Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

PD-YOLO: A High-Resolution Detection and Dilated-Receptive-Field Feature Enhancement Network for Small Objects in Low-Altitude Remote Sensing

Lulu Liu
Xijing University
Rui Zhu
zhu_r10@163.com

Xijing University
Chenjia Guo
Xijing University
Yushuai Zhang
Xijing University
Zhibo Shi
Xijing University

Article

Keywords: low-altitude remote sensing, UAV-based object detection, small object detection, VisDrone, PD-YOLO, multi-scale features

Posted Date: June 4th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9604670/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

PD-YOLO: A High-Resolution Detection and Dilated-Receptive-Field Feature Enhancement Network for Small Objects in Low-Altitude Remote Sensing

Lulu Liu[1,2], Rui Zhu[1,2,*], Chenjia Guo[1,2], YuShuai Zhang[1,2], Zhibo Shi[1,2]

[1]School of Electronic Information, Xijing University, Xi'an 710123, China
[2]Xi'an Key Laboratory of High Precision Industrial Intelligent Vision Measurement Technology, Xijing University, Xi'an 710123, China
*Correspondence: zhu_r10@163.com

Abstract: To address the challenges of large scale variation, low pixel proportion of small objects, and complex backgrounds in low-altitude remote sensing images, this paper proposes PD-YOLO based on the YOLO11 framework. The model introduces a four-scale (P2-P5) prediction branch in the neck, incorporating a high-resolution P2 layer to improve localization and classification density for extremely small objects. In the backbone, the conventional C3k2 module is replaced with C3k2_DWR, which integrates parallel dilated convolutions with multiple dilation rates to expand the effective receptive field and enhance cross-scale context representation. Experiments on the VisDrone2019-DET and UAVDT datasets show that PD-YOLO achieves 37.0% mAP@0.5 on the VisDrone test set, outperforming the listed YOLO baselines, and reaches 98.5% mAP@0.5 on UAVDT, with conclusions consistent across benchmarks. Ablation studies indicate that the P2 branch contributes the main accuracy gain, and C3k2_DWR further improves performance complementarily, with about 2.65M parameters and 10.5 GFLOPs. In summary, PD-YOLO provides a structurally concise improvement for aerial small-object detection in VisDrone-like scenarios, delivering clear and attributable performance gains.

Keywords: low-altitude remote sensing; UAV-based object detection; small object detection; VisDrone; PD-YOLO; multi-scale features

1 Introduction

Low-altitude remote sensing object detection lies upstream in the perception chain for applications such as smart cities, traffic regulation, emergency rescue, and facility inspection: only after pedestrians, vehicles, and other targets are stably detected can downstream trajectory association, flow statistics, and situation assessment obtain trustworthy inputs [1],[2]. The advancement of fifth- and sixth-generation mobile communications, the low-altitude economy, and integrated space-air-ground sensing continuously increases the deployment density of networked and onboard UAVs and the volume of imaging data; task scenarios and background complexity are rising, imposing stricter requirements on real-time algorithms, edge deployability, and reproducible evaluation. In bird's-eye street scenes and wide-area patrol, instances with extremely small pixel occupancy coexist with very large scale span; missed and false detections propagate directly into downstream analysis error. Therefore, developing efficient and robust aerial small-object detection has become one of the key topics in low-altitude visual intelligence and unmanned systems.

Deep learning-based detection has made substantial progress in generic scenes, yet under aerial, small-object-dense settings it still faces difficult measurement, unstable localization, and high miss rates [3],[4]; low-altitude remote sensing images often have a large field of view and complex background texture, and objects of the same class appear extremely small on the image plane, which makes the above contradictions more pronounced. The VisDrone series provides standardized evaluation protocols and large-scale annotations [5] and has become an important platform for testing aerial detection algorithms; for such data, without targeted multi-scale and receptive-field design, relying solely on inference-time compensations such as raising input resolution, test-time multi-scale fusion, or tiling-style inference often cannot simultaneously satisfy accuracy and throughput demands under limited compute and edge latency constraints.

Traditional two-stage detectors rely mainly on region proposal generation and cascaded refinement and historically hold advantages on several accuracy metrics [6]. Such methods still show clear limitations in real low-altitude video streams and onboard inference: first, long inference chains and high intermediate feature occupancy make it hard to meet millisecond-level latency and lightweight deployment; second, for extremely small objects, matching proposals to ground truth is sensitive to the IoU threshold, and early localization errors tend to accumulate through cascade stages; third, region proposal, feature cropping, and post-processing are often relatively independent, with high system coupling and experience-heavy tuning, which is unfavorable for forming an end-to-end detection pipeline that can be optimized in a unified manner.

Deep learning provides a new route of end-to-end representation learning for aerial object detection. Models can automatically learn multi-scale features, with compact inference structure, low reliance on hand-crafted features, and easy alignment and reproduction with open frameworks such as YOLO. Single-stage detectors represented by YOLO integrate localization and classification as dense prediction and are more attractive in speed and engineering [7],[8],[9],[10]; multi-scale neck structures such as the Feature Pyramid Network (FPN) [12] and Path Aggregation Network (PANet) [16] fuse high-level semantics and shallow detail through top-down paths, lateral connections, and bottom-up feedback, and have become a mainstream paradigm for alleviating scale variation. On this basis, the YOLO lineage continues to iterate in backbones, attention, and label assignment; survey-style organization of generations such as YOLO11 appears in [11]; engineering implementation and training configurations can be aligned and reproduced with the Ultralytics open-source repository [13]. On a nearby timeline there are also discussions such as YOLOv9 on gradient paths and architectural reversibility [14], and structural analyses targeting YOLOv8/YOLO-style implementations [15]. However, default YOLO configurations usually place detection heads on relatively deep P3-P5; shallow high-resolution information is easily weakened after repeated downsampling, and extremely small instances tend to be ill-posed for regression on a coarse prediction grid; meanwhile, if the backbone does not sufficiently expand the effective receptive field within the same feature resolution, large-scale targets and complex backgrounds compete for channel capacity, coupling insufficient high-resolution detail with receptive-field mismatch and limiting the performance ceiling on dense small-object data such as VisDrone2019.

While preserving the single-stage paradigm and strict comparability to the community YOLO11 implementation, introducing higher-spatial-resolution prediction branches in the neck-detection head and explicitly expanding the effective receptive field in the backbone via parallel dilated convolutions provide a unified structural answer to the above contradictions: on the one hand, higher-resolution detection layers can offer a denser regression and classification sampling plane for extremely small objects without increasing the physical resolution of the input image; on the other hand, dilated multi-branch structures can, while controlling kernel size and feature map spatial size, mix local texture and mid-to-long-range context within the same scale, forming a mechanistic complement to top-down/bottom-up cross-scale fusion. However, blindly stacking heterogeneous attention in the backbone or neck or deepening the network tends to rapidly increase parameters and latency, and gains are hard to attribute clearly in a paper; it is therefore necessary,

under a YOLO11 macro topology and training recipe that remain comparable, to select two structurally clear paths that can be independently toggled in implementation for reinforcement.

For the two coupled difficulties-insufficient high-resolution detail and mismatched effective receptive field-in low-altitude remote sensing small-object detection, this paper proposes PD-YOLO on the YOLO11 framework. On the one hand, we introduce a four-branch prediction path at the P2 scale in the detection head so that the network directly regresses extremely small objects on higher-spatial-resolution feature maps, thereby mitigating information loss from early downsampling; this idea is in the same line as multi-scale neck designs such as FPN and path aggregation (see citation indices for FPN and PANet above). On the other hand, in the backbone we replace conventional C3k2 modules with a dilated multi-branch structure inspired by DWR (Dilation-wise Residual), capturing context under multiple dilation rates in parallel to strengthen feature expression for objects with extremely large scale span. DWR-style designs were first systematically argued for multi-scale context acquisition in real-time semantic segmentation [17], with a mathematical basis consistent with dilated convolutions [18]. We combine them complementarily with the P2 detection branch to form PD-YOLO for low-altitude remote sensing small-object detection. Compared with stacking many heterogeneous attention modules in the backbone or neck or introducing high-complexity global self-attention, we strengthen only a fine-grained localization outlet and multi-receptive-field semantic mixing through two clear, independently switchable structural paths while keeping the YOLO11 macro topology and training recipe comparable, which facilitates attribution of performance gains in the main text. The experimental part takes VisDrone2019 as the primary benchmark, compares against baseline YOLO11 and staged improvements under a unified training setup, and conducts supplementary cross-dataset comparison on UAVDT.

The main contributions of this article are as follows.

(1) Addressing application needs of low-altitude remote sensing small-object detection in large-scale screening, wide-area deployment, and real-time inference, we propose PD-YOLO, a real-time object detector built on the single-stage YOLO11 framework.Four-scale P2-P5 prediction with an independent high-resolution outlet for tiny objects.

(2) We introduce four-scale (P2-P5) prediction along the neck-detection head path with P2 participation: tiny objects are regressed directly on higher-spatial-resolution feature maps, providing an independent detection outlet for targets with extremely low pixel occupancy and alleviating the weakening of shallow-layer detail after repeated downsampling.

(3) We replace conventional C3k2 in the backbone with C3k2_DWR, which integrates a DWR (Dilation-wise Residual) multi-rate dilated parallel structure to mix multiple effective receptive fields on single-scale feature maps and strengthen contextual representation.

(4) We conduct main comparisons and ablation studies on VisDrone2019, complementary cross-dataset comparisons on UAVDT, verify the effectiveness of the proposed method, and analyze the trade-off between computational cost and detection performance.

Paper organization: Section 2 systematically reviews general object detection, multi-scale feature fusion, receptive-field expansion, and small-object detection in UAV imagery; it summarizes strengths and weaknesses of existing methods in low-altitude remote sensing settings and clarifies the entry points of our improvements. Section 3 first presents the datasets and evaluation benchmarks, then unfolds the PD-YOLO architecture following overall principles-network integration-backbone DWR-neck P2-loss and training, and finally states the evaluation metrics in a unified manner. Section 4 gives the experimental environment and implementation details, reports experimental results of PD-YOLO, compares and analyzes performance of different models on VisDrone2019 and UAVDT, and presents ablation studies. Section 5 summarizes the main conclusions and contributions, discusses applicable conditions and limitations of the method, and outlines future work on more aerial benchmarks, cross-dataset generalization, and lightweight deployment.

2 Related Work

Traditional object detection pipelines are organized mainly around hand-crafted or shallow features and staged modules. Early work often chained candidate window generation, feature encoding, and classifier decisions: sliding windows and selective-search-style proposal strategies covered space; HOG, SIFT, and other hand-crafted descriptors encoded local texture; shallow classifiers such as support vector machines then performed category discrimination; deep two-stage networks further used region proposal networks (RPN) and cascaded refinement to improve localization, yet a clear staged division between backbone and detection head remains. The evolution from hand-crafted features to two-stage convolutional paradigms, the chained accumulation of proposal versus classification error, and the tension of real-time deployment were already surveyed in the introduction and are not re-numbered here. The above paradigms can still achieve stable accuracy under ideal distributions and medium object scales, but under low-altitude remote sensing with large field of view, extremely small instances, and complex backgrounds, reliance on manual priors and threshold tuning limits performance and transferability; besides recognition rate and recall, the disjoint proposal-feature-classification-postprocessing pipeline also lacks a unified optimization objective, and front-end localization bias strongly amplifies subsequent classification and box regression error. Therefore, on onboard edge devices and in video-stream scenarios, relying solely on classical multi-stage pipelines is hard to simultaneously balance accuracy, stability, and engineering deployability.

Deep convolutional networks provide a new route of end-to-end representation learning for generic object detection. Progress in the YOLO single-stage family, FPN/PANet-style multi-scale necks, and YOLO11 engineering has already been cited in the introduction; this section supplements several detection branch designs closely related to aerial transfer. The anchor-free paradigm FCOS replaces anchor boxes with per-pixel prediction [19]; ATSS alleviates training differences between anchor-based and anchor-free settings via adaptive training sample selection [20]. Query-based DETR uses a Transformer encoder-decoder and bipartite matching for set-style prediction [21]; RT-DETR and similar variants further trade speed and accuracy for real-time scenarios [22]. Task-aligned one-stage detection TOOD and Generalized Focal Loss improve stability of dense prediction heads from the angle of decoupling classification and regression heads and re-weighting losses [23],[24]. However, mainstream evaluation of the above works still concentrates on natural scenes and COCO-like distributions; when detectors are directly transferred to aerial imagery, difficulties in measuring tiny objects, long-tailed classes, and background interference become salient, and benchmarks such as VisDrone also show that data protocols and structural design need dedicated adaptation-consistent with aerial-domain difficulties already discussed in the introduction; we therefore do not repeat citation indices already used there.

Combining multi-scale feature fusion with the single-stage paradigm has become a mainstream technical route for alleviating scale variation. The topological ideas of FPN and PANet and their realization paths in the YOLO family; on this basis, Focal Loss by Lin et al. suppresses domination of the loss by easy negatives and stabilizes dense-prediction training [25]. EfficientDet by Tan et al. jointly adjusts depth, width, and feature resolution via compound scaling and balances efficiency and accuracy [26]; NAS-FPN by Ghiasi et al. explores learnable pyramid topology from a neural architecture search perspective [27]. Together these works show that design of the neck and of the set of detection scales is as important as loss form and label assignment strategy. Compared with configurations that place detection heads only on relatively deep features such as P3-P5, adding prediction branches at higher spatial resolutions such as P2 can provide a denser regression grid for extremely small objects without changing the input image size; this idea also echoes YOLO-family engineering practice described in the introduction. It must be stressed that data augmentation alone (e.g., copy-paste, mosaic) helps alleviate sample scarcity but does not change the internal scale layout

of the network during a single forward pass; under data and training settings that can be strictly aligned with baselines, we attribute structural gains mainly to the P2 branch and backbone receptive-field enhancement.

Expanding the effective receptive field is another design thread closely tied to small objects and large scale span. The theoretical motivation of dilated convolution and the multi-branch form of DWR modules. Detection research oriented to remote sensing and UAV viewpoints has systematically analyzed challenges such as arbitrarily oriented objects, dense layout, and extremely small instances on large-scale aerial benchmarks such as DOTA [28],[29], providing empirical support for incorporating multi-branch and dilated convolutions into detection backbones or necks. It must be stressed: dilated multi-branch structures mainly strengthen within-scale contextual mixing, whereas high-resolution detection layers mainly provide a prediction plane with finer spatial stride; the two are functionally complementary and cannot replace each other.

In recent years researchers have also reused generic detectors and mitigated aerial small-object misses via engineering means such as higher input resolution, test-time multi-scale fusion, or tiling. Such methods raise effective sampling density to some extent but often at the cost of memory, compute, and end-to-end latency, and they do not structurally increase the high-frequency spatial degrees of freedom available to the network under a fixed forward resolution. By contrast, explicitly introducing detection branches at higher stride resolution in the neck, or strengthening within-scale multi‑receptive-field mixing in the backbone, better serves the coexistence of extremely small “grain” targets and large-scale instances in a single forward pass [30].

Dedicated UAV-image datasets and survey work have driven domain methods (VisDrone-DET2019, UAV detection surveys, tiny-object metrics, and DOTA-like aerial benchmarks, etc.). Around these data, prior work has improved YOLO-style detectors from angles such as attention and feature re-parameterization, reparameterized convolution, lightweight downsampling, and super-resolution auxiliary branches. Compared with routes that stack heterogeneous modules or rely on multi-stage post-processing, this paper, under a topology comparable to the official YOLO11 implementation, strengthens structure only through two separable ablation paths-the P2 four-scale neck-detection head and backbone C3k2_DWR-so that performance gains can be clearly attributed in the paper and the same template can be reused on other aerial benchmarks such as DOTA.

Taken together, generic deep detection is relatively mature in natural scenes, yet under low-altitude remote sensing small-object settings one still needs, without significantly sacrificing single-forward efficiency, to jointly resolve insufficient high-resolution prediction planes and mismatched within-scale effective receptive fields; traditional staged pipelines and pure inference-time compensation are both hard to close these two gaps mechanistically. In label assignment and loss form, TOOD, GFL, and related studies provide task alignment and distribution modeling for dense prediction heads; this paper keeps the YOLO11 default configuration and concentrates innovation on multi-scale layout and backbone convolution mixing to ensure fair comparison with strong baselines.

A systematic literature review shows that although many YOLO improvements target VisDrone-like benchmarks, relatively few studies simultaneously depart from a P2-level high-resolution detection outlet and DWR-type multi-rate dilated backbone enhancement as two complementary paths while keeping the macro topology comparable and reproducible within the Ultralytics lineage; to improve detection of extremely small objects and context modeling under controlled complexity, this paper builds PD-YOLO on the YOLO11 framework and gives structural details and experimental validation in the following sections.

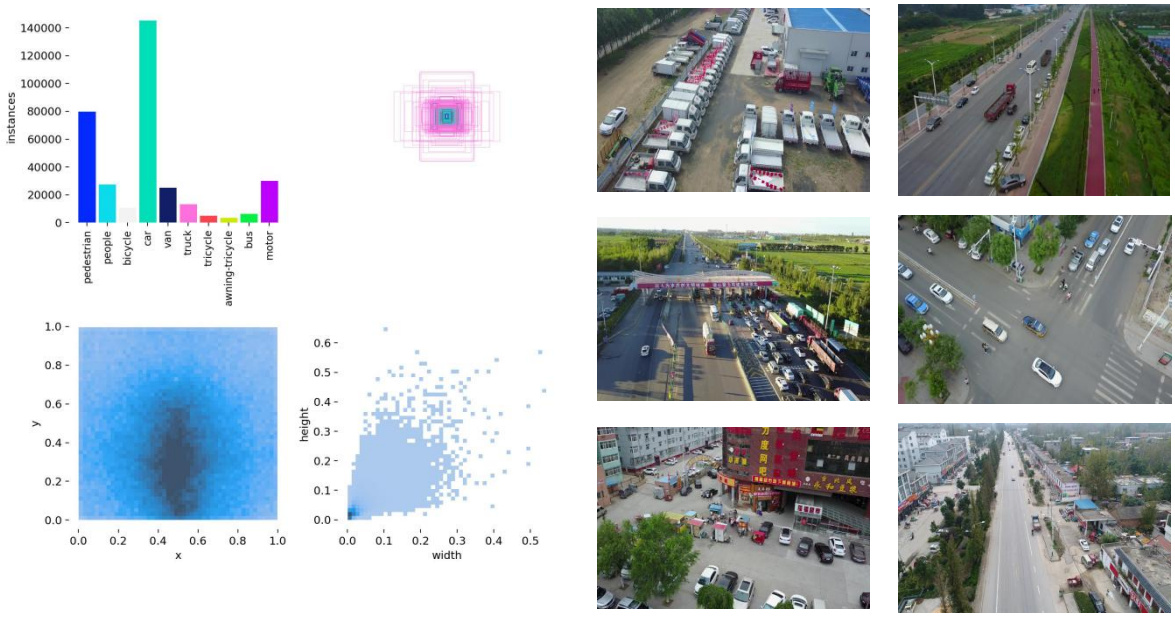
3 Methodology

Single-stage detectors cast classification and localization jointly as a dense prediction problem; their performance depends heavily on whether the backbone can encode sufficiently large context under a fixed

computational budget, and on whether the neck and detection heads can simultaneously preserve fine-grained boundary cues and high-level semantics at multiple spatial resolutions. On aerial benchmarks such as VisDrone2019, objects often occupy only a few to a few tens of pixels and are tightly coupled with backgrounds such as roads and buildings; if prediction units are allocated only on feature maps with large stride, misalignment between anchors or grid cells and target geometry readily appears, and small instances may be drowned by background responses. Motivated by these observations, this paper constructs PD-YOLO within the YOLO11 detection paradigm: the letter P denotes introducing a high-resolution P2 branch on top of the original P3-P5 setup to form four-scale detection; the letter D denotes replacing the standard C3k2 with C3k2_DWR, which carries multi-rate dilated branches, in backbone cross-stage refinement paths, so that at each scale the feature map explicitly mixes local texture with mid-to-long-range context within the same resolution, consistent with the design motivation of the DWR module in DWRSeg.

3.1 Datasets

The dataset used in this paper is VisDrone2019-DET. It is collected by UAVs in real urban scenes; images cover varying altitude, viewpoint, and illumination, exhibit large scale span, and show pronounced confusion between extremely small objects and the background, posing challenges to both localization and classification. The benchmark provides standardized splits for the detection task: a training set, a validation set, and a competition-oriented test subset. The training set contains 6471 images and the validation set 548 images. All main conclusions in this paper are reported with metrics on VisDrone2019. Ten detection categories are defined, in order: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. They include fine-grained targets such as pedestrians and cyclists as well as multiple vehicle types; dataset examples are shown in Figure 1. Class instance counts follow a clear long-tailed distribution-vehicles and pedestrians often dominate, whereas fine categories such as bicycles and tricycles are relatively scarce-which, superposed with scale differences under the aerial viewpoint, imposes additional requirements on training stability and on how results should be interpreted.

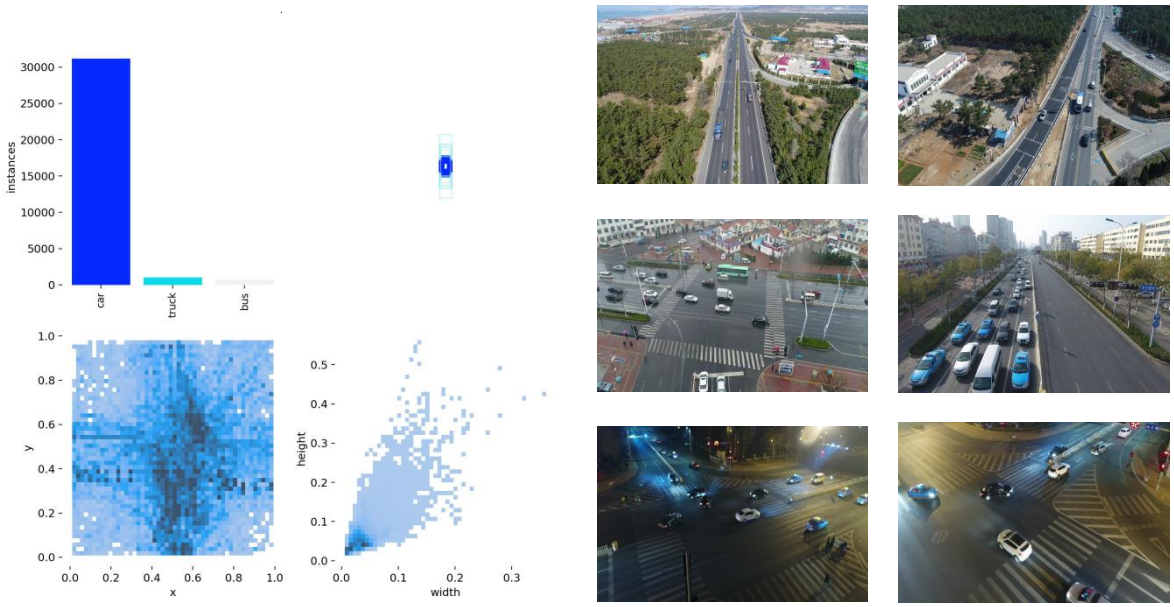


(a) (b)

Figure 1 Visualization of the VisDrone2019 dataset and sample images: (a) label schema of VisDrone2019; (b) sample images from VisDrone2019

To verify the effectiveness of the proposed method, we additionally conduct complementary cross-dataset comparison on UAVDT. UAVDT selects a large number of representative keyframes from roughly

ten hours of raw video and provides box-level annotations; the detection task focuses on vehicles in road environments, with categories generally corresponding to car, truck, and bus, and with multi-dimensional attributes such as flight altitude, camera viewpoint, weather, and occlusion, which help observe model behavior under another aerial regime in experiments; dataset examples are shown in Figure 2. The official protocol uses sequence-level splits for detection and tracking tasks: for detection, 30 training sequences and 20 test sequences are commonly used, with training and testing geographically disjoint to reduce overfitting to specific scenes.



(a) (b)

Figure 2 Visualization of the UAVDT dataset and sample images: (a) label schema of UAVDT; (b) sample images from UAVDT

3.2 Model architecture

This paper takes the YOLO11 single-stage detection pipeline as the backbone and embeds C3k2_DWR and P2 four-scale detection in stages: C3k2_DWR is placed in each backbone cross-stage refinement path, where parallel 3×3 convolution responses are fused, projected by 1×1 convolutions, and combined with residual connections to enlarge the effective receptive field and enrich same-scale multi-context semantics while controlling parameter count; the P2 branch sits on the neck-detection-head side, connecting with top-down (FPN-style) semantic injection and bottom-up (PANet-style) detail back-propagation, and adds a prediction outlet on the finest detection layer at roughly one-quarter input resolution to alleviate ill-posed regression for extremely small objects on coarse grids. Overall, the backbone emphasizes same-scale multi-receptive-field mixing, and the neck-head emphasizes cross-scale alignment and a high-resolution prediction plane, while preserving a macro topology comparable to Ultralytics for joint strengthening of fine-grained localization and long-range context modeling in aerial scenes.

PD-YOLO network integration

The PD-YOLO network is illustrated in Figure 3. RGB input is used; batch size and spatial resolution are denoted B , H , and W (in experiments we often set $H = W = 640$). The input tensor is $I \in \mathbb{R}^{B\times 3\times H\times W}$ .The stem applies several stride-2 convolutions and C3k2_DWR stages to downsample progressively, yielding multi-scale backbone features. Let s_2, s_3, s_4, s_5 denote the strides relative to the input at each scale. The backbone feature tuple and the stride values are summarized in Equation (1).

$$\{F_{P2}^{\mathrm{bb}}, F_{P3}^{\mathrm{bb}}, F_{P4}^{\mathrm{bb}}, F_{P5}^{\mathrm{bb}}\}, \quad s_2 = 4, \qquad s_3 = 8, \qquad s_4 = 16, \qquad s_5 = 32 \tag{1}$$

The neck first injects semantics from the deepest level in a top-down path: starting from P_5, features are upsampled stage by stage, concatenated with backbone P_4, P_3, P_2 along the channel dimension, and fused by C3k2 to obtain the shallowest neck feature F_{P2}^{neck}. It then follows a bottom-up path from P_2: repeated stride-2 downsampling together with side-branch features (Concat + C3k2) rebuilds the four-level feature tower $\{P_2, P_3, P_4, P_5\}$ used for detection.

The detection head outputs classification logits and bounding-box parameters (including DFL distribution coefficients) in parallel on these four paths; final boxes are obtained after decoding and non-maximum suppression. This hierarchical representation simultaneously preserves shallow high-resolution structural cues and deep large-receptive-field semantics, providing well-aligned dense prediction features for scenes where extremely small objects coexist with complex backgrounds.

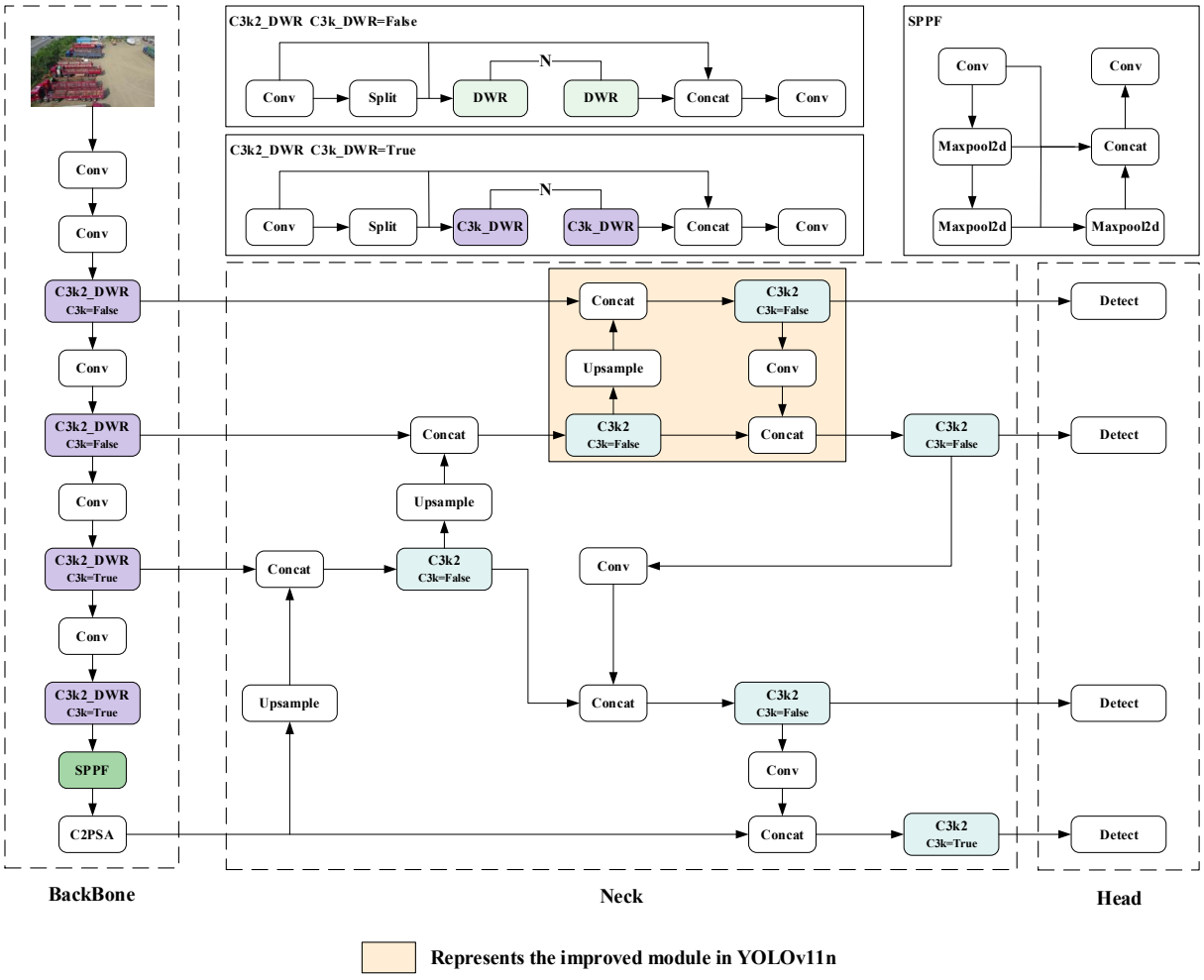


Figure 3 Overall architecture of PD-YOLO

Backbone encoding and C3k2_DWR

Shallow features should preserve localization cues such as edges and texture; deep features should cover larger image regions to encode object-level semantics; deep stacking and residual design in generic vision backbones are systematically discussed elsewhere [31]. Enlarging the receptive field solely by increasing kernel size or blindly stacking convolutions quickly raises parameter count and latency. Dilated convolution inserts gaps between sampling points of the kernel on the 2-D discrete grid so that a single convolution covers a larger input neighborhood with the same number of kernel elements. Let p denote an output lattice point, k an index over the kernel, and r the dilation rate ($r = 1$ reduces to standard convolution); in the single-channel case the forward map is as in Equation (2).

$$y(p) = \sum_{k} w_k \cdot x(p + r \cdot k) \tag{2}$$

When $r > 1$, the effective coverage grows approximately linearly with r; in aerial scenes with pronounced scale variation, parallel multi-rate dilation forms complementary receptive fields at the same resolution and supplies richer single-scale representations to the neck for cross-scale fusion.

While keeping the stem Conv downsampling chain, SPPF, and C2PSA unchanged, all C3k2 blocks used for cross-stage refinement in the backbone are replaced by C3k2_DWR. The DWR submodule first applies a 3×3 convolution to compress the input $x \in \mathbb{R}^{C \times H \times W}$ to $C/2$ channels, then runs three parallel 3×3 dilated convolutions with $d \in \{1,3,5\}$, concatenates the three outputs along the channel dimension, maps back to C channels with a 1×1 convolution, and adds the residual x. Let ϕ_d denote the composition of convolution and activation at dilation d, and ψ the 1×1 projection; the forward pass is as in Equations (3) and (4).

$$t = \mathrm{Conv}_{3\times3}(x), \quad t_d = \phi_d(t), \quad d \in \{1,3,5\} \tag{3}$$

$$y = x + \psi(\mathrm{Cat}(t_1, t_3, t_5)) \tag{4}$$

C3k2 is a widely used cross-stage partial-connection bottleneck stack in YOLO11; CSP-style local bottlenecks have a long history in YOLO-family backbones [32]. Replacing the inner bottleneck with the DWR block above yields C3k2_DWR, which strengthens local convolutional mixing without changing the number of stages, downsampling positions, or the channel-width schedule.

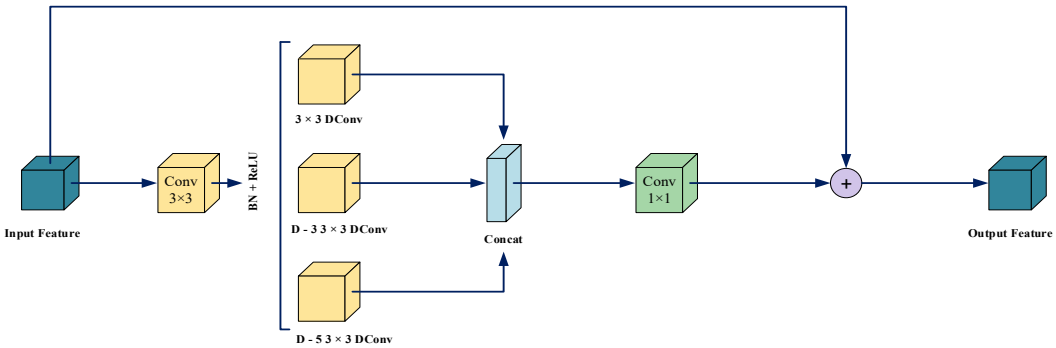


Figure 4 Structure of the DWR module

Neck and P2 four-scale detection branch

YOLO11 by default places detection heads only on P_3-P_5, so each cell covers a relatively wide input footprint; in VisDrone2019 many instances have long side below 32 pixels, and a coarse grid easily leads to ill-posed regression. Introducing the P_2 scale with $s_2 = 4$ amounts to adding a prediction plane at roughly one-quarter input resolution. The neck topology follows the community yolo11-p2 layout: top-down, semantics from P_5 are upsampled stage-wise and fused with backbone P_4, P_3, P_2; bottom-up, starting from the shallowest feature, stride-2 downsampling with side Concat + C3k2 rebuilds the P_2-P_5 four-level detection tower. Denote the backbone P_2 output, the upsampled feature from deeper levels, and the upsampling operator by F_{P2}^{bb}, $U(F_{\mathrm{in}})$, and $U(\cdot)$ as in Equation (5).

$$F_{P2}^{\mathrm{bb}}, \quad U(F_{\mathrm{in}}), \quad U(\cdot) \tag{5}$$

Here $U(\cdot)$ denotes bilinear or nearest-neighbor upsampling with optional 1×1 alignment (as implemented). The layer fused into P_2 is written as Equation (6).

$$F_{P2}^{\mathrm{neck}} = \mathrm{C3k2}(\mathrm{Cat}(U(F_{\mathrm{in}}), F_{P2}^{\mathrm{bb}})) \tag{6}$$

The Detect head outputs classification logits and box parameters on P_2, P_3, P_4, and P_5; the regression branch uses DFL consistent with YOLO11, discretizing continuous offsets into multiple bins and forming a softmax-weighted sum for smoother regression gradients. Four scales versus three increase Params and GFLOPs in the head and part of the neck, but on dense small-object data this is usually an acceptable cost for gains in mAP@0.5 and recall.

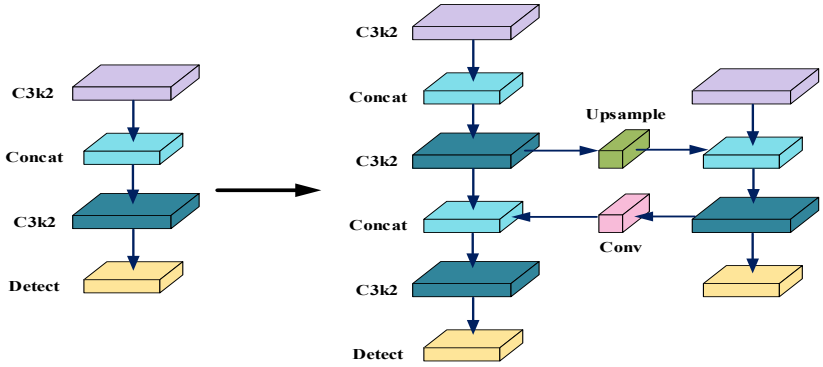


Figure 5 P2 four-scale neck and feature flow

3.3 Evaluation metrics

To evaluate model performance comprehensively, this paper adopts multiple metrics: computational cost (GFLOPs) and parameter count (Parameters) for complexity, and precision (P), recall (R), average precision (AP), mean average precision (mAP), and related detection scores as summarized in Table 1.

Table 1 Evaluation metrics

Metric	Symbol	Description
Parameter count	Parameters	Total number of learnable parameters; obtained from model statistics at a fixed input resolution.
Computational cost	GFLOPs	Approximate floating-point operations for one forward pass (order of 10^9); depends on input resolution and architecture.
Precision	P	Fraction of positive predictions that match ground truth.
Recall	R	Fraction of ground-truth instances that are detected.
Average precision	AP	Area under the precision-recall curve for one class.
Mean average precision	mAP	Arithmetic mean of per-class AP over all categories.
mAP at IoU 0.5	mAP@0.5	mAP computed at IoU threshold 0.5.
F1 score	F1	Harmonic mean of precision and recall.

GFLOPs and Parameters are counted under fixed input resolution; $P, R, \mathrm{AP}, \mathrm{mAP}$, etc. are computed from predictions and ground truth on the validation or test set under the prescribed IoU rules, with interpolation aligned to the Ultralytics evaluation scripts and COCO conventions.

Precision measures the fraction of samples predicted positive that are truly positive, as in Equation (7), where TP (true positive) is the number of correctly detected positives and FP (false positive) the number of negatives incorrectly predicted as positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

Recall measures the fraction of all true positives that are correctly detected, as in Equation (8), where FN (false negative) is the number of true positives missed.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

Average precision (AP) integrates precision over recall levels to summarize per-class detection quality, as in Equation (9).

$$\text{AP} = \int_0^1 \text{P(r)dr} \tag{9}$$

Mean average precision (mAP) averages AP over all classes for an overall multi-class score, as in Equation (10).

$$\text{mAP} = \frac{1}{\text{N}} \sum_{\text{i=1}}^{\text{N}} \text{AP}_\text{i} \tag{10}$$

Where N is the number of classes and AP_i the AP of class i.

We further report mAP@0.5: mAP computed at intersection-over-union (IoU) threshold 0.5-a prediction is considered correct when IoU with the ground-truth box exceeds 0.5; it jointly reflects localization and classification ability, as in Equation (11) .

$$\text{maP@0.5} = \frac{1}{\text{N}} \sum_{\text{i=1}}^{\text{N}} \text{AP}_\text{i}^{\text{IoU=0.5}} \tag{11}$$

The F1 score balances precision and recall and summarizes overall behavior, as in Equation (12).

$$\text{F1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

4 Experimental results

4.1 Experimental environment

All experiments reported in this paper were carried out on a deep-learning workstation; hardware and software are listed in Table 2.

Table 2 Experimental environment

Item	Specification
CPU	Intel(R) Core(TM) Ultra 7 265K
System	Windows 11
GPU	NVIDIA GeForce RTX 3090
RAM	64.0 GB
Python	3.9.23
CUDA	12.1
Pytorch	2.5.1

Under these conditions, training runs for 300 epochs with batch size 16, initial learning rate 0.01, momentum 0.937, weight decay 0.0005, input resolution 640×640, and four DataLoader worker threads.

4.2 Comparative experiments

Under the experimental environment described above, we compare PD-YOLO with multiple mainstream object detectors from the YOLO family on the VisDrone and UAVDT datasets; quantitative results are summarized in Table 3.

On the VisDrone split, PD-YOLO achieves 47.2% precision, 36.6% recall, and 37.0% mAP@0.5-all higher than any listed baseline. mAP@0.5 improves by 2.7 and 3.0 percentage points over the next-best YOLOv10n and YOLOv9t, respectively, and by 3.9 points over the same-series baseline YOLOv11n, ranking

first among the compared methods. Precision improves by 1.8 and 2.0 points over YOLOv8n and YOLOv10n; recall at 36.6% is also the highest in the table, indicating that overall detection performance is favorable in dense small-object aerial scenes.

On UAVDT, PD-YOLO attains 98.5% mAP@0.5, higher than YOLOv11n, YOLOv12n, and other listed baselines, consistent with the VisDrone conclusions and indicating strong classification and localization accuracy on this vehicle-centric aerial set.

For model complexity, PD-YOLO has 2.65M parameters and 10.5 GFLOPs, higher than YOLOv11n (2.58M, 6.4 GFLOPs), reflecting a deliberate trade of moderate extra compute and parameters for substantial accuracy gains. Compared with YOLOv6n, which has both more parameters and higher cost (4.63M, 11.3 GFLOPs), PD-YOLO achieves higher mAP@0.5 at lower GFLOPs, yielding a better accuracy-efficiency trade-off. Compared with the ultralight YOLOv5n (1.78M, 4.2 GFLOPs), PD-YOLO improves mAP@0.5 by 5.0 percentage points, making it suitable for UAV-view applications where detection accuracy is prioritized.

Table 3 Validation results of compared models

Model	VisDrone			UAVDT			Params	GFLOPs
	P(%)	R(%)	mAP@0.5	P(%)	R(%)	mAP@0.5		
YOLOv5n	42.3	32.3	32.0	96.8	93.6	98.1	1775401	4.2
YOLOv6n	41.5	31.7	31.0	93.7	93.8	97.5	4633568	11.3
YOLOv8n	45.4	33.0	33.3	94.5	95.2	98.0	3007598	8.2
YOLOv9t	44.8	33.9	34.0	95.6	93.7	97.9	2620460	10.7
YOLOv10n	45.2	34.0	34.3	93.9	96.9	98.3	2708210	8.4
YOLOv11n	44.4	33.2	33.1	94.4	93.7	97.9	2584102	6.4
YOLOv12n	44.6	32.8	32.8	96.5	96.8	97.6	2558678	6.4
YOLOv13n	43.4	32.5	32.4	91.9	95.4	97.8	2460496	6.3
YOLOv26n	43.9	33.6	33.1	94.4	93.7	97.9	2590409	6.4
PD-YOLO	47.2	36.6	37.0	96.2	95.4	98.5	2654456	10.5

Table 4 lists per-class AP@0.5 on VisDrone2019 for all ten categories and the class-mean summary. The proposed PD-YOLO reaches 37.0% mAP@0.5, the highest among the listed YOLO-family baselines-about 2.7 and 3.0 points above the next-best YOLOv10n and YOLOv9t, and about 3.9 points above YOLOv26n-so the overall margin is clear. Per-class results show that PD-YOLO does not rely on a single category to inflate the mean: for pedestrian, people, and bicycle, which are particularly sensitive in aerial detection, AP@0.5 is 42.9%, 33.8%, and 11.2%, respectively; pedestrian improves by 6.3 points over YOLOv10n and people by 4.4 points, which helps stably detect tiny pedestrians and crowded groups from the UAV viewpoint. Among vehicle-and riding-related classes, PD-YOLO is best in the table on car (80.2%), van (41.3%), tricycle (23.6%), bus (49.8%), and motor (43.5%); truck (30.2%) and awning-tricycle (13.0%) may trade slightly with individual baselines, yet the ten-class aggregate still raises mAP@0.5, indicating that the joint P2 high-resolution path and DWR-style mixing provide fuller scale coverage and deformation modeling for dominant traffic and riding targets without sacrificing balance across classes.

Table 4 Per-class mAP@0.5 on VisDrone2019

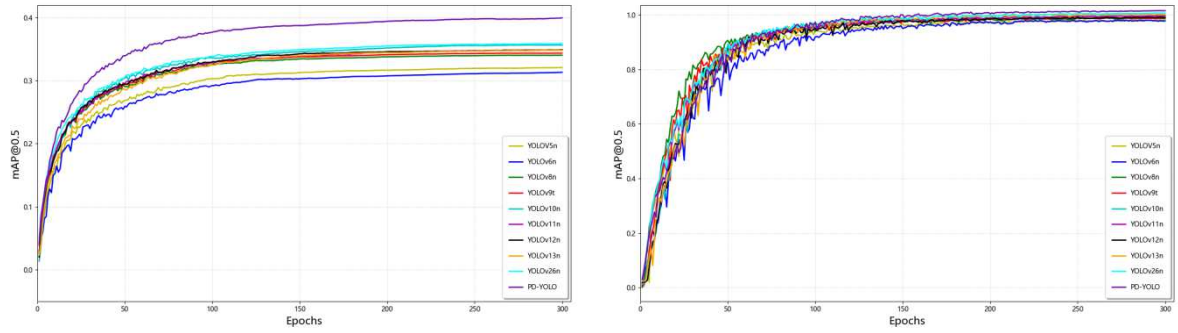
Model/Type	pedestrian	people	bicycle	car	van	truck	tricycle	aw-tr	aw-tr	motor	mAP@0.5
YOLOv5n	34.6	27.7	7.69	75.2	37.6	26.5	19.5	11.6	44.4	36.1	32.0
YOLOv6n	32.4	26.2	5.35	74.6	36.5	27.8	18.4	10.5	44.8	33.1	31.0
YOLOv8n	35.8	27.2	8.76	76.4	38.6	29.0	20.7	12.6	46.6	37.6	33.3
YOLOv9t	34.8	27.0	9.17	76.2	39.3	30.6	23.0	14.0	48.6	37.1	34.0

YOLOv10n	36.6	29.4	11.1	76.8	38.3	28.8	22.2	12.0	49.0	38.9	34.3
YOLOv11n	35.2	26.9	7.74	75.9	39.4	29.8	22.0	11.1	45.7	35.9	33.0
YOLOv12n	35.0	27.6	9.0	76.0	38.3	28.4	20.6	10.8	45.7	36.6	32.8
YOLOv13n	34.5	26.6	8.3	75.2	37.0	27.4	20.4	12.3	46.8	35.5	32.4
YOLOv26n	35.3	27.3	9.13	75.9	38.6	29.3	21.9	12.5	44.4	37.2	33.1
PD-YOLO	42.9	33.8	11.2	80.2	41.3	30.2	23.6	13.0	49.8	43.5	37.0

Figure 6 plots validation mAP@0.5 versus epoch on VisDrone2019 and UAVDT for nine YOLO-family baselines and PD-YOLO. Within roughly the first 50 epochs, mAP@0.5 rises quickly for all models; YOLOv9t, YOLOv11n, and PD-YOLO exhibit larger slopes and relatively smooth curves and enter a performance plateau earlier. YOLOv6n fluctuates more, suggesting weaker optimization stability under the same training recipe. At the end of training, PD-YOLO reaches 37.0% mAP@0.5 on VisDrone, surpassing the next-best YOLOv10n, YOLOv9t, and YOLOv8n.

In complexity, PD-YOLO uses 2.65M parameters and 10.5 GFLOPs-higher than YOLOv11n (2.58M, 6.4 GFLOPs) and YOLOv8n (3.01M, 8.2 GFLOPs), lower than YOLOv9t's 10.7 GFLOPs while still achieving higher mAP@0.5, and higher accuracy than YOLOv6n (4.63M, 11.3 GFLOPs) at lower GFLOPs.

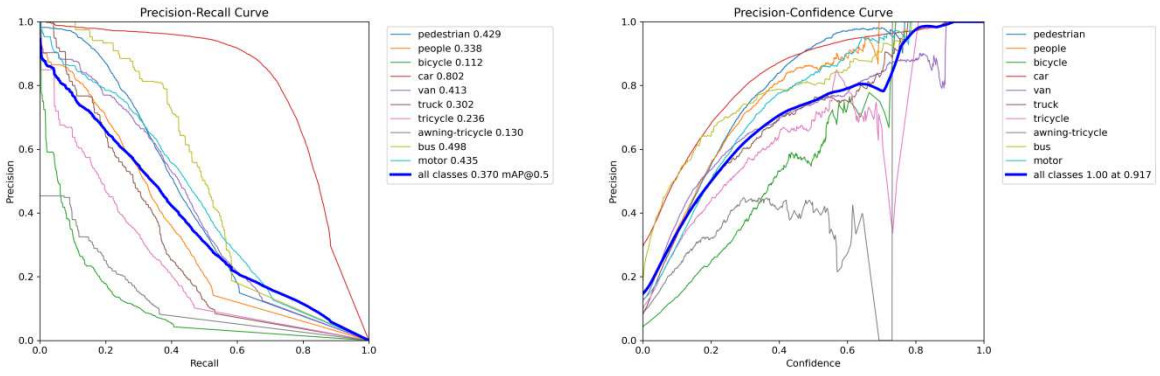
In summary, PD-YOLO remains leading at the end of the curves and achieves a parameter-compute-accuracy trade-off suited to our application scenario, appropriate for deployments that demand high accuracy on VisDrone-like aerial small-object detection.



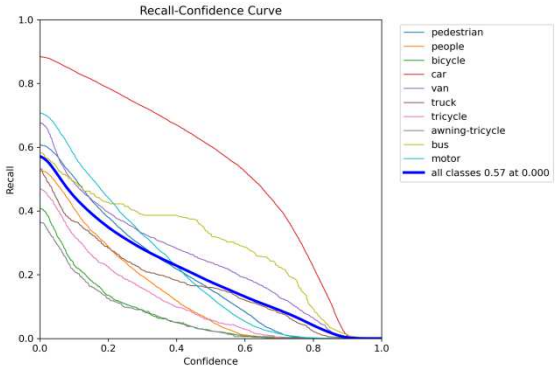
(a) VisDrone2019 (b) UAVDT

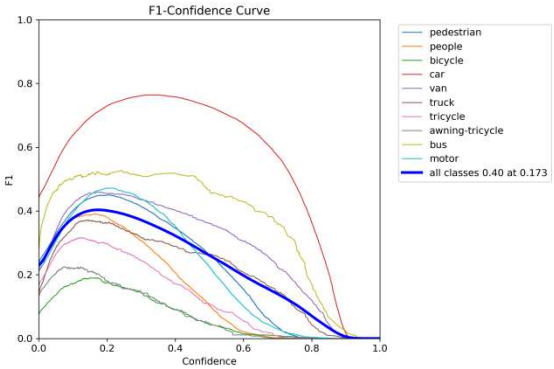
Figure 6 Training curves of mAP@0.5 for compared models

To further verify PD-YOLO's advantages and to assess detection performance qualitatively, Figure 7 plots multiple evaluation curves for the proposed PD-YOLO on the VisDrone2019 validation set.



(a) (b)





(c) (d)

Figure 7 Evaluation metrics of the PD-YOLO model (a) P-R curves for each category and overall. (b) Precision for each category and overall. (c) Recall rates for each category and overall. (d) F1-score

From the precision-recall curves: overall mAP@0.5 is about 0.37 with a full PR envelope; car reaches AP near 0.80 and keeps high precision even at high recall; pedestrian, bus, motor, van, and similar classes show PR shapes superior to many small or easily confused categories, reflecting PD-YOLO's advantage on aerial traffic and pedestrian-related targets. From precision versus confidence: as confidence increases, precision rises monotonically overall and approaches 1.0 in the high-confidence band, so high-confidence predictions are reliable-useful when raising thresholds to strictly control false alarms. From recall versus confidence: at low confidence, overall recall is high and dominant classes such as car maintain a recall edge across thresholds, indicating sufficient activation on major categories and reducing missed detections. From F1 versus confidence: overall F1 peaks near confidence 0.17 (around 0.40) with a clear peak location, which can serve as an initial detection threshold or for post-processing NMS, giving a practical precision-recall compromise. Taken together, the four panels show coherent per-class PR structure, confidence sensitivity, and F1 optima-strong performance on high-AP categories together with deployable threshold tunability.

4.3 Ablation experiments

To verify the proposed improvements, we conduct systematic ablations on VisDrone2019 and UAVDT. We use mAP@0.5 as the accuracy metric and report Parameters and FLOPs for complexity; detailed numbers appear in Table 5 and Table 6.

Table 5 Ablation study of PD-YOLO on VisDrone2019

Model	P2	C3k2_DWR	Parameters	GFLOPs	P(%)	R(%)	mAP@0.5
YOLO11n	-	-	2584102	6.4	44.4	33.2	33.1
YOLO11n-DWR	-	√	2575278	6.6	44.3	33.4	33.1
YOLO11n-P2	√	-	2671352	10.4	46.6	35.7	36.1
PD-YOLO	√	√	2654456	10.5	47.2	36.6	37.0

Table 6 Ablation study of PD-YOLO on UAVDT

Model	P2	C3k2_DWR	Parameters	GFLOPs	P(%)	R(%)	mAP@0.5
YOLO11n	-	-	2582737	6.4	94.4	93.7	97.9
YOLO11n-DWR	-	√	2565841	6.5	95.1	94.5	97.7

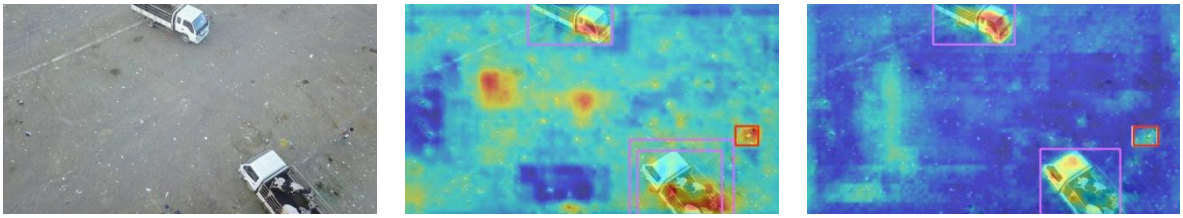
YOLO11n-P2	√	-	2670428	10.3	95.7	95.6	97.9
PD-YOLO	√	√	2653532	10.5	96.2	95.4	98.5

On VisDrone2019, adding only C3k2_DWR leaves mAP@0.5 equal to YOLO11n while precision and recall shift slightly and GFLOPs rise modestly to 6.6, indicating that backbone-only DWR under this protocol contributes limited mean improvement and mainly refines representations. Adding only the P2 four-scale neck (YOLO11n-P2) raises mAP@0.5 sharply to 36.1% with clear gains in P and R, confirming that a shallow high-resolution branch is critical for aerial small objects and dense instances at the cost of GFLOPs near 10.4. The full PD-YOLO, with slightly fewer parameters than YOLO11n-P2 (2.65M vs. 2.67M) and almost the same GFLOPs (10.5 vs. 10.4), further improves mAP@0.5 to 37.0% with P and R at 47.2% and 36.6%-a 3.9-point mAP@0.5 gain over YOLO11n-showing complementarity rather than naive stacking of P2 and DWR.

On UAVDT, DWR-only slightly reduces mAP@0.5 to 97.7%; P2-only restores 97.9% with higher P and R; PD-YOLO achieves the best mAP@0.5 and P in the table, indicating that the structure generalizes to cross-dataset vehicle-centric aerial scenes.

Jointly, the two tables suggest that P2 supplies the main increment for hard-to-detect targets, while DWR, on top of P2's expanded receptive field and scale layout, suppresses redundancy and sharpens boundaries so that PD-YOLO improves accuracy–complexity trade-offs versus P2 alone-supporting PD-YOLO as the final model of this paper.

Gradient-weighted class activation mapping (Grad-CAM) visualizes decision relevance by backpropagating gradients and fusing them into the final convolutional feature maps to form a heatmap correlated with class activation. We apply it to analyze PD-YOLO's decision behavior and randomly select one typical validation image from VisDrone2019 for side-by-side comparison (Figure 8).



(a) (b) (c)

Figure 8 Grad-CAM visualization results (a) original image; (b) YOLOv11n; (c) PD-YOLO

In the heatmaps, baseline YOLO11's overlay on the chosen layer shows diffuse activation and merged responses for distant tiny objects, with some minuscule instances nearly invisible; for large vehicles, high-response regions spill outward with unclear object-background boundaries, indicating insufficient shallow resolution and receptive field for fine localization. Under the same protocol, PD-YOLO yields nearly one peak per small target with concentrated responses and much weaker cross-target interference; for large vehicles, high activation hugs the body contour while background stays low. Overall, PD-YOLO's maps are spatially sharper and more scale-balanced, aligned with the motivation of adding a high-resolution shallow branch (e.g., P2) and DWR-style deformation modeling, and provide interpretability evidence for PD-YOLO's gains over YOLO11 on aerial small objects with multi-scale coexistence.

To further demonstrate superiority, we randomly sample typical validation images from VisDrone2019 and UAVDT, run YOLOv11n and PD-YOLO under the same environment, and compare detections in Figure 9.

Input Image YOLOv11n(baseline) PD-YOLO



(a)



(b)

Figure 9 Figure 9 Qualitative detection comparison (a) VisDrone2019; (b) UAVDT highlights clear differences. With YOLOv11n (middle column), obvious misses appear-for example, many vehicles on the right side of VisDrone2019 frames are not detected, and on UAVDT some vehicles on the road at night are also missed. PD-YOLO (right column) not only detects vehicles missed by YOLOv11n but is more accurate at night. Thus the model mitigates baseline misses and mis-detections, demonstrates superior performance, and corroborates the role of a shallow high-resolution branch for aerial small objects and dense instances.

Under fixed hardware and software and a unified training-evaluation protocol, this chapter presents horizontal comparisons between PD-YOLO and multiple YOLO-family baselines on VisDrone2019-DET as the primary benchmark and UAVDT as supplementary evaluation. Quantitatively, PD-YOLO exceeds all listed baselines in overall P, R, mAP@0.5, and per-class AP@0.5, with mAP@0.5 at 37.0% and stable margins over YOLOv10n, YOLOv11n, and others. Training curves of mAP@0.5 converge smoothly; PR, precision-confidence, recall-confidence, and F1-confidence plots are well behaved and support threshold selection for deployment. Ablations show that the P2 four-scale neck contributes the main accuracy gain, while C3k2_DWR further refines features at similar compute-the combination is consistent on VisDrone and UAVDT. Grad-CAM and similar visualizations align spatial attention with the quantitative gains. Together, these results indicate that PD-YOLO is suitable for VisDrone-like dense small-object aerial scenes.

5 Conclusion

This paper addresses large scale span in low-altitude remote sensing, the coexistence of small objects with dense backgrounds, and the loss of shallow detail through repeated downsampling. While preserving a single-stage YOLO11 paradigm that remains comparable and reproducible, we propose PD-YOLO, which couples a high-resolution P2 four-scale detection branch in the neck-head with C3k2_DWR dilated receptive-field enhancement in the backbone. Experiments show that the method achieves better overall and per-class detection accuracy than multiple YOLO-family baselines on the primary VisDrone2019 evaluation, and exhibits a consistent improvement trend on the supplementary vehicle-centric UAVDT benchmark; ablations further indicate that P2 contributes the main accuracy gain, whereas DWR refines features and acts in a complementary role at similar computational cost. At the same time, the improvements entail expected increases in FLOPs and parameter count; for extreme edge devices or ultra-low-latency deployments, pruning, knowledge distillation, or hardware-software co-design should be revisited. Current conclusions are mainly established under the adopted training protocol and data distribution; extension to other aerial benchmarks, different sensors, and seasonal conditions still requires additional validation. Future work may assess generalization on broader-area datasets, study lightweight designs and real-time trade-offs, and close the evaluation loop with downstream tasks such as tracking and counting.

Author Contributions: L.L. and R.Z. conceived the study and designed the methodology. L.L. developed the model, implemented the software, and conducted the formal analysis and investigation. C.G. and Y.Z. contributed to validation, data curation, and visualization. Z.S. contributed to methodological discussions and analysis. L.L. wrote the original draft. R.Z. supervised the project, acquired funding, and reviewed and edited the manuscript. All authors reviewed and approved the final manuscript.

Data Availability Statement: The VisDrone2019-DET and UAVDT datasets analysed during the cu rrent study are publicly available. VisDrone2019-DET can be accessed at: https://github.com/VisDro ne/VisDrone-Dataset. UAVDT can be accessed at: https://sites.google.com/view/grli-uavdt/. No new d atasets were generated.

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

Funding: The Research Foundation for Talented Scholars of Xijing University under Grant XJ19B01.

References

[1] Wu, X., Li, W., Hong, D. F. et al. Deep learning for UAV-based object detection and tracking: a survey. IEEE Geosci. Remote Sens. Mag. 10, 91-124; 10.1109/MGRS.2021.3115137 (2022).

[2] Tang, G. Y., Ni, J. J., Zhao, Y. H. et al. A survey of object detection for UAVs based on deep learning. Remote Sens. 16, 149; 10.3390/rs16010149 (2024).

[3] Wang, J., Xu, C., Yang, W. et al. A normalized Gaussian Wasserstein distance for tiny object detection. Preprint at https://arxiv.org/abs/2110.13389 (2021).

[4] Wang, J., Yang, W., Guo, H. W. et al. Tiny object detection in aerial images. In Proc. 25th Int. Conf. Pattern Recognit. 3791-3798; 10.1109/ICPR48806.2021.9413340 (2021).

[5] Du, D., Zhu, P., Wen, L. et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops 213-226; 10.1109/ICCVW.2019.00030 (2019).

[6] Zou, Z., Chen, K., Shi, Z. et al. Object detection in 20 years: A survey. Proc. IEEE 111, 257-276; 10.1109/JPROC.2023.3238524 (2023).

[7] Redmon, J. & Farhadi, A. YOLOv3: An incremental improvement. Preprint at https://arxiv.org/abs/1804.02767 (2018).

[8] Bochkovskiy, A., Wang, C. Y. & Liao, H. Y. M. YOLOv4: Optimal speed and accuracy of object detection. Preprint at https://arxiv.org/abs/2004.10934 (2020).

[9] Ge, Z., Liu, S., Wang, F. et al. YOLOX: Exceeding YOLO series in 2021. Preprint at https://arxiv.org/abs/2107.08430 (2021).

[10] Wang, C. Y., Bochkovskiy, A. & Liao, H. Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Preprint at https://arxiv.org/abs/2207.02696 (2022).

[11] Sapkota, R. & Karkee, M. Ultralytics YOLO evolution: An overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 object detectors for computer vision and pattern recognition. Preprint at https://arxiv.org/abs/2510.09653 (2025).

[12] Lin, T. Y. et al. Feature pyramid networks for object detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2117-2125; 10.1109/CVPR.2017.106 (2017).

[13] Jocher, G., Chaurasia, A. & Qiu, J. Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics (2023).

[14] Wang, C. Y., Yeh, I. H. & Liao, H. Y. M. YOLOv9: Learning what you want to learn using programmable gradient information. Preprint at https://arxiv.org/abs/2402.13616 (2024).

[15] Yaseen, M. What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. Preprint at https://arxiv.org/abs/2408.15857 (2024).

[16] Liu, S., Qi, L., Qin, H., Shi, J. & Jia, J. Path aggregation network for instance segmentation. In Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 8759-8768; 10.1109/CVPR.2018.00913 (2018).

[17] Wei, H., Liu, X., Xu, S. et al. DWRSeg: Rethinking efficient acquisition of multi-scale contextual information for real-time semantic segmentation. Preprint at https://arxiv.org/abs/2212.01173 (2022).

[18] Yu, F. & Koltun, V. Multi-scale context aggregation by dilated convolutions. Preprint at https://arxiv.org/abs/1511.07122 (2015).

[19] Tian, Z., Shen, C., Chen, H. & He, T. FCOS: Fully convolutional one-stage object detection. In Proc. IEEE/CVF Int. Conf. Comput. Vis. 9627-9636; 10.1109/ICCV.2019.00972 (2019).

[20] Zhang, S., Chi, C., Yao, Y., Lei, Z. & Li, S. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 9759-9778; 10.1109/CVPR42600.2020.00978 (2020).

[21] Carion, N. et al. End-to-end object detection with transformers. In Computer Vision-ECCV 2020 (eds Vedaldi, A., Bischof, H., Brox, T. & Frahm, J. M.) Lect. Notes Comput. Sci. vol. 12346, 213-229; 10.1007/978-3-030-58452-8_13 (Springer, Cham, 2020).

[22] Zhao, Y. et al. DETRs beat YOLOs on real-time object detection. In Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 16965-16974; 10.1109/CVPR52733.2024.01605 (2024).

[23] Feng, C., Zhong, Y., Gao, Y., Scott, M. R. & Huang, W. TOOD: Task-aligned one-stage object detection. In Proc. IEEE/CVF Int. Conf. Comput. Vis. 3510-3519; 10.1109/ICCV48922.2021.00310 (2021).

[24] Li, X. et al. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Preprint at https://arxiv.org/abs/2006.04388 (2020).

[25] Lin, T. Y. et al. Focal loss for dense object detection. In Proc. IEEE Int. Conf. Comput. Vis. 2980-2988; 10.1109/ICCV.2017.324 (2017).

[26] Tan, M., Pang, R. & Le, Q. EfficientDet: Scalable and efficient object detection. In Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 10781-10790; 10.1109/CVPR42600.2020.01079 (2020).

[27] Ghiasi, T., Lin, T. Y. & Le, Q. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 7036-7045; 10.1109/CVPR.2019.00720 (2019).

[28] Xia, G. et al. DOTA: A large-scale dataset for object detection in aerial images. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 3974-3983; 10.1109/CVPR.2018.00418 (2018).

[29] Ding, J. et al. Object detection in aerial images: A large-scale benchmark and challenges. IEEE Trans. Pattern Anal. Mach. Intell. 44, 7778-7795; 10.1109/TPAMI.2021.3117983 (2022).

[30] Xu, C., Wang, J., Yang, W. & Yu, L. Detecting tiny objects in aerial images: A normalized Wasserstein distance and a new benchmark. ISPRS J. Photogramm. Remote Sens. 190, 79-93; 10.1016/j.isprsjprs.2022.06.002 (2022).

[31] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 770-778; 10.1109/CVPR.2016.90 (2016).

[32] Wang, C. Y. et al. CSPNet: A new backbone that can enhance learning capability of CNN. In Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops 390-391; 10.48550/arXiv.1911.11929 (2020).