remote sensing

MDPI

Article

FLNet: A Deep Learning Detector for Floating-Debris Detection in Close-Range Aquatic Remote Sensing Imagery

Zhenyi Yang [1,2,3], Shizheng Xiao [1,2], Mingxin Hou [1,2,3,*], Duhao Zheng [1,2], Jing Zhang [1,2] and Haochun Liu [1,2]

1 School of Mechanical Engineering, Guangdong Ocean University, Zhanjiang 524088, China; zhyiyang@gdou.edu.cn (Z.Y.); xiaosi2777392797@stu.gdou.edu.cn (S.X.); zhengduhao2025@stu.gdou.edu.cn (D.Z.); zhangjing3@gdou.edu.cn (J.Z.); liuhaochun@gdou.edu.cn (H.L.)
2 Guangdong Engineering Technology Research Center of Ocean Equipment and Manufacturing, Zhanjiang 524088, China
3 Guangdong Provincial Key Laboratory of Intelligent Equipment for South China Sea Marine Ranching, Guangdong Ocean University, Zhanjiang 524088, China
* Correspondence: houmx@gdou.edu.cn

Highlights

What are the main findings?

- FLNet combines DPAR, AMRP, and DPS-Head to address water-surface optical interference, large variations in object scale, and irregular target boundaries.
- On the custom 12,237-image dataset, FLNet achieved 88.77% mAP@0.5, 72.94% mAP@0.5:0.95, 85.93% macro-Precision, and 85.08% macro-F1. In separately trained public-data benchmarks, it achieved an F1-score of 80.26% on FloW-Img and the highest macro-Recall and macro-F1 on MarineDebris-yolo; across three random seeds, both MarineDebris-yolo advantages over Hyper-YOLO exceeded seed-to-seed variability.

What are the implications of the main findings?

- The architecture offers an accurate and computationally efficient option for automated floating-debris monitoring and is designed to suppress false detections caused by reflections, ripples, and changing illumination.
- Results on the custom dataset and on two separately trained public benchmarks, FloW-Img and MarineDebris-yolo, support its use within the evaluated close-range USV setting.

Abstract

Floating debris threatens aquatic ecosystems, yet detecting it in close-range aquatic remote sensing imagery remains difficult because small, irregular, or partially submerged objects resemble specular highlights and ripples. This study proposes FLNet, a YOLO11-L-based detector for water-surface monitoring from unmanned surface vehicles (USVs). FLNet redesigns the backbone, neck, and head with three components: a Dynamic Polarity-Aware Robustness Block that calibrates deep activations and models similar and opposing feature relations; an Adaptive Multi-Receptive Pyramid that repeatedly fuses three adjacent scales through multi-branch depthwise convolutions; and a Detail-Preserving Shared Head that applies one boundary-sensitive prediction trunk to all scales. On a custom five-category dataset of 12,237 images, FLNet achieved 85.93% macro-Precision, 85.08% macro-F1, and 88.77% mean average Precision at an intersection-over-union threshold of 0.5 (mAP@0.5), the highest single-run values among the evaluated detectors, with 24M parameters and 83 GFLOPs. Across three random seeds, its macro-Precision and macro-F1 advantages over the strongest competitor, Hyper-YOLO, exceeded seed-to-seed variability, whereas its mAP@0.5 was statistically comparable to that of Hyper-YOLO and Hyper-YOLO retained a higher mAP@0.5:0.95. In separately trained benchmarks, FLNet achieved the highest F1-score on FloW-Img (80.26%) and the highest macro-F1 on MarineDebris-yolo (79.82%).

Check for updates

Academic Editor: Haopeng Zhang

Received: 28 July 2026
Revised: 14 October 2026
Accepted: 16 October 2026
Published: 18 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

With TensorRT half-precision (FP16) acceleration, detector-side inference reached 29.8, 111.2, and 56.4 frames per second on Jetson Xavier NX, AGX Orin, and Orin Nano Super, respectively. FLNet therefore offers a favorable accuracy–efficiency balance for close-range floating-debris detection.

Keywords: floating-debris detection; aquatic remote sensing; object detection; YOLO11; multi-scale feature fusion

1. Introduction

Plastic waste accumulates in aquatic environments faster than current waste-management systems can remove it. Global plastic production has reached 460 million metric tons per year, of which less than 10% is recycled [1], and an estimated 19–23 million metric tons enter aquatic ecosystems annually [2]. Floating waste includes plastic bottles, buoys, Styrofoam, glass, and metal objects, many of which persist for decades or centuries. Plastics also fragment into microplastics that accumulate through food chains [3], and aquatic debris affects more than 260 species and may threaten human health and food safety [4]. Reliable monitoring is therefore required to locate and remove debris from rivers and lakes.

Large-area surveys of aquatic debris commonly rely on satellites or high-altitude unmanned aerial vehicles (UAVs) [5], but their spatial resolution limits the detection of individual floating objects. Sentinel-2 (10 m/pixel) and Landsat-8 (30 m/pixel) can detect large aggregations of floating plastic in coastal and ocean waters [6] but cannot resolve individual bottles or Styrofoam boards that span only tens of centimeters. Even sub-meter commercial satellites ($\leq$0.5 m/pixel) struggle in inland waterways, where debris is sparse and frequently occluded by riparian vegetation or infrastructure shadows [7].

Object appearance further complicates detection. Transparent bottles blend into the water, metal cans produce strong reflections, and irregular Styrofoam fragments can resemble wave patterns. Sunlight reflected from the water creates saturated regions that obscure debris signatures in satellite images [8]. UAV imagery offers finer spatial resolution, but oblique views can enlarge specular highlights, and ripples and uneven illumination can produce debris-like textures [9].

Unmanned surface vehicles (USVs) provide a complementary close-range view because cameras mounted near the water surface resolve individual objects at centimeter-scale ground sampling distances. USV images, however, still contain highlights, reflection bands, ripple distortions, and vibration-induced motion [10], which shift features and reduce detection precision. A detector for this setting must therefore separate weak debris cues from strong water-surface artifacts.

The imaging modality also determines what a detector can recognize. Multispectral and hyperspectral sensors exploit the short-wave infrared absorption features of hydrocarbons to discriminate plastics from other floating materials [11,12], and Sentinel-2 benchmarks such as MARIDA support pixel-level mapping of marine debris [13]; however, these approaches depend on coarse spatial resolution or specialized sensors. Low-cost RGB cameras on bridges, riverbanks, or vessels instead record individual items at high spatial resolution and have been combined with deep learning for automated river-plastic monitoring [14]. Because RGB imagery cannot identify polymer composition, and because plastic colors vary widely and may blend with the water or vegetation, a close-range RGB detector should recognize debris mainly from shape and boundary cues rather than from a fixed color profile.

Several deep detectors have been adapted to close-range debris imagery. Yang et al. added the Convolutional Block Attention Module (CBAM) to YOLOv5 for river-debris

detection [15], PC-Net used high-resolution region-of-interest (RoI) pooling for small floating objects [16], and YOLOv8-CDW, YOLOv7-APM, and YOLOv8-SST introduced attention, adaptive fusion, small-target processing, and modified losses [17–19]. Three limitations remain. First, where attention is used, it mainly strengthens positively correlated responses and does not explicitly model evidence against a debris hypothesis, so saturated glint and ripple textures can still be amplified. Second, most of these detectors retain a feature pyramid that fuses adjacent scales pairwise with the same operation at every level, which can blur the weak boundaries of small or partially submerged objects. Third, accuracy gains are often obtained by adding modules to scale-specific necks and heads; in the present evaluation, YOLOv8-CDW, YOLOv7-APM, and YOLOv8-SST required 205, 178, and 163 giga floating-point operations (GFLOPs), respectively (Section 5.2), which limits their use on embedded USV computers.

To address these limitations, this study proposes the Floating Litter Network (FLNet), a detector for close-range water-surface imagery built on YOLO11-L [20]. FLNet modifies the backbone, neck, and detection head to reduce false positives, strengthen multi-scale representation, and limit computational cost.

The main contributions of this work are as follows:

- A Dynamic Polarity-Aware Robustness Block (DPAR) that combines Dynamic Tanh (DyT) with Polarity-Aware Linear Attention in a partial-channel topology after the Spatial Pyramid Pooling–Fast (SPPF) module. It calibrates extreme deep activations, represents both similar and opposing feature relations, and keeps a convolutional bypass for weak debris cues. Compared with a direct insertion of the PolaFormer block, it raised macro-Recall from 83.07% to 85.20% with 2.37M fewer parameters.
- An Adaptive Multi-Receptive Pyramid (AMRP) that concatenates three adjacent pyramid levels, filters them with multi-branch depthwise convolutions, and redistributes the result through two focus–diffusion cycles. In same-slot comparisons, it outperformed the default path aggregation feature pyramid network (PAFPN) of YOLO11-L, BiFPN, and HS-FPN on all four accuracy metrics.
- A Detail-Preserving Shared Head (DPS-Head) that applies one trunk of Detail-Preserving Convolutions (DPConv) with Group Normalization (GN) to all pyramid levels. Across three random seeds, it matched the accuracy of three independent DPConv towers with 1.66M fewer parameters.
- A controlled evaluation on a near-duplicate-screened, 12,237-image five-category dataset and on two public benchmarks, using validation-frozen operating thresholds, three-seed confidence intervals, and detector-side deployment tests on three Jetson platforms.

The remainder of this paper is organized as follows. Section 2 reviews related detectors, feature pyramids, and prediction heads; Section 3 describes FLNet and its three modules; Sections 4 and 5 present the experimental setup and results; Section 6 evaluates edge-device deployment; and Sections 7 and 8 provide the discussion and conclusions.

2. Related Work

2.1. Object Detection in Aquatic Remote Sensing

Close-range USV imagery supports fine-grained debris monitoring but is sensitive to changing illumination, specular reflection, and wave texture. Early methods relied on foreground–background modeling and image filtering: Jin et al. [21] combined an improved Gaussian mixture model with graph cuts to separate illumination effects from the water background, and other studies combined mean-shift filtering, color-space conversion, and thresholding to separate plastic debris from water [22]. These methods are fast but often fail under complex illumination or strong wave disturbance.

Deep detectors have largely replaced handcrafted features in aquatic monitoring. They include one-stage models such as the YOLO family [23–26] and SSD [27], two-stage models such as Faster R-CNN [28], Mask R-CNN [29], and Cascade R-CNN [30], and Transformer-based models such as DETR [31] and Deformable DETR [32]. For real-time debris monitoring, single-stage models offer a practical balance between speed and accuracy. Among debris-specific variants, Yang et al. [15] added CBAM to YOLOv5 for river-debris detection, Li et al. [16] proposed PC-Net with high-resolution RoI pooling for small floating objects, and YOLOv8-CDW [17], YOLOv7-APM [18], and YOLOv8-SST [19] explored attention, adaptive fusion, small-target processing, and modified losses.

Single-stage detectors nevertheless remain vulnerable to water-specific imaging artifacts, because strong reflections, uneven illumination, and ripples can produce debris-like patterns. Most attention modules strengthen positively correlated features but do not explicitly model evidence against a debris hypothesis, which motivates the polarity-aware feature processing in DPAR.

2.2. Multi-Scale Feature Fusion in Object Detection

Feature pyramids combine fine spatial detail with high-level semantics. The feature pyramid network (FPN) introduced a top-down pathway with lateral connections [33]. The path aggregation network (PANet) added a bottom-up path to shorten information flow between shallow and deep features [34]. The bidirectional FPN (BiFPN) used learnable cross-scale connections to weight information from different scales [35]. Other recent detectors have modified the SPPF interface between the backbone and the neck. GSPPF combines multi-kernel max and average pooling with channel-wise concatenation and GhostConv for multi-scale object detection in remote sensing images [36], whereas SPPF-LSKA applies large-kernel spatial attention in a road-defect detector [37]. These designs suggest ways to aggregate contextual features at the end of the backbone, but they were developed for other tasks and were not designed to handle water-surface interference such as sun glint and ripples. FLNet therefore retains the standard SPPF and places DPAR after it to calibrate the deepest features against such interference.

Many pyramid designs apply the same fusion operation at every scale, although shallow and deep features serve different roles: shallow features retain boundaries and textures, whereas deep features provide context for suppressing reflections and ripples. Uniform fusion can therefore blur the weak boundaries of irregular or partially submerged objects. AMRP instead concatenates three adjacent levels before filtering and redistributes the fused feature to each level, preserving the distinct information carried by each scale.

2.3. Detection Head Optimization for Irregular Targets

The detection head converts feature maps into class scores and bounding boxes. YOLOX [38] and recent YOLO models use decoupled heads that separate classification from box regression and usually assign an independent convolutional tower to each of P3, P4, and P5. This scale-specific capacity is useful but duplicates parameters.

Floating objects often have irregular shapes and blurred boundaries because only part of the object remains above water, and standard 3×3 convolutions do not explicitly encode directional boundary differences. Attention-based heads offer one alternative: Dynamic Head combines scale-, spatial-, and task-aware attention [39], and CBAM applies channel and spatial attention in sequence [40]. These approaches improve general detection but do not jointly address parameter duplication across scales and unstable Batch Normalization (BN) statistics in small batches.

Independent scale-specific towers increase model size and can overfit long-tailed debris datasets, and high-resolution training limits the batch size, making BN statistics

less reliable. A suitable head should therefore share parameters across scales, avoid batch-dependent normalization, and retain boundary information; DPS-Head is designed around these requirements.

3. Materials and Methods

3.1. Baseline and Task Formulation

Given a close-range aquatic RGB image, the detector predicts category-labeled boxes and confidence scores. YOLO11-L is adopted as an accuracy-oriented baseline: its backbone extracts hierarchical features, its neck combines P3–P5, and an anchor-free head predicts classes and boxes.

This general-purpose baseline has three limitations in the present setting. First, sun glint, foam, ripple boundaries, and debris all produce high-contrast responses, so the deepest features may confuse objects with water-surface artifacts. Second, object scale varies with distance, perspective, morphology, and submergence, and sequential pairwise fusion does not jointly model shallow detail, intermediate structure, and deep semantics. Third, independent prediction towers duplicate parameters and provide no explicit prior for weak or fragmented boundaries.

FLNet modifies each of these three stages, as shown in Figure 1. In the backbone, DPAR follows the SPPF module and combines DyT with Polarity-Aware Linear Attention in a partial-channel topology, while a parallel convolutional path preserves weak debris cues. In the neck, AMRP replaces sequential pairwise fusion with two focus–diffusion cycles: FocusFeature modules combine three adjacent pyramid levels through multi-branch depth-wise convolutions, and upward and downward diffusion returns the enriched information to each scale. In the head, DPS-Head aligns P3, P4, and P5 to a common hidden dimension and processes them with one shared DPConv–GN trunk, which reduces parameter duplication while retaining a boundary-sensitive representation.

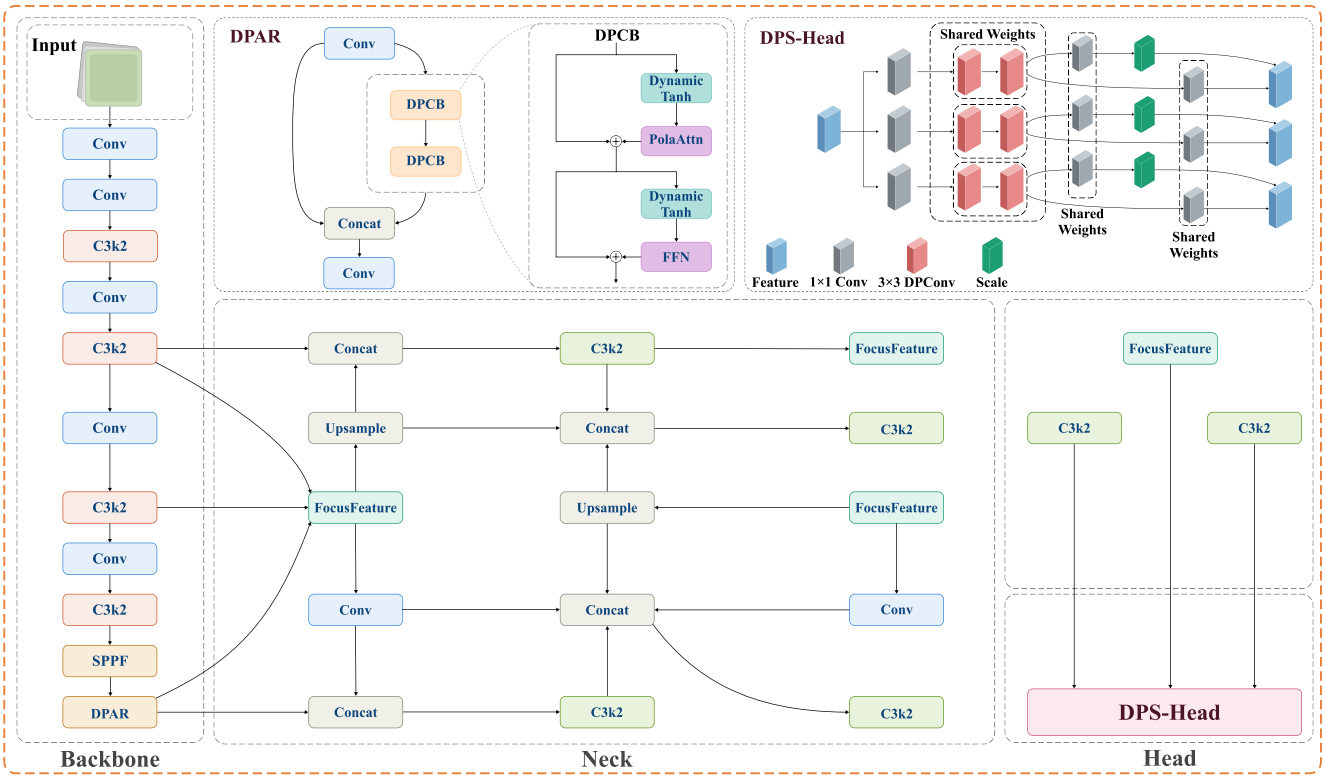

Figure 1. Overall architecture of FLNet.

FLNet builds on established operators: polarity-aware attention, Dynamic Tanh, and directional-difference convolution were proposed in earlier work. The contribution of this study lies in combining them with partial-channel processing, iterative tri-scale fusion, and cross-scale head sharing for close-range floating-debris detection.

Algorithm 1 summarizes the FLNet forward pipeline, including how the three modules connect to YOLO11-L and where training and inference diverge; Algorithm 2 gives the DPS-Head decoding procedure.

Algorithm 1 FLNet forward inference.

Require: RGB image $\mathbf{I} \in \mathbb{R}^{3\times H_0\times W_0}$ and class number n_c
Ensure: Per-scale predictions $\{\mathbf{L}_3, \mathbf{L}_4, \mathbf{L}_5\}$ during training, or decoded detections $\mathbf{D}$ at inference
$(\mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5) \leftarrow \text{Backbone}(\mathbf{I})$
$\mathbf{P}_5 \leftarrow \text{DPAR}(\text{SPPF}(\mathbf{P}_5))$ // deep-feature calibration; Section 3.2
$(\mathbf{O}_3, \mathbf{O}_4, \mathbf{O}_5) \leftarrow \text{AMRP}(\mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5)$ // two focus–diffusion cycles; Section 3.3
if training **then**
  return $\text{DPS-Head}_{\text{train}}(\mathbf{O}_3, \mathbf{O}_4, \mathbf{O}_5)$ // Algorithm 2, training branch
end if
$\mathbf{D} \leftarrow \text{DPS-Head}_{\text{infer}}(\mathbf{O}_3, \mathbf{O}_4, \mathbf{O}_5)$ // Algorithm 2, inference branch
return D

Algorithm 2 DPS-Head forward procedure.

Require: AMRP features $\{\mathbf{O}_3, \mathbf{O}_4, \mathbf{O}_5\}$, class number n_c, hidden dimension $d = 112$, and $R = \text{reg_max} = 16$
Ensure: Multi-scale training predictions or decoded detections $\mathbf{D}$
for each $i \in \{3, 4, 5\}$ **do**
  $\hat{\mathbf{O}}_i \leftarrow \text{ConvGN}_i^{1\times1}(\mathbf{O}_i)$
  $\mathbf{T}_i \leftarrow \text{SharedDPConvGN}_1(\hat{\mathbf{O}}_i)$
  $\tilde{\mathbf{O}}_i \leftarrow \text{SharedDPConvGN}_2(\mathbf{T}_i)$
  $\mathbf{B}_i \leftarrow \text{Scale}_i(\text{Conv}_{1\times1}^{4R}(\tilde{\mathbf{O}}_i))$
  $\mathbf{C}_i \leftarrow \text{Conv}_{1\times1}^{n_c}(\tilde{\mathbf{O}}_i)$
  $\mathbf{L}_i \leftarrow \text{Concat}(\mathbf{B}_i, \mathbf{C}_i)$
end for
if training **then**
  return $\{\mathbf{L}_3, \mathbf{L}_4, \mathbf{L}_5\}$
end if
for each $i \in \{3, 4, 5\}$ **do**
  $\mathbf{U}_i \leftarrow \text{Flatten}(\mathbf{L}_i)$
end for
$\mathbf{L} \leftarrow \text{Concat}(\mathbf{U}_3, \mathbf{U}_4, \mathbf{U}_5)$
if the feature shape has changed **then**
  $\mathbf{A}, \mathbf{S} \leftarrow \text{MakeAnchors}(\{\mathbf{L}_3, \mathbf{L}_4, \mathbf{L}_5\}, \text{stride}, 0.5)$
end if
$\mathbf{B}, \mathbf{C} \leftarrow \text{Split}(\mathbf{L}, [4R, n_c])$
$\mathbf{Q}_{\text{box}} \leftarrow \text{DFL}(\mathbf{B})$
$\text{Box} \leftarrow \text{dist2bbox}(\mathbf{Q}_{\text{box}}, \mathbf{A}) \odot \mathbf{S}$
$\text{Prob} \leftarrow \sigma(\mathbf{C})$
$\mathbf{D} \leftarrow \text{Concat}(\text{Box}, \text{Prob})$
return D

Before the module-level equations are presented, Table 1 summarizes the recurring notation needed to follow the three FLNet modules. Bold symbols denote tensors, whereas non-bold symbols denote scalar parameters, dimensions, or indices. Simple operators are defined at their first occurrence. Symbols used only in the forward pseudocode or a local derivation are listed separately in Appendix A, Table A1.

Table 1. Core notation of the FLNet modules; additional symbols are listed in Appendix A.

Symbol	Module	Definition
General notation		
$i \in \{3,4,5\}, \mathbf{P}_i$	General	Pyramid-level index and the corresponding backbone feature at strides 8, 16, and 32. Algorithm 1 updates the P5 entry in place after DPAR.
Dynamic Polarity-Aware Robustness Block		
$\mathbf{X}, \mathbf{X}_{\text{dyt}}$	DPAR	Input feature of a Dynamic Polarity Calibration Block (DPCB) and its DyT-calibrated representation.
$\alpha, \gamma_{\text{dyt}}, \boldsymbol{\beta}_{\text{dyt}}$	DPAR	Learnable DyT input scale, channel-wise affine scale, and channel-wise affine bias; α is initialized to 0.5.
$\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{G}$	DPAR	Query, key, value, and gate tensors produced from $\mathbf{X}_{\text{dyt}}$; H and W are feature-map dimensions and $N = H \times W$ is the token count.
$\mathbf{X}_{sim/opp}, \mathbf{Z}_{sim/opp}, \epsilon$	DPAR	Similarity/opposition representations, reciprocal kernel-normalization factors, and the small numerical-stability constant.
Adaptive Multi-Receptive Pyramid		
$\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3$	AMRP	Low-, middle-, and high-resolution input feature maps; in the first cycle they correspond to P5, P4, and P3.
$\mathbf{F}_{\text{cat}}, k_m, \mathbf{R}_m$	AMRP	Concatenated tri-scale feature, depthwise-kernel size $k_m \in \{5,7,9,11\}$, and response of the mth kernel branch.
$\mathbf{F}_{\text{agg}}, \mathbf{F}_{\text{feat}}, \mathbf{Y}$	AMRP	Aggregated multi-kernel feature, pointwise-mixed feature, and FocusFeature output.
Detail-Preserving Shared Head		
$\mathbf{O}_i, c_i, H_i, W_i, d$	DPS-Head	Incoming AMRP feature, its channel and spatial dimensions, and the unified hidden dimension ($d = 112$).
$\hat{\mathbf{O}}_i, \tilde{\mathbf{O}}_i$	DPS-Head	Features obtained after level-specific 1×1 Conv–GN alignment and the two-layer shared DPConv–GN trunk, respectively.
$\mathbf{W}_{\text{std/CD/AD/HD/VD}}, \mathbf{W}_{\text{DP}}$	DPS-Head	Standard, central-, angular-, horizontal-, and vertical-difference kernels and their structurally re-parameterized equivalent kernel; $\mathbf{b}_{\text{DP}}$ is the corresponding fused bias.
$s_i, R, n_c, \mathbf{B}_i, \mathbf{C}_i, \mathbf{L}_i$	DPS-Head	Regression scale, regression expansion factor ($R = 16$), class count, regression logits, classification logits, and concatenated per-level prediction.

3.2. Dynamic Polarity-Aware Robustness Block

Floating-debris detection requires deep features that remain discriminative under specular reflection, sun glint, and wave-induced clutter. The deepest backbone stage encodes high-level semantics at reduced spatial resolution, which makes it suitable for global dependency modeling but also vulnerable to activation dominance by saturated highlights and high-contrast water textures. Standard attention blocks mainly strengthen positive feature associations and do not explicitly separate target-consistent evidence from debris-like water artifacts.

3.2.1. Comparison with Existing Methods

DPAR integrates Dynamic Tanh (DyT) and Polarity-Aware Linear Attention (PolaAttention) into a Cross-Stage Partial (CSP) architecture. PolaAttention separates positive and negative query–key components to represent similarity and opposition relations [41], and DyT approximates the saturating response of normalization with a learnable element-wise mapping and a channel-wise affine transform, without estimating batch or token statistics [42]. DPAR keeps both operators unchanged but adapts their placement and routing. Polarity-aware modeling is applied only after SPPF, where the spatial size is manageable and semantic content is strong; separate DyT layers precede the attention and feed-forward sublayers to limit extreme responses from sun glint and saturated reflec-

tions; and PolaAttention processes only part of the channels, while a convolutional bypass preserves weak local evidence from small or partially submerged objects.

DPAR therefore differs from its source operators in three respects: its partial-channel topology, its DyT–PolaAttention ordering, and its placement at the deepest backbone stage. Together, these choices combine bounded activation calibration with polarity-aware relation modeling while preserving a convolutional path for weak debris evidence.

3.2.2. Architecture and Forward Process

As illustrated in Figure 2, DPAR follows a separation-and-fusion topology. A 1×1 convolution first splits the input feature map into two branches. The bypass branch preserves a direct convolutional path for weak or low-confidence debris cues, whereas the processed branch passes through n Dynamic Polarity Calibration Blocks (DPCBs), each containing a PolaAttention sublayer and a feed-forward sublayer with residual connections. The two branches are then concatenated and projected back to the original channel dimension.

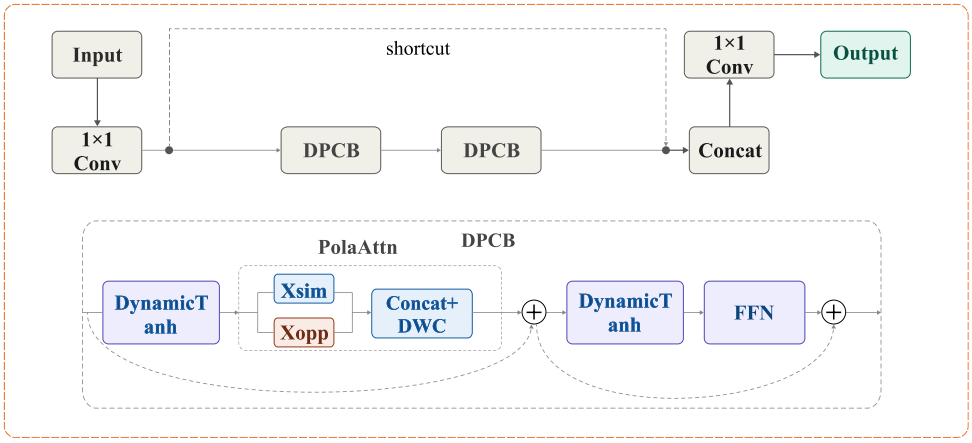

Figure 2. Architecture of the Dynamic Polarity-Aware Robustness Block (DPAR).

Within each DPCB, two independent DyT transformations precede the attention and feed-forward sublayers. Because DyT is adopted without modification, the following description explains why it suits the present task rather than presenting it as a new operator.

DyT is motivated by the empirical observation that LayerNorm in trained Transformers often produces a mapping similar to a scaled tanh curve, with a linear center and saturated tails [42]; DyT reproduces this behavior directly. For an input feature tensor $\mathbf{X}$, the operation is

$$\mathbf{X}_{\mathrm{dyt}} = \tanh(\alpha \mathbf{X}) \odot \gamma_{\mathrm{dyt}} + \beta_{\mathrm{dyt}}, \tag{1}$$

where α is a learnable scalar initialized to 0.5. The vectors γ_{dyt} and β_{dyt} provide channel-wise scale and bias, and $\odot$ denotes element-wise multiplication; α controls the width of the near-linear region.

This formulation has three useful properties at the deepest backbone stage. First, it preserves amplitude differences between positions. LayerNorm rescales every spatial position by its across-channel standard deviation, so a saturated highlight and a weak debris response can be pushed toward similar variance before attention observes their amplitude difference. DyT instead applies one shared α to the tensor: weak responses remain in the near-linear region, whereas the extreme tail produced by sun glint is compressed. Second, because α is global, the cutoff for extreme activations is learned from the overall data distribution rather than from each position. Third, DyT avoids batch statistics, which can be unreliable in the small-batch regime used for high-resolution training.

The two DyT instances do not share weights, because the attention and feed-forward sublayers receive activations at different scales; each learns its own α. DyT is applied only to the processed branch, leaving the convolutional bypass unbounded. It limits extreme activations before attention but does not guarantee that gradients cannot vanish; the ablations therefore test whether DyT and PolaAttention contribute complementarily.

The calibrated feature $\mathbf{X}_{\text{dyt}}$ is then projected into query, key, value, and gate tensors $(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{G})$. PolaAttention uses a parameterized ReLU kernel to model similar and opposing feature relations without explicitly constructing an $N \times N$ attention matrix. Its learnable power is $p = 1 + \gamma\,\sigma(\mathbf{W}_p)$, where $\gamma = 4$ controls the available power range and $\sigma(\cdot)$ is the sigmoid function. The positive and negative components of $\mathbf{Q}$ and $\mathbf{K}$ are separated as

$$\begin{cases} \mathbf{Q}_{pos} = \mathrm{ReLU}(\mathbf{Q})^p, \quad \mathbf{Q}_{neg} = \mathrm{ReLU}(-\mathbf{Q})^p \\ \mathbf{K}_{pos} = \mathrm{ReLU}(\mathbf{K})^p, \quad \mathbf{K}_{neg} = \mathrm{ReLU}(-\mathbf{K})^p \end{cases} \tag{2}$$

These components are concatenated along the channel dimension, denoted by $\|$, to construct the similarity query $\mathbf{Q}_{sim}$, the opposition query $\mathbf{Q}_{opp}$, and the polarity key $\mathbf{K}_{pola}$:

$$\begin{cases} \mathbf{Q}_{sim} = [\mathbf{Q}_{pos} \parallel \mathbf{Q}_{neg}] \\ \mathbf{Q}_{opp} = [\mathbf{Q}_{neg} \parallel \mathbf{Q}_{pos}] \\ \mathbf{K}_{pola} = [\mathbf{K}_{pos} \parallel \mathbf{K}_{neg}] \end{cases} \tag{3}$$

The value tensor $\mathbf{V}$ is split channel-wise into $\mathbf{V}_1$ and $\mathbf{V}_2$. Because the key–value product is evaluated before multiplication by the query, the computation avoids quadratic token-to-token attention and scales linearly with the number of tokens N for a fixed channel dimension. The similarity- and opposition-driven representations are

$$\begin{cases} \mathbf{X}_{sim} = \left[\mathbf{Q}_{sim}(\mathbf{K}_{pola}^T \mathbf{V}_1)\right] \odot \mathbf{Z}_{sim} \\ \mathbf{X}_{opp} = \left[\mathbf{Q}_{opp}(\mathbf{K}_{pola}^T \mathbf{V}_2)\right] \odot \mathbf{Z}_{opp}, \end{cases} \tag{4}$$

where $\mathbf{Z}_{sim}$ and $\mathbf{Z}_{opp}$ are reciprocal kernel-normalization factors computed from the corresponding query and the mean polarity key; a small constant ϵ is added to the denominator for numerical stability.

Finally, $\mathbf{X}_{sim}$ and $\mathbf{X}_{opp}$ are concatenated, supplemented with a local spatial response extracted from $\mathbf{V}$ by depthwise convolution, modulated by the learned gate $\mathbf{G}$, and projected back to the input dimension. Residual connections surround both the attention and feed-forward sublayers, and the processed branch is then fused with the bypass branch. Compared with replacing an entire backbone stage with attention, this partial-channel placement preserves a convolutional route for weak debris cues and restricts polarity-aware global modeling to the deepest feature. The mechanism is intended to reduce activation leakage from reflective water patterns; this interpretation is assessed by the component ablations and the Gradient-weighted Class Activation Mapping (Grad-CAM) analysis in Sections 5.1.3 and 5.8.

3.3. Adaptive Multi-Receptive Pyramid

Floating debris varies widely in scale and shape, even within one category: a bottle cap and a Styrofoam board require different receptive fields, and partial submergence can split one object into several visible regions. The feature pyramid must therefore retain high-resolution boundaries, intermediate structure, and low-resolution semantics. Conventional YOLO necks exchange information through sequential top-down and bottom-up paths in which each fusion node sees only two adjacent scales. This limited interaction can weaken

context, impose the same receptive-field pattern on different object shapes, and allow water texture to overwhelm small-object features during repeated resampling.

3.3.1. Comparison with Existing Methods

AMRP differs from existing pyramids in both its fusion unit and its routing pattern. BiFPN repeatedly combines aligned features with learnable scalar weights [35], and HS-FPN adds high-frequency perception and spatial-dependency modules to the lateral paths [43]. AMRP uses neither weighted pairwise summation nor explicit high-pass masking. Instead, its FocusFeature module aligns P5, P4, and P3 at the P4 resolution, concatenates them to preserve scale-specific channels, and filters the combined tensor with parallel depthwise kernels; the enriched feature then returns to both the higher- and lower-resolution branches, and a second cycle repeats this process on the updated pyramid.

These choices target floating debris whose apparent scale changes with camera distance, perspective, morphology, and partial submergence. The $\{5, 7, 9, 11\}$ depthwise kernels provide complementary local-to-broad context, and repeated diffusion gives weak high-resolution debris cues a second opportunity to interact with deep semantic evidence. These choices are evaluated through whole-neck and multi-kernel ablations and a same-slot comparison with BiFPN and HS-FPN.

3.3.2. Architecture and Forward Process

AMRP consists of two focus–diffusion cycles (Figure 1). In the first cycle, a FocusFeature module concentrates the backbone P5, P4, and P3 features into an enriched P4-resolution feature. A downsampled copy of this feature is combined with the low-resolution branch, and an upsampled copy is combined with the high-resolution branch; both are then refined by C3k2 modules. The second FocusFeature module receives the updated P5, P4, and P3 features and repeats the redistribution, so that weak high-resolution cues interact with deep semantic evidence again before reaching the detection head. Figure 3 details the FocusFeature module.

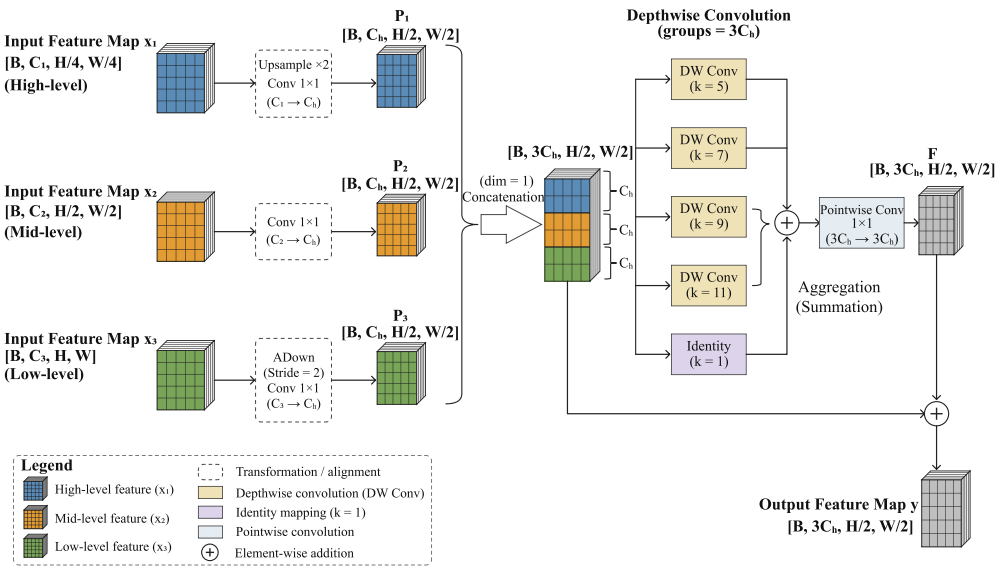

Figure 3. Architecture of the FocusFeature module in the Adaptive Multi-Receptive Pyramid (AMRP).

FocusFeature receives feature maps from three consecutive pyramid levels in low-to-high spatial-resolution order: a low-resolution semantic map $\mathbf{F}_1$, a middle-resolution map $\mathbf{F}_2$, and a high-resolution detail map $\mathbf{F}_3$. In the first cycle, these correspond to P5,

P4, and P3, respectively. All three maps are projected to a unified channel dimension $\tilde{C} = \lfloor C_2 e \rfloor$, with $e = 0.5$ by default, and aligned to the spatial resolution of $\mathbf{F}_2$:

$$\tilde{\mathbf{F}}_1 = \mathrm{Conv}_{1\times1}(\mathrm{Upsample}_{2\times}(\mathbf{F}_1)) \tag{5}$$

$$\tilde{\mathbf{F}}_2 = \mathrm{Conv}_{1\times1}(\mathbf{F}_2) \tag{6}$$

$$\tilde{\mathbf{F}}_3 = \mathrm{ADown}(\mathbf{F}_3) \tag{7}$$

where $\mathrm{ADown}(\cdot)$ denotes the adaptive downsampling operator used to map the high-resolution branch to the middle resolution. The three aligned representations are concatenated along the channel dimension to form a multi-scale feature:

$$\mathbf{F}_{\mathrm{cat}} = [\tilde{\mathbf{F}}_1 \parallel \tilde{\mathbf{F}}_2 \parallel \tilde{\mathbf{F}}_3] \tag{8}$$

Thus, each spatial position in $\mathbf{F}_{\mathrm{cat}}$ has access to fine texture, middle-level structure, and deep semantic context. To process this representation at multiple receptive-field sizes, FocusFeature applies four parallel depthwise convolution branches with kernel sizes $k_m \in \{5, 7, 9, 11\}$:

$$\mathbf{R}_m = \mathrm{DWConv}_{k_m \times k_m}(\mathbf{F}_{\mathrm{cat}}), \quad m \in \{1, 2, 3, 4\} \tag{9}$$

The branch outputs are aggregated with the original concatenated feature by element-wise summation:

$$\mathbf{F}_{\mathrm{agg}} = \mathbf{F}_{\mathrm{cat}} + \sum_{m=1}^{4} \mathbf{R}_m \tag{10}$$

The smaller kernels emphasize relatively local structure, whereas the larger kernels provide broader contextual support; these roles are complementary rather than exclusive to a particular object size. A pointwise convolution is subsequently used to mix the independently filtered channels:

$$\mathbf{F}_{\mathrm{feat}} = \mathrm{Conv}_{1\times1}(\mathbf{F}_{\mathrm{agg}}) \tag{11}$$

The context-enriched representation is fused with the original tri-scale concatenation through a residual connection, yielding

$$\mathbf{Y} = \mathbf{F}_{\mathrm{cat}} + \mathbf{F}_{\mathrm{feat}} \tag{12}$$

This residual shortcut preserves the aligned tri-scale representation and provides a direct gradient path. Through tri-scale focusing, residual multi-receptive-field filtering, and repeated upward and downward diffusion, AMRP is designed to expose the detection head to both local boundary cues and broader contextual evidence. The whole-neck and multi-kernel ablations evaluate its effect; the architecture alone does not establish that each branch specializes in a particular debris scale.

3.4. Detail-Preserving Shared Head

YOLO detection heads usually process P3/8, P4/16, and P5/32 with independent convolutional stacks, which supports scale-specific learning but duplicates parameters. These heads also commonly use BN, whose training statistics become less reliable with small batches, and their standard convolutions lack an explicit representation of the directional intensity changes that often mark fragmented, transparent, or partially submerged boundaries.

3.4.1. Comparison with Existing Methods

DPConv follows the detail-enhanced convolution design of DEA-Net [44], which computes central, angular, horizontal, and vertical difference branches alongside a standard convolutional branch. Because conventional YOLO heads assign independent towers to P3, P4, and P5, inserting DPConv into every tower would retain cross-scale parameter duplication and triple the cost of detail-sensitive processing. DPS-Head instead aligns all three pyramid levels to a common hidden dimension and applies the same two-layer DPConv–GN trunk and the same regression and classification projections at every scale; only the 1×1 input-alignment layers and the learnable scale factors for regression calibration remain level-specific. The resulting head combines batch-independent GN with a directional boundary prior shared across detection levels, avoiding three separate detail-enhanced towers.

3.4.2. Architecture and Forward Process

Figure 4 shows the DPS-Head architecture, in which GN provides batch-independent normalization, the shared trunk reuses parameters across scales, and DPConv encodes directional detail.

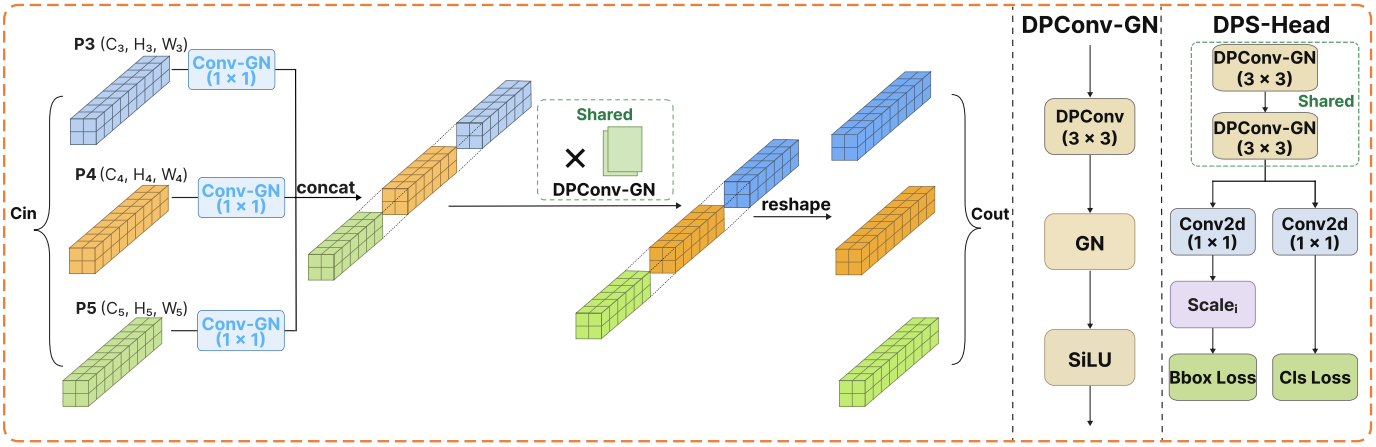

Figure 4. Architecture of the Detail-Preserving Shared Head (DPS-Head).

Each incoming AMRP feature $\mathbf{O}_i \in \mathbb{R}^{c_i \times H_i \times W_i}$, $i \in \{3, 4, 5\}$, is first projected by a dedicated 1×1 convolution with GN to a compact unified hidden dimension $d = 112$, which keeps the shared trunk lightweight:

$$\hat{\mathbf{O}}_i = \text{Conv-GN}_{1\times1}(\mathbf{O}_i) \in \mathbb{R}^{d \times H_i \times W_i}, \quad i \in \{3, 4, 5\} \tag{13}$$

Unlike BN, GN normalizes channel groups without batch-level statistics, which suits the small-batch regime of high-resolution detector training. After channel unification, all scales are processed by the same two-layer DPConv–GN trunk:

$$\tilde{\mathbf{O}}_i = \text{DPConv-GN}\big(\text{DPConv-GN}(\hat{\mathbf{O}}_i)\big) \in \mathbb{R}^{d \times H_i \times W_i} \tag{14}$$

To address the boundary ambiguity of partially submerged debris, DPConv combines one standard branch with four learnable difference branches: central difference (CD), angular difference (AD), horizontal difference (HD), and vertical difference (VD). The equivalent inference kernel and bias are obtained by summing the branch parameters:

$$\mathbf{W}_{\text{DP}} = \mathbf{W}_{\text{std}} + \mathbf{W}_{\text{CD}} + \mathbf{W}_{\text{AD}} + \mathbf{W}_{\text{HD}} + \mathbf{W}_{\text{VD}}, \tag{15}$$

$$\mathbf{b}_{\mathrm{DP}} = \mathbf{b}_{\mathrm{std}} + \mathbf{b}_{\mathrm{CD}} + \mathbf{b}_{\mathrm{AD}} + \mathbf{b}_{\mathrm{HD}} + \mathbf{b}_{\mathrm{VD}}. \tag{16}$$

The difference branches encode local intensity transitions in complementary directions, while the standard branch retains ordinary appearance information. Through structural re-parameterization, the branches are fused into the single equivalent kernel $\mathbf{W}_{\mathrm{DP}}$ and bias $\mathbf{b}_{\mathrm{DP}}$, so the deployed layer runs as one standard 3×3 convolution rather than as five parallel branches.

Because the scales retain different spatial resolutions and object-size distributions after weight sharing, a learnable per-scale scalar s_i calibrates the regression output magnitude:

$$\mathbf{B}_i = s_i \cdot \mathrm{cv2}(\tilde{\mathbf{O}}_i) \in \mathbb{R}^{4R \times H_i \times W_i} \tag{17}$$

$$\mathbf{C}_i = \mathrm{cv3}(\tilde{\mathbf{O}}_i) \in \mathbb{R}^{n_c \times H_i \times W_i} \tag{18}$$

where $R = 16$ is the regression expansion factor and n_c is the number of categories. The shared 1×1 branches $\mathrm{cv2}(\cdot)$ and $\mathrm{cv3}(\cdot)$ predict the regression and classification outputs, respectively. Apart from the level-specific 1×1 alignment layers and s_i, all feature and prediction parameters are shared across P3, P4, and P5, so no independent towers are added.

DPS-Head has different training and inference paths; the inference path additionally performs anchor generation, distribution focal loss (DFL) decoding, and box regression. Algorithm 2 summarizes both paths using the core notation in Table 1 and the implementation-specific notation in Appendix A, where $\mathbf{O}_i$ is the AMRP input, $\mathbf{L}_i$ is the head output, and SharedDPConvGN is the shared DPConv–GN layer.

3.4.3. Computational Complexity

Each FLNet module limits its contribution to model size and computation. DPAR applies PolaAttention to only part of the channels and evaluates the key–value product first, so its cost is $O(NC_a^2)$ rather than the $O(N^2C_a)$ of standard token-to-token attention, where $N = HW$ and C_a is the channel width of the attention branch.

AMRP uses depthwise kernels $\{5, 7, 9, 11\}$. For a tensor with C channels, its per-position cost is proportional to $C \sum_m k_m^2$, which grows linearly with the channel width rather than quadratically as in a dense convolution. DPS-Head applies the same two-layer DPConv–GN trunk to all three scales, so the trunk parameters, approximately $2 \times 9d^2 \approx 0.23$M for $d = 112$ after re-parameterization, are independent of the number of pyramid levels. Structural re-parameterization reduces DPConv to one equivalent 3×3 kernel at inference. The resulting parameter counts and GFLOPs are reported in the ablation and comparison tables, and the edge-device speeds are reported in Section 6.

4. Experimental Settings

4.1. Dataset Description

The custom dataset was built from two public datasets. The Garbage Collector Boat dataset on Roboflow provided 4308 images of water-surface debris and collection scenes. The HydroFloat dataset (https://github.com/SnailBoy0212/HydroFloat-Dataset, accessed on 20 September 2025) contains 4497 images; after 497 images unrelated to the studied debris categories were removed, 4000 images were retained. The two sources therefore contributed 8308 original images. Offline augmentation of the HydroFloat training images, used to increase variation in object appearance, illumination, and viewpoint, yielded 3929 additional images and a final benchmark of 12,237 images.

The 8308 original images were first divided into training, validation, and test subsets at the scene level: images of the same scene were assigned to the same subset, so that no scene appears in more than one subset. Offline augmentation was then applied only to the

HydroFloat training images. The validation and test subsets therefore contain only original images from scenes unseen during training, and no augmented copy of a validation or test image appears in the training subset.

Near-duplicate images were screened with a 64-bit perceptual hash (pHash) followed by manual review: image pairs with a Hamming distance d_{H} of at most 10 bits were treated as candidate duplicates and inspected visually before removal. After the split, the check was repeated between subsets, with flipped and rotated variants of each image also hashed to detect augmented copies. No identical hashes ($d_{\mathrm{H}} = 0$) were found between the test and training subsets, between the validation and training subsets, or within the test subset. No validation image had a training-subset neighbor within $d_{\mathrm{H}} \leq 4$, and manual inspection confirmed that the seven test images with such a neighbor depict different scenes. Because many water-surface scenes share large, similar backgrounds, the permissive $d_{\mathrm{H}} \leq 10$ criterion flagged 630 test images as having a candidate training neighbor; manual inspection attributed these candidates mainly to background similarity rather than to repeated scenes or object instances. No pair with $d_{\mathrm{H}} \leq 4$ was found between the custom dataset and the MarineDebris-yolo or FloW-Img images.

The dataset covers five surface-debris categories: Plastic (plastic bottles), Glass (glass bottles), Metal (metal cans), Plastic Buoy, and Styrofoam. Annotation was performed in Label Studio (version 1.22.0) under a unified protocol that defined category boundaries, box-placement criteria, and the treatment of occluded, truncated, reflective, and ambiguous objects; each box was required to tightly cover the visible extent of the target. A subsequent review identified missed objects, duplicate boxes, inaccurate boundaries, and inconsistent labels. The final annotations were exported in YOLO format (Figure 5).

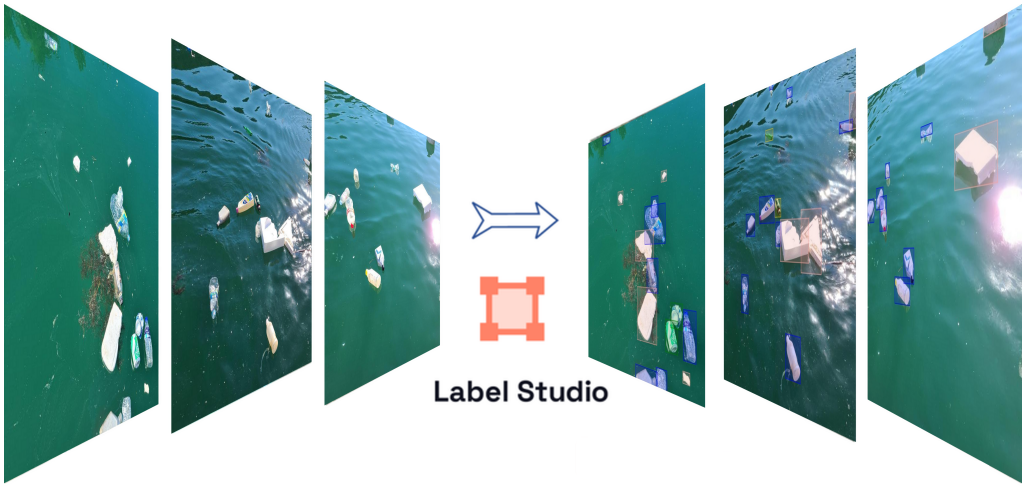

Figure 5. Manual annotation workflow for the custom floating-debris dataset using Label Studio.

The final benchmark comprised 12,237 images with 30,167 annotated objects, divided into predefined, mutually exclusive training, validation, and test subsets at an approximately 7:2:1 ratio. Because augmented images were added only to the training subset, this ratio refers to the final benchmark; the 8308 original images were distributed as 4637 training, 2447 validation, and 1224 test images. Table 2 reports the image and object counts for each split. The subsets were not reshuffled, and the held-out test subset was not used for training, checkpoint selection, or hyperparameter tuning.

Table 2. Image and object-instance counts of the custom dataset splits.

Split	Images	All Instances	Glass	Metal	Plastic	Plastic Buoy	Styrofoam
Training	8566	21,441	543	1098	11,357	6942	1501
Validation	2447	6537	177	274	3223	2469	394
Test	1224	2189	56	140	1249	602	142
Total	12,237	30,167	776	1512	15,829	10,013	2037

Figure 6 summarizes the dataset statistics. Plastic is the largest class (52.5% of all objects), followed by Plastic Buoy (33.2%), whereas Glass accounts for only 2.6% (Figure 6A). The overlaid bounding boxes (Figure 6B) and the normalized width–height distribution (Figure 6D) show that most boxes are small or medium, and Figure 6C shows the density of object centers.

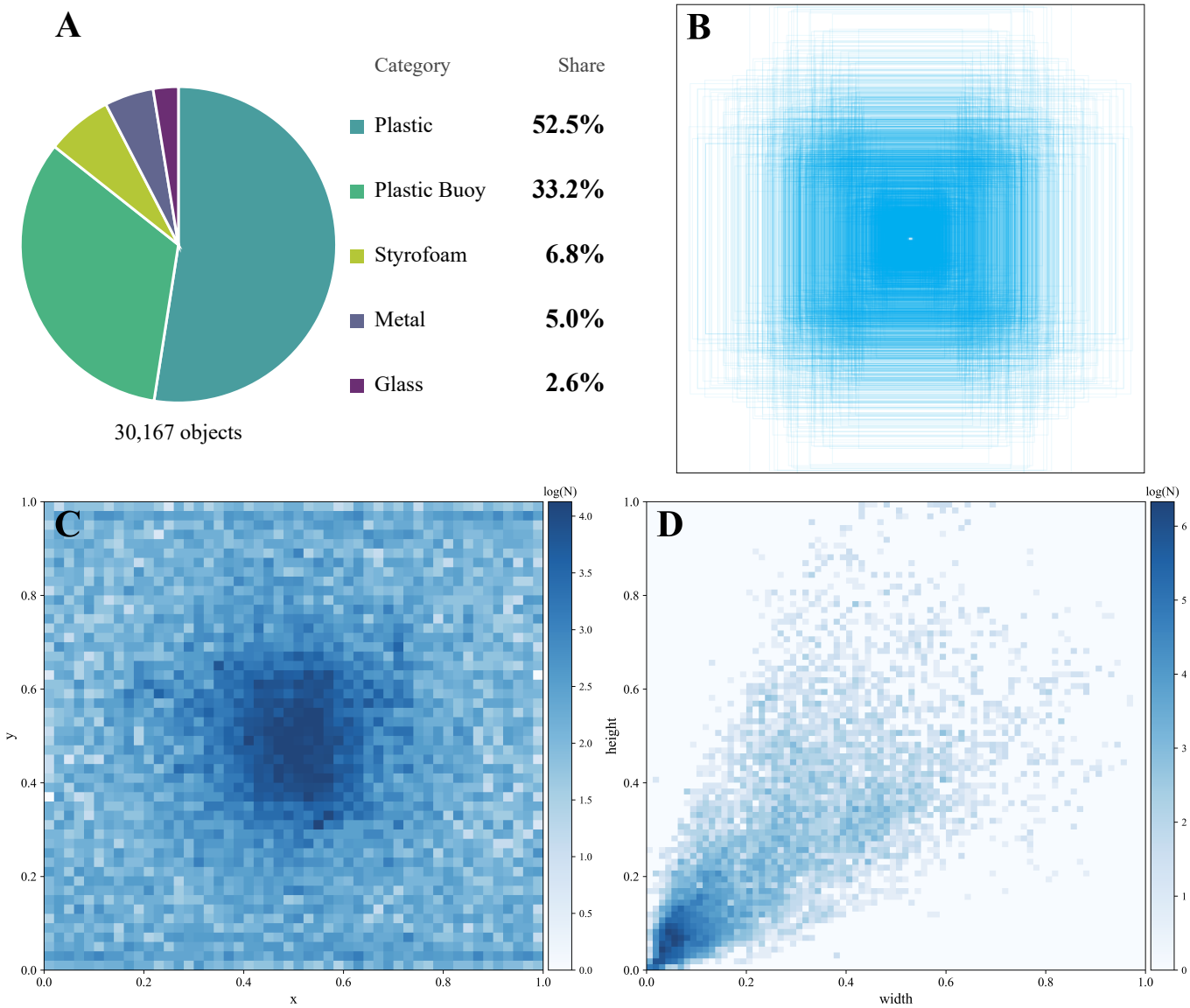

Figure 6. Statistics of the custom dataset: (**A**) category distribution of the 30,167 annotated objects; (**B**) overlay of all bounding boxes aligned at their centers; (**C**) density of object centers; (**D**) normalized bounding-box width–height distribution.

Rotation was used only in the offline augmentation of the HydroFloat training images. During training, all detectors used the same default Ultralytics online augmentation: hori-

zontal flipping with a probability of 0.5, Mosaic throughout training, random translation of up to 10% and scaling with a gain of 0.5, and hue, saturation, and value (brightness) jitter with gains of 0.015, 0.7, and 0.4, respectively. No online rotation, shear, vertical flipping, MixUp, or copy–paste augmentation was applied, and the validation and test splits were not augmented.

The public FloW-Img dataset [45], captured from a USV perspective, contains 2000 images of a single category, bottle, divided into 840 training, 360 validation, and 800 test images.

The Roboflow MarineDebris-yolo dataset contains 3775 images in five native classes (can, foam, plastic, plastic bottle, and other), divided into 2642 training, 756 validation, and 377 test images; its labels were not remapped to the custom categories. For both public datasets, all detectors were trained on the training split, and each detector's confidence threshold was selected on the validation split and frozen before evaluation on the held-out test split. The public-dataset experiments are therefore supervised within-dataset benchmarks, not evaluations of custom-dataset weights or zero-shot cross-dataset transfer tests.

4.2. Experimental Environment and Configuration

All experiments were conducted on Ubuntu 20.04 with Python 3.9, PyTorch 2.5.1, torchvision 0.20.1, and CUDA 12.4, using an Intel Core i9-14900KF CPU (Intel Corporation, Santa Clara, CA, USA), an NVIDIA RTX 4090 GPU with 24 GB of memory (NVIDIA Corporation, Santa Clara, CA, USA), and 64 GB of RAM.

All detectors were trained under the same configuration as YOLO11-L (Table 3), with a batch size of 16, at most 300 epochs, and 640 × 640 input images. Each model was built from its architecture configuration file and trained from randomly initialized weights, without COCO pre-training. The learning rate decayed linearly from η_0 to $\eta_0 \times r_f$ after a three-epoch warm-up (warm-up momentum 0.8; warm-up bias learning rate 0.1), automatic mixed precision was enabled, and early stopping was configured with a patience of 100 epochs. The checkpoint with the highest validation fitness, defined in Ultralytics as 0.1 mAP@0.5 + 0.9 mAP@0.5:0.95, was retained for evaluation; test-set annotations were not used for checkpoint selection. Hyper-YOLO, RT-DETR (RT-DETR-L configuration), YOLOv7-APM, YOLOv8-CDW, YOLOv8-SST, and the YOLO baselines were trained with their official implementations under the same input size, optimizer, schedule, epochs, batch size, and augmentation settings, without model-specific tuning.

Table 3. Training hyperparameters.

Hyperparameter	Setup
Input Size	640 × 640
Optimizer	SGD
Epochs	300
Batch Size	16
Initial Learning Rate (η_0)	0.01
Final Learning-Rate Factor (r_f)	0.01
Momentum	0.937
Weight Decay	0.0005
Learning-Rate Schedule	Linear
Warm-up Epochs	3
Early-Stopping Patience	100 epochs
Initialization	Random (no pre-training)
Mixed Precision (AMP)	Enabled
Random Seed	0 (0, 1, 2 for repeated runs)

Evaluation used Ultralytics 8.3.9. Each retained checkpoint was evaluated on the validation and test splits with a confidence threshold of 0.001, a non-maximum-suppression intersection-over-union (IoU) threshold of 0.70, and at most 300 detections per image. The value 0.001 served only as a candidate-retention floor: the retained predictions were used to construct the confidence-dependent Precision, Recall, and F1-score curves and the ranked Precision–Recall curves for average precision (AP), but not to report the operating-point metrics.

All detectors were evaluated with the same two-stage procedure. For each detector m, the operating threshold $\tau^*_{\mathrm{val},m}$ was selected on the validation split by maximizing the smoothed unweighted macro-F1 curve and was then applied unchanged to the held-out test split, at which the test-set macro-Precision, macro-Recall, and macro-F1 were computed. Test-set annotations were used neither for checkpoint selection nor for threshold calibration. AP and mAP were computed separately from the full confidence-ranked prediction list and therefore remain threshold-integrated measures.

Here, the final learning-rate factor $r_f = 0.01$ (the Ultralytics parameter lrf) is a multiplicative factor rather than an absolute final learning rate, so the scheduled terminal learning rate is $\eta_{\mathrm{target}} = \eta_0 \times r_f = 0.01 \times 0.01 = 1.0 \times 10^{-4}$.

Unless otherwise stated, each ablation and comparison result was obtained from a single deterministic training run with random seed 0. To quantify training-induced variability, the main comparisons on the custom dataset and MarineDebris-yolo and the key DPAR and DPS-Head ablations were repeated with seeds 0, 1, and 2, with all other settings unchanged (Section 5.7).

4.3. Evaluation Metrics

At a confidence threshold τ, the class-wise Precision, Recall, and F1-score for class c are defined as

$$P_c(\tau) = \frac{TP_c(\tau)}{TP_c(\tau) + FP_c(\tau)}, \tag{19}$$

$$R_c(\tau) = \frac{TP_c(\tau)}{TP_c(\tau) + FN_c(\tau)}, \tag{20}$$

$$F1_c(\tau) = \frac{2P_c(\tau)R_c(\tau)}{P_c(\tau) + R_c(\tau)}, \tag{21}$$

where $TP_c(\tau)$, $FP_c(\tau)$, and $FN_c(\tau)$ are the numbers of true positives, false positives, and false negatives of class c at threshold τ. For a dataset with C classes, the evaluation script reports unweighted macro-averages:

$$P_{\mathrm{macro}}(\tau) = \frac{1}{C}\sum_{c=1}^{C} P_c(\tau), \qquad R_{\mathrm{macro}}(\tau) = \frac{1}{C}\sum_{c=1}^{C} R_c(\tau), \tag{22}$$

$$F1_{\mathrm{macro}}(\tau) = \frac{1}{C}\sum_{c=1}^{C} F1_c(\tau). \tag{23}$$

For detector m, the operating threshold was calibrated on the validation split as

$$\tau^*_{\mathrm{val},m} = \arg\max_{\tau} \mathcal{S}_{0.1}\left[\frac{1}{C}\sum_{c=1}^{C} F1^{\mathrm{val}}_{c,m}(\tau)\right]. \tag{24}$$

Here, $\mathcal{S}_{0.1}$ denotes the Ultralytics curve-smoothing operation with a smoothing fraction of 0.1. Smoothing was used only to select the validation operating point; the reported class-wise and macro metrics were obtained from the unsmoothed curves at the frozen threshold.

Under this convention, macro-F1 is the arithmetic mean of the class-wise harmonic means and, for multi-class data, generally differs from the harmonic mean of macro-Precision and macro-Recall. For MarineDebris-yolo, $C = 5$ and the macro metrics average its five native classes; for the single-class FloW-Img dataset, F1 reduces exactly to the harmonic mean of Precision and Recall. All operating-point metrics were computed from unrounded values and rounded to two decimal places for presentation. At the class level, Precision is sensitive to false alarms from reflections and sun glint, whereas Recall measures the proportion of annotated objects that are detected, including partially submerged objects with weak visual evidence.

Mean average precision (mAP) is a threshold-integrated metric and is defined as

$$mAP = \frac{1}{C}\sum_{c=1}^{C} AP_c. \tag{25}$$

mAP@0.5 integrates each Precision–Recall curve at an IoU threshold of 0.5, whereas mAP@0.5:0.95 averages AP over IoU thresholds from 0.50 to 0.95 in steps of 0.05 and therefore places more emphasis on localization quality. Unlike the operating-point metrics, AP and mAP are computed over the full confidence-ranked prediction list retained above the 0.001 floor.

COCO-style AP@0.75 and the scale-specific AP_S, AP_M, and AP_L are also reported. Scale-specific AP is averaged over IoU thresholds from 0.50 to 0.95, with ground-truth boxes grouped by their area in the original image into small ($<32^2$ pixels), medium (32^2–96^2 pixels), and large ($\geq 96^2$ pixels) objects. These values are reported separately because COCO and Ultralytics evaluations can differ numerically.

For the multi-seed experiments, each metric is reported as the mean $\pm$ sample standard deviation (SD) over $n = 3$ training runs. The difference between two configurations, $\Delta = \bar{x}_A - \bar{x}_B$, is reported with a 95% confidence interval (CI) computed with the Welch t-interval,

$$\Delta \pm t_{0.975,\nu}\sqrt{\frac{s_A^2}{n} + \frac{s_B^2}{n}}, \qquad \nu = \frac{\left(s_A^2/n + s_B^2/n\right)^2}{\left(s_A^2/n\right)^2/(n-1) + \left(s_B^2/n\right)^2/(n-1)}, \tag{26}$$

where s_A and s_B are the corresponding SDs. A difference is described as exceeding seed-to-seed variability only when its 95% CI excludes zero; otherwise, the two configurations are described as comparable. With three seeds, these intervals are conservative and indicate the reliability of a difference rather than a formal ranking.

5. Results

5.1. Ablation Experiments

The ablation study addresses three questions: how much DPAR, AMRP, and DPS-Head each contribute relative to YOLO11-L (module-level ablation); how their internal components contribute, namely DyT and PolaAttention, the AMRP kernel set, and the normalization, sharing, and convolution choices in DPS-Head (internal ablation); and whether alternative modules perform equally well in the same position (same-slot comparison). Unless stated otherwise, all variants share the same data split, input resolution, and training configuration, and the macro metrics are unweighted averages over the five classes.

5.1.1. Module-Level Ablation

Table 4 reports the module-level results. DPAR alone raised macro-Precision from 77.21% to 80.54% and mAP@0.5 from 83.57% to 83.82% but lowered macro-Recall from 80.24% to 77.47%. AMRP and DPS-Head alone yielded larger gains, reaching macro-

Precision values of 86.05% and 85.12% and mAP@0.5 values of 87.19% and 87.39%, respectively. These changes are consistent with the intended roles of the modules but do not isolate their internal mechanisms.

Table 4. Module-level ablation on the custom test set (✓: module enabled; ×: YOLO11-L component).

Model	A (Baseline)	B	C	D	E	F	G	H (FLNet)
DPAR	×	✓	×	×	✓	✓	×	✓
AMRP	×	×	✓	×	✓	×	✓	✓
DPS-Head	×	×	×	✓	×	✓	✓	✓
macro-Precision	77.21	80.54	86.05	85.12	83.56	85.03	88.68	85.93
macro-Recall	80.24	77.47	80.38	81.49	81.57	83.15	82.03	85.20
macro-F1	76.03	77.43	82.91	82.57	82.44	83.56	84.79	85.08
mAP@0.5	83.57	83.82	87.19	87.39	87.23	86.85	87.84	88.77
mAP@0.5:0.95	70.33	70.62	73.02	72.68	72.11	72.41	73.08	72.94

The multi-module variants reveal interactions among the components. Model G, which combines AMRP and DPS-Head without DPAR, achieved the highest macro-Precision (88.68%) and mAP@0.5:0.95 (73.08%) among all variants but only a moderate macro-Recall (82.03%), suggesting that multi-scale fusion and detail-preserving prediction form a stricter decision boundary that may suppress ambiguous, low-confidence debris. The complete FLNet (Model H) achieved the highest macro-Recall (85.20%), macro-F1 (85.08%), and mAP@0.5 (88.77%). Adding DPAR to Model G raised macro-Recall by 3.17 percentage points, whereas DPAR alone lowered macro-Recall relative to the baseline. DPAR therefore mainly helps recover difficult targets and balance Precision and Recall when AMRP and DPS-Head are present, although its macro-F1 gain over Model G is modest (0.29 percentage points).

5.1.2. Fine-Grained Ablation of Internal Components

In Table 5, DyT and PolaAttention were removed separately while the rest of FLNet was kept fixed. Without DyT, macro-Precision rose to 87.56% but macro-Recall fell to 83.51%; without PolaAttention, macro-Recall was 84.47% and mAP@0.5:0.95 fell to 72.81%. Full DPAR yielded the highest macro-Recall (85.20%) and mAP@0.5 (88.77%) among these variants, whereas the w/o-DyT variant yielded the highest mAP@0.5:0.95. The benefit of DPAR therefore lies in the Precision–Recall balance and mAP@0.5 rather than in every metric. The three-seed repetition (Section 5.7) showed that the higher mAP@0.5:0.95 of the w/o-DyT variant was consistent across seeds, and the analysis in that section relates this difference to scale-specific localization behavior.

Table 5. Ablation of DPAR components. A checkmark indicates that the corresponding component or design choice is used; a cross indicates that it is not used.

Configuration	DyT	Pola-Attention	Macro-Precision	Macro-Recall	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs
w/o DyT	×	✓	87.56	83.51	88.52	73.59	24.41	83.4
w/o PolaAttention	✓	×	86.65	84.47	88.42	72.81	24.34	83.4
Full DPAR	✓	✓	85.93	85.20	88.77	72.94	24.41	83.4

Table 6 isolates the AMRP kernel set. A single 5×5 branch reached 84.17% macro-Recall, 87.47% mAP@0.5, and 72.64% mAP@0.5:0.95. Adding 7×7 and 9×9 branches raised macro-Precision to 85.66% and improved both mAP values, and the full $\{5, 7, 9, 11\}$ set performed best at the cost of an increase from 23.92M to 24.41M parameters and from 81.9 to 83.4 GFLOPs.

Table 6. Ablation of the AMRP kernel set.

Configuration	Kernel Set	Macro-Precision	Macro-Recall	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs
Single-scale kernel	{5}	84.05	84.17	87.47	72.64	23.92	81.9
Partial multi-kernel	{5,7,9}	85.66	84.85	88.42	72.85	24.18	82.7
Full multi-kernel	{5,7,9,11}	85.93	85.20	88.77	72.94	24.41	83.4

Table 7 examines normalization, DPConv, and cross-scale sharing. Replacing GN with BN reduced macro-Precision, macro-Recall, and both mAP metrics, and replacing DPConv with a standard 3×3 convolution lowered mAP@0.5:0.95 from 72.94% to 72.39%. Removing sharing raised the parameter count from 24.41M to 26.07M and slightly improved macro-Precision and mAP@0.5:0.95 but reduced macro-Recall and mAP@0.5. The full DPS-Head therefore provided the best balance between accuracy and parameter count among these variants.

Table 7. Ablation of DPS-Head components. A checkmark indicates that the corresponding component or design choice is used; a cross indicates that it is not used.

Configuration	GN	Shared Trunk	DPConv	Macro-Precision	Macro-Recall	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs
GN → BN	×	✓	✓	84.83	84.15	88.15	72.61	24.41	83.4
w/o shared trunk	✓	×	✓	86.28	84.52	88.64	73.17	26.07	83.4
DPConv → Conv	✓	✓	×	85.18	83.72	87.98	72.39	23.81	83.4
Full DPS-Head	✓	✓	✓	85.93	85.20	88.77	72.94	24.41	83.4

5.1.3. Fine-Grained Ablation Against Comparable Modules

The first two ablation levels do not show whether another module would perform equally well in the same position. In Table 8, one module was therefore replaced at a time while the remaining architecture, training schedule, augmentation, input resolution, random seed, and data splits were kept fixed. Alternatives were tested in three positions. At the post-SPPF stage, the published PolaFormer block, which retains full-channel attention and LayerNorm without a convolutional bypass, was used to separate the effect of polarity-aware attention from that of the DPAR adaptations described in Section 3.2. In the neck, BiFPN and HS-FPN replaced AMRP at P3–P5; in the head, three independent DPConv towers were used to separate the effect of weight sharing from that of Detail-Preserving Convolution. Each group also includes the corresponding YOLO11-L component.

Compared with C2PSA, the PolaFormer block improved macro-Recall from 82.03% to 83.07% and mAP@0.5 from 87.84% to 88.42% but required 2.41M more parameters and 4.95 more GFLOPs. DPAR reached 85.20% macro-Recall and 88.77% mAP@0.5 with 2.37M fewer parameters than the PolaFormer block, although its macro-Precision (85.93%) and mAP@0.5:0.95 (72.94%) were slightly lower.

The three post-SPPF configurations in Tables 5 and 8 form a controlled sequence that separates the two design choices of DPAR. The PolaFormer block and the w/o-DyT variant both omit DyT and differ mainly in channel routing, because the latter restricts attention to the processed branch and adds a convolutional bypass. This routing change left macro-Precision (87.62% vs. 87.56%) and mAP@0.5 (88.42% vs. 88.52%) nearly unchanged and raised macro-Recall slightly (from 83.07% to 83.51%) while reducing the cost by 2.37M parameters and 4.9 GFLOPs. Adding DyT before the attention and feed-forward sublayers then shifted the operating point: macro-Recall increased from 83.51% to 85.20% and mAP@0.5 from 88.52% to 88.77%, whereas macro-Precision decreased from 87.56% to 85.93%. In the three-seed repetition, this DyT step improved macro-F1 by 0.85 percentage

points (95% CI: 0.26–1.44; Section 5.7). The Recall gain of DPAR over the PolaFormer block therefore comes mainly from the DyT ordering, whereas the partial-channel routing mainly preserves accuracy at a lower cost. Overall, DPAR shifts the Precision–Recall balance at roughly the baseline cost rather than improving every metric.

Table 8. Comparison with alternative modules in the same position (proposed modules in **bold**). Italic rows indicate the module position.

Slot	Module	Macro-Precision	Macro-Recall	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs
Post-SPPF stage of the backbone							
	C2PSA (YOLO11-L default)	88.68	82.03	87.84	73.08	24.37	83.36
	PolaFormer block [41]	87.62	83.07	88.42	73.24	26.78	88.31
	DPAR	85.93	85.20	88.77	72.94	24.41	83.42
Neck							
	PAFPN (YOLO11-L default)	85.03	83.15	86.85	72.41	25.15	86.73
	BiFPN [35]	85.48	83.84	87.43	72.62	23.27	83.24
	HS-FPN [43]	84.65	84.13	87.62	72.31	24.63	83.74
	AMRP	85.93	85.20	88.77	72.94	24.41	83.42
Detection head							
	Decoupled head (YOLO11-L default)	83.56	81.57	87.23	72.11	24.97	84.37
	DPConv in three independent towers	86.28	84.52	88.64	73.17	26.07	83.44
	DPS-Head	85.93	85.20	88.77	72.94	24.41	83.42

The polarity kernel suggests a mechanism consistent with these results. PolaAttention raises the rectified query and key components to a learnable power $p \in [1, 5]$, which amplifies magnitude differences between tokens. Without DyT, the tokens entering this kernel either retain the heavy-tailed responses produced by sun glint or, after per-position normalization, lose the amplitude difference between glint and weak debris (Section 3.2.2). In the linear-attention aggregation $\mathbf{K}_{pola}^{T}\mathbf{V}$, a few saturated glint tokens can then dominate the global context shared by all positions. DyT compresses this tail with a single learned α before the power kernel while keeping weak debris responses in its near-linear region, so low-contrast and partially submerged targets retain a larger share of the aggregated context. More of these targets are then scored above the frozen threshold, which raises Recall but also admits some additional false positives, as the lower macro-Precision indicates. The bypass complements this calibration: because linear attention averages over all N tokens, a small target occupies only a small fraction of the aggregate, whereas the unbounded convolutional branch carries its local evidence to the fusion layer without this averaging. This interpretation agrees with the scale-specific ablation, in which the full model obtained the highest AP_S among the variants (Section 5.7), and with the Grad-CAM maps in Section 5.8, where FLNet concentrates activation on visible debris with less spillover onto glint and ripples (rows 1 and 3) and retains a response on the partially submerged Styrofoam board (row 2). Because these maps compare complete detectors and do not isolate DPAR, the mechanism is supported by the controlled ablations rather than demonstrated directly by the visualizations.

In the neck comparison, AMRP achieved the highest value on all four accuracy metrics. BiFPN and HS-FPN raised mAP@0.5 from 86.85% to 87.43% and 87.62%, respectively, but their macro-Recall remained below that of AMRP. AMRP was also lighter than PAFPN and HS-FPN and, compared with BiFPN, used 1.14M more parameters at nearly the same GFLOPs, because its depthwise multi-kernel branch grows only linearly with the channel width.

Both DPConv heads improved on the default decoupled head. Three independent towers reached slightly higher macro-Precision (86.28%) and mAP@0.5:0.95 (73.17%), whereas

DPS-Head reached higher macro-Recall (85.20%) and mAP@0.5 (88.77%) with 1.66M fewer parameters. Across three seeds, however, the macro-F1, mAP@0.5, and mAP@0.5:0.95 differences between the two heads remained within seed-to-seed variability (Section 5.7), so the main practical advantage of DPS-Head is the 1.66M parameter reduction at comparable accuracy.

5.2. Comparative Experiments

FLNet was compared with RT-DETR, Hyper-YOLO, several large YOLO variants, and three aquatic-debris detectors on the custom dataset and on the FloW-Img and MarineDebris-yolo benchmarks. For every dataset, each detector was trained on the training split and evaluated on the held-out test split with the same input resolution, metric definitions, and native label space, using its validation-selected and frozen confidence threshold (Table 9). This subsection reports the results on the custom dataset, and Section 5.6 reports those on the two public benchmarks. In these cross-model tables, model complexity is displayed as integers (24M parameters and 83 GFLOPs for FLNet), whereas the ablation tables retain more decimal places to resolve small component-level differences.

On the custom dataset (Table 9), FLNet achieved the highest macro-Precision (85.93%), macro-F1 (85.08%), and mAP@0.5 (88.77%), and its macro-F1 exceeded that of Hyper-YOLO by 2.18 percentage points. Its mAP@0.5:0.95 (72.94%) was lower than that of Hyper-YOLO (74.34%). The high macro-Precision is consistent with fewer false detections from highlights, sun glint, and ripple patterns, although false positives were not counted separately for each interference condition. In the three-seed analysis (Section 5.7), the macro-Precision and macro-F1 advantages over Hyper-YOLO exceeded seed-to-seed variability, whereas the mAP@0.5 difference did not; the two detectors are therefore regarded as comparable in mAP@0.5.

Table 9. Comparison of detectors on the custom test set (single run, seed 0). The best results are in **bold**.

Model	Macro-Precision	Macro-Recall	Macro-F1	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs
RT-DETR [46]	76.41	78.04	75.74	79.29	61.17	32	110
Hyper-YOLO [47]	82.10	84.96	82.90	88.64	**74.34**	52	191
YOLOv8-L	72.28	70.78	70.52	75.64	58.55	44	165
YOLOv9-C [48]	81.31	83.61	81.21	86.38	71.35	26	112
YOLOv10-L [49]	80.26	81.03	78.32	86.48	71.23	**24**	120
YOLO11-L	77.21	80.24	76.03	83.57	70.33	25	86
YOLOv12-L [50]	81.82	84.74	81.71	87.05	71.96	26	88
YOLOv13-L [51]	75.83	**87.04**	78.59	84.54	70.72	26	84
YOLO26-L [52]	82.95	81.37	80.34	84.79	70.46	**24**	86
YOLOv7-APM [18]	78.01	84.87	80.51	85.80	68.17	59	178
YOLOv8-CDW [17]	81.74	83.81	81.46	87.63	70.59	43	205
YOLOv8-SST [19]	80.25	82.34	79.47	85.01	68.64	41	163
FLNet	**85.93**	85.20	**85.08**	**88.77**	72.94	**24**	**83**

FLNet also offers a favorable accuracy–efficiency trade-off. It requires 24M parameters and 83 GFLOPs, substantially fewer than Hyper-YOLO (52M parameters and 191 GFLOPs). Compared with YOLOv10-L, which has a similar parameter count (24M) but requires 120 GFLOPs, FLNet improved macro-Precision by 5.67 percentage points and macro-F1 by 6.76 percentage points with fewer operations. These results suggest that sharing the head across scales reduces architectural redundancy while retaining sufficient representational capacity for this task.

Figure 7 compares the accuracy–efficiency profiles of YOLO11-L, YOLO26-L, Hyper-YOLO, and FLNet on five axes. Macro-Precision, macro-Recall, macro-F1, and mAP@0.5 are

divided by 100, and computational efficiency is defined as $83/G_i$, where G_i is the GFLOPs of model i and 83 GFLOPs is the FLNet reference value, so all five quantities are higher-is-better. FLNet encloses the other three profiles and has the highest single-run value on every axis in this four-model view; Hyper-YOLO approaches FLNet in macro-Recall and mAP@0.5 but has the lowest computational efficiency.

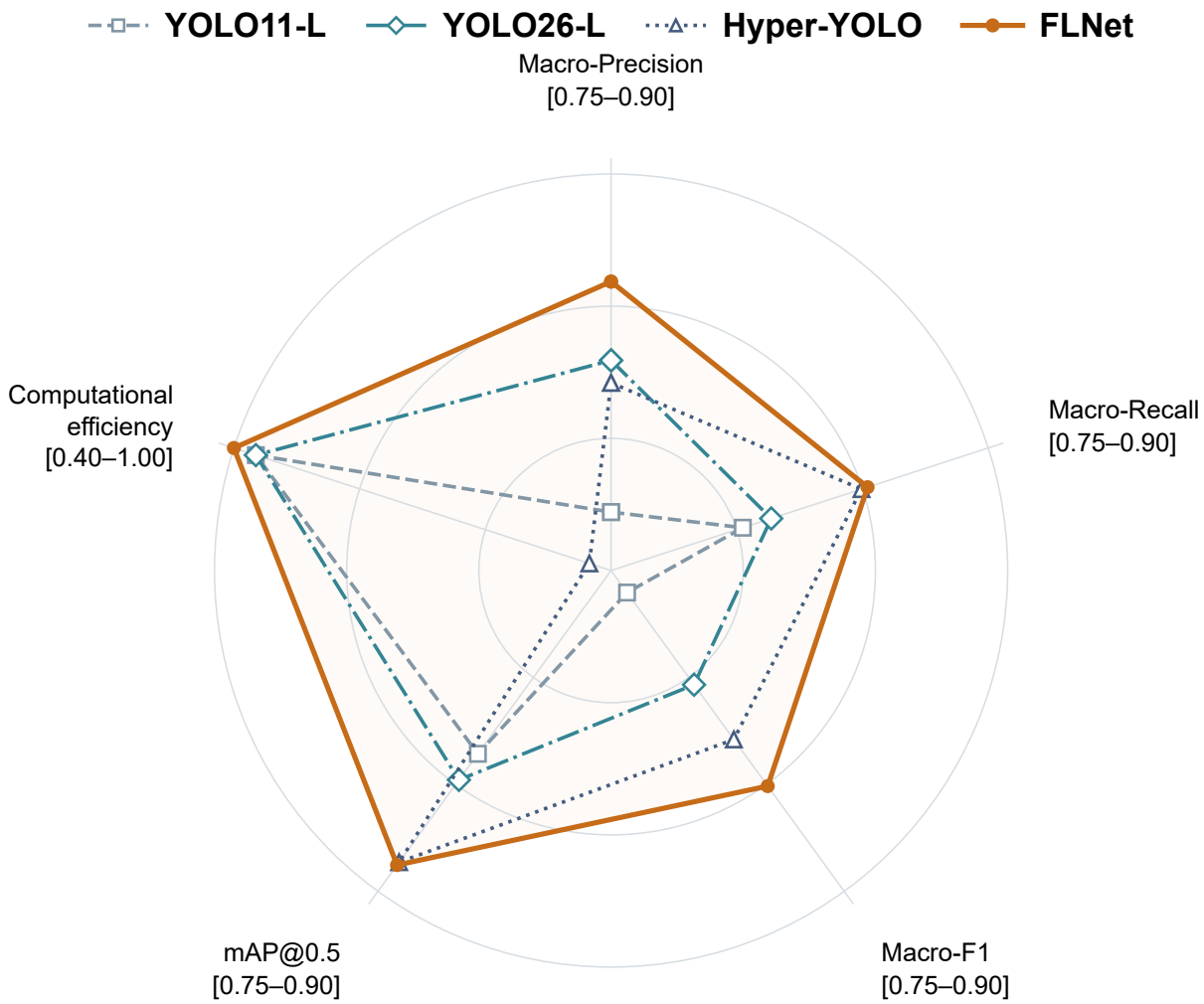

Figure 7. Radar comparison of representative detectors on the custom test set; each axis spans the range shown in brackets.

At the frozen operating points, FLNet had the second-highest macro-Recall (85.20%); only YOLOv13-L was higher (87.04%), at a substantially lower macro-Precision (75.83%). Because the confidence threshold shifts the balance between the two metrics, they should be interpreted jointly: fewer false positives can reduce unnecessary retrieval actions in automated cleanup, whereas applications that prioritize exhaustive discovery may require Recall-oriented calibration.

5.3. Confidence-Threshold Sensitivity Analysis

The results in Table 9 were obtained at detector-specific thresholds selected on the validation split. To characterize the stability of the comparison under threshold variation, FLNet, YOLO11-L, and YOLOv9-C were additionally evaluated over a range of thresholds on the custom test split: Figure 8 plots macro-Precision, macro-Recall, and macro-F1

against τ, and Table 10 reports five fixed thresholds. This post hoc sweep was not used for checkpoint selection, threshold calibration, model ranking, or selection of the reported operating points.

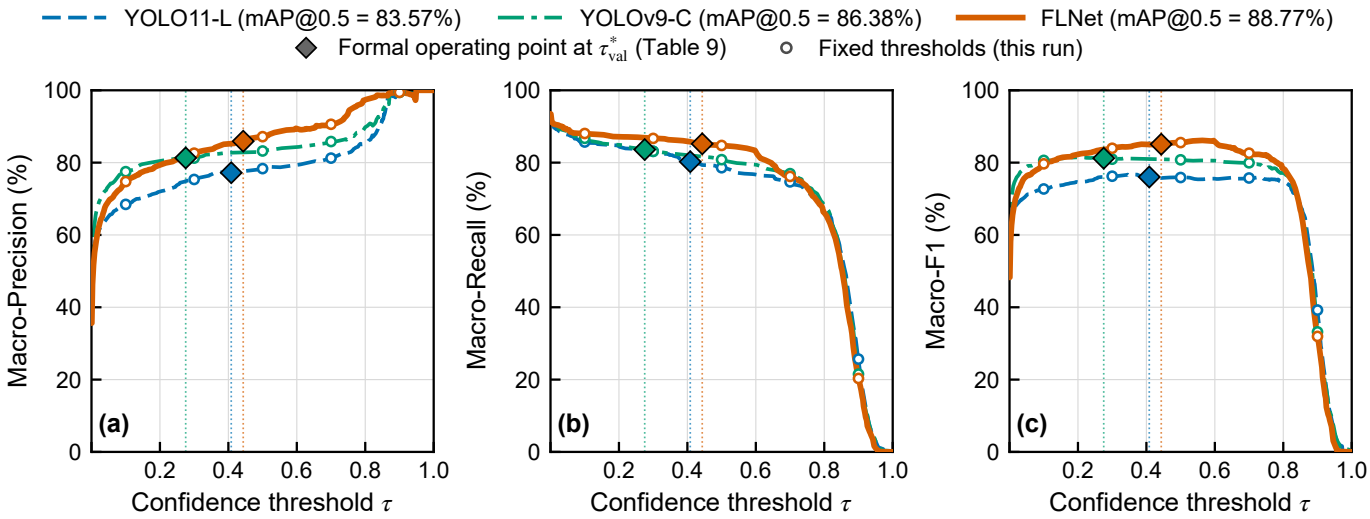

Figure 8. Post hoc confidence-threshold sensitivity on the custom test set: (**a**) macro-Precision, (**b**) macro-Recall, and (**c**) macro-F1 versus the confidence threshold τ. Open circles mark the fixed thresholds in Table 10; filled diamonds and dotted lines mark each detector's validation-selected threshold τ^*_{val}.

Table 10. Post hoc sensitivity to fixed confidence thresholds on the custom test set. The best result at each threshold is in **bold**.

Confidence τ	Model	Macro-Precision (%)	Macro-Recall (%)	Macro-F1 (%)
0.1	YOLO11-L	68.45	85.70	72.72
	YOLOv9-C	**77.50**	86.72	**80.62**
	FLNet	74.76	**88.06**	79.67
0.3	YOLO11-L	75.34	83.12	76.24
	YOLOv9-C	81.25	83.12	80.95
	FLNet	**82.62**	**86.70**	**83.97**
0.5	YOLO11-L	78.34	78.59	75.88
	YOLOv9-C	83.22	80.82	80.77
	FLNet	**87.18**	**84.75**	**85.54**
0.7	YOLO11-L	81.25	74.67	75.75
	YOLOv9-C	85.83	**76.95**	79.95
	FLNet	**90.60**	76.19	**82.61**
0.9	YOLO11-L	**99.63**	**25.67**	**39.22**
	YOLOv9-C	99.42	21.52	33.19
	FLNet	99.44	20.32	31.95

FLNet had higher macro-Precision than both baselines for $\tau \in [0.3, 0.7]$, with margins over YOLOv9-C and YOLO11-L of 3.96 and 8.84 percentage points at $\tau = 0.5$ and of 4.77 and 9.35 points at $\tau = 0.7$. YOLOv9-C was higher at $\tau = 0.1$ (77.50% vs. 74.76%), and all three detectors approached saturation at $\tau = 0.9$. The macro-Precision advantage of FLNet therefore holds across the mid-to-high threshold range rather than at a single operating point.

FLNet retained the highest macro-Recall of the three detectors for $\tau \in [0.1, 0.5]$, and all three curves decreased sharply beyond $\tau \approx 0.85$. At $\tau = 0.7$, YOLOv9-C reached a slightly higher macro-Recall (76.95% vs. 76.19%). At matched thresholds within $[0.1, 0.5]$, FLNet

is therefore not inferior to either baseline in macro-Recall. The operating-point values in Table 9 should be read at the detector-specific validation-selected thresholds, whereas the curves in Figure 8 characterize sensitivity to alternative thresholds.

The macro-F1 of FLNet remained above 82% for $\tau \in [0.3, 0.7]$, whereas neither baseline exceeded 82% in this interval, indicating lower sensitivity to threshold variation within the sampled range. This robustness observation complements, but does not replace, the validation-based threshold selection used for Table 9.

Table 10 confirms these trends: FLNet obtained the highest macro-F1 at $\tau \in \{0.3, 0.5, 0.7\}$ and remained competitive at $\tau = 0.1$, whereas at $\tau = 0.9$, all three detectors converged to near-perfect macro-Precision and low macro-Recall because most predictions were rejected. Table 9 reports formal test metrics at each detector's validation-selected threshold, whereas Table 10 reports a diagnostic sweep at five common thresholds. Although both tables were computed from the same predictions, IoU-matching rule, and postprocessing settings, their values coincide only when a diagnostic threshold equals the corresponding τ^*_{val}.

5.4. Class-Wise Performance and Class-Balancing Strategies

Because the long-tailed distribution in Figure 6A can make aggregate metrics dominated by frequent categories, YOLO11-L and FLNet were also compared class by class on the custom test split (Table 11), using the same protocol as Table 9.

Table 11. Class-wise F1-score and AP@0.5 (%) of YOLO11-L and FLNet on the custom test set.

Class	YOLO11-L F1	FLNet F1	ΔF1	YOLO11-L AP@0.5	FLNet AP@0.5	ΔAP@0.5
Glass	34.11	71.55	+37.44	56.76	78.32	+21.56
Metal	77.13	79.62	+2.48	79.60	82.00	+2.40
Plastic	87.56	88.18	+0.62	92.46	93.46	+1.00
Plastic Buoy	97.19	97.58	+0.39	98.88	98.88	0.00
Styrofoam	84.17	88.47	+4.30	90.15	91.19	+1.04

FLNet improved the F1-score for all five classes, and the unweighted class mean increased from 76.03% for YOLO11-L to 85.08%, matching the macro-F1 values in Table 9. Glass showed the largest change: its F1-score rose from 34.11% to 71.55% and its AP@0.5 from 56.76% to 78.32%, which is consistent with greater tolerance to reflections and ripples that resemble transparent debris, although it does not isolate the effect of any single module. For Metal, the F1-score increased from 77.13% to 79.62% and AP@0.5 from 79.60% to 82.00%. Plastic and Styrofoam gained 1.00 and 1.04 percentage points in AP@0.5, Styrofoam gained 4.30 points in F1-score, and Plastic Buoy retained 98.88% AP@0.5 while gaining 0.39 points in F1-score. The Glass improvement therefore did not reduce performance on the other classes. This evidence is limited to the custom test split, and external class-wise evaluation is needed before broader claims about long-tailed data can be made.

In the long-tailed training set (Figure 6A), Plastic and Plastic Buoy dominate, whereas Glass and Metal are minority classes; YOLO11-L performed worst on Glass (56.76% AP@0.5), whereas Plastic Buoy was near saturation. Two class-balancing strategies were therefore evaluated with both YOLO11-L and FLNet: inverse-frequency weighted classification loss, which increases the gradient contribution of minority classes, and minority-class oversampling, which increases the frequency of images containing Glass and Metal. All other settings, including augmentation, were held fixed (Table 12).

Table 12. Per-class AP@0.5 (%) under different class-balancing strategies.

Model	Strategy	Glass	Metal	Plastic	Plastic Buoy	Styrofoam	mAP@0.5
YOLO11-L	None	56.76	79.60	92.46	98.88	90.15	83.57
YOLO11-L	Weighted Loss	68.52	81.01	91.07	98.53	89.81	85.79
YOLO11-L	Oversampling	70.85	81.52	91.37	98.62	89.61	86.39
FLNet	None	78.32	82.00	93.46	98.88	91.19	88.77
FLNet	Weighted Loss	80.26	82.67	92.45	98.70	90.73	88.96
FLNet	Oversampling	81.57	82.94	92.62	98.81	90.58	89.30

Both strategies improved Glass detection for YOLO11-L: weighted loss raised Glass AP@0.5 from 56.76% to 68.52% and oversampling to 70.85%, increasing mAP@0.5 from 83.57% to 85.79% and 86.39%, respectively. The gains were smaller for FLNet, whose Glass AP@0.5 rose from 78.32% to 80.26% and 81.57% and whose mAP@0.5 rose from 88.77% to 88.96% and 89.30%; both strategies slightly reduced AP@0.5 for the frequent Plastic and Styrofoam classes.

Without re-balancing, FLNet already exceeded the best balanced YOLO11-L result for Glass by 7.47 percentage points. Oversampling produced the highest FLNet mAP@0.5 (89.30%) and Glass AP@0.5 (81.57%), whereas the unbalanced model retained higher AP@0.5 for Plastic and Styrofoam. The unbalanced configuration was therefore used for the main comparisons, and balancing is reported as an option for applications that prioritize minority-class performance.

5.5. Localization Accuracy and Scale-Specific Performance

To evaluate localization under a stricter overlap criterion and across object scales, Table 13 compares FLNet with YOLO11-L using COCO-style evaluation on the custom test split, which contains 262 small, 1257 medium, and 670 large instances. FLNet improved AP@0.75 from 73.94% to 79.27%, a gain of 5.33 percentage points (7.2% relative), showing that its advantage was retained under a substantially stricter localization criterion than AP@0.5. This is consistent with the intended boundary-preserving role of DPConv in DPS-Head, although the full-model comparison does not isolate that component.

Table 13. COCO-style AP@0.75 and scale-specific AP (%) on the custom test set. Better results are in **bold**.

Model	AP@0.75	AP_S	AP_M	AP_L
YOLO11-L	73.94	48.12	69.08	78.98
FLNet	**79.27**	**50.91**	**72.02**	**83.39**

FLNet also outperformed YOLO11-L in all three scale ranges, improving AP_S, AP_M, and AP_L by 2.79, 2.94, and 4.41 percentage points, respectively. The improvement was thus distributed across small, medium, and large targets, which is consistent with the design motivation of AMRP to strengthen cross-scale interaction.

5.6. Comparisons on Public Benchmarks

On FloW-Img (Table 14), whose comparison also includes the aquatic-debris detectors YOLOv7-APM, YOLOv8-CDW, and YOLOv8-SST, FLNet achieved the highest Precision (85.97%), Recall (75.26%), F1-score (80.26%), mAP@0.5 (83.59%), and mAP@0.5:0.95 (42.20%). Because FloW-Img contains only bottle annotations and every detector was trained and tested on its prescribed splits, these results constitute a supervised, dataset-specific bottle-detection benchmark rather than evidence of cross-dataset transfer.

Table 14. Comparison of detectors on the FloW-Img test set (single run, seed 0). The best results are in **bold**.

Model	Precision	Recall	F1-Score	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs
RT-DETR	73.32	69.81	71.52	74.32	33.88	32	110
Hyper-YOLO	85.47	68.99	76.35	80.83	41.42	52	191
YOLOv8-L	85.63	69.80	76.91	80.54	39.63	44	165
YOLOv9-C	85.77	70.78	77.56	81.90	39.61	26	112
YOLOv10-L	84.23	72.35	77.84	81.29	39.87	**24**	120
YOLO11-L	82.38	71.46	76.53	80.22	38.61	25	86
YOLOv12-L	83.87	68.22	75.24	80.74	39.75	26	88
YOLOv13-L	81.58	70.14	75.43	77.81	37.33	26	84
YOLO26-L	80.56	71.30	75.65	77.38	35.82	**24**	86
YOLOv7-APM	82.84	72.77	77.48	76.54	33.76	59	178
YOLOv8-CDW	82.39	73.96	77.95	77.90	35.00	43	205
YOLOv8-SST	84.09	74.07	78.76	77.07	35.57	41	163
FLNet	**85.97**	**75.26**	**80.26**	**83.59**	**42.20**	**24**	**83**

Among the three aquatic-debris baselines, YOLOv8-SST achieved the highest Precision (84.09%), Recall (74.07%), and mAP@0.5:0.95 (35.57%). Compared with YOLOv8-SST, FLNet improved the F1-score by 1.50 percentage points and mAP@0.5:0.95 by 6.63 percentage points while reducing the computational cost from 163 to 83 GFLOPs, indicating a favorable accuracy–efficiency trade-off on this benchmark.

MarineDebris-yolo provides a broader five-class label space than FloW-Img. On this benchmark (Table 15), FLNet achieved the highest macro-Precision (85.70%), macro-Recall (76.41%), macro-F1 (79.82%), and mAP@0.5:0.95 (49.76%) with 24M parameters and the lowest computational cost (83 GFLOPs), whereas its mAP@0.5 (77.74%) was lower than those of YOLOv13-L, Hyper-YOLO, and YOLO26-L. In the three-seed analysis (Section 5.7), its macro-Recall and macro-F1 advantages over Hyper-YOLO exceeded seed-to-seed variability, whereas its macro-Precision and mAP@0.5:0.95 were comparable to those of Hyper-YOLO.

Table 15. Comparison of detectors on the MarineDebris-yolo test set (single run, seed 0). The best results are in **bold**.

Model	Macro-Precision	Macro-Recall	Macro-F1	mAP@0.5	mAP@0.5:0.95	Params (M)	GFLOPs
RT-DETR	63.69	41.95	49.59	44.73	17.10	32	110
Hyper-YOLO	85.23	73.43	78.84	78.41	49.70	52	191
YOLOv8-L	84.56	72.03	77.67	77.65	47.18	44	165
YOLOv9-C	84.66	73.42	78.53	77.39	45.91	26	112
YOLOv10-L	84.02	67.87	73.74	74.65	43.71	**24**	120
YOLO11-L	80.31	71.80	75.71	75.99	44.36	25	86
YOLOv13-L	84.68	73.33	78.50	**79.57**	49.59	26	84
YOLO26-L	82.15	74.52	78.08	77.94	45.20	**24**	86
YOLOv7-APM [18]	83.09	71.34	76.62	76.27	44.99	59	178
YOLOv8-CDW [17]	83.07	73.10	77.74	77.63	45.59	43	205
YOLOv8-SST [19]	84.29	72.07	77.54	75.10	46.27	41	163
FLNet	**85.70**	**76.41**	**79.82**	77.74	**49.76**	**24**	**83**

5.7. Seed Variability and Statistical Reliability

Several single-run margins are small, notably the 0.13-percentage-point mAP@0.5 difference between FLNet and Hyper-YOLO on the custom dataset and the 0.06-percentage-point mAP@0.5:0.95 difference on MarineDebris-yolo; in addition, the w/o-DyT variant obtained a higher mAP@0.5:0.95 than the full model (Table 5). To determine whether such

differences exceed training-induced variability, the main comparisons and key ablations were repeated with seeds 0, 1, and 2: YOLO11-L, Hyper-YOLO, and FLNet on the custom dataset; Hyper-YOLO and FLNet on MarineDebris-yolo; and four ablation variants, namely w/o DPAR (the YOLO11-L C2PSA block at the post-SPPF position), w/o DyT, w/o PolaAttention, and three independent DPConv towers in place of DPS-Head. All other settings, including the data splits, training schedule, and validation-based threshold selection, were unchanged. Tables 16 and 17 report the mean ± SD and the 95% CI of each difference; the means differ slightly from the single-run values because they average three runs.

Table 16. Three-seed results (mean ± SD, %) and differences with 95% CIs for the main comparisons; macro-P and macro-R denote macro-Precision and macro-Recall. The best means are in **bold**.

Model	Macro-P	Macro-R	Macro-F1	mAP@0.5	mAP@0.5:0.95
Custom dataset					
YOLO11-L	77.43 ± 0.34	79.99 ± 0.38	76.18 ± 0.28	83.44 ± 0.21	70.45 ± 0.19
Hyper-YOLO	81.91 ± 0.28	**85.17 ± 0.32**	82.76 ± 0.24	88.57 ± 0.16	**74.45 ± 0.17**
FLNet	**86.11 ± 0.26**	85.04 ± 0.28	**85.19 ± 0.20**	**88.85 ± 0.15**	72.84 ± 0.16
Δ FLNet − YOLO11-L	+8.68 [7.97, 9.39]	+5.05 [4.27, 5.83]	+9.01 [8.43, 9.59]	+5.41 [4.98, 5.84]	+2.39 [1.99, 2.79]
Δ FLNet − Hyper-YOLO	+4.20 [3.59, 4.81]	−0.13 [−0.82, 0.56]	+2.43 [1.92, 2.94]	+0.28 [−0.07, 0.63]	−1.61 [−1.98, −1.24]
MarineDebris-yolo					
Hyper-YOLO	85.41 ± 0.30	73.15 ± 0.42	78.68 ± 0.28	**78.53 ± 0.20**	49.62 ± 0.15
FLNet	**85.55 ± 0.27**	**76.64 ± 0.35**	**79.96 ± 0.24**	77.63 ± 0.18	**49.82 ± 0.14**
Δ FLNet − Hyper-YOLO	+0.14 [−0.51, 0.79]	+3.49 [2.60, 4.38]	+1.28 [0.68, 1.88]	−0.90 [−1.33, −0.47]	+0.20 [−0.13, 0.53]

Table 17. Three-seed results (mean ± SD, %) of the key ablations and their differences from the full model with 95% CIs. The best means are in **bold**.

Variant	Macro-F1	mAP@0.5	mAP@0.5:0.95
w/o DPAR (C2PSA)	84.65 ± 0.24	87.95 ± 0.18	72.99 ± 0.16
w/o DyT	84.34 ± 0.29	88.61 ± 0.17	**73.70 ± 0.16**
w/o PolaAttention	84.77 ± 0.22	88.32 ± 0.17	72.73 ± 0.15
Three independent DPConv towers	84.96 ± 0.21	88.71 ± 0.15	73.06 ± 0.17
FLNet (full)	**85.19 ± 0.20**	**88.85 ± 0.15**	72.84 ± 0.16
Δ = *FLNet − variant [95% CI]*			
w/o DPAR (C2PSA)	+0.54 [0.03, 1.05]	+0.90 [0.52, 1.28]	−0.15 [−0.51, 0.21]
w/o DyT	+0.85 [0.26, 1.44]	+0.24 [−0.13, 0.61]	−0.86 [−1.22, −0.50]
w/o PolaAttention	+0.42 [−0.06, 0.90]	+0.53 [0.16, 0.90]	+0.11 [−0.24, 0.46]
Three independent DPConv towers	+0.23 [−0.24, 0.70]	+0.14 [−0.20, 0.48]	−0.22 [−0.59, 0.15]

On the custom dataset, FLNet exceeded Hyper-YOLO in macro-Precision by 4.20 percentage points (95% CI: 3.59–4.81) and in macro-F1 by 2.43 percentage points (95% CI: 1.92–2.94), with both intervals excluding zero. In contrast, the mAP@0.5 difference (0.28; 95% CI: −0.07 to 0.63) and the macro-Recall difference (−0.13; 95% CI: −0.82 to 0.56) included zero, so the two detectors are comparable in these metrics, and the 0.13-point single-run mAP@0.5 margin in Table 9 should not be interpreted as a reliable advantage. Hyper-YOLO retained a higher mAP@0.5:0.95, by 1.61 percentage points (95% CI: 1.24–1.98),

indicating a consistent strict-localization advantage of that larger model. Relative to YOLO11-L, FLNet improved all five metrics, with every 95% CI excluding zero.

On MarineDebris-yolo, the macro-Recall and macro-F1 advantages of FLNet over Hyper-YOLO were 3.49 percentage points (95% CI: 2.60–4.38) and 1.28 percentage points (95% CI: 0.68–1.88), respectively, whereas the macro-Precision difference (0.14; 95% CI: −0.51 to 0.79) and the mAP@0.5:0.95 difference (0.20; 95% CI: −0.13 to 0.53) included zero; the single-run mAP@0.5:0.95 margin in Table 15 is therefore within seed-to-seed variability. Hyper-YOLO obtained a higher mAP@0.5, by 0.90 percentage points (95% CI: 0.47–1.33). Across both datasets, the reproducible advantage of FLNet over Hyper-YOLO lies in the operating-point metrics, particularly macro-F1, and is achieved with less than half of the parameters and GFLOPs.

The ablation repetition supports three conclusions. First, replacing DPAR with the default C2PSA block reduced mAP@0.5 by 0.90 percentage points (95% CI: 0.52–1.28) and macro-F1 by 0.54 percentage points (95% CI: 0.03–1.05), whereas the mAP@0.5:0.95 difference was within seed-to-seed variability. Second, removing DyT reduced macro-F1 by 0.85 percentage points (95% CI: 0.26–1.44) but increased mAP@0.5:0.95 by 0.86 percentage points (95% CI: 0.50–1.22), so the higher strict-IoU mAP of the w/o-DyT variant is reproducible rather than a seed artifact; removing PolaAttention reduced mAP@0.5 by 0.53 percentage points (95% CI: 0.16–0.90), whereas its macro-F1 difference included zero. Third, the differences between DPS-Head and three independent DPConv towers were within seed-to-seed variability for all three metrics, so the shared head provides comparable accuracy with 1.66M fewer parameters.

To locate the strict-IoU difference, Table 18 evaluates the seed-0 checkpoints of the same variants with the COCO-style protocol of Table 13. Removing DyT increased AP@0.75, AP_M, and AP_L by 0.68, 0.69, and 1.06 percentage points, respectively, but decreased AP_S by 1.09 percentage points; the full model obtained the highest AP_S among all variants (50.91%), whereas the w/o-DyT variant obtained the highest AP@0.75, AP_M, and AP_L. The strict-IoU gain from removing DyT therefore came from medium and large objects, whereas DyT benefited small objects. This pattern is consistent with the intended role of DyT: its bounded mapping compresses the extreme activations produced by glint and large high-contrast targets, which can slightly reduce box refinement for large objects, while weak small-object responses remain in the near-linear range. Removing the full DPAR block and replacing DPS-Head with three independent towers produced changes in the same direction, with lower AP_S and slightly higher AP_L. DyT, and to a lesser extent the complete DPAR block and the shared head, therefore trade a small amount of strict-localization accuracy on medium and large objects for higher macro-F1, mAP@0.5, and small-object AP; among these differences, only the strict-IoU effect of DyT exceeded seed-to-seed variability. Because Table 18 is based on one seed and only 262 small instances, its scale-specific differences are indicative rather than statistically established.

Table 18. COCO-style ablation on the custom test set (seed 0, %). The best results are in **bold**.

Variant	AP@0.75	AP_S	AP_M	AP_L
w/o DPAR (C2PSA)	79.42	50.08	72.16	83.62
w/o DyT	**79.95**	49.82	**72.71**	**84.45**
w/o PolaAttention	79.12	50.35	71.88	83.22
Three independent DPConv towers	79.52	50.55	72.32	83.74
FLNet (full)	79.27	**50.91**	72.02	83.39

5.8. Experimental Visualization and Analytical Evaluation

Detection outputs and Gradient-weighted Class Activation Mapping (Grad-CAM) visualizations were compared to examine model behavior in difficult aquatic scenes.

Figure 9 compares the outputs of YOLO11-L, YOLO26-L, and FLNet on four scenes from the custom dataset. Each column corresponds to one detector and each row to one scene. Orange dashed circles mark missed objects, red dashed boxes mark false positives, and green solid boxes highlight the corresponding FLNet results.

Figure 9. Qualitative detection results of (**a**) YOLO11-L, (**b**) YOLO26-L, and (**c**) FLNet on four scenes (rows). Orange dashed circles: missed detections; red dashed boxes: false positives; green solid boxes: corresponding FLNet detections or, in row 2, the sun-glint region without a false detection.

In the first scene, YOLO11-L misses a white foam fragment and a partially visible object at the lower-right image boundary, and YOLO26-L additionally misses a small bottle and a red metal can. FLNet detects all four objects: the foam fragment as Styrofoam (0.90), the boundary object as Plastic (0.71), the small bottle as Plastic (0.87), and the can as Metal (0.84). In the second scene, strong sun glint appears next to a Styrofoam board. YOLO26-L labels the glint as Styrofoam (0.61), and YOLO11-L misses a small upright bottle. FLNet detects the bottle (Plastic, 0.93) and produces no detection in the glint region. In the third scene, debris floats near the far bank, whose reflection is visible on the water surface. YOLO11-L produces a false Styrofoam detection (0.62) in the reflection and misses the object beneath it, whereas YOLO26-L misses both this object and a bottle in the center of the image. FLNet detects the two objects as Styrofoam (0.84) and Plastic (0.89), respectively. In the fourth scene, both baselines miss a small bottle adjacent to a larger object and a partially visible object at the right image boundary. FLNet detects them as Plastic (0.83) and Styrofoam (0.58), respectively, although the latter confidence is low.

These examples are consistent with fewer missed detections of small, distant, and boundary-truncated objects and with fewer false positives caused by sun glint and reflections. As selected illustrations, however, they do not measure how often such errors occur across the test split.

Grad-CAM [53] was applied to the deepest feature map to inspect spatial activation, with red and yellow denoting high contribution and blue denoting low contribution. Figure 10 shows the original images and the maps of YOLO11-L, YOLO26-L, and FLNet.

In these examples, YOLO11-L produces diffuse activation over highlights, ripples, and sun glint in rows 1 and 3, and YOLO26-L, although more target-focused in row 3, still activates strongly on background water texture. FLNet concentrates activation more closely on visible debris: in row 2, its response remains near the partially submerged Styrofoam board with less spillover into the surrounding water, and in row 4, separate activation regions form around objects of different sizes.

5.9. Failure Case Analysis

To illustrate the remaining error types, three test-split scenes in which FLNet missed targets detected by at least one other detector were examined qualitatively: clustered minority-class objects caught in a fishing net, heavy occlusion by a nylon net, and distant low-contrast debris. The detections in Figure 11 were produced by the same trained checkpoints but rendered with display settings from an earlier qualitative analysis rather than at the operating points of Table 9. The boxes and confidence scores therefore show how each detector responds to the selected scenes but should not be used to infer Precision, Recall, or error frequency; the scenes were chosen to illustrate failure modes and are not a representative sample of the test split.

Figure 11 arranges the three scenes in a 3×3 grid, with one scene per row and the outputs of YOLOv9-C, Hyper-YOLO, and FLNet in columns (a)–(c). Green solid boxes indicate true positives (TP) recovered by a baseline under an ambiguous condition, and orange dashed circles identify the false negatives (FN) produced by FLNet at the same locations.

In the first scene, glass and plastic bottles are clustered inside a fishing net. YOLOv9-C produces a low-confidence Glass detection (0.48) for a net-entangled bottle, which both Hyper-YOLO and FLNet miss. FLNet detects the well-exposed bottles (0.88 and 0.85), but no FLNet detection is displayed for the marginal, net-entangled candidates, illustrating why the minority Glass class remains difficult under occlusion.

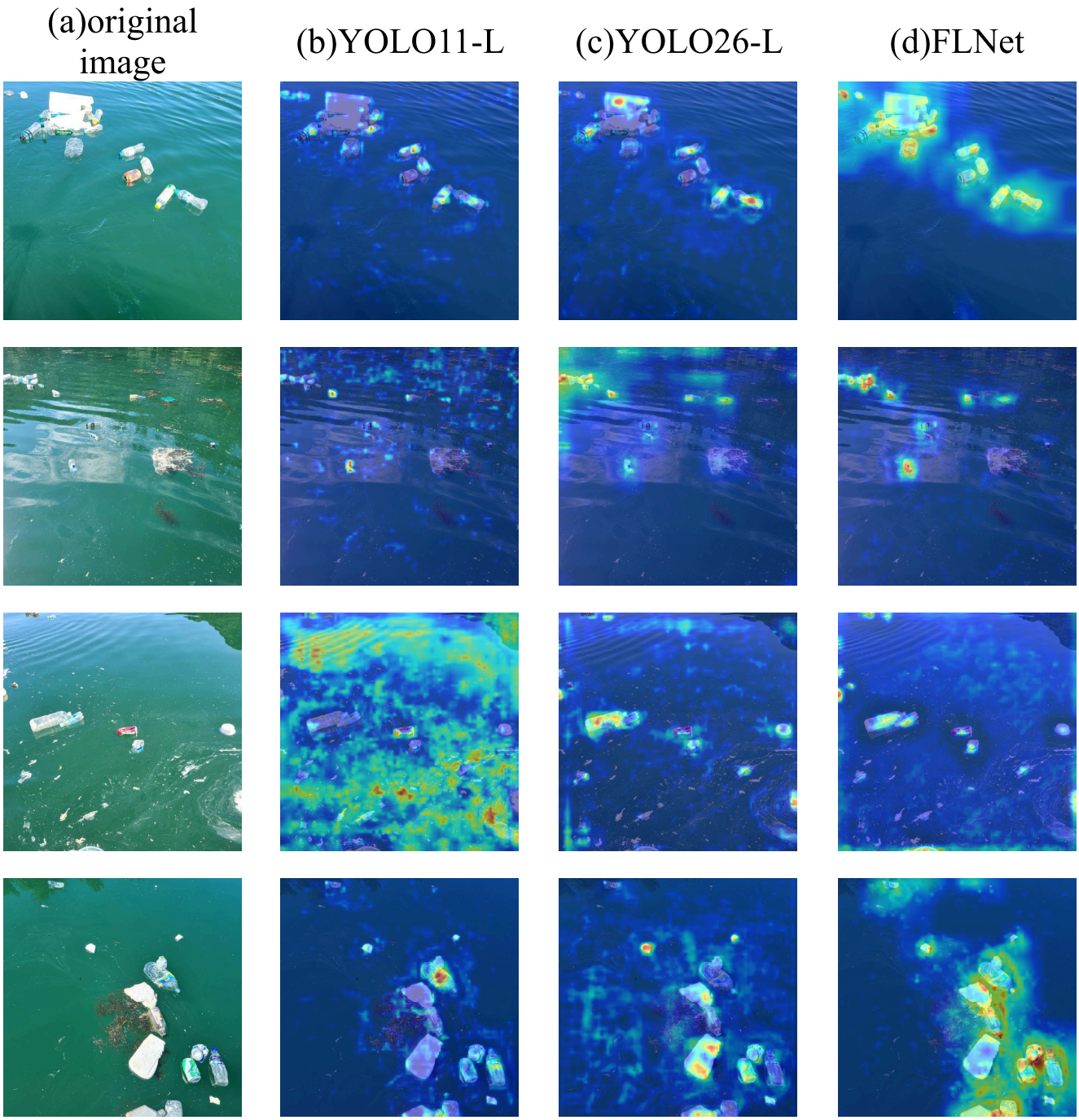

Figure 10. Grad-CAM maps for representative samples from the custom floating-debris dataset: (**a**) original image, (**b**) YOLO11-L, (**c**) YOLO26-L, and (**d**) FLNet.

In the second scene, a turquoise net occludes several bottles. YOLOv9-C and Hyper-YOLO detect one partially occluded bottle as Glass with low confidences of 0.37 and 0.32, respectively, whereas no FLNet detection is displayed at this location. Because the visible fragment resembles the surrounding mesh, suppressing this repetitive texture may reduce false positives elsewhere but also weaken the debris response.

In the third scene, transparent bottles have low contrast against the distant water surface. YOLOv9-C and Hyper-YOLO accept Plastic detections at 0.24 and 0.19, respectively, whereas no FLNet detection is displayed for this target, which is consistent with a trade-off between suppressing water-surface interference and recovering weak boundaries.

In these examples, the missed detections occurred where debris was ambiguous with respect to its background: net-entangled minority-class clusters, mesh-occluded targets, and low-contrast distant instances. Establishing how often each failure mode occurs across the test split would require a systematic, condition-stratified count of false negatives. Applications that prioritize exhaustive discovery may benefit from a lower confidence threshold, class-conditional calibration for minority categories, or temporal association across consecutive frames.

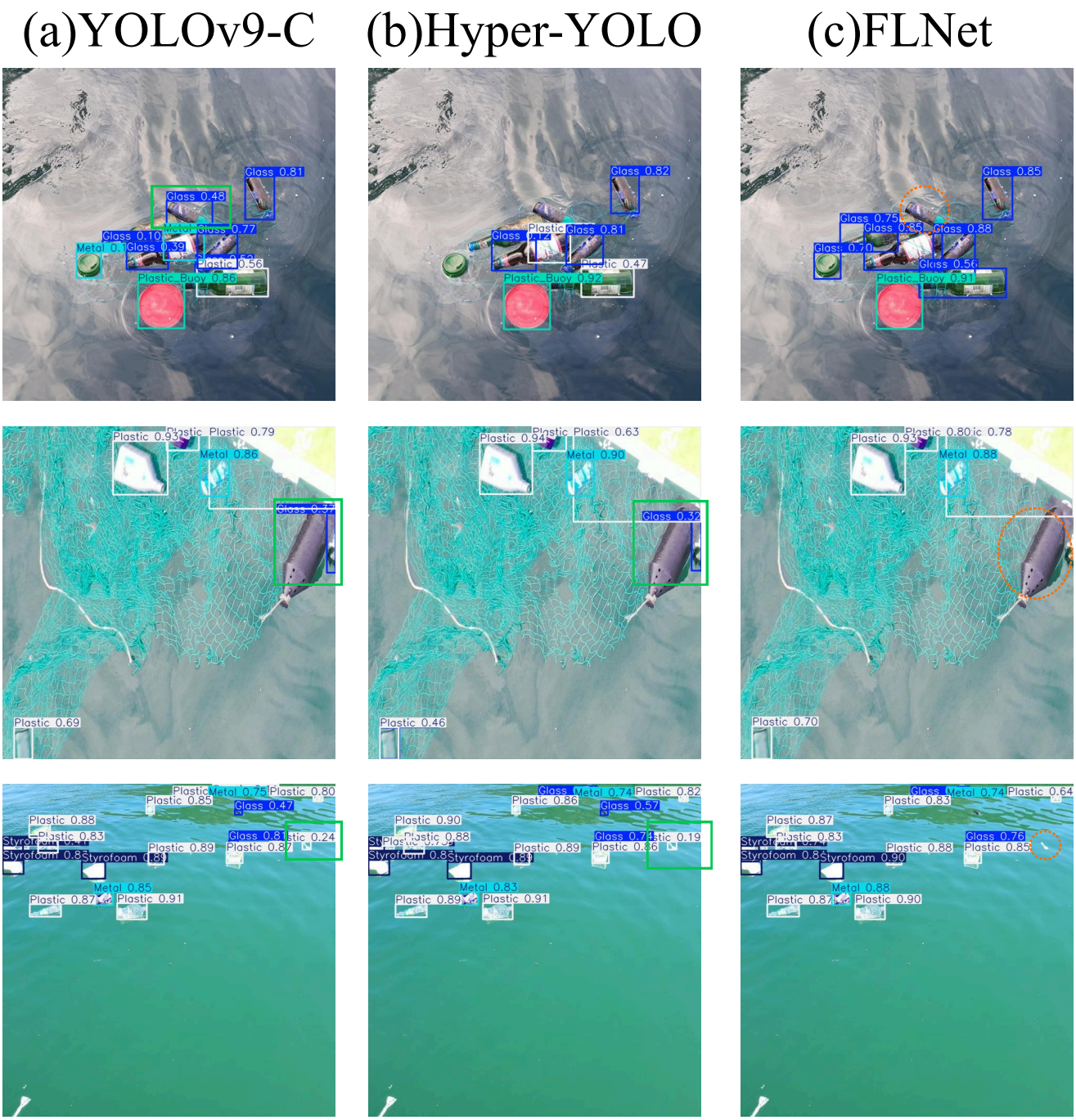

Figure 11. Failure examples of (**a**) YOLOv9-C, (**b**) Hyper-YOLO, and (**c**) FLNet on three test scenes: (1) glass bottles entangled in a fishing net, (2) occlusion by a nylon net, and (3) small, low-contrast debris at long range. Green solid boxes: true positives recovered by a baseline; orange dashed circles: the corresponding FLNet misses. The visualizations are illustrative and were not rendered at the operating points of Table 9.

6. Deployment Feasibility Analysis

FLNet and YOLO11-L were benchmarked under the same deployment protocol on Jetson Xavier NX 16GB, Jetson AGX Orin, and Jetson Orin Nano Super (all from NVIDIA Corporation, Santa Clara, CA, USA). Each device used the JetPack release and the CUDA, cuDNN, and TensorRT versions supplied with it and was set to its highest-performance power mode (Table 19). All tests used 640 × 640 input, batch size 1, and identical confidence and IoU thresholds. Inference was performed with ONNX Runtime (CUDA execution provider) in single precision (FP32) and with TensorRT in half precision (FP16); because TensorRT engines are specific to the device and TensorRT version, each FP16 engine was built on its evaluation device from the model after DPConv re-parameterization.

Table 19. Software stack and power mode of the Jetson devices.

Device	JetPack (Jetson Linux)	CUDA	cuDNN	TensorRT	ONNX Runtime-GPU	Power Mode
Xavier NX 16GB	5.1 (35.2.1)	11.4	8.6.0	8.5.2	1.15.1	MODE_20W_6CORE
AGX Orin	5.1 (35.2.1)	11.4	8.6.0	8.5.2	1.15.1	MAXN
Orin Nano Super	6.2 (36.4.3)	12.6	9.3.0	10.3.0	1.19.2	MAXN SUPER

Latency was measured from the decoded image tensor to the final detection output, including preprocessing, network inference, non-maximum suppression, and coordinate restoration; camera acquisition, video decoding, and lens undistortion were excluded so that the measurements reflect the detector rather than the full USV perception pipeline. Each model was warmed up for 200 iterations, and the GPU was synchronized before each high-resolution timestamp so that asynchronous kernel execution was included in the measured interval. Each configuration was timed over three complete passes of the 1224 test images, and the frame rate in frames per second (FPS) was calculated as $1000/\bar{t}$, where $\bar{t}$ is the mean per-image latency in milliseconds.

TensorRT FP16 increased the throughput of FLNet on every device, from 16.5 to 29.8 FPS on Xavier NX, from 62.7 to 111.2 FPS on AGX Orin, and from 30.9 to 56.4 FPS on Orin Nano Super (Table 20). FLNet was also faster than YOLO11-L on all three devices and with both backends: with TensorRT FP16, the frame rate increased from 26.8 to 29.8 FPS on Xavier NX, from 99.6 to 111.2 FPS on AGX Orin, and from 51.3 to 56.4 FPS on Orin Nano Super, while FLNet improved mAP@0.5 by 5.20 and macro-F1 by 9.05 percentage points on the custom dataset.

Table 20. Detector-side inference speed on Jetson devices (ONNX: ONNX Runtime, FP32; TRT: TensorRT, FP16).

Model	Xavier NX ONNX	Xavier NX TRT	AGX Orin ONNX	AGX Orin TRT	Orin Nano ONNX	Orin Nano TRT
YOLO11-L	14.9 FPS	26.8 FPS	56.9 FPS	99.6 FPS	28.1 FPS	51.3 FPS
FLNet	16.5 FPS	29.8 FPS	62.7 FPS	111.2 FPS	30.9 FPS	56.4 FPS

Table 21 shows that the TensorRT FP16 latency was stable: on every device, the 95th-percentile latency exceeded the mean by no more than 8%, and the SD was below 5% of the mean, indicating no pronounced long tail under the tested protocol. These measurements exclude camera acquisition, rectification, tracking, communication, navigation, and actuator control.

Table 21. TensorRT FP16 latency of FLNet on Jetson devices.

Device	Mean ± SD (ms)	P50 (ms)	P95 (ms)
Jetson Xavier NX 16GB	33.6 ± 1.2	33.4	35.7
Jetson AGX Orin	9.0 ± 0.4	8.9	9.7
Jetson Orin Nano Super	17.7 ± 0.7	17.5	18.9

Table 22 compares the training-time PyTorch FP32 model with the deployed TensorRT FP16 engine, which combines DPConv re-parameterization and half-precision conversion and was evaluated on AGX Orin at the same frozen threshold ($\tau = 0.443$). The conversion reduced mAP@0.5, mAP@0.5:0.95, and macro-F1 by 0.11, 0.16, and 0.13 percentage points, respectively, which is within the seed-to-seed SD of FLNet (0.15, 0.16, and 0.20; Table 16). The decreases were larger for the most sensitive targets, with AP_S decreasing by 0.45 points and Glass AP@0.5 by 0.38 points, indicating that half precision slightly affects small and transparent debris.

Table 22. Accuracy (%) of FLNet before and after TensorRT FP16 conversion on the custom test set.

Model Form	mAP@0.5	mAP@0.5:0.95	Macro-F1	AP_S	Glass AP@0.5
PyTorch FP32 (training-time structure)	88.77	72.94	85.08	50.91	78.32
TensorRT FP16 (re-parameterized, AGX Orin)	88.66	72.78	84.95	50.46	77.94
Δ (FP16 − FP32)	−0.11	−0.16	−0.13	−0.45	−0.38

AGX Orin provided the highest throughput and therefore the largest margin for other onboard tasks. For single-stream inference, Xavier NX reached 29.8 FPS, close to the common 30 FPS video rate, but offered less computational headroom, and Orin Nano Super provided intermediate throughput on a smaller platform. Resource sharing with calibration, tracking, Robot Operating System 2 (ROS2) middleware, navigation, and control was not measured.

The design of FLNet is well suited to TensorRT: DPAR uses linear polarity-aware attention, AMRP uses depthwise convolutions, DPS-Head shares its prediction trunk across scales, and DPConv is re-parameterized as a standard convolution before export, so its training-time branches add no deployment cost. Under the tested protocol, these choices made FLNet faster than YOLO11-L.

Several field-level factors must still be addressed before practical USV deployment. Camera distortion and mounting angle can change the apparent geometry and scale of floating debris; platform vibration, wave-induced motion, and insufficient shutter speed can introduce motion blur and weaken target boundaries; and strong sun glint, reflection bands, dynamic ripples, shadows, and partial submergence can reduce detection confidence. A practical deployment should therefore include intrinsic camera calibration, lens undistortion, vibration isolation, short-exposure acquisition, and continuous monitoring of device temperature and power consumption. TensorRT FP16 was adopted as the primary deployment optimization because it increased throughput by approximately 80% on every device while changing overall accuracy by less than 0.2 percentage points. INT8 quantization and structured pruning were not included because they require additional calibration and accuracy validation, particularly for small, transparent, or partially submerged debris; future work will examine them together with long-duration thermal-stability tests and field validation under diverse illumination, water-surface, and vessel-motion conditions.

7. Discussion

In the evaluated close-range setting, FLNet shifted the operating point toward higher macro-Precision while achieving the highest macro-F1. On the custom dataset, it achieved

the highest single-run macro-Precision, macro-F1, and mAP@0.5 with 24M parameters and 83 GFLOPs; across three random seeds, its macro-Precision and macro-F1 advantages over Hyper-YOLO exceeded seed-to-seed variability, whereas its mAP@0.5 was statistically comparable to that of Hyper-YOLO and Hyper-YOLO retained a higher mAP@0.5:0.95. The largest class-wise gain occurred for Glass, suggesting that the aggregate improvement was not driven only by frequent classes. The FloW-Img and MarineDebris-yolo results provide additional evidence from supervised, dataset-specific training and held-out testing in their native label spaces; because the models were retrained on each dataset, these experiments do not establish zero-shot cross-dataset transfer or generalization across all debris types, water bodies, seasons, or platform states.

The ablations suggest complementary roles for the three modules: DPAR shifts the operating point toward higher macro-Recall when AMRP and DPS-Head are present, AMRP improves information exchange across scales, and DPS-Head reduces parameter duplication while retaining directional boundary information. These interpretations are supported by component ablations but are not direct measurements of internal causal mechanisms.

The performance profile also reveals a Precision–Recall trade-off, which Section 5.9 illustrates with selected failure cases. FLNet achieved higher macro-Precision and macro-F1 than the compared models, and its macro-Recall was second only to that of YOLOv13-L, which attained higher Recall at a much lower macro-Precision. This pattern is consistent with a stricter decision boundary that suppresses false positives caused by water-surface interference but may miss some low-contrast, severely submerged, or boundary-blurred debris. For USV-based retrieval, this trade-off may be preferable when false positives would trigger unnecessary navigation and collection actions, although this operational benefit was not quantified, for example by comparing detectors at matched Recall or at a fixed false-positive rate. Applications that prioritize exhaustive debris discovery may instead require Recall-oriented tuning, a lower confidence threshold, or complementary temporal tracking.

Limitations and Future Work

Several limitations should be considered when interpreting these results. First, although the custom dataset contains 12,237 images (8308 original images) in five debris categories, it does not fully cover extreme weather, nighttime illumination, severe occlusion, dense debris accumulation, or strong motion-induced degradation, and the public source datasets do not provide complete metadata on season, weather, water-body type, camera pose, or vessel motion. These gaps may affect reliability in more diverse field environments. Second, FloW-Img contains only the bottle category and cannot establish multi-class robustness. Third, FloW-Img and MarineDebris-yolo were evaluated after dataset-specific supervised training in their native label spaces and therefore provide within-dataset benchmarks rather than zero-shot evidence of cross-dataset transfer. Fourth, the MarineDebris-yolo results are aggregate macro metrics and do not resolve class-specific uncertainty for minority categories. Finally, seed repetition was limited to three runs of the baseline, the strongest competitor, and the key ablations; the remaining baselines, the COCO-style ablation, and the class-wise results come from single runs, so small differences in those tables should be interpreted with caution.

The robustness of FLNet to water-surface interference is supported by aggregate metrics, component ablations, and selected qualitative examples rather than by a condition-stratified evaluation: false positives and missed detections were not counted separately for glare, low contrast, partial submergence, or occlusion, and debris-free water images were not evaluated separately. Statements about these conditions should therefore be read as design motivation and illustrative observations rather than as measured robustness.

In addition, although the custom dataset was divided at the scene level and screened for near-duplicates, the split was not organized by image source, and no source-disjoint evaluation, in which a detector is trained on one source and tested on the other, was performed; generalization to unseen image sources therefore remains to be verified.

FLNet relies only on RGB appearance. Its task is object-level detection of debris categories defined by visible form, such as bottles, cans, buoys, and foam boards; it does not identify the polymer composition of an object, which would require spectral information [11,12]. Material identification is therefore outside the scope of the detector, whereas a missed object is a false negative that the detector must control. Multicolored debris and debris whose color blends with the surrounding water or vegetation remain a likely source of such false negatives. Three aspects of the design and protocol mitigate this risk. First, training applies strong saturation and brightness jitter (gains of 0.7 and 0.4), exposing the detector to large variations in color intensity; the hue jitter, however, is small (gain 0.015), so arbitrary hue changes are not simulated. Second, the difference branches of DPConv in DPS-Head emphasize directional feature differences, which favor boundary information over absolute appearance. Third, Recall can be adjusted to the application: in the post hoc sensitivity analysis, lowering the confidence threshold to 0.1 increased FLNet's macro-Recall on the custom test split to 88.06%, at the cost of a lower macro-Precision of 74.76% (Table 10), and because the same object usually appears in several consecutive USV frames, multi-frame association could recover detections missed in individual frames, although this was not evaluated here. The class-wise results provide indirect support: the F1-score for transparent Glass, whose apparent color depends on the background, increased from 34.11% to 71.55% relative to YOLO11-L, although the test split contains only 56 Glass instances. Nevertheless, false negatives were not counted separately for multicolored or background-blended objects, and robustness to color variation was not tested with controlled color perturbations; such evaluations, together with multispectral, near-infrared, or polarization imaging for material-level discrimination, are left for future work.

Although MarineDebris-yolo broadens the evaluation beyond the single-category FloW-Img dataset, the experiments remain limited to close-range water-surface imagery. Trash-ICRA19 contains underwater imagery collected for robotic marine-litter detection [54] and differs from the present task in imaging medium, viewpoint, light attenuation, turbidity, background composition, and label space. Evaluating FLNet on it would therefore be a cross-domain transfer study rather than a comparable water-surface benchmark; future work will evaluate FLNet on Trash-ICRA19 and related underwater datasets under a dedicated protocol that defines the label mapping and transfer conditions.

The deployment results also have a clear boundary: the Jetson experiments evaluate detector-side inference speed under ONNX FP32 and TensorRT FP16 but do not constitute a complete onboard USV validation. A field deployment additionally involves camera acquisition, image rectification, motion blur, lens distortion, vibration, water spray, communication latency, navigation decision-making, and actuator control, all of which may change detection accuracy and system-level latency. Future work should therefore include integrated USV experiments under real vessel motion, different camera mounting angles, long-duration thermal conditions, and diverse water-surface states.

Finally, DPAR and AMRP are modular and may transfer to lightweight backbones such as MobileNet or EfficientNet, but this has not been verified experimentally. Future work will evaluate these modules with lightweight backbones, together with INT8 quantization, structured pruning, temporal tracking, and additional class-balanced training strategies for minority debris categories.

8. Conclusions

This study presented FLNet, a YOLO11-L-based detector for floating debris in close-range aquatic RGB imagery. FLNet redesigns three stages of the detection pipeline: DPAR calibrates deep activations and models similar and opposing feature relations after SPPF, AMRP repeatedly fuses three adjacent pyramid levels through multi-branch depthwise convolutions, and DPS-Head applies one boundary-sensitive DPConv–GN trunk to all scales. These components were designed to address water-surface interference, large variations in object scale, and weak or irregular target boundaries.

The experiments support four conclusions. First, on the custom five-category dataset, FLNet achieved the highest single-run macro-Precision (85.93%), macro-F1 (85.08%), and mAP@0.5 (88.77%) among the evaluated detectors with 24M parameters and 83 GFLOPs. Across three random seeds, its macro-Precision and macro-F1 advantages over Hyper-YOLO exceeded seed-to-seed variability, the two detectors were statistically comparable in mAP@0.5, and Hyper-YOLO, which requires 52M parameters and 191 GFLOPs, retained a higher mAP@0.5:0.95. Second, the ablations showed that each module contributed to this result: adding DPAR to the other two modules raised macro-Recall by 3.17 percentage points, AMRP outperformed PAFPN, BiFPN, and HS-FPN in the same position, and DPS-Head matched the accuracy of three independent DPConv towers with 1.66M fewer parameters. Third, relative to YOLO11-L, FLNet improved the F1-score for all five categories, most strongly for Glass (from 34.11% to 71.55%), and improved AP for small, medium, and large objects. In separately trained benchmarks, it also obtained the highest F1-score on FloW-Img (80.26%) and the highest macro-F1 on MarineDebris-yolo (79.82%). Fourth, with TensorRT FP16, detector-side inference reached 29.8–111.2 FPS on three Jetson platforms, and the conversion changed the overall accuracy metrics by at most 0.16 percentage points. Taken together, these results indicate that FLNet offers a favorable accuracy–efficiency balance for close-range floating-debris detection within the evaluated setting.

Several aspects were not established by the present experiments. At its validation-selected operating point, FLNet favors Precision, so applications that prioritize exhaustive debris discovery may require a lower confidence threshold. Robustness to glare, low contrast, partial submergence, and occlusion is supported by aggregate metrics, ablations, and selected examples rather than by a condition-stratified evaluation. Because FLNet relies on RGB appearance, it cannot identify polymer composition, and missed detections of multicolored or background-blended debris were not counted separately. The custom dataset was split by scene but not by source, and the public-dataset results reflect supervised within-dataset training rather than zero-shot transfer. Future work will therefore evaluate FLNet under condition-stratified and source-disjoint protocols, test its robustness to color variation, and validate it in integrated USV operation under adverse weather, nighttime illumination, and vessel motion. It will also examine temporal tracking, multispectral or polarization sensing, lightweight backbones, quantization, and cross-domain transfer to underwater datasets such as Trash-ICRA19.

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

Funding: This research was supported in part by the Key Research and Development Program Project in Hainan Province (ZDYF2026XDNY145), the Natural Science Foundation of Guangdong Province (2025A1515012901), the Fund of Guangdong Provincial Key Laboratory of Intelligent Equipment

for South China Sea Marine Ranching (2023B1212030003), the Scientific Research Start-up Funds of Guangdong Ocean University (060302062103 and 060302062202), and the Innovation Team Project for Ordinary Universities in Guangdong Province (2024KCXTD041).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source datasets used in this study are publicly available: the Garbage Collector Boat dataset on Roboflow, the HydroFloat dataset, FloW-Img [45], and MarineDebris-yolo on Roboflow. The processed custom dataset, including the scene-level training, validation, and test split lists and the offline-augmented images, together with the training configurations and evaluation scripts, is available from the corresponding author upon reasonable request.

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

Appendix A. Extended Notation for Algorithms and Local Derivations

Table A1 collects symbols that occur only in the forward pseudocode or within one local derivation. They are separated from Table 1 so that Section 3 emphasizes the notation needed repeatedly across modules.

Table A1. Extended notation used in the FLNet forward algorithms and local derivations.

Symbol	Context	Definition
$\mathbf{I}, H_0, W_0$	Forward pipeline	Input RGB image and its original spatial dimensions in Algorithm 1.
n	DPAR derivation	Number of repeated DPCBs in the processed partial-channel branch.
$p, \gamma, \mathbf{W}_p, \sigma(\cdot)$	DPAR derivation	Learnable kernel power $p = 1 + \gamma\sigma(\mathbf{W}_p)$, power-range coefficient ($\gamma = 4$), power parameter, and sigmoid function.
$\mathbf{Q}_{pos/neg}, \mathbf{K}_{pos/neg}$	DPAR derivation	Positive and negative ReLU-kernel components of the query and key tensors.
$\mathbf{Q}_{sim}, \mathbf{Q}_{opp}, \mathbf{K}_{pola}$	DPAR derivation	Similarity query, opposition query, and concatenated polarity key.
$C_2, e, \tilde{C}, \tilde{\mathbf{F}}_j$	AMRP alignment	Middle-level input channels, expansion ratio ($e = 0.5$), aligned channel count $\tilde{C} = \lfloor C_2 e \rfloor$, and aligned feature at level j.
$\mathbf{T}_i, \mathbf{U}_i, \mathbf{L}$	DPS-Head algorithm	Code-level tensors after the first shared DPConv–GN layer, prediction flattening, and concatenation across levels.
$\mathbf{A}, \mathbf{S}, \mathbf{Q}_{\text{box}}, \mathbf{D}$	DPS-Head algorithm	Anchor points, stride tensor, DFL-decoded box distribution, and final decoded detection tensor.

References

1. Canton, H. Organisation for economic co-operation and development—OECD. In *The Europa Directory of International Organizations 2021*; Routledge: London, UK, 2021; pp. 677–687.
2. Borrelle, S.B.; Ringma, J.; Law, K.L.; Monnahan, C.C.; Lebreton, L.; McGivern, A.; Murphy, E.; Jambeck, J.; Leonard, G.H.; Hilleary, M.A.; et al. Predicted growth in plastic waste exceeds efforts to mitigate plastic pollution. *Science* **2020**, *369*, 1515–1518.
3. Alimba, C.G.; Faggio, C. Microplastics in the marine environment: Current trends in environmental pollution and mechanisms of toxicological profile. *Environ. Toxicol. Pharmacol.* **2019**, *68*, 61–74. https://doi.org/10.1016/j.etap.2019.03.001.
4. Thompson, R.C.; Moore, C.J.; Vom Saal, F.S.; Swan, S.H. Plastics, the environment and human health: Current consensus and future trends. *Philos. Trans. R. Soc. B Biol. Sci.* **2009**, *364*, 2153–2166.
5. Maximenko, N.; Corradi, P.; Law, K.L.; Van Sebille, E.; Garaba, S.P.; Lampitt, R.S.; Galgani, F.; Martinez-Vicente, V.; Goddijn-Murphy, L.; Veiga, J.M.; et al. Toward the integrated marine debris observing system. *Front. Mar. Sci.* **2019**, *6*, 447.
6. Biermann, L.; Clewley, D.; Martinez-Vicente, V.; Topouzelis, K. Finding plastic patches in coastal waters using optical satellite data. *Sci. Rep.* **2020**, *10*, 5364.
7. Kikaki, A.; Karantzalos, K.; Power, C.A.; Raitsos, D.E. Remotely sensing the source and transport of marine plastic debris in Bay Islands of Honduras (Caribbean Sea). *Remote Sens.* **2020**, *12*, 1727.

8. Wang, Y.; Wang, M. Mitigating sun glint effects in Landsat OLI imagery for enhanced macroalgae biomass estimation. In *Proceedings of the Second International Conference on Remote Sensing and Global Positioning Algorithm (RSGPA 2025)*; SPIE: Bellingham, WA, USA, 2025; Volume 13818, pp. 125–132.
9. Wang, S.; Yu, C.; Sun, Y.; Gao, F.; Dong, J. Specular reflection removal of ocean surface remote sensing images from UAVs. *Multimed. Tools Appl.* **2018**, *77*, 11363–11379.
10. Wang, L.; Gao, Y.; Cai, G.; Shen, C. Energy-Aware Floating-Debris Detection for Battery-Powered Electric Unmanned Surface Vehicles: A Lightweight YOLO-Based Method with Embedded Profiling. *World Electr. Veh. J.* **2026**, *17*, 156. https://doi.org/10.3390/wevj17030156.
11. Garaba, S.P.; Dierssen, H.M. An airborne remote sensing case study of synthetic hydrocarbon detection using short wave infrared absorption features identified from marine-harvested macro- and microplastics. *Remote Sens. Environ.* **2018**, *205*, 224–235. https://doi.org/10.1016/j.rse.2017.11.023.
12. Tasseron, P.; van Emmerik, T.; Peller, J.; Schreyers, L.; Biermann, L. Advancing Floating Macroplastic Detection from Space Using Experimental Hyperspectral Imagery. *Remote Sens.* **2021**, *13*, 2335. https://doi.org/10.3390/rs13122335.
13. Kikaki, K.; Kakogeorgiou, I.; Mikeli, P.; Raitsos, D.E.; Karantzalos, K. MARIDA: A benchmark for Marine Debris detection from Sentinel-2 remote sensing data. *PLoS ONE* **2022**, *17*, e0262247. https://doi.org/10.1371/journal.pone.0262247.
14. van Lieshout, C.; van Oeveren, K.; van Emmerik, T.; Postma, E. Automated River Plastic Monitoring Using Deep Learning and Cameras. *Earth Space Sci.* **2020**, *7*, e2019EA000960. https://doi.org/10.1029/2019EA000960.
15. Yang, X.; Zhao, J.; Zhao, L.; Zhang, H.; Li, L.; Ji, Z.; Ganchev, I. Detection of river floating garbage based on improved YOLOv5. *Mathematics* **2022**, *10*, 4366.
16. Li, N.; Huang, H.; Wang, X.; Yuan, B.; Liu, Y.; Xu, S. Detection of floating garbage on water surface based on PC-Net. *Sustainability* **2022**, *14*, 11729.
17. Shi, C.; Lei, M.; You, W.; Ye, H.; Sun, H. Enhanced Floating Debris Detection Algorithm Based on CDW-YOLOv8. *Phys. Scr.* **2024**, *99*, 076019. https://doi.org/10.1088/1402-4896/ad5657.
18. Jiang, Z.; Wu, B.; Ma, L.; Zhang, H.; Lian, J. APM-YOLOv7 for Small-Target Water-Floating Garbage Detection Based on Multi-Scale Feature Adaptive Weighted Fusion. *Sensors* **2024**, *24*, 50. https://doi.org/10.3390/s24010050.
19. Dong, H.; Li, J.; He, X.; Zhang, M. Improved YOLOv8-SST for Accurate Detection of Small Floating Objects in Complex Water Environments. *PLoS ONE* **2026**, *21*, e0340822. https://doi.org/10.1371/journal.pone.0340822.
20. Sapkota, R.; Karkee, M. Ultralytics YOLO Evolution: An Overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 Object Detectors for Computer Vision and Pattern Recognition. *arXiv* **2025**, arXiv:2510.09653.
21. Jin, X.; Niu, P.; Liu, L. A GMM-based segmentation method for the detection of water surface floats. *IEEE Access* **2019**, *7*, 119018–119025. https://doi.org/10.1109/ACCESS.2019.2937129.
22. Olyaei, M.; Ebtehaj, A.; Ellis, C.R. A Hyperspectral Reflectance Database of Plastic Debris with Different Fractional Abundance in River Systems. *Sci. Data* **2024**, *11*, 1253. https://doi.org/10.1038/s41597-024-03974-x.
23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
24. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
25. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
26. Sohan, M.; Sai Ram, T.; Rami Reddy, C.V. A review on yolov8 and its advancements. In *Proceedings of the International Conference on Data Intelligence and Cognitive Informatics*; Springer: Singapore, 2024; pp. 529–545.
27. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 21–37.
28. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149.
29. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
30. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
31. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Proceedings of the Computer Vision—ECCV 2020*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12346, pp. 213–229. https://doi.org/10.1007/978-3-030-58452-8_13.
32. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
33. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125. https://doi.org/10.1109/CVPR.2017.106.

34. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768. https://doi.org/10.1109/CVPR.2018.00913.
35. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790. https://doi.org/10.1109/CVPR42600.2020.01079.
36. Du, Y.; Chen, L.; Hao, X. EViT-Net: An efficient vision transformer-inspired network for enhanced multi-scale remote sensing image features. *Expert Syst. Appl.* **2026**, *296*, 129123. https://doi.org/10.1016/j.eswa.2025.129123.
37. Yang, H.; Song, Y.; Liang, Y.; Tang, E.; Cao, D. SDC-YOLOv8: An Improved Algorithm for Road Defect Detection Through Attention-Enhanced Feature Learning and Adaptive Feature Reconstruction. *Sensors* **2026**, *26*, 609. https://doi.org/10.3390/s26020609.
38. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
39. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic Head: Unifying Object Detection Heads with Attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7373–7382.
40. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2018; pp. 3–19.
41. Meng, W.; Luo, Y.; Li, X.; Jiang, D.; Zhang, Z. PolaFormer: Polarity-aware Linear Attention for Vision Transformers. In Proceedings of the Thirteenth International Conference on Learning Representations, Singapore, 24–28 April 2025.
42. Zhu, J.; Chen, X.; He, K.; LeCun, Y.; Liu, Z. Transformers without Normalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 11–15 June 2025.
43. Shi, Z.; Hu, J.; Ren, J.; Ye, H.; Yuan, X.; Ouyang, Y.; He, J.; Ji, B.; Guo, J. HS-FPN: High Frequency and Spatial Perception FPN for Tiny Object Detection. *arXiv* **2025**, arXiv:2412.10116. https://doi.org/10.48550/arXiv.2412.10116.
44. Chen, Z.; He, Z.; Lu, Z.M. DEA-Net: Single Image Dehazing Based on Detail-Enhanced Convolution and Content-Guided Attention. *IEEE Trans. Image Process.* **2024**, *33*, 1002–1015.
45. Cheng, Y.; Zhu, J.; Jiang, M.; Fu, J.; Pang, C.; Wang, P.; Sankaran, K.; Onabola, O.; Liu, Y.; Liu, D.; et al. FloW: A Dataset and Benchmark for Floating Waste Detection in Inland Waters. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10933–10942. https://doi.org/10.1109/ICCV48922.2021.01077.
46. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974.
47. Feng, Y.; Huang, J.; Du, S.; Ying, S.; Yong, J.H.; Li, Y.; Ding, G.; Ji, R.; Gao, Y. Hyper-YOLO: When Visual Object Detection Meets Hypergraph Computation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 2388–2401. https://doi.org/10.1109/TPAMI.2024.3524377.
48. Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. Yolov9: Learning what you want to learn using programmable gradient information. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; pp. 1–21.
49. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.
50. Tian, Y.; Ye, Q.; Doermann, D. Yolov12: Attention-centric real-time object detectors. *arXiv* **2025**, arXiv:2502.12524.
51. Lei, M.; Li, S.; Wu, Y.; Hu, H.; Zhou, Y.; Zheng, X.; Ding, G.; Du, S.; Wu, Z.; Gao, Y. Yolov13: Real-time object detection with hypergraph-enhanced adaptive visual perception. *arXiv* **2025**, arXiv:2506.17733.
52. Sapkota, R.; Cheppally, R.H.; Sharda, A.; Karkee, M. YOLO26: Key architectural enhancements and performance benchmarking for real-time object detection. *arXiv* **2025**, arXiv:2509.25164.
53. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 618–626.
54. Fulton, M.; Hong, J.; Islam, M.J.; Sattar, J. Robotic Detection of Marine Litter Using Deep Visual Detection Models. In Proceedings of the 2019 International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019; pp. 5752–5758. https://doi.org/10.1109/ICRA.2019.8793975.

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