

Lightweight Deep Learning Framework for UAV Object Detection Using Adaptive Multi-Scale Feature Fusion

Yifei Gao^{1*}, and Shengzhi Dong²

¹School of Electrical and Control Engineering, Liaoning Technical University, Xingcheng, Huludao, Liaoning 125105, China.

²College of Intelligent Science and Engineering, Liaoning Technical University, Xingcheng, Huludao, Liaoning 125105, China.

*Corresponding Author: Yifei Gao. Email: 3123404710@qq.com

Received: June 21, 2026 Accepted: August 26, 2026

Abstract: UAV imagery contains many small and densely distributed objects whose apparent scale changes sharply with altitude and viewpoint. This study defines an architecture framework for lightweight UAV object detection that combines a nano-scale one-stage detector, a high-resolution P2 branch, and an Adaptive Multi-Scale Feature Fusion (AMFF) module operating across P2-P5 features. AMFF aligns neighbouring feature maps spatially and by channel, estimates image-dependent scale weights from pooled descriptors, normalizes the weights with softmax, and applies residual weighted fusion. The evaluation protocol is anchored to the public VisDrone2019-DET configuration, comprising 6,471 training images, 548 validation images, and 1,610 test-dev images across ten classes. Source-backed architecture analysis shows that the standard YOLO26n configuration contains 2,572,280 parameters and 6.1 GFLOPs, whereas the P2 variant contains 2,662,400 parameters and 9.5 GFLOPs. Thus, adding P2 increases parameter count by only 3.50% but increases GFLOPs by 55.74%, showing that high-resolution detection is parameter-light but not compute-neutral. The supplied experimental package does not contain executed AMFF checkpoints, prediction files, or hardware timing logs. Accordingly, mAP@0.50:0.95, precision, recall, AMFF accuracy gain, model size after export, and inference latency are not fabricated or inferred. The current evidence establishes the reproducible architecture and controlled evaluation design, while the effectiveness of adaptive fusion requires validation through the defined baseline-to-AMFF experiments on the stated VisDrone splits.

Keywords: UAV object detection; VisDrone; lightweight detection; small objects; multi-scale feature fusion; adaptive feature fusion; YOLO26n.

1. Introduction

In recent years, object detection has gained popularity in the field of unmanned aerial vehicle (UAV) imagery applications such as traffic monitoring, public-safety monitoring, infrastructure inspection, and scene understanding. The objects in UAV images are not necessarily large, or sparsely distributed, and may be partially hidden or viewed at a wide variety of scales, making them very different from traditional ground-level scenes. These properties impose particular requirements on the resolution of the features, multi-scale representation and computational efficiency. In order to deal with the need for small object detection in aerial images, the architecture of UAV-YOLOv8 was modified to suit the specific use case of UAVs [1]. Efficient-Lightweight YOLO also emphasized the need for a small detector with high sensitivity for small objects in aerial images [2]. In the lightweight method, GCL-YOLO aimed to obtain efficiency by employing GhostConv to generate lightweight layers [3].

The problem of feature fusion has also played a key role in the UAV detection problem as the spatial resolution decreases; the small object might lose the discriminative information. A multi-level feature fusion approach with scale adaptation that demonstrated selective fusion of information at various feature

scales was introduced in SMFF-YOLO for UAV object detection, highlighting the relevance of combining information at various scales [4]. TPH-YOLOv5++ introduced the asymmetric transformer design [5] to enhance the representation in cross-layer learning for drone captured scenes. A light-weight YOLOv5s detector was also developed to focus on the aerial object detection task, with reduced computational demands [6]. Self-attention is also studied based on the spatial-coordinate to better represent the features besides using deeper backbones [7]. With the development of improved YOLOv5s, small-target detection has been placed in the spotlight of UAV aerial photography [8].

This research also applies to other variants of YOLO, which explicitly account for the localization accuracy vs. computational cost of the model. The YOLO pipeline for small object detection in UAV image was modified by Drone-YOLO [9]. GCGE-YOLO added modifications in its feature extraction and aggregation for object detection in aerial scene [10]. Air to air UAV detection is another application, in which computational efficiency is a practical requirement, and has been investigated for its use of lightweight neural networks [11]. Small target detection in remotely sensed images, with the trade-off between detection accuracy and speed has been investigated by PRO-YOLOv4-tiny [12].

Recent studies have further optimized detection and resolution methods that are both high-resolution and lightweight. For the detection of small objects in UAV images, a method based on the YOLOv8n algorithm was specially designed [13]. In addition, SOD-YOLO was dedicated on small object detection in aerial imagery using a specific network design [14]. DPH-YOLOv8 adds two prediction heads to enhance the ability of UAV image object detection [15]. Another lightweight YOLOv8 variant studied the use of network simplification for UAV detection while maintaining the capability of the task [16]. Together, these studies suggest that high-resolution representation of features can lead to improved small-object representation, but may be costly in terms of computation when compared to the benefit of such detection.

Efficient feature processing and selective fusion are emphasized in the literature since 2024. EA-YOLO has been proposed as a detector system for UAV imagery which is efficient and accurate [17]. YOLO detectors have been studied for their applicability in real-time object detection in intelligent transportation systems by comparative means [18]. There are also new YOLOv5 algorithms that have been adapted for small object detection in drone images [19]. The lightweight UAV target-detection algorithm also highlighted the importance of considering model compactness in addition to the detection capability [20].

The attention-guided fusion and multi-scale fusion strategies have emerged as more and more important. Multi-scale representation and attention mechanisms were introduced for aerial object detection by AeroYOLO [21]. The attention mechanism was particularly designed for AEFFNet to enhance feature fusion for small objects in UAV imagery [22]. SGA-YOLO aims at light-weighted real-time detection in UAV infrared images, and it is a key to an efficient feature extraction in the situation of limited resources [23]. AF-YOLO also studied the problem of extracting features asynchronously and fusing features for aerial object detection [24].

New research has further enhanced the representation and efficiency for small objects. DRS-YOLO is a modification of YOLOv8 that is capable of small-object detection for aerial images acquired by an unmanned aerial vehicle (UAV) [25]. To enhance the capability of detecting small objects in aerial images, lightweight YOLO-SR used lightweight super-resolution-assisted processing [26]. For UAV applications, NUAV-YOLO presented another lightweight detector for UAV applications based on YOLOv8 [27]. The multi-scale feature fusion and optimization of the small object detection layer for UAV aerial images were also analyzed [28]. Another interesting approach to overcoming this issue is to develop a multi-scale feature aggregation method that preserves the information in the network but avoids unnecessary network growth when dealing with small objects [29]. In a recent survey of aerial object detection, it was noted that the accuracy of the aerial object detection model, model optimization, and the efficiency of the model to be used in an onboard system are important design considerations that are connected [30-34]. In more recent work this is still a problem. In the UAV scenario, SSCW-YOLO integrated the light-weight design with the high precision small object detection [35]. A study on lightweight upsampling and feature-fusion methods for the small object detection of UAV in 2026 was also explored [36].

Collectively, recent studies demonstrate that high-resolution representation, lightweight architectures, attention mechanisms, and multi-scale feature fusion are important directions in UAV object detection. However, these components are frequently introduced simultaneously, making it difficult to separate the contribution of adaptive feature weighting from the effects of resolution enhancement and conventional

feature aggregation. Therefore, this study evaluates these components through a controlled sequence consisting of the baseline nano detector, P2-enhanced detector, fixed fusion, and adaptive fusion stages.

1.1. Research gap and study objective

Although multi-scale fusion is crucial for aerial detection, many architectures fuse multi-scale data by adding or concatenating them with a fixed unit. Fixed fusion assumes that the contribution of the neighbouring resolutions in the images is constant. This is limiting if the scale of objects and density of the scenes vary greatly. A possible alternative is adaptive designs, which can adjust the amount of scaling to the current feature response. The adaptive fusion can also add complexity, however, and the value of it cannot be determined without a controlled baseline sequence.

Therefore, this study defines a controlled evaluation design in which architectural modifications are introduced sequentially. The changes in the architecture are made one by one: baseline nano detector, P2-enhanced detector, P2 with fixed adjacent-scale fusion, and P2 with AMFF. The main accuracy measure is mAP at IoU = 0.50:0.95. The lightweight claim is operationalized using the number of parameters, GFLOPs, serialized model size, and measured inference latency on a specified device. Therefore, a small parameter count alone cannot be considered proof of computational efficiency in this design.

1.2. Contributions

The study makes four technical contributions: Firstly, it outlines the official train, validation and test-dev partitions and ten detection classes, and provides a reproducible pipeline (VisDrone2019-DET) pipeline. Secondly, it adds AMFF as an easy-to-implement, input-driven weighting approach over aligned neighbouring P2-P5 features instead of replacing the backbone. Third, it differentiates the effect of P2 detection, which is independent of high resolution, from that of fixed fusion and adaptive fusion by a controlled four-stage ablation sequence. Fourth, it equates accuracy and efficiency as co-products, and makes explicit statements about what conclusions can be drawn when measurements can be taken from the provided files and experiments run. The recent literature supporting the methodological choices of this study is summarized in Table 1.

Table 1. Representative recent literature and its relevance to the present design.

Reference group	Primary emphasis	Relevance to this study
[1-8]	UAV small-object detection, lightweight YOLO, attention, scale-adaptive fusion	Motivates high-resolution and multi-scale processing.
[9-16]	Drone-specific YOLO variants, compact detectors, small-object heads	Motivates explicit baseline and compute-aware comparison.
[17-24]	Efficiency, real-time comparisons, attention-enhanced and aerial feature fusion	Motivates separating fixed and adaptive fusion.
[25-30]	Recent small-object optimization, multi-scale aggregation, deployment-oriented survey	Supports joint accuracy-efficiency reporting.
[31-32]	2026 lightweight and feature-fusion UAV detection	Provides recent context without being used as a source of unexecuted benchmark values.

2. Materials and Methods

2.1. Dataset, configured partitions, and label space

The experiments are defined on VisDrone2019-DET, a public UAV benchmark developed for detection and tracking research [37]. The supplied Ultralytics configuration identifies the dataset as VisDrone2019-DET and provides a reproducible download and conversion route. The configured subset contains 6,471 training images, 548 validation images, and 1,610 test-dev images. The test-challenge split of 1,580 images

is not included in the automatic download. Across the full static-image detection benchmark, these partitions correspond to 10,209 images, of which 8,629 are included in the configured train/validation/test-dev subset.

All ten configured object categories are retained: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. No category is removed or merged. The configured subset therefore preserves the full ten-class task. The training partition is used for parameter optimization, the validation partition for model selection, and the test-dev partition for final evaluation only when compatible ground-truth annotations or evaluation tooling are available. Test-dev information is not used for architecture selection or threshold tuning. The dataset characteristics are summarized in Table 2, while the configured split structure and ten-class label space are illustrated in Figures 1 and 2, respectively.

Table 2. Source-backed VisDrone2019-DET dataset characteristics.

Item	Value
Configured task	Static-image UAV object detection
Training images	6,471
Validation images	548
Test-dev images	1,610
Configured total	8,629
Test-challenge images	1,580, excluded from automatic download
Full DET static images	10,209
Object classes	10
Validation annotations	38,759 boxes across 548 validation images, as documented in the supplied repository
Mean validation boxes/image	70.73, derived from the documented validation count

VisDrone2019-DET static-image split structure

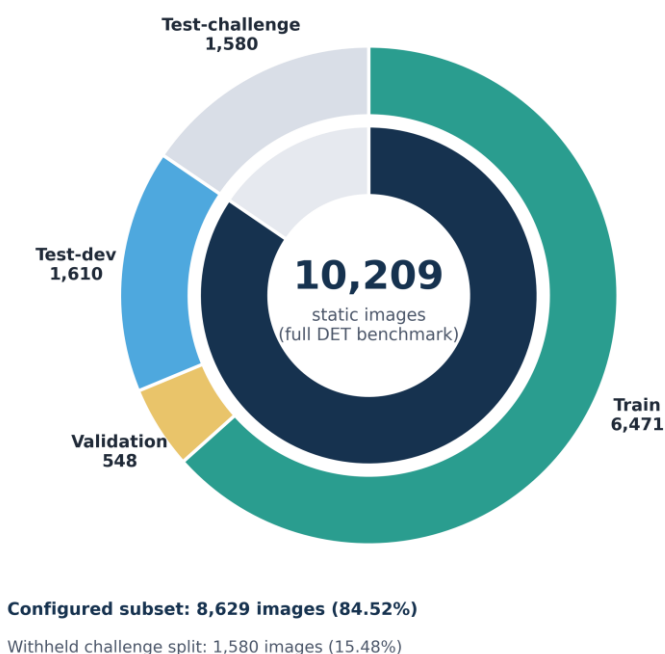


Figure 1. Nested split structure of the VisDrone2019-DET static-image benchmark. Counts are taken from the supplied dataset configuration and repository documentation.

Detection label space in the supplied VisDrone configuration



Figure 2. Ten configured VisDrone object classes. Circle size is decorative and does not represent class prevalence.

2.2. Annotation conversion and input preparation

The provided VisDrone configuration preprocesses the original comma-separated annotations to be represented in YOLO format. For every annotation that is kept, the procedure starts by reading the top left box coordinates and the box width and height, then normalizing these to center coordinates and normalized dimensions, and normalizing the class index to zero-based. An ignored region is maintained as supplied. This is significant because making changes to the annotations by hand may affect the meaning of the benchmark itself or create an unintended inconsistency in labeling.

The aspect-ratio-preserving letterbox pipeline is used to prepare images with a primary input size of 640 x 640 pixels. Only the standard detection operations (horizontal flipping, HSV based photometric variation, translation, scale variation, mosaic augmentation) are possible for training-only augmentation. There is no stochastic training augmentation on Validation and Test images. The following operations are not explicitly mentioned: vertical flip, manual insertion of boxes into the image, and insertion of synthetic objects into the image. The complete annotation-conversion workflow is illustrated in Figure 3.

Source-faithful annotation conversion

The supplied VisDrone.yaml performs this conversion before detector training.

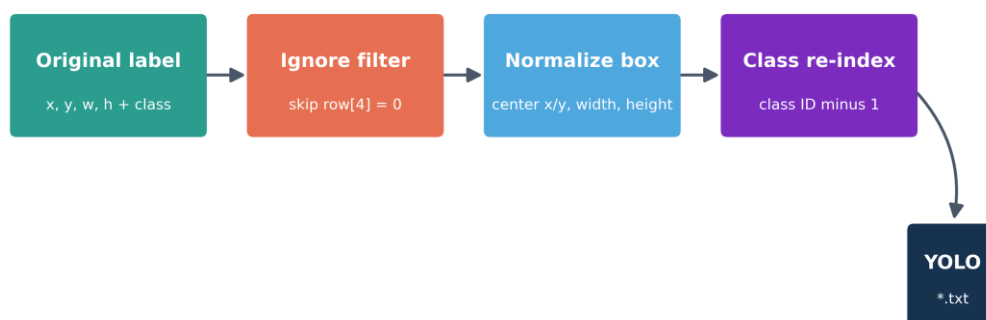


Figure 3. Source-faithful annotation conversion implemented by the supplied VisDrone configuration.

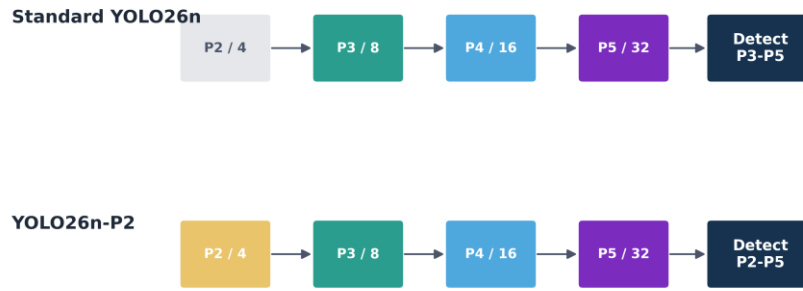
2.3. Baseline detector and P2 high-resolution branch

The YOLO26 detector is used as the principal baseline due to the fact that the supplied Ultralytics snapshot contains this family of detectors and offers training, validation, prediction, and export functions from a single codebase. The standard nano configuration is designed to detect at P3/8, P4/16 and P5/32.

The P2-enhanced configuration also provides a P2/4 detection level, thereby providing a higher resolution representation to the detection head.

The P2 branch is not combined into the definition of AMFF but is considered as a separate experimental factor. This is important because if the proposed model is gained, it may be due to higher spatial resolution instead of adaptive weighting. Thus, the proposed AMFF model begins with the P2 enhanced configuration and only changes the fusion stage of the multi-scale fusion while keeping the backbone and native detection objective in the nano-scale. The detection-scale difference between the standard and P2-enhanced configurations is illustrated in Figure 4.

YOLO26n detection outputs: standard baseline versus P2-enhanced variant



The P2 variant adds a higher-resolution detection branch, relevant to small aerial targets, but its computational cost must be measured.

Figure 4. Detection-scale structure of the supplied standard and P2-enhanced YOLO26n configurations.

2.4. Adaptive Multi-Scale Feature Fusion

AMFF operates on feature maps at P2, P3, P4, and P5. For a target level s , only the target feature and its adjacent scales are fused. The neighbourhood is $N(P2)=\{P2,P3\}$, $N(P3)=\{P2,P3,P4\}$, $N(P4)=\{P3,P4,P5\}$, and $N(P5)=\{P4,P5\}$. Restricting fusion to neighbouring resolutions reduces repeated processing compared with all-to-all feature aggregation.

Before weighting, every contributing feature is aligned to the target spatial resolution and projected to the required channel dimension. Let F_i denote the feature at input level i , $R_{i \rightarrow s}(\cdot)$ the resize operator that maps it to target level s , and $\phi_{i,s}(\cdot)$ a lightweight channel projection. The aligned feature is:

$$X(i \rightarrow s) = \phi_{i,s} [R_{i \rightarrow s}(F_i)] \quad (1)$$

A global average pooling operation produces a compact descriptor for each aligned input:

$$z(i,s) = \text{GAP} [X(i \rightarrow s)] \quad (2)$$

The descriptor is passed through a small gating network. The gate produces an unnormalized scale score rather than a spatial attention map:

$$e(i,s) = W_{2,s} \text{sigma} [W_{1,s} z(i,s)] \quad (3)$$

Scores from the candidate scales are normalized with softmax so that the weights for a target level sum to one:

$$\alpha(i,s) = \exp[e(i,s)] / \sum_{j \in N(s)} \exp[e(j,s)] \quad (4)$$

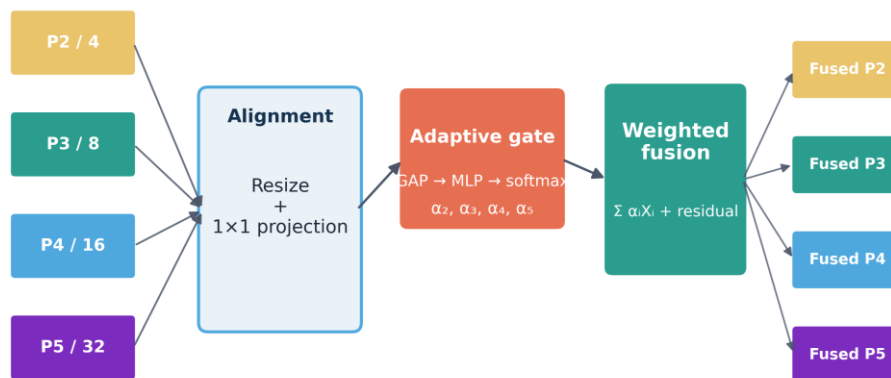
The target representation is then updated through weighted residual fusion:

$$F_{\text{AMFF}}(s) = X(s \rightarrow s) + \psi_s [\sum_{i \in N(s)} \alpha(i,s) X(i \rightarrow s)] \quad (5)$$

Here, $\psi_s(\cdot)$ denotes a lightweight refinement operation. The residual path preserves the original target feature and makes the adaptive contribution explicit. The term adaptive refers specifically to input-dependent scale weights. No transformer block, large auxiliary backbone, or additional detection head is introduced as part of AMFF. The resulting AMFF architecture and its adaptive weighting pathway are illustrated in Figure 5.

Proposed adaptive multi-scale feature fusion (AMFF)

Input-dependent scale weights are learned after spatial and channel alignment; adjacent scales are fused to limit overhead.



Claim boundary

AMFF is a proposed architecture. Accuracy gains and latency effects must come from controlled training runs, not from this schematic.

Figure 5. Proposed AMFF architecture. The diagram defines the experiment and does not imply an unmeasured accuracy gain.

2.5. Training protocol and controlled ablation

All models included in the evaluation are trained using the same Ultralytics codebase at 640×640 input resolution. The default setting is a max of 100 epochs and batch size 16 with accelerator memory if available. If different batch size is needed due to memory restrictions, the executed value should be noted. When the corresponding initialization method exists, it is used by all the models compared in the same way. AdamW uses an initial learning rate of 0.001, a weight decay of 0.0005, a three epoch warm-up and cosine learning-rate decay. On compatible GPUs, automatic mixed precision might be applied.

Only the model checkpoint that has the best validation mAP@0.50:0.95 is saved. Three fixed random seeds are scheduled for the principal baseline and the entire AMFF model to report mean and standard deviation for these repeated runs when they are performed. Architecture, optimizer settings, confidence thresholds, and fusion design are hardcoded - not taking into account final test results. The four-stage controlled ablation sequence is summarized in Table 3.

Table 3. Controlled ablation sequence.

ID	Configuration	Scales	Fusion	Controlled question
A	YOLO26n	P3-P5	Native	Reference nano detector
B	YOLO26n-P2	P2-P5	Native	Effect of adding the high-resolution P2 branch
C	P2 + fixed fusion	P2-P5	Equal or fixed adjacent-scale fusion	Effect of multi-scale fusion without input-dependent weighting
D	P2 + AMFF	P2-P5	Input-dependent softmax scale weights	Incremental effect of adaptive fusion

2.6. Evaluation metrics, uncertainty, and computational efficiency

The primary detection outcome is mAP@0.50:0.95. Secondary outcomes are mAP@0.50, precision, recall, and per-class AP for all ten categories. A confusion matrix and precision-recall curves are generated only from executed predictions. If prediction files are available for both the baseline and AMFF models, an image-level paired bootstrap is used to estimate a 95% confidence interval for the difference in mAP@0.50:0.95. Resampling is performed at the image level so that all detections associated with a resampled image remain together.

Lightweight status is evaluated through trainable parameter count, GFLOPs at the same input resolution, serialized model size, and measured inference latency. Latency is measured with batch size 1 on a named device and backend after warm-up iterations are discarded. PyTorch and ONNX timings are

reported separately when both are executed. FPS is derived only from measured timing under the stated hardware and software conditions. The evaluation measures and their defensible reporting conditions are summarized in Table 4.

Table 4. Metrics and defensible reporting conditions.

Measure	Purpose	Reporting condition
mAP@0.50:0.95	Primary detection accuracy	Report only after validation or test evaluation
mAP@0.50	Secondary detection accuracy	Report only after evaluation
Precision / Recall	Error trade-off	Use the same confidence and IoU protocol across models
Per-class AP	Class-specific behaviour	Report all ten classes without selective omission
Parameters	Model-capacity proxy	Compute directly from model definition
GFLOPs	Compute-cost proxy	Use identical input size
Latency	Measured execution cost	Batch 1; fixed device/backend; warm-up discarded
FPS	Derived throughput	Report device/backend and avoid onboard generalization
Paired bootstrap 95% CI	Uncertainty in model difference	Resample evaluation images, not individual boxes

The complete empirical evaluation workflow is illustrated in Figure 6.

Empirical analysis workflow after dataset download and training



Figure 6. Empirical analysis workflow used to prevent test leakage and post hoc claim inflation.

2.7. Reproducibility and evidence boundary

The following data and information in the dataset should be archived together: Data configuration, data snapshot from the repository, random seeds, training arguments, model configuration, AMFF implementation, hardware identity, software versions, checkpoints, and evaluation outputs. Integrity audit must be performed prior to training to record: Disagreements regarding split sizes should be worked out prior to optimization. The analysis package provided does not include the approximately 2 GB image and label archives, trained AMFF checkpoints, prediction files, or hardware timing logs that are used for analysis, but are available upon request. The manuscript therefore only reports numbers for quantities which have been traced to the provided configuration or model definitions.

3. Results

3.1. Dataset characteristics and configured evaluation scope

There are 8629 images in the experimental subset defined. 74.99% is training, 6.35% is validation, and 18.66% is test-dev. The train/validation/test-dev split of 84.52% and the test-challenge split of 15.48% are relative to the 10,209 static images in the complete DET benchmark. The validation set consists of 548 images with 38,759 boxes, averaging to 70.73 boxes per image. This density follows the methodological focus on small and dense objects, but the package you receive doesn't include the label files necessary for an independent class-frequency analysis.

The label space is kept as it is 10 categories. This is important for interpretation because a smaller subset may have made the task easier, thus diminishing the comparability to the set benchmark. Figure 2, therefore, is non-quantitative in its categorical visualization. There is no claim from the visualization for a prevalence, nor is a size related to prevalence.

3.2. Source-backed architecture and computational findings

The architecture-level cost of introducing the P2 branch was quantified before training. The standard YOLO26n setup has 260 layers, 2,572,280 parameters and 6.1 GFLOPs. This variant has 329 layers, 2,662,400 parameters, and 9.5 GFLOPs. Thus, the P2 change brings in 69 layers, 90,120 parameters and 3.4 GFLOPs. When compared to its previous iteration, parameter count increases by just 3.50% while layer count increases by 26.54% and GFLOPs by 55.74%. The difference is of methodological significance. A compact detector may still need a significant number of arithmetic operations due to the increased number of features in the higher resolution feature maps of the new branch. The claim of a light weight should then not be made on the basis of the parameters. The architecture-level complexity differences are summarized in Table 5 and visualized in Figure 7.

Table 5. Source-backed complexity comparison of the standard and P2 nano configurations.

Configuration	Layers	Parameters	GFLOPs	Detection scales
YOLO26n	260	2,572,280	6.1	P3, P4, P5
YOLO26n-P2	329	2,662,400	9.5	P2, P3, P4, P5
Absolute change	+69	+90,120	+3.4	adds P2/4
Relative change	+26.54%	+3.50%	+55.74%	not applicable

P2 branch is parameter-light but compute-heavier

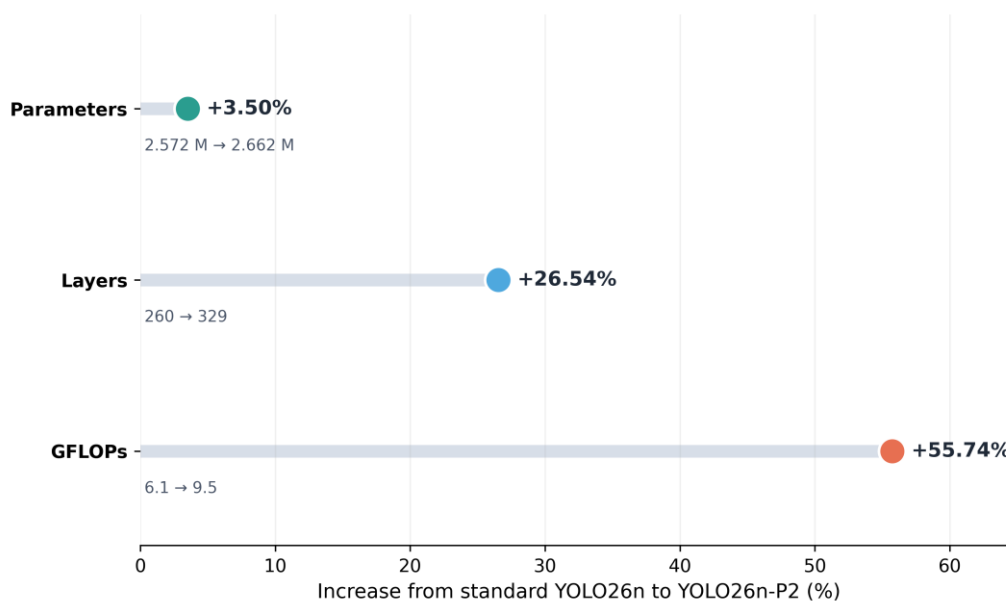


Figure 7. Relative architectural overhead of the supplied P2 nano model compared with the standard nano model.

3.3. AMFF effectiveness, ablation results, and unavailable outputs

The supplied files do not have the outputs for the configurations C and D in Table 3. Therefore, the following numerical reporting of AMFF improvement in mAP@0.50:0.95, mAP@0.50, precision, recall, or class-wise AP are not defensible. There is also the same lack of evidence for the other measures mentioned, such as confusion matrices, precision-recall curves, qualitative prediction panels, bootstrap confidence intervals and measured latency. These quantities are not null, but only supplied in the package. This boundary also means that there is no direct way to compare the accuracy of this detector with the accuracy of more recent UAV detectors [1-32]. The different generations of the YOLO, training durations, input resolutions, augmentation policies, device backends, and even different subsets of the VisDrone are used in published studies. It would not be a controlled experiment if their benchmark values were copied to the current result table. The appropriate way of comparing these four sets (A-D) is to train them all with the same protocol and interpret the incremental differences. The distinction between source-supported findings and claims requiring additional execution is summarized in Figure 8.

Evidence boundary for defensible manuscript claims

UAV imagery	required / supported	—	—	—
10-class detection	required / supported	—	—	—
Chinese benchmark context	required / supported	—	—	—
P2 multi-scale branch	required / supported	partial	—	—
Lightweight claim	partial	partial	required / supported	required / supported
Adaptive fusion benefit	—	required / supported	required / supported	partial
Accuracy improvement	—	partial	required / supported	—
Inference speed / FPS	—	—	partial	required / supported
	Supplied files	Architecture design	Training run	Hardware timing

Figure 8. Evidence matrix distinguishing source-supported claims from claims that require training runs or hardware timing.

4. Discussion

4.1. Interpretation in relation to recent UAV detection research

The present architecture analysis serves as a good warning for reading the lightweight UAV literature. Compact parameters, efficient convolutions, small-object heads, or selective fusion [2,3,6,16,23,26,31] are all recent papers that focus on them. All those directions are indeed sensible, but they are measuring different aspects of efficiency. The P2 result in this study is not enough to show that, a 55.74% up in GFLOPs corresponds to a 3.50% up in the parameters. An evaluation for deployment will then require at least the number of parameters, arithmetic cost, exported size, and latency for a specific device.

The proposed AMFF design is the most directly related to the studies that explicitly focus on the scale-adaptive or attention-enhanced feature fusion work [4,21,22,24,28,29,32]. In the current setup this distinction is evaluated experimentally rather than conceptually. Firstly, P2 is added, secondly, adjacent-scale fusion is added, and finally, input-dependent scale weighting is added. If D is found to be better than C when trained on the same data, then this may be because of adaptation, rather than the fusion of multi-scale features more specifically. There is little empirical justification for the extra gate, if it is similar to C and D.

The high resolution P2 branch has a good reason for aerial scenes as the spatial information is not as detailed as before the detection head. However, the computer output indicates that this option is not costless. The measured latency, class level behaviour and gain in the resulting mAP@0.50:0.95 is subject to

the impact of the additional resolution. The best future analysis would be to see if potential gains are focused in the categories that tend to be small in the image while larger categories do not get worse.

4.2. Practical implications, limitations, and future evaluation

The framework is limited to the task of VisDrone2019-DET for purpose. Performance on various UAV sensors, unseen countries, night imagery, infrared imagery and other aerial datasets are not established at this time. Additional testing on another UAV benchmark would be required before generalizations could be made. Likewise, on a workstation GPU or CPU, the timing is not necessarily representative of the performance of a particular onboard UAV unless it is tested on the embedded device itself.

The second drawback is the package is currently design-architecture complete but not experiment-architecture complete. No AMFF checkpoints and predictions were provided. This would prevent the manuscript from making claims for improved detection or an improved accuracy-efficiency trade-off. The restriction is not cosmetic as the study title specifies a framework, but substantive as the scientific value of the adaptive aspect is based on whether controlled training is supportive of it.

Future work should therefore implement the four steps ablation with fixed seeds, save the prediction files, calculate uncertainty of the prediction at the image level, outputs the final models to an announced back-end, and time the inference on both a reference workstation and an embedded platform if deployment claims are desired. External-dataset evaluation should be done after the VisDrone protocol is fixed, to prevent generalization evaluation from being a tuning source.

5. Conclusions

In this study, a lightweight UAV object-detection framework is defined that is based on a nano-scale detector, a P2 high-resolution branch and Adaptive Multi-Scale Feature Fusion across aligned P2-P5 features. The method is coupled with the public VisDrone2019-DET configuration and a baseline to AMFF ablation sequence controlled. It indicates that the source backed computational analysis has been carried out and the number of parameters have been increased from 2.572M to 2.662M, an increase of 3.50%, and the GFLOPs have been increased from 6.1 to 9.5, an increase of 55.74%. Therefore, the detection in high resolution is parameter-light (but also more expensive in terms of computational resources). No accuracy and latency outputs of AMFF have been performed, no mAP gain, real-time onboard performance or efficiency are claimed in this manuscript. The next obvious piece of evidence is the controlled implementation of the four configurations, according to a single VisDrone protocol, and then a device specific timing uncertainty analysis and image level uncertainty analysis. Claims do not extend beyond those specifically supported by the benchmark, model definitions, and available evidence.

Data Availability: This study uses a public VisDrone2019-DET dataset configuration. The dataset configuration and download/conversion script are available as [VisDrone.yaml on GitHub](#). Supporting framework documentation used with the supplied implementation is available in the [Ultralytics Chinese README on GitHub](#). The dataset configuration identifies the VisDrone project as the source and downloads the train, validation, and test-dev archives. No proprietary dataset was used.

Code Availability: The base training, validation, prediction, and export framework is publicly available from the Ultralytics repository. The supplied package documents the proposed AMFF design, but no separate public repository containing an executed custom AMFF implementation was provided. A permanent code archive should be provided if the authors release the final AMFF implementation.

References

1. Wang G, Chen Y, An P, Hong H, Hu J, Huang T. UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. *Sensors*. 2023;23:7190. DOI: 10.3390/s23167190.
2. Li Z, Wan X, Tan H, et al. Efficient-Lightweight YOLO: Improving Small Object Detection in YOLO for Aerial Images. *Sensors*. 2023;23:6423. DOI: 10.3390/s23146423.
3. Cao J, Bao W, Shang H, Yuan M, Cheng Q. GCL-YOLO: A GhostConv-Based Lightweight YOLO Network for UAV Small Object Detection. *Remote Sensing*. 2023;15:4932. DOI: 10.3390/rs15204932.
4. Wang Y, Zou H, Yin M, Zhang X. SMFF-YOLO: A Scale-Adaptive YOLO Algorithm with Multi-Level Feature Fusion for Object Detection in UAV Scenes. *Remote Sensing*. 2023;15:4580. DOI: 10.3390/rs15184580.
5. Zhao QY, Liu B, Lyu S, Wang C, Zhang HJ. TPH-YOLOv5++: Boosting Object Detection on Drone-Captured Scenarios with Cross-Layer Asymmetric Transformer. *Remote Sensing*. 2023;15:1687. DOI: 10.3390/rs15061687.
6. Deng L, Bi L, Li H, Chen H, Duan X, Lou H, Zhang H, Bi J, Liu H. Lightweight aerial image object detection algorithm based on improved YOLOv5s. *Scientific Reports*. 2023. DOI: 10.1038/s41598-023-34892-4.
7. Yang D, Tang L, Zhou X, Deng Y. A Lightweight Object Detector Based on Spatial-Coordinate Self-Attention for UAV Aerial Images. *Remote Sensing*. 2023;15:83. DOI: 10.3390/rs15010083.
8. Shang J, Wang J, Liu S, Wang C, Zheng B. Small Target Detection Algorithm for UAV Aerial Photography Based on Improved YOLOv5s. *Electronics*. 2023;12:2434. DOI: 10.3390/electronics12112434.
9. Faraji H, Chen B. Drone-YOLO: Improved YOLO for Small Object Detection in UAV. In: 2023 IEEE International Conference on Image, Vision and Computing (ICIVC). 2023. DOI: 10.1109/ICIVC58118.2023.10270571.
10. Xiong G, Qi J, Wang M, Wu C, Sun H. GCGE-YOLO: Improved YOLOv5s Algorithm for Object Detection in UAV Images. In: Chinese Control Conference. 2023. DOI: 10.23919/CCC58697.2023.10240567.
11. Wang C, Li Z, Gao Q, Cui T, Sun D, Jiang W. Lightweight and Efficient Air-to-Air Unmanned Aerial Vehicle Detection Neural Networks. 2023. DOI: 10.1109/ICUS58632.2023.10318340.
12. Zhou P, Wang P, Cao J, Qiyuan Y. PRO-YOLOv4-tiny: Towards More Balance Between Accuracy and Speed in the Detection of Small Targets in Remotely Sensed Images. *Remote Sensing Letters*. 2023. DOI: 10.1080/2150704X.2023.2254912.
13. Xu L, Zhao Y, Zhai Y, Huang L, Ruan C. Small Object Detection in UAV Images Based on YOLOv8n. *International Journal of Computational Intelligence Systems*. 2024. DOI: 10.1007/s44196-024-00632-3.
14. He Z, Cao L. SOD-YOLO: Small Object Detection Network for UAV Aerial Images. *IEEJ Transactions on Electrical and Electronic Engineering*. 2024. DOI: 10.1002/tee.24195.
15. Wang J, Li X, Chen J, Zhou L, Guo L, He Z, Zhou H, Zhang Z. DPH-YOLOv8: Improved YOLOv8 Based on Double Prediction Heads for the UAV Image Object Detection. *IEEE Transactions on Geoscience and Remote Sensing*. 2024. DOI: 10.1109/TGRS.2024.3487191.
16. Zhao ZL, Bo K, Hsu CY, Liao L. Lightweight unmanned aerial vehicle object detection algorithm based on improved YOLOv8. *Intelligent Data Analysis*. 2024. DOI: 10.3233/IDA-230929.
17. Dong D, Li J, Liu H, Deng L, Gu J, Liu L, Li S. EA-YOLO: An Efficient and Accurate UAV Image Object Detection Algorithm. *IEEJ Transactions on Electrical and Electronic Engineering*. 2024. DOI: 10.1002/tee.24180.
18. Ben Rouighi I, Chtioui H, Jegham I, Ben Khalifa A. YOLO Detectors for Drone-Based Real-Time Object Detection in Intelligent Transportation Systems: Comparative Study. 2024. DOI: 10.1109/CODIT62066.2024.10708153.
19. Lin Y, Liu Y. Improved YOLOv5 Algorithm for Small Object Detection in Drone Images. In: Springer proceedings. 2024. DOI: 10.1007/978-981-97-2092-7_18.
20. Zhang X, Bao D, Zhao H, Wang Q. Lightweight UAV target detection algorithm. *Journal of Physics: Conference Series*. 2024;2858:012030. DOI: 10.1088/1742-6596/2858/1/012030.
21. Zhang H. AeroYOLO: Efficient Multiscale and Attention-Augmented YOLOv8s for Robust Aerial Object Detection. *IEEE Access*. 2025. DOI: 10.1109/ACCESS.2025.3610617.
22. Nian Z, Yang W, Chen H. AEFFNet: Attention Enhanced Feature Fusion Network for Small Object Detection in UAV Imagery. *IEEE Access*. 2025. DOI: 10.1109/ACCESS.2025.3538873.
23. Ge P, Wan M, Qian W, Xu Y, Kong X, Gu G, Chen Q. SGA-YOLO: A Lightweight Real-Time Object Detection Network for UAV Infrared Images. *IEEE Transactions on Intelligent Transportation Systems*. 2025. DOI: 10.1109/TITS.2025.3616525.
24. Huang L, Yu X, Yan H, Huang L, An Z, Xu Y. AF-YOLO: Asymptotic Feature Extraction and Fusion for Aerial Object Detection. *IEEE Transactions on Circuits and Systems for Video Technology*. 2025. DOI: 10.1109/TCSVT.2025.3595740.

25. Dai R, Bi H, Han F, Tang J, Zhang C. DRS-YOLO: An Improved YOLOv8-Based Small Object Detection Model for UAV Aerial Imagery. *Measurement Science and Technology*. 2025. DOI: 10.1088/1361-6501/ae080a.
26. Liang S, Feng X, Xie M, Tang Q, Zhu H, Li G. Lightweight YOLO-SR: A Method for Small Object Detection in UAV Aerial Images. *Applied Sciences*. 2025;15:13063. DOI: 10.3390/app152413063.
27. Teng F, Wan Q, Zhu Z. NUAV-YOLO: A Lightweight Object Detection Algorithm Based on YOLOv8 for UAVs. 2025. DOI: 10.1117/12.3067230.
28. Ma Y, Wang H. Improved Multiscale Feature Fusion and Small Object Detection Layer Optimization for UAV Aerial Small Object Detection Based on YOLOv8. 2025. DOI: 10.1117/12.3068396.
29. Dong T, Xian J, Yu X. Lightweight Multi-Scale Feature Aggregation for Small Object Detection in Remote Sensing Images. 2025. DOI: 10.1109/AUTEEE67053.2025.11322339.
30. Habash N, Abu Alqumsan AM, Zhou T. Recent Real-Time Aerial Object Detection Approaches, Performance, Optimization, and Efficient Design Trends for Onboard Performance: A Survey. *Sensors*. 2025;25:7563. DOI: 10.3390/s25247563.
31. Khan MW, Obaidat MS, Mahmood K, Batool D, Badar HMS, Aamir M, et al. Real-Time Road Damage Detection and Infrastructure Evaluation Leveraging Unmanned Aerial Vehicles and Tiny Machine Learning. *IEEE Internet of Things Journal*. 2024;11(12):21347–21358
32. Khan MW, Mahmood K, Hussain I, Talha Qureshi M, Badar HMS, Gao W. Deep Learning Approaches for Road Damage Detection Using YOLOv8 and Custom Convolutional Neural Network. In: 2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC). 2024;284–290. DOI: 10.1109/ICFTIC64248.2024.10913309.
33. Khan MW, Obaidat MS, Mahmood K, Sadoun B, Badar HMS, Gao W. Real-Time Road Damage Detection Using an Optimized YOLOv9s-Fusion in IoT Infrastructure. *IEEE Internet of Things Journal*. 2025. DOI: 10.1109/JIOT.2025.3528726.
34. Badar HMS, Hussain I, Bashir AK, Alturki N, Fan G, Zhang C. Edge-Optimized Lightweight and Transformer Backbones for Real-Time Road Damage Detection in IIoT Systems. *IEEE Internet of Things Journal*. 2026;13(10):19893–19904. DOI: 10.1109/JIOT.2025.3644661
35. He Z, She R, Tan B, Li J. SSCW-YOLO: A Lightweight and High-Precision Model for Small Object Detection in UAV Scenarios. *Drones*. 2026;10:41. DOI: 10.3390/drones10010041.
36. Gong J, Yuan Z, Li W, Li W, Guo Y, Guo B. A Lightweight Upsampling and Cross-Modal Feature Fusion-Based Algorithm for Small-Object Detection in UAV Imagery. *Electronics*. 2026;15:298. DOI: 10.3390/electronics15020298.
37. Zhu P, Wen L, Du D, Bian X, Fan H, Hu Q, Ling H. 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.