Research on Engineering Structures & Materials
www.jresm.org

Research Article

High-Frequency detail injection and normalized Wasserstein distance for small object detection in UAV remote sensing using YOLOv8

Qifeng Han [*, a], Mingmin Liu [b], Xiaomin Han [c], Yuying Ma [d]

School of Electronic Engineering, Shandong University of Engineering and Vocational Technology, Jinan, China

Article Info

Article History:

Received 20 July 2026

Accepted 15 Sep 2026

Keywords:

UAV remote sensing;
Small object detection;
YOLOv8;
Normalized Wasserstein distance;
High-Frequency detail enhancement

Abstract

To address feature attenuation, scale imbalance, and background interference in UAV remote sensing images, an enhanced YOLOv8 framework is proposed. A high-frequency detail enhancement (HF) module preserves shallow texture information, adaptive feature fusion (AFF) improves cross-scale representation, and global context attention (GCA) strengthens semantic consistency. Normalized Wasserstein Distance (NWD) is introduced to improve small-object localization stability. On the VisDrone-DET2021 validation set, mAP@0.5 increases from 76.2% to 81.6%, while mAP-S increases from 48.7% to 54.5%. The model contains 15.3 M parameters and requires 38.9 GFLOPs, representing increases of 36.6% and approximately 36.0%, respectively, relative to YOLOv8s. It maintains an inference speed of 88.2 FPS at an input resolution of 640×640.

© 2026 MIM Research Group. All rights reserved.

1. Introduction

In recent years, computer vision has been widely used as an important tool to analyze UAV images [1-3]. Among them, YOLO series of single-stage detection frameworks have been widely used in aerial target recognition for their high end-to-end reasoning efficiency and usability [4-5]. However, the features of the UAV imagery — for example, the size of the target, the low effective pixel ratio, and the strong background interference-pose a significant challenge to the general detection network. Therefore, the performance of small object detection has been a major bottleneck in the implementation of the algorithm.

To address these problems, recent studies have focused on improving the YOLOv8 architecture [6-8]. Xu et al. [9] adapted YOLOv8n to UAV scenarios through lightweight structural optimization, improving small-object inference efficiency. Ni et al. [10] enhanced multiscale feature fusion in YOLOv8s, although high-level semantics may still suppress fine spatial details. Feng et al. [11] introduced multi-branch feature aggregation to enlarge the effective receptive field, but the additional branches increase computational complexity. Yuan et al. [12] designed an intra-group multiscale fusion attention mechanism to mitigate scale variation, whereas its localized attention provides limited global semantic modeling. Khalili et al. [13] proposed SOD-YOLOv8 for aerial object detection; however, IoU-based regression remains sensitive to extremely small and weakly overlapping targets.

Recent studies have investigated multiscale feature fusion, attention mechanisms, and lightweight network design [14-18]. However, fine-grained detail preservation, global semantic consistency, and scale-sensitive regression remain challenging for small objects [19-22]. These limitations

*Corresponding author: qifeng_han85@hotmail.com
[a]orcid.org/0009-0001-4905-2413; [b]orcid.org/0009-0007-2769-569X; [c]orcid.org/0009-0003-5254-2987; [d]orcid.org/0009-0007-7421-932X
DOI: http://dx.doi.org/10.17515/resm2026-1923vk0720rs

motivate the joint design of feature enhancement, context modeling, and regression constraints within YOLOv8.

This study improves YOLOv8 for UAV small-object detection through high-frequency detail enhancement, adaptive multiscale feature fusion, global context attention, and NWD-based localization regression. The resulting framework targets higher detection accuracy and localization stability while retaining real-time inference capability.

2. YOLOv8 Network Architecture and Small Object Detection Bottleneck Analysis

YOLOv8 adopts a three-stage architecture comprising a Backbone, Neck, and Head, consistent with mainstream single-stage detection frameworks [23-25]. The Backbone is the core feature extractor and uses stacked convolutional and cross-stage partial structures to progressively reduce spatial resolution while increasing semantic abstraction. This hierarchy produces feature maps at 1/8, 1/16, and 1/32 of the input resolution, corresponding to different receptive fields and semantic depths. The Neck integrates PAN-FPN to propagate features bidirectionally across scales. The FPN path transfers semantic information from high-level to high-resolution features, whereas the PAN path strengthens bottom-up transmission of spatial details. This multiscale fusion combines contextual and spatial information for objects of different sizes. The Head uses a decoupled structure, which means that classification and regression are carried out by separate branches. This architectural design reduces interference between classification and regression tasks. Figure 1 illustrates the overall structure and the function output hierarchy.

2.1 YOLOv8 Faces Three Principal Limitations in UAV Small-Object Detection

During progressive down sampling in the Backbone, small objects—defined as targets below 32×32 pixels under the COCO evaluation protocol and often below 20×20 pixels in high-altitude UAV imagery—can be compressed into only a few feature-map cells. On the other hand, while PAN-FPN is efficient at transmitting semantic and spatial cues, it suffers from semantic dominance: high-level features tend to overwhelm or mask low-level details during fusion. This results in the loss of edge clarity, boundary definition, and local contrast, which are critical for accurate recognition of small objects in a cluttered or textured background.

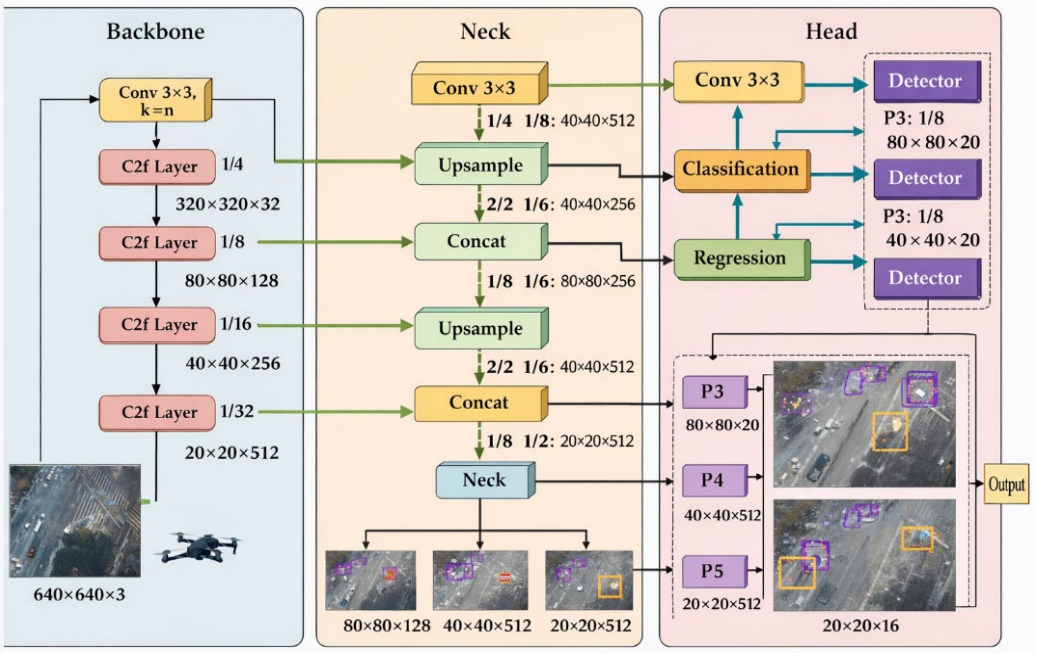


Fig. 1. Schematic of YOLOv8's overall network architecture and multi-scale feature outputs

Although the decoupled detection Head separates classification and regression, small-object sample assignment and bounding-box regression remain difficult. IoU-based regression becomes unstable when the overlap between predicted and ground-truth boxes is limited; negligible or zero overlap provides weak or discontinuous gradient feedback, reducing localization stability during

early training. Dense object distributions, partial occlusion, and severe scale imbalance further destabilize the regression branch in UAV remote sensing scenes. Consequently, predicted boxes are more likely to be missed or displaced during early optimization or under domain shift. Taken together, these structural characteristics reveal a trade-off between global semantic abstraction and preservation of local spatial detail. For high-resolution UAV inputs with dense layouts and severe scale imbalance, this trade-off limits detection robustness and recall. It motivates the proposed combination of feature refinement, context-aware attention, and scale-sensitive regression. The implementation uses the standard YOLOv8 detection scales: P3 at 1/8 resolution (80×80 for a 640×640 input), P4 at 1/16 resolution (40×40), and P5 at 1/32 resolution (20×20). These scales are used throughout the experiments.

3. Design of an Improved YOLOv8 Method for Small Object Detection in UAV Remote Sensing

3.1 Overview of the Overall Framework

As illustrated in Figure 2, the improved YOLOv8 framework retains the Backbone-Neck-Head architecture and integrates detail enhancement, multiscale fusion, global context modeling, and NWD-based regression. Input images are resized to 640×640. A high-frequency branch in the backbone preserves shallow texture responses, and the Neck adaptively fuses features across scales. GCA units are placed before the decoupled detection heads to recalibrate contextual features. The regression branch uses NWD to provide continuous geometric supervision for small objects.

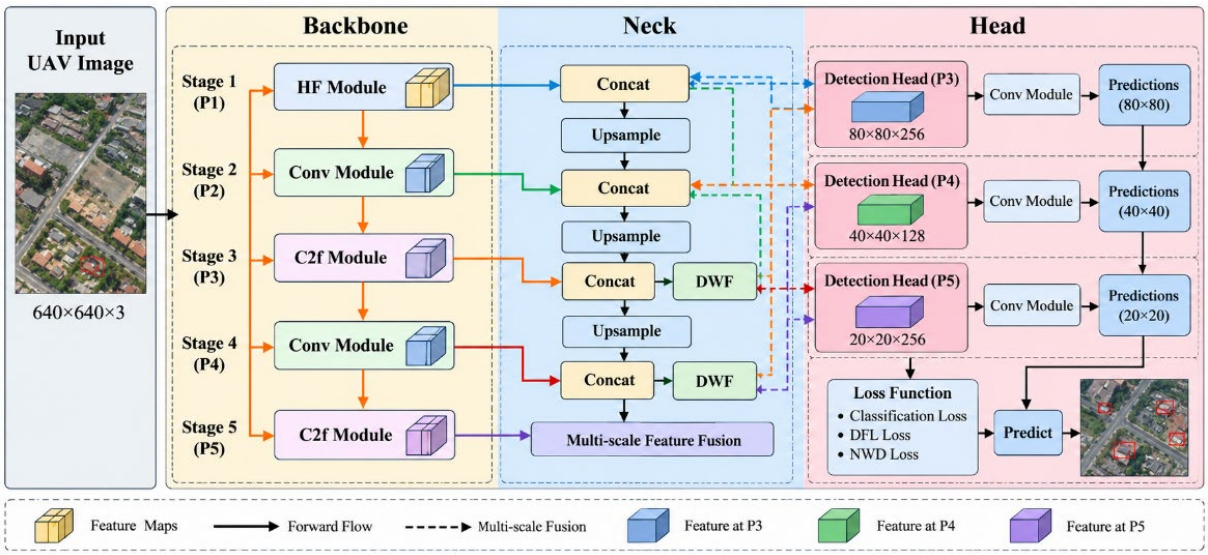


Fig. 2. Schematic of the improved YOLOv8 drone remote sensing small object detection framework

In Figure 2, DWF denotes the dynamic weighting operation within the adaptive feature fusion (AFF) module.

3.2 Detail-Enhanced Feature Pyramid Network

3.2.1 High-Frequency Detail Injection Branch

The High-Frequency Detail Injection Branch explicitly models edges, corners, and local contrast information—features prone to degradation during subsampling in YOLOv8's early-stage backbone features. It feeds these details into the feature pyramid via a "parallel injection + residual constraint" approach, preventing fine-grained textures from being obscured by higher-level semantics during PAN-FPN fusion. Specifically, lightweight high-pass response units are introduced at the shallow P2/P3 layers of the backbone. Local smoothing is first approximated via separable convolutions, followed by high-frequency residuals formed through differencing and amplitude normalization to generate detail feature maps:

$$H = Norm(F - DWConv(F; k_s)) \tag{1}$$

where F represents shallow features, $DWConv(F; k_s)$ denotes a separable convolution with kernel size k_s for low-frequency estimation, and $Norm$ performs channel normalization to suppress response scale drift caused by varying aerial illumination and ground material properties. To prevent high-frequency noise from being amplified indiscriminately, the branch further introduces a gated injection coefficient for adaptive adjustment of channel importance:

$$\tilde{F} = F + \alpha \odot H, \quad \alpha = \sigma(MLP(GAP(F))) \tag{2}$$

where α denotes the channel gating vector, σ represents the Sigmoid function, GAP denotes global average pooling, and $\odot$ indicates per-channel multiplication. This residual injection ensures the core semantic representation remains intact while providing a controllable source of detail enhancement for "adaptive weighted fusion." This enables each pyramid layer to distinguish between "true details" and "high-frequency interference" during fusion.

3.2.2 Adaptive Weighted Fusion

After high-frequency detail injection, features from different pyramid levels still exhibit differences in semantic depth, spatial resolution, and response intensity. Direct equal-weight fusion may cause high-level semantics to suppress shallow details, reducing the representation capability for small objects. Therefore, an adaptive weighted fusion (AFF) module is introduced to dynamically regulate the contribution of multi-scale features.

Let the adjacent input features beF_l, F_m, and F_h, representing different spatial and semantic levels. Channel dimensions are first unified using 1×1 convolution. Higher-level features are then aligned to the target resolution through nearest-neighbor interpolation, while lower-level features are down sampled using stride-2 convolution. The aligned features are denoted as $F_i^{'}$. Global average pooling is subsequently applied to obtain scale-specific descriptors:

$$z_i = GAP(F_i'), \quad i \in \{l, m, h\} \tag{3}$$

The descriptors are concatenated and processed by a lightweight MLP. Softmax normalization generates adaptive fusion weights:

$$[w_l, w_m, w_h] = Soft\,max(\,MLP([z_l, z_m, z_h])) \tag{4}$$

The final fused feature is expressed as:

$$F_{out} = w_l F_l' + w_m F_m' + w_h F_h' \tag{5}$$

where w_l, w_m and $w_h$denote the normalized contributions of different feature levels. The adaptive weighting mechanism preserves high-resolution structural details while maintaining semantic consistency, thereby reducing feature dominance during multi-scale fusion.

3.3 Lightweight Global Context Attention

3.3.1 Module Design

However, after the detail enhancement pyramid is used to aggregate the local structure information efficiently, the fusion features are still vulnerable to the effects of terrain texture and scale aliasing. This results in a similar response distribution in high level feature spaces. In order to solve this problem, a lightweight global context attention module is proposed. Based on the model of spatial dependencies and channel semantic associations, the proposed method imposes a global consistency constraint and performs semantic re-calibration on candidate regions. Specifically, let the input feature from the feature pyramid output be $F \in R^{C\times H\times W}$. First, a global context description vector is constructed via spatial compression mapping:

$$g = \sum_{i=1}^{H}\sum_{j=1}^{W} \alpha_{ij} F(:, i, j), \quad \alpha_{ij} = \frac{exp(q_{ij})}{\sum_{m,n} exp(q_{mn})} \tag{6}$$

where $F(:, i, j)$ denotes the channel vector at position (i,j), α_{ij} represents the spatial attention weights, and q_{ij} is the saliency score generated by a 1×1 convolution, capturing the contribution of different spatial locations to global semantic modeling. This weighted aggregation process avoids the excessive smoothing of fine-grained structural information inherent in traditional global average pooling, enabling small target regions to achieve higher participation in contextual representations. Building upon this, the global context vector g is fed into a channel re-calibration unit, where bottleneck mapping generates channel attention coefficients:

$$\beta = \sigma(W_2 \delta(W_1 g)) \tag{7}$$

where W_1 and W_2 denote the dimension-reduction and dimension-increase mapping matrices, respectively. δ represents the ReLU activation function, σ denotes the Sigmoid normalization operator, and $\beta \in R^C$ characterizes the importance weights of different semantic channels. Subsequently, the channel weights are modulated with the original features on a per-channel basis to form the context-enhanced feature representation:

$$F' = \beta \odot F \tag{8}$$

where $\odot$ denotes per-channel multiplication. In order to balance the lightweight requirements and the stability of the system, this module uses a single-head attention structure and a low rank decomposition strategy, which greatly reduces the additional parameters and computational complexity. Using residual connections, the enhanced features are combined with the original features to ensure that the global constraint is propagated to the detection head layer. This approach allows the attention module to achieve fine control of semantic consistency in a complex context, providing a stable context for further classification and localization of candidate regions.

3.3.2 Deployment Position

The lightweight global context attention (GCA) module is deployed at the interface between the Neck output and the decoupled detection Head rather than in the shallow Backbone layers. This placement avoids excessive smoothing of fine spatial details while applying semantic recalibration to the multi-scale features before classification and regression.

Specifically, the three Neck outputs P_3, P_4,and P_5 corresponding to different detection scales, are independently processed by scale-specific GCA units. The enhanced features are expressed as:

$$P_s' = P_s + A(P_s), \quad s \in \{3,4,5\} \tag{9}$$

where P_s denotes the feature map generated by PAN-FPN at scale ss, and $A_s(\cdot)$ represents the corresponding spatial aggregation and channel recalibration mapping. The residual connection preserves the original semantic information and improves optimization stability.

Considering the importance of high-resolution features for small-object detection, the P3 branch retains full context modeling. The P4 and P5 branches use branch-specific low-rank projections matched to their respective channel dimensions. For a branch with C_s channels and reduction ratio r, the bottleneck dimension is C_s/r; therefore, the projection matrices are scale specific and are not shared directly between P4 and P5. This formulation reduces computational overhead while preserving scale-specific representations.

The deployment strategy is illustrated in Figure 3. The three GCA units operate independently between the corresponding Neck outputs and decoupled detection heads, following the parallel flow $P_s \rightarrow A_s \rightarrow P_s' \rightarrow Head_s$. Context recalibration is therefore completed before classification and regression, providing more stable semantic features for subsequent NWD-based localization.

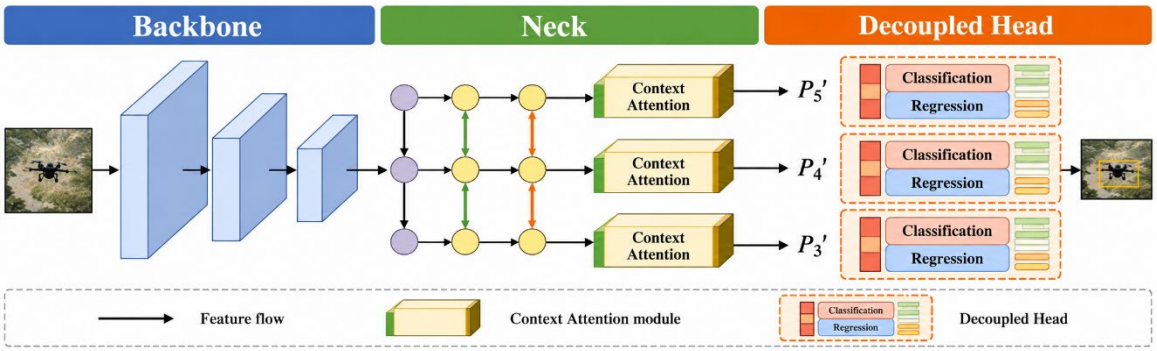


Fig. 3. Layout of lightweight global context attention at the neck-head interface

3.4 NWD-Based Small Object Regression Loss

3.4.1 Limitations of IoU Loss for Small Targets

In the UAV remote sensing scenario, the small target occupies a very small area in the feature map, and there is a significant difference in the size of the target. This has resulted in structural shortcomings in the representation of gradient and optimal stability of the classification regression loss based on IOU. Traditional IoU loss uses the intersection-over-union ratio between predicted and ground-truth boxes as its sole metric, expressed as

$$IoU(B_p, B_g) = \frac{|B_p \cap B_g|}{|B_p \cup B_g|} \tag{10}$$

where B_p and B_g denote the predicted and ground-truth bounding boxes, respectively. This metric effectively captures localization errors for medium-to-large-scale targets. However, when target scales shrink significantly, minor box translations or scale perturbations cause drastic changes in the intersection area. This results in non-continuous drops in IoU values or even complete degradation to zero, causing regression gradients to vanish frequently during early training stages. Particularly in aerial perspectives, small targets often appear against complex backgrounds with dense distributions. Even when prediction boxes closely match ground truth boxes in central position and aspect ratio, the IoU loss fails to provide effective penalty signals for "close but non-overlapping" states unless substantial overlap is achieved. This inherently limits the model's ability to correct early localization deviations. Although IoU and its variants (e.g., GIoU, DIoU, CIoU) are bounded by some additional terms (e.g., center distance, aspect ratio), area overlap still dominates: the additional terms are often dominated by the main area intersection-over-union term in small-object cases, leading to hard-to-control gradient directions during backpropagation. More importantly, IoU loss shows strong non-uniform scale sensitivity: pixel-wise object offsets result in small IoU changes on large objects but lead to orders-of-magnitude larger loss changes on small objects. This imbalance easily leads to regression difficulty disparities among objects of different scales within the same training batch, ultimately affecting the overall convergence behavior of the detection head. These issues indicate that relying solely on region-overlap-based IoU metrics struggles to capture the true geometric similarity between small bounding boxes. This limitation provides clear design motivation and a theoretical entry point for introducing the NWD regression loss, which centers on distribution distance.

3.4.2 Normalized Wasserstein Distance

To address gradient instability in small object localization caused by area-based overlap metrics, this paper introduces Normalized Wasserstein Distance (NWD) as the core geometric similarity metric in regression loss design. This transforms the bounding box matching process from modeling "area overlap relationships" to modeling "spatial distribution distances". Specifically, the predicted box and ground truth box are modeled as two-dimensional Gaussian distributions $N_p(\mu_p, \Sigma_p)$ and $N_g(\mu_g, \Sigma_g)$, respectively. The mean vector $\mu = (x, y)$ represents the target center coordinates, while the covariance matrix $\Sigma = diag(w^2/4, h^2/4)$ is constructed from the bounding box width$w$ and height$h$ to capture the spatial scale distribution characteristics of the target. Under

this modeling framework, the second-order Wasserstein distance between two boxes can be expressed as

$$W_2^2(N_p, N_p) = \left\|\mu_p - \mu_g\right\|_2^2 + \left\|\Sigma_p^{1/2} - \Sigma_g^{1/2}\right\|_F^2 \tag{11}$$

where $\|\|_2$ denotes the Euclidean norm and $\|\|_F$ denotes the Frobenius norm. The first term measures center displacement, while the second characterizes differences in box scale and shape. This formulation provides geometric supervision even when the predicted and ground-truth boxes have limited or no overlap.

To bound the geometric distance and improve numerical stability, the Wasserstein distance is mapped to a normalized similarity score:

$$NWD(B_p, B_g) = exp\left(-\frac{\sqrt{W_2^2(N_p, N_g)}}{C}\right) \tag{12}$$

where C is a fixed scaling constant that controls the mapping from Wasserstein distance to normalized similarity. It regulates sensitivity to geometric deviations and does not normalize the spatial resolutions of the P3, P4, and P5 feature maps. The resulting NWD value lies within (0,1], where a larger value indicates greater geometric similarity between two bounding boxes.

3.4.3 Loss Function Definition

Building upon the normalized Wasserstein distance modeling described earlier, this paper embeds the NWD metric into the bounding box regression branch of YOLOv8. This constructs a unified regression loss function that balances geometric consistency and optimization stability, replacing the traditional single overlap constraint based on IoU [26]. Let the predicted bounding box and ground truth bounding box for the i-th sample be denoted as B_p^i and B_g^i, respectively. Their normalized Wasserstein distance is recorded as $NWD(B_p^i, B_g^i)$. The distribution distance-based regression loss term is defined as

$$L_{NWD} = \frac{1}{N}\sum_{i=1}^{N}(1 - NWD(B_p^i, B_g^i)) \tag{13}$$

where N represents the number of positive samples involved in regression loss computation. NWD∈(0, 1] represents the overall similarity between the predicted and ground truth bounding boxes in terms of both center position and scale structure. Since the geometric differences between the two bounding boxes are mapped to an differentiable optimization term in "distance-similarity", the gradient feedback for these small target samples can still be effective even though they do not overlap with each other. The NWD term is incorporated into the YOLOv8 regression objective without introducing an independent object-confidence loss. The classification term is retained, while NWD provides geometric supervision for small-object localization together with the distribution-based regression term. The resulting loss is defined as:

$$L = \lambda_{cls}L_{cls} + \lambda_{dfl}L_{DFL} + \lambda_{nwd}L_{NWD} \tag{14}$$

where L_{cls} denotes the classification loss, L_{DFL} denotes Distribution Focal Loss, and L_{NWD} represents the proposed Wasserstein-distance-based localization loss. The coefficients λ_{cls}, λ_{dfl} and λ_{nwd} balance the corresponding optimization objectives.

4. Experimental Design and Results Analysis

4.1 Experimental Datasets and Evaluation Metrics

The VisDrone-DET2021 dataset is adopted to evaluate the detection performance and deployment feasibility of the proposed small-object detection framework. The dataset contains UAV images

acquired from diverse outdoor environments and provides dense annotations suitable for evaluating small-object detection under complex aerial scenes. The images cover urban, rural, road, and bridge environments and include challenging conditions such as occlusion, shadows, illumination variation, and background clutter. Image resolutions range from 640×480 to 2000×1500. Ten common UAV object categories are annotated, including pedestrians, cars, buses, motorcycles, and bicycles. The average target size in the training set is approximately 28×24 pixels, indicating that a large proportion of objects belong to the small-object range.

Following the official VisDrone-DET2021 partition, the dataset contains 6,471 training images, 548 validation images, and 1,610 unlabeled test images. The training set is used for model optimization, while the validation set is used for hyperparameter adjustment and accuracy evaluation. The test subset contains 1,610 unlabeled images and is used only for inference benchmarking; accuracy-related metrics are calculated on the labeled validation set. The dataset partitioning, average target size, and input resolution are summarized in Table 1. For compatibility with the YOLOv8 architecture, all input images are resized to 640×640. This unified resolution maintains consistent feature-map dimensions during multiscale fusion.

Table 1. Basic parameters of the VisDrone-DET2021 dataset

Dataset Partitioning	Number of Images	Number of Object Categories	Average Target Size (pixels)	Input Resolution
Training Set	6471	10	28×24	640×640
Validation Set	548	10	27×25	640×640
Test Set	1610	Unlabeled	-	640×640

Model performance is evaluated using mAP@0.5, mAP@0.5:0.95, Precision, Recall, and mAP-S. mAP@0.5 measures detection accuracy at an IoU threshold of 0.5, while mAP@0.5:0.95 averages the detection performance over IoU thresholds from 0.5 to 0.95 with a step of 0.05. Precision reflects the proportion of correct detections among all predictions, whereas Recall measures the proportion of ground-truth objects successfully detected. mAP-S is used to evaluate objects smaller than 32×32 pixels, emphasizing detection performance for small targets under dense, occluded, and cluttered UAV scenes.

4.2 Experimental Environment and Model Training Parameter Settings

All experiments were conducted on a unified hardware and software platform: Ubuntu 20.04, an NVIDIA RTX 3090 GPU with 24 GB VRAM, an Intel Xeon Gold 6226R CPU (2.9 GHz, 16 cores), and 128 GB RAM. The software environment comprised PyTorch 2.0, CUDA 11.8, and cuDNN 8.6. Training used a single GPU and the official open-source YOLOv8 implementation as the base framework. The proposed detail enhancement, contextual attention, and NWD regression components were integrated into the end-to-end model. All input images were resized to 640×640.

AdamW was used for optimization with an initial learning rate of 1.0×10^{-3}. Training ran for a maximum of 300 epochs, with a 10-epoch linear warmup that increased the learning rate from 1.0×10^{-6} to the initial value. The batch size was 32, the gradient-clipping threshold was 10, and decoupled weight decay was 1.0×10^{-4}. Exponential moving average updates used a coefficient of 0.999. The comparison models used the same input preprocessing and evaluation protocol.

4.3 Comparative Analysis with Mainstream Object Detection Methods

The proposed framework was compared with representative one-stage and two-stage detectors, including YOLOv5s, the standard YOLOv7 base configuration, YOLOv8s, RetinaNet, and Faster R-CNN. All models were evaluated on the same VisDrone-DET2021 partition at an input resolution of 640×640. The evaluation metrics were mAP@0.5, mAP@0.5:0.95, Recall, and mAP-S. Table 3 summarizes the quantitative comparison. The approximation sign denotes values rounded to one decimal place. The proposed model achieves higher mAP@0.5, mAP@0.5:0.95, Recall, and mAP-S than the YOLOv8s baseline. Figure 4 visualizes the four-evaluation metrics in Table 3 as point estimates. The numerical values are provided in Table 3.

Table 2. Experimental environment and training parameter settings

Item	Configuration/Value
Operating System	Ubuntu 20.04
Graphics Card	NVIDIA RTX 3090 (24GB)
Deep Learning Framework	PyTorch 2.0 + CUDA 11.8
Optimizer	AdamW
Initial learning rate	1.0×10^{-3}
Learning Rate Scheduling Strategy	Cosine Annealing + Warmup
Batch Size	32
Maximum Training Iterations	300 epochs
Regularization Method	Weight Decay: 1.0×10^{-4}
Gradient Clipping Threshold	10
EMA parameter update	Enabled (coefficient 0.999)
Input Image Resolution	640×640

Table 3. Quantitative comparison with mainstream object detection methods

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Recall (%)	mAP-S (%)
YOLOv5s	≈72.0	≈46.0	≈68.0	≈41.0
YOLOv7 (base)	≈74.0	≈49.0	≈71.0	≈45.0
YOLOv8s	76.2	50.4	73.1	48.7
Faster R-CNN	≈69.0	≈43.0	≈65.0	≈37.0
RetinaNet	≈70.0	≈45.0	≈67.0	≈40.0
Proposed Model	81.6	56.9	78.4	54.5

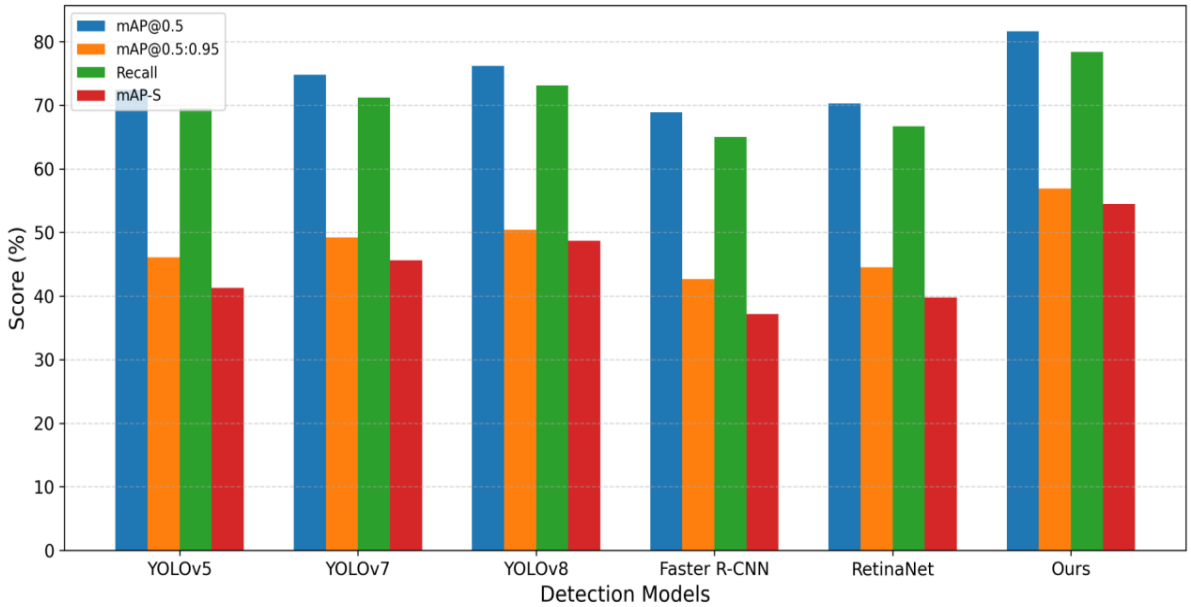


Fig. 4. Performance metrics comparison of object detection models for small target detection in remote sensing

To examine model behavior under challenging UAV scenes, Figure 5 presents qualitative detection comparisons between baseline YOLOv8s and the proposed method. Representative samples include dense-object distributions, partial occlusion, and small-target scenes. The proposed model shows more complete target responses and more stable bounding-box localization in these cases, providing qualitative support for the effectiveness of the proposed feature enhancement and contextual modeling mechanisms.

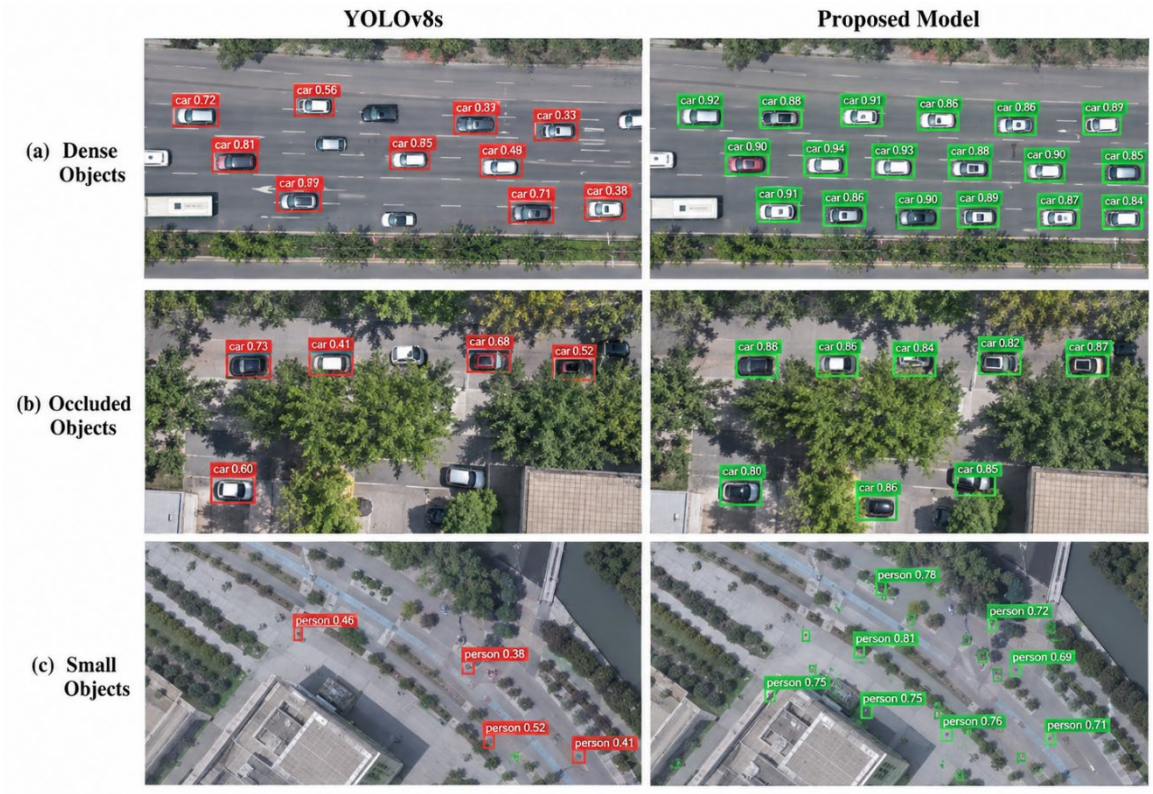


Fig. 5. Qualitative detection comparison between YOLOv8s and the proposed model

4.4 Ablation Studies and Effectiveness Validation of Improved Modules

The ablation study evaluates the original YOLOv8s baseline, variants containing HF, AFF, GCA, or NWD individually, and the Full model containing all four modules. All variants use a 640×640 input and the same training settings. The evaluation metrics include mAP@0.5, mAP@0.5:0.95, Recall, and mAP-S. The detailed results are presented in Table 4.

As shown in Table 4, the HF module increases mAP@0.5 from 76.2% to 77.6% and mAP-S from 48.7% to 51.4%, corresponding to absolute gains of 1.4 and 2.7 percentage points, respectively. This result indicates that high-frequency detail enhancement improves the preservation of texture and boundary information during shallow feature extraction. The AFF module improves mAP@0.5 to 78.4%, mAP@0.5:0.95 to 53.3%, Recall to 74.9%, and mAP-S to 52.7%. The gains indicate that adaptive cross-scale weighting strengthens the interaction between spatial details and high-level semantic features. The GCA module achieves the largest Recall improvement among the single-module variants, increasing Recall from 73.1% to 75.4%, corresponding to an absolute gain of 2.3 percentage points. The +AFF variant also improves Recall to 74.9%, representing a gain of 1.8 percentage points. The stronger Recall improvement produced by GCA indicates that global semantic modeling helps reduce missed detections in complex backgrounds.

The NWD variant produces a relatively limited Recall increase but improves mAP@0.5:0.95 from 50.4% to 54.7%, yielding an absolute gain of 4.3 percentage points. This result indicates that distribution-distance-based regression provides more stable geometric supervision for small-object localization.

The Full model achieves the best overall performance, with mAP@0.5, mAP@0.5:0.95, Recall, and mAP-S reaching 81.6%, 56.9%, 78.4%, and 54.5%, respectively. Compared with the baseline, the corresponding improvements are 5.4, 6.5, 5.3, and 5.8 percentage points. These results demonstrate that the four modules provide complementary benefits in detail preservation, multiscale fusion, contextual modeling, and bounding-box regression.

Table 4. Module ablation and performance comparison results

Model Structure	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Recall (%)	mAP-S (%)
Baseline YOLOv8s	76.2	50.4	73.1	48.7
+HF	77.6	52.1	74.2	51.4
+AFF	78.4	53.3	74.9	52.7
+GCA	78.1	52.9	75.4	52.1
+NWD	77.2	54.7	73.6	50.3
Full Model	81.6	56.9	78.4	54.5

4.5 Analysis of Algorithm Complexity and Engineering Deployment Feasibility

To evaluate the computational cost and deployment feasibility of the proposed model, six detection models are compared in terms of parameter count, FLOPs, inference speed, and GPU memory consumption. All inference-speed tests are conducted on an NVIDIA RTX 3090 GPU with an input resolution of 640×640 and a batch size of 1. The detailed results are presented in Table 5. As shown in Table 5, YOLOv8s contains 11.2 M parameters and requires 28.6 GFLOPs, achieving an inference speed of 96.5 FPS. After integrating HF, AFF, GCA, and NWD, the proposed model contains 15.3 M parameters and requires 38.9 GFLOPs. Compared with YOLOv8s, the parameter counts increases by 36.6%, while the computational cost increases by approximately 36.0%. Nevertheless, the proposed model maintains an inference speed of 88.2 FPS at 640×640 resolution.

The feature enhancement and context modeling modules increase computational cost, while the model maintains real-time inference performance. Compared with RetinaNet and Faster R-CNN, the proposed method has fewer parameters and a higher inference speed. These results show an accuracy-efficiency trade-off suitable for real-time UAV small-object detection. Table 5 reports peak training VRAM at a batch size of 32. Inference speed was measured separately at a batch size of 1 and an input resolution of 640×640; thus, the memory and speed measurements correspond to different execution settings.

Table 5. Comparison of complexity and deployment performance among detection models

Model	Parameters (M)	FLOPs (G)	FPS (640×640)	Peak Training VRAM (GB), Batch Size = 32
YOLOv5s	7.2	15.9	108.3	2.1
YOLOv7 (base)	37.2	105.2	82.7	4.8
YOLOv8s	11.2	28.6	96.5	3.2
RetinaNet	34.8	119.7	41.9	5.4
Faster R-CNN	42.3	180.5	27.4	6.1
Proposed Model	15.3	38.9	88.2	3.7

5. Conclusions

This study presents an enhanced YOLOv8 framework for small-object detection in UAV remote sensing images by jointly improving detail representation, multiscale feature fusion, contextual modeling, and localization regression. The framework integrates four complementary components. The high-frequency detail enhancement branch preserves shallow edge, texture, and local-contrast responses that are easily weakened during progressive down sampling. Adaptive feature fusion assigns scale-dependent weights to spatial and semantic features, reducing the suppression of fine details during cross-scale aggregation. Global context attention recalibrates the Neck outputs before classification and regression, which strengthens semantic consistency in cluttered scenes. The NWD-based regression term models bounding boxes as spatial distributions and provides continuous geometric supervision when small, predicted boxes have limited or no overlap with the ground-truth boxes. Together, these components address feature attenuation, scale imbalance, background interference, and unstable localization within a unified detection architecture.

Experiments on the VisDrone-DET2021 validation set show that the integrated model improves mAP@0.5 from 76.2% to 81.6%, mAP@0.5:0.95 from 50.4% to 56.9%, Recall from 73.1% to 78.4%, and mAP-S from 48.7% to 54.5% relative to the YOLOv8s baseline. The ablation results further distinguish the contribution of each component. High-frequency detail enhancement produces the largest single-module gain in mAP-S among the feature-enhancement variants, adaptive fusion improves performance across all four-evaluation metrics, global context attention provides the strongest Recall improvement among the individual modules, and NWD produces its largest gain in the stricter mAP@0.5:0.95 metric. The Full model achieves the highest values for all reported metrics, indicating that the modules provide complementary improvements rather than duplicating the same function. The performance gains are accompanied by an increase from 11.2 M to 15.3 M parameters and from 28.6 to 38.9 GFLOPs. Nevertheless, the model achieves 88.2 FPS at an input resolution of 640×640 on an RTX 3090, supporting real-time inference under the reported test conditions. Future work will extend the distribution-distance formulation to oriented bounding boxes and reduce deployment cost through structured pruning and INT8 quantization. Evaluation on additional UAV datasets and edge-computing platforms will further assess cross-scene generalization and practical deployment performance.

References

[1] Chen C, Zheng Z, Xu T, et al. YOLO-Based UAV Technology: A Review of the Research and Its Applications. Drones. 2023;7(3):190. https://doi.org/10.3390/drones7030190

[2] Jiang C, Ren H, Ye X, et al. Object Detection from UAV Thermal Infrared Images and Videos Using YOLO Models. International Journal of Applied Earth Observation and Geoinformation. 2022;112:102912. https://doi.org/10.1016/j.jag.2022.102912

[3] Zhai X, Huang Z, Li T, et al. YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection. Electronics. 2023;12(17):3664. https://doi.org/10.3390/electronics12173664

[4] Sangaiah A K, Yu F N, Lin Y B, et al. UAV T-YOLO-Rice: An Enhanced Tiny YOLO Network for Rice Leaf Disease Detection in Paddy Agronomy. IEEE Transactions on Network Science and Engineering. 2024;11(6):5201-5216. https://doi.org/10.1109/TNSE.2024.3394602

[5] Qiu Q, Lau D. Real-Time Detection of Cracks in Tiled Sidewalks Using a YOLO-Based Method Applied to Unmanned Aerial Vehicle Images. Automation in Construction. 2023;147:104745. https://doi.org/10.1016/j.autcon.2023.104745

[6] Zeng S, Yang W, Jiao Y, et al. SCA-YOLO: A New Small Object Detection Model for UAV Images. The Visual Computer. 2024;40(3):1787-1803. https://doi.org/10.1007/s00371-023-03030-x

[7] Cao J, Bao W, Shang H, et al. GCL-YOLO: A GhostConv-Based Lightweight YOLO Network for UAV Small Object Detection. Remote Sensing. 2023;15(20):4932. https://doi.org/10.3390/rs15204932

[8] Sadykova D, Pernebayeva D, Bagheri M, et al. IN-YOLO: Real-Time Detection of Outdoor High-Voltage Insulators Using UAV Imaging. IEEE Transactions on Power Delivery. 2019;35(3):1599-1601. https://doi.org/10.1109/TPWRD.2019.2945230

[9] Xu L Y, Zhao Y F, Zhai Y H, et al. Small Object Detection in UAV Images Based on YOLOv8n. International Journal of Computational Intelligence Systems. 2024;17(1):223. https://doi.org/10.1007/s44196-024-00609-w

[10] Ni J, Zhu S, Tang G, et al. A Small-Object Detection Model Based on Improved YOLOv8s for UAV Image Scenarios. Remote Sensing. 2024;16(13):2465. https://doi.org/10.3390/rs16132465

[11] Feng F, Hu Y, Li W, et al. Improved YOLOv8 Algorithms for Small Object Detection in Aerial Imagery. Journal of King Saud University - Computer and Information Sciences. 2024;36(6):102113. https://doi.org/10.1016/j.jksuci.2024.102113

[12] Yuan Z, Gong J, Guo B, et al. Small Object Detection in UAV Remote Sensing Images Based on Intra-Group Multi-Scale Fusion Attention and Adaptive Weighted Feature Fusion Mechanism. Remote Sensing. 2024;16(22):4265. https://doi.org/10.3390/rs16224265

[13] Khalili B, Smyth A W. SOD-YOLOv8: Enhancing YOLOv8 for Small Object Detection in Aerial Imagery and Traffic Scenes. Sensors. 2024;24(19):6209. https://doi.org/10.3390/s24196209

[14] Zhou Y, Jiang Y, Yang Z, et al. A Small Object Detection Method Based on the Attention Mechanism and Multi-Level Feature Fusion. International Conference on Image Processing and Artificial Intelligence (ICIPAI 2024). SPIE; 2024. p. 665-669. https://doi.org/10.1117/12.3042217

[15] Liu C, Gao G, Huang Z, et al. YOLC: You Only Look Clusters for Tiny Object Detection in Aerial Images. IEEE Transactions on Intelligent Transportation Systems. 2024;25(10):13863-13875. https://doi.org/10.1109/TITS.2024.3402432

[16] Su N, Zhao Z, Yan Y, et al. MMPW-Net: Detection of Tiny Objects in Aerial Imagery Using Mixed Minimum Point-Wasserstein Distance. Remote Sensing. 2024;16(23):4485. https://doi.org/10.3390/rs16234485

[17] Han Y, Wang C, Luo H, et al. LRDS-YOLO Enhances Small Object Detection in UAV Aerial Images with a Lightweight and Efficient Design. Scientific Reports. 2025;15(1):22627. https://doi.org/10.1038/s41598-025-06789-x
[18] Yang S, Ge H. Small Target Detection in UAV Images Based on SCMS-YOLOv8. 2025 10th International Conference on Automation, Control and Robotics Engineering (CACRE). IEEE; 2025. p. 182-187. https://doi.org/10.1109/CACRE64522.2025.10982341
[19] Li M, Chen Y, Zhang T, et al. TA-YOLO: A Lightweight Small Object Detection Model Based on a Multi-Dimensional Trans-Attention Module for Remote Sensing Images. Complex & Intelligent Systems. 2024;10(4):5459-5473. https://doi.org/10.1007/s40747-024-01440-4
[20] Fang M, Rui X, Cheng H, et al. Small Object Detection Algorithm Based on Improved Attention Mechanism and Feature Fusion of YOLOv8. Journal of Advanced Computational Intelligence and Intelligent Informatics. 2025;29(4):941-955. https://doi.org/10.20965/jaciii.2025.p0941
[21] Qin J, Yang D, Liu F. Lightweight Object Detection Algorithm for UAV Aerial Images Based on Improved YOLOv8. Proceedings of the 2024 8th International Conference on Electronic Information Technology and Computer Engineering. 2024. p. 134-137. https://doi.org/10.1145/3707252.3707279
[22] Zhang J, Song L Q, Gao M, et al. Interactive Feature Pyramid Network for Small Object Detection in UAV Aerial Images. Displays. 2026;91:103352. https://doi.org/10.1016/j.displa.2025.103352
[23] Wu S, Yang H, Liao L, et al. Dynamic Small Object Feature Enhancement and Detection for Remote Sensing Images. Scientific Reports. 2025;15(1):37225. https://doi.org/10.1038/s41598-025-12345-z
[24] Junos M H, Mohd Khairuddin A S, Thannirmalai S, et al. Automatic Detection of Oil Palm Fruits from UAV Images Using an Improved YOLO Model. The Visual Computer. 2022;38(7):2341-2355. https://doi.org/10.1007/s00371-021-02288-7
[25] Boudjit K, Ramzan N. Human Detection Based on Deep Learning YOLO-v2 for Real-Time UAV Applications. Journal of Experimental & Theoretical Artificial Intelligence. 2022;34(3):527-544. https://doi.org/10.1080/0952813X.2021.1895315
[26] Bao W, Zhu Z, Hu G, et al. UAV Remote Sensing Detection of Tea Leaf Blight Based on DDMA-YOLO. Computers and Electronics in Agriculture. 2023;205:107637. https://doi.org/10.1016/j.compag.2023.107637