



ARTICLE

TAEPM-Net: An Accurate and Lightweight Detector for Multi-Class Small Object Detection in UAV Imagery

Xinbin Hu^{1,2}, Mohd Shahrizal Sunar^{1,2,*} and Norhaida Mohd Suaib¹

¹Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia

²Media and Game Innovation Centre of Excellence, Institute of Human Centered Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia

*Corresponding Author: Mohd Shahrizal Sunar. Email: shahrizal@utm.my

Received: 30 May 2026; Accepted: 24 July 2026; Published: 22 October 2026

ABSTRACT: Unmanned aerial vehicle (UAV) imagery has made pedestrian and multi-class small-object detection a critical perception task, yet it remains difficult because targets often occupy fewer than 32×32 pixels. High accuracy on such targets is usually obtained with parameter-heavy models, whereas the compact detectors suited to a UAV's limited memory budget tend to sacrifice accuracy, leaving a persistent accuracy-compactness gap. To narrow this gap, we propose TAEPM-Net, a lightweight detector built on YOLOv11n that raises small-object accuracy while keeping the parameter count low. TAEPM-Net introduces three components: a Partial Multi-Scale Feature Aggregation module (CSPMA) that replaces the C3k2 backbone block and supplies parallel multi-scale receptive fields with residual fusion; a Task-Aligned Dynamic Detection Head (TALDH) that couples classification and localization on a shared spatial representation through deformable convolutions (DCNv2) and a layer-level attention mechanism, reducing feature conflict on densely packed small targets while using fewer head parameters; and an Efficient Up-Convolution Block (EUCB) in the neck that learns to reconstruct spatial detail during skip-connection upsampling. On the VisDrone-2019 benchmark, TAEPM-Net attains an mAP@0.5 of 37.3% and an mAP@0.5:0.95 of 22.3%, exceeding the YOLOv11n baseline by 4.4 and 3.1 percentage points while reducing the parameter count by about 10% to 2.33×10^6 . Against twelve representative detectors, it achieves the highest accuracy among nano-scale models while remaining more compact than the YOLOv11n baseline, offering a favorable accuracy-parameter trade-off for UAV pedestrian and multi-class small-object detection.

KEYWORDS: Small object detection; UAV imagery; multi-class detection; lightweight network; YOLOv11

1 Introduction

The rapid spread of low-altitude aerial platforms has reshaped the perception requirements of unmanned aerial vehicles (UAVs). Early work on UAV detection pursued raw accuracy under laboratory conditions, where model size was of little concern. As applications broadened to urban crowd monitoring, disaster relief, precision agriculture, and border surveillance, however, detectors are increasingly expected to run within a tight parameter and memory budget. This shift exposes a gap that current lightweight UAV detectors have not closed: no existing nano-scale detector simultaneously attains competitive accuracy on densely packed sub- 32×32 pedestrians and multi-class small objects, as found in VisDrone [1], while keeping the parameter count compact.

In recent years, deep learning-based object detection has advanced rapidly, spanning two-stage detectors such as (Faster Regions with Convolutional Neural Networks) Faster R-CNN [2] and one-stage detectors exemplified by the YOLO series. Transformer-based models, including Deformable DETR [3], (DETR with Improved deNoising anchor boxes) DINO [4], and RT-DETR [5], achieve strong accuracy, but their global self-attention imposes heavy computation and long training, making them ill-suited to resource-constrained UAV platforms. The YOLO series therefore remains the most practical framework for drone applications; YOLOv11 [6] in particular offers strong feature extraction with a streamlined design and has been quickly adapted for aerial tasks, e.g., YOLO11s-UAV [7] and YOLO-Drone [8]. Despite this progress, an accuracy-compactness gap persists: highly accurate detectors for sub- 32×32 UAV pedestrians (e.g., Faster R-CNN) require hundreds of GFLOPs, whereas compact detectors such as YOLOv11n remain limited in accuracy. Adding high-resolution branches or attention modules to lightweight backbones tends to enlarge the model again, restoring the trade-off. This indicates a structural limitation that must be addressed at the feature-representation level.

Under the standard YOLOv11 design, small pedestrians and multi-class targets in UAV imagery are hard to detect for three reasons. First, the repeated downsampling in the backbone progressively suppresses the already weak semantic features of small targets, causing irreversible information loss. Second, the default nearest-neighbor upsampling in the neck cannot reconstruct fine spatial detail, often producing aliasing and blurred object boundaries. Third, the conventional decoupled head processes classification and regression in separate streams; for small targets, even a slight spatial misalignment between the two branches causes severe feature conflict, raising both false negatives and false positives.

To address these limitations, we propose TAEPM-Net, a lightweight and accurate detector for pedestrian and multi-class small-object detection built on the YOLOv11 framework. We jointly optimize the backbone, the neck upsampling, and the detection head from the perspective of small-object detection.

The primary contributions of this paper are as follows:

- We propose CSPMA (Partial Multi-Scale Feature Aggregation), a feature-extraction module that replaces the C3k2 block in the backbone. CSPMA aggregates multi-scale features and fuses them through a residual structure, strengthening the representation of small pedestrians and multi-class targets at only a marginal parameter cost. The module is detailed in [Section 3.3](#).
- We design TALDH (Task-Aligned Dynamic Detection Head) to replace the conventional decoupled head. Building on the task-alignment idea of TOOD [9] and modulated deformable convolution (DCNv2) [10], TALDH couples classification and localization on a shared spatial representation and adds a layer-level attention mechanism so that both branches focus on the same informative locations, mitigating the feature conflict that is especially harmful for small aerial targets. By removing the redundant layers of the decoupled head, TALDH also lowers the parameter count. The mechanism is detailed in [Section 3.4](#).
- We introduce the Efficient Up-Convolution Block (EUCB) into the neck, a lightweight upsampling module inspired by recent learnable-upsampling operators [11]. EUCB performs learnable upsampling and uses depthwise convolution (DWC) to match the resolution required by the skip connections, recovering spatial detail lost during downsampling. Because it relies on depthwise rather than standard convolutions, it adds little parameter and computation overhead while preserving representational capacity. The implementation is described in [Section 3.5](#).

2 Related Works

2.1 Small Object Detection in UAV Imagery

Pedestrian and multi-class small-object detection in UAV imagery is challenging due to variation, dense crowding, and arbitrary viewing angles. Traditional detectors based on hand-crafted features are unsuitable here because of the severe background clutter in airborne imagery. Deep learning models built on CNNs, together with lightweight detectors such as EfficientDet [12], have therefore become the dominant paradigm. The YOLO series in particular has evolved from anchor-free design of YOLOX [13] and the decoupled head of YOLOv6 [14] to the efficient layer aggregation of YOLOv7 [15], steadily improving the speed-accuracy balance.

To handle the tiny scale of UAV targets, many task-specific improvements have been proposed: the Normalized Wasserstein Distance (NWD) [16] as a small-object regression loss, the Slicing Aided Hyper Inference (SAHI) framework [17] for high-resolution inputs, TPH-YOLOv5 [18] with transformer prediction heads for dense scenes, and YOLO-MS [19] with a multi-scale design. Recent YOLO versions continue this trend: YOLOv9 and YOLOv10 introduced Programmable Gradient Information and NMS-free training, YOLOv11 [6] refined the CSP bottleneck with C3k2 modules, YOLOv12 added attention, and YOLOv26 simplified the architecture. Yet applying these general-purpose detectors directly to UAV pedestrian and multi-class detection still yields suboptimal results, as their accuracy gains usually come from enlarging the model. A lightweight yet accurate architecture tailored to aerial small-object detection is therefore still lacking.

2.2 Multi-Scale Feature Fusion and Upsampling

Multi-scale feature extraction and fusion are critical for detecting small pedestrians. Combining deep semantic information with shallow spatial detail, the Feature Pyramid Network (FPN) [20] and the Path Aggregation Network (PANet) [21] have become standard practice in the YOLO series. However, repeated convolution and downsampling in the backbone still erode fine-grained information, and modules such as C2f and C3k2 spend a disproportionate share of parameters on low-resolution, semantically weak features. We therefore introduce CSPMA for partial multi-scale aggregation with a few additional parameters.

Upsampling in the neck is equally important. Typical nearest-neighbor upsampling simply copies pixel values and cannot restore the spatial detail that detection requires. Content-aware upsamplers such as CARAFE [22] and DySample [23] improve on this but add considerable computation, and general learnable upsampling operators such as FADE [11] have further advanced encoder-decoder feature reconstruction. Motivated by these designs, we adopt a lightweight Efficient Up-Convolution Block (EUCB) that uses depthwise convolution (DWC) for efficient resolution matching along skip connections, recovering spatial detail for small pedestrian and multi-class targets at little additional cost.

2.3 Detection Heads and Task Alignment

The detection head produces the final object-class and bounding-box predictions. Early YOLO versions used a single coupled head that handled the two tasks together, which limited accuracy. To relieve this, the most recent detectors adopt a decoupled head that predicts classification and regression through separate branches.

Although effective for general detection, decoupled heads introduce a spatial mismatch that is especially harmful for small targets. Classification favours translation invariance (whether an object is present), whereas localization requires translation variance (where it is). For small targets, the chance that a given location corresponds to a true positive is very low, and mismatched assignments are treated as background. To address this, Feng et al. proposed TOOD (task-aligned one-stage object detection) [9], which explicitly aligns the spatial locations of the two tasks.

In parallel, dynamic convolutions such as DCNv2 [10] have proven effective at adaptively adjusting the receptive field to fit irregular target shapes, while dense-crowd detection also suffers from class imbalance and background interference, as first highlighted by Focal Loss [24]. Building on these ideas, we design TALDH, which combines dynamic convolution with layer attention to steer the network toward the most informative spatial regions, thereby reducing the classification-localization conflict for small aerial objects.

3 Methods

Pedestrian and multi-class small-object detection in drone imagery suffers from severe scale variation, complex backgrounds, and dense occlusion. To address these challenges, we build TAEPM-Net upon the YOLOv11 framework for more precise small-object detection. We first review the baseline model structure, and then present specific modifications to each part of TAEPM-Net: the backbone, the neck, and the detection head.

3.1 Baseline YOLOv11 Architecture

We adopt YOLOv11 as the baseline architecture. It follows the standard one-stage detection paradigm and consists of three parts: a backbone that extracts features, a neck that fuses features across scales, and a decoupled head that produces the predictions. Although YOLOv11 introduces a C3k2 module (an optimized Cross Stage Partial bottleneck) that improves gradient flow, it is still designed mainly for general-size targets, and applying it directly to UAV small objects reveals three shortcomings. First, the repeated convolutions in the backbone severely weaken the already faint texture of small pedestrians. Second, the default nearest-neighbor upsampling in the neck fails to restore fine spatial edges. Third, the decoupled head handles classification and regression on spatially unsynchronized branches, which is particularly harmful for small objects. The YOLOv11 architecture is shown in Fig. 1.

3.2 Overview of TAEPM-Net

To address the challenges discussed above, we build upon the nano variant of YOLOv11 (YOLOv11n) and design a new lightweight network with three targeted improvements, which we term TAEPM-Net. We choose the nano scale because UAV platforms are resource-constrained, and our goal is to raise accuracy on small objects without inflating the model size.

First, in the backbone we replace the C3k2 modules with the proposed CSPMA (Partial Multi-Scale Feature Aggregation) module to strengthen multi-scale representation at a marginal parameter cost. Second, we replace the conventional decoupled head with the proposed TALDH (Task-Aligned Dynamic Detection Head), which aligns classification and localization and reduces feature interference. Finally, in the neck we adopt a lightweight Efficient Up-Convolution Block (EUCB), inspired by recent learnable-upsampling operators [11], to replace the nearest-neighbor upsampling and recover the spatial detail of small objects more accurately. The overall architecture of TAEPM-Net is shown in Fig. 2.

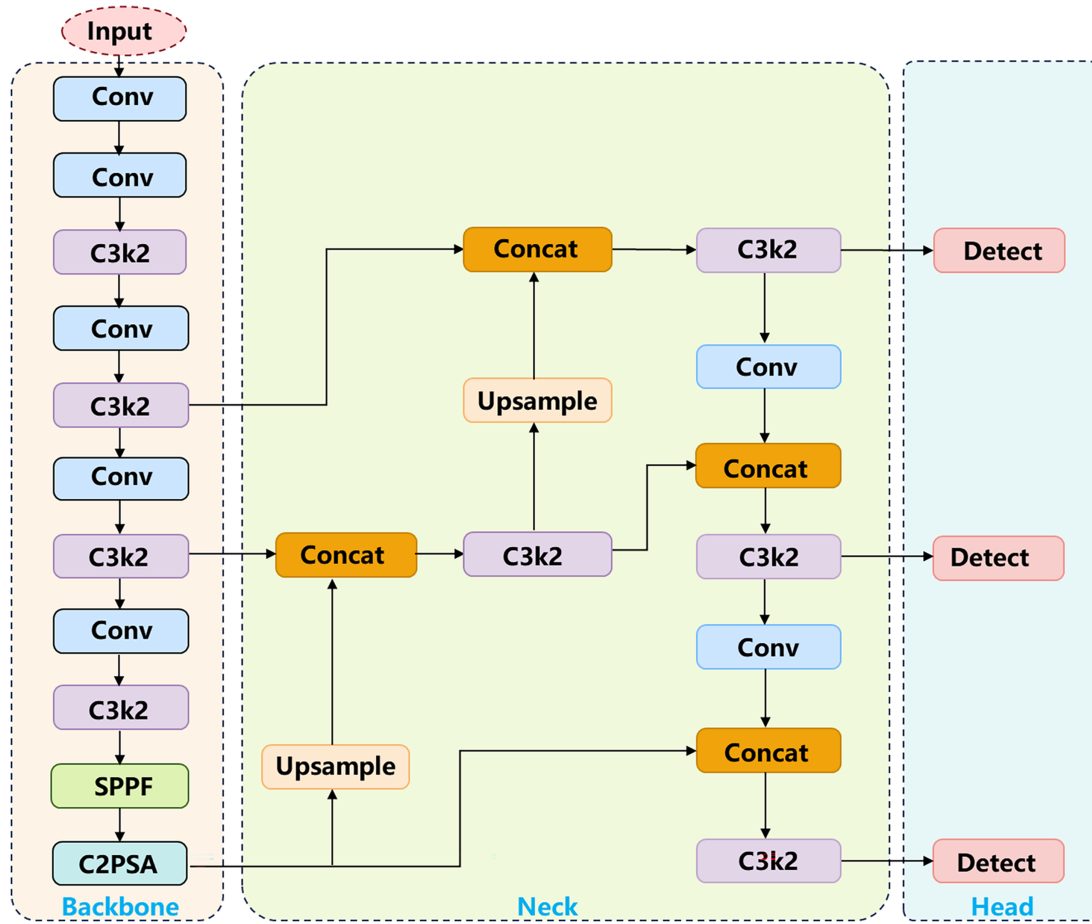


Figure 1: The architecture of YOLOv11.

3.3 CSPMA: Partial Multi-Scale Feature Aggregation

In UAV imagery, small targets such as pedestrians may occupy only about 10×10 pixels, so standard convolutions struggle to extract reliable features for them.

YOLOv11 extracts features mainly through its C3k2 module, which relies on a chain of sequential bottlenecks. This sequential design behaves like a low-pass filter: the weak, low-resolution features of small pedestrians are gradually weakened layer by layer and are largely lost before reaching the detection head. To avoid this, CSPMA is designed to capture features at several scales at the same time, rather than passing them down one layer after another.

The architecture of CSPMA is shown in Fig. 3. A 3×3 convolution first mixes all channels, and the result is split into two halves; one half is refined by a 5×5 depthwise convolution and split again, and one of the resulting quarters is further refined by a 7×7 depthwise convolution. The three scales (7×7 , 5×5 , and 3×3 branches) are concatenated back to the original number of channels, fused by a 1×1 convolution, and added to the input via a residual connection. Every convolution uses stride 1 with padding equal to half the kernel size (1, 2, and 3 for the 3×3 , 5×5 , and 7×7 kernels), so all branches keep the same spatial size and can be concatenated and added without mismatch. Because the larger 5×5 and 7×7 receptive fields use depthwise convolutions, CSPMA enlarges the receptive field at only a small parameter and computation cost. Eqs. (1) and (2) summarize this process.

$$X1, X2 = \text{Split}(\text{Conv}3 \times 3(X)), X3, X4 = \text{Split}(\text{DWConv}5 \times 5(X1)) \quad (1)$$

$$Y = \text{Conv}1 \times 1(\text{Concat}[\text{DWConv}7 \times 7(X3), X4, X2]) + X \quad (2)$$

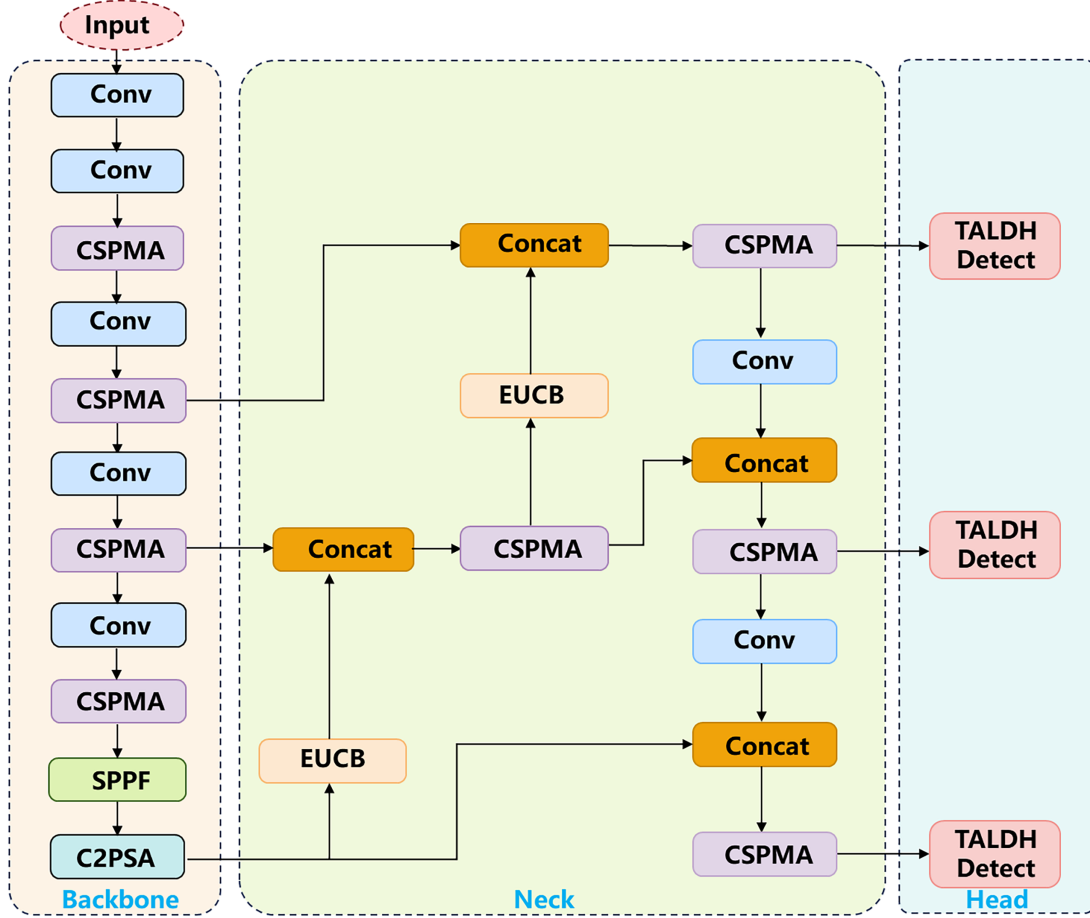


Figure 2: The architecture of TAEPM-Net.

In summary, CSPMA captures multi-scale small-object features while remaining compact, and replacing C3k2 with it keeps the model size almost unchanged while providing a richer representation.

3.4 TALDH: Task-Aligned Dynamic Detection Head

The separation head in the current YOLOv11 structure has drawbacks when used for UAV image data. It forcibly splits the feature map into two independent branches that cannot exchange information: one branch predicts the object class, while the other predicts the bounding-box coordinates. For large, common object categories, this decoupled strategy performs adequately; however, for sub-pixel aerial targets, it introduces spatial misalignment that severely degrades detection accuracy. Classification favours translation invariance (whether an object is present), whereas localization requires translation variance (where exactly it is). These two tasks are inherently conflicting in their spatial requirements. A spatial offset of as little as two pixels causes the classification branch to respond to background texture rather than the target silhouette, yielding a bounding box centered on an empty region. This spatial misalignment is the primary cause of the elevated false-positive rates observed in baseline models.

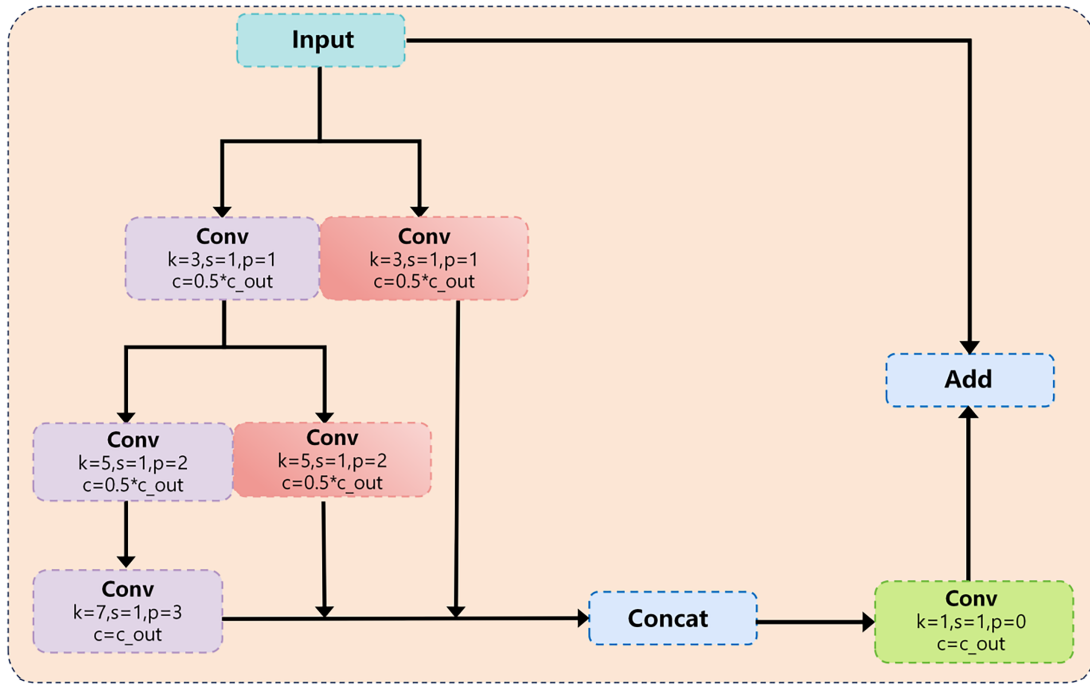


Figure 3: The proposed CSPMA module.

To remove this bottleneck, TALDH forces the classification and localization branches to operate on a shared spatial representation before they are decoupled. As shown in Fig. 4, at each detection scale the input feature is first passed through a shared stem of two 3×3 group-normalized convolutions; the outputs of the two layers are concatenated to form an inter-level feature F in $\mathbb{R}^{(2C \times H \times W)}$, where $C = \text{hidc}/2$ is the head width. This shared feature F is the common source from which both the alignment offsets and the task-specific features are derived, so the two tasks are spatially consistent by construction.

On top of this shared feature, the two tasks are aligned separately. For localization, a 3×3 convolution predicts a set of sampling offsets and modulation weights, which drive a deformable convolution (DCNv2) so that the regression branch samples exactly the region covered by each small target (Eqs. (3) and (4)). For classification, a lightweight branch (1×1 conv, ReLU, 3×3 conv, sigmoid) produces a spatial weight map that re-weights the classification feature, emphasizing the locations that truly contain an object (Eq. (5)). In addition, a task-decomposition module applies a simple channel attention (global pooling followed by a small bottleneck) to each branch, so that the classification and localization branches focus on the same informative regions. Together, these steps let both tasks attend to consistent spatial locations, which is exactly what small aerial targets need.

$$O, M = \text{Split}(\text{Conv}_{3 \times 3}(F)), M = \text{sigmoid}(M) \quad (3)$$

$$F_{\text{reg}} = \text{GN}(\text{DCNv2}(F_{\text{reg}}, O, M)) \quad (4)$$

$$P_{\text{cls}} = \text{sigmoid}(\text{Conv}_{3 \times 3}(\text{ReLU}(\text{Conv}_{1 \times 1}(F)))), F_{\text{cls_out}} = F_{\text{cls}} * P_{\text{cls}} \quad (5)$$

Because the shared stem, the deformable regression branch and the classification branch all draw on the same aligned feature, TALDH suppresses background interference and reduces the classification-localization conflict that is especially harmful for small aerial targets. Compared with the conventional decoupled head of YOLOv11, TALDH replaces the two separate stacks of standard convolutions with a single shared stem plus

lightweight alignment branches, which also lowers the parameter count. The complete structure of TALDH is shown in Fig. 4.

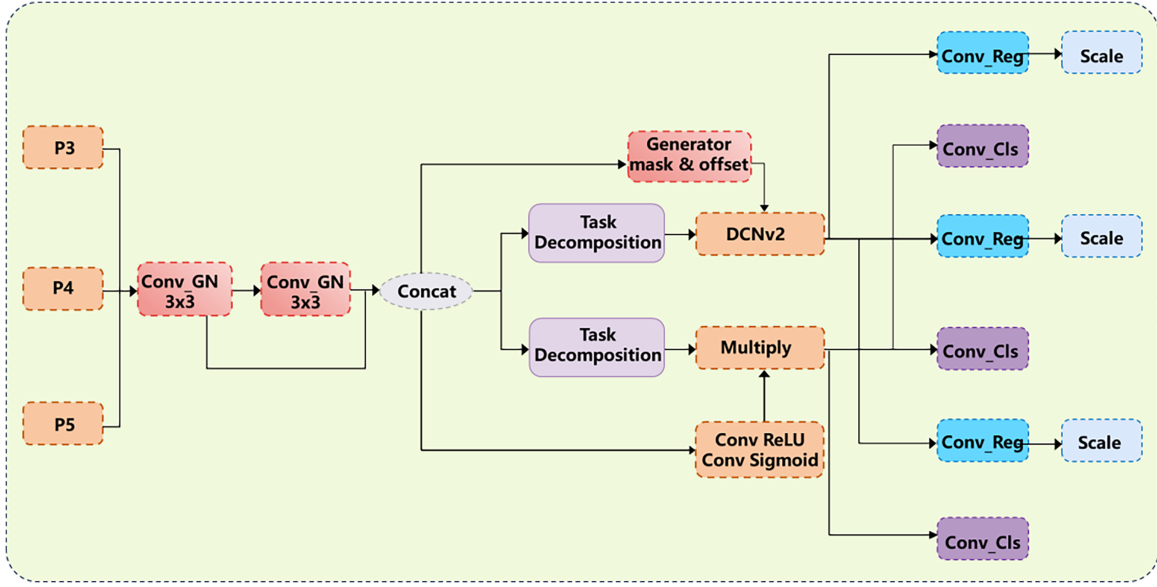


Figure 4: The proposed TALDH module.

3.5 EUCB Upsampling Module

Downsampling in deep networks inevitably discards high-frequency spatial detail. YOLOv11 mitigates this only partially, using nearest-neighbor interpolation at the upsampling stage, which simply replicates pixels to double the resolution. This coarse resizing is acceptable for large objects, but for small and heavily occluded pedestrians it blurs the very boundaries that matter, producing jagged and ambiguous edges.

We therefore replace the default upsampler with the Efficient Up-Convolution Block (EUCB) in the neck. Unlike fixed interpolation, EUCB not only enlarges the feature map but also learns to reconstruct the fine spatial detail lost during downsampling.

The internal structure of EUCB is shown in Fig. 5. Given a low-resolution feature map, EUCB first doubles the spatial resolution by nearest-neighbor upsampling, then refines it with a 3×3 depthwise convolution (with groups equal to the number of channels) that learns fine spatial detail while adding almost no parameters. A channel-shuffle operation then promotes information exchange across the depthwise groups, and a final 1×1 pointwise convolution mixes the channels and matches the depth of the corresponding backbone skip connection. The channel count is left unchanged, so only the spatial resolution is doubled. Because the refinement uses a depthwise separable design rather than a dense convolution, the extra cost is small.

The EUCB calculation method is as follows:

$$\text{EUCB}(X) = \text{Conv1} \times 1(\text{Shuffle}(\text{DWConv3} \times 3(\text{Upsample_2} \times (X)))) \quad (6)$$



Figure 5: The EUCB module.

4 Experiments

To evaluate TAEPM-Net for UAV-based multi-class small-object detection, we conducted extensive experiments. This section describes the dataset, the experimental environment, and the evaluation metrics, and then presents the ablation studies, per-class and per-scale analyses across input resolutions, comparison experiments, and qualitative visualizations.

4.1 Datasets

VisDrone-2019 is a publicly available UAV detection benchmark from the AISKYEYE team at Tianjin University, containing 10,209 images with more than 2.6 million annotated bounding boxes over ten categories: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. Like other challenging aerial benchmarks such as UAVDT [25], it is characterized by severe scale variation and heavy occlusion. Following the COCO convention, objects with an area below 32×32 pixels are treated as small, between 32×32 and 96×96 as medium, and above 96×96 as large. Because most VisDrone targets, especially the pedestrian and people classes, fall into the small range, the benchmark is heavily biased toward small-object detection. We follow the official partition: the training set (6471 images) is used for training, the validation set (548 images) for all quantitative evaluation (Tables 1–6), and the test-dev set (1610 images) only for the qualitative visualizations in the visualization analysis. The subsets are disjoint, so all reported results are on data not seen during training.

Table 1: Object size distribution of the VisDrone-2019 validation set (COCO scale bins).

Class	Small	Medium	Large	Total
Pedestrian	7746	1075	23	8844
People	4631	489	5	5125
Bicycle	1053	231	3	1287
Car	6910	6375	779	14,064
Van	937	923	115	1975
Truck	289	374	87	750
Tricycle	621	412	12	1045
Awning-tricycle	305	222	5	532
Bus	116	103	32	251
Motor	3978	901	7	4886
All	26,586	11,105	1068	38,759

Table 2: Ablation experiments.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Precision (%)	Recall (%)	Params (M)	GFLOPs (G)
YOLOv11n (Baseline)	32.9	19.2	43.4	33.2	2.58	6.3
YOLOv11n+EUCB	33.3	19.4	44.8	32.9	2.67	6.8
YOLOv11n+TALDH	35.3	21.0	45.6	35.0	2.20	7.9
YOLOv11n+CSPMA	34.5	20.3	45.8	34.4	2.63	7.6
YOLOv11n+TALDH+EUCB	36.1	21.3	46.2	35.9	2.27	8.1
YOLOv11n+CSPMA+EUCB	34.9	20.5	44.9	35.2	2.70	7.9
YOLOv11n+CSPMA+TALDH	36.9	22.0	47.2	36.4	2.26	9.2
YOLOv11n+CSPMA+TALDH+EUCB (Ours)	37.3	22.3	48.2	37.2	2.33	9.5

Table 3: Per-class per-scale AP at 640 input (COCO AP; B = YOLOv11n baseline, O = TAEPM-Net).

Class	AP_S(B)	AP_S(O)	AP_M(B)	AP_M(O)	AP_L(B)	AP_L(O)
Pedestrian	0.100	0.127	0.333	0.405	0.490	0.614
People	0.073	0.092	0.138	0.196	0.018	0.011
Bicycle	0.017	0.025	0.079	0.135	0.003	0.003
Car	0.281	0.317	0.653	0.693	0.828	0.850
Van	0.093	0.116	0.375	0.451	0.488	0.554
Truck	0.055	0.070	0.213	0.250	0.435	0.550
Tricycle	0.073	0.089	0.164	0.182	0.347	0.423
Awning-tricycle	0.050	0.052	0.110	0.108	0.187	0.136
Bus	0.084	0.140	0.415	0.468	0.667	0.811
Motor	0.111	0.140	0.231	0.290	0.121	0.213
All	0.094	0.117	0.271	0.318	0.358	0.416

Table 4: Per-class per-scale AP at 1024 input (COCO AP; B = YOLOv11n baseline, O = TAEPM-Net).

Class	AP_S(B)	AP_S(O)	AP_M(B)	AP_M(O)	AP_L(B)	AP_L(O)
Pedestrian	0.187	0.208	0.438	0.419	0.605	0.421
People	0.135	0.136	0.230	0.191	0.036	0.009
Bicycle	0.054	0.051	0.187	0.127	0.034	0.000
Car	0.393	0.416	0.715	0.722	0.849	0.837
Van	0.159	0.189	0.475	0.482	0.513	0.517
Truck	0.091	0.112	0.299	0.266	0.521	0.419
Tricycle	0.135	0.131	0.204	0.212	0.425	0.342
Awning-tricycle	0.073	0.088	0.154	0.119	0.153	0.062

(Continued)

Table 4 (continued)

Class	AP_S(B)	AP_S(O)	AP_M(B)	AP_M(O)	AP_L(B)	AP_L(O)
Bus	0.193	0.262	0.525	0.532	0.751	0.743
Motor	0.199	0.205	0.349	0.325	0.387	0.061
All	0.162	0.180	0.357	0.339	0.427	0.341

Table 5: Effect of input resolution on the YOLOv11n baseline (VisDrone-2019 val).

Model	Input	mAP@0.5	mAP@0.5:0.95	GFLOPs
YOLOv11n	640	32.9	19.2	6.3
YOLOv11n	1024	33.1	19.3	16.5
TAEPM-Net (Ours)	640	37.3	22.3	9.5

Table 6: Comparison experiments.

Model	mAP@0.5 (%)	mAP@ 0.5:0.95 (%)	Params (M)	GFLOPs (G)
Faster-RCNN [2]	37.4	22	41.2	208
Cascade-RCNN [26]	39	24.2	69.1	236
D-FINE-HGNETv2 [27]	36.2	20.1	3.7	7.1
DEIM-HGNETv2-n [28]	36.1	20.1	3.7	7.1
YOLOv5n	20.5	9.9	1.77	4.2
YOLOv5s	25.7	13.0	7.04	15.8
YOLOv8n	32.4	18.7	3.0	8.1
YOLOv9-T [29]	33.7	19.5	2.0	7.6
YOLOv10n [30]	32.1	18.6	2.7	8.2
YOLOv11n (Baseline)	32.9	19.2	2.58	6.3
YOLOv12n [31]	33.5	19.6	2.6	6.3
YOLOv26n [32]	30.5	16.9	2.4	5.2
TAEPM-Net (Ours)	37.3	22.3	2.33	9.5

Table 1 summarizes the object size distribution of the validation set under the COCO scale definition. Small objects dominate the benchmark (68.6% of all instances), and the pedestrian and people classes are almost entirely small (87.6% and 90.4%, respectively), which makes small-object accuracy the decisive metric for this dataset.

4.2 Experimental Environment

To ensure fairness and reproducibility, all models—including the baseline and the proposed TAEPM-Net—were trained and evaluated under the exact same hardware and software conditions.

Hardware configuration: The experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 4090 GPU (24 GB VRAM) and an Intel Core i9 processor, running the Ubuntu 22.04 operating system.

Software environment: The models were implemented using Python 3.9 and the PyTorch 2.1 deep learning framework, accelerated by CUDA 12.1.

Hyperparameter settings: The baseline model was YOLOv11n. During the training phase, the input image resolution was uniformly resized to 640×640 . The models were trained from scratch for 250 epochs using the SGD (Stochastic Gradient Descent) optimizer, with an initial learning rate of 0.01 and a batch size of 16. To keep the ablation strictly controlled, every ablation variant and the full model share exactly the same training protocol (640 input, 250 epochs, SGD, batch 16, official train/val split); the only difference between rows in [Table 2](#) is which module is enabled.

4.3 Evaluation Metrics

We evaluate performance using three metrics: mean Average Precision (mAP), the total parameter count, and GFLOPs. Precision and Recall, computed from True Positives (TP), False Positives (FP), and False Negatives (FN), form the basis of the mAP computation. Precision measures how many of the predicted detections are correct and thus reflects the false-detection rate, whereas Recall measures how many of the true targets are found and thus reflects the miss rate. Their definitions are given in [Eqs. \(7\) and \(8\)](#).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (7)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (8)$$

Average Precision (AP) measures the detection quality for an individual target category, as defined in [Eq. \(9\)](#). Evaluating an aerial detector, however, requires a global perspective: mean Average Precision (mAP) aggregates the AP scores across all K categories and computes their arithmetic mean, as formalized in [Eq. \(10\)](#). A higher mAP indicates a more robust network that accurately localizes targets across diverse classes.

$$\text{AP} = \int_0^1 \text{Precision}(\text{Recall}) d\text{Recall} \quad (9)$$

$$\text{mAP} = \frac{\sum_{k=1}^K \text{AP}_k}{K} \quad (10)$$

We also report the total parameter count and GFLOPs. The parameter count reflects model size and storage cost, and a lower value indicates a more lightweight model; GFLOPs measures the computation of a single forward pass. Reporting both alongside mAP lets accuracy be interpreted together with model complexity.

4.4 Ablation Experiments

To validate the effectiveness and necessity of the integrated EUCB module and our proposed TALDH and CSPMA modules, detailed ablation studies were conducted based on the YOLOv11n baseline. The quantitative results are presented in [Table 2](#).

As shown in [Table 2](#), YOLOv11n baseline achieves an mAP@0.5 of 32.9% with 2.58M parameters.

- (1) Effect of EUCB: EUCB replaces the nearest-neighbor upsampling with a learnable depthwise up-convolution. Adding it to the CSPMA + TALDH model raises mAP@0.5 from 36.9% to 37.3% and mAP@0.5:0.95 from 22.0% to 22.3% ([Table 2](#)); a consistent gain is also observed when EUCB is added to the baseline alone (mAP@0.5 32.9% to 33.3%, mAP@0.5:0.95 19.2% to 19.4%). This confirms that recovering the fine spatial detail lost during downsampling benefits detection, which is exactly the regime that matters for densely packed aerial objects. Because the spatial reconstruction is realized with depthwise separable convolutions, this gain is obtained at only a small parameter and computation cost (2.26M to 2.33M parameters, 9.2G to 9.5G).
- (2) Effect of TALDH: Applying TALDH alone raises mAP@0.5 from 32.9% to 35.3% and mAP@0.5:0.95 to 21.0%, while reducing the parameter count from 2.58M to 2.20M (about 8.0 GFLOPs). By aligning the classification and regression tasks on a shared spatial representation and removing the redundant layers of the conventional decoupled head, TALDH alleviates the classification-localization conflict on small targets and simultaneously lowers the parameter count.
- (3) Effect of CSPMA: Replacing the C3k2 backbone modules with the proposed CSPMA improves mAP@0.5 to 34.5% and mAP@0.5:0.95 to 20.3%, at a marginal parameter change (2.63M). This indicates that the partial multi-scale aggregation strategy yields richer gradient flow and stronger multi-scale features than the standard CSP bottleneck, with essentially no change in model size.
- (4) Combination effect: Combining all three modules (CSPMA + TALDH + EUCB) yields the best overall performance. TAEPM-Net reaches an mAP@0.5 of 37.3% (an increase of 4.4 percentage points over the baseline) and an mAP@0.5:0.95 of 22.3% (+3.1 points). Owing to the lightweight design of TALDH, the total parameter count drops to 2.33M (about 10% lighter than the baseline 2.58M), while GFLOPs are 9.5G. This shows that TAEPM-Net achieves a favorable accuracy-parameter trade-off.

4.5 Per-Class Per-Scale Analysis

To address the small-object performance at the class level, [Table 3](#) reports the per-class per-scale AP at the standard 640 input, evaluated with the official COCO API. TAEPM-Net improves the small-object AP_S for every one of the ten classes without exception (e.g., pedestrian 0.100 to 0.127, car 0.281 to 0.317, bus 0.084 to 0.140, motor 0.111 to 0.140), and the mean AP_S rises from 0.094 to 0.117 (+24.5%). We note that, in aggregate, the absolute AP_S gain may appear smaller than the medium/large gains; this is expected because small objects are intrinsically the hardest regime (limited pixels and weak texture), while the large-object AP is estimated from very few instances for most classes (see the size distribution in [Table 1](#)) and is therefore less stable. Measured by relative improvement, and across every class, the small-object regime benefits the most, which is consistent with the design goal of the proposed modules.

4.6 Per-Class Results at Higher Resolution

To confirm that the advantage is not merely an artifact of low resolution, [Table 4](#) repeats the per-class per-scale evaluation at a higher 1024 input. At this resolution, and at essentially identical computation to the baseline, TAEPM-Net still improves the small-object AP_S on eight of the ten classes (e.g., pedestrian 0.187 to 0.208, car 0.393 to 0.416, bus 0.193 to 0.262, van 0.159 to 0.189), and the mean AP_S rises from 0.162 to 0.180. This demonstrates that the gains of TAEPM-Net hold across object sizes, classes, and input resolutions, rather than coming from a particular resolution setting.

4.7 Effect of Input Resolution

A natural question is whether the accuracy gain simply comes from a larger input resolution. To rule this out, we also trained the YOLOv11n baseline at 1024×1024 under the same protocol (Table 5). Enlarging the baseline from 640 to 1024 raises its mAP@0.5 only marginally (32.9% to 33.1%) while inflating computation about $2.6\times$ (6.3 to roughly 16.5 GFLOPs), whereas TAEPM-Net reaches 37.3% at the standard 640 input with far fewer FLOPs (9.5G). The gain therefore stems from the architecture, not from input resolution.

4.8 Comparison Experiments

To further assess TAEPM-Net, we compared it under identical training and evaluation conditions on VisDrone-2019 against representative detectors spanning three families: two-stage detectors (Faster R-CNN [2], Cascade R-CNN [26]), recent DETR-based real-time detectors (D-FINE [27] and its successor DEIM [28]), and the YOLO series (YOLOv5n and YOLOv5s, YOLOv8n, YOLOv9-T [29], YOLOv10n [30], YOLOv11n [6], YOLOv12n [31], and YOLOv26n [32]). The YOLO comparison is centered on the nano (n/t) variants; for YOLOv5 we additionally include the small (s) variant as a stronger, larger reference. The results are summarized in Table 6.

As shown in Table 6, Faster R-CNN and Cascade R-CNN reach slightly higher mAP@0.5 but with parameter counts and GFLOPs more than an order of magnitude larger, which is impractical for lightweight UAV settings. The DETR-based D-FINE and DEIM attain competitive accuracy (36.2% and 36.1% mAP@0.5) at a moderate 3.7M parameters. Among the YOLO models, even YOLOv5s, with about three times our parameter count (7.04M), stays far below our accuracy, showing that our advantage holds across model scales, while the nano detectors YOLOv10n-YOLOv12n reach at most 33.5%. In comparison, TAEPM-Net attains the highest mAP@0.5 (37.3%) among all nano-scale models with a compact 2.33M parameters, smaller than the YOLOv11n baseline (2.58M), giving the best accuracy-parameter trade-off among the compared detectors.

4.9 Visualization Analysis

To intuitively understand the improvements brought by TAEPM-Net, visualization comparisons between the baseline YOLOv11n and TAEPM-Net were conducted on the VisDrone test-dev set, which is not used during training.

As can be seen from the Fig. 6 visualizations, the baseline YOLOv11n is noticeably deficient under several conditions: regions with many people, or areas occluded by vehicles and shadows, often suffer missed detections at lower confidence thresholds. In addition, because its head is poorly aligned with the underlying features, it frequently misidentifies background regions as objects.

TAEPM-Net generalizes better across these conditions. Through the CSPMA backbone and the EUCB upsampling layer, the spatial edges of extremely small targets are preserved, enabling precise localization of pedestrians even when they are small and far from the camera. In addition, guided by the TALDH dynamic detection head, the predicted bounding boxes better match the shape of pedestrians, eliminating some of the false positives caused by background interference.

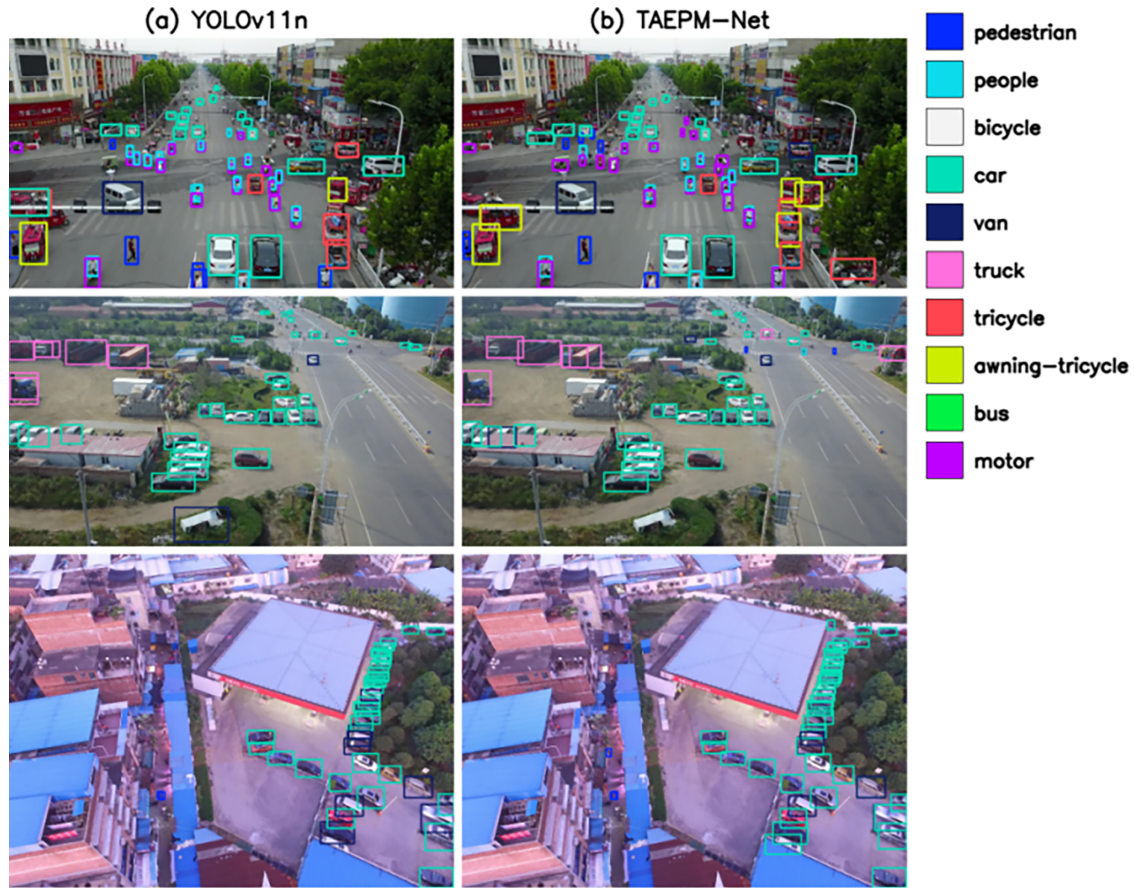


Figure 6: Visualization of the detection results of YOLOv11n and TAEPM-Net.

5 Conclusion

This paper presents TAEPM-Net, a lightweight detector built on YOLOv11 that improves the accuracy of UAV-based pedestrian and multi-class small-object detection while remaining compact.

We redesign the network in three aspects. First, the CSPMA module replaces the C3k2 blocks in the backbone to strengthen multi-scale feature extraction. Second, the proposed TALDH head aligns the classification and regression branches on a shared spatial representation, mitigating their spatial misalignment and improving generalization to irregularly shaped pedestrians and dense crowd scenes; because it removes the redundant layers of the conventional decoupled head, TALDH also reduces the overall parameter count. Third, the EUCB upsampling unit is added to the neck to restore fine-scale spatial detail lost during downsampling.

Extensive experiments on the VisDrone-2019 benchmark show that TAEPM-Net consistently outperforms the compared nano-scale detectors. Relative to the YOLOv11n baseline, it raises $mAP@0.5$ by 4.4 percentage points (to 37.3%) and $mAP@0.5:0.95$ by 3.1 points (to 22.3%), while reducing the parameter count from 2.58M to 2.33M (about 10%), mainly through the lightweight TALDH head. GFLOPs increase moderately to 9.5G, which we regard as an acceptable trade-off for the accuracy gained on small targets.

Overall, TAEPM-Net offers a favorable accuracy-parameter trade-off for UAV-based pedestrian and multi-class small-object detection, improving accuracy over the YOLOv11n baseline while using fewer parameters. We note that the present study evaluates the model on a desktop GPU; measuring on-device

latency and energy consumption on embedded UAV hardware is left for future work. In future research, besides improving robustness under adverse aerial conditions such as low light or poor weather (for example, through visible-infrared image fusion), we plan to apply acceleration techniques such as TensorRT and to deploy and test the model on physical UAV platforms under real-world conditions.

Acknowledgement: The authors express their gratitude to Universiti Teknologi Malaysia for its administrative and technical support.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Xinbin Hu led the analysis of the experimental results. Xinbin Hu and Norhaida Mohd Suaib jointly prepared the manuscript. Mohd Shahrizal Sunar provided overall direction and supervision. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data will be made available on request.

Ethics Approval: Not applicable.

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

References

1. Zhu P, Wen L, Du D, Bian X, Fan H, Hu Q, et al. Detection and tracking meet drones challenge. *IEEE Trans Pattern Anal Mach Intell.* 2022;44(11):7380–99. doi:10.1109/TPAMI.2021.3119563.
2. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell.* 2017;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
3. Zhu X, Su W, Lu L, Li B, Wang X, Dai J. Deformable DETR: deformable transformers for end-to-end object detection. In: *Proceedings of the 9th International Conference on Learning Representations (ICLR)*; 2021 May 4; Vienna, Austria; 2021. p. 1–16.
4. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, et al. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In: *Proceedings of the 11th International Conference on Learning Representations (ICLR)*; 2023 May 1; Online. p. 1–19.
5. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, et al. DETRs beat YOLOs on real-time object detection. In: *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2024 Jun 16–22; Seattle, WA, USA. New York, NY, USA: IEEE; 2024. p. 16965–74. doi:10.1109/cvpr52733.2024.01605.
6. Ma Y, Xi C, Ma T, Sun H, Lu H, Xu X, et al. I-YOLOv11n: a lightweight and efficient small target detection framework for UAV aerial images. *Sensors.* 2025;25(15):4857. doi:10.3390/s25154857.
7. Mi Q, Chao J, Chen A, Zhang K, Lai J. YOLO11s-UAV: an advanced algorithm for small object detection in UAV aerial imagery. *J Imag.* 2026;12(2):69. doi:10.3390/jimaging12020069.
8. Peng J, He H, Zhang D. YOLOv8s-DroneNet: small object detection algorithm based on feature selection and ISIoU. *Comput Mater Contin.* 2025;84(3):5047–61. doi:10.32604/cmc.2025.066368.
9. Feng C, Zhong Y, Gao Y, Scott MR, Huang W. TOOD: task-aligned one-stage object detection. In: *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*; 2021 Oct 10–17; Montreal, QC, Canada. New York, NY, USA: IEEE; 2021. p. 3490–9. doi:10.1109/iccv48922.2021.00349.
10. Zhu X, Hu H, Lin S, Dai J. Deformable ConvNets V2: more deformable, better results. In: *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2019 Jun 15–20; Long Beach, CA, USA. New York, NY, USA: IEEE; 2019. p. 9300–8. doi:10.1109/cvpr.2019.00953.
11. Lu H, Liu W, Fu H, Cao Z. FADE: a task-agnostic upsampling operator for encoder–decoder architectures. *Int J Comput Vis.* 2025;133(1):151–72. doi:10.1007/s11263-024-02191-8.

12. Tan M, Pang R, Le QV. EfficientDet: scalable and efficient object detection. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. New York, NY, USA: IEEE; 2020. p. 10778–87. doi:10.1109/cvpr42600.2020.01079.
13. Tang R, Sun H, Liu D, Xu H, Qi M, Kong J. EYOLOX: an efficient one-stage object detection network based on YOLOX. Appl Sci. 2023;13(3):1506. doi:10.3390/app13031506.
14. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: a single-stage object detection framework for industrial applications. arXiv:2209.02976. 2022.
15. Wang CY, Bochkovskiy A, Liao HM. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. New York, NY, USA: IEEE; 2023. p. 7464–75. doi:10.1109/cvpr52729.2023.00721.
16. Wang J, Xu C, Yang W, Yu L. Detecting tiny objects in aerial images: a normalized Wasserstein distance and a new benchmark. ISPRS J Photogramm Remote Sens. 2022;190(9):79–93. doi:10.1016/j.isprsjprs.2022.06.002.
17. Akyon FC, Onur Altinuc S, Temizel A. Slicing aided hyper inference and fine-tuning for small object detection. In: Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP); 2022 Oct 16–19; Bordeaux, France. New York, NY, USA: IEEE; 2022. p. 966–70. doi:10.1109/icip46576.2022.9897990.
18. Zhu X, Lyu S, Wang X, Zhao Q. TPH-YOLOv5: improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW); 2021 Oct 11–17; Montreal, BC, Canada. New York, NY, USA: IEEE; 2021. p. 2778–88. doi:10.1109/iccvw54120.2021.00312.
19. Chen Y, Yuan X, Wang J, Wu R, Li X, Hou Q, et al. YOLO-MS: rethinking multi-scale representation learning for real-time object detection. IEEE Trans Pattern Anal Mach Intell. 2025;47(6):4240–52. doi:10.1109/TPAMI.2025.3538473.
20. Lin TY, Dollar P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. New York, NY, USA: IEEE; 2017. p. 936–44. doi:10.1109/cvpr.2017.106.
21. Liu S, Qi L, Qin H, Shi J, Jia J. Path aggregation network for instance segmentation. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. New York, NY, USA: IEEE; 2018. p. 8759–68. doi:10.1109/cvpr.2018.00913.
22. Wang J, Chen K, Xu R, Liu Z, Loy CC, Lin D. CARAFE: content-aware ReAssembly of FEatures. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. New York, NY, USA: IEEE; 2019. p. 3007–16. doi:10.1109/iccv.2019.00310.
23. Liu W, Lu H, Fu H, Cao Z. Learning to upsample by learning to sample. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV); 2023 Oct 1–6; Paris, France. New York, NY, USA: IEEE; 2023. p. 6004–14. doi:10.1109/iccv51070.2023.00554.
24. Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy. New York, NY, USA: IEEE; 2017. p. 2999–3007. doi:10.1109/iccv.2017.324.
25. Du D, Qi Y, Yu H, Yang Y, Duan K, Li G, et al. The unmanned aerial vehicle benchmark: object detection and tracking. In: Computer vision—ECCV 2018. Cham, Switzerland: Springer International Publishing; 2018. p. 375–91. doi:10.1007/978-3-030-01249-6_23.
26. Cai Z, Vasconcelos N. Cascade R-CNN: delving into high quality object detection. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. New York, NY, USA: IEEE; 2018. p. 6154–62. doi:10.1109/cvpr.2018.00644.
27. Peng Y, Li H, Wu P, Zhang Y, Sun X, Wu F. D-FINE: redefine regression task in DETRs as fine-grained distribution refinement. In: Proceedings of the 13th International Conference on Learning Representations (ICLR); 2025 Apr 24–28; Singapore. p. 44015–31.

28. Huang S, Lu Z, Cun X, Yu Y, Zhou X, Shen X. DEIM: DETR with improved matching for fast convergence. In: Proceedings of the 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2025 Jun 10–17; Nashville, TN, USA. New York, NY, USA: IEEE; 2025. p. 15162–71. doi:10.1109/cvpr52734.2025.01412.
29. Wang CY, Yeh IH, Mark Liao HY. YOLOv9: learning what you want to learn using programmable gradient information. In: Computer vision—ECCV 2024. Cham, Switzerland: Springer Nature; 2024. p. 1–21. doi:10.1007/978-3-031-72751-1_1.
30. Wang A, Chen H, Liu L, Chen K, Lin Z, Han J, et al. YOLOv10: real-time end-to-end object detection. In: Proceedings of the Advances in Neural Information Processing Systems 37; 2024 Dec 10–15; Vancouver, BC, Canada. Sydney, Australia: NeurIPS; 2024. p. 107984–8011. doi:10.52202/079017-3429.
31. Tian Y, Ye Q, Doermann D. YOLOv12: attention-centric real-time object detectors. arXiv:2502.12524. 2025.
32. Chakrabarty S. YOLO26: an analysis of NMS-free end to end framework for real-time object detection. arXiv:2601.12882. 2026.