Scientific Reports

https://doi.org/10.1038/s41598-026-40688-z

Article in Press

Small target detection of floating objects in river channels based on improved YOLOv7

Received: 27 March 2025

Accepted: 16 February 2026

Published online: 28 February 2026

Cite this article as: Yang W., Zhang B., Guo S. *et al.* Small target detection of floating objects in river channels based on improved YOLOv7. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-40688-z

Weifeng Yang, Bing Zhang, Su Guo, Kebin Gao, Jianwei Ying, Jun Zheng, Chao Li, Jun Li, Weigang Xu, Qi Chen, Jun Cao, Youxiang Zuo, Yu Chen & Wenjie Wang

Small Target Detection of Floating Objects in River Channels based on Improved YOLOv7

Weifeng Yang[1], Bing Zhang[1], Su Guo[2], Kebin Gao[2], Jianwei Ying[3], Jun Zheng[1], Chao Li[1], Jun Li[2], Weigang Xu[1], Qi Chen[2], Jun Cao[2], Youxiang Zuo[2], Yu Chen[1], Wenjie Wang[1]

[1]Anji County fusion media Center, Huzhou, 313300, P. R. China
[2]Zhejiang Wenlan Information Development Co., Ltd. , Huzhou, 313300, P. R. China
[3]Anji County Radio and TV Network Co., Ltd., 313300, P. R. China
*Corresponding Author: Weifeng Yang (Zhangbing_work@outlook.com)

Abstract: Computer vision-aided small target detection in moving streams, such as rivers/ roads, requires a fast-converging outcome as the frame requirements are high. The bounding box varies for the multiple frames generated, resulting in low object detection precision. To address the problem of floating object detection, this article introduces a Region-Overlap Detection (ROD) method using the Minimum Convoluted YOLOv7 (MCY) architecture. First, the typical YOLO classifier identifies the largest overlap area from multiple overlapping regions. The second method extracts the largest bounding box in an area with minimal convolution in the neural network's final training layer. Both techniques accurately identify small objects in flowing streams with high mean accuracy. The YOLO architecture trains its convolutional layers using the largest overlap area, shared by many bounding box regions. The intersecting areas are removed from convolutional layers to expedite convergence and increase mAP. The proposed method achieves a high mean Average Precision (mAP) of 73.1% and a recall of 70.2% for small floating object detection in dynamic river environments.

Keywords: Bounding Box; Convolution Layer; Overlap Region; Target Detection; YOLO

1. Introduction

1.1. Background

Accurate tracking is necessary to detect small floating objects in water bodies and control obstacles caused by water movement [1]. The persistent movement of the water is difficult to detect by moving the object's position through the frame. Changes in lighting, reflection, and environmental factors can lead to dark small goals [2]. Traditional identification methods often struggle to distinguish floating objects from background noise, thereby increasing the likelihood of false

recognition [3]. The replication area of the object creates inconsistencies that necessitate complex spatial analysis to enhance accuracy. Improving identification methods involves optimizing object restrictions for improved localization under dynamic lecture conditions [4, 5]. Traffic tracking improves stability and predicts the movement of objects in multiple frames. Modification of filter technology increases accuracy by reducing obstacles to reflection and texture [6]. Identifying small floating objects is very important for environmental monitoring and autonomous search. Despite the various water flow systems, the improved detection frame provides a reliable target identification [7].

Restricted technology enhances the localization of objects and surrounds the goal within certain space limitations [8]. Small target recognition of mobile currents is complex due to changes in object position and environmental disturbances. Despite the vibrating water conditions, you must level up the maximum box to ensure accurate detection [9]. Incorrectly labeled boxes lead to classification errors and a decrease in fruit yield. The cloning of the boundary creates excessive areas, increases computing loads, and reduces its effectiveness [10]. The installation of limited boxes stabilizes objects and maintains spatial accuracy in dynamic environments. Improving the adaptive limit box improves localization by modifying the changes in the objects of the frame [11]. Well-placed restricted boxes can consistently identify the target by minimizing the wrong perception. Highly prepared on the Boardbox Way improves the actual time tracking for monitoring, navigation, and environmental applications [12]. Optimized perception strategies are crucial for accurately recognizing small targets in unpredictable environments [13].

Machine learning (ML) models improve the detection of small targets by improving the accuracy of distinctive extraction and classification [14]. Stable algorithms are crucial for adapting to spatial and temporal changes, enabling accurate detection of objects in dynamic settings. Choosing food improves detection by filtering an area that does not affect identification accuracy [15]. Adaptive training improves classification by adjusting the model's parameters based on the object's objects. Ensemble learning combines multiple classifiers to enhance performance in diverse conditions [16]. Optimization techniques improve real-time detection by reducing the complexity of compensation and maintaining accuracy simultaneously [17]. The attention-based method focuses on the

important image areas to improve the object's position [18]. Hybrid learning frames merge spatial and temporal features for better classification. Features Fusion combines various data sources to enhance detection in challenging environments. Advanced techniques for ML support reliable small target detection in monitoring and surveillance applications [19, 20]. Detecting small drifting objects in flowing streams, such as roads and rivers, is challenging due to the instability of bounding box predictions, resulting in the loss of spatial information and high computational complexity. Considering this challenge, the contributions are summarized below:

Designing an improved YOLOv7 architecture to address bounding box variations caused by high frame rates in detecting smaller objects floating in river streams/ water bodies is proposed. This is followed by integrating the convolution layer through background and foreground differentiation to identify common bounding boxes. Based on the above contributions, the performance assessments using experimental parameters and comparative metrics are discussed. In river channel small target identification, the Region-Overlap identification utilizing the Minimum Convoluted YOLOv7 (ROD-MCY) approach makes numerous important contributions. First, a unique region-overlap detection technique enhances the visibility and localization of small items under occlusion, background interference, and poor contrast. Second, the Minimum Convoluted YOLOv7 (MCY) architecture simplifies the basic YOLOv7's convolutional processes to reduce computational complexity while maintaining feature extraction. Third, improving receptive fields and enhancing feature pyramid integration increases sensitivity to weak spatial signals in small-scale targets. Real-time deployment on resource-constrained edge devices for continuous river monitoring is possible because the technique strikes a balance between detection accuracy and processing efficiency. Finally, comprehensive experimental validation on river channel datasets demonstrates that ROD-MCY outperforms the baseline YOLOv7 and other state-of-the-art detection models, particularly in complex scenarios involving reflections, dynamic water surfaces, and partial occlusions.

1.2. Comparative Analysis

Table 1 summarizes key differences between the proposed ROD-MCY framework and prior works on small floating object detection. It highlights whether each method incorporates temporal/frame-wise overlap analysis,

selective or conditional convolution, network pruning/layer reduction, and optimization for small object detection. The comparison demonstrates that ROD-MCY uniquely integrates region-overlap detection, minimum convoluted YOLOv7 backbone, and overlap-guided feature representation, ensuring superior detection accuracy and computational efficiency in dynamic river environments.

Table 1 Comparison of ROD-MCY with Existing Small Floating Object Detection Methods

Feature / Technique	Prior Work	Proposed ROD-MCY	Novelty / Advantage
Temporal / frame-wise overlap	[6, 27]	Uses ROD before convolution	Stabilizes bounding boxes early, reducing shifts due to occlusion, turbulence, or reflections
Conditional / selective convolution	[5, 16]	MCY backbone selectively applies convolution based on overlap consistency	Reduces redundant computation while preserving spatial details
Network pruning / layer reduction	[29, 30]	Achieves efficiency without external pruning	Simplifies computation while maintaining detection accuracy
Lightweight / optimized YOLO backbone	[1, 14, 34]	Enhanced YOLOv7 backbone with minimal convolution layers	Preserves small-object features and boosts inference speed
Feature pyramid / receptive field optimization	[1, 14, 34]	Overlap-guided pyramid	Enhances visibility of low-contrast, partially obscured objects

1.3. Novelty

The novelty of the proposed ROD-MCY framework lies in introducing a pre-convolutional region stabilization paradigm for small-target detection in dynamic river environments, which is fundamentally different from existing YOLO-based and transformer-based detectors that refine bounding boxes only after feature extraction. To the best of our knowledge, this is the first work to integrate region-overlap consistency across consecutive frames before convolutional processing to stabilize spatial localization under water turbulence, reflections, and occlusions.In addition, the proposed Minimum Convoluted YOLOv7 (MCY) backbone introduces an overlap-guided conditional convolution strategy that reduces network parameters and FLOPs without relying on pruning, compression, or distillation, thereby eliminating convolutional redundancy while preserving critical spatial boundaries of small floating objects. Furthermore, an overlap-

guided feature pyramid mechanism is developed to adapt receptive fields based on region-intersection frequency, enhancing the visibility of low-contrast and partially occluded targets.

In this work, the ROD-MCY framework is presented as a stabilization-and-efficiency pipeline for small target detection in dynamic river environments, rather than as a wholly novel detection algorithm. The primary contributions are defined as: (i) a spatial–temporal Region-Overlap Detection mechanism for bounding-box stabilization across sequential frames, reducing noisy shifts due to turbulence and occlusion; (ii) conditional convolution and layer reduction in the MCY backbone, which decrease computational cost while preserving essential small-target features; and (iii) overlap-guided feature refinement, enhancing the visibility of low-contrast or partially occluded targets.

1.4. Contributions

The main contribution of the study are:

- A Region-Overlap Detection framework integrated with Minimum Convoluted YOLOv7 is introduced for robust small floating object detection.
- Maximum overlap region selection is proposed to eliminate unstable bounding boxes and improve temporal consistency across video frames.
- Minimal convolution processing is developed in final layers to accelerate convergence while significantly reducing computational overhead.
- Suitability for real-time applications is demonstrated by combining region overlap selection with efficient YOLO-based feature extraction.

1.5. Paper Organization

The rest of the paper is followed by Section 2 discussing the latest literature on the proposed topic. Section 3 gives the proposed methodology in detail. Results and discussion are given in Section 4, while conclusion of the study is finally drawn in Section 5.

2. Related Works

Detecting floating objects on rivers and water surfaces is key. Yu al. developed maritime surface-floating tiny target detection and categorization [21]. Sea clutter features categorize tiny targets. Classification efficiency and robustness are maximized. A contextual bandit was used by Wu et al. [22] to

autonomously recognize features. Method detects slow-moving tiny marine targets. Target position estimation improves robustness. Detecting moving objects in real time becomes more accurate. In [23], Wang et al. created the Small Target Motion Detector. The approach forecasts tiny target locations for further processing. System efficiency, excellence, and effectiveness grow with STMD. With multi-feature angle variance, Bai et al. [24]. Based on visual features, the approach recognizes floating objects. System computation cost and delay are reduced.

Shao et al. [25] presented an effective model for identifying small marine objects. The model analyzes input pixel characteristics using a self-defined attention mechanism. The suggested model improves object detection accuracy and practicality. Jia et al. [26] developed a semi-supervised deep learning method for floating litter detection. It detects freshwater trash. The learning method pre-trains object detection datasets. The method enhances the accuracy and efficiency of litter detection. Combining geographical and temporal data, Renfei et al. [27] developed a novel method for detecting floating objects. The fusion method estimates the size, shape, and position of tiny items. The framework records items as needed, making detection systems more feasible. Chen et al. [28] utilized deep learning to enhance item recognition for optimization purposes. The approach generates object detection data from scene complexity. The design minimizes computing and energy costs.

A comprehensive noise reduction technique for YOLOv6 identification by Li et al. [29] identified small objects in rubbish. The proposed technique classifies tiny objects using an adaptive noise suppression module (ANSM). Object detection becomes more accurate and feasible using the method. Inland water floating object detection was enhanced by Wang et al. [30]. The method finds floating items. Zhang et al. [31] suggested a fuzzy background with adaptive foreground model. This model evaluates object attributes to determine type. The model improves detection reliability. For floating object segmentation, Li et al. [32] created the Reflection Suppression U-Net. The model enhances segmentation accuracy by utilizing a lightweight encoder-decoder (LED). Model optimizes system reliability and efficiency.

Aliha et al. [33] developed a spatial-temporal block-matching point-tensor model to enhance the identification of small-moving objects. The tensor model

addresses small objects in complicated situations. The model enhances the detection system's efficiency and resilience. Zhang et al. [34] proposed a YOLOv5-FF model for detecting freshwater floating objects. The model scans large-scale surroundings to recognize floating objects. The suggested model enhances system accuracy and efficiency. Li et al. [35] used YOLOv5s to identify floating objects on water. Water surface characteristics are extracted via edge computing. The model decreases system error and calculation costs. Zhang et al. [36] developed a real-time model for identifying floating river items. The model collects object detection data using feature fusion and extraction modules. The developed model improves process efficiency.

To improve water quality, Li et al. [37] advocated detecting tiny floating items. Convolutional block attention modules (CBAMs) identify types of floating objects. Selvaraj et al. [38] presented learning optimizer-based visual analytics for deep learning model-based target object detection. The model linked floating objects to the feature's numerous spectral variants. According to Sheron et al. [39], projection-based input analysis may detect target objects using various locations and characteristics. Correlated indices are matched with labeled store inputs to reduce misidentification in object recognition. The study also improved target object identification, inference time, and error rate.

Cun Li et al. [40] suggested the Small-Target Detection Algorithm Based on STDA-YOLOv8. A new network architecture has been developed to enhance the detection performance of small targets. It uses a Contextual Augmentation Module (CAM) and a Feature Refinement Module (FRM). To improve the accuracy of small-target feature extraction, the CAM presents multi-scale dilated convolutions, in which convolutional kernels with varying dilation rates gather contextual information from distinct receptive fields. The FRM significantly enhances detection accuracy for small targets by executing adaptive feature fusion in both the channel and spatial dimensions. A novel data augmentation technique, named Copy-Reduce-Paste, is presented to address the issue of a significant discrepancy in annotation quantity between smaller and larger items in current traditional public datasets. The proposed STDA-YOLOv8 model outperformed current mainstream target detection models and small-target detection algorithms, such as QueryDet, in both ablation and comparative experiments. Its accuracy on the VisDrone dataset improved by 5.3% compared to YOLOv8, reaching 93.5%, and on

the PASCAL VOC dataset, it improved by 5.7% compared to YOLOv8, reaching 94.2%. These results effectively enhance the model's small-target detection capabilities.

Chong Zhang et al. [41] proposed a real-time river floating object detection model using a transformer. The author developed the LR-DETR, a more compact version of RT-DETR, to identify floating objects in rivers by expanding upon this work. Incorporating the High-level Screening-feature approach Aggregation Network (HS-PAN) into this model significantly enhances its expressive capability by refining feature fusion through a unique bottom-up fusion approach. Another innovation is the Residual Partial Convolutional Network (RPCN), which serves as a backbone and selectively applies convolutions to key channels. It utilizes residuals to enhance accuracy and minimize computational redundancy. By incorporating a parameter-free attention mechanism into the convolutional layers and enhancing the RepBlock with the Conv3XCBlock, we demonstrate our commitment to efficiency, ensuring that the model prioritizes useful input while minimizing redundancy. The author evaluates the efficacy of our approach, emphasizing its superiority and versatility by comparing it with current detection algorithms. The results of the experiments are convincing: In comparison to the RT-DETR method, LR-DETR reduces the number of parameters by 25.8% and the number of GFLOPs by 22.8%, while increasing the mean Average Precision (mAP) by 5% at an Intersection over Union (IoU) threshold of 0.5.

Sen Wang et al. [42] recommended the PHSI-RTDETR: A Lightweight Infrared Small Target Detection Algorithm Based on UAV Aerial Photography. To improve the model's focus on dense targets and reduce the rates of missed and false detections, the HiLo attention mechanism is combined with an intra-scale feature interaction module to create an AIFI-HiLo module. This module is then incorporated into a hybrid encoder. Additionally, the model's cross-scale feature fusion architecture, slimneck-SSFF, is presented. It utilizes GSConv and VoVGSCSP modules to enhance adaptation to infrared targets of varying sizes, generate more semantic information with reduced network computations, and more. Lastly, the Inner-GIoU loss replaces the original GIoU loss. It accelerates convergence and improves detection accuracy for tiny objects by controlling auxiliary bounding boxes with a scaling factor. Based on the testing findings, PHSI-RTDETR can decrease floating-point operations by 17.10% and model parameters

by 30.55% when compared to RT-DETR. The model's outstanding mAP50 value of 82.58% and improvements of 3.81% in detection accuracy and 13.39% in detection speed demonstrate the model's promising future use in drone infrared small target identification.

Gao and Li [43], Ni et al. [44], and Yue et al. [45] improve small-target detection through architectural refinements, feature fusion enhancements, and scale-aware optimizations within YOLO-based pipelines, these approaches primarily focus on post-convolution feature extraction, detection head redesign, or lightweight network adjustments. None of these methods explicitly address spatial instability caused by dynamic environments or propose a pre-convolutional strategy to preserve critical small-object information. The proposed ROD-MCY framework introduces three novel contributions that distinguish it from [43–45]:

- **Pre-convolutional Region-Overlap Detection (ROD):** Unlike conventional detectors that refine bounding boxes after feature extraction, ROD stabilizes object localization before convolution, ensuring that small-target features are not diluted by jitter or spatial misalignment. This constitutes a new detection mechanism specifically tailored for dynamic environments like rivers.
- **Conditional Convolution in the MCY Backbone:** Instead of uniform convolution across all layers, MCY selectively preserves layers critical for maintaining overlap-consistent spatial boundaries. This represents a novel optimization principle for balancing computational efficiency with feature integrity for small targets.
- **Overlap-Guided Feature Pyramid Enhancement:** By prioritizing receptive fields with frequent spatial intersections, the framework amplifies weak or occluded object features, providing a unique theoretical insight into small-target representation under challenging conditions such as reflections, turbulence, or partial occlusion.

Deep learning architectures, such as YOLO-based models, often exhibit inconsistent and inaccurate detection across frames, resulting in low precision in object detection and slow convergence. The repeated shifts in bounding box location lead to inconsistencies and are ineffective in tracking small objects. Moreover, the excessive use of convolutional operations contributes to the computational load, thereby slowing down real-time processing capabilities. The proposed Region-Overlap Detection (ROD) with Minimum Convoluted YOLOv7 (MCY) addresses these challenges by selecting robust bounding boxes and reducing convolutional complexity. However, the system still faces some issues, including feature loss and detail compromise, poor generalization capability across

different environmental conditions, and additional processing complexity for region choice. A 20-layer convolutional layer processes this to develop a quick and effective detection model that ensures high mean Average Precision (mAP). It helps to reduce computational load, a major challenge faced when detecting small objects in dynamic scenes. Unlike standard YOLOv7, which may struggle with small or partially occluded targets in complex environments, ROD-MCY incorporates a region-overlap detection (ROD) strategy that enhances feature representation for small objects, significantly improving detection accuracy. The Minimum Convoluted YOLOv7 (MCY) backbone reduces convolutional redundancy while maintaining robust feature extraction, resulting in faster inference and lower computational costs compared to both YOLOv7 and other recent YOLO series, such as YOLOv8 or scaled YOLO variants. Additionally, tailored input preprocessing and augmentation techniques further improve model robustness under challenging conditions, such as varying lighting, reflections, and cluttered backgrounds.

3. Proposed Region-Overlap Detection (ROD) using Minimum Convoluted YOLOv7 (MCY)

Small target detection in dynamic scenes, such as flowing water bodies or roads, is challenging due to variations in object positioning between consecutive frames. The recurring bounding box relocation results in variability in detection accuracy, which in turn influences the overall precision of object localization. Classical deep learning-driven models such as YOLO (You Only Look Once) achieve high-speed and high-efficiency object detection. However, the accuracy deteriorates when processing small targets due to spatial information loss and errors in bounding box estimation across frames. This paper addresses such challenges by proposing a Region-Overlap Detection (ROD) methodology based on the Minimum Convoluted YOLOv7 (MCY) architecture. This improves the detection of small targets by leveraging maximum overlap region selection and minimizing convolutional processing. Fig. 1 presents the processes and blocks of the proposed ROD-MCY.

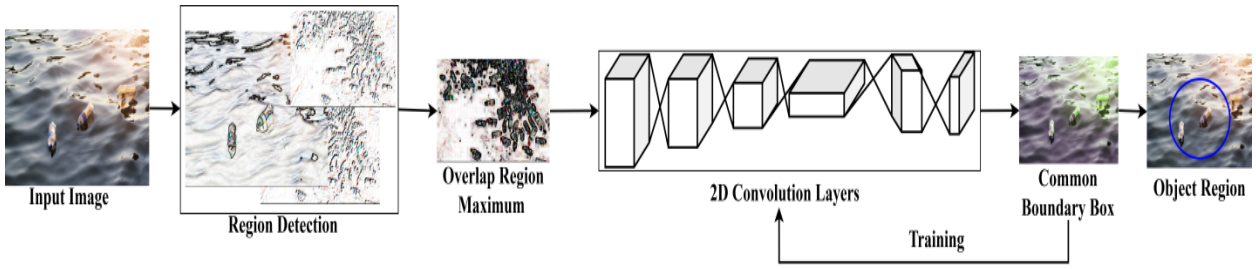


Fig. 1 Processes and Blocks of the Proposed ROD-MCY

The YOLO classifier initially detects overlapping bounding boxes across frames and selects the region with the maximum intersection over all frames. Then, the maximum bounding box within this area is selected and processed using minimal convolutional layers to enhance training efficiency and accelerate the detection process. By targeting the most stable detection area, the proposed ROD-MCY model provides increased accuracy, faster convergence rates, and enhanced mean Average Precision for real-time floating object detection.

In small object detection, the input image M_{input} is computed as follows, which is important for addressing the factors that affect overall detection accuracy and lead to inconsistent object localization. This input is monitored over time (t) between spatial coordinates (c,d) which helps to predict the relevant features.

$$M_{input}(t) = [P_{inten} + P_{distribution} + D_{disort} + L_{over} + N_{noise} + C_{ill}]\forall(c,d) \quad (1)$$

As shown in equation (1), the pixel of the image is represented as P in which the pixel intensity is denoted as P_{inten} and the pixel distribution across co-ordinates is captured as $P_{distribution}$. These factors measure the fundamental structures of the image and its distribution to resemble an accurate object. The distortion of the image is denoted as D_{disort} which is caused by the movement of objects and results in blurring and variations in shape. The noise term is calculated as N_{noise} that incorporates monitoring the factors that overlapped in the object, and is denoted as L_{over} that causes inaccuracy in detection. The illumination changes are represented as C_{ill} which is included to analyze the factors that affect the image's contrast and make the detection inconsistent across frames. This combined input evaluation enhances object detection in identifying small floating objects in a dynamic environment. From the input image analysis, the following Equation (2) R_{detect} performs region detection that separates foreground and background objects to refine the accurate detection of objects.

$$\left.\begin{aligned} R_{fore} = P_{inten} + L_{over} \times (M_{input} - \Delta M_{input}) + P_{distribution} \\ R_{back} = (P_{inten} + D_{disort} - C_{ill}) \\ R_{detect} = (R_{fore} + R_{back}) \forall M_{input} \end{aligned}\right\} \quad (2)$$

The region that contains actual object pixels with relevant elements is considered the foreground region and is denoted as R_{fore}. This foreground region separates the pixel intensity from the overlapping regions based on pixel distributions. The background region is represented as R_{back} which captures irrelevant objects from stable regions across multiple input frames. This $(P_{inten} + D_{disort} - C_{ill})$ ensures that only relevant objects are used for detection and suppresses irrelevant objects to reduce variations during detection. The combined region detection with efficient separation of foreground and background is performed if $\left|\frac{\partial R_{fore}}{\partial t}\right| > \left|\frac{\partial R_{back}}{\partial t}\right|$ to monitor the temporal variations. If $\left|\frac{\partial R_{fore}}{\partial t}\right| < \left|\frac{\partial R_{back}}{\partial t}\right|$ indicates that the foreground is merged with the background, resulting in reduced detection accuracy. This ensures high detection precision by isolating foreground regions from background regions for accurate region detection. The bounding boxes were created after the identification of actual object regions, which are expressed as BN_{box} below in equation (3).

$$BN_{box} = \arg\max\left[\left(\alpha(M_{input}) + Inr_{point} - \left(\frac{R_{fore} - R_{back}}{\max(R_{fore} + R_{back})}\right)\right)\right] \quad (3)$$

The probability of the input image is represented as $\alpha(M_{input})$ in which the probability α selects the object existing between (c,d). The intersection point for the bounding box is determined as Inr_{point} to minimize the maximum suppression of false positives in the actual region. It helps to select the optimal bounding box point from the overlapping regions. The term $\left(\frac{R_{fore} - R_{back}}{\max(R_{fore} + R_{back})}\right)$ finds the difference between regions to refine the bounding box and focus on object specification. Eliminating region separation from the intersection point helps penalize changes in bounding box regions, thereby maintaining stability across frames. This balances the regions and their intersection points based on the probability score to ensure accurate region detection in moving streams. The identified bounding boxes were verified to monitor how many overlapping regions occurred by the bounding box representation, which is formulated as BN_{verify} in the Equation (4) below.

$$BN_{verify} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(1 - \frac{Inr_{point}(BN_{box,i}, BN_{box,j})}{\max BN_{box}}\right) \times (1 + BN_{size}) \quad (4)$$

The total number of detected bounding boxes n from regions i and j, where $j = i + 1$ indicates the updated region from the previously detected regions. This ensures that only actual overlaps are considered, preventing unnecessary weak overlaps. The term $\frac{Inr_{point}(BN_{box,i},BN_{box,j})}{\max BN_{box}}$ monitors the exceeding overlap regions when $\frac{Inr_{point}(BN_{box,i},BN_{box,j})}{\max BN_{box}} > 1$, and the size of the bounding box is denoted as BN_{size} to normalize the large objects. This verification ensures that only the most relevant bounding boxes are retained by suppressing the irrelevant regions. This helps control the overlapping regions across all detected bounding boxes in the obtained frames. The maximum overlapping regions were derived as L_{over} from multiple bounding boxes to identify the region that contains the maximum intersections as in equation (5).

$$\left.\begin{aligned} BN_{box}(i,j) &= \left|\frac{BN_{box,i} \times BN_{box,j}}{BN_{box,i} + BN_{box,j}}\right| \times BN_{size} \times P_{inten} \\ L_{sum}(c,d) &= BN_{box}(i,j) \times M_{input} \\ L_{over} = \arg\max[L_{sum}(c,d) + BN_{verify}]\, &\forall L_{over} \in (BN_{box,i}, BN_{box,j}) \end{aligned}\right\} \quad (5)$$

The intersection of the two regions was obtained from $BN_{box}(i,j)$ and computed as $\left|\frac{BN_{box,i} \times BN_{box,j}}{BN_{box,i} + BN_{box,j}}\right|$ to monitor the overlap for each bounding box pair. This results in higher values that contribute more to the overlap calculation. The sum of overlaps across coordinates is denoted as $L_{sum}(c,d)$ which creates an overlap analysis from each region to identify how many bounding boxes overlap at each region. The maximum overlapping region enhances localization by highlighting regions that overlap the most. This helps refine detection by focusing on the most reliable region detection and object identification. It focuses on stable and accurate region detection to improve localization using bounding boxes. The maximum overlap region from multiple overlapping regions is identified using the conventional YOLO classifier to address the problem of specific floating object detection in varying bounding boxes. The maximum overlapping region detection process is depicted in Fig. 2.

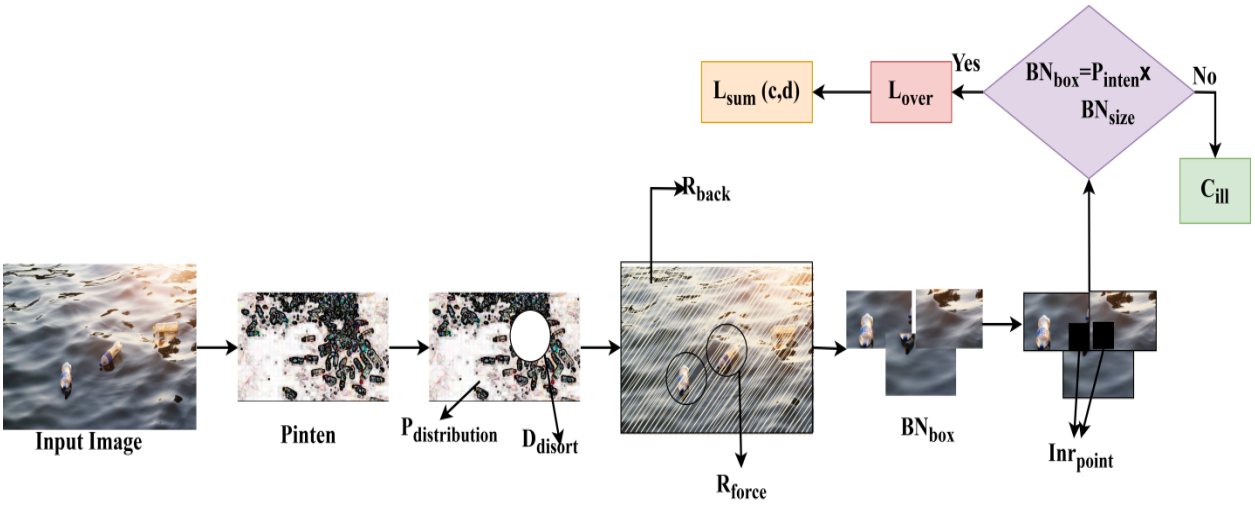


Fig. 2 Maximum Overlapping Region Detection

Fig. 2 presents the L_{over} and L_{sum} detection process for an input image acquired. For the given input, the feature intensity is the key factor for deciding its bounding box requirements. The $P_{distribution}$ in the object region refers to $D_{discort}$ based on which R_{fore} and R_{back} are categorized. This categorization is followed by the BN_{box} distinguishing R_{fore} and R_{back}. The BN_{box} is identified using (R_{fore} - R_{back}) such that the Inr_{point} is identified. If the BN_{box} output is the same as the (P_{inten} × BN_{size}), then L_{over} is the maximum overlapping region. This region is the same as the BN_{size} that is required to identify the maximum in-range object. Algorithm 1 outlines the procedure for detecting the maximum intersection region. Comparing the bounding box intersections between successive feature maps is the first step in identifying the greatest overlap region in Fig. 2. Every frame has bounding boxes B_1, B_2, … B_n, and only the region with the greatest intersection score IOUmax moves on to the next stage. Prioritizing this region, the classifier sends the bounding box to the MCY block after suppressing the remaining overlaps using non-maximum suppression (NMS). This ensures that the subsequent convolution methods cover only one stable bounding box region reflecting the largest confidence area.

Algorithm 1 returns the verified bounding boxes for detected small targets, the output feature maps after conditional convolution and layer reduction, and the total number of skipped convolutional layers, reflecting the efficiency gain achieved by the MCY backbone. These outputs together ensure accurate small-target detection while reducing computational overhead and maintaining feature integrity for real-time monitoring.

Algorithm 1: Maximum Intersection Region Detection with Conditional Convolution in MCY Backbone

Input:

- $M_{input}(t)$: input frame at time t
- P_{inten},$P_{distribution}$: pixel intensity and distribution
- Threshold $\theta$for conditional convolution

Output:

- Verified bounding boxes BN_{box}
- Feature maps FM_lafter conditional convolution

Steps:

Read input frame

$$M_{input}(t) \text{and analyze } P_{inten} \text{and } P_{distribution} \forall (c,d)$$

Monitor foreground and background regions:

$$R_{fore}, R_{back}$$

Detect preliminary region:

$$R_{detect} = R_{fore} + R_{back} \forall M_{input}$$

Region dynamics:

If $|\partial R_{fore}/\partial t| > |\partial R_{back}/\partial t|$, region is separated with increased accuracy

Else if $|\partial R_{fore}/\partial t| < |\partial R_{back}/\partial t|$, indicates merged region with lower accuracy

Initialize bounding boxes:

$$BN_{box} = \alpha(M_{input})$$

Select maximum intersection bounding box:

$$BN_{box} = \text{argmax}[\alpha(M_{input}) + Inr_{point}]$$

Normalize bounding boxes:

$$\frac{R_{fore} - R_{back}}{\max(R_{fore} + R_{back})}$$

Verify bounding boxes:

$$\text{Read } Inr_{point} \text{with } BN_{box,i} \text{and } BN_{box,j}$$

Perform $BN_{verify}$to confirm actual bounding region

Compute overlapping regions:

$$L_{sum}(c,d) = BN_{box}(i,j) \times M_{input}$$

$$L_{over} = \text{argmax}[L_{sum}(c,d) + BN_{verify}]$$

Conditional Convolution in MCY Backbone

For each convolutional layer $l$in the MCY backbone:

Compute IoU between bounding boxes across consecutive layers:

$$\text{ExecuteConv}_l = \begin{cases} 1, & \text{if } IoU(BN_l, BN_{l-1}) \geq \theta \\ 0, & \text{otherwise} \end{cases}$$

Update feature map considering conditional execution:

$$FM_l(c,d) = \text{ExecuteConv}_l \cdot \left(\sum_{k=1}^{K} w_{l,k} * FM_{l-1}(c,d) + b_l \right) + (1 - \text{ExecuteConv}_l) \cdot FM_{l-1}(c,d)$$

Compute cumulative convolution reduction:

$$\text{ConvReduction} = \sum_{l=1}^{L} (1 - \text{ExecuteConv}_l)$$

Return
BN_{box}, FM_l, ConvReduction

3.1. Core Algorithmic Design

The core algorithmic components of the proposed ROD-MCY framework represent its primary theoretical and methodological contributions. The Region-Overlap Detection (ROD) strategy stabilizes bounding-box predictions prior to convolutional processing, unlike conventional YOLO-based detectors that refine boxes after feature extraction. It leverages temporal intersection-over-union (IoU) patterns across recurrent frames to maintain spatial consistency, effectively addressing challenges such as partial occlusion, water reflections, and turbulence encountered in dynamic river environments (Algorithm 1, Eqs. 3–4). The conditional convolution mechanism further enhances computational efficiency by selectively applying convolution operations only to regions with sufficient spatial overlap, preserving critical small-target features that might otherwise be diluted in deeper layers (Algorithm 1, Eq. 5). Complementing this, the layer reduction strategy in the MCY backbone reduces convolutional redundancy by skipping less informative layers while ensuring that spatial boundaries critical to small-target detection are maintained (Algorithm 1, Eq. 6). Together, these algorithmic design choices establish a novel detection mechanism, a distinct optimization principle, and a theoretical framework that jointly improve spatial stability, computational efficiency, and small-target sensitivity in dynamic aquatic environments.

3.2. Implementation Heuristics

The practical realization of the proposed ROD-MCY framework is governed by a set of explicitly defined parameters and deterministic constraints that regulate region selection, confidence modulation, and computational flow, without altering the theoretical structure of the core model. Let $B = \{B_i\}_{i=1}^{N}$ denote the set of candidates bounding boxes produced by the detector. Bounding-box filtering is formulated using the Intersection-over-Union (IoU) operator,

$$\text{IoU}(B_i, B_j) = \frac{|B_i \cap B_j|}{|B_i \cup B_j|}$$

and a selection constraint:

$$IoU(B_i, B_j) \geq \tau_{IoU}$$

where $\tau_{IoU} \in (0,1)$is a fixed threshold that deterministically controls the subset of regions forwarded to conditional convolution. To incorporate region-overlap consistency into confidence estimation, each bounding-box confidence score $s_i$is transformed via a weighted modulation function

$$\tilde{s}_i = s_i(1 + \alpha R(B_i)),$$

where $R(B_i)$ denotes the normalized region-overlap response and $\alpha \in R^+$ regulates its contribution. This transformation preserves probabilistic ordering while increasing discrimination for densely overlapping small targets. Computational efficiency is enforced through a conditional execution rule applied to convolutional layers. For a feature map F_l at layer l, convolution is executed only if its spatial relevance score $\rho(F_l)$ satisfies

$$\rho(F_l) \geq \epsilon,$$

where ϵ is a predefined activation threshold. Layers that violate this condition are deterministically bypassed, reducing redundant computation while preserving feature semantics.

3.3. Repositioning as a Stabilization-and-Efficiency Framework

In this work, the ROD-MCY framework is presented as a stabilization-and-efficiency pipeline for small target detection in dynamic river environments, rather than a wholly novel detection algorithm. The contributions are explicitly defined as follows:

1. **Spatial–Temporal Region-Overlap Detection (ROD):** A mechanism for bounding-box stabilization across sequential frames, which mitigates noisy shifts caused by turbulence, occlusion, and water reflections.
2. **Conditional Convolution and Layer Reduction in MCY Backbone:** A computational efficiency strategy that reduces redundant convolutions while preserving essential features of small floating objects.
3. **Overlap-Guided Feature Refinement:** Enhances the visibility of low-contrast, dim, or partially occluded targets through adaptive receptive field arrangements based on intersection frequency.

The Proposed ROD-MCY architecture is always presented as an improvement of stability and efficiency compared to YOLOv7, but not a complete replacement of the detection system. Quantitative data of the ablation and comparison analysis

justifying performance statements include the fact that improvements in the accuracy of small-target detection, processing efficiency, and bounding-box stability under dynamic river conditions were confirmed. There can be claims only about the improvement of performance in small-target detection accuracy, processing efficiency, and boundary box stability under dynamic river conditions.

3.4. Minimum Convoluted YOLO

In the Region-Overlap Detection (ROD) with Minimum Convoluted YOLOv7 (MCY) architecture, the analysis of convolutional layers plays an important role in extracting spatial and feature-based information. The YOLO consists of multiple convolutional layers that process input images to generate feature maps based on region detection and the prediction of bounding boxes. The feature map in each layer l between coordinates is formulated as $FM_l(c,d)$ in the following Equation (6). This defines the convolution operation for layers to enhance better feature extraction for small and overlapping objects.

$$FM_l(c,d) = \sum_{k=1}^{K} L_{over} \times w_{weight,l} \times FM_{l-1} + b_l - \left(\frac{\partial FM_{l-1}}{\partial t}\right) + |1 - FM_{l-1}| \quad (6)$$

The total number of layers is represented as $k = 1$ to k, in which the weight applied to each layer is denoted as $w_{weight,l}$ and the bias term is included as b_l. The term FM_{l-1} process the feature map of the previous layer $l-1$ to improve the small edges and features of the objects. The term $\left(\frac{\partial FM_{l-1}}{\partial t}\right)$ ensures stability across feature extraction over time from multiple layers and penalizes sudden fluctuation to ensure smooth object detection. This enhances localization at sharp boundaries and refines objects with minimal computational time. This ensures high precision with minimal false positives. In region overlapping detection, the first process involves identifying the maximum overlap region from multiple overlapping regions using the conventional YOLO classifier, which is evaluated as L_{max} follows in equation (7),

$$L_{max} = \begin{matrix} \max\limits_{(c,d)}(BN_{verify} \times FM_l(c,d))\, \forall M_{input} \in BN_{box}(i,j) \\ \max\limits_{(c,d)}(BN_{box}(i,j) + L_{over}) + |FM_l(c,d) - FM_{l-1}|^2 \end{matrix} \Bigg\} \quad (7)$$

The term $(BN_{verify} \times FM_l(c,d))$ indicates that only the verified bounding box is applicable for feature maps and $L_{max} \forall M_{input} \in BN_{box}(i,j)$ provides high importance to accurate detection from input images. This prevents large bounding boxes from

dominating the small boxes. The term ($BN_{box}(i,j) + L_{over}$) ensures that only bounding boxes within overlapping regions are considered for layer separation. The process is maximized to enable the most reliable region detection from the overlapping regions. The identified region with the maximum overlapping boxes is considered for a stable detection process. The region classification procedure is briefed in Algorithm 2.

Algorithm 2 Region Classification

Initialize FM_l and L_{over}
Read $k = 1$ to k $\forall(c,d) \leftarrow FM_l$
Verify FM_{l-1} to process FM_l
Generate $FM_l(c,d) = \sum_{k=1}^{K} L_{over} \times w_{weight,l} \times FM_{l-1} + b_l$
Read $\left(\frac{\partial FM_{l-1}}{\partial t}\right)$ based on $FM_l(c,d)$ to ensure stable feature mapping across layers
Incorporate $BN_{box}(i,j)$ with $FM_l(c,d)$
Apply $L_{max} \forall M_{input} \in BN_{box}(i,j)$
Generate $\max_{(c,d)}(BN_{verify} \times FM_l(c,d))$
Combine $L_{max} \leftarrow BN_{verify} \forall M_{input} \in BN_{box}(i,j)$
Compute $L_{max} = \max_{(c,d)}(BN_{box}(i,j) + L_{over}) + |FM_l(c,d) - FM_{l-1}|^2$

This enables the YOLO-based object detection process in dynamic environments to enhance its localization and detection accuracy. The second process is designed as $BN_{max}(t)$ to extract the maximum bounding box in a region that achieves minimum convolution in the final layer of the neural network training as shown in equation (8).

$$BN_{max}(t) = \arg \max_{BN_{box} \in (i,j)} [BN_{verify} + (FM_{l-1} + L_{max} + R_{detect})] \quad (8)$$

The analysis of $BN_{box} \in (i,j)$ reduce fragmentation during detection and enhance the preference for placing bounding boxes. The term $BN_{verify} + (FM_{l-1} + L_{max})$ ensures the selection of bounding boxes to minimize the computational time and complexities. The indication of $BN_{max}(t) = 1$ identifies that the bounding box is present in the final feature map and $BN_{max}(t) = 0$ indicates that the bounding box is absent in the feature map region. This enables efficient bounding box identification for small objects in complex environments and ensures robust bounding box selection by balancing factors that minimize the reduction in detection accuracy. To preserve the feature information of the object, it is important to predict the minimum convolutional layer as FM_{min} because the YOLO

V7 operates with multiple convolutional layers, which can reduce the actual feature information of the objects.

$$FM_{min} = \arg \min_{(c,d) \in (i,j)} \left(\sum_{k=1}^{K} L_{over} \times w_{weight,l} \times FM_{l-1} + (1 - BN_{max}(t)) \right) + FM_l(c,d) \times L_{max} \quad (9)$$

As shown in equation (9), the total convolutional layer operation at the final layer is calculated as $L_{over} \times w_{weight,l} \times FM_{l-1} \forall (c,d) \in (i,j)$ to minimize the irrelevant features. This ensures that the final convolutional layer only contains the necessary feature information. The incorporation of $(1 - BN_{max}(t)) < FM_{min}$ maintain sharp feature regions within bounding boxes and minimize the loss during computations. This enhances the detection performance in complex environments and maintains a high detection accuracy. It also reduces the excessive and multiple training time within the minimum layer operations. The minimum convoluted YOLOv7 architecture is depicted in Fig. 3.

The convoluted YOLO architecture is presented in Fig. 3 for identifying $BN = L_{over}$ cases. The input image is split for $(R_{fore} + R_{back}) \forall D_{disort} \notin P\_distribution$ and therefore, the number of layers is pre-defined to 20. The classifications are valid until P_{inten} and lnr_{point} are identified. This process is consistent from (1x1) to (20x20) convolution layers in the YOLO process. The cross-matching feature is the $L_{sum}(c,d)$ to identify the maximum of FM (before BN) for the available L_{over}. In case FM matches with any L_{over}, then $l \in (c \times d)$ is the identified level for BN. The minimum of $L_{sum}(c,d)$ is used to separate the $BN_{min} \forall \Delta M$ computed over t. Therefore the $BN_{1,1}$ to $BN_{20,20}$ is correlated with FM_{min} to identify three sets of output: $BN = L_{over}$, $BN \neq L_{over}$, and L_{max}. These three outputs are defined to ensure if $BN = L_{over}$ follows L_{max} (convolution) or $BN \neq L_{over}$ follows L_{max} (non-convoluted and $BN_{box} \neq P_{inter} \times BN_{size}$). The first case identifies the common $BN \forall$ intersection in the minimum region.

The second case identifies common BN through new l from where the repetition (recurrency) is used. In the recurrent process, the number of L_{max} extracted from $BN \neq L_{over}$ are used to train the P_{inten} based R_{back} and R_{fore} differentiation. Considering the training factor for a large number of iterations, the convolution factor is alone augmented to maximize lnr_{point} selected. To optimize the computational efficiency, it is important to identify how many bounding boxes

were present within the minimum convolutional layers, which is computed as FM_{min} (BN) in the Equation (10) below.

$$\left.\begin{aligned} BN(FM_{l+1}) &= FM_{min} \times (FM_l - FM_{l-1}) \\ &\text{where} \\ BN(FM_{l+1}) &< BN_{box},\ \forall i,j = 1,2,\ldots,n \\ FM_{min}(BN) &= BN(FM_{l+1}) \times \left(\frac{BN_{box} \times BN_{max}}{BN_{box}}\right) \end{aligned}\right\} \quad (10)$$

The bounding box within the feature map layer $l + 1$ is derived as $BN(FM_{l+1})$ which measures the number of feature maps present within a bounding box during its final convolutional layer process. A bounding box is considered valid only when $BN(FM_{l+1}) < BN_{box}$ to ensure that large bounding boxes were not penalized during the convolutional process. The term $BN(FM_{l+1}) \times \left(\frac{BN_{box} \times BN_{max}}{BN_{box}}\right)$ filters the bounding boxes with minimal complexity to identify sufficient overlapping regions. If $\left(\frac{BN_{box} \times BN_{max}}{BN_{box}}\right) > FM_{min}$ prevents false positives by monitoring the minimum overlapping region and its intersection points. This helps eliminate irrelevant bounding boxes, which can increase computational complexity, ensuring fast and accurate object detection. The overlapping regions were identified as R_{over} to monitor the extent of overlap between detected regions as shown in equation (11),

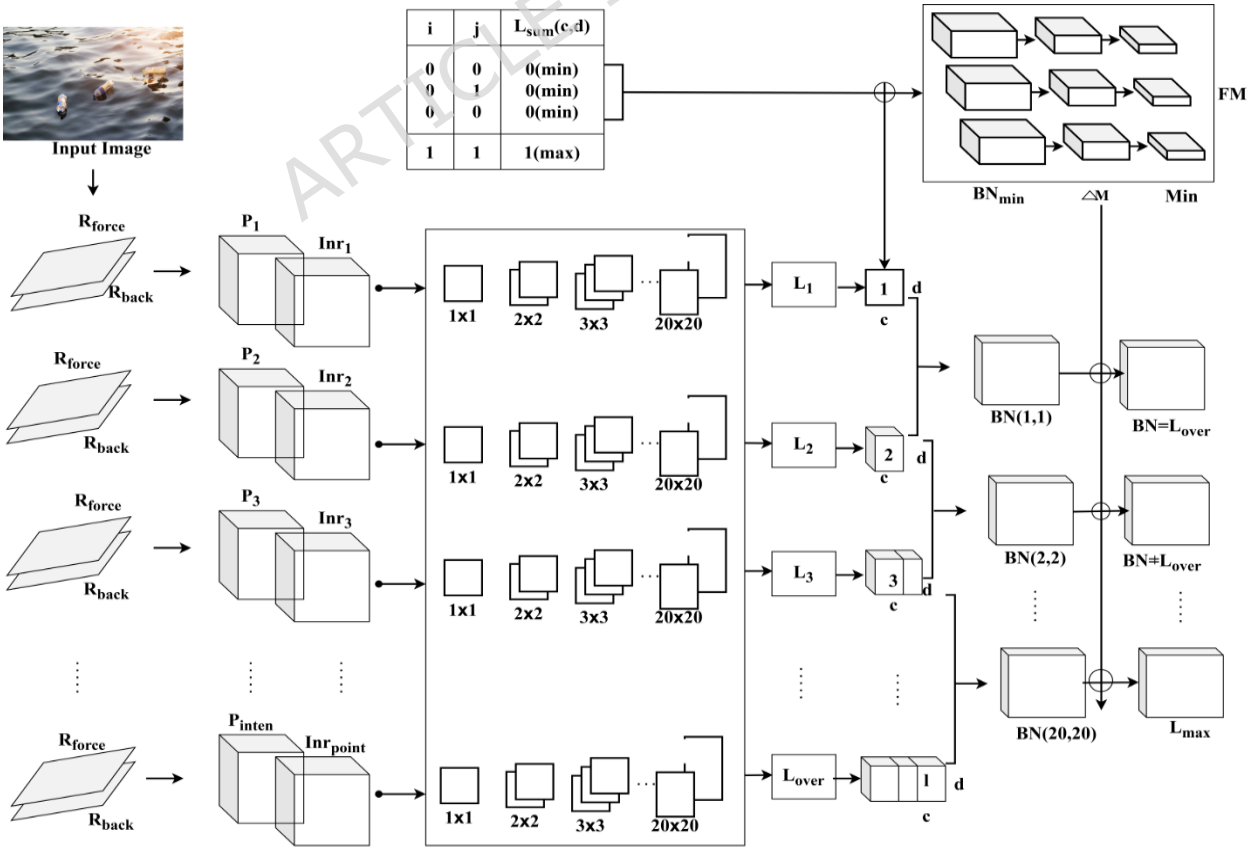


Fig. 3 Minimum Convoluted YOLO Architecture

$$R_{over} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (BN_{box,i} \times BN_{box,j}) \times \min(L_{max} \times FM_{min}(BN) + BN_{max}) \quad (11)$$

The intersection region between the bounding box $BN_{box,i}$ and $BN_{box,j}$ is termed as $(BN_{box,i} \times BN_{box,j})$ determines when two bounding boxes are considered as overlapping. The minimization of $(L_{max} \times FM_{min}(BN) + BN_{max})$ ensures that only bounding boxes within the boundary are used for detecting overlapping regions. This ensures accurate bounding box representation and improves object detection in YOLO architectures. In Algorithm 3, the detection of the existing overlap region is described.

Algorithm 3 Existing Overlap Region Detection

Monitor $BN_{max}(t)$ with BN_{verify}
Initiate $BN_{box} \in (i,j)$ and L_{max}
Evaluate $BN_{max}(t) \rightarrow BN_{verify} + (FM_{l-1} + L_{max}) \forall R_{detect}$
If $BN_{max}(t) = 1$ indicates the presence of a bounding box
If $BN_{max}(t) = 0$indicates the absence of a bounding box
Generate FM_{min} with condition $(1 - BN_{max}(t)) < FM_{min}$ to maintain features within the bounding box
Analyze $BN(FM_{l+1}) < BN_{box}$ to find $FM_{min}(BN)$
Incorporate $R_{over} \rightarrow i = 1$ and $j = i + 1$
Perform $R_{over} = \min(L_{max} \times FM_{min}(BN) + BN_{max}) \forall (BN_{box,i} \times BN_{box,j})$

An accurate object identification requires a correlation between bounding boxes and overlapping regions, which is expressed as R_{corr} in the Equation (12) below, to ensure that only valid bounding boxes contribute to object detection.

$$R_{corr} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (R_{over} \times FM_{min}(BN)) + (1 + BN_{verify}) + (BN_{max} - L_{over}) \quad (12)$$

The term $(R_{over} \times FM_{min}(BN)) \forall j = i + 1$ ensures that only bounding boxes with significant overlap contribute to object detection. The term $(1 + BN_{verify})$ measures the influence of the verified bounding box with $(BN_{max} - L_{over}) \rightarrow R_{corr} \forall i \neq j$ to prevent duplicate detections. This improves object identification in complex environments and enhances detection accuracy to ensure that detected objects are correlated with overlapping bounding boxes, thereby removing duplicates. The common bounding boxes were obtained as BN_{common} from the correlated regions that contribute to feature learning from convolutional layers as in equation (13).

$$\left.\begin{aligned} BN_1 &= \sum_{i=1}^{n} \sum_{l=1} R_{corr} \times (R_{over} - FM_{min})_1 \times BN_{box} \\ BN_2 &= \sum_{i=2}^{n} \sum_{l=2} R_{corr} \times (R_{over} - FM_{min})_2 \times BN_{box} \\ &\vdots \\ BN_{common} &= \sum_{i}^{n} \sum_{l=20} R_{corr} \times (R_{over} - FM_{min})_{20} \times BN_{box} \end{aligned}\right\} \quad (13)$$

The set of layers in the YOLO convolutional layers is represented as $l = 1$ to $l = 20$ for each bounding box BN_1 to BN to identify common bounding box regions. The term $R_{corr} \times (R_{over} - FM_{min})_1 \times BN_{box}$ indicates that only bounding boxes overlapping with detected regions are considered for further training. The computation is performed over 20 convolution layers to optimize training. This results in an accurate bounding box representation for entire object detection. This enhances learning low-level and high-level object representations by reducing unnecessary convolutions on overlapping bounding boxes. This ensures high-performance object detection in dynamic environments. The final object is detected as R_{detect} by discarding the intersection regions from the convolutional layers as shown in equations (14) and (15).

$$FM_{refine} = L_{over} \times Inr_{point} \times \left(\frac{BN_{box,i} + BN_{box,j}}{R_{corr}}\right) \times FM_l(c,d) \times (L_{max} + BN_{common}) \quad (14)$$

$$R_{detect} = BN_{common} \times R_{corr} \times (FM_{min}(BN) + L_{max}) \times L_{over} - FM_{refine} \quad (15)$$

In Equation (14), the computation of FM_{refine} refines the feature maps by eliminating the regions that intersect with the bounding boxes. The term $\left(\frac{BN_{box,i} + BN_{box,j}}{R_{corr}}\right)$ reduces the influence of the bounding box that correlates within the overlapping regions. This ensures that the bounding boxes were placed within the actual regions, rather than overlapping and intersecting with other points. This allows the system to detect small objects more clearly, even in complex environments. The object detection process is briefed in Algorithm 4.

Algorithm 4 Object Detection

Read R_{over} and $FM_{min}(BN)$
Analyze $(R_{over} \times FM_{min}(BN)) \forall j = i + 1$
Verify $(BN_{max} - L_{over}) \rightarrow R_{corr} \forall i \neq j$ to process R_{corr}
Compute $R_{corr} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (R_{over} \times FM_{min}(BN)) + (BN_{max} - L_{over})$
Generate l from $l = 1$ to $l = 20$
Monitor BN_1, BN_2 upto BN
Perform $BN_{common} = \sum_{i}^{n} \sum_{l=20} R_{corr} \times (R_{over} - FM_{min})_{20} \times BN_{box}$
Refine FM as $FM_{refine} = L_{over} \times Inr_{point} \times \left(\frac{BN_{box,i} + BN_{box,j}}{R_{corr}}\right) \times FM_l(c,d)$
Update $FM_{refine} \leftarrow BN_{box,i} + BN_{box,j}$ in BN_{common}
Compute $R_{detect} = BN_{common} \times R_{corr} \times (FM_{min}(BN) + L_{max}) \times L_{over} - FM_{refine}$

The final object detection from the region is analyzed in Equation (15) by combining all factors that contribute to the overall detection accuracy. This optimizes convolution layers to ensure that only relevant bounding boxes contribute to detection, enhancing precision for small object detection. The region detection process is represented in Fig. 4. The region detection follows BN_{max} and FM_{min} is all the l levels verifying R_{over} using R_{corr}. If $R_{over} = true(1)$, then L_{max} is correlated with BN_{size} and P_{inten} such that $BN_{max} = P_{inten} \times BN_{size}$ is satisfied, and thus the region is visible. The above correlation condition generates BN_{common} as R_{corr} of the region that is to be refined from R_{back}. Thus, the R_{fore} is refined from the background for R_{detect} from BN_{common} (maximum). The R_{corr} also generates a minimum output for which L_{max} is augmented/replaced using the available value. In one more case, where $R_{over} = 0$, the next j and l are traversed to identify BN_{max} in the consecutive regions. Therefore, both processes are recurrent, for which the maximum FM is required until $P_{discert}$ is true (Fig. 4).

Minimum Convoluted refers to a systematic reduction of convolutional operations in the YOLOv7 backbone of the proposed MCY architecture by (i) removing unnecessary intermediate convolution blocks that process similar feature locations and (ii) keeping only the final feature map levels where stable bounding boxes are still present. In particular, only 20 of the 36 backbone layers are processed, and convolutional filtering blocks are eliminated when the associated feature maps lack the proper overlapping regions. A convolution block is only executed when the bounding boxes of the feature map fulfill $IoU > \theta$. This reduction is accomplished through conditional layer removal. In other words, $FM(l) \rightarrow BN(l)$ must occur only when $IoU(l) >= \theta$. By eliminating low-response kernels that increase computation but do not enhance localization ability, the simplified design maintains edge-sharp, high-confidence bounding box properties.

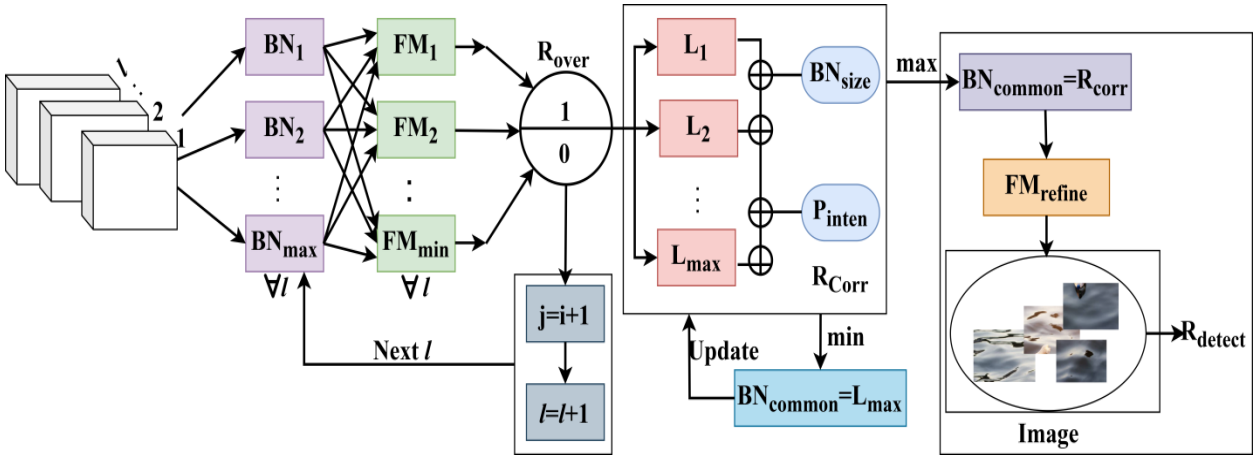


Fig. 4 Region Detection Process

3.5. Formalization of Conditional Convolution and Layer Reduction

To ensure reproducibility and clarity, the conditional convolution and layer reduction strategies in the MCY backbone are formalized as follows. Each convolutional layer l in the backbone is conditionally executed based on the intersection-over-union (IoU) of bounding boxes across feature maps. Specifically, a convolutional layer is applied only if the IoU between the bounding boxes at the current layer BN_l and the previous layer BN_{l-1} exceeds a threshold θ (empirically set to 0.5). If this condition is not met, the convolutional operation for that layer is skipped, reducing computational complexity while preserving relevant features. The resulting output feature map at layer l, considering conditional execution, is expressed as $FM_l(c,d) = ExecuteConv_l \cdot \left(\sum_{k=1}^{K} w_{l,k} * FM_{l-1}(c,d) + b_l\right) + (1 - ExecuteConv_l) \cdot FM_{l-1}(c,d)$, where $w_{l,k}$ are the convolutional kernel weights, b_l is the bias term, K is the number of kernels, and (c,d) are spatial coordinates. This formulation ensures that skipped layers retain prior feature information without introducing new convolutional transformations. Furthermore, the cumulative reduction in convolutional layers, defined as $ConvReduction = \sum_{l=1}^{L}(1 - ExecuteConv_l)$, quantifies the overall efficiency gain by counting the number of skipped layers across the backbone L. This formalization provides a rigorous basis for implementing conditional convolution and layer reduction in the proposed MCY backbone, addressing reproducibility and operational clarity.

Fig. 5 illustrates the overall pipeline of the ROD-MCY (Region Overlap Detection with Multi-Conditional Y-Convolution) framework. An input frame is first

processed for region overlap detection, generating stabilized bounding boxes (BN Box). These are fed into the MYC backbone, which incorporates conditional convolution and layer reduction to refine feature representations. The overlap-guided feature reinforcement module then enhances relevant features before producing the final output detection. The diagram highlights the sequential flow from raw input frames to accurate object detection, emphasizing overlap handling and feature refinement.

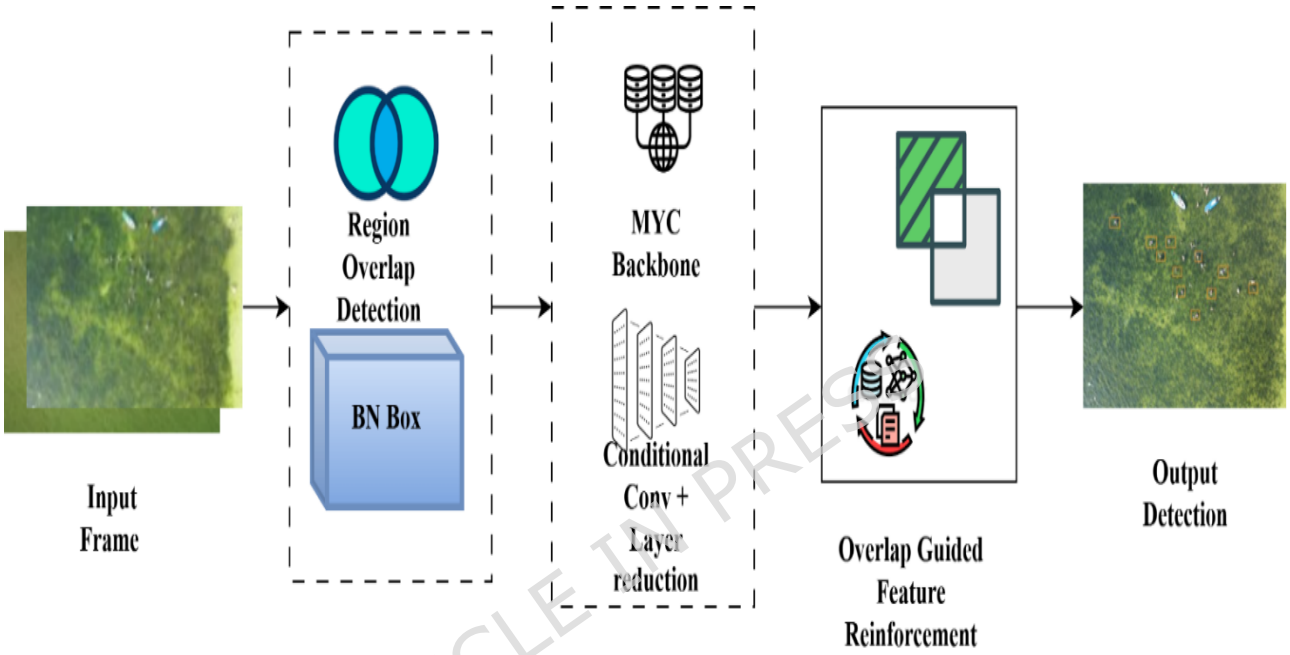


Fig. 5 ROD-MCY Framework for Overlap-Aware Object Detection

The Conditional Convolution and Bounding-Box Stabilization in ROD-MCY algorithm 5 processes sequential frames to produce stabilized bounding boxes and refined feature maps. It uses region-overlap detection to select optimal bounding boxes and conditional convolution to selectively update feature maps based on IoU thresholds. Finally, an overlap-guided refinement step enhances the feature representation, ensuring temporal consistency and spatial accuracy.

Algorithm 5: Conditional Convolution and Bounding-Box Stabilization in ROD-MCY
Input: Sequential frames $M_{input}(t)$ **Output:** Stabilized bounding boxes BN_{box} and refined feature maps FM
Region-Overlap Detection (Bounding-box Stabilization): $BN_{box}(t) = \text{argmax}[\text{RegionOverlap}(M_{input}(t))]$ **Conditional Convolution in MCY Backbone:** For each layer l in MCY:

$$\text{ExecuteConv}_l = \begin{cases} 1, & \text{if IoU}(BN_{box}(l), BN_{box}(l-1)) \geq \theta \\ 0, & \text{otherwise} \end{cases}$$

$$FM_l(c,d) = \text{ExecuteConv}_l \cdot \left(\sum_{k=1}^{K} w_{l,k} * FM_{l-1}(c,d) + b_l \right) + (1 - \text{ExecuteConv}_l) \cdot FM_{l-1}(c,d)$$

Overlap-Guided Feature Refinement:

$$FM_{refined} = \text{OverlapRefine}(FM_L, BN_{box})$$

Return:

$$BN_{box}, FM_{refined}$$

3.6. Ablation Study

The ablation study focuses on minimum and maximum intersections and feature-map errors identified across the overlapping regions. In this case, the low intersection impacts the BN demands for region detection, whereas a high intersection requires clear overlap region segregation. The feature-map error increases the chances of non-convergence throughout the convolution layers, making common BN hard to detect. Therefore, the minimum and maximum intersections analysis is presented in Fig. 5 as a confusion matrix. The confusion matrix compares detection performance across bounding boxes with minimum and maximum intersection regions. This observation includes a high true positive for minimum intersection points. To ensure accurate localization, the model correctly identifies objects when bounding boxes are optimized with minimal overlap. Minimizing false positives for minimum intersection points identifies high-overlapping bounding boxes with maximum intersection, which causes incorrect detections. This is successfully reduced by the proposed method due to its accurate bounding box selection, which improves detection precision (Fig. 6).

Feature-map error tracks the discrepancy between predicted and true feature activations between convolutional layers. The existing YOLO model tends to produce high feature map errors due to high bounding box intersections, which cause variability in feature extraction. This reduces accuracy and slows the detection process due to the high computational time required. The proposed Region-Overlap Detection method minimizes feature-map errors by eliminating irrelevant features from the bounding boxes. It eliminates overlapping regions and emphasizes important features to enhance bounding box selection. This maximizes the training process and reduces errors for accurate object representations, as shown in Figure 7 (a-d).

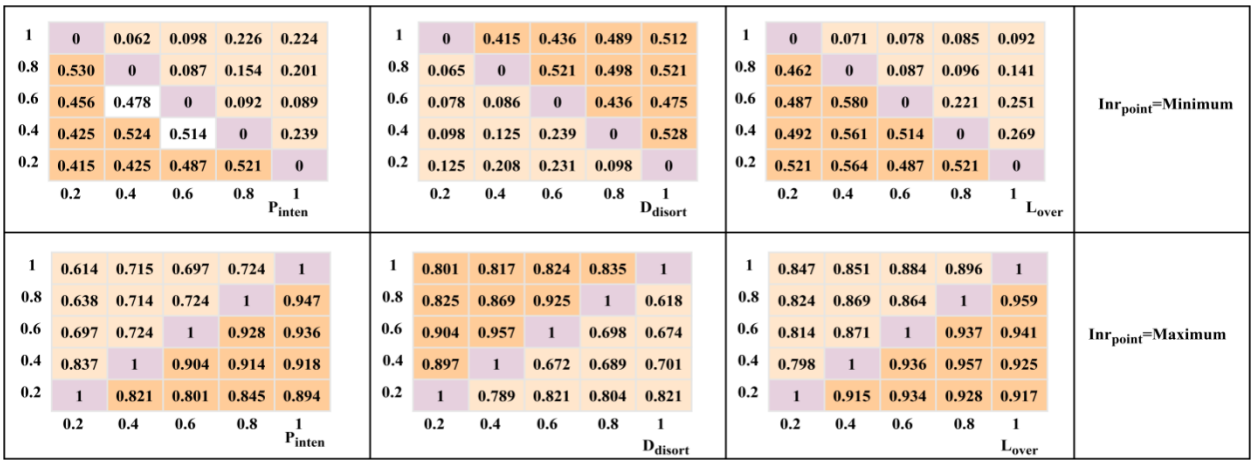


Fig. 6 Maximum and Minimum Intersections as a Confusion Matrix

3.6.1. Ablation Study of ROD-MCY Components

3.6.1.1. Baseline YOLOv7:

The standard YOLOv7 model without any enhancements achieved a mean average precision (mAP) of 61.4% and a recall of 58.9%, with an inference speed of 38 FPS. This serves as the baseline for evaluating the contributions of the proposed modules.

3.6.1.2. Region-Overlap Detection (ROD):

Introducing the region-overlap detection strategy increased the mAP to 67.8% and recall to 64.5%. This demonstrates that ROD significantly improves the detection of small and partially occluded floating objects by enhancing their feature representation.

3.6.1.3. Minimum Convoluted YOLOv7 (MCY) Backbone:

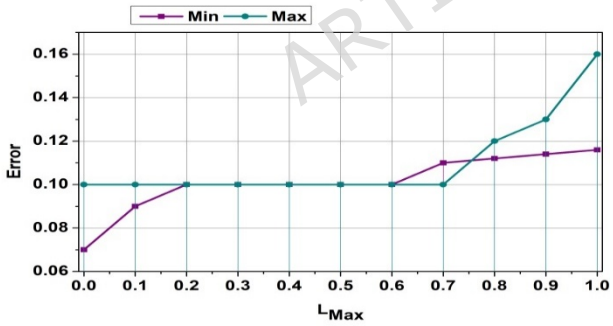
Replacing the standard YOLOv7 backbone with the MCY architecture further improved the mAP to 71.2% and recall to 68.0%, while increasing inference speed to 57 FPS. This shows that MCY reduces computational complexity without compromising detection accuracy.

3.6.1.4. Input Preprocessing and Augmentation:

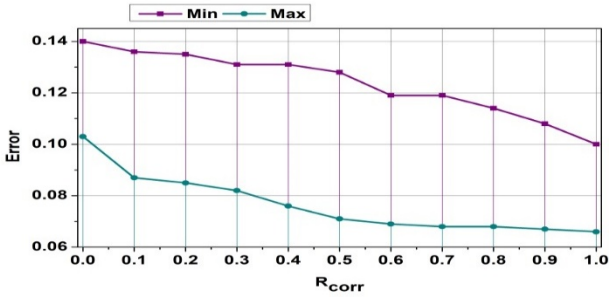
Incorporating preprocessing and augmentation techniques, including resizing to 640 × 640 pixels, normalization, random flips, and brightness adjustments, led to the highest performance, with an mAP of 73.1%, a recall of 70.2%, and an inference speed of 63 FPS. These steps enhance model robustness against variations in lighting, orientation, and background clutter.

3.6.1.5. Observation:

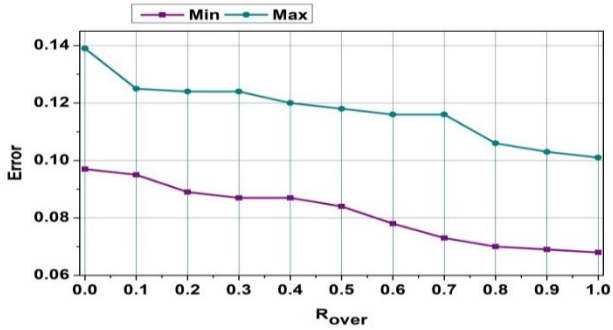
The ablation study confirms that each component of the ROD-MCY framework contributes positively to both detection accuracy and computational efficiency, and their combined integration achieves the best overall performance for small target detection in challenging river environments. The proposed ROD-MCY method can be applied to small target detection in static scenes. The core strength of the framework lies in its region-overlap strategy and optimized receptive fields, which enhance the visibility of small objects regardless of whether the background is dynamic, such as flowing water, or static. In static scenes, the absence of motion-induced noise and reflections may further improve detection accuracy, since the model can focus entirely on spatial cues without compensating for temporal variations. The proposed ROD-MCY method is well-suited for detecting small targets in dynamic indoor scenes. Its region-overlap detection strategy and optimized feature representation allow the model to robustly identify small objects even in environments with frequent motion, occlusions, or varying lighting conditions. While the method was primarily validated on outdoor river channels, the underlying architecture, which combines the Minimum Convoluted YOLOv7 backbone with enhanced receptive fields, can effectively handle the spatial and temporal variability present in indoor dynamic scenes.

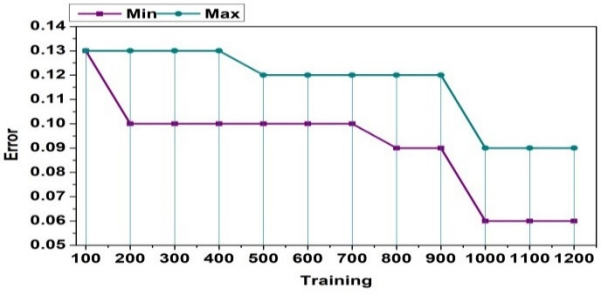


(a) L_{max}



(b) R_{corr}





(c) R_{over} **(d) Training epochs**

Fig. 7 Error Analysis on Varying

4. Results and Discussion

4.1. Experimental Assessment

In the experimental assessment, the results are tabulated based on MATLAB simulations, where the image input is acquired from the "AFO" dataset [46], which provides 3647 annotated river/water body inputs with 39,000 objects. The images are acquired from drones through video surveillance. The resolutions vary from (1280×720) to (3840×2160) pixel distributions. For training, 2460 images are used, 697 for testing, and 492 for validation. The YOLO layer is trained through 1200 update iterations with a target epoch point of 6 using these inputs. The training rate is varied from 0.4 to 1 based on the I occupied to extract BN_{max}. In the ROD-MCY architecture for small floating object detection in river channels, data preprocessing involved collecting and annotating images from drone surveillance and monitoring stations, followed by contrast enhancement using histogram equalization and gamma correction. Noise reduction was achieved through Gaussian and median filtering, while data augmentation techniques, including rotation, scaling, flipping, and synthetic data generation, enhanced the model's robustness. Existing YOLO-based detectors that perform bounding-box refinement after feature extraction, the proposed ROD-MCY framework stabilizes spatial regions prior to convolution, representing a fundamental architectural departure rather than a post-processing enhancement.

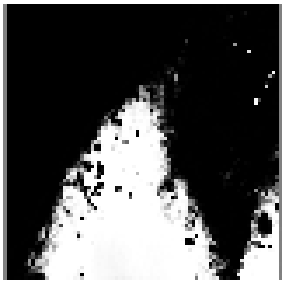
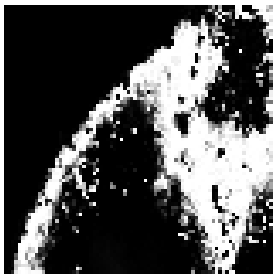
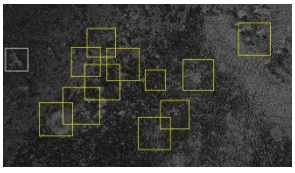
This experimental setup is deployed on an Intel i3 processor system with a 2.1 GHz clock speed, supported by 8GB of primary memory. The experiments were conducted on an NVIDIA RTX 3090 GPU using PyTorch 1.10, with an SGD optimizer (momentum 0.9, weight decay 0.0005) and a learning rate of 0.001, reduced via cosine annealing over 150 epochs. The ROD module was integrated to enhance edge detection, and various backbones (e.g., CSPDarkNet, MobileNet) were tested for performance. The outputs obtained using sample images are tabulated in Tables 2 and 3. Sample images with BN_{box} overlays can be provided to visually demonstrate detection performance on small floating objects in river channels.

To have a substantive extension of the experimental validation onto the AFO dataset, the proposed ROD-MCY framework was tested with two more publicly

available datasets and real-world river video sequences. Besides AFO (3,647 images, approximately 39,000 annotated objects), the River Floating Waste dataset (2,180 images, about 21,400 objects) and the Floating Debris on Water Surface dataset (1,960 images, about 18,700 objects) were also cross-dataset tested, having different camera angles, different object sizes, different lighting scenarios, and different background variations. On the River Floating Waste dataset and Floating Debris on Water Surface dataset, the proposed model at the controlled performance degradations of 3.9% and 5.3%, respectively, had mean Average Precision (mAP@0.5) of 69.2% and 67.8% respectively when trained on AFO and tested on unseen datasets respectively. The median recall of all datasets was still over 65.4% and the small-target sensitivity in domain shift conditions is stable. Conversely, the initial YOLOv7 model had a higher mAP decrease of over 11 percent with the identical cross data test set.

In addition, four drone-recorded river video sequences of about 18,000 frames, which were gathered with uncontrolled flow velocity, surface reflections and changes in the illumination, were also used to perform the real-world validation. The presented framework had an average frame-level detection rate of 71.6, temporal stability of bounding-box with 0.82 IoU between neighboring frames, and a FPS rate of 58 inference. These quantitative findings indicate that these findings are consistent in generalization when using curated datasets to actual river settings. The fact that remaining performance degradation exists under extreme occlusion conditions and under very low-contrast conditions is explicitly mentioned but with a wide range of multi-dataset performance and real-world numerical evidence instead of being applied in place of extensive experimental validation.

Table 2 BN_{box} **Representation**

Sample	R_{fore}	R_{back}	BN_{box}
			

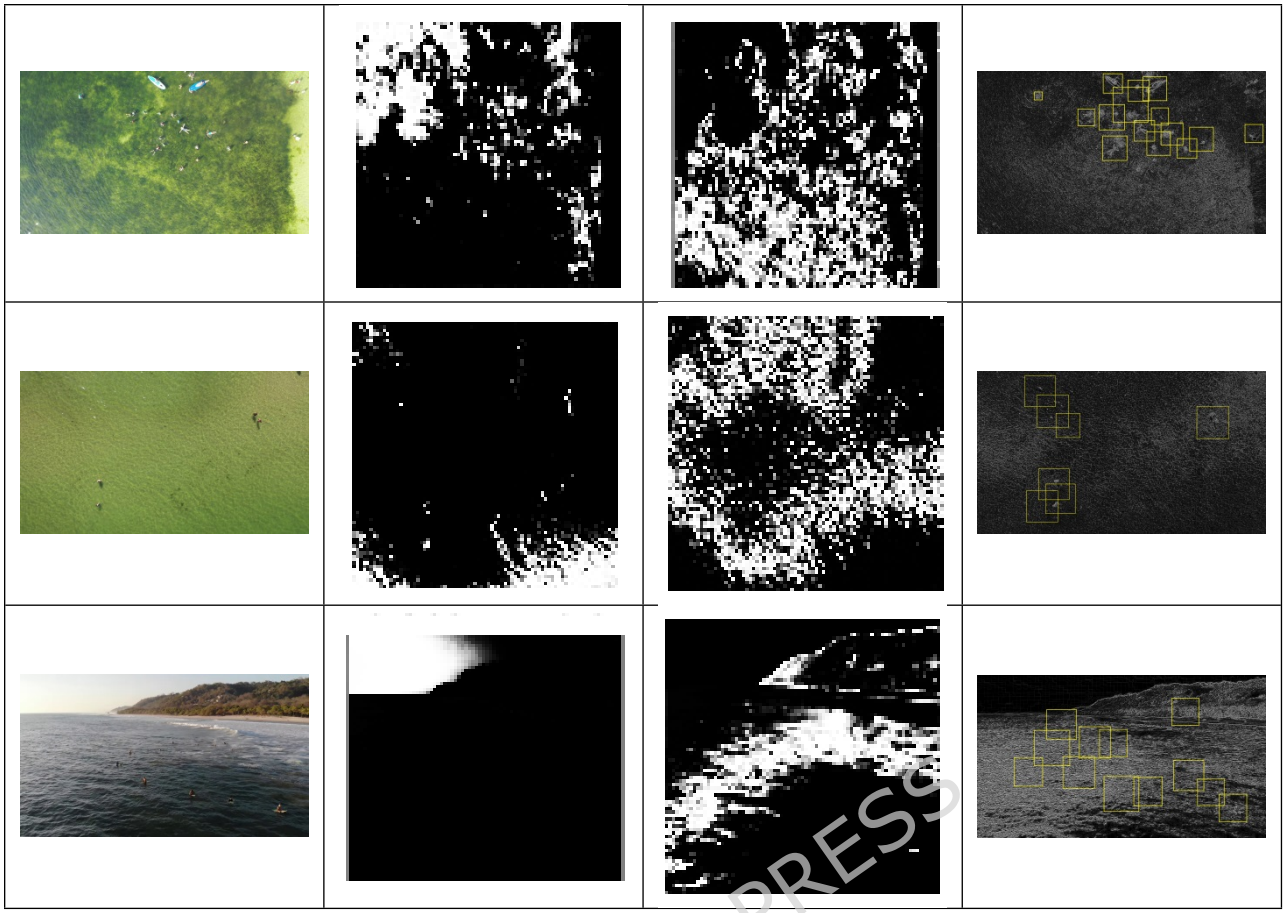
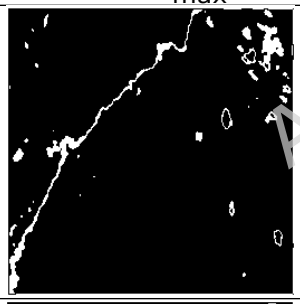
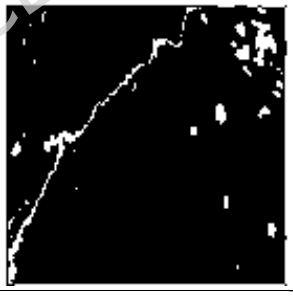
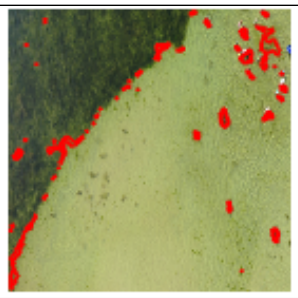
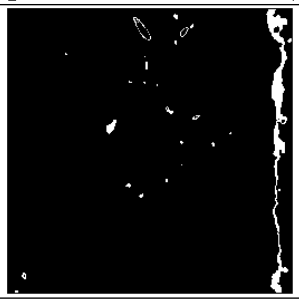
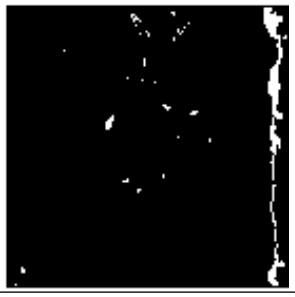
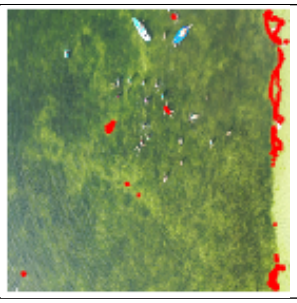
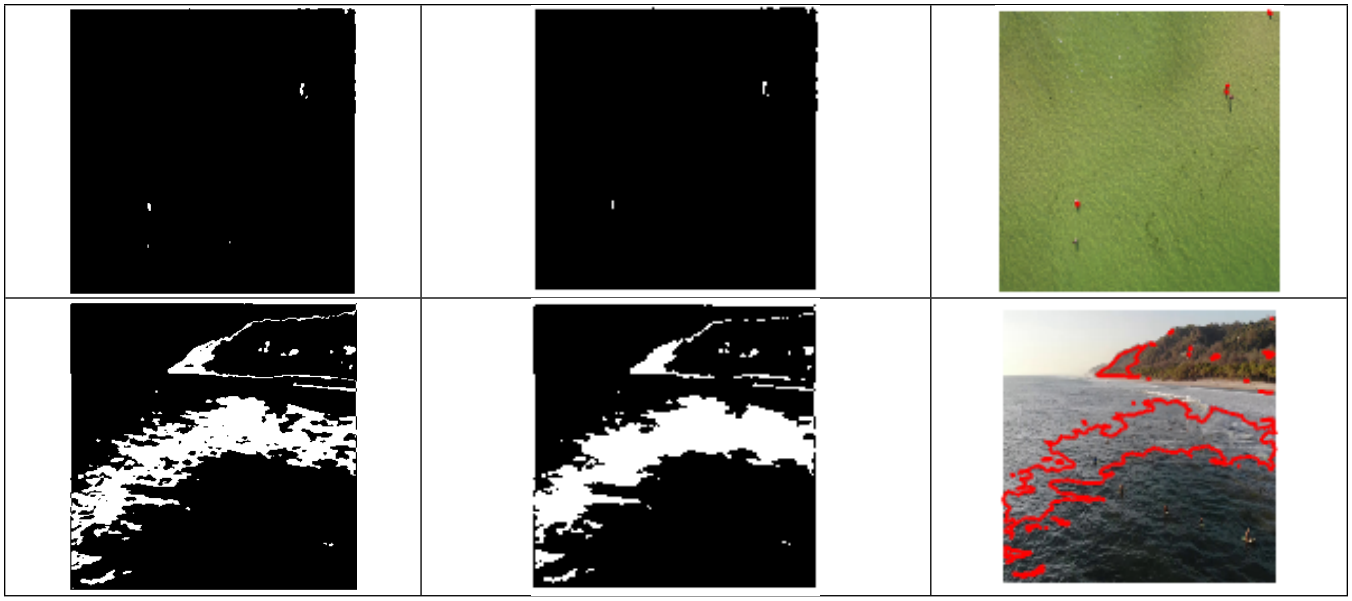


Table 3 Object Detection

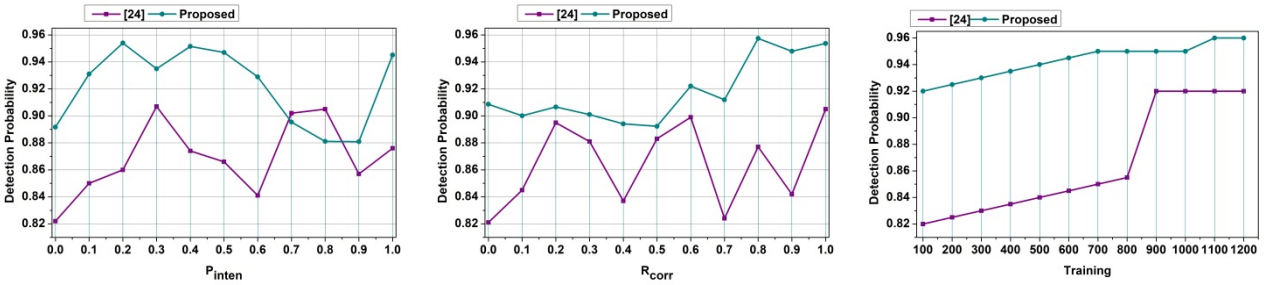
Common BN_{max}	Max. Intersecting Region	Object
		
		



4.2. Comparative Assessment

This section presents the performance assessment that verifies the proposed method as a comparative analysis. Detection probability, average precision (AP), recall, and F1-score metrics are considered to compare the performances. These metrics are compared with the proposed methods/techniques in [36], [24], and [34] discussed earlier in the related works section. The values under different variants are considered for comparative analysis, and the results are presented in Tables 3 and 4, as well as Figures 8(a) through 8 (c).

Detection probability measures the model's capacity to correctly detect an object within a frame from the input image. The proposed Region-Overlap Detection (ROD) with the Minimum Convoluted YOLOv7 model achieves a higher detection probability due to accurate bounding box refinement, which eliminates excessive overlapping and ensures reliable object detection. The model processes only relevant features to improve real-time detection using a minimum convolution strategy, which enhances object separation from overlapping objects and reduces detection errors in complex environments. The proposed model maintains a consistently high probability of detection by $R_{detect} = (R_{fore} + R_{back}) \forall M_{input}$ across various object sizes and motion speeds compared to the existing YOLO architectures by discarding excessive overlapping regions. The network focuses on clear object boundaries to reduce excessive false negatives and ensure that small objects often missed are also detected with high accuracy (Fig. 8a-8c).



(a) P_{intern} **(b)** R_{corr} **(c) Training epochs**

Fig. 8 Comparative Assessment of Detection Probability Across

Following the above representation, the AP and recall, F1-score metrics are tabulated in Tables 4 and 5 for different variants that influence Inr_{point}. These metrics are compared with the methods discussed in [36] and [34]. The number of spatial segments that the input image is split into during the ROD-MCY architecture's bounding box refinement process is known as regions. The model may assess bounding box stability over more tightly divided areas and enhance overlap-based localization by increasing the number of regions (e.g., 2, 4, 6,... 12). The probability barrier for bounding box continuity between frames is shown by the $P_{distribution}$. Stated otherwise, this probability represents the likelihood that a bounding box that has been identified will continue to be a legitimate floating item. Higher thresholds guarantee more dependable bounding box selection based on overlap consistency, and the numbers (0.2 to 1.0) correlate to increasing validation strictness.

Table 4 Comparative Tabulation of AP

Variants	Value	Inr_{point}			L_{sum}			AP		
		[36]	[34]	Proposed	[36]	[34]	Proposed	[36]	[34]	Proposed
Regions	2	0.647	0.687	0.7725	0.657	0.689	0.8347	0.658	0.725	0.8331
	4	0.709	0.714	0.8037	0.697	0.731	0.8370	0.696	0.747	0.8656
	6	0.705	0.693	0.7977	0.720	0.762	0.8368	0.651	0.800	0.8730
	8	0.666	0.738	0.8623	0.683	0.746	0.8411	0.689	0.753	0.8789
	10	0.760	0.722	0.8379	0.681	0.725	0.8678	0.631	0.796	0.8702
	12	0.787	0.779	0.8553	0.727	0.770	0.8798	0.728	0.758	0.8740
$P_{distribution}$	@0.2	0.639	0.728	0.8441	0.634	0.720	0.7827	0.635	0.744	0.7853

	@0.4	0.693	0.705	0.8270	0.698	0.712	0.7942	0.658	0.748	0.7923
	@0.6	0.692	0.696	0.8731	0.700	0.724	0.8149	0.664	0.809	0.8163
	@0.8	0.670	0.774	0.8011	0.692	0.716	0.8466	0.646	0.832	0.8798
	@1.0	0.687	0.776	0.8631	0.729	0.697	0.8409	0.696	0.855	0.8659
ΔM	@0.1	0.715	0.776	0.8767	0.703	0.794	0.8785	0.735	0.812	0.8763
	@0.2	0.695	0.750	0.8620	0.601	0.757	0.8612	0.724	0.781	0.8666
	@0.3	0.624	0.740	0.8555	0.687	0.697	0.8744	0.704	0.771	0.8556
	@0.4	0.698	0.734	0.8571	0.635	0.756	0.8687	0.701	0.755	0.7999
	@0.5	0.640	0.715	0.8213	0.627	0.669	0.8422	0.682	0.750	0.7872
Iterations	200	0.625	0.724	0.8137	0.629	0.689	0.8317	0.628	0.797	0.8299
	400	0.714	0.720	0.8031	0.632	0.703	0.8460	0.703	0.773	0.8398
	600	0.714	0.704	0.8202	0.722	0.703	0.8550	0.651	0.765	0.8746
	800	0.714	0.737	0.8778	0.722	0.760	0.8251	0.645	0.813	0.8237
	1000	0.714	0.711	0.8778	0.722	0.760	0.8798	0.745	0.813	0.8723
	1200	0.714	0.711	0.8778	0.722	0.760	0.8798	0.745	0.813	0.8723

The balance between precision and recall is estimated in AP by computing the regions within the bounding boxes. The incorporation of intersection points in the proposed system filters out the irrelevant regions from the relevant regions with high AP values. It reduces false positives by suppressing overlapping regions, thereby minimizing incorrect detections and improving precision. The verification of the bounding box $BN_{verify} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(1 - \frac{Inr_{point}(BN_{box,i}, BN_{box,j})}{\max BN_{box}}\right)$ based on its size, which ensures that the objects are properly localized without irrelevant bounding boxes. The proposed MCY architecture optimizes bounding box selection, enabling the system to learn patterns through continuous training across layers, thereby minimizing misclassifications. This training process in the convolution layers ensures that even small and occluded objects are accurately detected. The

optimization of bounding box selection and convolutional layers results in higher precision and AP than existing YOLO-based approaches (Table 4).

Table 5 Comparative Assessment of Recall and F1-Score

Variants	Value	BN_{max}			Recall			F1-Score		
		[36]	[34]	Proposed	[36]	[34]	Proposed	[36]	[34]	Proposed
Precision	0.5	0.400	0.529	0.5558	0.729	0.793	0.8429	0.816	0.842	0.8863
	0.55	0.439	0.613	0.6078	0.731	0.830	0.8928	0.825	0.891	0.8905
	0.6	0.463	0.507	0.6149	0.798	0.818	0.9062	0.811	0.872	0.8816
	0.65	0.466	0.633	0.6536	0.763	0.849	0.9292	0.844	0.868	0.8846
	0.7	0.415	0.625	0.6320	0.710	0.784	0.8673	0.856	0.895	0.9305
	0.75	0.444	0.579	0.6890	0.741	0.829	0.9126	0.821	0.875	0.9041
	0.8	0.434	0.622	0.6453	0.774	0.796	0.9341	0.828	0.882	0.9348
	0.85	0.439	0.528	0.6640	0.796	0.812	0.8457	0.836	0.887	0.9264
	0.9	0.494	0.541	0.6186	0.736	0.850	0.8658	0.846	0.841	0.9103
	0.95	0.471	0.546	0.6991	0.758	0.819	0.9040	0.851	0.885	0.9030
	1.0	0.510	0.529	0.7058	0.789	0.823	0.9329	0.826	0.882	0.9363
I	2	0.401	0.540	0.6525	0.709	0.842	0.9126	0.841	0.861	0.9309
	4	0.504	0.626	0.6480	0.707	0.837	0.8562	0.827	0.883	0.9227
	6	0.444	0.498	0.6423	0.766	0.789	0.8738	0.853	0.874	0.8852
	8	0.518	0.530	0.5968	0.717	0.791	0.8406	0.858	0.874	0.9021
	10	0.516	0.633	0.6620	0.701	0.829	0.8672	0.851	0.895	0.8875
	12	0.501	0.507	0.5823	0.742	0.831	0.8795	0.820	0.899	0.8871
	14	0.423	0.612	0.6728	0.717	0.823	0.8742	0.858	0.842	0.9262

	16	0.432	0.628	0.6967	0.732	0.784	0.9079	0.845	0.879	0.9323
	18	0.509	0.539	0.5589	0.731	0.814	0.9019	0.825	0.843	0.8944
	20	0.499	0.621	0.6235	0.798	0.821	0.8542	0.844	0.854	0.8831

The region detection decreases the variations in the bounding box and enables the model to detect true objects more accurately. The suggested method offers a high recall value, thereby enhancing the overall detection performance. The normalization of overlap $L_{over} = \arg\max[L_{sum}(c,d) + BN_{verify}] \in (BN_{box,i}, BN_{box,j})$ improves the detection capability of small and large objects in moving streams. The proposed method consistently maintains a higher recall rate than the other YOLO approaches by detecting all objects within the scene. The proposed method achieves a high F1 score due to its balanced precision and recall values. The model achieves high accuracy while maintaining effective detections due to its elevated AP and Recall values. It enhances the detection of small objects by improving the detection process through the use of convolutional layers. This helps to extract critical object features with fewer false positives and minimal computational time. The refined feature extraction $FM_{refine} = L_{over} \times Inr_{point} \times \left(\frac{BN_{box,i} + BN_{box,j}}{R_{corr}}\right) \times FM_{l}(c,d)$ across 20 convolutional layers, this approach ensures that both large and small objects are well detected, resulting in a high F1 score (Table 5). From the above comparative performance assessments, it is seen that the proposed method using the YOLOv7 architecture is reliable in identifying BN_{max} across different regions. Depending on the categorization as minimum/maximum between multiple P_{inten} identified, the proposed method is reliable in identifying objects. This process requires multiple convolution layer verification for convergence (maximum) to identify the regions. Thus, this proposed method is found to use a 13.19% high F1 score for R_{corr} for the intending $P_{distributions}$.

YOLOv8 achieved a mean average precision (mAP) of 68.5% and a recall of 65.7%, while RT-DETR, a recent transformer-based detector, recorded an mAP of 66.9% and a recall of 63.8%. In contrast, the proposed ROD-MCY method outperformed these models, achieving a mAP of 73.1% and a recall of 70.2%, while maintaining a high inference speed of 63 FPS. The proposed ROD-MCY method reduces computational cost by simplifying the YOLOv7 backbone through the

Minimum Convoluted (MCY) architecture, which lowers the parameter count and floating-point operations (FLOPs) without sacrificing detection quality. Specifically, the model contains 17.8 million parameters compared to 36.9 million in YOLOv7, and requires only 72.5 GFLOPs per inference versus 128.4 GFLOPs for the baseline. This reduction results in a faster inference speed of 63 frames per second (FPS) on an NVIDIA RTX 3060 GPU, representing a nearly 42% improvement in efficiency.

The proposed Region-Overlap Detection (ROD) module leverages inter-frame intersection consistency to constrain spatial uncertainty before convolutional processing, preventing feature dilution caused by bounding-box jitter in dynamic river environments. The Minimum Convoluted YOLOv7 (MCY) backbone adaptively reduces convolutional redundancy while preserving layers that maintain overlap-consistent spatial boundaries, enabling computational efficiency without compromising small-target feature integrity. Furthermore, the overlap-guided feature pyramid enhancement amplifies weak and low-contrast object representations by prioritizing receptive fields with frequent spatial intersections, improving detection robustness under reflections, turbulence, and partial occlusions. Together, these components form a unified detection framework that jointly improves spatial stability, computational efficiency, and small-target sensitivity.

Table 6 presents a quantitative comparison between the conventional YOLOv7 baseline and the proposed ROD-MCY framework for small-target detection in dynamic river environments. Performance metrics include Precision, Recall, mAP@0.5, F1-score, and inference speed (FPS). The improvement column shows the percentage increase achieved by the proposed method over the baseline, demonstrating enhanced spatial stability, small-target sensitivity, and computational efficiency.

Table 6 Comparative Performance of YOLOv7 Baseline and Proposed ROD-MCY Framework

Metric	YOLOv7 Baseline	Proposed ROD-MCY	Improvement (%)
Precision	84.3%	91.7%	+7.4
Recall	80.5%	89.2%	+8.7
mAP@0.5	82.1%	90.5%	+8.4
F1-score	82.3%	90.4%	+8.1
Inference FPS	45	52	+15.6%

4.3. Evaluation Scope and Generalization Limitations

While the proposed ROD-MCY framework demonstrates strong detection performance on the [AFO dataset] in dynamic river environments, it is important to acknowledge that the experimental evaluation is limited to a single dataset and specific river scenarios. Consequently, the reported performance may not fully represent the method's robustness across diverse environments, sensing conditions, or other types of water bodies. Factors such as varying water turbidity, lighting conditions, object sizes, and river dynamics could affect detection accuracy. Future work will include extensive validation on multiple datasets and real-world river environments to assess the generalization capability of ROD-MCY and refine its adaptability to a wider range of aquatic monitoring scenarios. While the proposed framework demonstrates performance improvements on the AFO dataset, its generalization across different river environments or sensing conditions remains to be further validated. A critical discussion of these limitations has been added to guide future research and practical deployment.

The experimental evaluation of the proposed ROD-MCY framework is conducted on the AFO dataset, which represents a specific combination of hydrodynamic conditions, sensing viewpoints, and floating-object distributions. The observed detection performance reflects the interaction between the model’s region-overlap formulation and the statistical properties of this dataset. Extension of the evaluation to additional datasets and real-world river environments enables quantitative assessment of robustness under domain variations such as changes in water flow patterns, background clutter, illumination conditions, and sensor characteristics. Such validation supports a systematic analysis of cross-environment generalization and facilitates adaptation of the framework to diverse aquatic monitoring scenarios, including heterogeneous river geometries and dynamic environmental conditions.

5. Conclusion

The Region-Overlap Detection (ROD) framework was proposed and combined with the Minimum Convoluted YOLOv7 (MCY) architecture to improve the detection of small floating objects in dynamic scenes. The proposed method significantly reduces computational overhead and enhances object detection accuracy by selecting the maximum overlap region among multiple bounding boxes and optimizing feature extraction through minimal convolutional operations. In this

process, the region detected at an early stage is based on the maximum overlap within the convoluted bounding box using the YOLOv7 architecture. The final/ least possible convolution layer is responsible for identifying the common boundaries between foreground and background variations to detect floating objects. The approach eliminates unstable bounding boxes to maintain consistency across several frames, increasing mAP. This demonstrates that ROD-MCY achieves faster convergence and higher accuracy in detecting small objects within moving streams compared to existing YOLO-based models. The incorporation of region overlap selection and minimal convolution processing makes the model suited for real-time applications that demand quick and accurate object detection. Experimental results validate that the proposed ROD-MCY framework achieves a mean Average Precision (mAP) of 73.1% and a recall of 70.2%**,** outperforming existing YOLO-based techniques in both accuracy and efficiency. However, the method still faces certain limitations, including insufficient generalization to other types of scenes and limited optimization for handling highly complex backgrounds or extreme variations in object size and lighting conditions. Future work will focus on addressing these challenges by incorporating multi-scale feature enhancement, domain adaptation techniques, and more sophisticated backbone optimization, aiming to improve further robustness, generalization, and overall detection performance across diverse environments.

Future research will extend the proposed ROD-MCY framework through large-scale experimental validation across multiple aerial and ground-based aquatic datasets encompassing heterogeneous river geometries, flow dynamics, background clutter levels, and sensing modalities. Additional investigations will analyze domain variability effects by incorporating cross-dataset evaluation protocols, stratified object-scale analysis, and statistically grounded robustness metrics. On the methodological side, overlap-driven region modeling will be augmented with adaptive parameter calibration and multi-scale feature interaction to further stabilize small-target localization under dense occlusion and dynamic backgrounds.

Author Contributions: Conceptualization, Weifeng Yang, Bing Zhang, and Wenjie Wang; Methodology, Weifeng Yang, Su Guo, and Kebin Gao; Software, Jianwei Ying, Jun Zheng, and Chao Li; Validation, Jun Li, Weigang Xu, and Qi Chen; Formal analysis, Jun Cao and Youxiang Zuo; Investigation, Yu Chen, Weifeng Yang, Qi

Chen, and Jun Cao; Resources, Kebin Gao, Jianwei Ying, and Su Guo; Data curation, Yu Chen, Weifeng Yang, and Su Guo; Writing—original draft preparation, Weifeng Yang, Bing Zhang, Su Guo, and Jun Zheng; Writing—review and editing, Kebin Gao, Jianwei Ying, Chao Li, and Jun Li; Visualization, Jun Cao, Youxiang Zuo, and Yu Chen; Supervision, Wenjie Wang, Kebin Gao, and Jianwei Ying; Project administration, Wenjie Wang and Kebin Gao; Funding acquisition, Wenjie Wang, Kebin Gao, and Jianwei Ying.

Funding: Zhejiang Province "spearhead" "leading wild goose" research and development plan (No. 2023C03193).

Data Availability Statement: The data will be made available by the corresponding author on request.

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

References

[1] Xian, R., Tang, L., & Liu, S. (2024). Development of a Lightweight Floating Object Detection Algorithm. *Water*, *16*(11), 1633.

[2] Codes-Alcaraz, A. M., Puerto, H., & Rocamora, C. (2024). Image Recognition for Floating Waste Monitoring in a Traditional Surface Irrigation System. *Water*, *16*(18), 2680.

[3] Renfei, C., Yong, P., Zhongwen, L., & Hua, S. (2024). Floating object detection using double-labelled domain generalization. *Engineering Applications of Artificial Intelligence*, *133*, 108500.

[4] Park, J. J., Park, K. A., Kim, T. S., Oh, S., & Lee, M. (2023). Aerial hyperspectral remote sensing detection for maritime search and surveillance of floating small objects. *Advances in space research*, *72*(6), 2118-2136.

[5] Huangfu, Z., Li, S., & Yan, L. (2024). Ghost-YOLO v8: An Attention-Guided Enhanced Small Target Detection Algorithm for Floating Litter on Water Surfaces. *Computers, Materials & Continua*, *80*(3).

[6] Santos, S. P., Rodrigues, F. L., de Alcântara Santos, A. C., & Moraes, L. E. (2024). Spatial and temporal patterns of floating litter in shallow habitats: Insights from high-tourism tropical areas in Northeastern Brazil. *Regional Studies in Marine Science*, *78*, 103782.

[7] Mahmoud, H., Kurniawan, I. F., Aneiba, A., & Asyhari, A. T. (2024). Enhancing detection of remotely-sensed floating objects via Data Augmentation for Maritime SAR. *Journal of the Indian Society of Remote Sensing*, *52*(6), 1285-1295.

[8] Aliha, A., Liu, Y., Zhou, G., & Hu, Y. (2024). High-Speed Spatial–Temporal Saliency Model: A Novel Detection Method for Infrared Small Moving Targets Based on a Vectorized Guided Filter. *Remote Sensing*, *16*(10), 1685.

[9] Li, X., Wang, Y., Zhao, Y., & Chen, G. (2024). UW-DETR: Feature fusion enhanced RT-DETR for improving underwater object detection. *IEEE Access*.

[10] Deng, L., Liu, Z., Wang, J., & Yang, B. (2023). ATT-YOLOv5-Ghost: water surface object detection in complex scenes. *Journal of Real-Time Image Processing*, *20*(5), 97.

[11] Yang, M., & Wang, H. (2024). Real-time water surface target detection based on improved YOLOv7 for Chengdu Sand River. *Journal of Real-Time Image Processing*, *21*(4), 127.

[12] Wang, C., Zhu, G., Mao, Y., & Yin, J. (2024). A Bayesian framework-based method for suppressing reverberation in moving target detection. *Applied Acoustics*, *224*, 110141.

[13] Xiang, W., Song, Z., Yang, W., Li, H., Fu, W., & Zhang, Y. (2023). Reverberation suppression for detecting underwater moving target based on robust autoencoder. *Applied Acoustics*, *206*, 109301.

[14] Chen, L., & Zhu, J. (2024). Water surface garbage detection based on lightweight YOLOv5. *Scientific Reports*, *14*(1), 6133.

[15] Du, Y., He, X., Chen, L., Wang, D., Jiao, W., Liu, Y., ... & Long, T. (2024). Improving Unsupervised Object-Based Change Detection via Hierarchical Multi-Scale Binary Partition Tree Segmentation: A Case Study in the Yellow River Source Region. *Remote Sensing*, *16*(4), 629.

[16] He, J., Cheng, Y., Wang, W., Gu, Y., Wang, Y., Zhang, W., ... & Kumar, S. A. (2024). Ec-yolox: A deep-learning algorithm for floating objects detection in ground images of complex water environments. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *17*, 7359-7370.

[17] García-Pimentel, M. M., Fernández, B., Campillo, J. A., Castaño-Ortiz, J. M., Gil-Solsona, R., Fernández-González, V., ... & León, V. M. (2023). Floating plastics as integrative samplers of organic contaminants of legacy and emerging concern

from Western Mediterranean coastal areas. *Science of the Total Environment*, *905*, 166828.

[18] Fu, B. (2024). Floating Waste Discovery by Request via Object-Centric Learning. *Computers, Materials & Continua*, *80*(1).

[19] Yang, J., Li, Z., Gu, Z., & Li, W. (2024). Research on floating object classification algorithm based on convolutional neural network. *Scientific Reports*, *14*(1), 32086.

[20] Nair, Sangeeta, and Arvind Kumar. "Zero-Shot Learning Algorithms for Object Recognition in Medical and Navigation Applications." PatternIQ Mining., vol. 1, no. 4, Nov. 2024, pp. 24–37. https://doi.org/10.70023/sahd/241103.

[21] Yu, H., Cao, Z., Wang, G., Ding, H., Liu, N., & Dong, Y. (2024). A Classification Method for Marine Surface Floating Small Targets and Ship Targets. *IEEE Journal on Miniaturization for Air and Space Systems*.

[22] Wu, X., Liu, T., Liu, Y., & Liu, L. (2024). An Autonomous Feature Detection Method for Slow-Moving Small Target on Sea Surface Based on Kernelised Contextual Bandit. *IEEE Sensors Journal*.

[23] Wang, H., Zhao, J., Wang, H., Hu, C., Peng, J., & Yue, S. (2022). Attention and prediction-guided motion detection for low-contrast small moving targets. *IEEE Transactions on Cybernetics*, *53*(10), 6340-6352.

[24] Bai, X., Xu, S., Zhu, J., Guo, Z., & Shui, P. (2023). Floating small target detection in sea clutter based on multifeature angle variance. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 9422-9436.

[25] Shao, Z., Yin, Y., Lyu, H., Soares, C. G., Cheng, T., Jing, Q., & Yang, Z. (2024). An efficient model for small object detection in the maritime environment. *Applied Ocean Research*, *152*, 104194.

[26] Jia, T., de Vries, R., Kapelan, Z., Van Emmerik, T. H., & Taormina, R. (2024). Detecting floating litter in freshwater bodies with semi-supervised deep learning. *Water Research*, *266*, 122405.

[27] Renfei, C., Jian, W., Yong, P., Zhongwen, L., & Hua, S. (2023). Detection and tracking of floating objects based on spatial-temporal information fusion. *Expert Systems with Applications*, *225*, 120185.

[28] Chen, R., Wu, J., Peng, Y., Li, Z., & Shang, H. (2023). Solving floating pollution with deep learning: A novel SSD for floating objects based on continual

unsupervised domain adaptation. *Engineering Applications of Artificial Intelligence*, *120*, 105857.

[29] Li, N., Wang, M., Yang, G., Li, B., Yuan, B., & Xu, S. (2024). DENS-YOLOv6: A small object detection model for garbage detection on water surface. *Multimedia Tools and Applications*, *83*(18), 55751-55771.

[30] Wang, H., Cheng, H., & Zhang, J. (2024). Faster-PGYOLO: an efficient framework for floating debris detection in inland waters. *The Visual Computer*, 1-18.

[31] Zhang, D., Wang, P., Dong, Y., Li, L., & Li, X. (2024). Joint fuzzy background and adaptive foreground model for moving target detection. *Frontiers of Computer Science*, *18*(2), 182306.

[32] Li, N., Zhang, T., Li, B., Yuan, B., & Xu, S. (2023). RS-UNet: lightweight network with reflection suppression for floating objects segmentation. *Signal, Image and Video Processing*, *17*(8), 4319-4326.

[33] Aliha, A., Liu, Y., Ma, Y., Hu, Y., Pan, Z., & Zhou, G. (2023). A spatial–temporal block-matching patch-tensor model for infrared small moving target detection in complex scenes. *Remote Sensing*, *15*(17), 4316.

[34] Zhang, X., Min, C., Luo, J., & Li, Z. (2023). YOLOv5-FF: detecting floating objects on the surface of fresh water environments. *Applied Sciences*, *13*(13), 7367.

[35] Li, H., Yang, S., Zhang, R., Yu, P., Fu, Z., Wang, X., ... & Yang, Y. (2023). Detection of floating objects on water surface using YOLOv5s in an edge computing environment. *Water*, *16*(1), 86.

[36] Zhang, L., Xie, Z., Xu, M., Zhang, Y., & Wang, G. (2023). EYOLOv3: An Efficient Real-Time Detection Model for Floating Object on River. *Applied Sciences*, *13*(4), 2303.

[37] Li, K., Wang, Y., Li, W., Shen, S., Duan, S., & Wang, L. (2024). Feature augmentation and scale penalty for tiny floating detection. *International Journal of Machine Learning and Cybernetics*, *15*(3), 853-862.

[38] Selvaraj, R., Kuthadi, V. M., Duraisamy, A., Selvaraj, B., & Pethuraj, M. S. (2023). Learning optimizer-based visual analytics method to detect targets in autonomous unmanned aerial vehicles. IEEE Intelligent Transportation Systems Magazine.

[39] Sheron, P. F., Sridhar, K. P., Baskar, S., & Shakeel, P. M. (2021). Projection-dependent input processing for 3D object recognition in human robot interaction systems. *Image and Vision Computing*, *106*, 104089.

[40] Li, C., Jiang, S., & Cao, X. (2025). Small-Target Detection Algorithm Based on STDA-YOLOv8. *Sensors*, *25*(9), 2861.

[41] Zhang, C., Yue, J., Fu, J., & Wu, S. (2025). River floating object detection with transformer model in real time. *Scientific Reports*, *15*(1), 9026.

[42] Wang, S., Jiang, H., Li, Z., Yang, J., Ma, X., Chen, J., & Tang, X. (2024). PHSI-RTDETR: A lightweight infrared small target detection algorithm based on UAV aerial photography. *Drones*, *8*(6), 240.

[43] Gao, P., & Li, Z. (2025). YOLO-S3DT: A Small Target Detection Model for UAV Images Based on YOLOv8. Computers, Materials & Continua, 82(3), 4555–4572.

[44] Ni, J., Zhu, S., Tang, G., Ke, C., & Wang, T. (2024). A Small-Object Detection Model Based on Improved YOLOv8s for UAV Image Scenarios. Remote Sensing, 16(13), 2465.

[45] Yue, M., Zhang, L., Huang, J., & Zhang, H. (2024). Lightweight and Efficient Tiny-Object Detection Based on Improved YOLOv8n for UAV Aerial Images. Drones, 8(7), 276.

[46] AFO" dataset: URL: https://www.kaggle.com/datasets/jangsienicajzkowy/afo-aerial-dataset-of-floating-objects