

Article

Shipwake-YOLO: Ship Wake Detection and Instance Segmentation for Visual Navigational-State Cue Extraction

Shaoxi Li ¹, Xingchen Ji ^{1,2}, Chuankao Yang ¹ and Ruolan Zhang ^{1,*}¹ Navigation College, Dalian Maritime University, Dalian 116026, China; lishaoxi@dlmu.edu.cn (S.L.); jxc970606@gmail.com (X.J.); yck@dlmu.edu.cn (C.Y.)² Graduate School of Maritime Sciences, Kobe University, Kobe 658-0022, Japan

* Correspondence: zhangruolan@dlmu.edu.cn

Abstract

Visual perception is an important component of close-range maritime situational awareness, particularly when conventional sources such as AIS and radar are delayed, incomplete, or unavailable. Ship wakes provide motion-related visual cues, but their segmentation remains difficult because wake regions are elongated, weakly textured, and frequently mixed with water-surface clutter. This study develops Shipwake-YOLO, a wake-oriented adaptation of YOLOv9-Seg for ship and wake instance segmentation in inland-waterway images. The task is formulated as visual navigational-state cue extraction rather than validated future manoeuvre prediction. The model segments hull and wake instances and provides mask-derived spatial cues for possible downstream state interpretation. The architecture introduces iAFF into cross-scale feature fusion, adapts the high-level SPPELAN aggregation block with SAConv-enhanced convolution, replaces selected downsampling paths with iSACADown, and adopts MPD-IoU as the bounding-box regression loss. On a 2100-image dataset collected from the Wuhu Channel of the Yangtze River, Shipwake-YOLO improves Box-mAP@50 from 77.7% to 84.6% and Mask-mAP@50 from 67.6% to 79.8% relative to the YOLOv9-Seg baseline. Under stricter IoU thresholds, the model reaches 48.9 in Box-mAP@[0.50:0.95] and 46.2 in Mask-mAP@[0.50:0.95]. The parameter count is reduced by 7.5%, and GFLOPs decrease from 144.2 to 137.1. These results indicate that the proposed adaptation improves ship-wake perception within the collected inland-waterway setting and provides a visual basis for downstream navigational-state estimation.

Keywords: ship wake; instance segmentation; small target detection; feature extraction



Academic Editor: Simone Mancini

Received: 3 June 2026

Revised: 22 June 2026

Accepted: 26 June 2026

Published: 30 June 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The identification and prediction of vessel behaviour are pivotal for maritime transportation and navigation safety. They aid in collision avoidance and mitigating other maritime incidents, ensuring the safety of crew and cargo [1]. By discerning the behaviour of other vessels, ships can take appropriate measures to avert collisions and enact emergency protocols when necessary. Understanding the behaviour of other vessels facilitates navigation optimization. By forecasting the trajectories and actions of other vessels, ships can select optimal routes and speeds, mitigating traffic congestion, saving fuel, and enhancing navigation efficiency. The recognition and prediction of vessel behaviour assist emergency and rescue services in promptly responding to maritime incidents [2]. It can utilize this information to forecast the potential location and impact of accidents, undertaking suitable measures to reduce/mitigate losses.

As a traditional tool for situational awareness of surrounding vessels, AIS has wide applications in the maritime field. Gao et al. [3] developed an online real-time ship behaviour prediction model by constructing a Bidirectional Long Short-Term Memory Recurrent Neural Network (BI-LSTM-RNN) suitable for AIS date and time series features. The model was utilized for online real-time predictions. Results demonstrated that BI-LSTM-RNN effectively predicted ship navigation behaviours. Yan et al. [4] proposed a method for extracting ship docking information based on AIS data, integrating trajectory features and geographic scenario semantics. Initially, under the constraints of port geographic knowledge, ship trajectory features were excavated, enabling the identification of docking points within port areas. This method efficiently extracted ship docking information, offering knowledge support for understanding ship behaviours and maritime traffic analysis. Ljunggren et al. [5] employed three baseline methods to classify ship trajectories for ship type identification. Evaluation results revealed that the classifier based on deep learning performed optimally.

In the maritime domain, especially in inland waterways and areas with poor water clarity, vessel crews often rely on visual cues, such as wake patterns, to gauge the behaviour of nearby vessels. Experienced crew members can swiftly and intuitively predict the potential movement direction of a target vessel based on its wake [6]. This reliance on visual cues underscores the importance of vessel behaviour identification and prediction, particularly in scenarios where traditional methods may be hindered by environmental factors.

Traditional methods typically rely on the experience and intuition of seafarers to analyze and forecast vessel motion by observing factors such as the shape, length, direction, and speed of a vessel's wake [7]. Common analytical approaches include observing wake shape, length, direction, and changes to infer potential vessel motion and intent. However, this approach is subject to subjective interpretation and uncertainty due to its reliance on the personal experience and skills of the crew, and it relies solely on limited visual information, making it challenging to provide accurate predictions in complex maritime environments.

The existing literature indicates that ship wakes are an important tool for situational awareness of surrounding vessels. Nielsen et al. [8] employed wave buoy analogues to analyze ship-induced motions and subsequently calculate directional wave spectra. This method can accurately estimate sea state parameters by considering simple and cost-effective ship instruments, based on measurements from motion response units installed near the ship's forward perpendicular. Almström et al. [9] compared ship waves simulated by XBeach with high-quality field measurements to predict hydrodynamics of ship waves along waterways accurately. Additionally, Ma et al. [10] investigated the influence of primary ship dimensions on the generation of stable waves and proposed practical methods to inversely predict the ship's length, width, draught, and related parameters. Their findings indicate a correlation between ship length and the quasi-period of the divergent wave amplitude function. Based on the considered two-point ship stern model, the predictive accuracy of ship length can be achieved to a certain extent overall.

Another limitation of traditional methods arises in adverse weather conditions or poor visibility, which results in their accuracy and reliability being compromised [11]. Moreover, traditional methods fail to fully utilize other available perceptual data sources, such as radar, AIS, and satellite imagery. Hence, there is a pressing need to develop more accurate and reliable techniques to supplement or replace traditional methods, thereby enhancing the accuracy and reliability of vessel behaviour identification and prediction.

With the advent of smart and autonomous vessels, there is a growing need to integrate computer vision-based perception systems into maritime operations [12]. Recent JMSE studies have also shown that shipping-domain knowledge can be incorporated into computer vision models for maritime transportation, and that multi-scale object detection models can

support autonomous ship navigation in complex maritime environments [13,14]. However, one area that remains relatively unexplored is the utilization of computer vision for wake pattern recognition. Incorporating visual wake recognition into vessel perception systems could significantly enhance their ability to characterize the visual motion-related cues of surrounding vessels, thereby improving navigation safety and efficiency. By leveraging computer vision techniques to analyze wake patterns, intelligent vessels will be able to supplement or even replace human intuition in predicting vessel behaviour.

Existing investigations reveal that most researchers have focused on integrating AIS with artificial intelligence. Recent JMSE work has further explored deep-learning-based vessel trajectory prediction using AIS data, such as dual-path spatial-temporal attention networks with multi-attribute information [15]. Volkova et al. [16] explored the use of neural networks to predict ship trajectories using AIS data. The study highlighted the importance of considering vessel inertia and external conditions in trajectory predictions. Shi et al. [17] addressed the detection of abnormal ship behaviours using AIS data to enhance maritime safety. It used various data-driven techniques to model and detect unusual movements or patterns that could indicate potential risks or navigational errors. This study provided a comprehensive approach to understanding and mitigating factors leading to maritime accidents. Murray and Perera [18] presented a deep learning framework intended to enhance maritime safety by predicting regional ship behaviours using historical AIS data. The framework aimed to support proactive collision avoidance, addressing both the technical and operational aspects of maritime traffic systems.

In summary, this study focuses on ship and wake instance segmentation as a visual perception problem for inland-waterway navigation. Rather than directly predicting future collision-avoidance manoeuvres, the proposed framework extracts hull and wake masks and derives image-level spatial cues that may support downstream navigational-state interpretation. This scope is consistent with COLREGs Rule 5 on proper lookout, where visual observation forms one component of situational awareness. For Maritime Autonomous Surface Ships, reliable perception of hull-wake structures can supplement conventional sensors under close-range conditions, but temporal intention recognition and future manoeuvre prediction require synchronized navigation sensors and labelled motion states beyond the image-level dataset used in this work.

Object detection and instance segmentation technologies based on computer vision play a crucial role in identifying ship wakes. These technologies allow for the precise localization and classification of objects within an image, enabling the accurate identification of ship wakes amid various environmental conditions.

This paper develops Shipwake-YOLO as a task-oriented adaptation of YOLOv9-Seg for ship and wake instance segmentation in inland-waterway visual images. The methodological contribution does not lie in proposing iAFF, SAConv, or MPD-IoU as new standalone operators; rather, it lies in placing these established components at resolution-sensitive and context-sensitive stages of the YOLOv9-Seg pipeline to address the visual properties of ship wakes. Specifically, iAFF is introduced into cross-scale feature fusion to suppress redundant water-surface responses while retaining weak wake textures; SAConv is embedded into the downsampling and pyramid aggregation modules to expand the effective receptive field without a proportional increase in computational cost; and MPD-IoU is adopted as the bounding-box regression loss to improve geometric alignment for elongated ship and wake regions. This task-specific recombination yields a lightweight perception model tailored to low-contrast, diffuse, and scale-varying wake patterns, and its effectiveness is examined through comparisons with representative instance segmentation baselines and ablation variants. The effectiveness of this model is attributed to three main novel contributions:

- (1) Integration of SPPELAN with iAFF and SAC convolution modules: This combination effectively retains cross-channel information while reducing computational demands, ensuring efficient and comprehensive feature analysis.
- (2) iSACADown: By integrating the iAFF attention mechanism, the new framework can achieve the accuracy and speed of ship wake detection while reducing the overall parameter count of the model. This balance between efficiency and effectiveness is crucial for real-time maritime navigation applications.
- (3) Adoption of the MPD-IoU loss function: The introduction of the MPD-IoU loss function significantly enhances target recognition capabilities. This enhancement improves the regression accuracy of bounding boxes, enabling precise identification of ships, and markedly improving overall detection performance.

The structure of this paper is as follows. Section 2 provides an overview of the related work. Section 3 presents the proposed Shipwake-YOLO framework and its key modules. Section 4 reports the experimental results and ablation studies. Section 5 discusses the findings and limitations, and Section 6 concludes the paper and outlines future work.

2. Literature Review

AIS serves as a pivotal maritime surveillance technology, facilitating the exchange of crucial data such as location, course, speed, and vessel type between ships and shore stations. Although AIS does not inherently possess predictive capabilities for vessel behaviour, the rich information it provides underpins complex behaviour analyses and navigational decision-making processes. By collecting dynamic and static information, AIS not only tracks the immediate and historical voyage paths of vessels but also enables the analysis of these data to forecast future positions and behaviours, such as directional changes and speed adjustments [19].

There have been numerous research outcomes on obtaining situational awareness data of surrounding vessels through radar and AIS. Chen et al. [20] explored the use of radar data in the management of maritime traffic, focusing on how radar technology helps predict ship behaviours and trajectories in regional waters. Radar data, including the continuous tracking of ships' positions, speeds, and headings, could be analyzed to predict movements and potential conflicts. Wang et al. [21] delved into various data-driven methods, including radar, for detecting abnormal ship behaviour, discussing advancements and applications in maritime safety. The focus would be on how radar data contributes to identifying deviations from normal ship behaviour, which could indicate potential risks or emergency situations. The research explored different data-driven techniques such as machine learning models that process radar data to recognize patterns indicative of abnormal behaviour, such as unexpected turns, sudden stops, or straying from common traffic lanes. Tang et al. [22] reviewed how radar and other sensor data are used to predict ship behaviour in mixed waterborne traffic scenarios, providing insights into the interaction between manned and unmanned vessels. This study reviews how radar technology is used to understand and predict interactions between different types of vessels, particularly in areas where commercial, recreational, and unmanned vessels operate together. It could discuss the challenges of using radar to accurately capture dynamic scenarios and how these data are integrated with other sensor data to enhance predictive accuracy and reliability in mixed traffic environments.

Traditional AIS technology has several significant limitations [23]. Firstly, the AIS data may be inaccurate or outdated if multiple vessels fail to update their AIS devices or do not turn on their AIS equipment, leading to potential gaps in data. Secondly, there is an inherent delay in data transmission from AIS devices, ranging from 3 s to several minutes. This latency hinders the effective handling of hazardous and emergency situations where timely information is crucial. Lastly, integrating AIS data with radar data is challenging due

to the different sources of information, complicating the correlation and overall situational awareness required for navigational safety.

Visual tracking technology leverages cameras and other image capturing devices to monitor ships, providing an effective means for predicting maritime behaviours [24]. Initially, real-time images of vessels are captured via cameras installed along coastlines, on nearby vessels, or on aircraft. These images undergo preprocessing, such as noise reduction and contrast enhancement, to improve image quality [25]. Advanced algorithms, such as Convolutional Neural Networks, are then applied for object detection and recognition, differentiating ships from other maritime objects [26]. Additionally, key features of ships, including length, width, colour, and shape, are extracted and techniques like optical flow are utilized to estimate the velocity and direction of movement. Continuous tracking algorithms, such as multi-object tracking, are employed to monitor the positions and movements of multiple vessels in real time [27]. Ultimately, by analyzing the motion patterns of ships and employing time series analysis or machine learning models to predict future positions and behaviours, visual tracking technology not only enhances maritime safety but also boosts the efficiency of decision support systems, especially in complex or busy port areas. This integrative approach, which combines visual data with other sources of information such as AIS and radar, offers comprehensive decision support for navigators and traffic management systems.

Instance segmentation, leveraging computer vision and deep learning techniques, provides a nuanced approach for predicting ship behaviour by segmenting and analyzing individual objects within images [28]. This method excels in delivering high-resolution detection and precise outlining of ships, which enables a detailed analysis of ship features and movements crucial for accurate behaviour prediction. Unlike AIS and radar, which primarily focus on positional data such as speed and heading, instance segmentation dynamically recognizes and tracks changes in ships' shape and size, offering insights into specific manoeuvres or behaviours. Furthermore, it demonstrates robust performance under a variety of visual conditions, effectively handling environmental factors such as fog or poor lighting by utilizing extensive image data. Additionally, instance segmentation can be integrated with AIS and radar data, providing a comprehensive view of maritime traffic that merges precise visual details with the reliability and standardization of traditional maritime navigation systems. This integration enriches the contextual understanding of maritime scenarios, enhancing both the precision and applicability of ship behaviour predictions in diverse maritime environments. Figure 1 and Table 1 illustrate the characteristics of situational awareness for surrounding vessels using AIS images, radar detection images, and ship wake images.

Table 1. Comparison of AIS, radar, visual tracking, and instance segmentation in predicting ship behaviour.

Feature	AIS	Radar	Visual Tracking	Instance Segmentation
Coverage Range	Global	Up to 100 nm	Line of Sight (approx. 1–10 km)	Line of Sight (approx. 1–10 km)
Measurement Accuracy	±10 m	±1 m	±5 m	±1 m
Instance Differentiation	×	✓	✓	✓
Visual Information	×	×	✓	✓
Real-Time Processing	✓	✓	✓	✓
Integration with Other Data	✓	✓	✓	✓
Environmental Robustness	Good	Excellent	Good	Good
Cost	Low	High	Medium	Medium-High

Note: ✓ indicates that the feature is supported or included, whereas × indicates that the feature is not supported or not included.

Traditional AIS and radar systems exhibit significant deficiencies in environmental perception. AIS data can be inaccurate or outdated due to delays and incomplete updates, while radar data often remains uncorrelated with AIS information, complicating situational awareness. Existing computer vision methods predominantly focus on classical tasks such as object detection, classification, and instance segmentation. However, there has been a lack of research specifically addressing the detection and application of ship wakes based on seamanship principles. This study aims to fill this gap by exploring new methodologies for ship wake detection and its practical applications in maritime navigation.

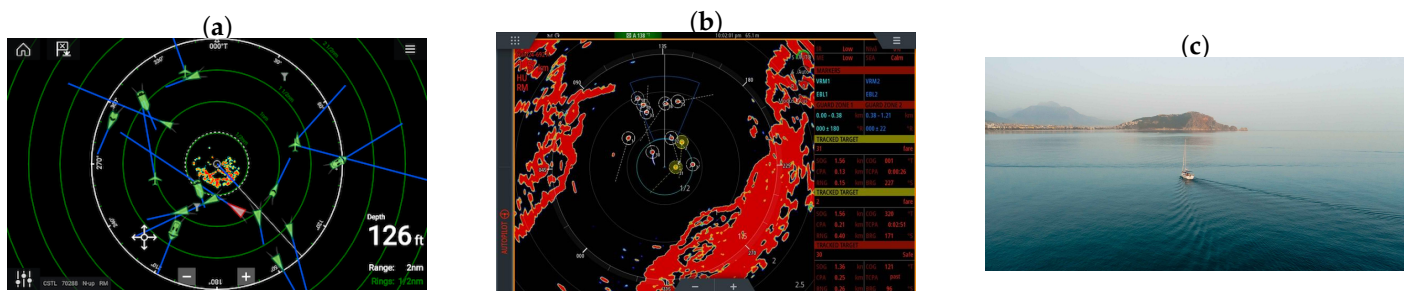


Figure 1. (a) AIS perception: positions, headings, and speeds of multiple vessels in a given area. (b) Radar perception: detected objects and possible danger zones or obstacles. (c) Vision perception: a ship wake image that supports inference of dynamic vessel characteristics, such as changes in heading and speed.

3. Methodology

This section presents the development of an enhanced algorithm based on the YOLOv9 visual object detection system. Each proposed improvement is evaluated in terms of its impact on the overall performance of the object detection system, setting the stage for empirical validation in subsequent sections.

3.1. Ship-Wake Instance Segmentation and Visual Cue Extraction

Instance segmentation is a fundamental computer vision task that combines object detection with pixel-level semantic segmentation. It localizes individual object instances and assigns each instance a category label. The YOLO series has been widely used for real-time detection and segmentation because of its balance between accuracy and inference efficiency [29]. In this study, YOLOv9-Seg is used as the baseline model for extracting two visual classes from inland-waterway images: the ship hull and the ship wake.

The purpose of the proposed perception model is to provide reliable hull and wake masks from visual images. These masks can be used to derive geometric cues related to the spatial relationship between a vessel and its wake. Such cues may support downstream navigational-state interpretation, but they are not treated in this paper as a validated manoeuvre-prediction module. The present study focuses on the perception stage, namely ship and wake detection and instance segmentation.

For a segmented image at frame t , let M_t^s and M_t^w denote the binary masks of the ship and wake, respectively. The centroid of each mask is computed as

$$\mathbf{c}_t^k = \frac{1}{|M_t^k|} \sum_{\mathbf{p} \in M_t^k} \mathbf{p}, \quad k \in \{s, w\}, \quad (1)$$

where $\mathbf{p} = (x, y)^\top$ denotes a pixel coordinate and $|M_t^k|$ is the number of pixels in the corresponding mask.

$$\Sigma_t^k = \frac{1}{|M_t^k|} \sum_{\mathbf{p} \in M_t^k} (\mathbf{p} - \mathbf{c}_t^k)(\mathbf{p} - \mathbf{c}_t^k)^\top, \quad (2)$$

and the principal eigenvector of Σ_t^k is used to estimate the main axis direction θ_t^k . The angular deviation between the hull and wake is then defined as

$$\Delta\theta_t = |\text{atan2}(\sin(\theta_t^s - \theta_t^w), \cos(\theta_t^s - \theta_t^w))|. \quad (3)$$

The centroid offset between the hull and wake can be written as

$$d_t = \|\mathbf{c}_t^s - \mathbf{c}_t^w\|_2. \quad (4)$$

These quantities describe the image-level hull–wake spatial relationship. In the revised manuscript, they are used only as visual cues extracted from the segmentation output, rather than as direct labels for manoeuvre prediction.

Figure 2 presents the revised workflow of the proposed ship–wake perception and visual cue extraction process.

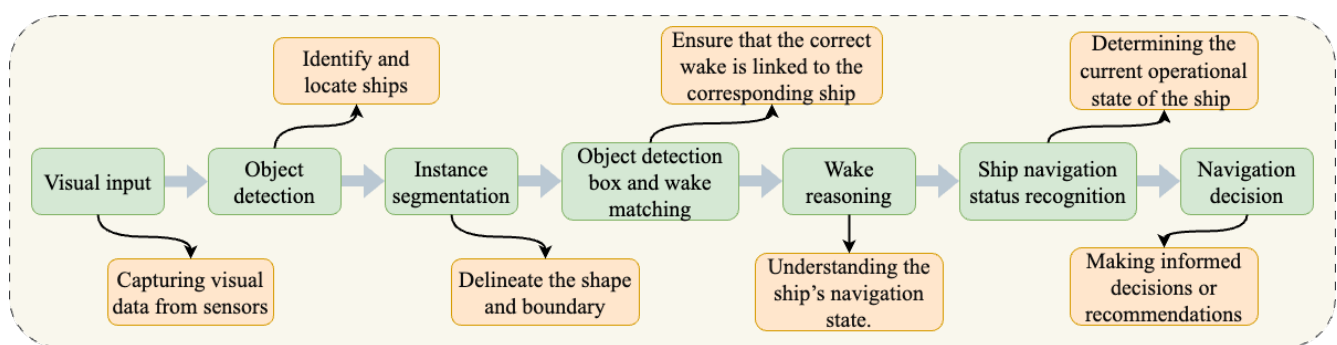


Figure 2. Workflow of the proposed ship–wake perception and visual cue extraction process. Arrows indicate the processing flow from visual input to ship/wake segmentation, hull–wake matching, wake reasoning, and possible downstream navigational-state interpretation. Green boxes denote main processing modules, and yellow boxes denote explanatory functions or intermediate interpretation cues.

It should be noted that the current image-level dataset does not provide synchronized ground-truth manoeuvre labels for states such as acceleration, course keeping. Therefore, this paper does not quantitatively evaluate a temporal intention-recognition or future manoeuvre-prediction model. The speed-threshold rule in the previous version has been removed to avoid overinterpreting the segmentation output. A complete navigation-state recognition framework would require synchronized visual frames, AIS/GNSS/INS-derived speed and heading, temporal filtering or sequence modelling, and labelled manoeuvre states for quantitative validation.

3.2. Dataset and Baseline

The instance segmentation dataset used in this study was collected using cameras in the Wuhu Channel of the Yangtze River. The dataset comprises 2100 images derived from real ship navigation scenarios. It includes two categories: “Ship” and “Shiptrack”, and “Ship” refers to the vessel from the bridge perspective, whereas “Shiptrack” denotes the wake generated on the water surface during vessel navigation. Recent work has also emphasized the importance of shipborne first-view visual datasets for autonomous navigation. For example, MASS-LSVD provides a large-scale first-view maritime vessel detection dataset collected from onboard cameras, supporting the development and evaluation of visual perception algorithms for Maritime Autonomous Surface Ships [30].

The dataset was divided into training, validation, and test subsets at the image level, with no image shared across different subsets. The final split contains 1470 training images, 420 validation images, and 210 test images. All images were annotated for two instance

categories, Ship and Shiptrack. Ship masks delineate the visible hull region, while Shiptrack masks delineate the visible wake envelope on the water surface. For diffuse wake boundaries, the annotation followed the outermost visually continuous foam, ripple, or brightness-transition region associated with the corresponding vessel, while ambiguous water textures not attributable to a specific vessel were kept as background. Overlapping wakes were annotated as separate instances only when their visible spatial extents could be associated with different hulls; otherwise, the ambiguous region was excluded from the instance mask. The annotation results were checked through a quality-control procedure in which incomplete masks, wrong class assignments, and inconsistent wake boundaries were manually reviewed before model training.

Instance segmentation based on computer vision offers several advantages over AIS, radar, and visual tracking methods in predicting ship behaviour. Popular instance segmentation algorithms include YOLO series, transformer-based models [31], Mask R-CNN [32], and YOLACT [33]. YOLOv5 is known for its fast inference speed, good accuracy, and ease of deployment, though it may struggle with small or overlapping objects. YOLOv6 [34] improves on accuracy and efficiency, but it requires more computational resources. YOLOv7 [35] offers enhanced speed and accuracy, particularly for small objects, but at the cost of increased model size and computational needs. YOLOv8 [36] provides state-of-the-art performance, balancing speed, and accuracy, albeit with higher complexity and potential costs. YOLOv9 [37], the latest in the YOLO series, boasts cutting-edge accuracy and efficiency with the latest neural network advancements, though it demands significant computational power. Transformer-based models excel at capturing long-range dependencies and offer high flexibility but are computationally intensive and slower. Mask R-CNN is highly accurate and well-established but slower and more complex to train and deploy. YOLACT combines the speed of YOLO with instance segmentation capabilities but may be less accurate than specialized models. Given these considerations, YOLOv9-seg is formally defined as the Baseline model in this study for detecting ship wakes. While newer iterations exist, YOLOv9 offers an optimal balance of programmable gradient information and segmentation accuracy, serving as a stable foundation for the proposed architectural enhancements.

The architecture of YOLOv9-Seg is structured into a backbone, neck, and head to facilitate efficient instance segmentation. The backbone integrates Conv modules, ADown for pooling-based downsampling, and the RepNCSPeLAN4 module which minimizes computational redundancy while enhancing feature extraction through layer aggregation. For feature fusion, the neck employs a dual-path structure combining a Feature Pyramid Network (FPN) and a Path Aggregation Network (PAN) to synthesize semantic richness from top-down pathways with detailed localization information from bottom-up propagation. Additionally, the SPPELAN module synergizes Spatial Pyramid Pooling Fast (SPPF) with the Efficient Local Aggregation Network (ELAN) to capture multi-scale spatial data and improve model robustness. Finally, the head network utilizes three distinct segment detectors operating on multi-scale feature matrices to generate precise object localization and pixel-level segmentation outputs.

3.3. Task-Oriented Use of iAFF and SAConv

The iAFF and SAConv modules are adopted from previous computer vision studies and are not claimed as new operators in this work. Their role in Shipwake-YOLO is task-oriented: iAFF is used to regulate cross-scale feature fusion under water-surface clutter, while SAConv is used to expand the receptive field of selected convolutional blocks for elongated and low-contrast wake regions. YOLOv9's instance segmentation capability provides a strong baseline for ship wake detection owing to its advanced architecture and

segmentation accuracy. To further enhance performance, this study integrates iterative Attentional Feature Fusion (iAFF) [38] and Switchable Atrous Convolution (SAConv) [39] into the feature extraction and fusion process. iAFF emphasizes informative features and suppresses redundant background responses, whereas SAConv captures contextual information at multiple receptive-field scales. Their combination improves the localization and segmentation of ships and wakes under complex and variable maritime backgrounds, thereby reducing false positives and missed detections and improving the quality of downstream visual navigational-state cue interpretation.

3.3.1. The Model of iAFF

When incorporating point features, we observe that they are easily affected by complex backgrounds or adjacent targets. Specifically, the subset of point features corresponding to a single target is not only dispersed across the target itself but also scattered within the surrounding background clutter or adjacent target regions. This phenomenon adversely impacts the overall efficiency of object detection. To address this issue, as illustrated in Figure 3, an iAFF mechanism is introduced within the Feature Pyramid Network. The proposed iAFF mechanism mitigates the adverse effects of non-object elements by attenuating redundant features through incentive weight vectors. Shallow feature maps contain rich texture details but also meaningless speckle noise, while deep feature maps have a larger receptive field but suffer from significant resolution loss, making accurate predictions at a fine-grained level challenging. Balancing the representation of shallow texture details and deep semantic features, along with robust suppression of interference, is key to effectively extracting point features. Directly combining shallow texture details with deep semantic features introduces substantial redundancy due to their differences. To harmonize these features, this paper incorporates SAConv within the iAFF, which enhances the model's multi-scale representation capability.

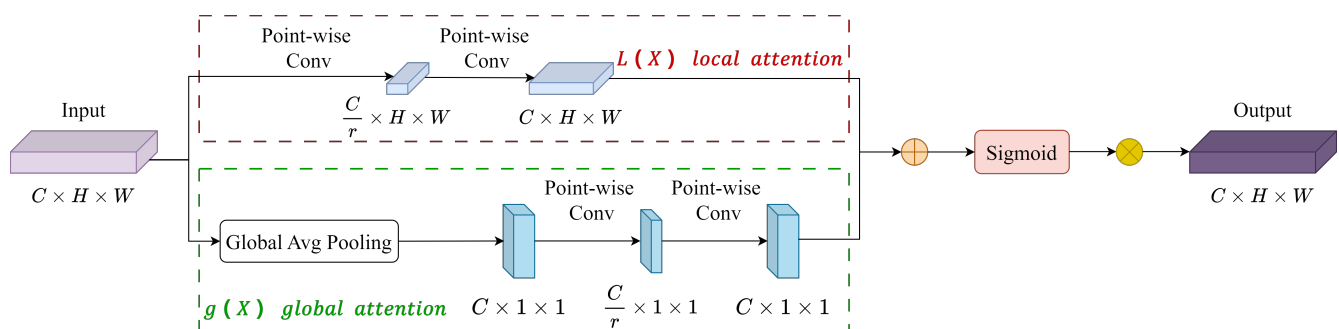


Figure 3. Structure of the Multi-Scale Channel Attention Module. The upper branch denotes local attention, and the lower branch denotes global attention. Different colors distinguish convolution, pooling, attention-weight generation, and output features; circular symbols indicate feature addition and element-wise multiplication.

Dai et al. [38] proposed a Multi-Scale Channel Attention Module (MS-CAM) to bridge the gap between features at different scales, achieving attentive feature fusion. The iAFF mechanism aided in balancing the representation of shallow texture details and deep semantic features while suppressing interference from non-object elements. Here is a detailed explanation of the iAFF mechanism with the relevant formulas [38].

The variation in object scales was a fundamental research topic in computer vision. MSCAM combined global context information from global features and local context information from local features through point-wise convolutional fusion of the input feature maps. This enhanced the adaptiveness of multi-scale features, allowing the network

to focus on significant information in image objects while suppressing background noise more effectively, as shown in Figure 3.

The global channel feature attention is denoted as $g(X)$, with H and W representing the height and width of the feature map, respectively. Unlike $L(X)$, $g^*(X)$ denotes global average pooling; W_1 and W_2 denote the convolution operations of the descending and ascending layers, respectively; B denotes the batch normalization (BN) layer; and δ denotes the rectified linear unit (*ReLU*) activation function:

$$g(X) = B(W_2\delta(B(W_1(g^*(X))))) \quad (5)$$

Local channel attention applies two point-wise convolution (PWConv) operations directly on the input feature map to obtain the final local attention feature map. The local channel attention formula is given by

$$L(X) = B(\text{PWConv}_2(\delta(B(\text{PWConv}_1(X))))) \quad (6)$$

where X is the input feature map, PWConv_1 represents a 1×1 convolution, B denotes BN, and δ denotes *ReLU*. PWConv_2 upscales the number of channels back to the original number, with r denoting the scaling factor of the channel dimensions.

After applying attention to the input X , the computed weights are used to obtain the output X' as follows:

$$X' = X \otimes M(X) = X \otimes \sigma(L(X) \oplus g(X)) \quad (7)$$

where $M(X)$ represents the attention weights assigned by MSCAM, σ denotes the sigmoid function, $\oplus$ denotes the broadcasting addition, and $\otimes$ represents element-wise multiplication.

Given two feature maps $X, Y \in \mathbb{R}^{C \times H \times W}$, it is assumed by default that Y is the feature map with a larger receptive field and C represents the channel. This can be exemplified in three scenarios: (1) in the same-layer scenario, X is the output of a 3×3 kernel and Y is the output of a 5×5 kernel in InceptionNet; (2) in the short skip connection scenario, X is the identity mapping and Y is the learned residual in a ResNet block; (3) in the long skip connection scenario, X is the low-level feature map and Y is the high-level semantic feature map in a feature pyramid.

Based on the Multi-Scale Channel Attention Module (MSCAM), the Attentional Feature Fusion (AFF) can be expressed as

$$Z = M(X \parallel Y) \otimes X + (1 - M(X \parallel Y)) \otimes Y \quad (8)$$

where ($Z \in \mathbb{R}^{C \times H \times W}$) is the fused feature, and $\parallel$ denotes the initial feature integration. For simplicity, element-wise summation is chosen as the initial integration. This process is illustrated in Figure 4, where the dashed line represents $1 - M(X \parallel Y)$.

It should be noted that the fusion weights $M(X \parallel Y)$ consist of real numbers between 0 and 1. This allows the network to perform a soft selection or weighted averaging between X and Y , enabling a more flexible and adaptive feature fusion process.

Unlike partially context-aware methods such as the Pyramid Attention Network for semantic segmentation, fully context-aware approaches face a significant challenge: how to initially integrate input features. The quality of this initial integration can greatly influence the final fusion weights of the attention module. Given that this remains a feature fusion problem, a straightforward solution is to employ another attention module to fuse the

input features. This two-stage method is referred to as iAFF, as depicted in Figure 5. Consequently, the initial integration $X \parallel Y$ in Equation (9) can be reformulated as

$$X \parallel Y = M(X + Y) \otimes X + (1 - M(X + Y)) \otimes Y \quad (9)$$

This iterative approach enables a more refined and effective feature fusion process, leveraging attention mechanisms at multiple stages to enhance the overall integration quality.

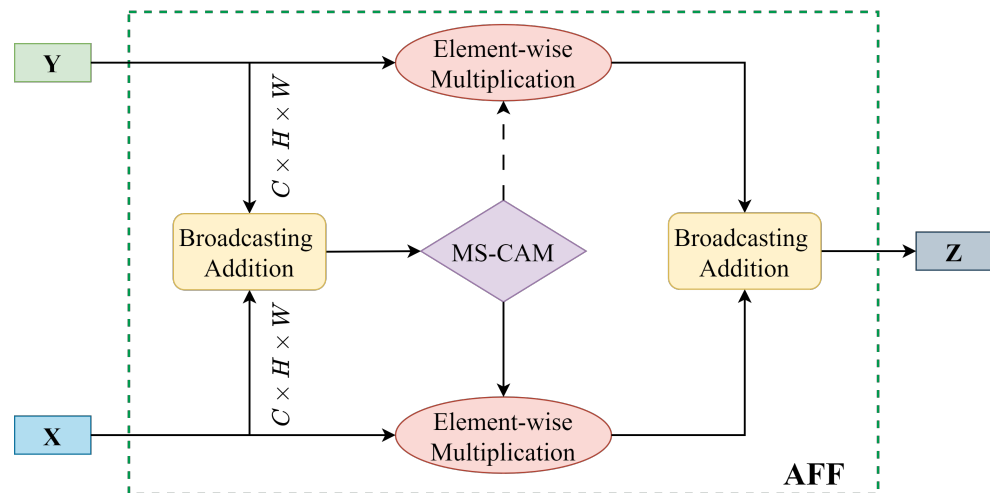


Figure 4. Structure of the Attentional Feature Fusion (AFF) module. The two inputs, X and Y , are adaptively fused through the MS-CAM attention module. Different colors distinguish input features, attention weights, broadcasting addition, element-wise multiplication, and the fused output Z ; the dashed path represents $1 - M(X \parallel Y)$.

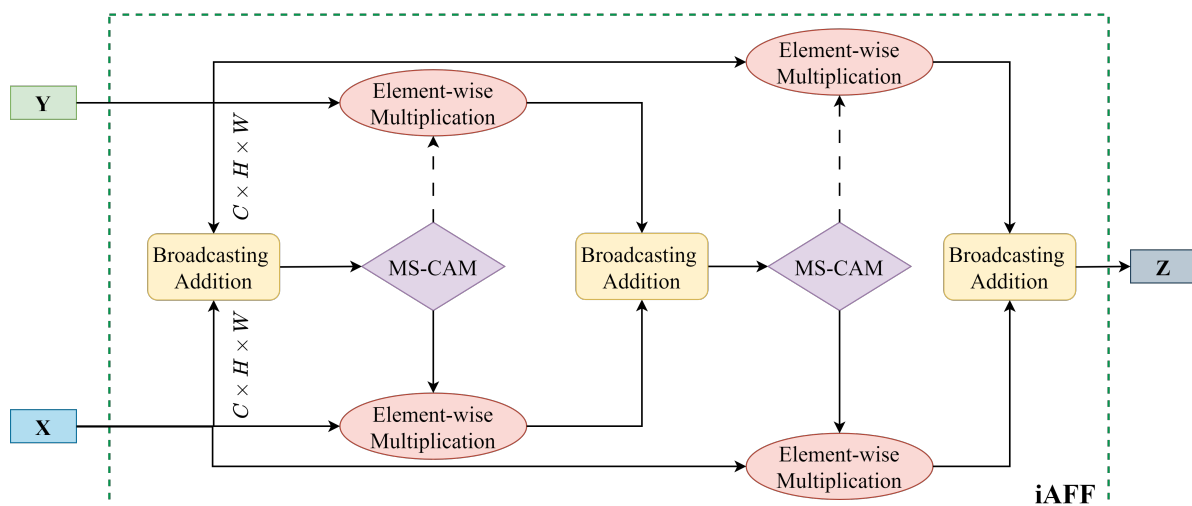


Figure 5. Structure of the iterative Attentional Feature Fusion (iAFF) module. The module performs two-stage attention-based feature fusion to refine the integration of X and Y . Different colors distinguish input features, MS-CAM attention modules, broadcasting additions, element-wise multiplications, and the fused output Z .

The iAFF mechanism enhances the representation of significant features while suppressing redundant ones through attention maps and incentive weights. By fusing shallow and deep features using weighted contributions, the iAFF mechanism effectively balances texture details and semantic information, leading to more accurate and robust feature extraction for tasks such as ship wake detection.

3.3.2. The Model of SAConv

Considering the need to extract target features while reducing model redundancy and model inference speed, this study proposes improvements to the model's convolutional modules. The SAConv convolutional module offers higher accuracy and faster speed. SAConv enhances the feature representation by capturing long-range dependencies and contextual information within a feature map. Figure 6 is a detailed explanation of the SAConv mechanism with the relevant formulas [39].

Atrous convolution, also known as dilated convolution, allows for an increased effective receptive field size by introducing an atrous rate. For an atrous rate of l in atrous convolution, $l - 1$ zeros are inserted between consecutive filter values. Consequently, a $k \times k$ filter kernel expands to a size of $k + (k - 1)(l - 1)$ without altering the number of network parameters. Figure 6 illustrates an example of a 3×3 atrous convolution with an atrous rate of 1 (shown in red).

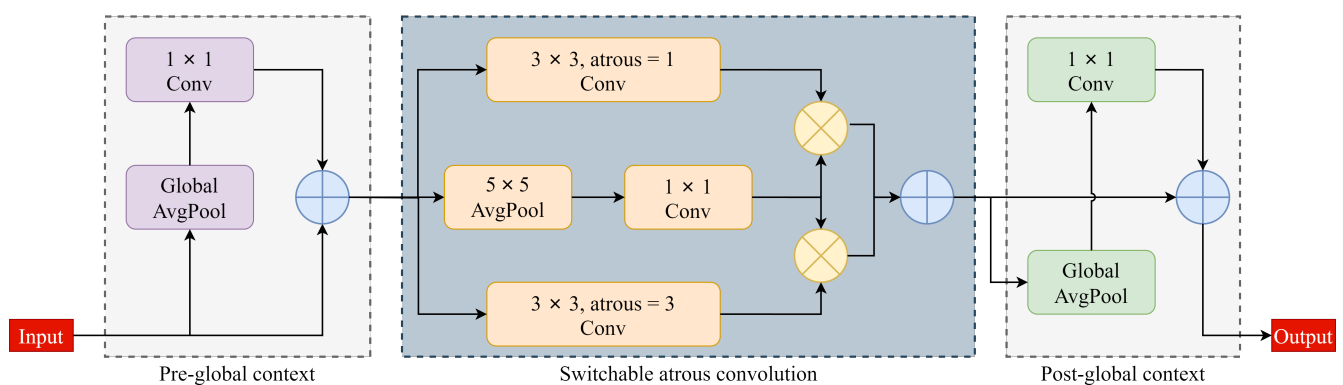


Figure 6. Model of Switchable Atrous Convolution (SAConv). The left and right parts represent the pre-global and post-global context branches, while the central part represents the switchable atrous convolution branch. Different colors distinguish convolution, average pooling, switching, feature addition, and weighted fusion operations.

In this subsection, the proposed SAC is detailed. Figure 6 illustrates the overall architecture of SAC, which consists of three main components: two global context modules positioned before and after the SAC component. This section focuses on the central SAC component, while the global context modules are explained subsequently.

The convolutional operation with weight $\mathbf{w}$ and atrous rater, which takes $\mathbf{x}$ as input and outputs $\mathbf{y}$, is represented as $\mathbf{y} = \text{Conv}(\mathbf{x}, \mathbf{w}, r)$. The conversion of a convolutional layer to SAC can be formulated as follows:

$$\text{Conv}(\mathbf{x}, \mathbf{w}, 1) \xrightarrow{\text{Convert to SAC}} \mathbf{S}(\mathbf{x}) \cdot \text{Conv}(\mathbf{x}, \mathbf{w}, 1) + (1 - \mathbf{S}(\mathbf{x})) \cdot \text{Conv}(\mathbf{x}, \mathbf{w} + \Delta\mathbf{w}, r) \quad (10)$$

Here, r is a hyper-parameter of SAC, $\Delta\mathbf{w}$ is a trainable weight, and the switch function $\mathbf{S}(\cdot)$ is implemented as an average pooling layer with a 5×5 kernel followed by a 1×1 convolutional layer (see Figure 6). The switch function depends on both the input and location, enabling the backbone model to adapt to various scales as needed.

A locking mechanism is proposed by setting one weight as $\mathbf{w}$ and the other as $\mathbf{w} + \Delta\mathbf{w}$ for several reasons. Object detectors typically use pretrained checkpoints to initialize weights. For an SAC layer converted from a standard convolutional layer, the weight for the larger atrous rate is missing. Since objects at different scales can be roughly detected by the same weight with different atrous rates, it is natural to initialize the missing weights with those from the pretrained model. The implementation uses $\mathbf{w} + \Delta\mathbf{w}$ for the missing weight, where $\mathbf{w}$ is from the pretrained checkpoint and $\Delta\mathbf{w}$ is initialized to 0. Setting $\Delta\mathbf{w} = 0$ results in a 0.1% decrease in AP, whereas using $\Delta\mathbf{w}$ without the locking mechanism significantly degrades AP.

The SAConv significantly enhances feature representation by capturing long-range dependencies and contextual information. By transforming the input feature map into query, key, and value matrices, computing attention scores, and combining the values based on these scores, SAConv provides a robust method to improve feature extraction for tasks such as instance segmentation.

3.4. Layer-Level Adaptation of YOLOv9-Seg for Wake Segmentation

The architectural adaptation in Shipwake-YOLO is designed around the spatial properties of ship wakes rather than a generic stacking of existing modules. In inland-waterway images, wake regions are usually elongated, weakly textured, and partially mixed with water-surface clutter, so their boundary cues can be weakened during downsampling or diluted during coarse feature aggregation. To address this issue, selected ADown blocks in the baseline network are replaced by iSACADown, allowing weak wake textures to be better preserved during resolution reduction. The iAFF module is inserted into the cross-scale fusion path of the neck, where shallow ripple-like textures and deeper hull/wake semantics need adaptive weighting before feature concatenation. At the high-level aggregation stage, the original SPPELAN block is adapted into iSACSPPELAN, where SAConv-enhanced convolution enlarges the effective receptive field and helps associate fragmented wake regions with the corresponding vessel. The segmentation head is kept unchanged, and MPD-IoU is adopted only in the bounding-box regression branch. This design confines the modification to feature extraction, downsampling, feature fusion, and localization optimization, thereby positioning Shipwake-YOLO as a wake-oriented adaptation of YOLOv9-Seg rather than a new generic instance segmentation framework.

In the YOLOv9-Seg framework, the utilization of a Feature Pyramid Network (FPN) is a strategic choice to facilitate multi-scale feature fusion, effectively enabling the model to detect objects at various scales with high precision. This approach is critical in maintaining the robustness of the model across different object sizes and resolutions, a key factor in applications such as real-time object detection and segmentation.

To enhance this architecture, the paper introduces significant modifications by incorporating the SAConv module within the SPPELAN and ADown frameworks. SAConv, known for its ability to capture long-range dependencies within feature maps, enriches the model's ability to discern more nuanced patterns and contextual information. This is particularly beneficial in complex visual scenes where standard convolution might miss subtle but crucial details.

Integrating SAConv into SPPELAN and ADown aims to streamline the model further by reducing the overall parameter count and simplifying the model's complexity without compromising performance. SPPELAN, which typically focuses on optimizing spatial hierarchies for more efficient feature extraction, and ADown, an adaptive down-sampling method, both benefit from the inclusion of SAConv by gaining a more dynamic capability to focus on relevant features while reducing computational load.

3.4.1. The Model of iSACSPPELAN

SPPF (Spatial Pyramid Pooling Fast) is an innovative pooling technique introduced in YOLOv5. SPP (Spatial Pyramid Pooling) was originally introduced in YOLOv3 as a pooling strategy capable of capturing spatial information at different scales, thereby enhancing model robustness. However, the computational complexity of SPP adversely affected model inference speed. To address this issue, YOLOv5 proposed SPPF, which implements SPP with optimized algorithms for fast computation. The core idea of SPPF is to maintain the functionality of SPP while improving model inference speed by reducing computational redundancy. Specifically, SPPF adopts strategies such as point-wise convolution and

grouped convolution to merge pooling operations performed at multiple scales into a single scale, thus reducing computational complexity. Additionally, SPPF employs point-wise max pooling to replace traditional Spatial Pyramid Pooling, further reducing computational overhead. By introducing SPPF, YOLOv5 significantly improves model inference speed while maintaining high accuracy. This enhancement makes YOLOv5 more competitive in practical applications, especially in scenarios requiring real-time processing.

As advancements in object detection technology persist, researchers continually explore avenues to further enhance model performance. Within YOLOv9, there is potential for the introduction of an innovative technique termed SPPELAN. SPPELAN amalgamates SPP with ELAN with the aim of augmenting object detection efficacy through the fusion of their respective strengths. ELAN, characterized by its lightweight network architecture, adeptly enhances the model's feature extraction capability through localized aggregation and global integration methodologies. By integrating SPP with ELAN, SPPELAN effectively harnesses the Spatial Pyramid Pooling prowess of SPP alongside the efficient feature aggregation capabilities of ELAN, consequently further enhancing model performance. The introduction of SPPELAN is anticipated to enable YOLOv9 to maintain high accuracy while further bolstering model robustness and generalization capabilities. Furthermore, owing to the lightweight nature of ELAN, SPPELAN also aids in reducing model computational complexity, thereby enhancing inference speed. This baseline SPPELAN design motivates the iSACSPPELAN adaptation described below.

3.4.2. iSACSPPELAN: SACConv-Enhanced SPPELAN Aggregation

In this manuscript, SPPELAN refers only to Spatial Pyramid Pooling Enhanced with ELAN, consistent with the YOLOv9 terminology, and it is not redefined as a self-projection or pair-wise-embedding attention network. The SPPELAN block implemented in the baseline code follows the YOLO-family design principle of combining Spatial Pyramid Pooling with ELAN-style feature aggregation, so that high-level feature maps can retain multi-scale spatial context without introducing a separate transformer-like attention formulation. In Shipwake-YOLO, the corresponding adaptation is termed iSACSPPELAN. This module preserves the input–output interface and aggregation logic of the original SPPELAN block, while replacing selected standard convolutional paths with SACConv-enhanced convolutional paths. The purpose of this replacement is to enlarge the effective receptive field at the high-level aggregation stage, where fragmented wake regions need to be associated with the corresponding hull and where weak wake textures have already passed through several downsampling operations. Therefore, iSACSPPELAN should be interpreted as a wake-oriented modification of the YOLOv9 SPPELAN block, rather than a newly defined attention network. Figure 7 illustrates the implemented iSACSPPELAN adaptation.

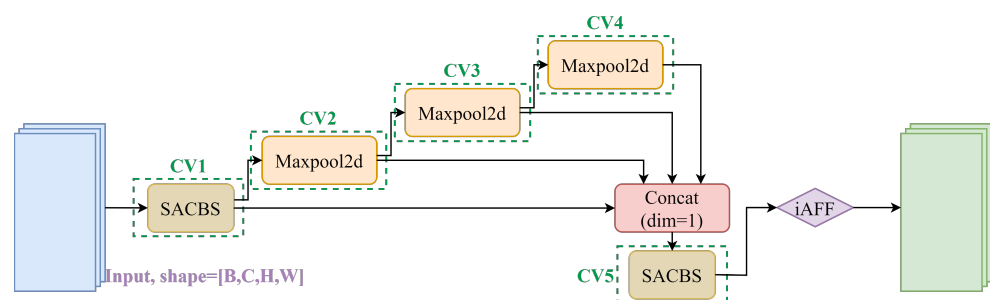


Figure 7. Structure of the proposed iSACSPPELAN module. The module preserves the SPPELAN aggregation pathway and introduces SACConv-enhanced convolutional paths for wake-oriented high-level feature aggregation. Different colors distinguish input/output features, convolutional blocks, pooling branches, iAFF-based fusion, concatenation, and output projection.

3.4.3. The Model of iSACADown

The ADown module performs downsampling on input features and splits the feature channels into two parts, followed by different convolution operations on each part, and finally concatenates the results for output. Initially, the module calculates half of the input channel count and creates two convolutional layers, *cv1* and *cv2*. *cv1* applies a 3×3 convolutional kernel with a stride of 2 and padding of 1, while *cv2* uses a 1×1 convolutional kernel with a stride of 1 and no padding. Subsequently, the input undergoes average pooling using PyTorch's *avg_pool2d* function, reducing the input size by half. The pooled features are then split into two parts along the channel dimension using the chunk method. One part undergoes a convolutional operation using the convolution kernel *cv1*, while the other part first undergoes max pooling and then a convolutional operation using the kernel *cv2*. Finally, the convolution results of the two parts are concatenated and returned as output.

Attention-based Downsampling enhances feature representation by adaptively downsampling features while preserving critical information. The ADown mechanism involves several key steps: feature extraction, attention map computation, weighted downsampling, and feature aggregation.

Given an input feature map $F \in \mathbb{R}^{H \times W \times C}$, the first step is to extract features using convolutional layers:

$$F_{\text{conv}} = \text{Conv}(F) \quad (11)$$

where $F_{\text{conv}} \in \mathbb{R}^{H \times W \times C'}$ represents the feature map after convolution and C' denotes the convolutional operation.

Next, we compute the attention map A to identify the importance of different spatial locations within the feature map. This is done by applying a series of convolutional layers followed by a sigmoid activation function:

$$A = \sigma(\text{Conv}_A(F_{\text{conv}})) \quad (12)$$

where $A \in \mathbb{R}^{H \times W}$ is the attention map, σ denotes the sigmoid function, and Conv_A represents the convolutional layers used to compute the attention map.

The weighted downsampling is performed by applying the attention map to the feature map, followed by a pooling operation. The attention map acts as weights to emphasize important regions:

$$F_{\text{weighted}} = A \odot F_{\text{conv}} \quad (13)$$

where $F_{\text{weighted}} \in \mathbb{R}^{H \times W \times C'}$ is the weighted feature map and $\odot$ denotes element-wise multiplication.

Next, we apply a downsampling operation, such as average pooling, to the weighted feature map:

$$F_{\text{down}} = \text{Pool}(F_{\text{weighted}}) \quad (14)$$

where $F_{\text{down}} \in \mathbb{R}^{H' \times W' \times C'}$ represents the downsampled feature map, and Pool denotes the pooling operation. H' and W' indicate the reduced spatial dimensions (height and width) after pooling, and C' denotes the number of channels after the weighted downsampling operation.

Finally, we aggregate the downsampled features to obtain the output feature map. This can involve concatenating or adding the downsampled features with other feature maps from different levels:

$$F_{\text{out}} = \text{Aggregate}(F_{\text{down}}, F_{\text{other}}) \quad (15)$$

where $F_{\text{out}} \in \mathbb{R}^{H' \times W' \times C''}$ is the output feature map, Aggregate represents the aggregation operation, which can be concatenation or addition, and F_{other} represents other feature maps involved in the aggregation.

The ADown mechanism enhances feature representation by adaptively downsampling features while preserving critical information through attention mechanisms. By computing an attention map, performing weighted downsampling, and aggregating the resulting features, ADown provides a robust method to improve feature extraction for various tasks. The detailed formulation illustrates how ADown integrates these elements to enhance feature representation and improve performance.

However, ADown divides the input feature channels into two parts with a fixed ratio, which may not be suitable for all types of input data and tasks. The fixed splitting ratio may fail to effectively capture the diversity of features between different parts. Furthermore, ADown only concatenates the features after applying different convolution operations to each part. This simplistic fusion strategy might not fully exploit the complementary information between the two parts, potentially limiting the module's performance in capturing complex patterns.

The integration of SACConv and ADown modules may contribute to enhancing the model's performance and efficiency. SACConv facilitates the reduction of model parameters and complexity, while ADown enables downsampling and segmentation of input features into two parts. By combining SACConv and ADown, their respective advantages can be better utilized, leading to improved model performance and efficiency. SACConv exhibits lower parameter count and computational complexity, aiding in reducing model complexity while preserving critical feature information through appropriate design. On the other hand, ADown enhances the model's perception of different scales and features by performing feature downsampling and segmentation. Consequently, the fusion of SACConv and ADown modules holds the potential to enhance model performance, generalization capabilities, and computational efficiency. The structure of iSACADown is shown in Figure 8.

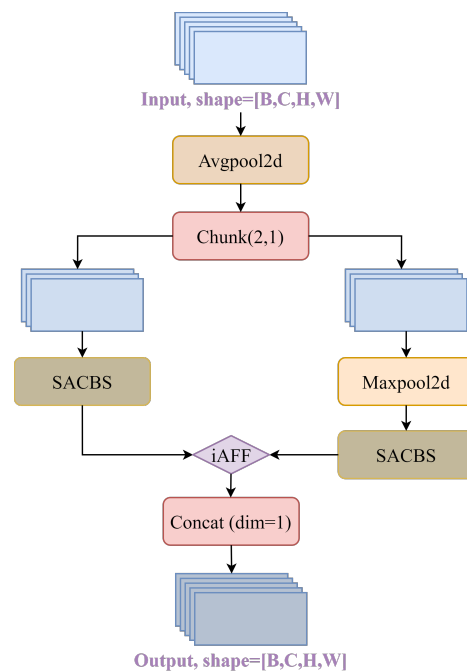


Figure 8. Structure of the proposed iSACADown module. The module combines pooling-based downsampling, SACConv-enhanced convolution, and iAFF-based feature fusion. Different colors distinguish input/output features, average pooling, max pooling, convolutional branches, concatenation, and attention-based fusion operations.

3.5. MPD-IoU Loss for Wake-Oriented Bounding-Box Regression

Bounding-box regression is critical for ship and wake localization because the mask branch still depends on reliable region proposals and box-level spatial constraints. In the present task, wake regions often appear as elongated, low-contrast, and partially deformable structures, so small endpoint shifts can cause a visible mismatch between the predicted region and the annotated wake extent even when the box centre remains close to the ground truth. Conventional IoU-based losses mainly optimize overlap area, while GIoU, DIoU, and CIoU introduce enclosing-region, centre-distance, or aspect-ratio penalties to improve convergence. However, these terms do not directly constrain the two diagonal endpoints of an elongated rectangle, and this limitation can be problematic for wake boxes whose length, orientation-related projection, and weak terminal boundary are difficult to localize. Therefore, this study adopts MPD-IoU as the bounding-box regression loss. MPD-IoU augments the overlap term with the normalized distances between the top-left and bottom-right points of the predicted and ground-truth boxes, thereby introducing coordinate-level penalties that remain sensitive to endpoint displacement. In Shipwake-YOLO, MPD-IoU is not presented as a new loss function; it is used as a task-motivated localization loss for reducing corner drift in elongated ship and wake regions.

The CIoU loss is defined as

$$\text{CIoU} = \text{IoU} - \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} - \alpha v \quad (16)$$

where IoU is the Intersection over Union between the predicted bounding box and the ground truth box. $\rho(\mathbf{b}, \mathbf{b}^{gt})$ is the Euclidean distance between the centre points of the predicted box $\mathbf{b}$ and the ground truth box $\mathbf{b}^{gt}$. c is the diagonal length of the smallest enclosing box that covers both the predicted and ground truth boxes. α is a positive trade-off parameter that adjusts the impact of the aspect ratio term. v is a measure of the consistency of the aspect ratio, defined as

$$v = \frac{4}{\pi^2} \left(\arctan\left(\frac{w^{gt}}{h^{gt}}\right) - \arctan\left(\frac{w}{h}\right) \right)^2 \quad (17)$$

and α is defined as

$$\alpha = \frac{v}{(1 - \text{IoU}) + v} \quad (18)$$

where w and h are the width and height of the predicted bounding box. w^{gt} and h^{gt} are the width and height of the ground truth bounding box.

However, when the predicted box and the ground truth box share the same aspect ratio but have completely different width and height values, CIoU might hinder effective optimization. To address this issue, this research proposes replacing CIoU with MPD-IoU in the network, as depicted in Figure 9.

Compared to existing loss functions, MPD-IoU incorporates the minimum point distance along with other relevant factors like centroid distances and deviations in width and height. This redefines the loss function to better capture the nuances of bounding box alignment and reduce the total degrees of freedom. The formula for LMPD-IoU is shown below:

$$\mathcal{L}_{\text{MPD-IoU}} = 1 - \text{MPD-IoU} \quad (19)$$

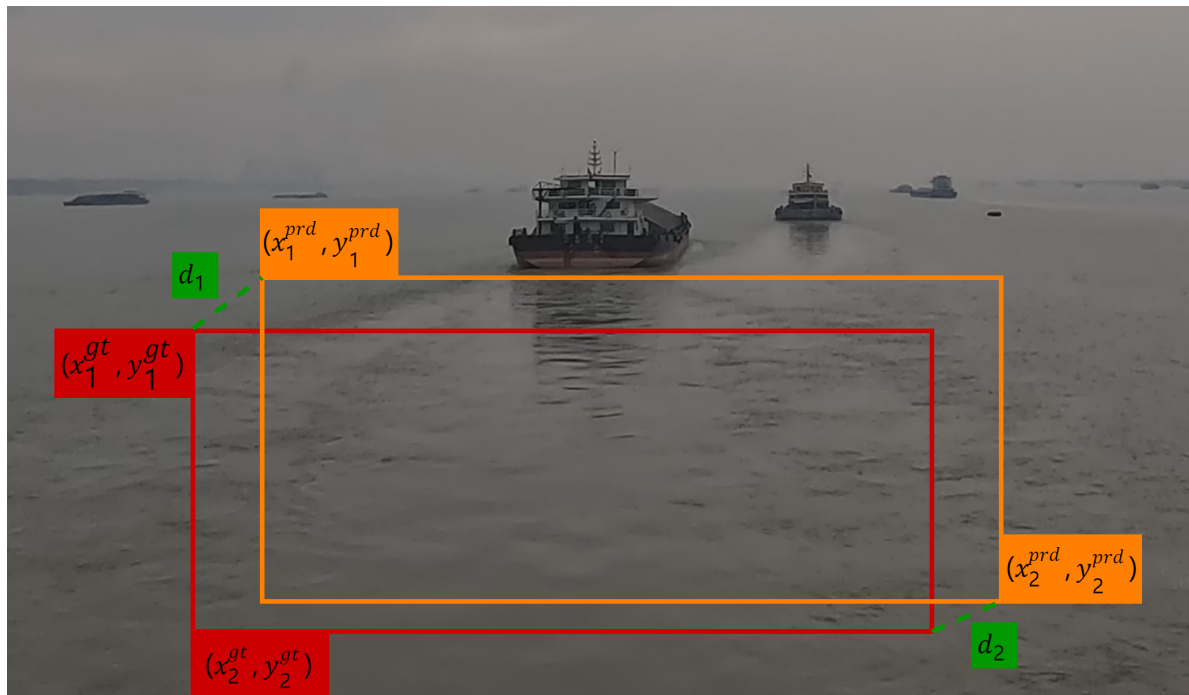


Figure 9. Structure of MPD-IoU for wake-oriented bounding-box regression. The red box denotes the ground-truth bounding box, the orange box denotes the predicted bounding box, and the green annotations indicate the point-distance terms d_1 and d_2 between corresponding diagonal endpoints.

Consequently, the factors in existing loss functions for bounding box regression can be entirely determined by the coordinates of four points. The conversion formulas are presented below. These equations help determine various factors considered in existing loss functions, including non-overlapping area, central point distance, and deviations in width and height.

$$|C| = \left(\max(x_2^{gt}, x_2^{prd}) - \min(x_1^{gt}, x_1^{prd}) \right) \cdot \left(\max(y_2^{gt}, y_2^{prd}) - \min(y_1^{gt}, y_1^{prd}) \right) \quad (20)$$

$$x_c^{gt} = \frac{x_1^{gt} + x_2^{gt}}{2}, y_c^{gt} = \frac{y_1^{gt} + y_2^{gt}}{2}, x_c^{prd} = \frac{x_1^{prd} + x_2^{prd}}{2}, y_c^{prd} = \frac{y_1^{prd} + y_2^{prd}}{2} \quad (21)$$

$$w_{gt} = x_2^{gt} - x_1^{gt}, h_{gt} = y_2^{gt} - y_1^{gt}, w_{prd} = x_2^{prd} - x_1^{prd}, h_{prd} = y_2^{prd} - y_1^{prd} \quad (22)$$

In the above, the area $|C|$ represents the minimum enclosing rectangle's area covering both the ground truth bounding box B_{gt} and the predicted bounding box B_{prd} . The coordinates (x_c^{gt}, y_c^{gt}) and (x_c^{prd}, y_c^{prd}) represent the central points of the ground truth and predicted bounding boxes, respectively. The dimensions w_{gt} and h_{gt} represent the width and height of the ground truth bounding box, while w_{prd} and h_{prd} represent the width and height of the predicted bounding box, and subscript c denotes "centroid".

The formula for MPD-IoU is shown below:

$$\text{MPD-IoU} = \text{IoU} - \frac{d_1^2}{H_l^2 + W_l^2} - \frac{d_2^2}{H_l^2 + W_l^2} \quad (23)$$

$$d_1^2 = (x_1^{\text{pred}} - x_1^{gt})^2 + (y_1^{\text{pred}} - y_1^{gt})^2, \quad (24)$$

$$d_2^2 = (x_2^{\text{pred}} - x_2^{gt})^2 + (y_2^{\text{pred}} - y_2^{gt})^2 \quad (25)$$

where H_I and W_I denote the input image height and width, respectively. They are used only for distance normalization and should not be confused with the width and height of the predicted bounding box.

Equations (23)–(25) indicate that MPD-IoU introduces two explicit point-distance penalties in addition to the overlap term. During optimization, these penalties generate coordinate-level gradients for the two diagonal endpoints of the predicted box, so the loss remains sensitive to box-size and endpoint displacement even when the box centre is not severely biased. This property is relevant to wake detection because wake boxes are usually long and thin, and a small displacement at the far end of the wake can alter the covered water-surface region without producing a large centre shift. Compared with a pure overlap loss, the endpoint-distance terms penalize this type of corner drift more directly. Compared with centre-distance or aspect-ratio penalties, they provide a more localized geometric constraint on the two boundary-defining corners. The adopted loss is still a box-level localization term and cannot by itself solve mask-boundary ambiguity; its role is to improve the spatial alignment of the detection branch, which subsequently supports more stable instance segmentation of elongated ship and wake regions.

3.6. Shipwake-YOLO

Figure 10 summarizes the architecture of Shipwake-YOLO. Relative to YOLOv9-Seg, the input stem and segmentation head are retained, while the feature extraction and fusion pathway is adapted at selected resolution-transition and context-aggregation stages. The original ADown blocks are replaced by iSACADown to reduce the loss of weak wake textures during downsampling. In the neck, iAFF is used to weight shallow texture responses and deeper semantic features before feature concatenation, which is important when wake boundaries are visually close to water-surface ripples. The original SPPELAN module is further adapted into iSACSPPELAN by introducing SAConv-enhanced convolutional paths, allowing the high-level feature maps to encode both compact hull structures and elongated wake regions. MPD-IoU is used only in the bounding-box regression branch, while the mask prediction head remains unchanged. This design keeps the comparison with YOLOv9-Seg controlled and positions the contribution as a wake-oriented architectural adaptation rather than a new generic detector.

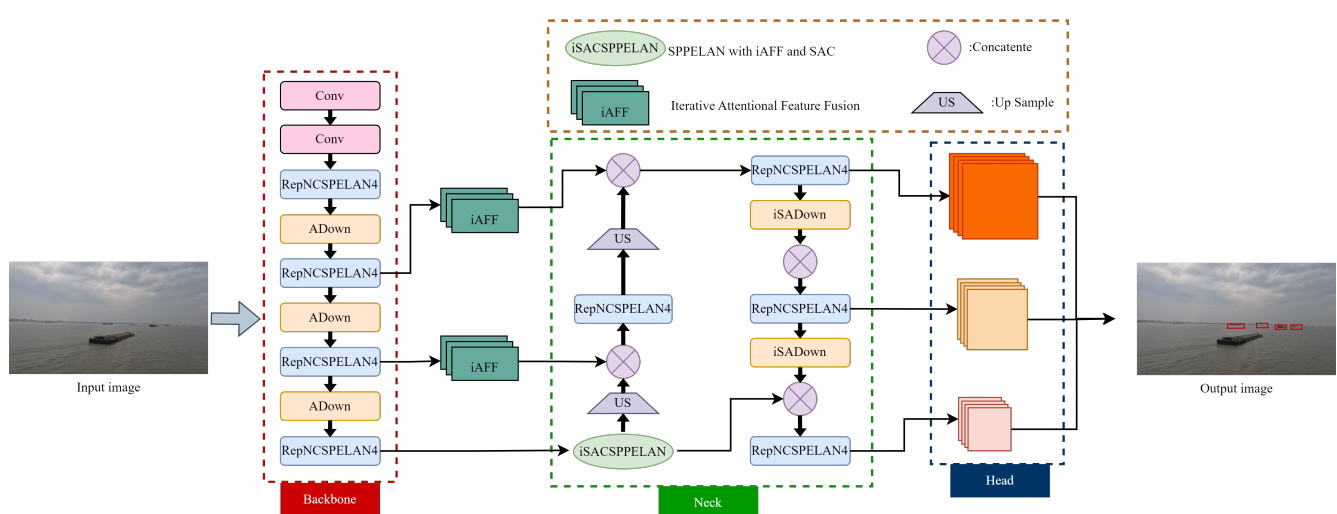


Figure 10. Structure diagram of the proposed Shipwake-YOLO model. The architecture integrates iSACSPPELAN and iSACADown into the YOLOv9-Seg framework. Different colors distinguish the backbone, neck, head, proposed modules, feature maps, and output branches; arrows indicate feature propagation, and special symbols denote upsampling, concatenation, and feature-fusion operations.

4. Experimental Evaluation

4.1. Evaluation Metrics and Experimental Protocol

The performance of the proposed Shipwake-YOLO model was assessed using precision, recall, F1-score, average precision (AP), number of parameters, GFLOPs, and frames per second (FPS). Precision measures the proportion of predicted positive instances that are correct, whereas recall measures the proportion of ground-truth positive instances that are successfully detected. In ship wake detection and instance segmentation, both metrics are important because false detections and missed wake regions may affect navigation-state inference. Parameters indicate model size, GFLOPs measure computational complexity for a single forward pass, and FPS evaluates inference speed. The F1-score is the harmonic mean of precision and recall, while AP is the area under the precision–recall curve. In this study, ship and wake pixels were treated as positive instances, and background pixels were treated as negative instances. The formulas for calculating these metrics are provided in Equations (26)–(30):

$$F_1 = \frac{2 \times P \times R}{P + R} \quad (26)$$

$$AP = \int_0^1 P(r)dr \quad (27)$$

$$P = \frac{TP}{TP + FP} \quad (28)$$

$$R = \frac{TP}{TP + FN} \quad (29)$$

$$FPS = \frac{1}{\text{Process time per frame}} \quad (30)$$

In these equations, *TP* (true positive) refers to correctly predicted positive instances. *FP* (false positive) refers to negative instances incorrectly predicted as positive. *FN* (false negative) refers to positive instances incorrectly predicted as negative.

The experiments were conducted on a server equipped with a NVIDIA Tesla V100 GPU with 32 GB of RAM, running a 64-bit Ubuntu operating system. The implementation was based on the YOLOv9-Seg training framework using Python 3.10, PyTorch 2.8.0, CUDA 12.6, Batch size 64, Training epochs 300, and cuDNN 12.6. All models were trained and evaluated under the same dataset split, class definition, annotation files, and test protocol.

For a fair comparison, all baseline models were trained and evaluated using the same dataset split, annotation files, class definitions, and test set. The primary comparison was conducted at an input size of 640×640 , and all models used the same Ship and Shiptrack categories without class merging or post hoc filtering. Data augmentation was kept consistent across trainable models, including image resizing, horizontal flipping, colour perturbation, and mosaic-style augmentation when supported by the corresponding framework. Pretrained weights were used only as initialization, and all models were fine-tuned on the proposed ship-wake dataset under the same evaluation protocol. For models with different native architectures, such as Mask R-CNN, YOLACT, and YOLO-family segmentors, the output masks and bounding boxes were evaluated using the same IoU thresholds and the same AP calculation pipeline. The lower scores of Mask R-CNN and YOLACT should therefore be interpreted within this controlled protocol: these models were not excluded or penalized by a different test set, but their heavier two-stage or prototype-based segmentation designs are less aligned with the small, low-contrast, and diffuse wake patterns in the collected inland-waterway images.

4.2. Baseline Comparison with Representative Segmentation Models

The experiments were organized in three parts. First, Tables 2 and 3 compare the baseline model with representative instance segmentation methods. Second, Tables 4 and 5 evaluate the contribution of the proposed iAFF, SAConv, and MPD-IoU components. Finally, Table 6 reports the parameter count, inference speed, and computational complexity of different models.

Table 2. Performance of various models on object detection.

Model	Input Size	All Box (mAP@50%)	Ship (Box-mAP@50%)	Shiptrack (Box-mAP@50%)
Mask-RCNN	640 × 640	62.2	65.8	58.6
YOLACT	640 × 640	64.7	68.8	60.6
YOLOv5m-seg	448 × 448	67.1	70.4	65.0
YOLOv5m-seg	544 × 544	70.4	75.9	65.5
YOLOv5m-seg	640 × 640	71.0	74.5	67.5
YOLOv7-seg	448 × 448	70.9	73.4	68.5
YOLOv7-seg	544 × 544	72.7	77.1	68.2
YOLOv7-seg	640 × 640	73.7	79.5	68.0
YOLOv11m-seg	448 × 448	68.8	69.1	68.6
YOLOv11m-seg	544 × 544	73.5	79.4	67.6
YOLOv11m-seg	640 × 640	75.2	79.6	70.8
Baseline	448 × 448	67.3	69.0	65.6
Baseline	544 × 544	77.1	79.6	74.5
Baseline	640 × 640	77.7	79.8	75.6

Table 2 presents a comparative analysis of object detection performance. Among the compared models, the Baseline model achieves the highest overall Box-mAP@50 values, reaching 77.1% at 544 × 544 and 77.7% at 640 × 640. For the Ship class, it obtains 79.6% and 79.8% at these two resolutions, respectively. These results show that the Baseline model provides stable and competitive detection performance across input scales and is therefore selected as the foundational network for subsequent architectural enhancements.

Table 3 reports the instance segmentation results. The Baseline model again performs strongly, obtaining an overall Mask-mAP@50 of 66.5% at 544 × 544 and 67.6% at 640 × 640. At 640 × 640, it achieves 74.7% for the Ship class and 60.5% for the Shiptrack class. These results indicate that YOLOv9-Seg is a suitable baseline for ship and wake segmentation, especially considering its balance between segmentation accuracy and model deployability.

Table 3. Performance of various models on the instance segmentation task.

Model	Input Size	All Mask (mAP@50%)	Ship (Mask-mAP@50%)	Shiptrack (Mask-mAP@50%)
Mask-RCNN	640 × 640	39.8	43.3	36.3
YOLACT	640 × 640	41.3	45.7	36.9
YOLOv5m-seg	448 × 448	55.7	61.0	50.4
YOLOv5m-seg	544 × 544	60.6	65.2	55.9
YOLOv5m-seg	640 × 640	63.9	71.4	56.5
YOLOv7-seg	448 × 448	58.5	62.9	54.2
YOLOv7-seg	544 × 544	61.4	66.8	56.0
YOLOv7-seg	640 × 640	64.5	71.9	57.1
YOLOv11m-seg	448 × 448	57.3	58.1	56.6
YOLOv11m-seg	544 × 544	58.5	68.9	48.2
YOLOv11m-seg	640 × 640	66.1	74.3	57.9
Baseline	448 × 448	60.7	65.8	55.5
Baseline	544 × 544	66.5	71.4	61.7
Baseline	640 × 640	67.6	74.7	60.5

As shown in Figure 11, YOLOv9 demonstrates significantly better detection accuracy compared to other models. It excels in boundary localization, target identification, and segmentation precision, evidenced by clearer target contours and fewer false positives and missed detections. This study has chosen YOLOv9 as the baseline network model. With further improvements and optimizations, the model can play a crucial role in predicting ship behaviour and monitoring navigational safety. This choice is not only based on its current high detection accuracy but also on its superior performance in handling complex backgrounds and multiple targets.

The experimental results of the baseline network model are significantly better than those of other models. Therefore, the Baseline model is adopted as the foundational network for improvement, focusing on integrating the iAFF mechanism, SAC, and MPD-IoU loss function. Integrating iAFF will enhance feature representation through refined feature integration. SAC will help capture multi-scale contextual information by dynamically switching atrous rates, improving segmentation accuracy. The MPD-IoU loss function will provide more accurate bounding box predictions by considering multiple points and their distances. To validate these enhancements, ablation experiments will be conducted to analyze the individual and combined effects of iAFF, SAC, and MPD-IoU on the Baseline model's performance.

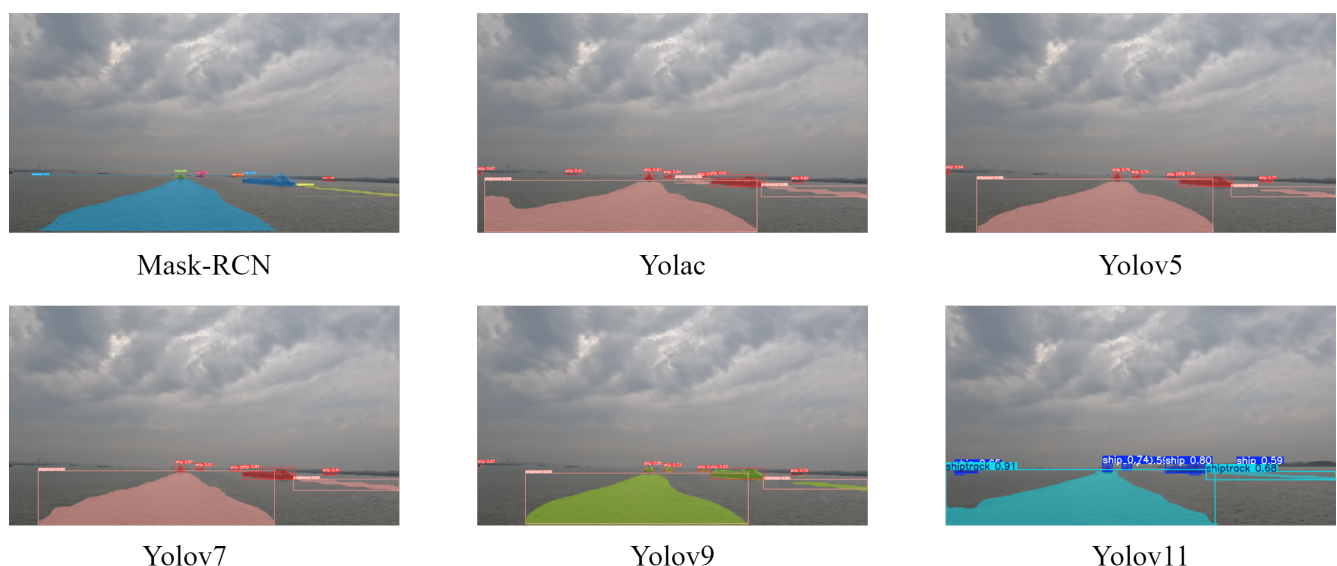


Figure 11. Examples showing that YOLOv9 can accurately identify and segment ship wakes under varying environmental and lighting conditions, supporting its selection as the baseline network model.

4.3. Combined Ablation of the Proposed Adaptations

Additionally, Table 4 presents the object detection performance of different ablation models. The Baseline-Sac-iAFF model improves the overall Box-mAP@50 from 77.7% to 80.5% at 640×640 , showing that the combined feature fusion and Switchable Atrous Convolution modules enhance detection accuracy. After introducing MPD-IoU, the Baseline-Sac-iAFF-MPD model further increases the overall Box-mAP@50 to 84.6%. At the same resolution, the Ship and Shiptrack scores reach 84.5% and 85.0%, respectively. These results demonstrate that the proposed modules are effective not only for ship localization but also for wake-related detection.

Table 4. Comprehensive performance indicators of different models on object detection.

Model	Input Size	All Box (mAP@50%)	Ship (Box-mAP@50%)	Shiptrack (Box-mAP@50%)
Baseline	448 × 448	67.3	69.0	65.6
Baseline	544 × 544	77.1	79.6	74.5
Baseline	640 × 640	77.7	79.8	75.6
Baseline-Sac-iAFF	448 × 448	78.4	78.8	77.9
Baseline-Sac-iAFF	544 × 544	79.2	80.1	78.3
Baseline-Sac-iAFF	640 × 640	80.5	81.2	79.8
Baseline-Sac-iAFF-MPD	448 × 448	81.4	81.5	81.3
Baseline-Sac-iAFF-MPD	544 × 544	83.2	83.0	83.3
Baseline-Sac-iAFF-MPD	640 × 640	84.6	84.5	85.0

For the Ship class, the Baseline-Sac-iAFF-MPD model achieves a Box-mAP@50 of 83.0% at 544 × 544, which is higher than the Baseline model's 79.6% and the Baseline-Sac-iAFF model's 80.1%. The most notable improvements are observed for the Shiptrack class. At 544 × 544, the Baseline-Sac-iAFF-MPD model reaches 83.3%, compared with 74.5% for the Baseline model and 78.3% for the Baseline-Sac-iAFF model. At 640 × 640, the Shiptrack score further increases to 85.0%, whereas the Baseline and Baseline-Sac-iAFF models reach 75.6% and 79.8%, respectively.

These data enhancements illustrate that the Baseline-Sac-iAFF-MPD model can not only improve overall performance but also provide significant gains in accurately detecting and segmenting ships and their tracks. The full configuration attains the strongest performance among the tested variants, while the isolated results indicate that the improvement arises from multiple stages of the network.

Table 5 presents the instance segmentation performance of the ablation models. At 640 × 640, the Baseline model obtains an overall Mask-mAP@50 of 67.6%, while Baseline-Sac-iAFF improves it to 78.6%. With MPD-IoU, the Baseline-Sac-iAFF-MPD model reaches the best overall Mask-mAP@50 of 79.8%, with 83.3% for Ship and 76.3% for Shiptrack. These results indicate that iAFF and SACnv substantially improve feature representation for segmentation, and MPD-IoU further stabilizes localization and mask prediction for complex wake regions.

Table 5. Comprehensive performance indicators of different models on instance segmentation tasks.

Model	Input Size	All Mask (mAP@50%)	Ship (Mask-mAP@50%)	Shiptrack (Mask-mAP@50%)
Baseline	448 × 448	60.7	65.8	55.5
Baseline	544 × 544	66.5	71.4	61.7
Baseline	640 × 640	67.6	74.7	60.5
Baseline-Sac-iAFF	448 × 448	66.5	67.6	65.5
Baseline-Sac-iAFF	544 × 544	76.0	78.8	73.1
Baseline-Sac-iAFF	640 × 640	78.6	82.9	74.4
Baseline-Sac-iAFF-MPD	448 × 448	67.5	66.3	68.6
Baseline-Sac-iAFF-MPD	544 × 544	74.1	81.7	66.6
Baseline-Sac-iAFF-MPD	640 × 640	79.8	83.3	76.3

For the Ship class, the Baseline-Sac-iAFF-MPD model reaches a Mask-mAP@50 of 83.3% at 640 × 640, higher than the 74.7% achieved by the Baseline model and the 82.9% achieved by the Baseline-Sac-iAFF model. For the Shiptrack class, the proposed model achieves 76.3%, compared with 60.5% for the Baseline model and 74.4% for Baseline-Sac-iAFF. This improvement highlights the effectiveness of integrating MPD-IoU for more challenging wake segmentation targets.

Overall, the Baseline-Sac-iAFF-MPD model demonstrates consistent and significant improvements over other models, particularly in its handling of complex segmentation tasks, making it the superior choice for instance segmentation.

As shown in Figure 12, the improved model significantly enhances wake detection accuracy in columns (a) and (c). In columns (b) and (d), the improved model reduces instances of missed detections and overlapping detections. Overall, the improved model achieves more precise segmentation of ship wakes and other relevant features, demonstrating enhanced performance and robustness under various environmental conditions. These improvements are reflected in clearer and more accurate boundaries, and a reduction in false positives.

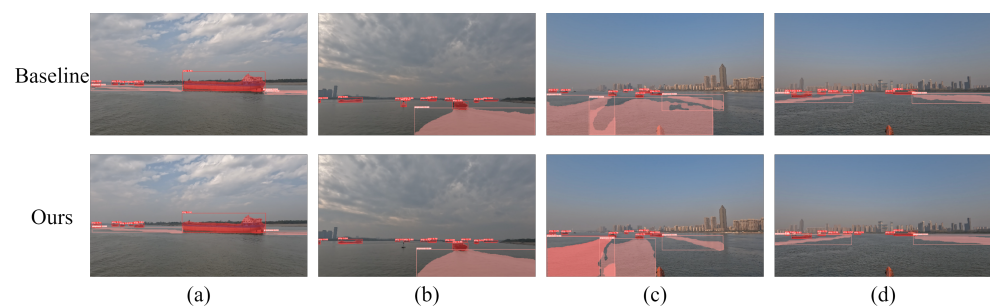


Figure 12. Instance segmentation results. The bottom row (Ours) represents the improved model, and the top row (Baseline) represents the original model. Columns (a–d) correspond to four different navigation scenarios.

4.4. Model Complexity and Server-Side Runtime

Moreover, as shown in Table 6, the Baseline-Sac-iAFF-MPD model reduces the number of parameters from 27.36 million to 25.32 million compared with the Baseline-seg model. GFLOPs refers to Giga Floating-point Operations and represents the computational cost required for a single forward pass, whereas FPS measures inference speed. Compared with the Baseline-seg model, the proposed model improves FPS from 51.5 to 54.6 and decreases GFLOPs from 144.2 to 137.1. This indicates that the proposed model is computationally lighter while achieving higher processing speed.

Table 6. Comparison of model performance metrics.

Model	Parameters	FPS (f/s)	GFLOPs
Mask-RCNN	336,679,283	2.1	258.2
YOLOACT	282,713,751	3.4	235.8
YOLOv5m-seg	21,656,399	61.8	68.8
YOLOv7-seg	37,847,870	38.6	141.9
Baseline-seg	27,363,670	51.5	144.2
Baseline-Sac-iAFF	25,321,814	54.6	137.1
Baseline-Sac-iAFF-MPD	25,321,814	54.6	137.1

To further examine the class-level behaviour of the proposed perception model, a confusion matrix was added, as shown in Figure 13. The matrix evaluates the recognition of the two annotated visual classes, namely Ship and Shiptrack, together with background-related false positives and false negatives. The proposed model correctly identifies 114 Ship instances and 28 Shiptrack instances. Direct confusion between Ship and Shiptrack is limited, indicating that the model can distinguish hull regions from wake regions in most correctly detected cases.

The main errors are associated with the background. Specifically, 35 Ship instances and 12 Shiptrack instances are assigned to the background, while 44 background regions

are predicted as Ship and 11 background regions are predicted as Shiptrack. This pattern suggests that missed detections and background-induced false positives remain the main sources of error. These errors are consistent with the visual characteristics of inland-waterway wake images, where wake boundaries are often diffuse and may be affected by water-surface reflection, foam, low contrast, and partial occlusion.

This confusion-matrix analysis provides additional evidence for the reliability and limitations of the visual perception stage. It should be noted that this matrix is not a manoeuvre-intention confusion matrix. The present work evaluates ship and wake detection and instance segmentation, while quantitative validation of navigational-state recognition or future manoeuvre prediction requires additional ground-truth labels, such as AIS/INS-derived motion states or human-annotated manoeuvre categories.

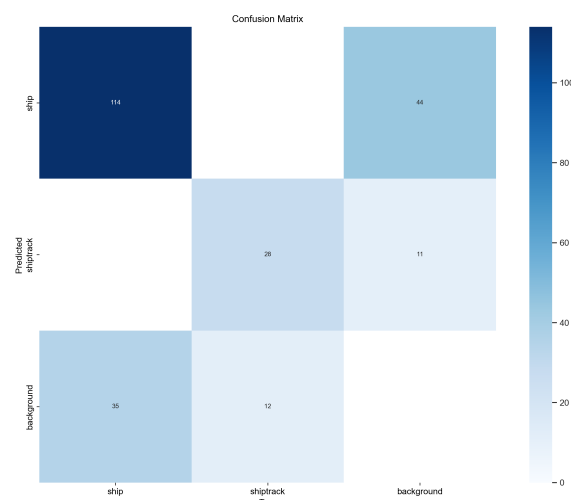


Figure 13. Class-level confusion matrix of the proposed Shipwake-YOLO model for ship and wake perception. Rows indicate predicted classes, and columns indicate ground-truth classes.

4.5. Isolated Ablation Under Strict IoU Criteria

To isolate the contribution of each modification and broaden the evaluation beyond mAP@50, an additional ablation experiment was conducted at an input size of 640×640 under matched training settings. The data split, augmentation strategy, optimizer, batch size, training epochs, and inference protocol were kept unchanged, and only one component was introduced in each single-component variant. V1 corresponds to the YOLOv9-Seg baseline. V2 inserts iAFF into the feature-fusion pathway, V3 introduces SAConv, V4 replaces the selected downsampling path with iSACADown, V5 adapts the high-level aggregation module into iSACSPPELAN, and V6 replaces the bounding-box regression loss with MPD-IoU. The full Shipwake-YOLO model combines these five adaptations. Besides Box-mAP@50 and Mask-mAP@50, Box-mAP@[0.50:0.95] and Mask-mAP@[0.50:0.95] are included to penalize boundary drift and mask misalignment under stricter IoU thresholds, which is particularly relevant for thin and diffuse wake regions.

Table 7 indicates that the performance gain is distributed across feature fusion, receptive-field adaptation, downsampling refinement, high-level aggregation, and localization loss. Compared with V1, the iAFF-only variant V2 raises Box-mAP@50 to 81.2 and Mask-mAP@50 to 73.6, while its stricter IoU scores also increase to 43.3 for boxes and 41.9 for masks. This pattern suggests that adaptive cross-scale fusion helps retain weak wake textures that would otherwise be absorbed by water-surface clutter. V3, which introduces only SAConv, reaches 73.8 in Mask-mAP@50 and 43.2 in Mask-mAP@[0.50:0.95], indicating that receptive-field expansion benefits the spatial continuity of wake masks across stricter IoU thresholds. The iSACADown-only variant V4 produces a more limited improvement,

especially in Box-mAP@[0.50:0.95], which remains close to the baseline; downsampling refinement alone therefore preserves part of the thin wake structure but does not fully resolve high-IoU alignment. V5 secures the highest single-component Box-mAP@[0.50:0.95] of 43.5, suggesting that high-level aggregation contributes to more accurate box geometry by associating fragmented wake regions with the corresponding hull. V6, which changes only the localization loss, obtains the highest single-component Mask-mAP@50 of 74.1, but its Mask-mAP@[0.50:0.95] does not exceed the SAConv-only variant, implying that endpoint-sensitive regression improves coarse wake localization while strict mask alignment still depends on feature quality. The full configuration achieves 84.6 in Box-mAP@50, 48.9 in Box-mAP@[0.50:0.95], 79.8 in Mask-mAP@50, and 46.2 in Mask-mAP@[0.50:0.95]. Relative to V1, these correspond to gains of 6.9, 7.4, 12.2, and 9.1 percentage points, respectively. The stricter IoU results support the revised interpretation that Shipwake-YOLO improves not only loose-overlap detection but also higher-threshold localization and segmentation quality for elongated wake regions.

Table 7. Isolated ablation analysis of Shipwake-YOLO at 640×640 under matched training settings. Both mAP@50 and mAP@[0.50:0.95] are included to evaluate loose and strict IoU criteria.

Model	iAFF	SAConv	iSACADown	iSACSPPELAN	MPD-IoU	Box-mAP@50	Box-mAP@[0.50:0.95]	Mask-mAP@50	Mask-mAP@[0.50:0.95]
V1	×	×	×	×	×	77.7	41.5	67.6	37.1
V2	✓	×	×	×	×	81.2	43.3	73.6	41.9
V3	×	✓	×	×	×	79.9	42.7	73.8	43.2
V4	×	×	✓	×	×	79.3	41.6	71.4	39.5
V5	×	×	×	✓	×	80.3	43.5	72.7	40.7
V6	×	×	×	×	✓	78.8	42.8	74.1	41.3
Ours	✓	✓	✓	✓	✓	84.6	48.9	79.8	46.2

Note: ✓ indicates that the component is included in the variant, whereas × indicates that the component is not included.

5. Discussion

5.1. Integration of Multi-Source Data for Enhanced Situational Awareness

The integration of radar, AIS, and visual data in the MASS framework represents a significant advancement in maritime technology. Recent research on multimodal ship behaviour anomaly detection also demonstrates that jointly modelling AIS, video imagery, LiDAR point clouds, and water-level signals can improve the robustness of inland-waterway early warning under occlusion, illumination variation, and signal noise. By combining these sources of information, autonomous ships can achieve a higher level of situational awareness and operational safety. This multi-faceted approach not only enhances the understanding of the navigational scene but also plays a critical role in the development and deployment of reliable and efficient autonomous maritime systems.

Single-stage object detection techniques offer an effective approach to detecting ship wakes in real-time within maritime navigation systems. Their ability to balance high detection precision with fast processing time and optimized resource consumption makes them highly applicable for operational use. These methods satisfy the stringent real-time demands of maritime activities, improving both navigational safety and situational awareness. As a result, they play a critical role in ensuring efficient and secured maritime transport.

5.2. Environmental Robustness and Wake Ambiguity

Despite the progress made by single-stage object detection methods in ship wake detection, challenges such as camera resolution limitations and harsh maritime conditions persist, necessitating ongoing research and innovation. To tackle these obstacles, it is essential to improve the quality of visual data, integrate multiple sensors, and create more resilient detection algorithms. By focusing on these areas, the reliability and accuracy of

ship wake detection can be significantly enhanced, leading to better navigational safety and more informed decision-making within the maritime domain.

The adaptability of the proposed model to different maritime environments is a critical aspect of its practical application. In harsh weather conditions, such as heavy rain, fog, or rough seas, the visibility and quality of visual data can be significantly affected, thereby reducing the model's detection accuracy. To address this limitation, integrating data from multiple sensors, such as radar and AIS, can enhance the robustness of the detection system, providing complementary sensor information when visual data is compromised. In busy port environments, the presence of multiple overlapping vessels and complex wake interactions poses another challenge. The model's instance segmentation capability helps distinguish between different ship wakes, but further optimization is needed to improve performance in these highly dynamic environments. Additionally, the model's reliance on visual data makes it vulnerable to occlusions and other visibility issues, which may occur frequently in congested maritime areas.

5.3. Limits of Navigational-State Interpretation

A limitation of the current study is that the downstream navigational-state interpretation has not yet been quantitatively validated using labelled manoeuvre states. The proposed model focuses on ship and wake perception, and the hull–wake relationship is used only as a visual cue extracted from the segmentation masks. In the revised manuscript, hull–wake alignment is defined by mask centroids and principal-axis orientations, while the previous speed-threshold rule has been removed because the current image-level dataset does not provide synchronized ground-truth speed, heading, or manoeuvre labels. Therefore, the present results should not be interpreted as direct evidence of reliable intention recognition or future manoeuvre prediction.

The selection of baseline models was also guided by the intended deployment scenario. The proposed perception module is designed for future deployment on resource-constrained maritime edge platforms, such as Jetson Nano-class onboard devices, where memory footprint, inference latency, and power consumption are more restrictive than on server GPUs. Transformer-based segmentation models, such as Mask2Former, SegFormer, and DINO-derived detectors, can provide strong global-context modelling, but their attention mechanisms and larger memory demand are less aligned with the lightweight real-time objective of the present study. For this reason, the main comparisons focus on Mask R-CNN, YOLACT, and YOLO-family segmentation models, which represent conventional two-stage, prototype-based, and single-stage real-time segmentation paradigms. This choice does not imply that transformer-based methods are irrelevant to ship-wake perception; rather, they are better suited to future cross-domain benchmarking when sufficient computational resources and larger multi-scene datasets are available. The present work prioritizes a lightweight architecture that can be further compressed, accelerated, and deployed on maritime edge hardware.

5.4. Boundary-Aware Evaluation and Dataset Generalization

Wakes are also affected by temporal and environmental factors. River currents, wind waves, bank effects, shallow-water hydrodynamics, overlapping wakes, glare, foam, and partial occlusion may change the apparent direction and shape of a wake in a single image. As a result, a single-frame hull–wake offset may not necessarily correspond to an intentional manoeuvre. A complete validation of navigational-state recognition would require consecutive image frames, AIS-derived motion states, temporal smoothing or sequence models, classification accuracy, state-level F1-score, manoeuvre confusion matrices, false-alarm rates, and lead-time analysis. These elements are left for future work. Although the revised

experiments include mAP@[0.50:0.95] to evaluate stricter IoU thresholds, the present evaluation still does not cover all boundary-aware aspects of wake segmentation. Wake masks often have ambiguous margins because their visual extent changes gradually across foam, water-surface reflection, and low-contrast regions. Future work should therefore include boundary-specific metrics, such as Boundary F-score, size-stratified AP for small, medium, and large wake instances, precision–recall curves for each class, and a larger set of failure cases under glare, low light, occlusion, rain, fog, and overlapping wakes.

The dataset scope is another limitation of the present study. The 2100 images used in this work were collected from the Wuhu Channel of the Yangtze River, so the experiments constitute a single-scene validation of wake-oriented instance segmentation rather than a cross-domain evaluation across maritime environments. This setting is suitable for examining whether ship and wake masks can be learned from inland-waterway visual data, but it does not cover the full variability of vessel types, lighting conditions, seasonal appearance, camera viewpoints, water textures, traffic density, weather, and sea states. Building on that direction, future work will extend the wake dataset to multiple waterways, time periods, weather conditions, and camera viewpoints, and will evaluate cross-scene, cross-weather, cross-time-of-day, and cross-dataset transfer performance. Therefore, the present results should be interpreted as evidence for the feasibility of Shipwake-YOLO in the collected inland-waterway setting, not as a complete demonstration of universal generalization.

5.5. Runtime Constraints for Edge Deployment

The runtime results should also be interpreted with caution. The FPS values in this study were measured on an NVIDIA Tesla V100 server GPU, which does not represent the memory bandwidth, thermal envelope, or power budget of onboard maritime edge devices. Although the proposed model reduces parameters and GFLOPs relative to the YOLOv9-Seg baseline, these indicators alone do not establish real-time deployability on Jetson Nano-class hardware. For this reason, the revised manuscript avoids claiming validated onboard deployment. Future work will profile the model on representative embedded platforms, including Jetson Nano, Jetson Xavier, or Jetson Orin, and will record end-to-end latency, FPS, peak memory usage, and power consumption after deployment-oriented acceleration such as TensorRT conversion, mixed-precision inference, pruning, or quantization.

A future multi-source baseline can be formulated as a late-fusion state estimator. For each detected vessel at frame t , the visual branch would extract wake-related geometric features, including hull–wake centroid offset, angular deviation between hull and wake principal axes, wake length, and wake-mask area. The navigation branch would provide AIS/GNSS/INS-derived speed-over-ground, course-over-ground, heading, yaw-rate proxy, and temporal speed variation. These features could then be concatenated and passed to a lightweight classifier or temporal filter to estimate coarse navigational states such as course keeping, acceleration, deceleration, or turning. Such a baseline would provide a more defensible evaluation of navigational-state recognition than a single-frame visual rule, but it requires synchronized sensor records and labelled manoeuvre states that are not available in the current image-level dataset.

6. Conclusions

This study presents Shipwake-YOLO, a wake-oriented adaptation of YOLOv9-Seg for ship and wake instance segmentation in inland-waterway visual images. The model focuses on the perception stage and extracts hull–wake spatial cues from visual data, rather than directly validating navigation intention recognition or future manoeuvre prediction. The task-oriented adaptations improve ship and wake perception while keeping the segmentation head unchanged. At an input size of 640×640 , the full model reaches 84.6 in

Box-mAP@50 and 79.8 in Mask-mAP@50, with corresponding mAP@[0.50:0.95] scores of 48.9 for boxes and 46.2 for masks. These results indicate improved loose-overlap detection and stricter localization quality for elongated wake regions within the collected Wuhu Channel dataset. The study remains limited by its single-scene dataset, image-level evaluation, and lack of cross-weather, cross-time-of-day, and cross-dataset validation. Future work will expand the dataset across multiple waterways and environmental conditions, incorporate synchronized AIS/GNSS/INS information, and evaluate temporal state-estimation models together with boundary-aware wake segmentation metrics.

Author Contributions: Conceptualization, S.L., X.J. and R.Z.; methodology, X.J.; software, X.J.; validation, X.J. and C.Y.; formal analysis, X.J.; investigation, X.J. and C.Y.; resources, S.L. and R.Z.; data curation, C.Y. and X.J.; writing—original draft preparation, X.J.; writing—review and editing, S.L., X.J., C.Y. and R.Z.; visualization, X.J.; supervision, S.L. and R.Z.; project administration, C.Y. and R.Z.; funding acquisition, S.L. and R.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the Guangxi Science and Technology Infrastructure and Talent Development Program, grant number GUIKE AD25069109. This work was also supported by the International Association of Maritime Universities (IAMU) and The Nippon Foundation in Japan, grant number 20240201.

Data Availability Statement: Data will be made available on reasonable request.

Acknowledgments: The authors would like to acknowledge the support of the International Association of Maritime Universities and The Nippon Foundation in Japan.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations and symbols are used in this manuscript:

w^{gt}, h^{gt}	Width and height of the ground truth bounding box
$\Delta \mathbf{w}$	Trainable weight
δ	Rectified linear unit activation function
B	Batch normalization layer
$g(X)$	Global channel feature attention
H, W	Height and width of the feature map
$M(X)$	Attention weights assigned by MSCAM
r	Hyper-parameter of SAC
W_1, W_2	Convolution operations of the descending and ascending layers
ADown	Attention-based Downsampling
AFF	Attentional Feature Fusion
AIS	Automatic Identification System
BBR	Bounding box regression
CNN	Convolutional Neural Network
ELAN	Efficient Local Aggregation Network
FPN	Feature Pyramid Network
GFLOPs	Giga Floating-point Operations
iAFF	Iterative Attentional Feature Fusion
iSACADown	Feature extraction module based on iAFF with SACConv and ADown modules
MASS	Maritime Autonomous Surface Ship
MPD-IoU	Multi-Point Distance Intersection over Union
MSCAM	Multi-Scale Channel Attention Module

NMS	Non-Maximum Suppression
PAN	Path Aggregation Network
PWConv	Point-wise convolution
SAConv	Switchable Atrous Convolution
SPPELAN	Spatial Pyramid Pooling Enhanced with ELAN
SPPF	Spatial Pyramid Pooling Fast

References

1. Sepehri, A.; Vandchali, H.R.; Siddiqui, A.W.; Montewka, J. The impact of shipping 4.0 on controlling shipping accidents: A systematic literature review. *Ocean Eng.* **2022**, *243*, 110162. [\[CrossRef\]](#)
2. Andreassen, N.; Borch, O.J.; Sydnæs, A.K. Information sharing and emergency response coordination. *Saf. Sci.* **2020**, *130*, 104895. [\[CrossRef\]](#)
3. Gao, M.; Shi, G.; Li, S. Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network. *Sensors* **2018**, *18*, 4211. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Yan, Z.; Cheng, L.; He, R.; Yang, H. Extracting ship stopping information from AIS data. *Ocean Eng.* **2022**, *250*, 111004. [\[CrossRef\]](#)
5. Ljunggren, H. Using deep learning for classifying ship trajectories. In Proceedings of the 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; pp. 2158–2164.
6. Veitch, E.; Alsos, O.A. A systematic review of human-AI interaction in autonomous ship systems. *Saf. Sci.* **2022**, *152*, 105778. [\[CrossRef\]](#)
7. Sutulo, S.; Guedes Soares, C. Review on ship manoeuvrability criteria and standards. *J. Mar. Sci. Eng.* **2021**, *9*, 904. [\[CrossRef\]](#)
8. Nielsen, U.D.; Dietz, J. Estimation of sea state parameters by the wave buoy analogy with comparisons to third generation spectral wave models. *Ocean Eng.* **2020**, *216*, 107781. [\[CrossRef\]](#)
9. Almström, B.; Roelvink, D.; Larson, M. Predicting ship waves in sheltered waterways—An application of XBeach to the Stockholm Archipelago, Sweden. *Coast. Eng.* **2021**, *170*, 104026. [\[CrossRef\]](#)
10. Ma, C.; Liu, Z.; Yang, Y.; Zhan, C.; Chen, X.; Feng, B.; Chang, H. The reverse prediction of the ship principal dimensions based on the Kelvin ship waves. *Ocean Eng.* **2023**, *285*, 115308. [\[CrossRef\]](#)
11. Zhang, Y.; Carballo, A.; Yang, H.; Takeda, K. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS J. Photogramm. Remote Sens.* **2023**, *196*, 146–177. [\[CrossRef\]](#)
12. Qiao, D.; Liu, G.; Lv, T.; Li, W.; Zhang, J. Marine vision-based situational awareness using discriminative deep learning: A survey. *J. Mar. Sci. Eng.* **2021**, *9*, 397. [\[CrossRef\]](#)
13. Yang, Y.; Yan, R.; Wang, S. Integrating Shipping Domain Knowledge into Computer Vision Models for Maritime Transportation. *J. Mar. Sci. Eng.* **2022**, *10*, 1885. [\[CrossRef\]](#)
14. Shao, Z.; Lyu, H.; Yin, Y.; Cheng, T.; Gao, X.; Zhang, W.; Jing, Q.; Zhao, Y.; Zhang, L. Multi-Scale Object Detection Model for Autonomous Ship Navigation in Maritime Environment. *J. Mar. Sci. Eng.* **2022**, *10*, 1783. [\[CrossRef\]](#)
15. Huang, F.; Liu, Z.; Li, X.; Mou, F.; Li, P.; Fan, Z. Vessel Trajectory Prediction Based on AIS Data: Dual-Path Spatial–Temporal Attention Network with Multi-Attribute Information. *J. Mar. Sci. Eng.* **2024**, *12*, 2031. [\[CrossRef\]](#)
16. Volkova, T.A.; Balykina, Y.E.; Bepalov, A. Predicting ship trajectory based on neural networks using AIS data. *J. Mar. Sci. Eng.* **2021**, *9*, 254. [\[CrossRef\]](#)
17. Shi, Y.; Long, C.; Yang, X.; Deng, M. Abnormal ship behavior detection based on AIS data. *Appl. Sci.* **2022**, *12*, 4635. [\[CrossRef\]](#)
18. Murray, B.; Perera, L.P. An AIS-based deep learning framework for regional ship behavior prediction. *Reliab. Eng. Syst. Saf.* **2021**, *215*, 107819. [\[CrossRef\]](#)
19. Thombre, S.; Zhao, Z.; Ramm-Schmidt, H.; García, J.M.V.; Malkamäki, T.; Nikolskiy, S. Sensors and AI techniques for situational awareness in autonomous ships: A review. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 64–83. [\[CrossRef\]](#)
20. Chen, P.; Yang, F.; Mou, J.; Chen, L.; Li, M. Regional ship behavior and trajectory prediction for maritime traffic management: A social generative adversarial network approach. *Ocean Eng.* **2024**, *299*, 117186. [\[CrossRef\]](#)
21. Wang, Y.; Liu, J.; Liu, R.W.; Liu, Y.; Yuan, Z. Data-driven methods for detection of abnormal ship behavior: Progress and trends. *Ocean Eng.* **2023**, *271*, 113673. [\[CrossRef\]](#)
22. Tang, Y.; Mou, J.; Chen, L.; Zhou, Y. Review of ship behavior characteristics in mixed waterborne traffic. *J. Mar. Sci. Eng.* **2022**, *10*, 139. [\[CrossRef\]](#)
23. Yang, D.; Wu, L.; Wang, S.; Jia, H.; Li, K.X. How big data enriches maritime research—A critical review of Automatic Identification System (AIS) data applications. *Transp. Rev.* **2019**, *39*, 755–773.
24. Jahanbakht, M.; Xiang, W.; Hanzo, L.; Azghadi, M.R. Internet of underwater things and big marine data analytics—A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 904–956. [\[CrossRef\]](#)

25. Lin, J.; Diekmann, P.; Framing, C.E.; Zweigel, R.; Abel, D. Maritime environment perception based on deep learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 15487–15497. [[CrossRef](#)]
26. Shao, Z.; Yin, Y.; Lyu, H.; Soares, C.G. A robust method for multi object tracking in autonomous ship navigation systems. *Ocean Eng.* **2024**, *311*, 118560. [[CrossRef](#)]
27. Wang, Y.; Ning, X.; Leng, B.; Fu, H. Ship detection based on deep learning. In Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 275–279.
28. Zhang, R.; Li, S.; Ji, G.; Zhao, X.; Li, J.; Pan, M. Survey on deep learning-based marine object detection. *J. Adv. Transp.* **2021**, *2021*, 5808206. [[CrossRef](#)]
29. Vijayakumar, A.; Vairavasundaram, S. YOLO-based object detection models: A review and its applications. *Multimed. Tools Appl.* **2024**, *83*, 83535–83574. [[CrossRef](#)]
30. Fan, Y.; Ju, D.; Han, B.; Sun, F.; Shen, L.; Gao, Z.; Mu, D.; Niu, L. MASS-LSVD: A Large-Scale First-View Dataset for Marine Vessel Detection. *J. Mar. Sci. Eng.* **2025**, *13*, 2201. [[CrossRef](#)]
31. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [[CrossRef](#)] [[PubMed](#)]
32. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
33. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9157–9166.
34. 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.
35. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
36. Jocher, G.; Chaurasia, A.; Qiu, J. *Ultralytics YOLO*, Version 8.0.0; Computer Software. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 3 June 2026).
37. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning what you want to learn using programmable gradient information. *arXiv* **2024**, arXiv:2402.13616.
38. Dai, Y.; Gieseke, F.; Oehmcke, S.; Wu, Y.; Barnard, K. Attentional feature fusion. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 3560–3569.
39. Qiao, S.; Chen, L.C.; Yuille, A. DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 10213–10224.

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.