drones

MDPI

Article

HSF-Net: A Hierarchical Edge Enhancement and Sparse-Aware Fusion Network for Small Object Detection in UAV Aerial Imagery

Jiaqi Li [1], Qinghua Zeng [1,*], Songnan Duan [1], Junjie Wu [1], Yilin Li [1], Yudi Sun [1] and Qian Gao [2,3,*]

[1] Navigation Research Center, College of Automation Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China; lijiaqinrc@nuaa.edu.cn (J.L.); duansongnan@nuaa.edu.cn (S.D.); wu_junjie@nuaa.edu.cn (J.W.); lieo@nuaa.edu.cn (Y.L.); sunyd1126@nuaa.edu.cn (Y.S.)
[2] School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen 518172, China
[3] Laboratory for Computational Sensing and Robotics (LCSR), Johns Hopkins University, Baltimore, MD 21218, USA
* Correspondence: zengqh@nuaa.edu.cn (Q.Z.); qgao23@jh.edu (Q.G.)

Highlights

What are the main findings?

- HSF-Net enhances YOLOv11s for small-object detection in UAV imagery through a Hierarchical Edge Enhancement Module, a Sparse-Aware Feature Modulation module, and a small-object-oriented {P2, P3, P4} detection head, attaining higher accuracy than the baseline with substantially fewer parameters.
- Ablation studies confirm that the three components are complementary, with the edge-aware backbone enhancement and the foreground-selective neck fusion yielding their largest gains when coupled with the high-resolution P2 detection pathway.

What are the implications of the main findings?

- Explicitly compensating for fine-scale detail loss and foreground–background imbalance offers a parameter-efficient alternative to enlarging model capacity for small-object detection in aerial imagery.
- By combining a compact parameter and storage footprint with consistent accuracy gains across two markedly different aerial benchmarks, HSF-Net provides a promising architecture for UAV-based perception.

Abstract

Object detection in unmanned aerial vehicle (UAV) imagery is severely challenged by extremely small object scales, cluttered backgrounds, and pronounced foreground–background imbalance, which jointly degrade the accuracy of general-purpose detectors. This paper presents HSF-Net, a small-object detection network built upon YOLOv11s through three complementary enhancements. First, a Hierarchical Edge Enhancement Module (HEEM) employs orthogonal strip-convolution decomposition with zero-initialized residual fusion to enhance and re-weight the fine-scale edge and texture cues that are progressively attenuated in deep convolutional backbones. Second, a Sparse-Aware Feature Modulation (SAFM) module replaces concatenation-based fusion in the top-down neck pathway, coupling a sparse foreground gate with channel-wise scale modulation to confine cross-scale aggregation to object-bearing regions. Third, the detection head is restructured from {P3, P4, P5} to {P2, P3, P4}, introducing a high-resolution pathway for tiny targets while removing the original low-resolution P5 branch. On VisDrone, HSF-Net attains 46.4% mAP_{50} and 28.5% $mAP_{50–95}$, exceeding the baseline by 7.1 and 4.8 percentage points,

Check for updates

Academic Editor: San Jiang

Received: 26 June 2026
Revised: 25 August 2026
Accepted: 26 August 2026
Published: 27 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

respectively, while reducing the parameter count from 9.4 million to 3.6 million. The model achieves an end-to-end throughput of 131.3 FPS on an NVIDIA RTX 3090, although the high-resolution P2 branch increases the computational cost to 39.5 GFLOPs. After dataset-specific training and evaluation on the markedly different TinyPerson benchmark, HSF-Net outperforms YOLOv11s by 5.2 percentage points in mAP_{50}, indicating that its relative performance advantage persists under a substantially different target-scale distribution and maritime background.

Keywords: UAV aerial imagery; small object detection; edge enhancement; sparse foreground; feature modulation

1. Introduction

The rapid maturation of unmanned aerial vehicle (UAV) technology has transformed aerial imagery into an indispensable data modality across a broad spectrum of applications, including traffic surveillance, precision agriculture, disaster response, environmental monitoring, and smart-city management [1–3]. As a foundational perception capability underpinning these applications, object detection in UAV-captured imagery has attracted sustained research interest. However, despite the remarkable progress of general-purpose detectors on standard benchmarks such as MS-COCO [4], transferring these models to UAV scenarios remains far from straightforward, as their detection accuracy typically degrades substantially when confronted with the distinctive characteristics of aerial imagery.

Three intertwined challenges fundamentally constrain object detection performance in UAV aerial imagery. First, owing to typical flight altitudes ranging from 100 to 500 m, the overwhelming majority of objects appear at extremely small scales; statistics on mainstream UAV benchmarks such as VisDrone [5] show that a substantial proportion of instances occupy fewer than 32×32 pixels, with a non-negligible fraction smaller than 16×16 pixels. Such targets retain only weak structural cues, which are easily diluted by repeated convolutional downsampling. Second, aerial scenes are dominated by complex, heterogeneous backgrounds, including road surfaces, vegetation canopies, water bodies, rooftops, and shadows, against which small objects exhibit low contrast, a difficulty further compounded by their pronounced intra-class variation. This severe foreground–background imbalance allows uninformative background activations to overwhelm discriminative object signals during feature aggregation. Third, objects in UAV imagery are typically distributed in a highly non-uniform and densely clustered manner, with substantial scale variation even within a single category. Collectively, these challenges impose far more stringent requirements on detection models than those encountered in conventional ground-level scenarios.

To meet these challenges, a wide range of detectors has been investigated. Classical two-stage detectors such as Fast R-CNN [6] and Faster R-CNN [7] achieve strong localization accuracy but rely on multi-stage pipelines that are computationally prohibitive for real-time UAV deployment. One-stage detectors, exemplified by SSD [8] and the YOLO series [9], instead cast detection as a single-pass dense prediction problem, offering a markedly more favorable accuracy–efficiency trade-off that has made them the de facto choice for aerial applications. More recently, Transformer-based detectors such as DETR [10] and its real-time variant RT-DETR [11] have exploited global self-attention to model long-range dependencies and attain competitive accuracy on UAV benchmarks; their large parameter footprints and high inference latency, however, hinder practical deployment, and their advantage on small objects remains limited. Building on the YOLO framework, numerous UAV-specific variants have been proposed: CF-YOLO [12] redesigns the feature pyramid with cross-scale interaction pathways to alleviate small-object information attenu-

ation, and WCDB-YOLO [13] employs a wavelet-enhanced dual-backbone to jointly model object and contextual cues.

Despite these advances, two fundamental limitations persist in existing UAV-oriented detectors. (1) Insufficient preservation and exploitation of fine-scale structural detail. The discriminative cues of small objects, namely edges, corners, and fine textures, reside in localized, high-contrast structures rather than in coarse semantic outlines. Standard convolutional backbones, however, rely on repeated downsampling and successive smoothing operations that progressively erode these fine structures as the network deepens; consequently, the shallow feature maps that natively retain such cues are underexploited by subsequent detection heads, leading to severe representation loss for tiny targets. Existing remedies either employ spatial attention that re-weights existing responses without explicitly reinforcing the attenuated detail or introduce heavy modules such as Transformer blocks that incur substantial computational overhead. (2) Inefficient multi-scale feature fusion under foreground–background imbalance. Most current feature pyramid networks aggregate multi-scale features through indiscriminate channel concatenation followed by convolution, treating every spatial location and channel with equal importance. Such uniform fusion allows the abundant background activations characteristic of aerial imagery to dilute the sparse object-related signals during cross-scale propagation. Even recent feature pyramid designs such as BiFPN [14], AFPN [15], and CS-FPN [12] primarily refine connectivity patterns or apply weighted scalar fusion, without explicitly addressing the spatial sparsity and semantic disparity inherent to UAV scenes.

To overcome these limitations, this paper proposes the Hierarchical Edge Enhancement and Sparse-Aware Fusion Network (HSF-Net), a small-object detection framework built upon YOLOv11s. HSF-Net integrates two task-specific modules, the Hierarchical Edge Enhancement Module (HEEM) and the Sparse-Aware Feature Modulation (SAFM) module, with a small-object-oriented detection-head reconfiguration. HEEM explicitly restores attenuated edge and texture information through residual-detail extraction and hierarchical zero-initialized fusion. SAFM performs sparse foreground-selective and channel-adaptive cross-scale aggregation to suppress background-dominated responses. The original {P3, P4, P5} detection head is then reconfigured as {P2, P3, P4} so that these refined high-resolution representations can be used directly for tiny-object prediction. The principal contributions of this work are summarized as follows:

(1) A Hierarchical Edge Enhancement Module (HEEM) is proposed that, through a strip-based detail extraction with zero-initialized residual fusion and a coarse-to-fine kernel schedule, enhances and re-weights the edge and texture cues of small objects at a modest additional parameter and computational cost.

(2) A Sparse-Aware Feature Modulation (SAFM) module is designed, coupling a Sparse Foreground Gate (SFG) with a Channel Scale Modulation (CSM) unit, to replace naive neck concatenation with active, content-aware fusion that suppresses background interference under severe foreground–background imbalance.

(3) The detection head is restructured into a small-object-oriented {P2, P3, P4} configuration, in which a high-resolution P2 branch over HEEM/SAFM-enriched 4×-downsampled features is added and the redundant P5 branch is pruned, yielding a lighter yet more accurate detector.

The remainder of this paper is organized as follows. Section 2 reviews related work on generic and UAV-oriented small-object detection. Section 3 elaborates the proposed HEEM, SAFM, and restructured {P2, P3, P4} detection head, while Section 4 presents the experimental setup and a comprehensive evaluation comprising ablation, comparative, cross-benchmark, and qualitative studies. Section 5 concludes the paper and discusses future work.

2. Related Work

2.1. Object Detection Methods

Early object detection predominantly relied on conventional pipelines, which can be broadly grouped into template-matching and hand-crafted-feature approaches. The former localizes candidates by computing similarity measures between predefined templates and image patches, whereas the latter constructs discriminative descriptors from expert priors (e.g., SIFT [16], HOG [17], and saliency cues [18]) and couples them with conventional classifiers, as exemplified by deformable part-based models [19]. Such pipelines depend heavily on intricate prior design and multi-stage processing, which renders them sensitive to scene variation and computationally demanding under complex backgrounds, scale changes, and occlusions.

With the rise of deep learning, modern detectors are commonly divided into two-stage and one-stage paradigms, distinguished by whether candidate boxes are explicitly generated. Two-stage detectors, exemplified by the R-CNN family [6,7,20], first produce region proposals and then perform classification and bounding-box regression, achieving strong localization accuracy and discriminative capability. As a landmark work that introduced CNN-based feature extraction into detection, R-CNN [20] markedly advanced representation quality and yielded a substantial accuracy leap over hand-crafted-feature methods, while its successors Fast R-CNN [6] and Faster R-CNN [7] progressively unified feature sharing and region-proposal generation into an end-to-end trainable framework. Nevertheless, their multi-stage inference pipelines incur considerable computational overhead, limiting their applicability to real-time tasks such as UAV-based aerial image analysis.

In contrast, one-stage detectors, including SSD [8] and the YOLO series [9,21], cast detection as a dense prediction problem that simultaneously outputs class labels and bounding-box coordinates, thereby offering markedly higher inference efficiency and greater suitability for real-time and edge deployment. As one of the earliest one-stage detectors, SSD [8] performs classification and localization in a single forward pass and exploits multi-layer convolutional feature maps for multi-scale detection; its accuracy, however, remains constrained in complex scenes dominated by extremely small and densely clustered targets, such as aerial imagery. The subsequently introduced YOLO series has fundamentally reshaped the field. By reformulating detection as a unified regression task that jointly predicts object categories and bounding-box coordinates, it achieved real-time yet high-accuracy detection for the first time. From YOLOv1 [9] onward, the series has progressively incorporated advances such as refined anchor mechanisms, multi-scale feature fusion [22,23], dynamic label assignment, attention modules [24], and efficient backbones, steadily improving accuracy without sacrificing inference speed. Consequently, YOLO has become one of the most widely adopted detection frameworks in both academia and industry.

Despite these advances, one-stage detectors still suffer from insufficient preservation of shallow-level details, inadequate cross-scale feature fusion, and sample imbalance [25] when applied to scenes containing tiny or densely distributed targets, all of which jointly degrade recall and localization accuracy. Overcoming these bottlenecks while preserving the inference efficiency of one-stage networks has therefore become a critical research challenge.

2.2. YOLO-Based Methods for Small-Object Detection in UAV Imagery

YOLO-based detectors have been extensively adapted to UAV imagery, with existing efforts largely targeting three intertwined challenges: the feature scarcity of small targets, blurred boundaries under occlusion, and cluttered-background interference.

Small-target feature scarcity. Because ground targets often span only a few pixels, shallow feature maps carry limited spatial and textural cues. To amplify such cues, Drone-YOLO [26] adds a high-resolution detection head and a sandwich-fusion module that injects shallow features into multi-level heads, while Li et al. [27] reorganize the neck into a Bi-PAN-FPN with bidirectional reuse of shallow features and a GhostblockV2 module for low-cost long-range modeling. More recently, LYA-YOLO [28] counters downsampling-induced response decay through split-interactive spatial–channel modulation and a lightweight deformable head. These designs improve tiny-target recall but largely reuse or redistribute existing shallow responses, leaving the fine-scale edge and texture cues most diagnostic of small objects insufficiently restored.

Blurred boundaries and severe occlusion. The overhead viewpoint and densely populated scenes give rise to indistinct boundaries and pervasive mutual occlusion. Borrowing from face detection, YOLO-FaceV2 [29] couples a separated-enhancement attention module with a repulsion loss to disentangle inter- and intra-class occlusion, whereas LAM-YOLO [30] introduces a lighting-occlusion attention mechanism and an SIB-IoU loss to stabilize detection under variable illumination. DOMino-YOLO [31] further decomposes occlusion degradation into spatial, visibility, and contextual factors, addressing each via deformable convolutions, visibility-aware aggregation, and an occlusion-aware repulsion loss. Such methods enhance robustness to occlusion yet typically rely on heavier attention or deformable modules that raise computational cost.

Complex background interference. Expansive aerial scenes introduce rich context and inter-class similarity that obscure small objects. SOD-YOLO [32] suppresses background noise through receptive-field attention in the backbone and a balanced spatial–semantic pyramid with a high-resolution P2 branch in the neck; CF-YOLO [12] and SRTSOD-YOLO [33] instead redesign the feature pyramid (via cross-scale interaction pathways and gated convolutional fusion, respectively) to curb information attenuation during multi-scale propagation. WCDB-YOLO [13] adopts a target–context decoupled dual-backbone in which wavelet convolutions expand the receptive field to model global context. Although effective, these fusion schemes still aggregate features in a content-agnostic manner and do not explicitly counter the spatial sparsity and foreground–background imbalance of aerial scenes.

Overall, existing YOLO-based variants face a persistent accuracy–efficiency trade-off, and few simultaneously address fine-scale detail attenuation and foreground–background imbalance within a lightweight framework. Motivated by this gap, HSF-Net is proposed to jointly tackle feature scarcity, occlusion-induced response loss, and background interference while preserving real-time efficiency.

3. Methodology

YOLOv11 inherits the cumulative refinements of preceding YOLO versions and achieves competitive accuracy, but its performance remains suboptimal on UAV aerial imagery, where targets are predominantly small, densely distributed, and entangled with cluttered backgrounds. Although YOLO26 was available at the time of submission [21], YOLOv11s was deliberately retained as the baseline because the objective of this study is not to compare successive YOLO generations, but to evaluate targeted architectural modifications within a fixed detector framework. HSF-Net is defined on the specific YOLOv11s backbone–neck–head topology, whereas YOLO26 adopts a different native end-to-end detection formulation and updated training design.

To address the limitations of YOLOv11s in UAV aerial imagery, HSF-Net incorporates three targeted enhancements to strengthen sensitivity to small, densely distributed targets while preserving lightweight, real-time inference. First, to counteract the loss of fine-scale de-

tail induced by repeated downsampling, HEEM is embedded at three backbone stages (P2/4, P3/8, and P4/16), recovering and re-weighting the edge and texture cues on which small objects depend at a modest additional parameter and computational cost. Second, to mitigate the severe foreground–background imbalance of aerial scenes, SAFM replaces the original concatenation operations along the top-down neck, localizing sparse foreground regions and adaptively modulating channel-wise responses to yield spatially focused, semantically refined features. Third, the detection head is restructured from {P3, P4, P5} to a small-object-oriented {P2, P3, P4} configuration: a high-resolution P2 prediction branch is added to capture fine structural cues, while the semantically redundant P5 prediction branch is removed. In addition, the C2 block with Position-Sensitive Attention (C2PSA) is removed from the backbone. By design, the three enhancements are complementary: HEEM enriches texture-sensitive representations in the backbone, SAFM redirects neck capacity toward object-bearing regions, and the P2 prediction branch provides a dedicated pathway through which these refined features are exploited for accurate localization. Their integration enables HSF-Net to achieve markedly higher small-object detection accuracy with a lighter model than the YOLOv11s baseline. The overall architecture of HSF-Net is illustrated in Figure 1.

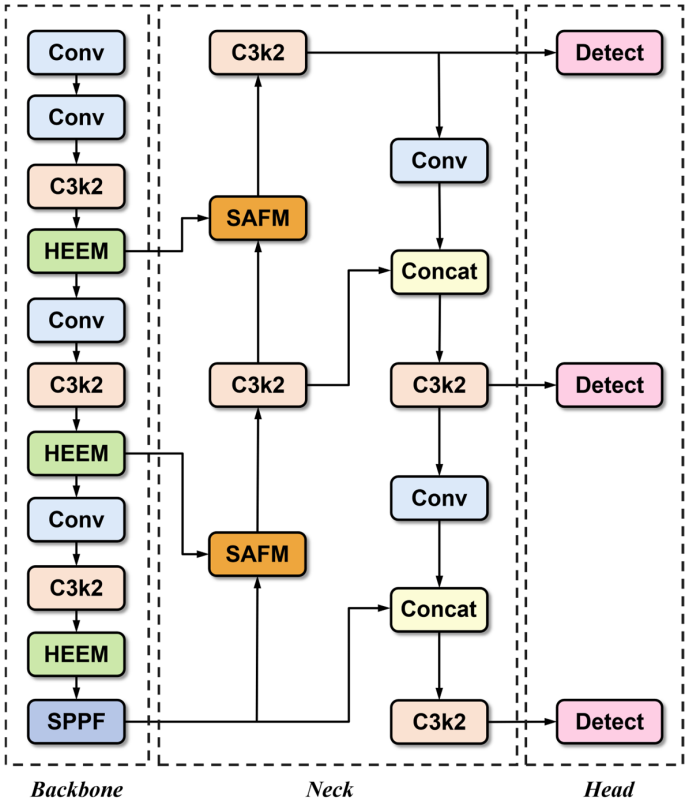


Figure 1. Overall architecture of the proposed HSF-Net: the Hierarchical Edge Enhancement Module (HEEM) embedded at the P2/4, P3/8, and P4/16 backbone stages; the Sparse-Aware Feature Modulation (SAFM) modules replacing the concatenation operations in the top-down neck pathway; the restructured detection head, in which a high-resolution P2 head is added and the original P5 head is removed; and the removal of the C2PSA module from the backbone.

3.1. Hierarchical Edge Enhancement Module (HEEM)

In UAV aerial imagery, small objects typically occupy only a handful of pixels and exhibit weak structural cues that are easily diluted by repeated convolutional downsampling. The discriminative information of such objects resides largely in fine-scale structures such as edges, corners, and textures. Because standard convolutional layers tend to act as smoothing operators that progressively suppress fine-grained detail as the network

deepens, the local structural cues intrinsically preserved in shallow feature maps are often under-exploited by subsequent detection heads.

To address this limitation, HEEM is proposed to explicitly reinforce fine-scale structural cues at multiple backbone stages through a lightweight detail-extraction strategy. The overall architecture of HEEM is illustrated in Figure 2. In contrast to conventional attention mechanisms that operate purely in the spatial domain, HEEM performs structure-aware feature refinement by means of two parallel orthogonal strip convolutions, an explicit residual-detail extraction, and a learnable zero-initialized fusion. The module is composed exclusively of depth-wise and point-wise operations, so the additional parameters and FLOPs are negligible, and it can be inserted into the YOLOv11 backbone without altering any feature-map dimensions, thereby preserving full architectural compatibility.

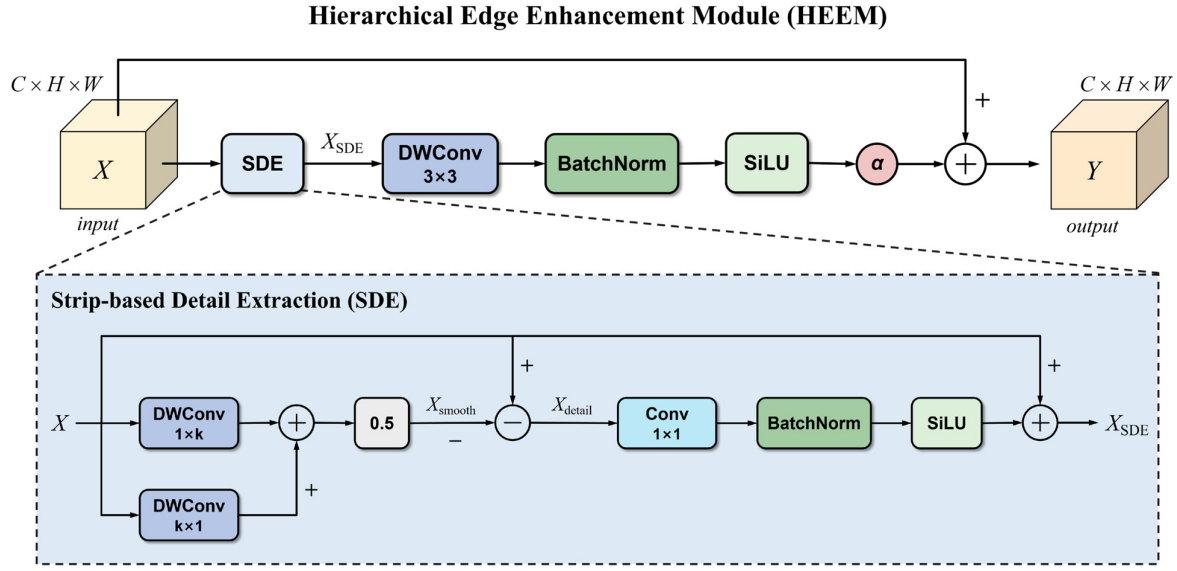


Figure 2. Structure of the proposed HEEM. Two orthogonal depth-wise strip convolutions, $1 \times k$ and $k \times 1$, are averaged with equal weights to form the smoothed estimate X_{smooth}. The residual detail $X_{\text{detail}} = X - X_{\text{smooth}}$ is recalibrated by a point-wise convolution with batch normalization and SiLU and added back to X inside the Strip-Based Detail Extraction (SDE) unit. The resulting SDE output is then refined by a 3×3 depth-wise convolution followed by batch normalization and SiLU, and injected into the backbone through a learnable, zero-initialized scalar α. Conv, DWConv, BatchNorm (BN), and SiLU denote convolution, depth-wise convolution, batch normalization, and the Sigmoid Linear Unit activation, respectively.

3.1.1. Strip-Based Detail Extraction (SDE)

The SDE unit is designed to construct a broad spatial smoothing reference with lower parameter and computational costs than a dense two-dimensional convolutional kernel. Following the principle of asymmetric strip convolution [34], SDE employs two parallel depthwise convolutions with orthogonal one-dimensional kernels:

$$X_{\text{smooth}} = \frac{1}{2}(\text{DWConv}_{1\times k}(X) + \text{DWConv}_{k\times 1}(X)), \tag{1}$$

where $X \in \mathbb{R}^{C\times H\times W}$ denotes the input feature map, and $\text{DWConv}_{1\times k}(\cdot)$ and $\text{DWConv}_{k\times 1}(\cdot)$ denote depth-wise convolutions with horizontal and vertical strip kernels of length k, respectively. Owing to their large spatial extent, the two strips respond mainly to slowly varying, spatially extended patterns: the horizontal strip aggregates context along the horizontal direction, whereas the vertical strip aggregates context along the vertical direction. Averaging the two yields an anisotropic, cross-shaped aggregation that provides a parameter-efficient

smoothed reference at only $O(2Ck)$ parameters, markedly fewer than the $O(Ck^2)$ cost of a full $k \times k$ depth-wise convolution. Rather than acting as an exact isotropic low-pass filter, this estimate serves only as a smoothing reference from which informative residual details are subsequently extracted.

The residual detail is then obtained by subtracting the smoothed component from the original input:

$$X_{\text{detail}} = X - X_{\text{smooth}}. \tag{2}$$

This residual emphasizes locally contrastive structures, namely edges, contours, and fine-grained textures, which are precisely the cues decisive for distinguishing small objects from cluttered backgrounds. A point-wise convolution followed by batch normalization and a SiLU activation is then applied to recalibrate the residual detail across channels, and the recalibrated signal is superimposed on the original input through a short residual connection:

$$X_{\text{SDE}} = X + \delta(\text{BN}(\text{Conv}_{1\times1}(X_{\text{detail}}))), \tag{3}$$

where $\delta(\cdot)$ denotes the SiLU activation and $\text{BN}(\cdot)$ denotes batch normalization. The residual term X in Equation (3) is essential: it guarantees that the full information of the input is retained, so that SDE amplifies fine-scale details on top of the original representation rather than replacing it, which prevents the loss of coarse semantic context.

3.1.2. Hierarchical Adaptive Fusion

To inject the enhanced features into the backbone while strictly preserving the original feature semantics, HEEM adopts a zero-initialized residual fusion scheme. The SDE output is first aggregated spatially by a 3×3 depth-wise convolution with batch normalization and SiLU activation, and is then fused with the input via a learnable scalar gate:

$$Y = X + \alpha \cdot \delta(\text{BN}(\text{DWConv}_{3\times3}(X_{\text{SDE}}))), \tag{4}$$

where $\alpha \in \mathbb{R}$ is a learnable scalar initialized to zero and Y is the module output. The zero-initialization strategy is critical for training stability. At the beginning of optimization, $\alpha = 0$ reduces HEEM to an exact identity mapping, so the module introduces no perturbation to the well-conditioned backbone weights. As training proceeds, the network gradually learns an appropriate magnitude of α that balances detail enhancement against feature stability. This mechanism follows the warm-start principle widely adopted in modern residual designs [35] and introduces no additional hyperparameter.

Batch normalization (BN) is retained at both normalization sites in HEEM to remain consistent with the surrounding convolutional blocks of YOLOv11s and to support standard convolution–BN folding during inference. The complete model exhibited stable convergence under the fixed batch size of 8 used in the reported experiments. Layer normalization (LN) is independent of batch statistics and applies the same normalization rule during training and inference; it may therefore be advantageous when training with extremely small or dynamically varying batch sizes.

3.1.3. Multi-Scale Deployment

Small objects in aerial imagery span a wide range of pixel extents, from a few pixels for distant pedestrians to several dozen pixels for nearby vehicles. To accommodate this scale diversity, HEEM is instantiated at three hierarchical backbone stages (P2/4, P3/8, and P4/16), immediately after the C3k2 block of each stage, with a stage-specific strip-kernel length k:

- HEEM-P2 ($k = 15$): Applied to the shallowest stage, where the spatial resolution is highest. The long strip kernel emulates a wide smoothing extent, allowing the module

to estimate the slowly varying context against which the fine textures of tiny objects are accentuated.

- HEEM-P3 ($k = 11$): Deployed at the intermediate stage to enhance mid-scale objects such as motorcycles and bicycles, for which mid-scale edges carry most of the discriminative information.
- HEEM-P4 ($k = 7$): Used at the deeper stage, where receptive fields are already large. A shorter strip kernel suffices to capture the residual fine-scale cues while avoiding over-smoothing of the semantic representations.

3.2. Sparse-Aware Feature Modulation (SAFM)

The neck of a detector is responsible for fusing features across resolutions, yet the conventional fusion operators in YOLO-family necks treat every spatial location and every channel identically. In UAV imagery, where tiny foreground objects are sparsely distributed over vast and cluttered backgrounds, such undifferentiated fusion allows background responses from the coarse, low-resolution branch to overwhelm the weak fine-scale detail carried by the high-resolution branch, blurring the very cues on which small-object detection depends. To overcome this limitation, the SAFM block is proposed to perform foreground-aware and channel-adaptive cross-scale fusion. As illustrated in Figure 3, SAFM takes a high-resolution feature $F_{hi} \in R^{C\times H\times W}$ and a low-resolution feature $F_{lo} \in R^{C'\times(H/2)\times(W/2)}$ and emits a fused map of the same spatial size as F_{hi}. It comprises three cooperating components: an SFG that localizes object-bearing regions, a CSM unit that adaptively weights the two scales per channel, and a foreground-selective fusion followed by a zero-initialized residual output.

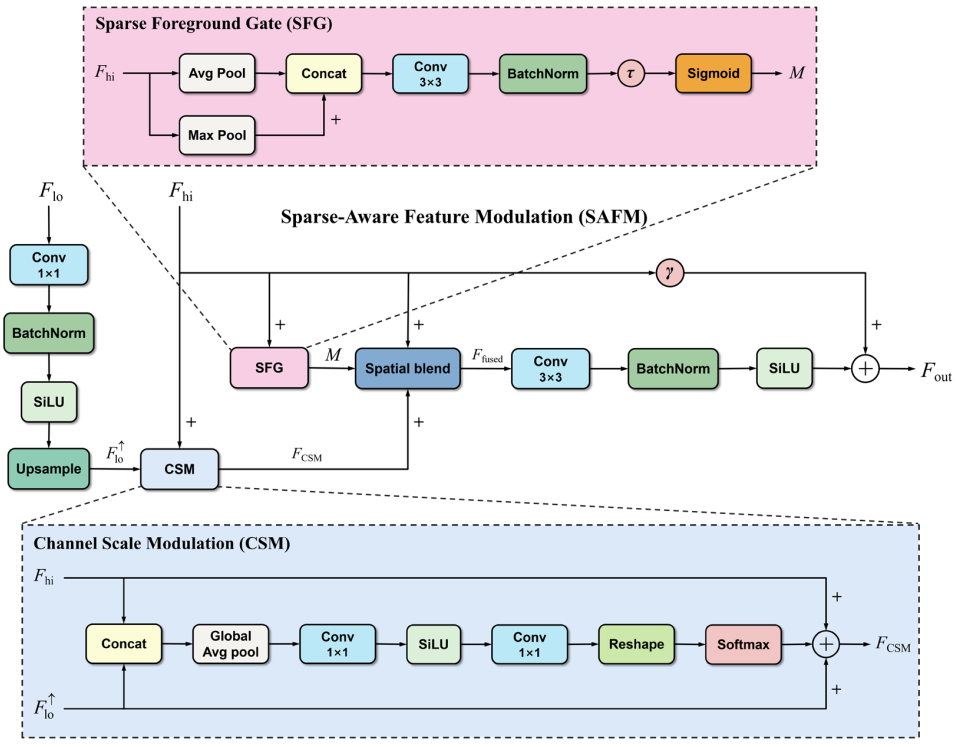


Figure 3. Structure of the proposed SAFM block, comprising the Sparse Foreground Gate (SFG), the Channel Scale Modulation (CSM) unit, a foreground-selective spatial blend, and a zero-initialized residual output. Avg Pool, Max Pool, Global Avg Pool, Concat, Upsample, Reshape, Softmax, BatchNorm, SiLU, and Sigmoid denote average pooling, max pooling, global average pooling, channel-wise concatenation, nearest-neighbor upsampling, tensor reshaping, softmax normalization, batch normalization, the Sigmoid Linear Unit activation, and the logistic sigmoid function, respectively.

3.2.1. Sparse Foreground Gate (SFG)

To restrict cross-scale fusion to informative regions, SFG estimates a spatial selection map intended to emphasize object-bearing regions from the high-resolution feature. Two channel-wise statistics, namely average pooling and max pooling along the channel axis, are extracted to summarize the mean activation and the most salient response at each spatial location, respectively:

$$m_{\text{avg}}(h,w) = \frac{1}{C}\sum_{c=1}^{C} F_{hi}(c,h,w), \tag{5}$$

$$m_{\max}(h,w) = \max_{1\le c\le C} F_{hi}(c,h,w), \tag{6}$$

where $h = 1,\ldots,H$, $w = 1,\ldots,W$ and $m_{\text{avg}}, m_{\max} \in \mathbb{R}^{1\times H\times W}$. The two maps are concatenated along the channel dimension and transformed into a foreground gate by a 3×3 convolution, batch normalization, and a temperature-controlled sigmoid:

$$M = \sigma\big(\tau \cdot \mathrm{BN}\big(\mathrm{Conv}_{3\times 3}\big([m_{\text{avg}}; m_{\max}]\big)\big)\big), \tag{7}$$

where $[\cdot;\cdot]$ denotes channel concatenation, $\sigma(\cdot)$ is the sigmoid function, $\mathrm{BN}(\cdot)$ is batch normalization, and τ is a learnable temperature (initialized to one) that rescales the pre-activation logits before the sigmoid, thereby controlling the sparsity of the gate map $M \in (0,1)^{1\times H\times W}$. A larger τ sharpens M toward a near-binary, highly selective mask, whereas a smaller τ yields softer gating; since τ is optimized end-to-end, the foreground selectivity adapts automatically to the data without any manual threshold.

3.2.2. Channel Scale Modulation (CSM)

Before fusion, the low-resolution feature is projected and upsampled to match the channel width and spatial resolution of the high-resolution branch. Plain concatenation followed by convolution leaves scale selection to a fixed-parameter transformation and does not explicitly provide sample-dependent, normalized scale weights. CSM instead predicts a pair of per-channel weights that adaptively balances the aligned high- and low-resolution features for each input.

$$F_{lo}^{\uparrow} = \mathrm{Up}(\delta(\mathrm{BN}(\mathrm{Conv}_{1\times 1}(F_{lo})))) \tag{8}$$

where $\delta(\cdot)$ denotes the SiLU activation and $\mathrm{BN}(\cdot)$ denotes batch normalization. The two aligned features are concatenated, globally pooled, and passed through a lightweight two-layer bottleneck (reduction ratio $r = 8$); the resulting logits are reshaped into two branches and normalized by a softmax so that, for every channel, the high- and low-resolution weights sum to one:

$$[\omega_{hi};\omega_{lo}] = Softmax\Big(\mathrm{Conv}_{1\times 1}\Big(\delta\Big(\mathrm{Conv}_{1\times 1}\Big(\mathrm{GAP}\Big([F_{hi}; F_{lo}^{\uparrow}]\Big)\Big)\Big)\Big)\Big) \tag{9}$$

where $\mathrm{GAP}(\cdot)$ is global average pooling, and ω_{hi}, $\omega_{lo} \in \mathbb{R}^{C\times 1\times 1}$ with $\omega_{hi} + \omega_{lo} = 1$ per channel. The channel-modulated feature is then the convex combination:

$$F_{CSM} = \omega_{hi} \odot F_{hi} + \omega_{lo} \odot F_{lo}^{\uparrow} \tag{10}$$

where $\odot$ denotes channel-wise multiplication with spatial broadcasting. This data-dependent weighting lets each channel decide whether fine detail (F_{hi}) or coarse semantics ($F_{lo}^{\uparrow}$) is more informative, replacing the fixed fusion ratio of a plain concatenation at the cost of only two 1×1 convolutions.

3.2.3. Foreground-Selective Fusion and Residual Output

The foreground gate M and the channel-modulated feature F_{CSM} are combined so that adaptive cross-scale fusion is applied only inside object-bearing regions, while background locations fall back to the unaltered high-resolution feature:

$$F_{fused} = M \odot F_{CSM} + (1 - M) \odot F_{hi}. \tag{11}$$

This selective scheme prevents background responses from the low-resolution branch from contaminating clean background regions, concentrating the fusion capacity on the sparse foreground. Finally, the fused map is refined by a 3×3 convolution (with batch normalization and SiLU) and combined with a zero-initialized, channel-wise skip connection from the high-resolution input:

$$F_{out} = \delta\left(\mathrm{BN}\left(\mathrm{Conv}_{3\times3}\left(F_{fused}\right)\right)\right) + \gamma \odot F_{hi}, \tag{12}$$

where $\gamma \in \mathbb{R}^{C\times1\times1}$ is a learnable per-channel vector initialized to zero. The two terms serve distinct purposes. The first term, $\delta\left(\mathrm{BN}\left(\mathrm{Conv}_{3\times3}\left(F_{fused}\right)\right)\right)$, is the foreground-selective cross-scale fusion and constitutes the output of the module at the start of training, when $\gamma = 0$; SAFM therefore begins by emitting the fully formed fused representation rather than an identity mapping. The second term, $\gamma \odot F_{hi}$, is a zero-initialized, channel-wise skip from the high-resolution input. Because γ grows from zero during optimization, the unmodified high-resolution detail is re-admitted gradually and on a per-channel basis, with each channel learning how much raw fine-scale information to reinstate alongside the fused feature. In contrast to the identity-preserving zero-initialization adopted in HEEM, where $\alpha = 0$ yields an exact identity branch, the zero-initialized γ here does not protect an identity mapping; instead, it schedules a gradual re-injection of high-resolution detail, allowing the freshly initialized fusion branch to stabilize before the raw skip begins to contribute. This staged scheme mitigates early competition between the fusion and skip pathways and yields stable convergence without any additional hyperparameter.

At the network level, SAFM is selectively deployed at the P4-to-P3 and P3-to-P2 top-down fusion nodes, where coarse semantic features are injected into higher-resolution maps and sparse object responses are most susceptible to background interference. The P4-to-P3 SAFM fuses the P3 lateral feature with the Spatial Pyramid Pooling-Fast (SPPF) output, while the P3-to-P2 SAFM further refines the high-resolution P2 pathway. The resulting features are subsequently propagated through the bottom-up neck to construct the P3 and P4 prediction branches. Consequently, the P4 branch already receives SAFM-refined information indirectly, while the reused SPPF output provides complementary deep semantic context. A direct SAFM-to-P4 shortcut is therefore omitted to avoid redundant feature transfer and additional complexity. Unlike conventional pyramid necks that apply uniform fusion throughout the hierarchy, HSF-Net reserves SAFM for the two top-down nodes most affected by foreground–background imbalance and retains conventional concatenation in the bottom-up pathway.

3.3. Small-Object Detection Head

YOLO-family detectors conventionally employ a three-level detection head comprising the P3, P4, and P5 prediction branches, which operate at downsampling strides of 8, 16, and 32, respectively. Although effective for generic object detection, this configuration is not optimally matched to UAV aerial imagery, where objects are predominantly small. The deepest P5 level provides strong semantic information but has the lowest spatial resolution; consequently, the fine structural cues of tiny targets may be substantially attenuated after

repeated downsampling. Even the higher-resolution P3 level, operating at a stride of 8, may remain too coarse for objects occupying only a few pixels. Therefore, the original detection head lacks a dedicated high-resolution prediction branch for fine-grained localization, while the wide-channel P5 prediction branch retains a relatively large parameter burden despite its coarse spatial representation.

To better match the object-scale distribution of UAV imagery, the original {P3, P4, P5} detection head is restructured into a small-object-oriented {P2, P3, P4} configuration with strides of 4, 8, and 16, respectively, as illustrated in Figure 1. Specifically, a high-resolution P2 prediction branch, fed by the shallowest backbone stage through the top-down SAFM pathway, is introduced, while the original P5 prediction branch is removed. Because P2 has twice the spatial resolution of P3 along each axis, it contains four times as many spatial locations. This finer prediction grid preserves more low-level structural cues, including edges, corners, and textures, while providing denser candidate locations for minute objects, thereby improving their localization in cluttered aerial scenes.

From a complexity perspective, the addition of the P2 prediction branch and the removal of the P5 prediction branch have different effects on the parameter count and computational cost. For a convolutional layer with kernel size k, the number of learnable parameters is proportional to $k^2 C_{in} C_{out}$, whereas the computational cost is additionally proportional to the spatial size of the output feature map, namely $HWk^2 C_{in} C_{out}$. Therefore, the high-resolution P2 prediction branch increases GFLOPs because its convolutional operations are performed over a substantially larger feature map. In contrast, the low-resolution P5 prediction branch uses the widest channels in the original detection hierarchy; removing it eliminates a comparatively large number of convolutional weights while reducing fewer spatial computations. At the complete-network level, the removal of the C2PSA module further contributes to the reduction in parameters.

The restructured detection head also works synergistically with HEEM and SAFM. HEEM reinforces edge- and texture-sensitive cues across the backbone, while SAFM redirects cross-scale feature fusion toward sparse object-bearing regions. The newly introduced P2 prediction branch then provides a dedicated high-resolution pathway through which these enhanced representations can be directly exploited, thereby further improving small-object detection performance.

4. Experiments and Results

4.1. Experimental Setup

All experiments conducted in this study were performed in a unified software and hardware environment, whose detailed specifications are summarized in Table 1. The detailed training hyperparameters and data-augmentation configuration are summarized in Table 2. All models were trained and evaluated under identical settings to ensure a fair and reliable comparison.

Table 1. Software and hardware configuration of the experimental environment.

Component	Specification
GPU	NVIDIA GeForce RTX 3090
CPU	Intel Core i9-12900K (12th generation)
GPU memory	24 GB
Operating system	Windows 11
Python	3.9
PyTorch	1.10.0
CUDA	11.6

Table 2. Training hyperparameters and data-augmentation settings.

Parameter	Value
Epochs	300
Input image size	640 × 640
Batch size	8
Number of workers	4
Optimizer	SGD
Optimizer momentum	0.937
Initial learning rate	0.01
Final learning rate	0.0001
Weight decay	0.0005
Translation fraction	0.1
Horizontal-flip probability	0.5
Mosaic probability	1.0
Random seed	0
Deterministic mode	True

4.2. Datasets

The proposed method was evaluated on two publicly available UAV-imagery benchmarks: VisDrone [5] and TinyPerson [36]. Quantitative evaluations, including the ablation and comparative studies, were conducted primarily on VisDrone, whereas TinyPerson was used as an additional benchmark to evaluate the effectiveness and robustness of the proposed architecture under markedly different object-scale distributions and background conditions. In addition, representative images from both datasets were selected for qualitative visualization, providing intuitive insight into the detection behavior of the model under varied real-world conditions.

The VisDrone dataset, released by the AISKYEYE team at Tianjin University, comprises 6471 training images and 548 validation images captured by various drone-mounted cameras across a wide range of urban and suburban scenes and viewing angles. It provides bounding-box annotations for ten object categories: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. Owing to the prevalence of densely distributed small objects, severe occlusions, and cluttered backgrounds, VisDrone has become one of the most challenging and widely adopted benchmarks for object detection in UAV imagery.

The TinyPerson dataset was developed specifically for detecting person instances with extremely small image-plane extents in long-distance, large-field-of-view imagery, with particular emphasis on maritime search-and-rescue scenarios. Here, the term "tiny person" describes the apparent image scale of the annotated human instances; it does not refer to children or persons of small physical stature, and the benchmark does not define age-based categories. Most annotated persons occupy fewer than 20 pixels and exhibit substantial variation in pose and viewpoint, low signal-to-noise ratios, and severe interference from extensive and complex backgrounds. The dataset contains 1610 images and 72,651 annotated instances. Following the official two-way partition, 794 images containing 42,197 annotations were used for training, while the remaining 816 images containing 30,454 annotations were used for validation and evaluation.

4.3. Experimental Evaluation Metrics

In this study, the performance of the proposed network is quantitatively evaluated using three widely adopted object-detection metrics: precision (P), recall (R), and mean average precision (mAP) [4].

Precision denotes the proportion of correctly identified targets among all detected instances and thus reflects the reliability of the positive predictions. It is defined as:

$$P = \frac{TP}{TP + FP}, \tag{13}$$

where TP denotes the number of true positives, corresponding to correctly detected objects, and FP denotes the number of false positives, corresponding to predicted objects that do not match any ground-truth instance under the specified matching criterion.

Recall measures the proportion of correctly detected targets among all ground-truth instances and therefore indicates the model's ability to identify positive samples without omission. It is defined as:

$$R = \frac{TP}{TP + FN}, \tag{14}$$

where FN denotes the ground-truth targets that are missed by the detector.

Average precision (AP) corresponds to the area under the precision–recall (P–R) curve obtained by varying the confidence threshold, and provides a comprehensive assessment of detection performance for a given class. The mean average precision (mAP) is then defined as the mean of the AP values over all object categories. The two metrics are computed as:

$$AP = \int_0^1 P_i(R)\, dR, \tag{15}$$

$$mAP = \frac{1}{K}\sum_{i=1}^{K} AP_i, \tag{16}$$

where K is the total number of object categories and AP_i is the average precision of the i-th category.

In addition, to quantify the computational complexity introduced by the proposed modules, the number of parameters (Params) and the giga floating-point operations (GFLOPs) are reported as auxiliary indicators of model efficiency.

4.4. Overall and Class-Wise Performance Analysis

To verify the overall effectiveness of the proposed enhancements, the baseline YOLOv11s and HSF-Net were trained and evaluated on the VisDrone dataset under identical settings. The results are reported in Table 3 and visualized in Figure 4. HSF-Net consistently outperforms the baseline across all four metrics, improving precision, recall, mAP_{50}, and $mAP_{50–95}$ by 5.2, 6.3, 7.1, and 4.8 percentage points, respectively, which confirms a substantial gain in overall detection performance.

Table 3. Performance comparison between the baseline YOLOv11s and the proposed HSF-Net on the VisDrone dataset.

Model	P (%)	R (%)	mAP_{50} (%)	$mAP_{50–95}$ (%)
YOLOv11s	50.7	38.2	39.3	23.7
HSF-Net	55.9	44.5	46.4	28.5

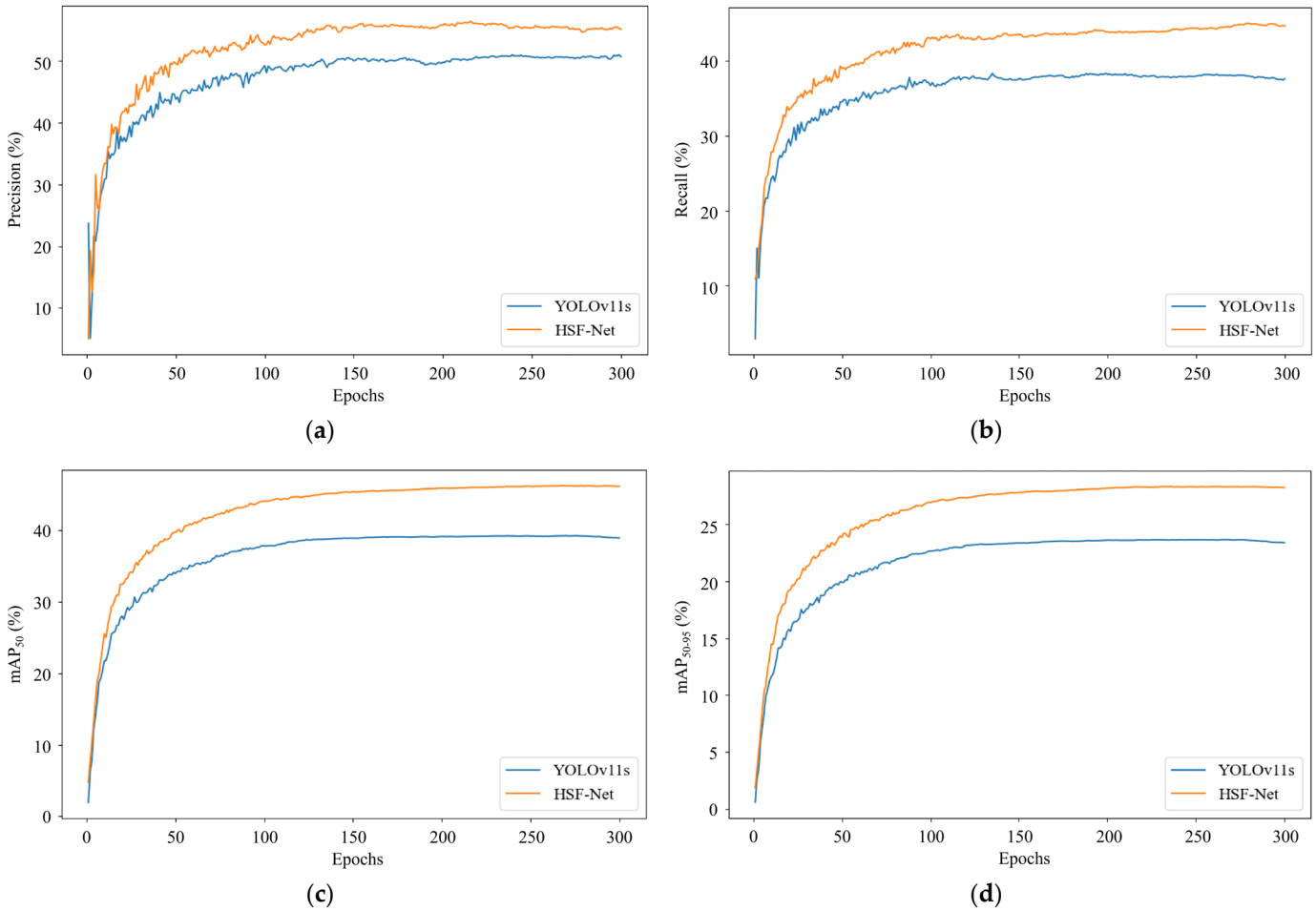


Figure 4. Training curves of the baseline YOLOv11s and the proposed HSF-Net on the VisDrone dataset: (**a**) precision; (**b**) recall; (**c**) mAP_{50}; (**d**) $mAP_{50–95}$.

To provide a more targeted category-level assessment, Table 4 reports AP_{50} and $AP_{50–95}$ for each of the ten VisDrone classes. HSF-Net improves both metrics for every category. The largest AP_{50} gains are observed for people, pedestrian, and motor, with improvements of 11.6, 11.0, and 10.6 percentage points, respectively. Their corresponding $AP_{50–95}$ values increase by 6.1, 6.5, and 6.5 percentage points. These pronounced improvements are consistent with the intended design of HSF-Net, which strengthens high-resolution structural cues and suppresses background interference in small-object-dominated aerial scenes.

Consistent improvements are also obtained for the remaining seven categories, including classes with relatively high baseline accuracy, such as car and bus. Although bicycle and awning-tricycle remain the most challenging categories in terms of absolute AP, HSF-Net still improves their AP_{50} by 6.4 and 4.1 percentage points, respectively. Overall, the class-wise results demonstrate that the aggregate mAP improvement is broadly distributed across object categories rather than being dominated by only one or two classes, thereby providing more targeted evidence of the effectiveness of HSF-Net on the VisDrone benchmark.

Table 4. Class-wise average precision of YOLOv11s and HSF-Net on the VisDrone validation set. Δ denotes the absolute performance difference between HSF-Net and YOLOv11s, measured in percentage points. The final row reports the mean AP over all ten classes.

Class	YOLOv11s AP_{50} (%)	HSF-Net AP_{50} (%)	Δ AP_{50} (pp)	YOLOv11s $AP_{50–95}$ (%)	HSF-Net $AP_{50–95}$ (%)	Δ $AP_{50–95}$ (pp)
Pedestrian	42.9	53.9	11.0	19.8	26.3	6.5
People	31.9	43.5	11.6	12.2	18.3	6.1
Bicycle	12.7	19.1	6.4	5.4	8.8	3.4
Car	79.8	84.9	5.1	57.5	61.8	4.3
Van	45.5	50.3	4.8	32.5	36.3	3.8
Truck	36.7	42.3	5.6	23.9	28.9	5.0
Tricycle	26.9	34.1	7.2	14.9	19.5	4.6
Awning-tricycle	14.7	18.8	4.1	9.0	11.6	2.6
Bus	58.0	62.2	4.2	42.0	46.7	4.7
Motor	44.1	54.7	10.6	20.0	26.5	6.5
All classes (mAP)	**39.3**	**46.4**	**7.1**	**23.7**	**28.5**	**4.8**

4.5. Ablation Experiments

HSF-Net introduces three key enhancements over the YOLOv11s baseline: (i) HEEM; (ii) SAFM; and (iii) a restructured, small-object-oriented detection head that adds a high-resolution P2 branch while removing the redundant P5 branch (denoted {P2, P3, P4} in Table 5, and abbreviated as "restructured head" hereafter). To quantify the contribution of each component, the three were added to the baseline both individually and in combination, under identical training settings on the same dataset to ensure a fair comparison. The exact endpoint results of the ablation configurations are reported in Table 5. Figure 5 provides complementary information by showing the evolution of mAP_{50} and $mAP_{50–95}$ over the complete training process. In addition to the final ranking, the curves allow the relative convergence behavior, optimization stability, and persistence of the performance differences among the individual, pairwise, and complete configurations to be examined.

Table 5. Ablation study of the proposed components. A check mark (√) indicates that the corresponding component is enabled. The best result in each column is shown in bold.

Baseline	HEEM	SAFM	{P2, P3, P4}	P (%)	R (%)	mAP_{50} (%)	$mAP_{50–95}$ (%)	GFLOPs	Parameters (M)
√				50.7	38.2	39.3	23.7	**21.3**	9.4
√	√			51.1	39.0	40.2	24.1	23.7	9.1
√		√		52.9	39.3	41.0	25.0	30.2	10.7
√			√	53.8	41.8	43.5	26.5	22.5	**2.6**
√	√	√		52.3	40.7	41.8	25.4	32.6	10.4
√	√		√	52.6	43.5	44.4	27.2	24.8	2.8
√		√	√	53.5	43.6	45.3	27.9	37.2	3.5
√	√	√	√	**55.9**	**44.5**	**46.4**	**28.5**	39.5	3.6

As shown in Figure 5, all configurations progressively enter stable performance regions without pronounced late-stage deterioration. The complete HSF-Net establishes the highest mAP_{50} and $mAP_{50–95}$ trajectories during the later stage of training and maintains this advantage until the end of optimization. In addition, configurations incorporating the restructured detection head generally reach higher stable performance levels than their counterparts without it. These observations complement the endpoint values reported in Table 5 by showing that the performance gains are sustained throughout the later training stage.

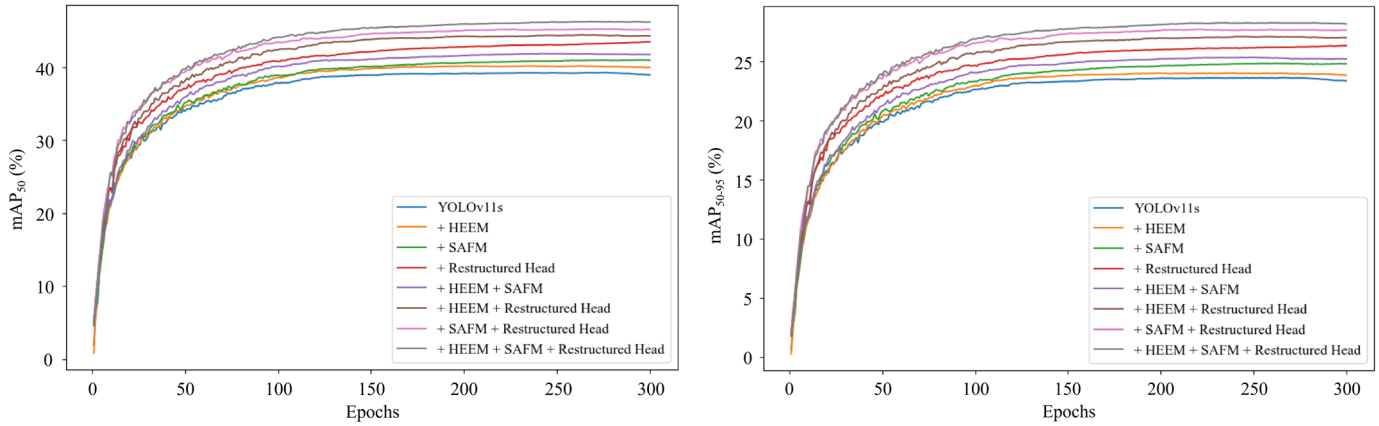


Figure 5. Evolution of the validation metrics during training for the different ablation configurations on the VisDrone dataset: (**a**) mAP_{50}; (**b**) $mAP_{50–95}$. The exact final values are reported in Table 5, whereas the curves illustrate the relative convergence behavior, optimization stability, and persistence of the performance differences over 300 training epochs.

Effect of individual components. When introduced separately, all three components yield consistent improvements over the baseline across every metric, confirming that each addresses a distinct deficiency of the original detector. HEEM provides a modest but consistent gain, indicating that reinforcing attenuated fine-scale cues is beneficial on its own. SAFM produces a larger improvement, reflecting the benefit of suppressing background interference during multi-scale fusion. The restructured head delivers the most pronounced single-component gain, raising mAP_{50} to 43.5% and recall to 41.8%, consistent with its role in supplying a dedicated high-resolution pathway for tiny objects whose details would otherwise be lost through repeated downsampling.

Effect of pairwise combinations. Combining any two components further amplifies performance, demonstrating that the three modules are complementary rather than redundant. The gains are most evident when a component is paired with the restructured head, with HEEM + Restructured Head and SAFM + Restructured Head reaching 44.4% and 45.3% mAP_{50}, respectively. By contrast, pairing HEEM with SAFM without the restructured detection head attains only 41.8% mAP_{50}. This supports the design hypothesis that HEEM and SAFM enrich the feature representation, while the P2 prediction branch within the restructured detection head provides the pathway through which these enhanced cues are exploited.

Marginal gains beyond the restructured detection head. As shown in Table 5, the restructured {P2, P3, P4} detection head is the strongest individual modification, improving mAP_{50} and $mAP_{50–95}$ by 4.2 and 2.8 percentage points, respectively. Nevertheless, HEEM and SAFM provide additional gains when the restructured detection head is kept unchanged. Relative to the restructured-head-only configuration, adding HEEM increases mAP_{50} and $mAP_{50–95}$ by 0.9 and 0.7 percentage points, respectively, whereas adding SAFM increases them by 1.8 and 1.4 percentage points. Integrating both modules with the restructured detection head to form the complete HSF-Net further increases the two metrics by 2.9 and 2.0 percentage points. These joint increments account for 40.8% and 41.7% of the total baseline-to-HSF-Net improvements in mAP_{50} and $mAP_{50–95}$, respectively. These results indicate that the three components provide complementary and largely additive benefits at the levels of feature quality, fusion selectivity, and detection resolution.

4.6. Comparative Experiments

To evaluate the proposed method comprehensively, two complementary forms of comparison are reported in this section. Table 6 and Figure 6 present a controlled comparison with general-purpose detectors, whereas Table 7 compares HSF-Net with specialized UAV-oriented small-object detectors using a combination of reproduced and literature-reported results.

Table 6. Comparison with general-purpose detectors on VisDrone. The best result in each column is shown in bold.

Model	mAP_{50} (%)	$mAP_{50–95}$ (%)	GFLOPs	Parameters (M)	Model Size (MB)	FPS
Faster R-CNN [7]	31.5	19.2	253.3	28.4	108.5	34.0
SSD [8]	20.2	11.6	61.8	25.0	95.1	45.2
RT-DETR-L [11]	43.5	27.7	103.5	32.0	63.2	89.8
RetinaNet [25]	19.8	11.8	166.7	36.5	139.6	65.8
YOLOv3	44.8	27.6	261.8	98.5	188.2	86.4
YOLOv5s	38.4	23.0	**18.8**	7.8	15.2	251.9
YOLOv5m	42.2	25.7	52.5	22.1	42.6	172.5
YOLOv5l	43.5	26.9	114.2	47.9	91.9	127.3
YOLOv6s	37.2	22.4	42.5	16.0	30.7	**253.5**
YOLOv6m	41.0	25.1	157.6	51.3	98.1	126.0
YOLOv6l	42.5	26.2	384.9	109.6	209.5	77.0
YOLOv8s	39.1	23.4	23.4	9.8	19.0	261.0
YOLOv8m	42.3	26.0	67.5	23.2	44.6	163.8
YOLOv8l	44.0	27.3	145.2	39.4	75.7	118.5
YOLOv11s	39.3	23.7	21.3	9.4	18.3	212.6
YOLOv11m	43.5	26.6	67.7	20.0	38.7	150.1
YOLOv11L	45.0	28.0	86.6	25.3	48.8	129.7
HSF-Net	**46.4**	**28.5**	39.5	**3.6**	**7.3**	131.3

Table 7. Literature-based comparison with specialized small-object detectors on the VisDrone dataset. NR denotes that the corresponding metric was not reported in the source publication. The best reported value for each metric is shown in bold.

Model	mAP_{50} (%)	$mAP_{50–95}$ (%)	GFLOPs	Parameters (M)	Model Size (MB)	FPS
Hyper-YOLO-S [37]	40.9	24.7	33.8	13.5	26.2	169.6
Hyper-YOLO-M [37]	42.9	26.2	91.8	30.7	59.1	124.6
LUD-YOLO-S [38]	41.7	NR	NR	10.3	20.5	**194.0**
Modified YOLOv8 [27]	42.2	NR	NR	9.7	19.2	167.0
PC-YOLO11s [39]	43.8	26.3	NR	7.1	NR	NR
MSD-YOLO11s [40]	43.8	26.9	NR	11.4	NR	60.1
EBC-YOLO [41]	44.3	26.7	35.5	10.2	20.8	95.6
Drone-YOLO-S [26]	44.3	27.0	NR	10.9	NR	NR
CF-YOLO [12]	44.9	27.5	23.9	3.8	NR	NR
AD-YOLO [42]	45.4	27.8	**14.1**	15.3	NR	192.0
WCDB-YOLO [13]	45.5	27.9	60.1	19.8	NR	30.3
HSF-Net	**46.4**	**28.5**	39.5	**3.6**	**7.3**	131.3

To ensure a fair comparison, every detector in Table 6 was retrained and evaluated under a unified protocol using the same VisDrone training and validation splits, an input resolution of 640 × 640, the training and data-augmentation settings summarized in Section 4.1 and Table 2, and the same NVIDIA RTX 3090 platform. Accordingly, all accuracy and efficiency metrics reported in Table 6, including mAP, GFLOPs, parameter count, model size, and FPS, were obtained within the same experimental pipeline.

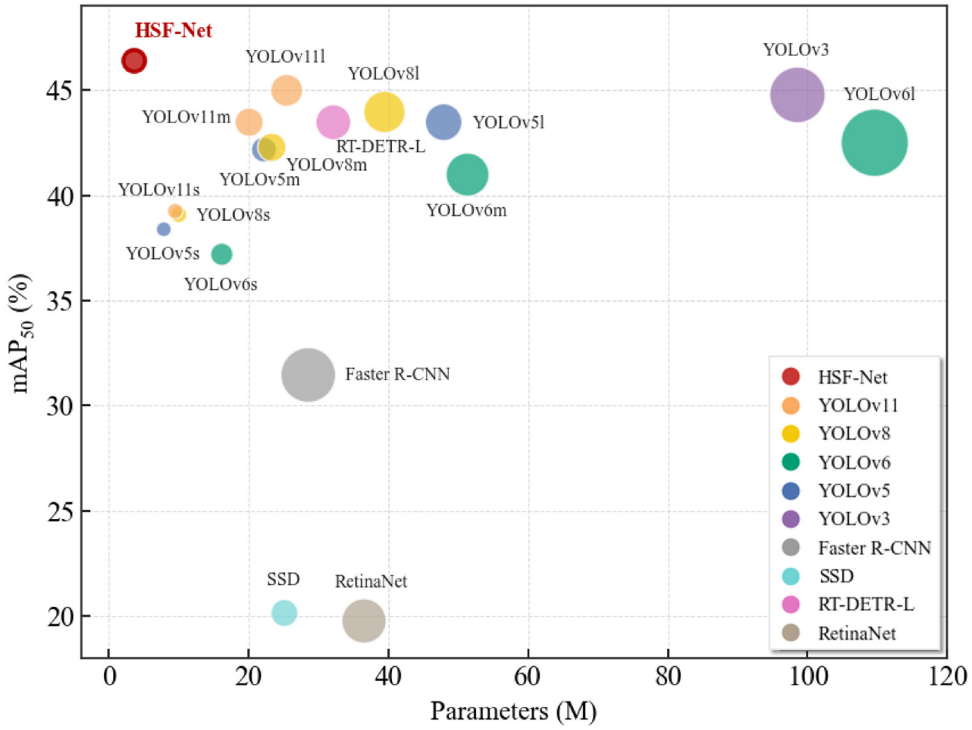


Figure 6. Accuracy–complexity comparison of the general-purpose detectors on the VisDrone dataset. The horizontal coordinate denotes the parameter count, the vertical coordinate denotes mAP_{50}, and the bubble area is scaled according to GFLOPs. All detectors were trained and evaluated under the unified protocol described in Section 4.1, and the corresponding numerical values are reported in Table 6.

Table 7 combines results obtained under the unified protocol with results reported in the literature. Specifically, the results for Hyper-YOLO-S, Hyper-YOLO-M, and HSF-Net were obtained under the same unified protocol, whereas the results for all other specialized detectors were taken from their corresponding original publications. Consequently, comparisons involving the latter methods should be interpreted as contextual rather than strictly controlled, particularly for efficiency metrics that may depend on the input resolution, software environment, and inference hardware.

Among the general-purpose detectors, YOLOv11L attains the highest mAP_{50} at 45.0% yet still falls 1.4 percentage points short of HSF-Net; moreover, it incurs roughly 2.2× the computational cost (86.6 vs. 39.5 GFLOPs) and about seven times as many parameters (25.3 M vs. 3.6 M), indicating that its accuracy is obtained at a markedly higher computational price. The Transformer-based RT-DETR-L exhibits a similar trade-off, trailing HSF-Net by 2.9 percentage points in mAP_{50} while requiring roughly nine times the parameters and over 2.5× the computation. To further verify that the gains do not stem merely from increased capacity, HSF-Net was compared with YOLOv11m, which reaches only 43.5% mAP_{50} despite a substantially larger parameter count and higher computational cost. Collectively, these comparisons confirm that the proposed enhancements improve detection in a far more parameter- and computation-efficient manner, attaining the highest mAP_{50} and $mAP_{50–95}$ while using the fewest parameters among all detectors in this comparison.

Recent UAV-oriented lightweight detectors address small-object detection through different architectural priorities. LYA-YOLO emphasizes split-interactive spatial–channel modulation and a lightweight deformable prediction head; AD-YOLO combines orientation-aware feature extraction with dual-path cross-scale fusion; and WCDB-YOLO adopts wavelet-enhanced target–context dual-backbone modeling. HSF-Net instead combines explicit edge-detail reinforcement, sparse foreground-aware cross-scale fusion, and a high-

resolution P2 prediction branch. These methods alleviate different sources of information loss and background interference, while extremely small object scales, occlusion, and low contrast remain common performance bottlenecks on VisDrone. Therefore, small differences in mAP should be interpreted jointly with model complexity and runtime efficiency. Among the methods reported in Table 7, HSF-Net achieves 46.4% mAP_{50} and 28.5% $mAP_{50–95}$ with 3.6 M parameters and 39.5 GFLOPs, whereas WCDB-YOLO achieves 45.5% and 27.9%, respectively, with 19.8 M parameters and 60.1 GFLOPs. These results indicate that HSF-Net provides a favorable accuracy–compactness trade-off.

In summary, HSF-Net delivers detection accuracy that is competitive with, and in most cases superior to, both general-purpose and specialized detectors, while requiring the fewest parameters and the smallest model size among the compared methods. These results demonstrate that the proposed architectural enhancements strike an effective balance between detection performance and computational cost.

4.7. Cross-Benchmark Evaluation on TinyPerson

To determine whether the performance advantage of HSF-Net remains consistent under a substantially different object-scale distribution and scene background, HSF-Net and the comparison models were further trained and evaluated on the TinyPerson benchmark using its dataset-specific annotations. Compared with VisDrone, TinyPerson contains considerably smaller targets and substantially different maritime backgrounds. Therefore, this experiment evaluates the effectiveness and robustness of the proposed architecture after dataset-specific training on an additional benchmark. The quantitative results and corresponding training curves are reported in Table 8 and Figure 7, respectively.

Table 8. Performance comparison on the TinyPerson dataset. The best result for each metric is shown in bold.

Model	P (%)	R (%)	mAP_{50} (%)	$mAP_{50–95}$ (%)
YOLOv5s	35.2	17.1	15.3	5.3
YOLOv5m	35.2	17.2	15.8	5.5
YOLOv5l	35.5	17.3	15.6	5.5
YOLOv8s	35.0	17.6	15.9	5.6
YOLOv8m	36.0	18.4	16.7	5.8
YOLOv8l	35.6	18.4	17.1	6.1
YOLOv11s	33.5	17.6	15.7	5.5
YOLOv11m	35.7	18.0	16.6	6.0
YOLOv11l	36.0	18.2	16.7	6.0
HSF-Net	**37.1**	**24.4**	**20.9**	**7.4**

As summarized in Table 8, HSF-Net attains the best performance on all four metrics among the compared detectors. Relative to the YOLOv11s baseline, the improvement is most pronounced in recall, which rises from 17.6% to 24.4%, indicating that HSF-Net recovers substantially more of the tiny targets missed by the baseline. Notably, HSF-Net surpasses not only its YOLOv11s baseline but also the larger YOLOv8l and YOLOv11m models, despite their considerably higher capacity, confirming that its advantage stems from the proposed architectural enhancements rather than from increased model size.

Figure 7 complements the endpoint comparison in Table 8 by presenting the evolution of precision, recall, mAP_{50}, and $mAP_{50–95}$ during dataset-specific training. The curves show when the performance differences emerge, whether they remain stable during later optimization, and whether the compared models exhibit pronounced late-stage performance deterioration. HSF-Net reaches a higher stable performance region, particularly in recall, mAP_{50}, and $mAP_{50–95}$, and maintains its advantage over YOLOv11s throughout the later training stage.

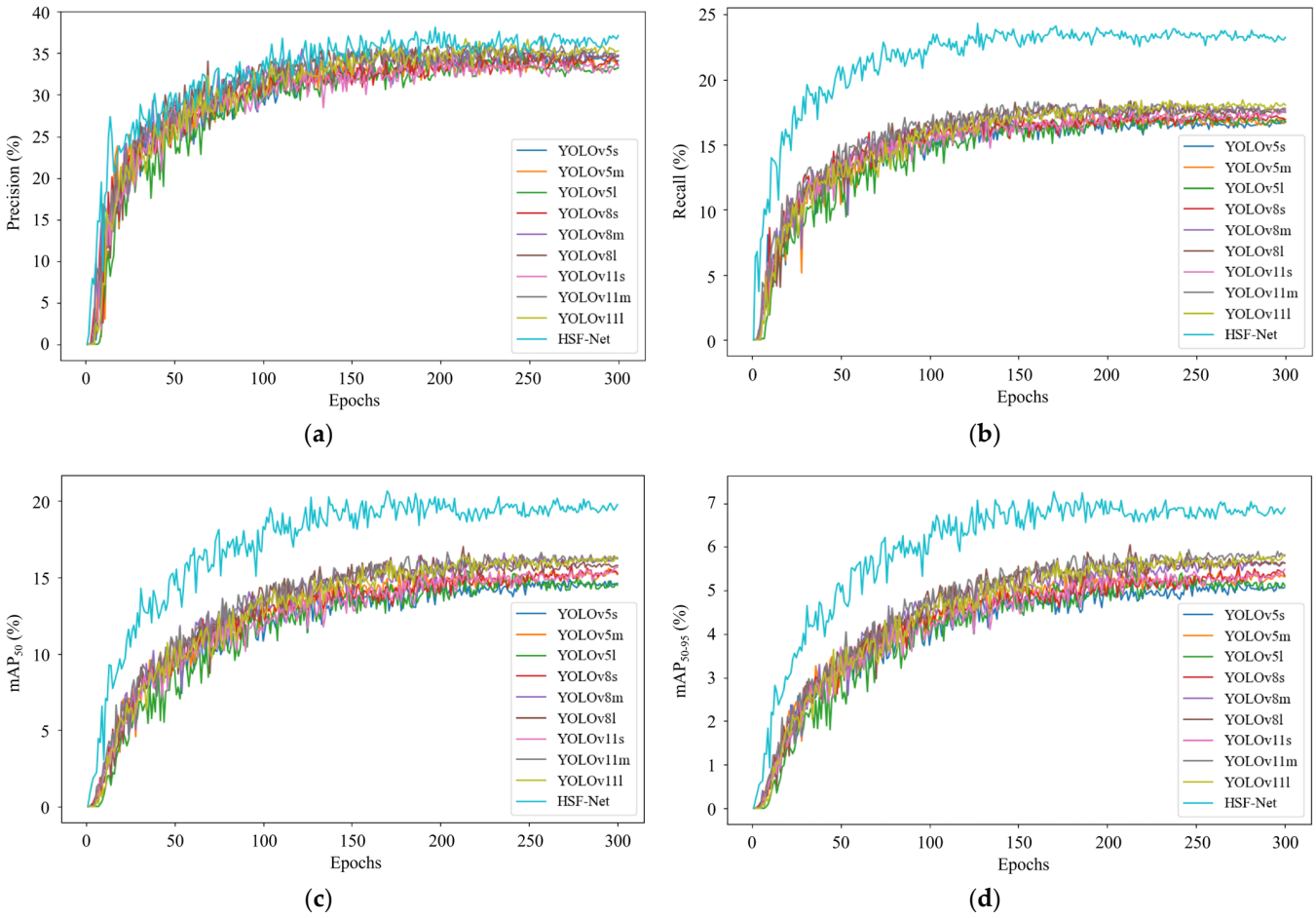


Figure 7. Evolution of the validation metrics of the compared detectors during dataset-specific training on the TinyPerson dataset: (**a**) precision; (**b**) recall; (**c**) mAP_{50}; and (**d**) $mAP_{50–95}$. The exact final values are reported in Table 8, whereas the curves show the optimization trajectories and the persistence of the performance differences over 300 training epochs.

Collectively, these results indicate that the proposed architecture remains effective after dataset-specific training on a benchmark with markedly different scale and background characteristics, demonstrating its robustness under diverse aerial-detection scenarios.

4.8. Qualitative Visualization

To demonstrate the detection performance of the proposed model, several representative aerial images from the VisDrone and TinyPerson datasets were selected, and the detection results of HSF-Net were compared with those of the YOLOv11s baseline, as presented in Figure 8.

As shown in Figure 8a, YOLOv11s fails to detect distant vehicles; similarly, in Figure 8c, it misses pedestrians, bicycles, and motors obscured by tree shadows. In contrast, HSF-Net accurately identifies the targets in all these scenes, exhibiting stronger robustness, particularly for extremely small, long-range objects and in cluttered scenes involving occlusion and shadow.

Figure 8e,f further compare the two models on TinyPerson, whose target-scale distribution and maritime backgrounds differ markedly from those of VisDrone. Persons on the water surface occupy only a few pixels and are easily confused with the surrounding background. YOLOv11s misses several such instances in Figure 8e, whereas HSF-Net detects and localizes them in Figure 8f. These observations are consistent with the quantitative results reported in Section 4.7 and indicate that the proposed architecture remains effective

after dataset-specific training under substantially different tiny-object distributions and background conditions.

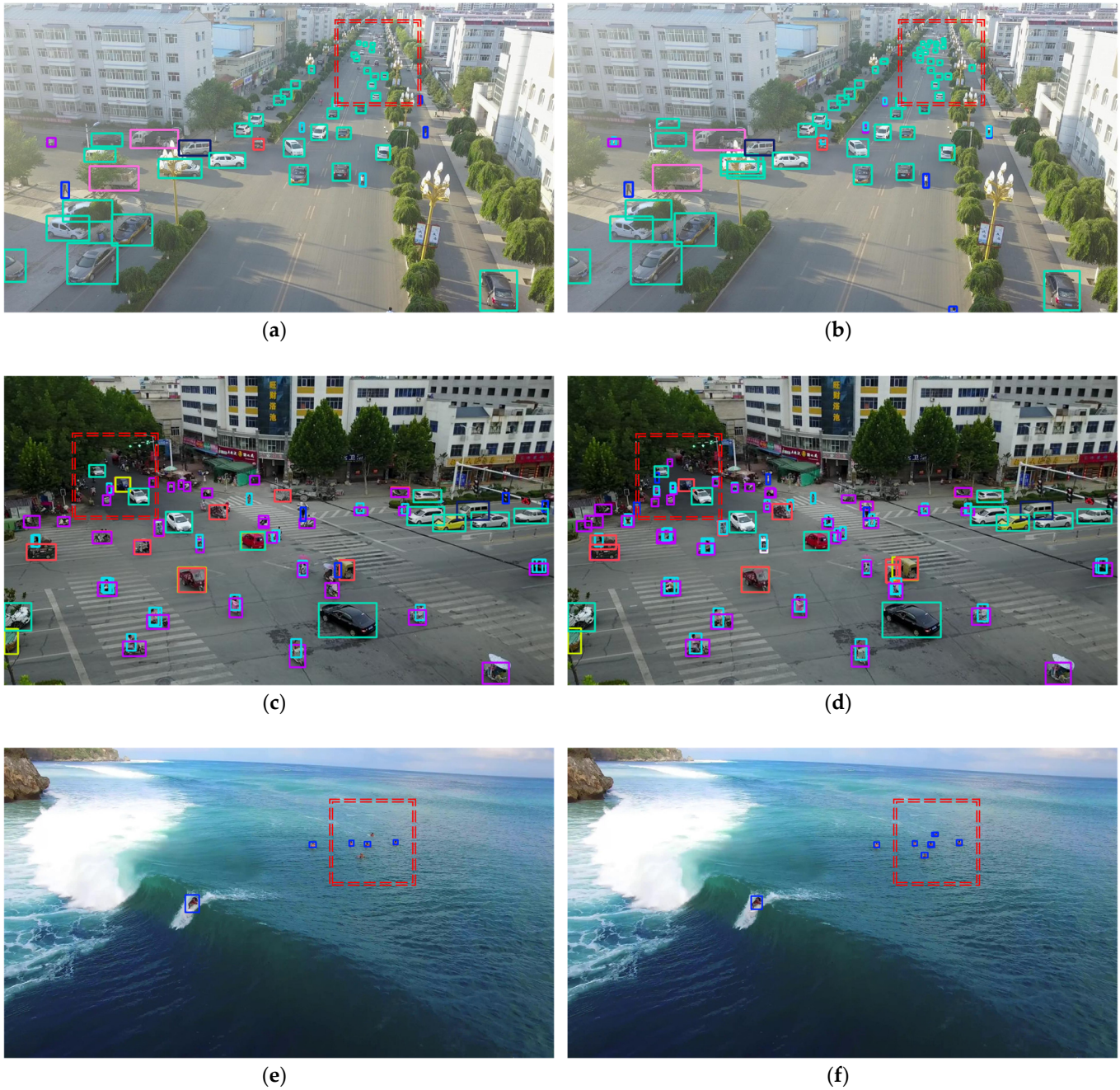


Figure 8. Qualitative comparison between the baseline YOLOv11s and the proposed HSF-Net on representative images from the VisDrone and TinyPerson datasets. Panels (**a**,**c**,**e**) show the results of YOLOv11s, and panels (**b**,**d**,**f**) show the corresponding results of HSF-Net. The red dashed rectangles are author-defined regions of interest placed at corresponding locations in each paired comparison to highlight the regions in which the detection differences between the two models are most evident; they are not detector-generated bounding boxes. The colored solid rectangles denote the predicted object bounding boxes.

5. Conclusions

Small-object detection in UAV imagery is hindered by tiny, densely distributed targets that are easily overwhelmed by cluttered backgrounds. To address this challenge, this paper presented HSF-Net, a lightweight YOLOv11s-based detector incorporating three

complementary enhancements: HEEM, which enhances and re-weights fine-scale edge and texture cues; SAFM, which suppresses background interference and redirects neck capacity toward the sparse foreground regions occupied by small objects; and a restructured {P2, P3, P4} head that adds a high-resolution P2 branch while removing the redundant P5 branch.

On VisDrone, HSF-Net improved mAP_{50} by 7.1 percentage points over the YOLOv11s baseline while reducing the parameter count from 9.4 M to 3.6 M and the model size from 18.3 MB to 7.3 MB. On the markedly different TinyPerson benchmark, it achieved a 5.2-point gain in mAP_{50} and a particularly large 6.8-point gain in recall, and surpassed the larger YOLOv8l and YOLOv11m models, confirming that its advantage stems from the proposed designs rather than from increased capacity.

These improvements in detection accuracy and model compactness are nevertheless accompanied by an increase in computational cost from 21.3 to 39.5 GFLOPs, primarily because of the high-resolution P2 prediction branch. Moreover, because runtime performance was evaluated only on an NVIDIA RTX 3090, the onboard real-time capability of HSF-Net on resource-constrained embedded UAV processors remains to be established. Although the fixed-seed deterministic training protocol improves reproducibility and supports controlled within-study comparisons, it does not quantify variability across independent training runs or establish formal statistical significance. Future work will therefore focus on reducing the computational overhead through lightweight edge-aware operators, structured pruning, quantization, and hardware-aware optimization; evaluating the optimized model on representative embedded platforms; and conducting multi-seed retraining of the principal configurations, with run-to-run variability quantified using standard deviations or confidence intervals, as appropriate, and supported by formal statistical comparisons.

Author Contributions: Conceptualization, J.L.; methodology, J.L.; software, J.L. and J.W.; validation, J.L. and Q.Z.; formal analysis, J.L.; investigation, J.L. and Y.L.; resources, Q.Z.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, Q.Z. and Q.G.; visualization, S.D. and Y.S.; supervision, Q.Z. and Q.G.; project administration, Q.Z.; funding acquisition, Q.Z. and Q.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Defense Industrial Technology Development Program (Grant No. JCKY2025605C007).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Osco, L.P.; Junior, J.M.; Ramos, A.P.M.; de Castro Jorge, L.A.; Fatholahi, S.N.; de Andrade Silva, J.; Matsubara, E.T.; Pistori, H.; Gonçalves, W.N.; Li, J. A review on deep learning in UAV remote sensing. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *102*, 102456. [CrossRef]
2. Mittal, P.; Singh, R.; Sharma, A. Deep learning-based object detection in low-altitude UAV datasets: A survey. *Image Vis. Comput.* **2020**, *104*, 104046. [CrossRef]
3. Cazzato, D.; Cimarelli, C.; Sanchez-Lopez, J.L.; Voos, H.; Leo, M. A survey of computer vision methods for 2D object detection from unmanned aerial vehicles. *J. Imaging* **2020**, *6*, 78. [CrossRef] [PubMed]
4. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the 13th European Conference on Computer Vision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014; pp. 740–755. [CrossRef]
5. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7380–7399. [CrossRef] [PubMed]
6. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]

7. 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*, 1137–1149. [CrossRef] [PubMed]
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot MultiBox detector. In Proceedings of the 14th European Conference on Computer Vision (ECCV 2016), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37. [CrossRef]
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
10. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the 16th European Conference on Computer Vision (ECCV 2020), Glasgow, UK, 23–28 August 2020; pp. 213–229. [CrossRef]
11. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs beat YOLOs on real-time object detection. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974. [CrossRef]
12. Wang, C.; Han, Y.; Yang, C.; Wu, M.; Chen, Z.; Yun, L.; Jin, X. CF-YOLO for small target detection in drone imagery based on YOLOv11 algorithm. *Sci. Rep.* **2025**, *15*, 16741. [CrossRef] [PubMed]
13. Luan, D.; Dong, Y.; Zhou, J.; Li, A.; Xie, L.; Liu, H.; Zhu, J. WCDB-YOLO: Wavelet-enhanced contextual dual-backbone network for small object detection in UAV aerial imagery. *Drones* **2026**, *10*, 155. [CrossRef]
14. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 14–19 June 2020; pp. 10781–10790. [CrossRef]
15. Yang, G.; Lei, J.; Zhu, Z.; Cheng, S.; Feng, Z.; Liang, R. AFPN: Asymptotic feature pyramid network for object detection. In Proceedings of the 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, HI, USA, 1–4 October 2023; pp. 2184–2189. [CrossRef]
16. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
17. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [CrossRef]
18. Itti, L.; Koch, C.; Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1254–1259. [CrossRef]
19. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef] [PubMed]
20. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
21. Sapkota, R.; Karkee, M. Ultralytics YOLO evolution: An overview of YOLO26, YOLO11, YOLOv8 and YOLOv5 object detectors for computer vision and pattern recognition. *arXiv* **2025**, arXiv:2510.09653. [CrossRef]
22. Lin, T.-Y.; Dollár, 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), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
23. 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 (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768. [CrossRef]
24. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV 2018), Munich, Germany, 8–14 September 2018; pp. 3–19. [CrossRef]
25. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
26. Zhang, Z. Drone-YOLO: An efficient neural network method for target detection in drone images. *Drones* **2023**, *7*, 526. [CrossRef]
27. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A modified YOLOv8 detection network for UAV aerial image recognition. *Drones* **2023**, *7*, 304. [CrossRef]
28. Wu, P.; Xu, Y.; Ma, Y.; Zhang, Y.; Xu, Y. LYA-YOLO: A lightweight and accurate YOLO model in drone aerial image scenes. *Expert Syst. Appl.* **2026**, *321*, 132166. [CrossRef]
29. Yu, Z.; Huang, H.; Chen, W.; Su, Y.; Liu, Y.; Wang, X. YOLO-FaceV2: A scale and occlusion aware face detector. *Pattern Recognit.* **2024**, *155*, 110714. [CrossRef]
30. Zheng, Y.; Jing, Y.; Zhao, J.; Cui, G. LAM-YOLO: Drones-based small object detection on lighting-occlusion attention mechanism YOLO. *Comput. Vis. Image Underst.* **2025**, *261*, 104489. [CrossRef]

31. Fu, T.; Dong, H.; Yang, B.; Deng, B. DOMino-YOLO: A deformable occlusion-aware framework for vehicle detection in aerial imagery. *Remote Sens.* **2026**, *18*, 66. [CrossRef]
32. Li, Y.; Li, Q.; Pan, J.; Zhou, Y.; Zhu, H.; Wei, H.; Liu, C. SOD-YOLO: Small-object-detection algorithm based on improved YOLOv8 for UAV images. *Remote Sens.* **2024**, *16*, 3057. [CrossRef]
33. Xu, Z.; Zhao, H.; Liu, P.; Wang, L.; Zhang, G.; Chai, Y. SRTSOD-YOLO: Stronger real-time small object detection algorithm based on improved YOLO11 for UAV imageries. *Remote Sens.* **2025**, *17*, 3414. [CrossRef]
34. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, USA, 4–9 February 2017; Volume 31, pp. 4278–4284. [CrossRef]
35. Bachlechner, T.; Majumder, B.P.; Mao, H.; Cottrell, G.; McAuley, J. ReZero is all you need: Fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), Online, 27–30 July 2021*; de Campos, C., Maathuis, M.H., Eds.; PMLR: Cambridge, MA, USA, 2021; Volume 161, pp. 1352–1361.
36. Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; Han, Z. Scale match for tiny person detection. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 1257–1265. [CrossRef]
37. Feng, Y.; Huang, J.; Du, S.; Ying, S.; Yong, J.-H.; Li, Y.; Ding, G.; Ji, R.; Gao, Y. Hyper-YOLO: When visual object detection meets hypergraph computation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 2388–2401. [CrossRef] [PubMed]
38. Fan, Q.; Li, Y.; Deveci, M.; Zhong, K.; Kadry, S. LUD-YOLO: A novel lightweight object detection network for unmanned aerial vehicle. *Inf. Sci.* **2025**, *686*, 121366. [CrossRef]
39. Wang, Z.; Su, Y.; Kang, F.; Wang, L.; Lin, Y.; Wu, Q.; Li, H.; Cai, Z. PC-YOLO11s: A lightweight and effective feature extraction method for small target image detection. *Sensors* **2025**, *25*, 348. [CrossRef] [PubMed]
40. Zhang, X.; Du, B.; Jia, Y.; Luo, K.; Jiang, L. MSD-YOLO11n: An improved small target detection model for high precision UAV aerial imagery. *J. King Saud Univ. Comput. Inf. Sci.* **2025**, *37*, 100. [CrossRef]
41. Luo, H.; Wang, Y.; Chen, Y.; Li, X.; Zhan, J.; Zuo, D. EBC-YOLO: A remote sensing target recognition model adapted for complex environments. *Earth Sci. Inform.* **2025**, *18*, 282. [CrossRef]
42. Deng, Y.; Hu, Y.; Ye, Y.; Xu, P. AD-YOLO: A unified method for traffic-dense and small object detection in UAV images. *Drones* **2026**, *10*, 338. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.