

P2LITE-YOLO: DETECTION-SCALE REALLOCATION FOR TINY OBJECT DETECTION IN UAV IMAGERY

KaiQi Mi

School of Automation and Software Engineering, Shanxi University, Taiyuan 030000, Shanxi, China.

Abstract: In UAV imagery, object scales are heavily biased towards the tiny region, yet general-purpose object detectors typically employ fixed multi-scale prediction configurations. There is a significant mismatch between the spatial detail loss caused by continuous downsampling and the prediction scale setup. To address this issue, this paper proposes P2Lite-YOLO, a scale-prior-driven prediction resource reallocation method for tiny UAV objects. First, the prediction requirements of different resolution features are explicitly characterized based on the object scale distribution in the training data. While keeping the number of detection heads unchanged, the original P3/P4/P5 prediction scales of YOLO11s are reallocated to P2/P3/P4, allowing the limited prediction capacity to prioritize high-resolution regions where tiny objects dominate. Second, a P2 collaborative representation is constructed via top-down deep semantic propagation and shallow spatial detail fusion, enabling high-resolution features to retain both fine-grained localization information and contextual discriminative ability. Simultaneously, a detection head capacity alignment strategy is adopted to mitigate the impact of structural adjustments on pre-trained parameter transfer. Statistics from the VisDrone2019 training set show that 68.30% of the 343,205 valid objects have an equivalent size of less than 16 pixels, confirming the necessity of allocating high-resolution prediction resources. Under a unified experimental protocol, P2Lite-YOLO achieves 43.11% mAP50 and 26.38% mAP50-95 with 9.129 M parameters, outperforming YOLO11s by 2.88 and 2.47 percentage points, respectively, and surpassing comparative models such as YOLO11m, YOLOv8m, Faster R-CNN, and RetinaNet. Further experiments indicate that retaining the independent P5 prediction branch after introducing the P2 high-resolution prediction does not yield performance gains. These results demonstrate that, compared to simply expanding the detection scales, reconfiguring multi-scale representation and prediction resources under a fixed prediction budget based on data scale priors is an effective strategy for improving tiny object detection in UAV imagery.

Keywords: UAV imagery; Tiny object detection; Scale prior; Resolution-semantic collaboration; Prediction-capacity reallocation

1 INTRODUCTION

UAV platforms offer advantages such as high maneuverability, wide field of view, and low deployment cost, making aerial object detection valuable for traffic monitoring, urban governance, emergency search and rescue, and public safety [1-2]. However, flight altitude, overhead perspective, and imaging distance cause pedestrians, vehicles, and non-motorized vehicles to occupy only a few pixels in the image; dense distribution, mutual occlusion, motion blur, and complex backgrounds further degrade the separability between objects and their environment. For such scenarios, detection performance depends not only on deep semantic representations but also on whether the network can retain sufficient spatial detail after continuous downsampling.

Deep learning detectors are usually divided into two-stage and one-stage families. Two-stage methods such as Faster R-CNN first propose candidate regions and then classify and regress them, which gives strong localization [3]. The YOLO family, in contrast, casts detection as a single-stage dense prediction task, which is fast and convenient to deploy [4]. For scale variation, FPN injects deep semantics into high-resolution features through a top-down path with lateral connections [5], and PANet shortens the route by which shallow localization cues travel to deeper layers [6]. As YOLO architectures evolved, CSPNet improved gradient flow and efficiency with cross-stage partial connections [7], YOLOX brought in an anchor-free decoupled head and SimOTA label assignment [8], YOLOv8 kept the anchor-free decoupled paradigm [9], and YOLO11 stacked C3k2, SPPF, and C2PSA on an FPN-PAN neck while still predicting at three scales, P3/P4/P5 [10]. These designs were mostly tuned for general scenes, though, and do not necessarily suit UAV data where small objects dominate.

Earlier work has tried to improve UAV small-object detection through higher-resolution prediction, stronger features, or better label assignment. TPH-YOLOv5 adds a fourth detection head and uses a Transformer head to capture global context [11]. NWD treats bounding boxes as 2D Gaussian distributions so that IoU is less sensitive to tiny positional shifts [12]. Zhao et al. improved VisDrone results by reducing backbone downsampling, enlarging the detection feature maps, and adding context together with spatial-channel enhancement modules [13]. What these works share is that high-resolution representations and more stable sample matching help small objects. But the extras—new heads, attention modules, or assignment schemes—also make the network more complex, and their benefits over long training runs are not always reliable.

Existing studies typically treat scale coverage as an inherent property of a general architecture, with less attention paid to whether the prediction resource configuration aligns with the specific dataset's scale prior. Our statistical analysis of

the VisDrone training set reveals that, under a unified imaging scale, the median equivalent size of the 343,205 valid objects is only 11.09 pixels, with 68.30% smaller than 16 pixels. This implies that a substantial number of instances retain only minimal spatial responses after continuous downsampling, while low-resolution prediction representations aimed at large objects may have low marginal value. The research question that arises is: under constrained prediction budget, how can spatial resolution and semantic abstraction be coordinated based on the object scale distribution, so that limited capacity preferentially serves the dominant tiny objects?

To address this question, we propose P2Lite-YOLO. Its core is not to add detection branches, but to reorganize multi-scale representations and prediction resources using the scale distribution as a prior: on one hand, it constructs a collaborative representation that preserves both fine-grained localization cues and deep context through cross-layer information propagation; on the other hand, while keeping the number of detection heads unchanged, it shifts part of the capacity from the low-resolution end to the high-resolution end. The method also constrains the capacity changes of the detection branches to reduce the disturbance of structural adjustments on the transfer of pre-trained knowledge.

The contributions of this paper are threefold: (1) Starting from the empirical scale distribution, we formulate UAV tiny object detection as a problem of representation and capacity allocation under a fixed prediction budget; (2) We propose a resolution-semantic collaborative representation and a scale-prior-constrained prediction resource reallocation mechanism, enhancing the discriminability of tiny objects without increasing the number of prediction heads; (3) We construct a unified comparison protocol covering YOLO11s, YOLO11m, YOLOv8m, Faster R-CNN, and RetinaNet, and demonstrate the effectiveness of our method through comparative experiments.

2 METHODS

2.1 Problem Formulation and Overall Framework

Multi-scale detection can be viewed as allocating limited prediction capacity between spatial resolution and semantic abstraction. For an input image I , the backbone produces hierarchical features, and the feature pyramid integrates them into fused representations. As the layer depth increases, the spatial resolution of the representation decreases while the semantic receptive field expands. The generic YOLO11s selects three mid-to-low resolution layers for dense prediction, with the representation set given by:

$$\mathcal{F}_{\text{base}} = \{F_3, F_4, F_5\} \quad (1)$$

Let the equivalent scale of training objects follow an empirical distribution $p(d)$, with a fixed prediction head budget K . For a fixed K , layer selection should simultaneously satisfy two requirements: the high-probability scale interval obtains sufficient spatial sampling, and the corresponding representation retains the context necessary for category discrimination. Accordingly, P2Lite-YOLO no longer treats the prediction layers as immutable architectural constants but as capacity allocation variables constrained by the data's scale prior. The model retains the deep semantic propagation path and introduces a finer-grained collaborative representation, transforming the final prediction set to:

$$\mathcal{F}_{\text{P2Lite}} = \{F_2, F_3, F_4\} \quad (2)$$

Equations (1) and (2) represent a transfer of prediction resources under a fixed budget, rather than an expansion of detection heads. When mapped to YOLO11s specifically, the new collaborative representation corresponds to the higher-resolution P2 layer, while the original deep P5 features still participate in top-down semantic propagation but no longer independently undertake classification and regression; the three prediction units are adjusted from P3/P4/P5 to P2/P3/P4. Figure 1 summarizes the framework from the three perspectives of data prior, collaborative representation, and resource allocation.

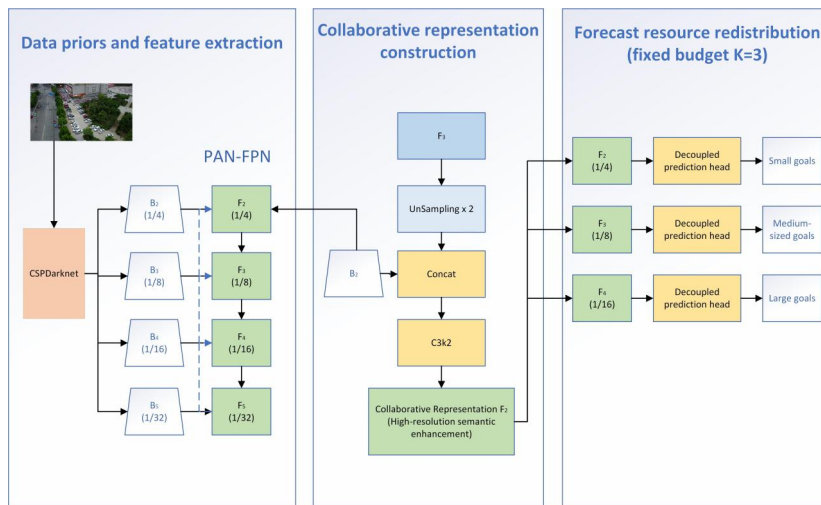


Figure 1 Framework of Scale-Distribution-Driven Collaborative Representation and Prediction Resource Reallocation

2.2 Resolution-Semantic Collaborative Representation

Tiny object representation faces two complementary deficiencies. Shallow features preserve edges, center positions, and local textures but are prone to misclassifying road textures, building boundaries, and shadows as objects. Deep features possess larger receptive fields and more stable category semantics, but may dissolve tiny instances during continuous downsampling [5]. Therefore, a high-resolution branch can serve both precise localization and reliable classification only after receiving deep contextual constraints. This paper constructs a collaborative representation through top-down semantic propagation and lateral spatial detail injection, implemented as:

$$F_2 = \Phi_{P2} (\text{Concat} [\text{Up}_{2\times}(F_3), B_2]) \quad (3)$$

Here, Up is the scale-alignment operation, Concat merges information along the channel dimension, and Φ_{P2} stands for cross-layer feature integration. In practice, the semantic representation from the deeper fusion layer is upsampled by $2\times$ and concatenated with the high-resolution spatial features from the backbone, after which $C3k2$ jointly encodes the local and contextual information [10]. The resulting F_2 is not simply an enlarged copy of the shallow features; it is a fine-grained representation modulated by deep semantics. This lets later predictions sample more spatial locations while suppressing high-frequency responses that carry no categorical meaning.

2.3 Scale-Prior-Constrained Prediction Resource Reallocation

Adding a high-resolution branch on its own would enlarge both the number of heads and the optimization space, making it hard to tell whether any gains come from better scale matching or just from more capacity [11]. P2Lite-YOLO instead sticks to a fixed-budget rule: the total number of prediction units stays at $K=3$, and one low-resolution unit is moved to the finer collaborative representation according to $p(d)$. Let H_i be the decoupled prediction operator at the i -th layer; the model output is:

$$\mathcal{Y} = \bigcup_{i \in \{2,3,4\}} \mathcal{H}_i(F_i) \quad (4)$$

Each H_i emits a category score and a distributed bounding-box regression. This only changes where the prediction resources sit, not how deep the semantic encoder is: low-resolution features still feed the fine-grained representation through the pyramid, and only their independent prediction duty is handed off. For a 640×640 input, the four-scale scheme has 34,000 spatial prediction locations, while the fixed-budget three-scale scheme has 33,600, and in both cases the high-resolution layers dominate the computation.

2.4 Transfer Consistency and Optimization Constraints

Adjusting the prediction layers alters the input dimensions of the detection heads. If the internal width is recalculated according to the channel count of the newly added high-resolution branch, the parameter shapes of multiple classification and regression branches change simultaneously, causing large portions of pre-trained knowledge to be discarded due to tensor incompatibility. To maintain parameter space continuity before and after structural adjustment, this paper does not determine the detection head capacity using the original channel count of the new layer, but instead uses the channel count C_3 of the original middle scale as a common reference. Let N_c be the number of categories and $R=16$ the number of discrete units for distribution regression. The hidden channels for the regression and classification branches are respectively defined as:

$$C_{\text{reg}} = \max \left(16, \left\lceil \frac{C_3}{4} \right\rceil, 4R \right) \quad (5)$$

$$C_{\text{cls}} = \max (C_3, \min (N_c, 100)) \quad (6)$$

This capacity constraint maintains internal dimensions close to the baseline for the original mid- and low-resolution branches and provides shape-compatible initialization for the newly added high-resolution branch. When transferring from the COCO 80-class pre-trained model to the VisDrone 10-class task, 517 out of 523 structurally compatible tensors are successfully loaded; the non-transferred items correspond only to classification output weights and biases that change due to the category space variation.

To control methodological variables, P2Lite-YOLO does not alter the learning objectives of YOLO11s, continuing to use bounding box regression, classification, and Distribution Focal Loss (DFL) [10, 14]. The total objective function is:

$$\mathcal{L} = \lambda_{\text{box}} \mathcal{L}_{\text{box}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{dfl}} \mathcal{L}_{\text{dfl}} \quad (7)$$

where $\mathcal{L}_{\text{box}}$, $\mathcal{L}_{\text{cls}}$, and $\mathcal{L}_{\text{dfl}}$ denote the bounding box regression loss, classification loss, and DFL, respectively, with weights λ_{box} , λ_{cls} , and λ_{dfl} set to 7.5, 0.5, and 1.5. The model introduces no additional inference branches or post-processing operations, so the primary structural variables in the experiments are limited to the collaborative representation and prediction resource configuration.

3 EXPERIMENTS AND RESULTS

3.1 Dataset and Scale Distribution

This paper evaluates on the VisDrone2019-DET dataset [1]. It contains UAV images captured under various cities, flight altitudes, and weather conditions, annotated with 10 object categories: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. The training, validation, and test-dev sets contain 6,471, 548, and 1,610 images, respectively. We train on the training set and report results on the publicly annotated validation set; annotations are converted to YOLO format while ignoring instances marked as ignored regions in the official labels, resulting in 343,205 valid training objects.

To unify object scales under different original image sizes, images are resized to 640 pixels along the longer edge, with object boxes scaled accordingly. Let the original image width and height be W and H , and the object box width and height be w and h . The scaling factor and the object's equivalent size are respectively:

$$r = \frac{640}{\max(W, H)} \quad (8)$$

$$d = \sqrt{(rw)(rh)} = r\sqrt{wh} \quad (9)$$

The equivalent size d , described as the side length of a square with the same area as the object box, mitigates the influence of aspect ratio on single-side statistics. Table 1 shows that the mean and median equivalent sizes of training objects are 15.26 and 11.09 pixels, respectively, with 68.30% of objects smaller than 16 pixels and 90.55% smaller than 32 pixels. This distribution provides data-driven support for introducing stride-4 features.

Table 1 Object Scale Statistics on the VisDrone Training Set

Metric	Value
Number of Valid Objects	343 205
Mean Equivalent Size	15.26 px
Median Equivalent Size	11.09 px
Equivalent Size < 16 px	68.30%
Equivalent Size < 32 px	90.55%
Equivalent Size < 64 px	98.66%

3.2 Comparative Models and Unified Evaluation Protocol

Comparative objects are implemented using their official names and reproducible implementations. YOLO11s, YOLO11m, and YOLOv8m use Ultralytics' yolo11s.pt, yolo11m.pt, and yolov8m.pt, respectively [9-10]. Faster R-CNN and RetinaNet both employ ResNet-50-FPN, with the former being a two-stage implementation and the latter using Focal Loss [3, 15]. P2Lite-YOLO is based on YOLO11s, implementing the proposed collaborative representation and resource reallocation mechanism. Table 2 defines the models.

Table 2 Comparative Model Definitions

Model	Detection Paradigm	Implementation or Backbone	Pre-training	Unified Input
YOLO11s	One-stage	Ultralytics yolo11s	COCO	640×640
YOLO11m	One-stage	Ultralytics yolo11m	COCO	640×640
Faster R-CNN	Two-stage	ResNet-50-FPN	COCO	640×640
RetinaNet	One-stage	ResNet-50-FPN	COCO	640×640
YOLOv8m	One-stage	Ultralytics yolov8m	COCO	640×640
P2Lite-YOLO	One-stage	YOLO11s + P2Lite	COCO	640×640

The unified comparison protocol stipulates that all models use the same VisDrone training and validation splits, 640×640 letterbox input, COCO pre-trained initialization, and 50 training epochs; the effective batch size is uniformly set to 16. To reduce confounding from framework-specific augmentations, the main comparison retains only random horizontal flipping, scale perturbation, and color perturbation, without using Mosaic; Faster R-CNN and RetinaNet have both their shortest and longest sides fixed to 640. All models use the same annotation conversion script and COCO evaluation program.

The optimizer is uniformly SGD with an initial learning rate of 0.01, momentum of 0.9, weight decay of 0.0005, and a 3-epoch warmup followed by cosine decay. Training employs automatic mixed precision and deterministic mode. Training and testing are uniformly conducted on an NVIDIA GeForce RTX 4090 with PyTorch 2.0.0 and CUDA 11.8. Evaluation metrics include mAP50 and mAP50-95. mAP50 is the mean of per-category average precisions at an IoU threshold of 0.5; mAP50-95 averages over multiple IoU thresholds from 0.50 to 0.95 with a step of 0.05.

3.3 Comparative Experiments

Table 3 presents the validation results at epoch 50 under the unified protocol for all comparative models. P2Lite-YOLO achieves 43.11% mAP50 and 26.38% mAP50-95 with 9.129 M parameters, ranking first among all six models. Compared to the baseline YOLO11s (40.23% mAP50, 23.91% mAP50-95), P2Lite-YOLO improves mAP50 and mAP50-95 by 2.88 and 2.47 percentage points, respectively, indicating that reallocating detection scales from P3/P4/P5 to P2/P3/P4 with similar parameter counts effectively improves detection accuracy in tiny-object-dominated scenes. The simultaneous improvements in both mAP metrics further suggest that the proposed structure enhances not only the detection capability at IoU=0.5 but also the comprehensive localization performance under stricter thresholds.

Table 3 Cross-Model Comparison under the Unified Protocol

Model	Detection Paradigm	Reference Param. (M)	mAP50(%)	mAP50-95(%)
YOLO11s	One-stage	9.4	40.23	23.91
YOLO11m	One-stage	20.1	42.82	25.81
Faster R-CNN (R50-FPN)	Two-stage	41.8	41.32	25.03

Model	Detection Paradigm	Reference Param. (M)	mAP50(%)	mAP50-95(%)
RetinaNet (R50-FPN)	One-stage	34.0	39.93	23.67
YOLOv8m	One-stage	25.9	41.82	24.92
P2Lite-YOLO	One-stage	9.129	43.11	26.38

Table 4 provides per-category AP for the independently re-evaluated best weights of P2Lite-YOLO. Cars and buses achieve AP50 of 82.533% and 58.859%, respectively, while bicycles and awning-tricycles reach only 16.058% and 15.818%. These latter categories are generally smaller in area, more similar in appearance, and more susceptible to occlusion by vehicles and pedestrians. High-resolution features can alleviate downsampling information loss but cannot simultaneously eliminate difficulties posed by category imbalance, occlusion, and annotation scale.

Table 4 Per-Category Detection Results of P2Lite-YOLO with Best Weights

Category	AP50(%)	AP50-95(%)
pedestrian	48.025	22.815
people	39.504	15.900
bicycle	16.058	7.029
car	82.533	58.843
van	44.838	32.207
truck	40.232	26.959
tricycle	27.450	15.931
awning-tricycle	15.818	10.080
bus	58.859	42.075
motor	48.879	22.253

3.4 Ablation Experiments

To check what the P5 head is actually worth, we build a P2/P3/P4/P5 four-head model that is otherwise identical to P2Lite-YOLO and compare it with the P2/P3/P4 three-head model during training; Table 5 lists the numbers. The three-head setup reaches 43.11% mAP50 and 26.38% mAP50-95, while the four-head setup gets 42.05% and 25.17%, leaving the three-head model ahead by 1.06 and 1.21 points. In other words, once the stride-4 P2 collaborative representation is in place and the P5 prediction resources are moved to P2, keeping an independent P5 branch does not pay off; if anything, it adds optimization burden and spreads prediction resources needlessly across four scales. On the efficiency side, the P5 head carries its own classification and regression parameters, so dropping it trims parameters and compute and leaves a cleaner configuration at essentially no accuracy cost.

Table 5 Comparison with and without the P5 Prediction Branch

Prediction Scales	Number of Heads	mAP50(%)	mAP50-95(%)	Selection
P2/P3/P4/P5	4	42.05	25.17	Not adopted
P2/P3/P4	3	43.11	26.38	Adopted

4 CONCLUSION

To address the issue of an extremely high proportion of tiny objects in UAV imagery and the insufficient default prediction scale configuration of generic detectors, this paper proposes the P2Lite-YOLO detection model. By statistically analyzing object scales on the VisDrone training set, we find that 68.30% of objects have equivalent sizes smaller than 16 pixels. Accordingly, we construct a stride-4 high-resolution P2 collaborative representation branch, fusing shallow spatial details from the backbone with deep semantic information from the neck. We also adopt a P5 to P2 detection scale reallocation strategy, adjusting the P3/P4/P5 prediction configuration to P2/P3/P4 while keeping the number of prediction heads unchanged, retaining the P5 deep semantic propagation path but removing its low-resolution prediction branch, thereby avoiding parameter and computational redundancy from simply adding a fourth detection head. The pre-training-compatible detection heads ensure a high reuse rate of 517/523 YOLO11s pre-trained parameters by using the P3 channel count as the capacity reference. Under the unified protocol, P2Lite-YOLO achieves 43.11% mAP50 and 26.38% mAP50-95 with 9.129 M parameters, ranking first among all six comparative models, surpassing the same-scale YOLO11s by 2.88 and 2.47 percentage points, respectively, and exceeding larger models such as YOLO11m and YOLOv8m with less than half their parameters. Ablation experiments demonstrate that the four-head configuration retaining P5 prediction drops by 1.06 and 1.21 percentage points in mAP50 and mAP50-95, respectively, validating the effectiveness of the P5 to P2 prediction resource reallocation. Current experiments are primarily based on single-run results on the VisDrone validation set and do not yet claim statistical significance across datasets. Future work can validate the model's generalization ability on other UAV datasets such as UAVDT, investigate lower-computation P2 feature generation methods, and further reduce the deployment pressure of high-resolution predictions on edge devices through model pruning, quantization, and knowledge distillation.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

- [1] Zhu P, Wen L, Du D, et al. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 44(11): 7380-7399. DOI: 10.1109/TPAMI.2021.3119563.
- [2] Tang G, Ni J, Zhao Y, et al. A survey of object detection for UAVs based on deep learning. *Remote Sensing*, 2024, 16(1): 149. DOI: 10.3390/rs16010149.
- [3] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(6): 1137-1149. DOI: 10.1109/TPAMI.2016.2577031.
- [4] Wang C Y, Liao H Y M. YOLOv1 to YOLOv10: The fastest and most accurate real-time object detection systems. *APSIPA Transactions on Signal and Information Processing*, 2024, 13(1): 1-38. DOI: 10.1561/116.20240058.
- [5] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017: 2117-2125.
- [6] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018: 8759-8768.
- [7] Wang C Y, Liao H Y M, Wu Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020: 390-391.
- [8] Ge Z, Liu S, Wang F, et al. YOLOX: Exceeding YOLO series in 2021. arXiv:2107.08430, 2021.
- [9] Terven J, Córdova-Esparza D M, Romero-González J A. A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 2023, 5(4): 1680-1716. DOI: 10.3390/make5040083.
- [10] Khanam R, Hussain M. YOLOv11: An overview of the key architectural enhancements. arXiv:2410.17725, 2024.
- [11] Zhu X, Lyu S, Wang X, et al. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2021: 2778-2788.
- [12] Xu C, Wang J, Yang W, et al. Detecting tiny objects in aerial images: A normalized Wasserstein distance and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2022, 190: 79-93. DOI: 10.1016/j.isprsjprs.2022.06.002.
- [13] Zhao J D, Zhen G Y, Chu C Q. Unmanned aerial vehicle image target detection algorithm based on YOLOv8. *Computer Engineering*, 2024, 50(4): 113-120. DOI: 10.19678/j.issn.1000-3428.0068268.
- [14] Li X, Wang W, Wu L, et al. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 2020, 33: 21002-21012.
- [15] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2017: 2980-2988.