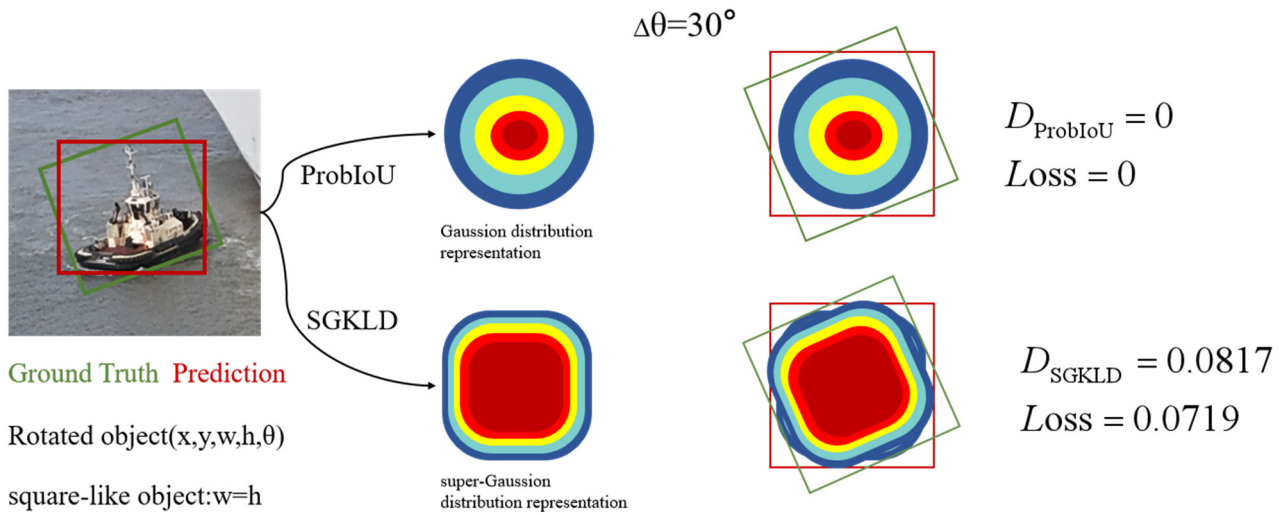


Figure 8. Difference in angle regression between ProbIoU and SGKLD for a square-like object. D and Loss represent the probability distribution distance and loss function, respectively. When detecting a square-like object, the equiprobability curve of the Gaussian distribution becomes a circle. In this case, the probability distributions of the predicted bounding box and the ground-truth box completely overlap, resulting in $D = 0$ and $\text{Loss} = 0$, thereby preventing the model from learning the object orientation. In contrast, the super-Gaussian distribution maintains anisotropy under all conditions. Consequently, D and Loss never degenerate to zero, allowing the model to effectively learn the correct directional information regardless of the object's aspect ratio.

Equations (5) and (6) defines the super-Gaussian distribution function adopted in the SGKLD loss.

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = SR \left(\begin{pmatrix} x \\ y \end{pmatrix} - \mu \right) = \text{diag} \left(\frac{w}{2}, \frac{h}{2} \right) \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \left(\begin{pmatrix} x \\ y \end{pmatrix} - \mu \right) \quad (5)$$

$$f(x, y) = \frac{n^2}{wh} \Gamma \left(\frac{1}{n} \right)^{-2} \exp(-(|\hat{x}|^n + |\hat{y}|^n)), \quad n = 4 \quad (6)$$

In this formulation, S denotes the scaling matrix; R denotes the rotation matrix; and Γ represents the Gamma function.

Equations (7) and (8) represents the computation of the Kullback–Leibler divergence between two super-Gaussian distributions as the final localization loss.

$$D_{\text{KL}}^n(\text{obj}_1 \parallel \text{obj}_2) = \log \frac{w_2 h_2}{w_1 h_1} + \frac{n^2}{w_1 h_1} \Gamma \left(\frac{1}{n} \right)^{-2} \times \iint_{\mathbb{R}} \mathbb{Z} \exp \left(- \left| \frac{2x}{w_1} \right|^n - \left| \frac{2y}{h_1} \right|^n \right) dx dy, \quad n = 4 \quad (7)$$

$$\mathbb{Z} = - \left| \frac{2x}{w_1} \right|^n - \left| \frac{2y}{h_1} \right|^n + \left| \frac{(x - \mu_x) \cos \hat{\theta} - (y - \mu_y) \sin \hat{\theta}}{w_2/2} \right|^n + \left| \frac{(x - \mu_x) \sin \hat{\theta} + (y - \mu_y) \cos \hat{\theta}}{h_2/2} \right|^n \quad (8)$$

4. Experiments

4.1. Dataset Marship-OBB9

In this study, we have a self-constructed and drone-captured ship dataset, Marship-OBB9. This dataset not only covers the ships commonly found in high-traffic maritime regions but also takes into account the visual characteristics of ships under different weather and lighting conditions so as to enhance the detection and recognition ability of the model for all kinds of ships in practical applications. Through the construction of this high-

quality dataset, we established a robust foundation for subsequent model training and experimental evaluations.

4.1.1. Image Acquisition and Preprocessing

As shown in Figure 9, to ensure the dataset aligns with real-world maritime UAV detection scenarios, we employed two UAV platforms with different spatial resolutions (DJI M3E and DJI MINI4 Pro) to collect ship images under real operational conditions. The data collection was conducted at varying altitudes (ranging from 40 m to 400 m), angles, distances, and lighting conditions, covering a wide range of high-traffic zones. To further enhance dataset diversity, we also collected supplementary images from professional ship photography websites (e.g., ShipSpotting: <http://www.shipspotting.com>, accessed on 19 March 2025) and Google Image Search.

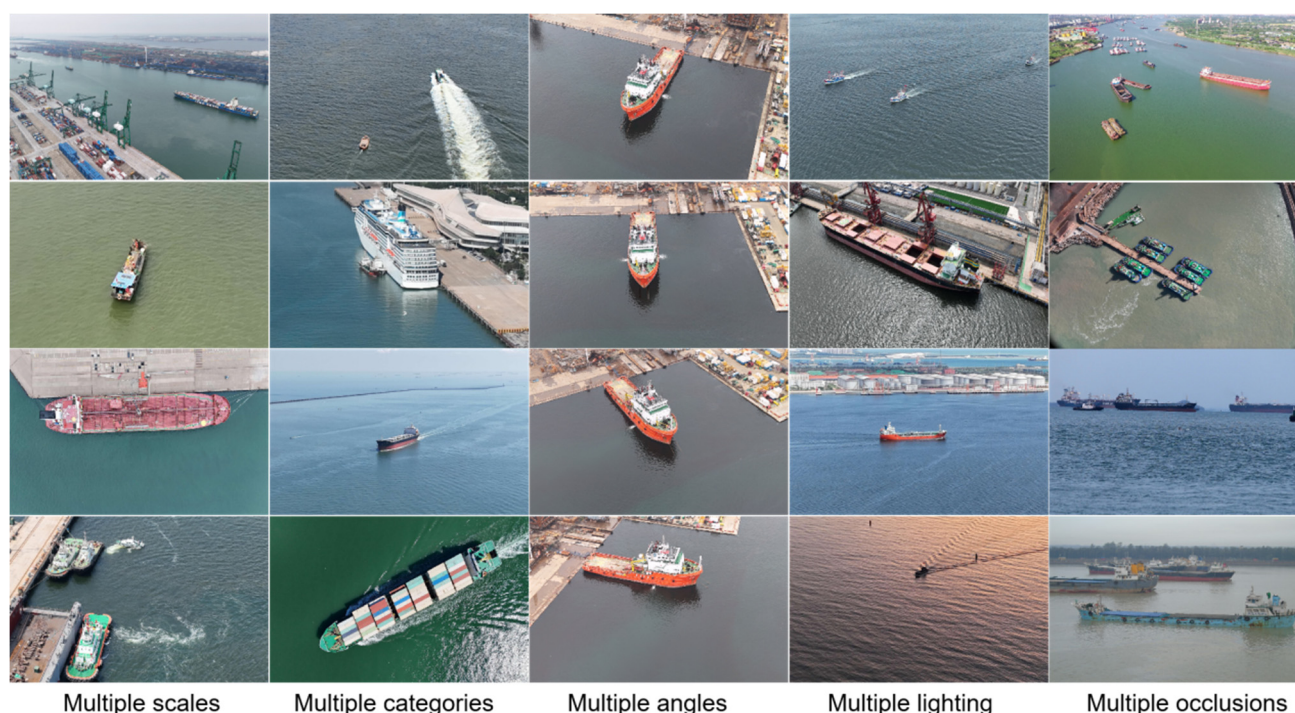


Figure 9. Example images of different scales, categories, viewing angles, lighting conditions, and occlusions.

The collected images were subjected to a series of preprocessing steps, which included (1) removing corrupted and duplicated images and (2) removing images that did not match the UAV viewpoint.

The preprocessed dataset is named Marship-OBB9. As shown in Figure 9, its main features are as follows: (1) multiple angles: contains images from different viewpoints, such as front, back, side, and top views, of ships; (2) multiple categories: covers 9 representative ship categories commonly observed in high-traffic maritime zones; (3) multiple scales: the percentage of footage of ships captured by a drone changes dynamically with height and distance; and (4) multiple scenarios: incorporates diverse conditions including occlusion, complicated backgrounds, varied lighting, and typical dense-traffic scenarios.

4.1.2. Data Annotation

We employed the X-Anylabeling annotation tool to finely label the targets in the image with HBB and OBB. Annotation results were saved in plain text (.txt) format, with each entry containing the class label and corresponding location parameters. To ensure

annotation consistency and accuracy, the following labeling guidelines were strictly enforced: (1) Bounding boxes should tightly enclose the target object, with category labels accurately matching the annotated instance; (2) for the occluded images, only the visible portion was annotated; and (3) all annotations underwent a strict review by professional volunteers to identify and correct any labeling errors or positional deviations, thereby ensuring reliability across the dataset. Based on the above process, a marine ship dataset named Marship-OBB9 was constructed to support the research of ship detection based on a UAV platform. As shown in Figure 10, the dataset contains nine categories: fishing boat (fc), general cargo vessel (gc), bulk carrier (bc), tug, passenger ships (ps), coast guard ship (cg), oil tanker (ot), container ships (cs), and other ships (e.g., sailboats, engineering vessels, etc.). The dataset contains a total of 11,268 images and 18,632 labeled objects, with a maximum of 24 instances per image. Table 2 illustrates the distribution of the number of each ship category in Marship-OBB9. Specifically, representative samples from each category are presented in Figure 10. To ensure model generalization performance, the dataset was partitioned into training, validation, and test subsets at an 8:1:1 ratio via random sampling. Strict separation was maintained between subsets to prevent sample overlap and mitigate overfitting. Figure 11 illustrates the geometric feature distribution of ship targets in the dataset. The normalized density estimation plot provides an intuitive visualization of both the spatial positions and scale characteristics of the targets.

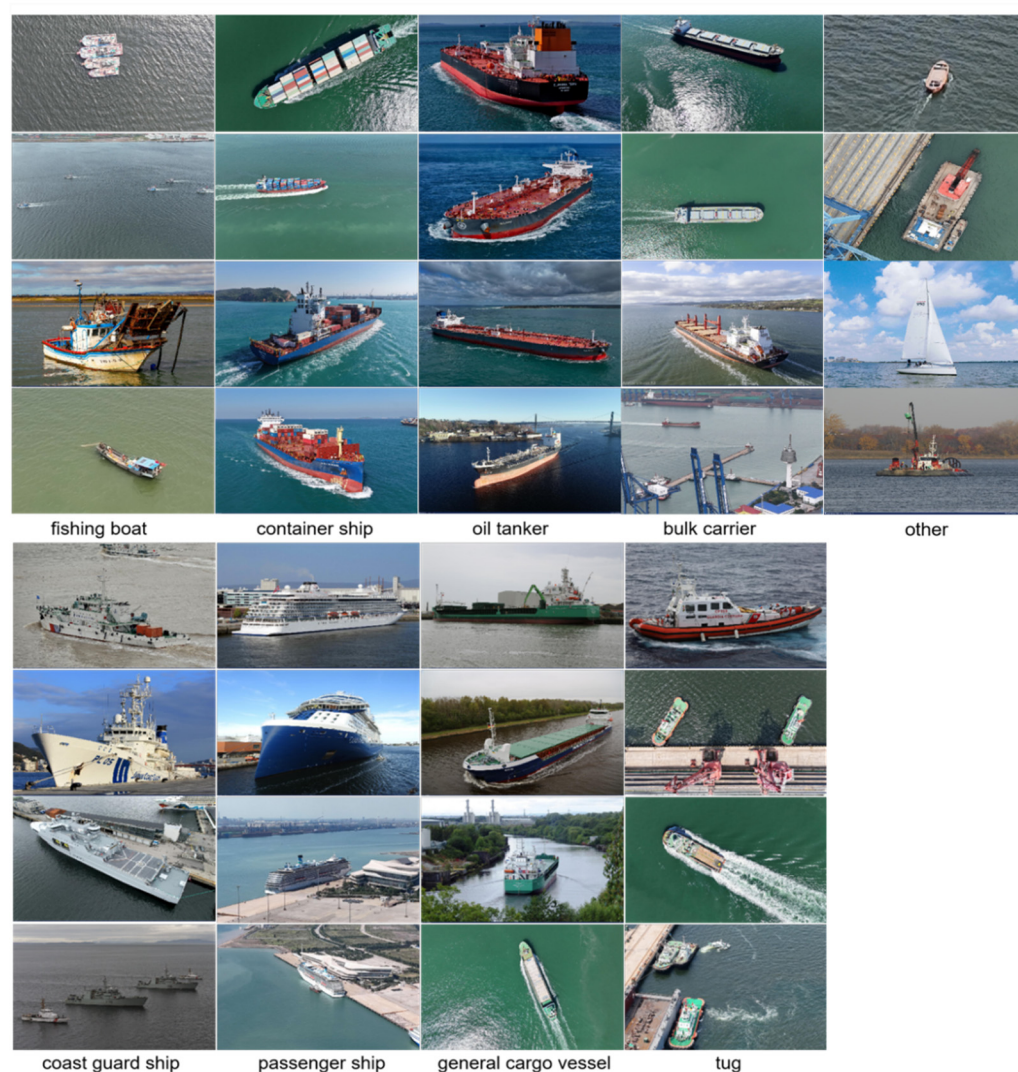
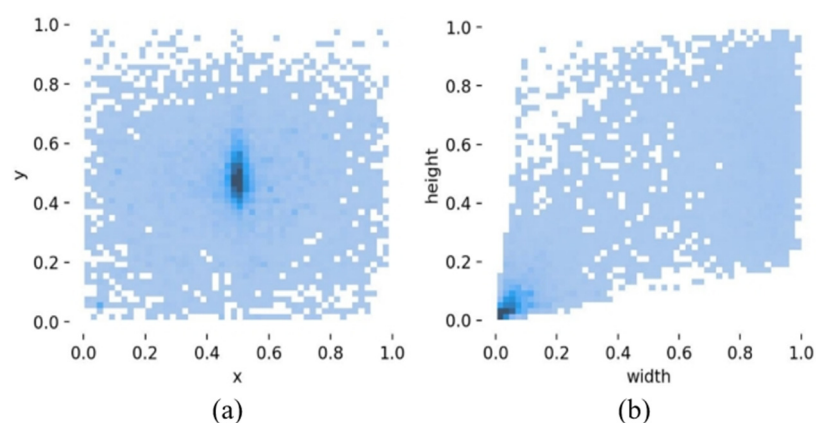


Figure 10. Examples of the Marship-OBB9 ship categories.

Table 2. Distribution of each ship category in Marship-OBB9.

Group	Category	Quantity	Ratio
#1	fishing boat	3391	0.182
#2	tug	2521	0.135
#3	general cargo vessels	2367	0.127
#4	bulk carrier	2309	0.124
#5	container ship	2196	0.118
#6	oil tanker	2013	0.108
#7	passenger ship	2001	0.107
#8	coast guard ship	970	0.052
#9	other	864	0.046

**Figure 11.** (a) The coordinate distribution diagram of the center point. (b) The width and height distribution diagram of the boundary box.

4.1.3. Evaluation Metrics

To comprehensively evaluate the performance of the proposed model, a series of metrics, including Precision, Recall, mean Average Precision (mAP), Parameters (Params), Gigaflops Per Second (GFLOPs), and Frames Per Second (FPS), were chosen to measure the model's effectiveness across different aspects.

Precision and Recall: Precision indicates the proportion of true, correct samples among all the results predicted as positive samples by the model. Recall, on the other hand, indicates the proportion of the number of positive samples correctly detected by the model to the total number of true positive samples in the category. In Equations (9) and (10), true positive (TP) denotes the case where the sample is truly positive and is also predicted as positive by the model; false positive (FP) refers to the case where the sample is truly negative but is incorrectly predicted as positive by the model; and false negative (FN) denotes the case where the sample is truly positive but is incorrectly predicted as negative by the model.

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{\text{All detections}} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{\text{All ground truth}} \quad (10)$$

Average Precision (AP): AP represents the area under the Precision–Recall (P–R) curve and serves as a key performance metric in object detection tasks. As shown in Equation (11), it quantifies the trade-off between precision and recall across different confidence thresholds.

$$AP = \int_0^1 p(r)dr \quad (11)$$

mAP: mAP refers to the mean of Average Precision (AP) across all object classes. It is commonly used to evaluate the overall detection performance of a model. As shown in Equation (12), where n denotes the total number of target classes, mAP is calculated as follows:

$$\text{mAP} = \frac{1}{n} \sum_{i=1}^n \text{AP}_i \quad (12)$$

Params and GFLOPs: Params and GFLOPs are key indicators used to assess the complexity of a model. In concrete terms, Params quantify the total number of trainable weights in the model, while GFLOPs reflect the computational cost required for a single forward pass. Generally, lower values of Params and GFLOPs indicate a more lightweight model, which is particularly advantageous for deployment on resource-constrained platforms such as UAV.

FPS: FPS measures the number of images a model can process per second, serving as a key indicator of the model's inference speed and real-time performance.

4.2. Experimental Details

The experiments in this study were conducted on the following software and hardware environment: the operating system was Ubuntu 18.04, with the deep learning framework PyTorch 2.4.1, CUDA version 12.1, and Python version 3.10.14. The hardware platform was equipped with an Intel(R) Xeon(R) Gold 6430 (manufactured by Intel Corporation, Santa Clara, CA, USA) processor and an NVIDIA RTX 4090 GPU with 24GB of VRAM (manufactured by NVIDIA Corporation, Santa Clara, CA, USA). Detailed configuration information is presented in Table 3.

Table 3. Experimental environment parameters.

Parameter	Values
Computer operating system	Ubuntu 18.04
CPU	Intel (R) Xeon (R) Gold6430
GPU	NVIDI ARTX 4090 (24 GB)
CUDA	V12.1
Python	V3.10.14
Pytorch	V2.4.1

During training, hyperparameters were optimized through a random-search-like exploration, where multiple combinations were tested and validation performance guided the final selection. The input image size was fixed at 640×640 pixels to balance accuracy and computational efficiency. A batch size of 16 was chosen to maintain stable training under memory constraints; while smaller batches may improve generalization, they often introduce instability. The initial learning rate was set to 0.01, as overly large values risk divergence and excessively small values slow convergence; prior literature and preliminary experiments confirmed that 0.01 achieves stable and efficient convergence. Training was conducted for 200 epochs, with early stopping applied to prevent overfitting once validation performance plateaued. Momentum was set to 0.937 to accelerate gradient descent and smooth parameter updates, thereby improving convergence efficiency. Stochastic Gradient Descent (SGD) was selected as the optimizer for its stable convergence and compatibility with weight decay in our dataset and model. The complete hyperparameter configuration is summarized in Table 4.

The loss curves during training are illustrated in Figure 12. The overall trend shows a continuous decrease followed by convergence to a stable level. By the end of training, the loss values remained consistently low. Throughout the entire training process, no

noticeable fluctuations in the loss were observed, and there were no evident signs of overfitting or underfitting.

Table 4. Model hyperparameter information.

Hyperparameters	Values
Images Size	640 × 640
Learning Rate	0.01
Momentum	0.937
Epochs	200
Batch Size	16
Optimizer	SGD

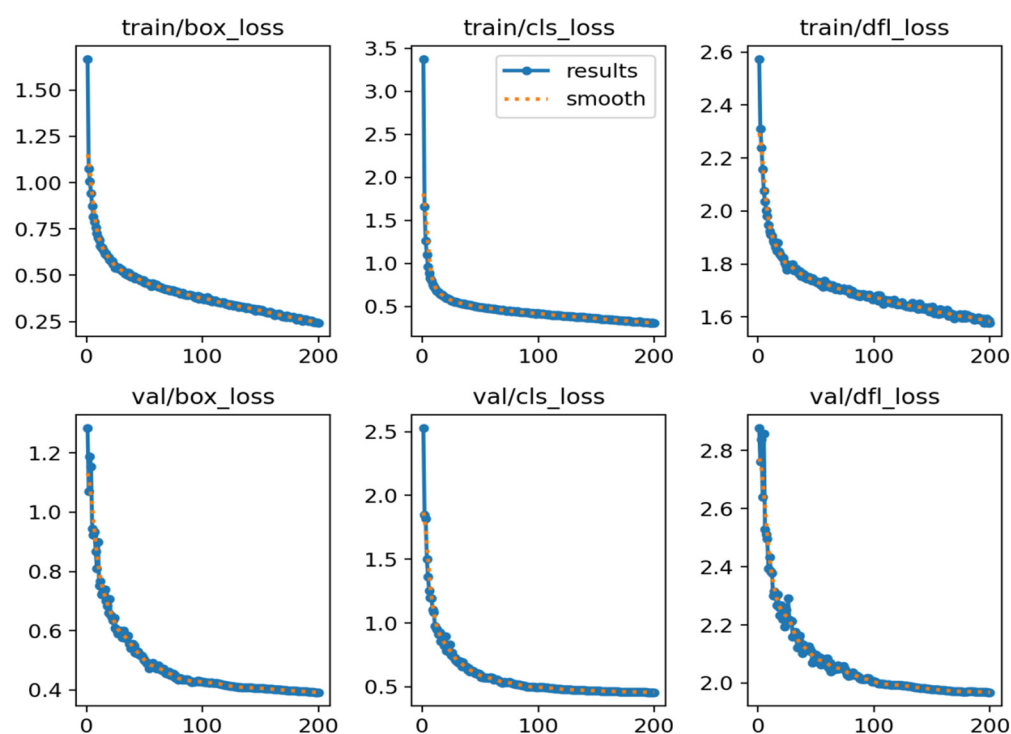


Figure 12. Network training loss function curve.

4.3. Ablation Study

To verify the effectiveness of the proposed improvement strategy, an ablation study was conducted on the Marship-OBB9 dataset. Using YOLO11n-OBB as the baseline model, we incrementally incorporated each proposed module under the same experimental settings and conducted a comprehensive comparison to analyze the performance changes before and after each improvement. As shown in Table 5, the symbol $\checkmark$ indicates that the corresponding module was added to the model.

Table 5. Results of ablation experiments. The best values are highlighted in bold.

YOLO11n-obb	DCNv4	SGKLD	P (%)	R (%)	mAP@0.5 (%)	mAP@0.5–0.95 (%)	Model Size (M)	GFLOPs (G)	FPS
$\checkmark$			88.9	82.2	86.1	71.2	5.46	6.6	94
$\checkmark$	$\checkmark$		89.5	84.4	86.7	71.5	5.57	6.7	83
$\checkmark$		$\checkmark$	90.2	82.2	86.3	71.7	5.46	6.6	100
$\checkmark$	$\checkmark$	$\checkmark$	91.0	84.5	87.8	72.8	5.57	6.7	89

The experimental results indicate that each proposed improvement strategy contributes to enhanced ship detection performance with varying efficacy. After replacing the

C3k2 module in the YOLO11n-OBb backbone with the CK_DCNv4 module, the model exhibited a slight increase in parameters and GFLOPs due to the higher computational complexity of deformable convolutions compared with standard convolutions. Nevertheless, detection accuracy improved by 0.6%, recall increased by 2.2%, mAP@0.5 rose by 0.6%, and mAP@0.5–0.95 improved by 0.3%. These findings demonstrate that deformable convolutions can effectively capture the geometric variations in ships under complex backgrounds and multi-view scenarios, achieving more robust and precise detection while maintaining a favorable balance between accuracy and efficiency. Upon substituting the original loss function in YOLO11n-OBb with SGKLD, the model demonstrated a 1.3% increase in precision, a 0.2% improvement in mAP, and a 0.5% increase in mAP@0.5–0.95, alongside enhanced inference efficiency. SGKLD contributes to more accurate localization in rotated bounding box regression. When combining both improvements, the model achieved a 2.1% gain in detection accuracy, a 1.7% increase in mAP@0.5, and a 1.6% boost in mAP@0.5–0.95, with a slight increase in model complexity.

In order to more clearly demonstrate the robustness of the improved strategy of this study, Figure 13 shows the inference results of the above different models. Four representative scenes are selected, including different lighting conditions, flight altitudes, and backgrounds. These scenes cover multi-view, dense, and occluded ship targets.

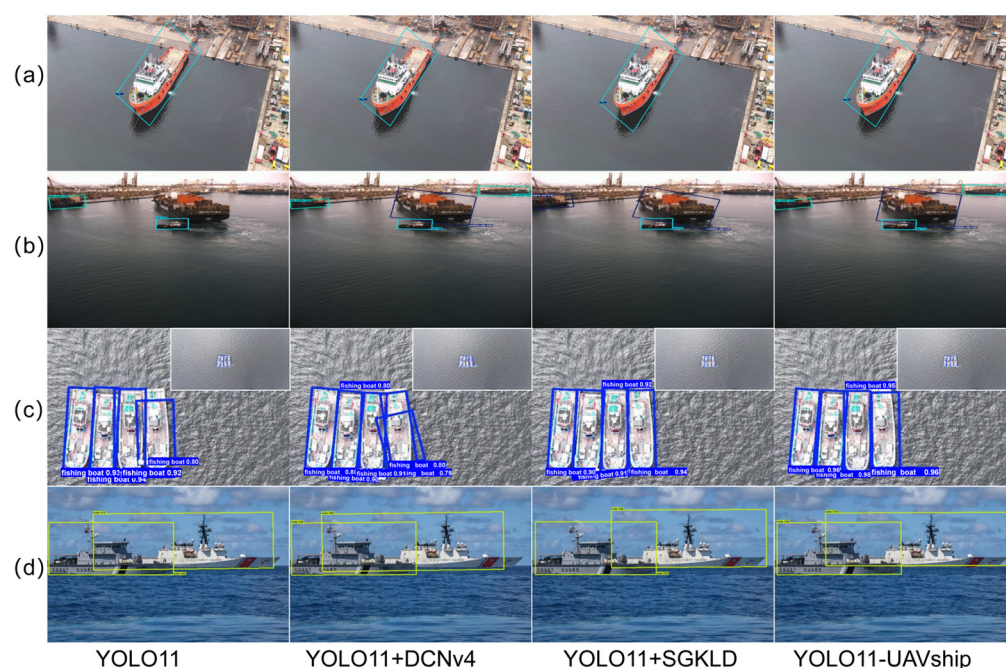


Figure 13. Visualization of ablation study result. Row (a–d) show the the ship from multiple perspectives. Rows (a,b) show the shape variations in ships caused by UAV oblique imaging, row (b) represents low light conditions, row (c) represents dense condition, row (d) represents occluded ship targets.

It can be clearly seen from the comparison of the detection results in Figure 13 that the improved YOLO11-UAVship exhibits superior ship detection performance relative to the original architecture. By comparing the images in rows (a) and (b), it can be observed that the CK_DCNv4 module more accurately predicts bounding boxes when coping with shape variations in ships caused by UAV oblique imaging, while effectively reducing the missed detection rate. Meanwhile, the comparison in rows (c) and (d) demonstrates that the SGKLD loss function improves detection accuracy in ship-intensive areas and exhibits stronger occlusion robustness.

In summary, the improved YOLO11-UAVship model shows superior adaptability and robustness for ship detection in both multi-view and dense maritime scenarios. Although the incorporation of deformable convolutions has modestly increased model complexity, the system maintains satisfactory real-time performance (89 fps), achieving an optimal balance between precision and efficiency for practical UAV applications.

4.4. Comparative Experiment

To evaluate the performance improvements of the proposed model, we conducted comparative experiments against several current advanced object detection algorithms. Table 6 shows the comparative test results of different algorithms. In this experiment, we evaluated three HBB methods and seven OBB methods. The methods encompass not only conventional CNN-based approaches, including both single-stage and two-stage object detection methods, but also incorporate increasingly prevalent transformer-based detection algorithms. The HBB models were trained on datasets annotated with standard rectangular bounding boxes, while the OBB models were trained using datasets annotated with rotated bounding boxes.

Table 6. Results of different detection methods in Marship-OBB9.

	Method	Size (Pixels)	Recall (%)	mAP@0.5 (%)	Params (M)	GFLOPs (G)	FPS
HBB	YOLOv8n	640	79.5	80.7	3.01	8.1	133
	YOLO11n	640	80.4	81.5	2.58	6.3	110
	RT-DETR [52]	640	86.5	86.3	32.00	103.5	36
OBB	rotated-faster-rcnn [33]	640	73.3	68.04	41.13	91.01	15
	RoI-Transformer [21]	640	77.4	74.3	55.08	104.96	13
	Redet [19]	640	82.7	80.7	31.60	48.3	6
	Retinanet-obbb	640	80.6	71.9	36.29	83.28	23
	S2Anet	640	80.7	79.1	38.57	76.96	15
	YOLOv8n-OBB	640	81.4	84.9	2.76	7.1	108
	YOLO11n-OBB	640	82.2	86.1	2.66	6.6	94
	Ours	640	84.5	87.8	2.72	6.7	89

The results show that, within the HBB group, YOLOv8n achieves 80.7% mAP@0.5 and 79.5% recall, while YOLO11n achieves 81.5% mAP and 80.4% recall. RT-DETR achieved the highest precision in this group with an mAP of 86.3% and a recall of 86.5%, but at the cost of a substantially increased model complexity—32 million parameters, 103.5 GFLOPs, and an inference time of 27.9 ms—making it less suitable for real-time or resource-constrained applications. In the OBB group, models such as Rotated-Faster-RCNN, RoI-Transformer, and ReDet showed moderate detection accuracy but suffered from extremely low FPS. YOLOv8n-OBB and YOLO11n-OBB achieved mAP values of 84.9% and 86.1%, respectively, which are better than YOLOv8n and YOLO11n. However, the FPS of YOLOv8n-OBB and YOLO11n-OBB are lower than YOLOv8n and YOLO11n, which shows that after the introduction of the rotation box, the detection accuracy of the model is improved, but the detection efficiency is slightly reduced. Our proposed method further pushes mAP@0.5 to 87.8% with a recall of 84.5%, achieving the highest detection performance among all compared models. In general, under the condition of maintaining real-time performance and lightweight deployment, the single-stage detection framework proposed in this paper, which combines the rotation bounding box mechanism with the improved module, can achieve superior detection effects in complex marine scenes.

To further test the robustness of the proposed detection algorithm in identifying different types of ships, we provide the mAP@0.5 results for each ship category using

various detection methods, as detailed in Table 7. The results demonstrate that the proposed YOLO-UAVship model consistently outperforms existing methods across all ship categories. Compared to the benchmark YOLO11n-OBB, our model achieves an improvement in detection accuracy ranging from 0.4% to 2.7% across different ship types. The model demonstrates strong performance across most categories. Specifically, for large, structurally stable, and visually distinctive ship types such as container ships, general cargo vessels, oil tankers, bulk carriers, passenger ships, and coast guard ships, the mAP exceeds 95%, which shows that our algorithm has a strong ability to identify and locate these types of targets. In contrast, the mAP for fishing boats and tugs is slightly lower, which may be attributed to their smaller size and greater difference in shape and appearance, making them more challenging to detect. Notably, the performance in the “other” category is suboptimal. This can be explained by several factors. First, the “other” class exhibits high internal heterogeneity, encompassing diverse ship types such as engineering vessels, government ships, small yachts, and sailboats, which differ markedly in structural appearance and scale. The lack of consistent visual patterns makes it difficult for the model to learn effective decision boundaries. Second, the “other” class has the fewest samples in the dataset (accounting for only 4.6%), making it susceptible to long-tailed effects, which in turn limit the model’s performance. At last, during the annotation process, this category often serves as a fallback for ships that cannot be clearly classified, introducing semantic ambiguity and label inconsistency, which further hinders the model’s learning.

Table 7. Comparison of test results (mAP0.5 (%)) for each category with different methods. The best values are highlighted in bold. The first line is the abbreviation of each ship type, such as fb for fishing boat.

Category	fb	tug	gc	bc	cs	ot	ps	cg	Other
YOLOv8n	90.7	92.7	93.2	96.7	94.3	97.0	94.9	97.8	51.7
YOLO11n	91.0	92.9	95.3	98.4	95.3	98.7	94.9	97.5	52.1
RT-DETR	91.7	93.9	95.5	96.3	93.8	97.2	94.5	98.1	55.3
rotated-faster-rcnn	62.0	65.4	72.4	80.2	77.2	76.7	80.6	78.8	19.0
RoI-Transformer	68.5	76.9	84.2	85.0	84.7	82.4	81.5	77.7	27.6
redet	79.0	80.9	88.5	89.5	89.9	87.6	90.1	88.5	32.7
Retinanet-obb	70.1	69.7	79.3	77.5	80.3	80.9	83.6	82.0	24.4
S2Anet	77.0	79.5	86.4	86.1	87.4	80.1	89.6	79.0	46.7
YOLOv8n-OBB	92.0	93.1	95.8	96.7	94.7	97.6	95.8	98.3	48.7
YOLO11n-OBB	91.7	92.9	95.9	96.6	94.6	98.0	95.7	98.4	55.0
Our	92.1	94.7	96.1	98.9	96.9	99.1	96.7	98.5	57.7

Figure 14 presents the visualized detection results of different algorithms in complicated and ship-intensive areas. The analysis reveals that YOLOv8n, YOLO11n, and RT-DETR generate horizontal bounding boxes that include a substantial amount of non-target background, leading to frequent missed and false detections in crowded areas. Faster R-CNN, ROI-Transformer, and ReDet exhibit redundant bounding boxes, indicating over-detection. Furthermore, YOLOv8n-OBB and YOLO11n-OBB suffer from misclassifications; for example, YOLO11n-OBB erroneously classifies background regions as tugboats. In contrast, the proposed method in this study accurately localizes and classifies ships, achieving high-precision recognition across multiple ship categories and thus verifying the accuracy of the proposed improvements.

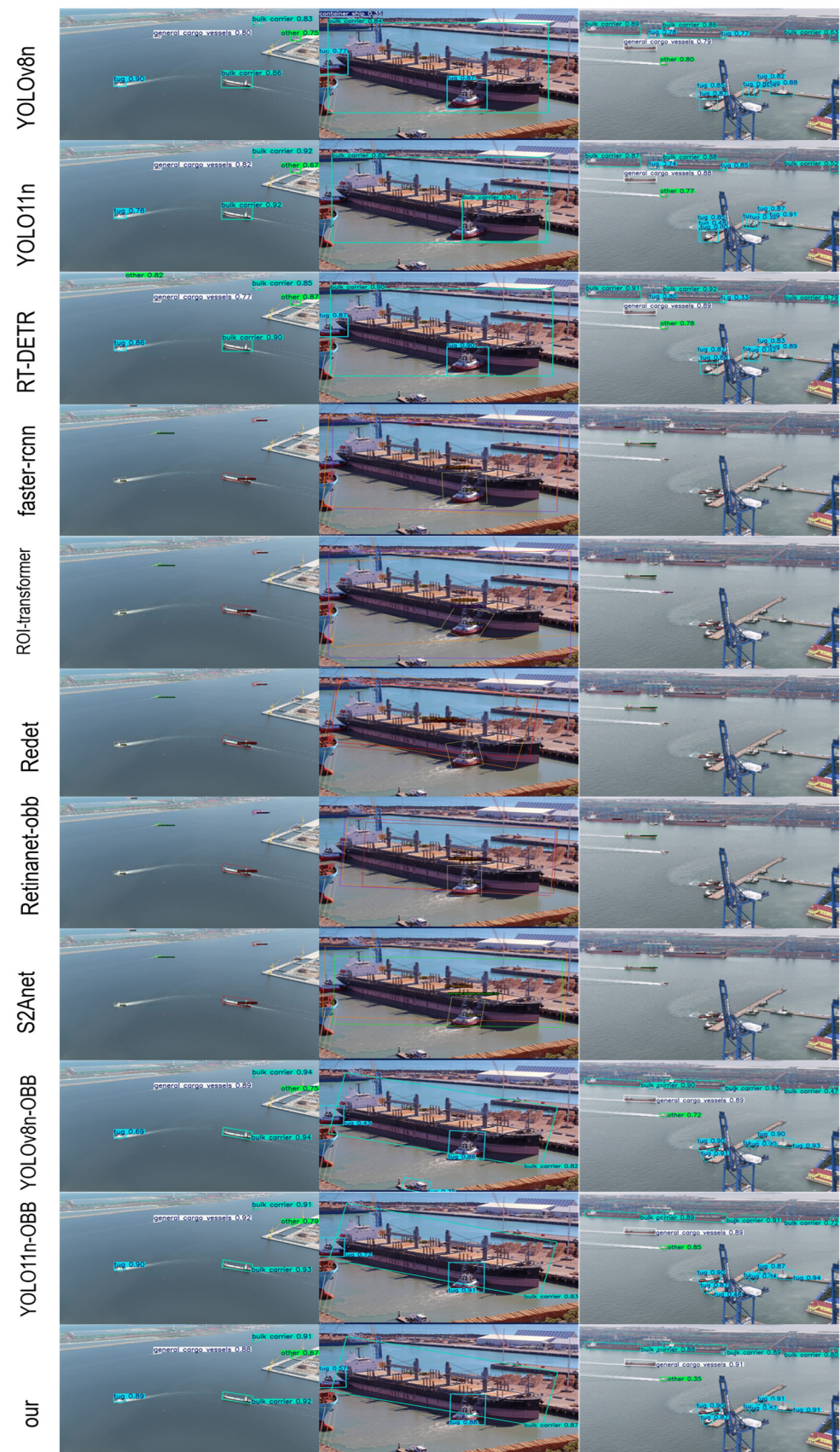


Figure 14. Comparison of detection results of different methods in complicated scenarios.

5. Discussion

Through both ablation and comparative experiments, our proposed method—which integrates a rotated bounding box detection mechanism, designs the CK_DCNv4 module, and introduces the SGKLD loss for robust rotated box regression—demonstrates significant improvements in adaptability and detection accuracy under diverse perspectives, geometric distortions, and occlusion scenarios.

To further assess the robustness of YOLO-UAVShip, we also conducted evaluations in additional challenging maritime environments. These results demonstrate that, despite the model's strong performance, certain limitations remain to be addressed.

First, as illustrated in Figure 15, the model exhibits limited performance in detecting extremely small or distant targets that frequently appear in UAV imagery. This limitation is largely attributable to the inherently low resolution of such objects and the insufficient contextual information available for reliable identification. Second, UAV imagery often involves highly complex environments, such as nearshore port areas where docks, cranes, and other infrastructure exhibit visual characteristics similar to ships. This visual similarity can mislead the detector and result in false positives, highlighting the challenge of robust detection in cluttered maritime scenes.



Figure 15. Visualization of false negatives and false positives. Red circles indicate missed detections, while yellow circles indicate false detections.

6. Conclusions

UAV ship detection fulfills a critical function in enhancing maritime traffic safety and facilitating efficient maritime supervision. Aiming at the problem of UAV ship detection in complicated, high-density navigational areas, we propose an improved detection algorithm named YOLO-UAVship, based on the YOLO11 architecture. First, an oriented bounding box detection mechanism is incorporated to precisely fit ship contours and reduce background interference. Second, a newly designed CK_DCNv4 module, integrating deformable convolution v4 (DCNv4) and a C3k2 backbone structure, is developed to enhance geometric feature extraction under aerial oblique view. At last, for ships with large aspect ratios, SGKLD effectively addresses the localization challenges in dense environments, achieving robust position regression. In addition, we construct a diverse and high-quality UAV ship detection dataset, Marship-OBB9, which provides a solid data foundation for future research on maritime ship detection from a UAV perspective. Experimental results demonstrate that our method substantially improves detection accuracy and robustness compared to the baseline model while maintaining competitive inference efficiency. The algorithm achieves an optimal balance between precision and computational cost, demonstrating considerable promise for maritime applications such as port surveillance, maritime law enforcement, and search-and-rescue operations.

Future research will focus on the following directions: First, reconstructing the “other” category using a fine-grained classification strategy, while employing generative adversarial network (GAN) for data augmentation to improve the consistency of samples within category; Next, designing a hybrid network architecture that combines multi-scale feature pyramids (FPN) with adaptive attention mechanisms, aiming to enhance the extraction of geometric features for small and distant ship targets; At last, exploring more lightweight and efficient network designs that maintain strong feature representation capabilities, thereby improving the deployment adaptability of the algorithm in real-world applications.

Author Contributions: Conceptualization, Y.L. and C.Y.; data curation, Y.L., F.L. and K.Y. (Kun Yu); formal analysis, Y.L.; investigation, Y.L. and C.Y.; methodology, Y.L., H.H. and C.Y.; resources, Y.L., C.Y. and Y.T.; supervision, C.Y.; visualization, Y.L.; writing—original draft, Y.L.; writing—review and editing, Y.L., C.Y., Y.T., G.Y., K.Y. (Kai Yin) and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the “Maritime Science and Technology Key Project Preliminary Basic Research Project” of the Maritime Safety Administration of the People’s Republic of China (E3DZ113401) and the “Research and Development of Key Technologies and Equipment for the New Generation of Marine Intelligent Transportation Management System” of the Hebei Transportation Investment Group Company Limited (E3E2113601).

Data Availability Statement: The datasets and code are available from the corresponding author upon reasonable request.

Acknowledgments: The authors are grateful to Xinyi Feng and Yingqi Wang for their valuable support and constructive suggestions throughout this work.

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

References

1. MerchantFleet. Available online: <https://unctadstat.unctad.org/datacentre/dataviewer/US.MerchantFleet> (accessed on 26 June 2025).
2. Feng, S.; Huang, Y.; Zhang, N. An Improved YOLOv8 OBB Model for Ship Detection through Stable Diffusion Data Augmentation. *Sensors* **2024**, *24*, 5850. [\[CrossRef\]](#)
3. Gonçalves, L.; Damas, B. Automatic Detection of Rescue Targets in Maritime Search and Rescue Missions Using UAVs. In Proceedings of the 2022 International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, Croatia, 21–24 June 2022; pp. 1638–1643.
4. Liu, Y.; Yan, J.; Zhao, X. Deep Reinforcement Learning Based Latency Minimization for Mobile Edge Computing with Virtualization in Maritime UAV Communication Network. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4225–4236. [\[CrossRef\]](#)
5. Cheng, S.; Zhu, Y.; Wu, S. Deep Learning Based Efficient Ship Detection from Drone-Captured Images for Maritime Surveillance. *Ocean Eng.* **2023**, *285*, 115440. [\[CrossRef\]](#)
6. Robicquet, A.; Sadeghian, A.; Alahi, A.; Savarese, S. Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 549–565.
7. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7380–7399. [\[CrossRef\]](#)
8. Zhao, C.; Liu, R.W.; Qu, J.; Gao, R. Deep Learning-Based Object Detection in Maritime Unmanned Aerial Vehicle Imagery: Review and Experimental Comparisons. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107513. [\[CrossRef\]](#)
9. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. *IEEE Trans. Multimed.* **2018**, *20*, 2593–2604. [\[CrossRef\]](#)
10. 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. [\[CrossRef\]](#)
11. Zheng, Y.; Zhang, S. Mcships: A Large-Scale Ship Dataset for Detection and Fine-Grained Categorization in the Wild. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
12. Iancu, B.; Soloviev, V.; Zelioli, L.; Lilius, J. ABOships—An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations. *Remote Sens.* **2021**, *13*, 988. [\[CrossRef\]](#)

13. Sun, X.; Wang, P.; Yan, Z.; Xu, F.; Wang, R.; Diao, W.; Chen, J.; Li, J.; Feng, Y.; Xu, T.; et al. FAIR1M: A Benchmark Dataset for Fine-Grained Object Recognition in High-Resolution Remote Sensing Imagery. *ISPRS J. Photogramm. Remote Sens.* **2022**, *184*, 116–130. [\[CrossRef\]](#)
14. 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. [\[CrossRef\]](#)
15. Wang, N.; Wang, Y.; Wei, Y.; Han, B.; Feng, Y. Marine Vessel Detection Dataset and Benchmark for Unmanned Surface Vehicles. *Appl. Ocean Res.* **2024**, *142*, 103835. [\[CrossRef\]](#)
16. Yu, C.; Yin, H.; Rong, C.; Zhao, J.; Liang, X.; Li, R.; Mo, X. YOLO-MRS: An Efficient Deep Learning-Based Maritime Object Detection Method for Unmanned Surface Vehicles. *Appl. Ocean Res.* **2024**, *153*, 104240. [\[CrossRef\]](#)
17. Han, J.; Ding, J.; Li, J.; Xia, G.-S. Align Deep Features for Oriented Object Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5602511. [\[CrossRef\]](#)
18. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: New York, NY, USA, 2017; pp. 2999–3007.
19. Han, J.; Ding, J.; Xue, N.; Xia, G.-S. ReDet: A Rotation-Equivariant Detector for Aerial Object Detection. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2 November 2021; IEEE: Nashville, TN, USA, 2021; pp. 2785–2794.
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.U.; Polosukhin, I. Attention is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
21. Ding, J.; Xue, N.; Long, Y.; Xia, G.-S.; Lu, Q. Learning RoI Transformer for Oriented Object Detection in Aerial Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
22. Dai, L.; Liu, H.; Tang, H.; Wu, Z.; Song, P. AO2-DETR: Arbitrary-Oriented Object Detection Transformer. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 2342–2356. [\[CrossRef\]](#)
23. Zeng, Y.; Chen, Y.; Yang, X.; Li, Q.; Yan, J. ARS-DETR: Aspect Ratio-Sensitive Detection Transformer for Aerial Oriented Object Detection. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5610315. [\[CrossRef\]](#)
24. Yang, X.; Yan, J.; Ming, Q.; Wang, W.; Zhang, X.; Tian, Q. Rethinking Rotated Object Detection with Gaussian Wasserstein Distance Loss. In Proceedings of the 38th International Conference on Machine Learning, Online, 18–24 July 2021; PMLR: New York, NY, USA, 2021; pp. 11830–11841.
25. Yang, X.; Yang, X.; Yang, J.; Ming, Q.; Wang, W.; Tian, Q.; Yan, J. Learning High-Precision Bounding Box for Rotated Object Detection via Kullback-Leibler Divergence. In Proceedings of the 35th International Conference on Neural Information Processing Systems, Online, 6–14 December 2021; Curran Associates, Inc.: New York, NY, USA, 2021; Volume 34, pp. 18381–18394.
26. Zhang, B.; Liu, J.; Liu, R.W.; Huang, Y. Deep-Learning-Empowered Visual Ship Detection and Tracking: Literature Review and Future Direction. *Eng. Appl. Artif. Intell.* **2025**, *141*, 109754. [\[CrossRef\]](#)
27. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
28. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [\[CrossRef\]](#)
29. Chen, B.; Liu, L.; Zou, Z.; Shi, Z. Target Detection in Hyperspectral Remote Sensing Image: Current Status and Challenges. *Remote Sens.* **2023**, *15*, 3223. [\[CrossRef\]](#)
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [\[CrossRef\]](#)
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
32. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
33. 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.* **2017**, *39*, 1137–1149. [\[CrossRef\]](#)
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York, NY, USA, 2016; pp. 779–788.

36. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
37. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. [[CrossRef](#)]
38. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934. [[CrossRef](#)]
39. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
40. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
41. Zwemer, M.H.; Wijnhoven, R.G.J.; de With, P.H.N. Ship Detection in Harbour Surveillance Based on Large-Scale Data and CNNs. In Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018), Funchal, Madeira, 27–29 January 2018; Volume 5, pp. 153–160. [[CrossRef](#)]
42. Wang, Y.; Ning, X.; Leng, B.; Fu, H. Ship Detection Based on Deep Learning. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 275–279.
43. Tang, G.; Liu, S.; Fujino, I.; Claramunt, C.; Wang, Y.; Men, S. H-YOLO: A Single-Shot Ship Detection Approach Based on Region of Interest Preselected Network. *Remote Sens.* **2020**, *12*, 4192. [[CrossRef](#)]
44. Li, H.; Deng, L.; Yang, C.; Liu, J.; Gu, Z. Enhanced YOLO v3 Tiny Network for Real-Time Ship Detection from Visual Image. *IEEE Access* **2021**, *9*, 16692–16706. [[CrossRef](#)]
45. Guo, F.; Ma, H.; Li, L.; Lv, M.; Jia, Z. FCNet: Flexible Convolution Network for Infrared Small Ship Detection. *Remote Sens.* **2024**, *16*, 2218. [[CrossRef](#)]
46. Dolgoplov, A.V.; Kazantsev, P.A.; Bezuhliy, N.N. Ship Detection in Images Obtained from the Unmanned Aerial Vehicle (UAV). *Indian J. Sci. Technol.* **2016**, *9*, 1–7. [[CrossRef](#)]
47. Wang, Q.; Wang, J.; Wang, X.; Wu, L.; Feng, K.; Wang, G. A YOLOv7-Based Method for Ship Detection in Videos of Drones. *J. Mar. Sci. Eng.* **2024**, *12*, 1180. [[CrossRef](#)]
48. Han, Y.; Guo, J.; Yang, H.; Guan, R.; Zhang, T. SSMA-YOLO: A Lightweight YOLO Model with Enhanced Feature Extraction and Fusion Capabilities for Drone-Aerial Ship Image Detection. *Drones* **2024**, *8*, 145. [[CrossRef](#)]
49. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725. [[CrossRef](#)]
50. Llerena, J.M.; Zeni, L.F.; Kristen, L.N.; Jung, C. Gaussian Bounding Boxes and Probabilistic Intersection-over-Union for Object Detection. *arXiv* **2021**, arXiv:2106.06072. [[CrossRef](#)]
51. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
52. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-Time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.

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.