remote sensing

MDPI

Article

DPF-DETR: Enhancing Drone Image Detection with Density Perception and Multi-Scale Feature Fusion

Sidi Lai [1], Zhensong Li [1,*], Xiaotan Wei [1], Yutong Wang [1] and Shiliang Zhu [2]

[1] Center for Target Cognition Information Processing Science and Technology, College of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100080, China; laisidi@bistu.edu.cn (S.L.); 2023020554@bistu.edu.cn (X.W.); w18515656301@163.com (Y.W.)

[2] The State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; 2024186190094@whu.edu.cn

* Correspondence: lizhensong@bistu.edu.cn

Highlights

What are the main findings?

- We propose a novel density sensing mechanism that adaptively adjusts object queries based on object density, and introduce a multi-scale feature fusion network to enhance detection performance.
- The proposed method delivers superior performance on the Visdrone2019 dataset, surpassing competing methods across all key metrics.

What is the implication of the main finding?

- By introduction fine-gravened supervision, our approach effective resolution the parse supervision limited of traditional methods, provides distinct benefits for DETR models.
- It provides technical support for key applications such as urban transportation, crop classification, and environmental monitoring.

Abstract

The DPF-DETR model has been designed to address the challenges encountered in object detection within drone imagery, particularly in scenarios involving significant target scale variations, dense targets, and complex backgrounds. To overcome the limitations of traditional object detection methods, the Density Sensing Mechanism (DSM) and Adaptive Density Map Loss (AdaptiveDM Loss) have been incorporated into the model to provide fine-grained supervision signals. The DSM optimizes the query selection mechanism by utilizing density maps, enabling the number of queries to be adaptively adjusted based on the distribution density of targets, thus improving detection accuracy in dense regions. Furthermore, the precision of the model in detecting dense targets is enhanced by AdaptiveDM Loss, which dynamically adjusts the weights for object localization and classification. Multi-scale feature fusion capabilities are also improved by the Multi-Scale Feature Fusion Network (MSFFN) and the Selective Feature Integration Module (SFIM). The MSFFN refines the fusion of features, which improves the detection of targets across various scales, particularly in complex scenes. Additionally, SFIM enhances the detection accuracy for small targets and complex backgrounds by integrating low-level spatial features with high-level semantic information. The Context-Sensitive Feature Interaction Module (CSFIM) further optimizes multi-scale feature fusion through context-guided interactions, bridging the semantic gap between features of different scales, thus improving the robustness of the model in dense scenarios. Experimental results have shown that DPF-DETR outperforms traditional models and state-of-the-art detection methods across

Check for updates

Academic Editors: Maofan Zhao, Shouhang Du, Jancsó Tamás, Abhishek Singh and Jiahao Wu

Received: 17 March 2026
Revised: 7 April 2026
Accepted: 15 April 2026
Published: 17 April 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.

multiple datasets, demonstrating superior robustness and accuracy, especially in dense target detection and complex background scenarios.

Keywords: aerial remote sensing; multi-scale fusion; density-aware mechanism; real-time object detection

1. Introduction

With the rapid development of drone technology, object detection based on drone imagery has become an important research direction in the field of computer vision [1]. Objects in drone imagery often exhibit significant scale variations, and in complex environments, they may be subject to occlusion, blurriness, and lighting changes, which make object detection particularly challenging [2]. Traditional object detection methods, such as Faster R-CNN [3] and the YOLO series [4–7] based on convolutional neural networks (CNNs), have achieved significant progress on standard datasets; however, their performance remains limited when handling targets with highly variable scales, complex backgrounds, and dense targets [8]. Most existing methods rely on anchor box mechanisms or post-processing steps to locate targets [9], but false positives [10] or missed detections [11] often arise, particularly when dense or small targets are detected.

To address these challenges, the DPF-DETR model is introduced in this study, which is based on the DETR [12] (Detection Transformer) architecture and incorporates several innovative techniques. First, the Density Sensing Mechanism (DSM) is introduced, which generates density maps [13] to provide fine-grained supervision signals [14] for the model, thereby enabling a better capture of the spatial distribution of targets. Compared with traditional methods, DSM employs a dynamic query selection mechanism that adaptively adjusts the number of queries based on target distribution density, effectively improving detection accuracy in dense regions [15]. Second, the Multi-Scale Feature Fusion Network (MSFFN) and the Selective Feature Integration Module (SFIM) are proposed to optimize multi-scale feature fusion [16]. The former strengthens the fusion of multi-scale features through an improved pyramid structure, while the latter optimizes the combination of low-level spatial features and high-level semantic information, enhancing the model's ability to detect targets of different scales [17]. To further enhance the model's multi-scale adaptability, the Context-Sensitive Feature Interaction Module (CSFIM) is introduced, which optimizes multi-scale feature fusion by guiding it with contextual information, bridging the semantic gap between features of different scales and thereby improving the model's performance in complex scenes [18]. Finally, the Adaptive Density Map Loss (AdaptiveDM Loss) is proposed, which adaptively adjusts the weights to optimize target localization and classification performance, excelling in detecting dense and small targets [19]. The contributions of this study are summarized as follows:

1. The DSM framework is proposed, utilizing density maps as fine-grained supervision signals and dynamically adjusting the number of queries to enhance detection accuracy in dense target regions. This module adaptively focuses on dense regions based on the target distribution density while reducing queries in sparse regions to avoid unnecessary computational costs. In this manner, DSM significantly improves detection accuracy in complex backgrounds, while simultaneously optimizing computational efficiency and minimizing computational burden.
2. The MSFFN and SFIM are combined to optimize multi-scale feature fusion, allowing the model to better handle targets of varying scales. MSFFN enhances the efficiency of multi-scale feature fusion through an improved pyramid structure, while the SFIM

effectively integrates low-level spatial features with high-level semantic information. Through the collaboration of these two modules, DPF-DETR demonstrates superior detection accuracy in complex scenes, while maintaining a low computational load and reduced parameter count.

3. The CSFIM is proposed to bridge the semantic gap between features of different scales through context-guided feature fusion. The CSFIM intelligently adjusts the fusion ratio of features at different scales, thus improving multi-scale target detection accuracy. This module further strengthens the robustness of DPF-DETR in dense scenes and complex backgrounds, enhancing the overall detection performance.

2. Related Works

2.1. Object Detection in Drone Imagery

Object detection in drone imagery is often confronted with significant challenges due to large variations in target scales, complex backgrounds, and dense targets. While these scale variations fundamentally follow from traditional camera geometry [20] (i.e., perspective projection, where object size is inversely proportional to distance), their magnitude and impact are substantially amplified in drone-captured images because of the diverse heights, angles, and environments in which these images are acquired [18]. These amplified geometric effects—such as extreme scale changes, frequent occlusions, and highly variable object densities—pose substantial difficulties for conventional object detectors, which are typically designed for more moderate conditions. We also note that the presence and severity of these challenges depend on specific scene characteristics; in low-altitude or sparse scenes [21], such difficulties may be less pronounced. Early object detection approaches, such as Faster R-CNN [3] and SSD [22], employed sliding windows and region proposal mechanisms, but they performed inadequately on small and overlapping targets in drone imagery. Subsequently, the YOLO series [23–25] improved detection speed by utilizing single-stage, end-to-end prediction. YOLOv7 [26] and YOLOv8 [24] further enhanced detection capabilities for targets of varying scales; however, these models still depend on predefined anchor boxes, which fail to sufficiently address the challenges posed by drastic scale changes and dense targets in drone imagery. With the advent of Transformer-based architectures, end-to-end methods such as RT-DETR [27] and DINO [28] eliminate the need for anchor boxes by using set-based prediction and bipartite matching loss [12]. However, these models still encounter difficulties in detecting dense small targets and complex backgrounds in drone imagery due to a lack of strong constraints on local details [29]. As a baseline model, D-Fine [30] addresses robustness issues in complex scenarios using FDR (Fine-grained Distribution Refinement) and GO-LSD (Global Optimal Localization Self-Distillation) techniques, which effectively handle multi-scale targets and complex backgrounds in drone imagery. Despite this, D-Fine continues to face challenges in detecting dense and small targets, particularly when targets are closely spaced or occluded, resulting in a significant reduction in detection accuracy.

2.2. Fine-Grained Supervision Signals and Dynamic Query Mechanism

Traditional object detection methods often rely on coarse bounding box supervision, which fails to offer sufficient detailed information for enhancing detection accuracy in dense target scenarios. DETR [12], employing an end-to-end Transformer architecture, successfully removes anchor boxes and post-processing steps, but continues to struggle with dense object detection. DINO [28] further enhances DETR by refining the query selection mechanism, improving target localization accuracy. However, DINO still faces challenges in fine-grained target capture in dense scenes. DQ-DETR [31] introduces a density-aware query selection mechanism, leveraging density maps to optimize the spatial distribution

modeling of targets. Although this approach improves the model's performance to some extent, it still lacks precise target boundary information and exhibits limitations when handling dense targets [32]. To overcome these issues, the Density Sensing Mechanism (DSM) proposed in this study introduces fine-grained supervision signals through adaptive density map generation and dynamic query mechanisms. DSM not only provides the model with detailed supervision signals but also adapts the number of queries based on the distribution density of targets. This significantly enhances detection accuracy in dense target regions, thereby improving the overall performance of the model.

2.3. Multi-Scale Feature Fusion

Multi-scale feature fusion plays a critical role in target detection, particularly when dealing with targets of varying sizes [33]. FPN [34] and PANet [35] are recognized as classic solutions for multi-scale feature fusion. FPN constructs a feature pyramid to enhance detection capabilities across different target scales, yet it still faces issues with information loss in complex scenes [36]. PANet improves information flow by utilizing path aggregation, but continues to encounter redundancy and information loss, especially when processing small targets and complex backgrounds [37]. BiFPN [38] enhances feature fusion efficiency through bidirectional information flow, yet it does not fully address the semantic gap between multi-scale features during the fusion process [39]. In response to these challenges, this paper proposes the Multi-Scale Feature Fusion Network (MSFFN) and the Selective Feature Integration Module (SFIM). MSFFN optimizes the multi-scale feature fusion process by employing an improved pyramid structure, while SFIM further enhances the integration of low-level spatial features with high-level semantic information, thereby improving detection accuracy for small targets and complex backgrounds. Finally, the Context-Sensitive Feature Interaction Module (CSFIM) guides the multi-scale feature fusion process using contextual information, effectively bridging the semantic gap between features at different scales and enhancing the robustness of the model in complex background and dense target scenarios [40].

3. Methods

This section presents a comprehensive description of the DPF-DETR model, which is built upon the D-Fine framework. Section 3.1 outlines the overall architecture of DPF-DETR, emphasizing the innovations within each component and their interconnections. In Section 3.2, the implementation details and operational principles of the Density Sensing Mechanism (DSM) are discussed. Section 3.3 examines the specifics of the feature aggregation module and explains the functionality of the Feature-Focused Diffusion Pyramid Network (FFDPNet). Section 3.4 introduces the principles behind the Context-Sensitive Feature Interaction Module (CSFIM) and explains its role in bridging the semantic gap between multi-scale features by replacing concatenation in multi-scale feature fusion. Finally, Section 3.5 details how the Adaptive Density Map Loss (AdaptiveDM Loss) enhances both localization and classification, followed by an analysis of the stability of its hyperparameters.

3.1. Overall Architecture

The architecture of the DPF-DETR model is illustrated in Figure 1. The design of DPF-DETR is specifically intended to address the challenges posed by significant target scale variations, complex feature extraction, and dense object detection in drone imagery. Drone-captured images typically present intricate scenes in which target size, pose, lighting, and occlusions vary considerably, resulting in the poor performance of traditional object detection methods in such environments. To overcome these challenges, DPF-DETR integrates

several innovative modules, significantly enhancing detection accuracy and efficiency, particularly in scenarios with dense objects and complex backgrounds. The model incorporates the Density Sensing Mechanism (DSM), Multi-Scale Feature Fusion Network (MSFFN), Context-Sensitive Feature Interaction Module (CSFIM), and Adaptive Density Map Loss (AdaptiveDM Loss), optimizing target localization and detection accuracy through multi-level and multi-scale feature extraction and fusion.

To address the shortcomings of traditional bounding box supervision, the Density Sensing Mechanism (DSM) introduces density maps as fine-grained supervision signals for the object detection task. Density maps not only provide object-level spatial information but also effectively model the distribution characteristics of objects, offering richer gradient signals for network optimization. MSFFN replaces the conventional feature pyramid structure in the baseline model and, in particular, when handling multi-scale targets, integrates the Selective Feature Integration Module (SFIM). This module enables precise adjustments to the fusion of low-level spatial features and high-level semantic information. Furthermore, MSFFN optimizes information flow and fusion through a feature diffusion mechanism, significantly improving the model's performance in multi-scale target detection. The CSFIM replaces the feature fusion process of the traditional baseline model. Through context-guided feature fusion, CSFIM more effectively integrates low-level spatial features with high-level semantic information, refining the feature fusion process, especially in dense target and complex scenarios. Finally, the Adaptive Density Map Loss (AdaptiveDM Loss) is proposed to balance overestimation and underestimation errors via asymmetric pixel-wise weighting based on density map prediction errors. Unlike existing re-weighting methods that focus on anchor assignment [41] or sample-category redistribution [42], AdaptiveDM Loss applies distinct recall and precision penalties to underestimation and overestimation regions, respectively, with particular emphasis on higher-error regions in the density map.

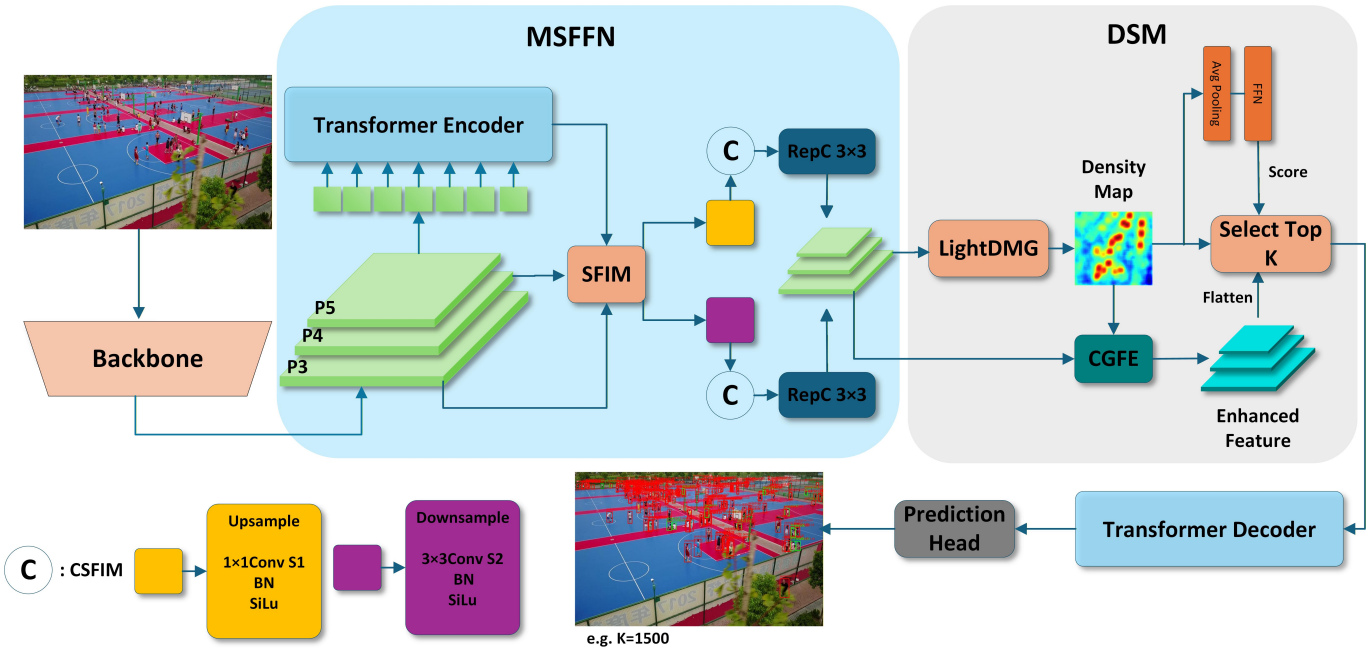


Figure 1. This is the overall architecture of DPF-DETR, where RepC3×3 is the same as RepC3×3 in RT-DETR.

3.2. Density Sensing Mechanism

In object detection tasks, traditional decoder models based on Transformers [43–45] have been widely applied to remote sensing imagery. However, their limitation in utilizing

fixed query numbers reduces their effectiveness in handling remote sensing datasets with significant variations in object densities. Traditional decoders fail to consider the object density distribution within the image, leading to unnecessary computational resource consumption in dense object regions, while missing important targets in sparse regions [46]. To address this limitation, the Density Sensing Mechanism (DSM) is proposed. This framework introduces density maps along with a dynamic query selection mechanism that adaptively adjusts the number of queries based on the object density distribution, thereby improving both detection efficiency and accuracy. The architecture of DSM is illustrated in Figure 2.

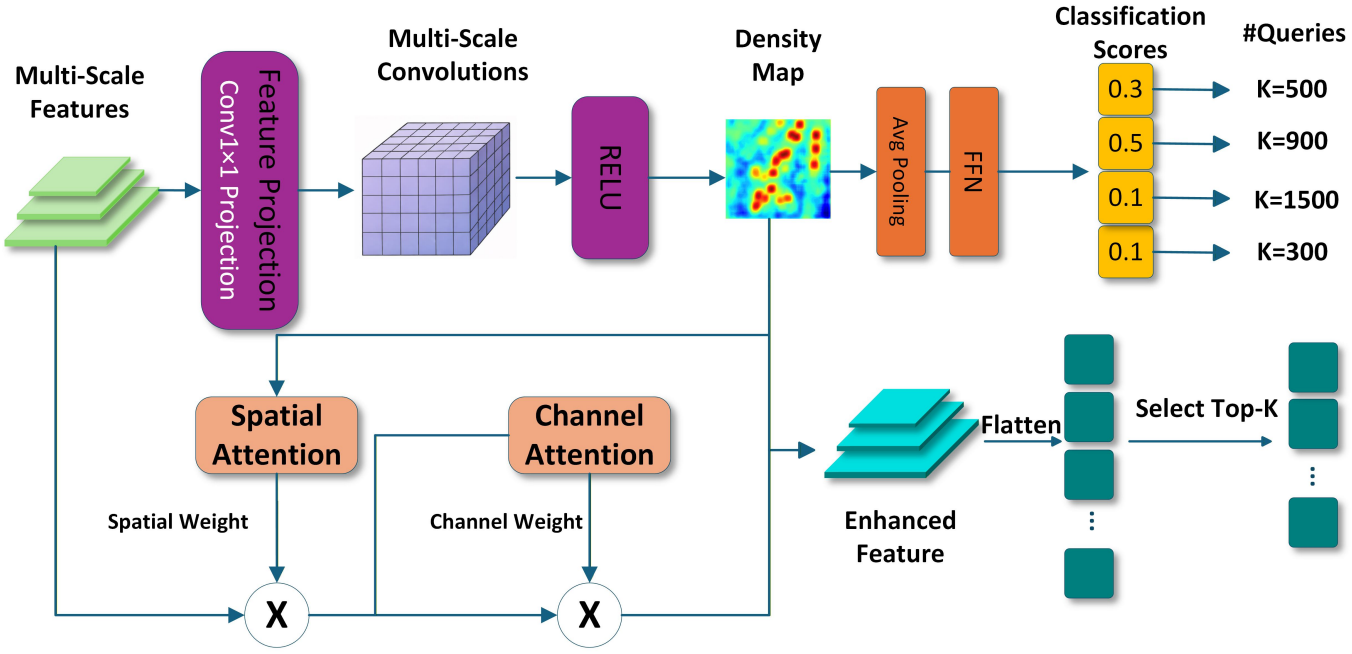


Figure 2. The architecture of DSM.

In the initial stage of object detection, the original remote sensing image undergoes feature extraction through a Convolutional Neural Network (CNN) [47–49]. These feature maps are then processed by the DSM framework. Traditional decoders typically use feature maps of fixed dimensions for subsequent reasoning, which may lead to suboptimal performance for objects of different scales. To solve this problem, we incorporate multi-scale feature processing in the DSM framework. Through `1×1 convolutions`, features from different layers and scales are projected onto a unified hidden dimension (`hidden_dim`), allowing the subsequent query selection and decoding process to better integrate information from various scales. The input feature map F is projected through a `1×1 convolution`, resulting in the projected feature map F_{proj}, as expressed by the following equation:

$$F_{\text{proj}} = \text{Conv1}\times1(F), \tag{1}$$

where F is the input feature map and F_{proj} is the projected feature map, and `Conv1×1` denotes the 1×1 convolution operation.

To address object detection in dense regions, a lightweight Density Map Generator (LightDMG) is proposed. This module models the spatial distribution of objects using an adaptive multi-scale density-aware mechanism, explicitly capturing density characteristics from shallow features. LightDMG employs parallel deep convolutions [50] with varying kernel sizes and progressively increases the receptive field ($3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9, 11 \times 11$) to capture multi-scale density features. Smaller kernels are utilized to capture the details

in sparse regions, while larger kernels aggregate contextual information, thus modeling trends in dense regions. The concatenation of multi-scale features allows the network to adaptively respond to density variations at different spatial locations. The architecture of LightDMG is depicted in Figure 3.

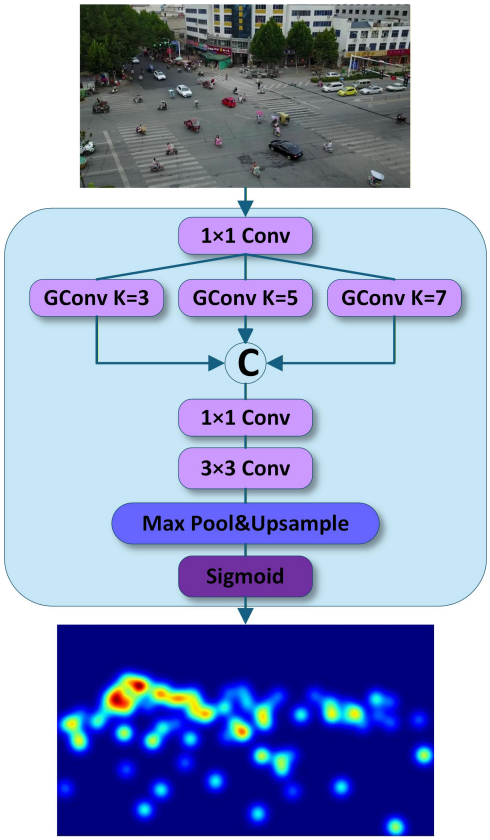


Figure 3. The architecture of Lightweight Density Map Generator (LightDMG).

In the density map generation process, the predicted density map D_{pred} and the ground truth density map D_{gt} are first normalized to ensure that their values lie within the range of [0, 1]. The normalization operation is performed as follows:

$$\hat{D}_{\text{gt}} = \frac{D_{\text{gt}}}{\max(D_{\text{gt}}) + \epsilon}, \quad \hat{D}_{\text{pred}} = \frac{D_{\text{pred}}}{\max(D_{\text{pred}}) + \epsilon}, \tag{2}$$

where ϵ is a small constant (e.g., 10^{-6}) to avoid division by zero. The normalized density maps are $\hat{D}_{\text{gt}}$ and $\hat{D}_{\text{pred}}$, and the difference ΔD between the two maps is computed as the input for the loss function:

$$\Delta D = \hat{D}_{\text{pred}} - \hat{D}_{\text{gt}}, \tag{3}$$

Figure 4 illustrates an example of dense crowd detection in a UAV aerial image. The left panel displays the original scene containing pedestrians and stalls; the middle panel presents the predicted bounding boxes (count: 101), marked in green; and the right panel shows the corresponding density map, where warmer colors represent higher target density. This figure highlights the model's ability to accurately capture both the locations and spatial distribution of dense targets.

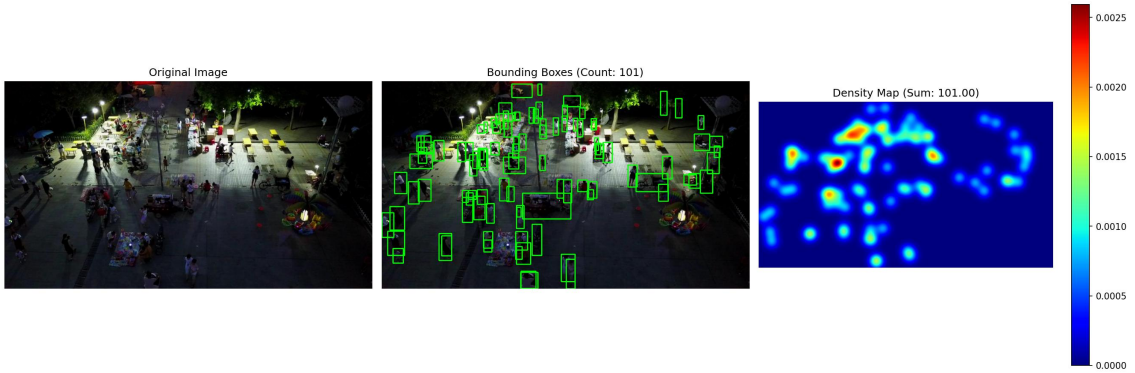


Figure 4. Visualization of density estimation on a UAV aerial image.

Next, we perform a weighted calculation of the density map difference to assign different weights to various regions of the map during the loss calculation. The adaptive weight $W(i,j)$ is computed as follows, incorporating penalties for recall and precision:

$$W(i,j) = 1 + \lambda_{\text{recall}} \cdot \hat{D}_{\text{gt}}(i,j) \cdot \mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) < \hat{D}_{\text{gt}}(i,j))} + \lambda_{\text{precision}} \cdot \hat{D}_{\text{gt}}(i,j) \cdot \mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) > \hat{D}_{\text{gt}}(i,j))}, \tag{4}$$

where $\mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) < \hat{D}_{\text{gt}}(i,j))}$ and $\mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) > \hat{D}_{\text{gt}}(i,j))}$ are indicator functions determining whether recall or precision penalties are applied based on the relationship between the predicted and ground truth density maps. The weight coefficients λ_{recall} and $\lambda_{\text{precision}}$ control the influence of recall and precision penalties, respectively.

The density loss L_{density} is calculated as the average of the weighted squared differences for each pixel, as shown in the following equation:

$$L_{\text{density}} = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} W(i,j) \cdot (\Delta D(i,j))^2, \tag{5}$$

where H and W represent the height and width of the density map, $W(i,j)$ is the adaptive weight at position (i,j), and $\Delta D(i,j)$ is the difference between the predicted and ground truth density maps at that position. The final density loss is calculated using the weighted squared difference, encouraging the model to fit more precisely in the dense regions of the density map while avoiding computational waste in sparse regions.

Next, the DSM framework enters the query selection phase. Unlike traditional decoders, which utilize a fixed number of queries, queries are dynamically selected in DSM through the introduction of density maps. Specifically, in each inference round, queries are selected based on the predicted category logits and density map values. In dense regions, a larger number of queries are chosen to enhance attention to objects in those areas, while fewer queries are selected in sparse regions to avoid unnecessary computational waste. This mechanism is implemented as follows:

$$Q_k = \text{topk}(\text{Logits}_i \times \text{DensityMap}_i^{(1/T)}), \tag{6}$$

where Logits_i represents the category prediction logits, DensityMap_i is the density map, and T is the temperature parameter. By applying temperature scaling (i.e., exponentiation) to the density map, the model is guided to focus more on query selection in dense regions.

Anchor generation is considered an essential part of object detection, as candidate regions are provided through it. In traditional decoders, fixed-size anchors are used, which fail to adapt to varying object sizes. In DSM, multi-level anchors are generated at different

scales and are optimized based on the density map. The formula for anchor generation is as follows:

$$\text{Anchor}_i = (x_i, y_i, W_i, H_i, w_i, h_i), \tag{7}$$

where (x_i, y_i) represents the center coordinates of the anchor, W_i and H_i are the width and height of the feature map, and w_i and h_i are the width and height of the anchor. The density map influences anchor generation, ensuring that dense regions receive more anchors.

In the decoding phase, object localization is performed by the DSM framework through the utilization of query information and input features via the Transformer decoder. Unlike traditional Transformer decoders, which do not account for density distribution, the DSM framework directs the decoder to update queries based on dynamic query selection and density maps. This results in more accurate predictions for both bounding boxes and object classes. The decoding process is expressed as follows:

$$B_{\text{pred}} = \text{Decoder}(Q_k, F, A), \tag{8}$$

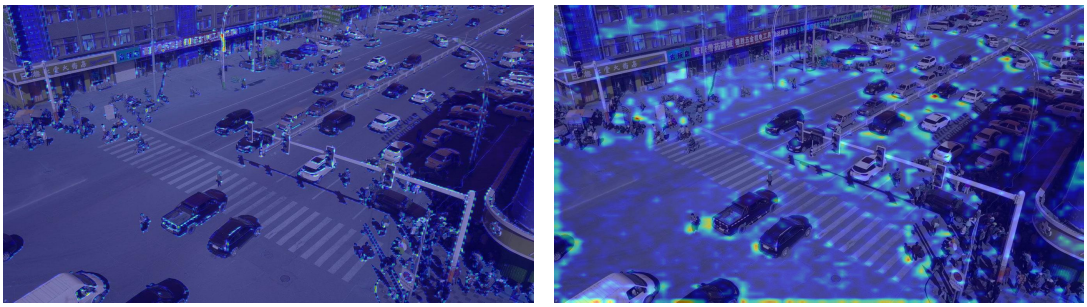
where B_{pred} is the predicted bounding box, Q_k is the selected query set, F is the input feature map, and A is the generated anchors.

The DSM framework also incorporates denoising training to improve the model's robustness when facing noisy or partially missing targets. Denoising training is achieved through contrastive loss computation, as shown in the following formula:

$$L_{\text{denoise}} = \text{Contrastive Loss}(Y_{\text{denoised}}, Y_{\text{original}}), \tag{9}$$

where Y_{denoised} is the denoised prediction and Y_{original} is the original ground truth.

Figure 5 compares the thermal map responses of the original D-Fine model and the model with the added DSM module. As shown in Figure 5a, the thermal distribution of the D-Fine model is relatively diffuse in dense vehicle areas, with obvious response adhesion between multiple targets, making it difficult to distinguish vehicle boundaries. After adding DSM (Figure 5b), the thermal values are significantly concentrated towards the target center, the highlighted areas of each vehicle are more compact, and the thermal differentiation between adjacent targets is significantly improved. This indicates that DSM can guide the model to more accurately locate target positions and effectively alleviate the feature confusion problem in dense scenes.



(**a**) The Heatmap of D-Fine (**b**) The Heatmap of D-Fine after adding DSM

Figure 5. Comparison of visual heat maps in scenarios with confusing objects.

In conclusion, the DSM framework, by utilizing density maps and dynamic query selection, effectively overcomes the computational limitations present in traditional models when handling substantial variations in object density. As a result, detection efficiency and accuracy are significantly enhanced, along with improved robustness in complex scenes. When compared to D-Fine, DSM exhibits superior performance and greater computational

efficiency, particularly in scenarios characterized by uneven object distribution, such as in remote sensing imagery.

3.3. Multi-Scale Feature Fusion Network

In object detection tasks for remote sensing images, traditional convolutional neural networks [3] and Transformer [12] models have made considerable advancements, yet they continue to encounter challenges in handling multi-scale feature fusion, discrepancies in feature map sizes, and overall detection accuracy [51]. This issue is particularly pronounced in remote sensing images, where there are significant differences in object sizes and regions with varying scales and densities, necessitating the effective handling and fusion of information from multiple scales. To address these challenges, the Multi-Scale Feature Fusion Network (MSFFN) framework is proposed. This framework, which combines Transformer and convolutional layers, employs a two-stage feature extraction and multi-scale feature fusion mechanism, providing an effective solution for remote sensing object detection. A key innovation within the MSFFN framework is the Selective Feature Integration Module (SFIM), which plays an essential role in multi-scale feature processing and fusion. The specific structure of MSFFN is depicted in Figure 6.

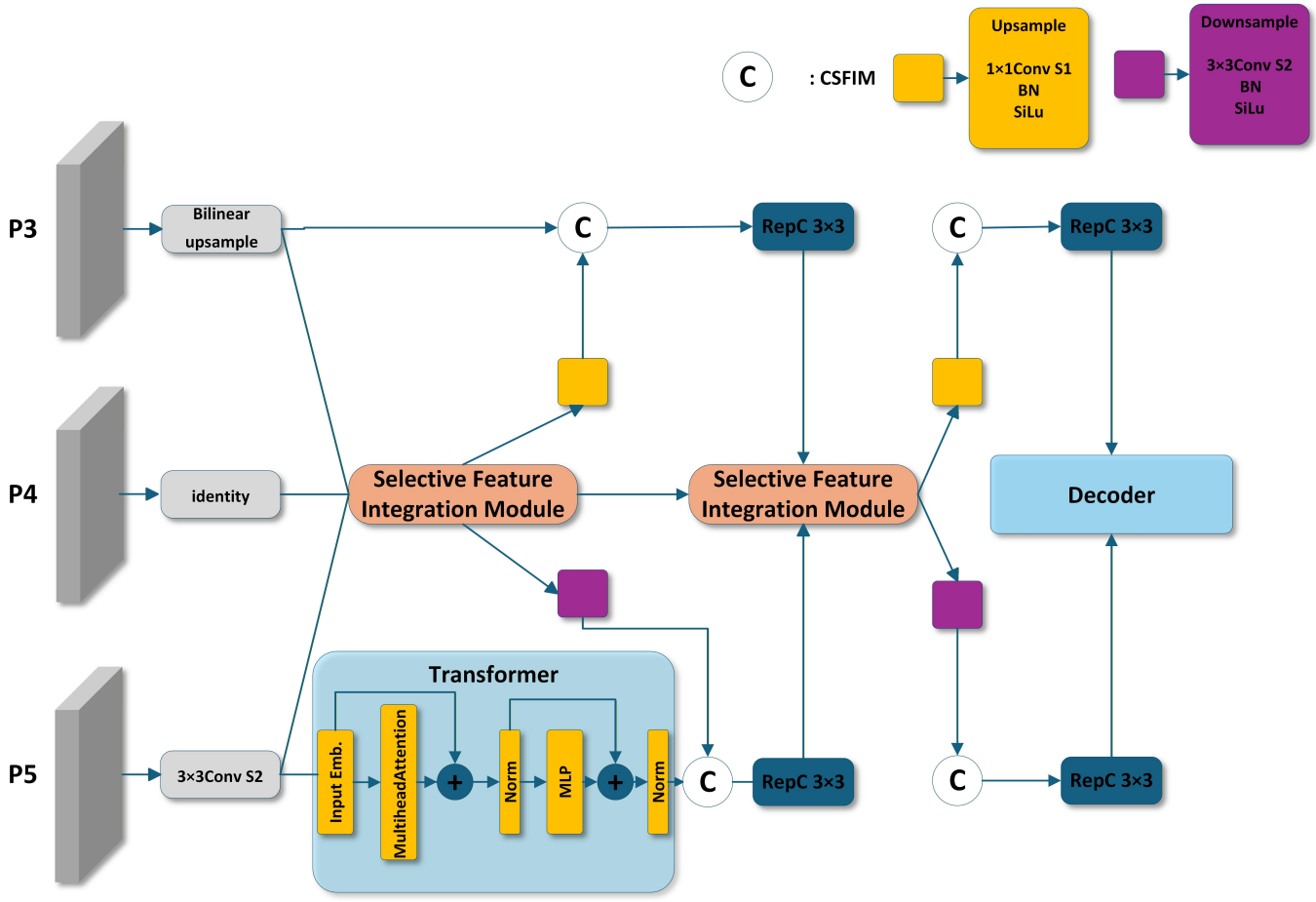


Figure 6. This is the architecture of MSFFN, where RepC3×3 is the same as RepC3×3 in RT-DETR. It projects multi-scale backbone features to a unified dimension, enhances them with a Transformer encoder, and fuses them via the Selective Feature Integration Module (SFIM) to produce robust multi-scale representations.

The input to the MSFFN framework consists of feature maps at different scales from the backbone network. First, each input feature map is adjusted to a unified hidden dimension (`hidden_dim`) using a `1×1 convolution`, ensuring that features from different scales can

be effectively fused during the subsequent processing stages. The specific operation is as follows:

$$F^i_{\text{proj}} = \text{Conv1}\times1(F^i), \tag{10}$$

where F^i is the input feature map at the i-th layer, and F^i_{proj} is the feature map after applying the 1×1 convolution, where the channel size is adjusted to `hidden_dim`.

Subsequently, a Transformer encoder is incorporated into MSFFN to enhance the features of each feature map layer. In contrast to traditional models, MSFFN applies the Transformer encoder to the feature maps of each individual layer, enabling the model to capture long-range dependencies through self-attention mechanisms and thereby improving the representation of object features. The operation is expressed as follows:

$$M^i = \text{TransformerEncoder}(F^i_{\text{proj}}, P^i_{\text{pos}}), \tag{11}$$

where F^i_{proj} is the input feature map after the 1×1 convolution, P^i_{pos} is the corresponding positional encoding at the i-th layer, and M^i is the feature map after processing by the Transformer encoder.

In the MSFFN framework, the SFIM represents one of the key innovations, responsible for processing and integrating feature maps from various scales. Initially, the SFIM adjusts the channel size of each input feature map to "hidc" via convolutions, followed by downsampling using the ADown module [52]. These operations enable SFIM to effectively capture multi-scale features and perform the fusion of features from different scales. The structure of SFIM is illustrated in Figure 7. For each input feature map, the channel size adjustment is performed as described by the following formula:

$$F_1' = \text{Conv}(F_1, \text{hidc}), \quad F_2' = \text{Conv}(F_2, \text{hidc}), \quad F_3' = \text{ADown}(F_3, \text{hidc}), \tag{12}$$

where F_1, F_2, and F_3 are the input feature maps from different scales, and "hidc" is the adjusted channel size.

Next, SFIM further processes the features through depthwise convolutions [53] and pointwise convolutions [54], as expressed by the following formula:

$$F_{\text{feature}} = \sum_{k=1}^{K} [F + \text{Conv}_{\text{dw}}(F)], \tag{13}$$

where $\text{Conv}_{\text{dw}}(F)$ denotes the depthwise convolution operation, and K represents the number of convolution layers. The fused feature map is then processed through pointwise convolutions:

$$F_{\text{feature}} = \text{Conv}_{\text{pw}}(F_{\text{feature}}), \tag{14}$$

Finally, SFIM adjusts the processed feature map to the final output through a 1×1 convolution:

$$F_{\text{out}} = \text{Conv}_{1\times1}(F_{\text{feature}}), \tag{15}$$

where F_{feature} is the fused feature map, and F_{out} is the final output feature map.

Multi-scale feature fusion in MSFFN is achieved through downsampling and upsampling operations. Specifically, operations such as $p_{4\to5_down1}$ and $p_{4\to3_up1}$ downsample high-resolution feature maps to low resolution or upsample low-resolution feature maps to higher resolution. These operations ensure that feature maps at different scales can be effectively fused in the final detection process. The downsampling operation is formulated as:

$$F_{\text{down}} = \text{Conv}(F_{\text{high}}, \text{hidc}, 3, 2), \tag{16}$$

where F_{high} is the high-resolution input feature map, F_{down} is the downsampled feature map, and the channel size is adjusted to "hidc" with a stride of 2. The upsampling operation is performed through interpolation:

$$F_{up} = \text{Upsample}(F_{low}), \tag{17}$$

where F_{low} is the low-resolution input feature map, and F_{up} is the upsampled feature map.

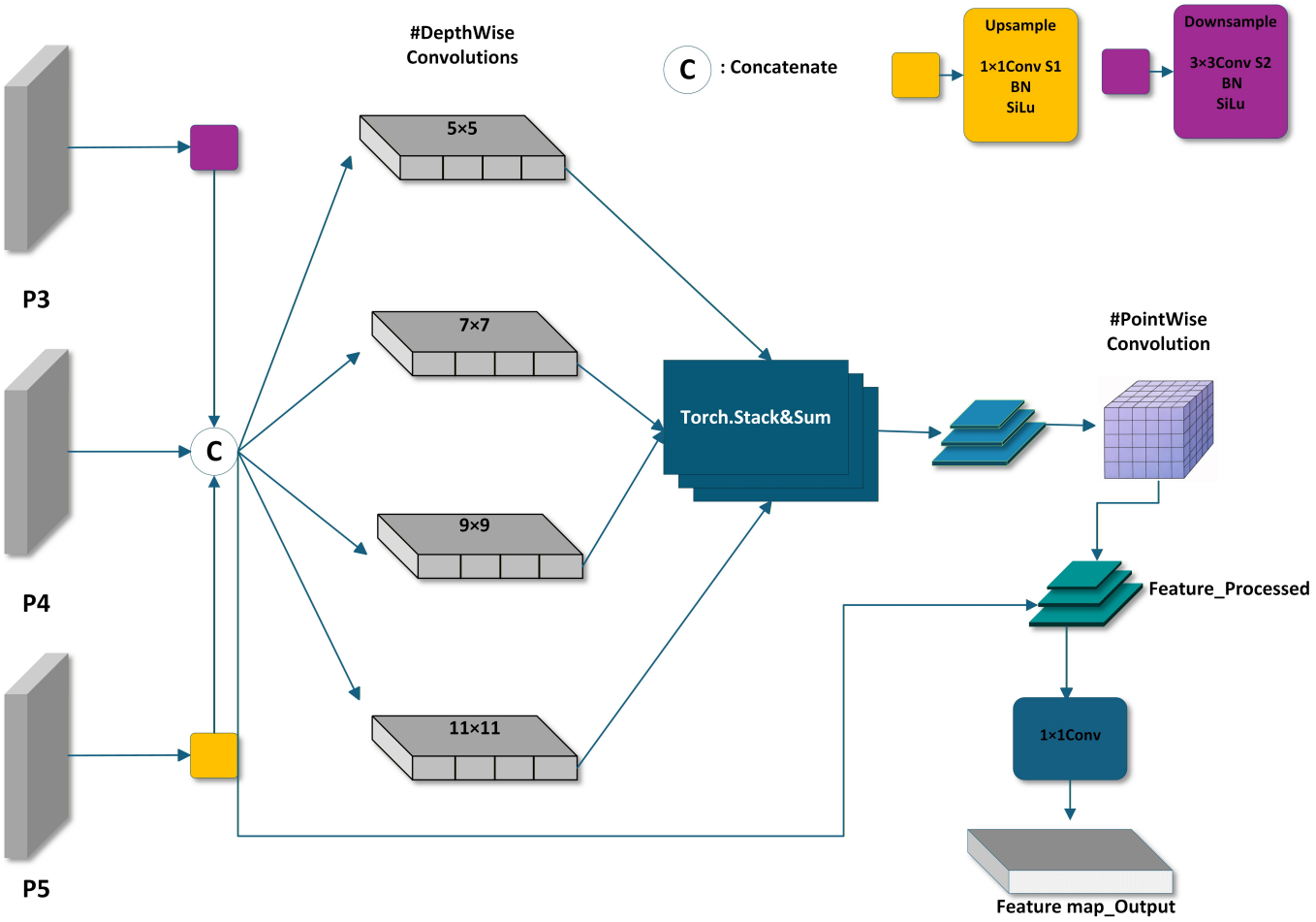


Figure 7. This is the architecture of SFIM. It adjusts channel dimensions of input features, applies depthwise and pointwise convolutions, and adds a residual connection to preserve original information, enabling effective fusion of low-level spatial details and high-level semantics.

At the conclusion of the MSFFN framework, all processed and fused feature maps are integrated to produce the final multi-scale object detection results. Specifically, feature maps from varying scales are subjected to upsampling, downsampling, and fusion operations, resulting in the output feature map. This process can be mathematically expressed as:

$$F_{output} = \text{Conv}_{1x1}\left(\sum_{i=1}^{N}\left(\text{Upsample}(F_{low}^{(i)}) + \text{Conv}(F_{high}^{(i)}, \text{hidc}, 3, 2)\right)\right), \tag{18}$$

where $F_{high}^{(i)}$ are feature maps from different scales, $\text{Upsample}(F_{low}^{(i)})$ is the upsampling operation applied to the low-resolution feature maps, and $\text{Conv}(F_{high}^{(i)}, \text{hidc}, 3, 2)$ is the downsampling operation applied to the high-resolution feature maps. F_{output} is the final output feature map, with the channel size adjusted by the 1×1 convolution.

In conclusion, the MSFFN framework, by integrating Transformer encoders and the SFIM, addresses the challenge of multi-scale feature fusion in remote sensing object detection. The SFIM, through depthwise convolution, pointwise convolution, and focused feature fusion strategies [55–57], effectively captures multi-scale object information. By employing downsampling and upsampling operations, MSFFN efficiently handles feature maps of varying resolutions and ultimately generates multi-scale object detection results. When compared to traditional methods, MSFFN demonstrates superior multi-scale handling capabilities in remote sensing images, leading to enhanced object detection accuracy and robustness.

3.4. Context-Sensitive Feature Interaction Module

In deep learning models, particularly in computer vision tasks such as object detection, feature fusion is a critical step [58]. The Context-Sensitive Feature Interaction Module (CSFIM) framework effectively addresses the challenges of feature misalignment and information loss by adaptively integrating feature maps from multiple network branches or different sources. CSFIM excels in handling heterogeneous features, especially in complex backgrounds. As shown in Figure 8, CSFIM enhances multi-scale feature fusion through the use of a dimensionality reduction attention mechanism [59] and a weighted feature recombination process [60]. This mechanism facilitates the adaptive modulation of contextual information, ensuring that features from different scales are accurately integrated. By improving the handling of complex contextual relationships, the approach results in enhanced robustness and performance in feature fusion tasks.

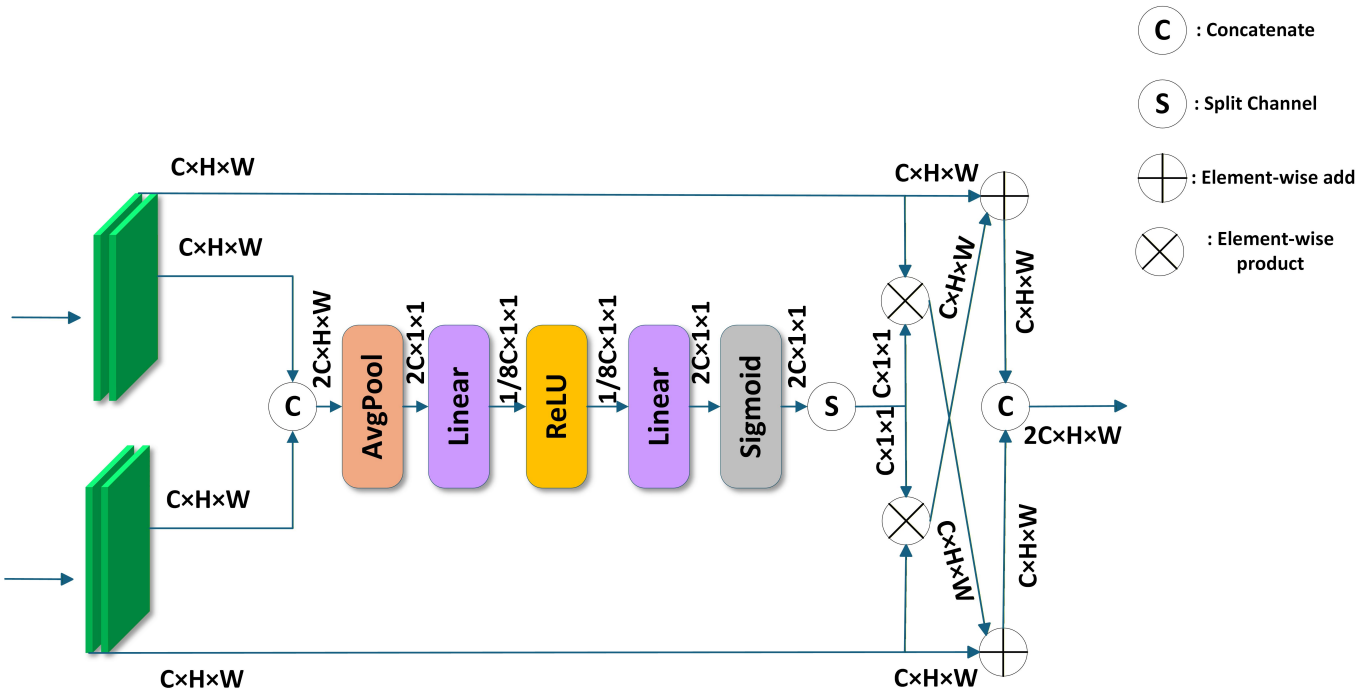


Figure 8. This is the architecture of CSFIM.

The first step of the CSFIM framework is the adaptation of input feature maps. Suppose the inputs are two feature maps F_1 and F_2 from different network branches. Since their channel dimensions might not match, CSFIM uses a convolution operation to adjust the channel sizes, making them have the same dimension. The specific operation is:

$$F_1' = \mathrm{Conv}(F_1, \mathrm{hidc}), \quad F_2' = \mathrm{Conv}(F_2, \mathrm{hidc}), \tag{19}$$

where F_1 and F_2 are the input feature maps, and "hidc" represents the adjusted channel size. After this adjustment, the two feature maps F_1' and F_2' can be fused in the subsequent steps.

If the input feature maps already have matching channel sizes, the CSFIM skips this operation and directly uses F_1 and F_2 as inputs.

The SEAttention [61] module is incorporated into CSFIM to perform feature recalibration, enhancing the key information associated with the target while suppressing irrelevant features. Through adaptive learning, the SEAttention module computes a weight for each channel and adjusts the feature map accordingly, emphasizing crucial information. For the input feature maps F_1' and F_2', they are first weighted by the SEAttention module, with the following calculation:

$$F_1' = F_1' \times \mathrm{SE}(F_1'), \quad F_2' = F_2' \times \mathrm{SE}(F_2'), \tag{20}$$

where $\mathrm{SE}(F)$ denotes the context-aware recalibration of the feature map F through the adaptive SE module. The SE module recalibrates the feature map by first calculating the global average pooling of the input feature map and then using a fully connected layer to generate weight coefficients for each channel. These weight coefficients are used to adjust the input feature map.

The fusion process is the core of CSFIM, where the module fuses feature maps from different sources with adaptive weights. Specifically, CSFIM performs weighted summation to fuse the adjusted feature maps F_1' and F_2'. The weighted feature for each location is computed as:

$$F_{\text{fusion}} = F_1' + F_2', \tag{21}$$

In this way, the module is able to reasonably fuse the information from different input feature maps.

CSFIM further optimizes multi-modal feature fusion through a feature-guided mechanism, enhancing the contextual information in the feature map. Specifically, in the fused feature map F_{fusion}, the module adaptively adjusts the weights $W(i,j)$ for each location, assigning different weights to different regions. The calculation is given by:

$$W(i,j) = 1 + \lambda_{\text{recall}} \cdot \hat{D}_{\text{gt}}(i,j) \cdot \mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) < \hat{D}_{\text{gt}}(i,j))} + \lambda_{\text{precision}} \cdot \hat{D}_{\text{gt}}(i,j) \cdot \mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) > \hat{D}_{\text{gt}}(i,j))}, \tag{22}$$

where $\hat{D}_{\text{gt}}(i,j)$ and $\hat{D}_{\text{pred}}(i,j)$ are the ground truth and predicted density map values at position (i,j), respectively. $\mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) < \hat{D}_{\text{gt}}(i,j))}$ and $\mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) > \hat{D}_{\text{gt}}(i,j))}$ are indicator functions that determine whether recall or precision penalties are applied based on the relationship between the predicted and ground truth density maps. The weight coefficients λ_{recall} and $\lambda_{\text{precision}}$ control the influence of recall and precision penalties, respectively.

The fused feature map, after adaptive weight calculation, is processed using a 1×1 convolution to produce the final output feature map. This process adjusts the feature map to ensure compatibility with downstream tasks. The final output feature map is represented as:

$$F_{\text{out}} = \mathrm{Conv}_{1\times1}(F_{\text{fusion}}), \tag{23}$$

where F_{fusion} is the fused feature map, and F_{out} is the final output feature map, with the channel size adjusted through the 1×1 convolution.

The CSFIM effectively addresses the challenges encountered in traditional feature fusion methods through the use of adaptive convolution layers, context-aware feature recalibration, and bidirectional feature enhancement. Unlike conventional methods, CSFIM is capable of adapting flexibly to various input configurations. By leveraging the collaborative fusion of contextual information, it enhances object detection accuracy and robustness. Additionally, through its flexible output mapping, CSFIM ensures compatibility

with downstream tasks, thereby improving both the model's computational efficiency and overall performance.

3.5. Adaptive Density Map Loss

In remote sensing image object detection tasks, significant progress has been made with traditional convolutional neural networks and Transformer models. However, challenges persist in handling multi-scale feature fusion, discrepancies in feature map sizes, and achieving high object detection accuracy. Specifically, traditional localization loss functions exhibit considerable limitations when dealing with small objects, complex backgrounds, and irregularly shaped targets [62]. For instance, L1 loss [63] often fails to provide sufficient positional information in scenarios involving small targets and dense backgrounds. This is because it merely computes the coordinate differences of bounding boxes, without accounting for the complexity of object shapes and backgrounds. Moreover, while Generalized IoU (GIoU) loss [64] optimizes the overlap between predicted and ground truth boxes, it struggles to accurately capture the true position of targets, especially when faced with complex shapes or substantial background interference. These limitations render traditional localization losses inadequate in certain scenarios, thereby diminishing overall detection accuracy [65].

To overcome these challenges, the proposed AdaptiveDM Loss framework integrates a multi-loss function design aimed at enhancing the performance of DETR-based object detection models, particularly for density-aware object detection tasks. The Density Map Loss serves as the core component of this framework, designed to optimize localization and enhance detection through the use of density maps. In contrast to traditional bounding box regression methods, the density map assigns a density value to each pixel, representing the object distribution. This approach allows the model to more precisely locate objects in dense regions. The model is trained to generate density maps that correlate with target region density, adapting to variations in object distributions across different areas.

First, following the normalization defined in Equation (2), the predicted density map $\hat{D}_{\text{pred}}$ and the ground truth density map $\hat{D}_{\text{gt}}$ are both normalized to [0, 1]. with a small constant $\epsilon = 10^{-6}$ to ensure numerical stability.

After normalization, the difference ΔD between the predicted density map and the ground truth density map is computed:

$$\Delta D = \hat{D}_{\text{pred}} - \hat{D}_{\text{gt}}, \tag{24}$$

Next, dynamic adjustment of the weight for each location is performed based on the difference between the predicted and ground truth density maps. The weight $W(i,j)$ is adjusted based on the relationship between the predicted and ground truth values and is computed as follows:

$$W(i,j) = 1 + \lambda_{\text{recall}} \cdot \hat{D}_{\text{gt}}(i,j) \cdot \mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) < \hat{D}_{\text{gt}}(i,j))} + \lambda_{\text{precision}} \cdot \hat{D}_{\text{gt}}(i,j) \cdot \mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) > \hat{D}_{\text{gt}}(i,j))}, \tag{25}$$

where $\mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) < \hat{D}_{\text{gt}}(i,j))}$ and $\mathbb{1}_{(\hat{D}_{\text{pred}}(i,j) > \hat{D}_{\text{gt}}(i,j))}$ are indicator functions that determine whether recall or precision penalties should be applied based on the relationship between the predicted and ground truth density maps. The weight coefficients λ_{recall} and $\lambda_{\text{precision}}$ control the influence of recall and precision penalties, respectively.

The final loss function is based on the weighted error computed for each pixel. The density loss L_{density} is the weighted mean squared error (MSE) and is computed globally as:

$$L_{\text{density}} = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} W(i,j) \cdot (\Delta D(i,j))^2, \tag{26}$$

where H and W are the height and width of the density map.

The final density map loss function, incorporating both recall and precision penalties, is given by:

$$L_{\text{density}} = \mathbb{E}(i,j)\left[(1 + \lambda_{\text{recall}} \cdot \hat{D}_{\text{gt}}(i,j) \cdot 1_{\text{under}} + \lambda_{\text{precision}} \cdot \hat{D}_{\text{gt}}(i,j) \cdot 1_{\text{over}}) \cdot (\hat{D}_{\text{pred}} - \hat{D}_{\text{gt}})^2\right], \quad (27)$$

where 1_{under} and 1_{over} are indicator functions for underestimation and overestimation, respectively.

In this loss calculation, several key hyperparameters have a significant impact on the model's stability and convergence:

- λ_{recall} and $\lambda_{\text{precision}}$: These hyperparameters control the strength of the recall and precision penalties. A larger recall penalty value encourages the model to focus more on underestimated areas, while a larger precision penalty value penalizes excessively high density predictions. These parameters should be adjusted according to the characteristics of the dataset.
- ϵ: The stability term in normalization to avoid division by zero. This value is small but critical for numerical stability.
- Loss weight L_{density}: The weighted mean squared error loss is the core loss, and the choice of hyperparameters affects the model's focus on different regions.

The role of AdaptiveDM Loss is to optimize the training process of object detection models by incorporating density map generation and multi-scale feature fusion, particularly in scenarios characterized by uneven object distribution or dense regions. Detection accuracy and computational efficiency are enhanced by focusing the model's attention on dense regions and dynamically adjusting the allocation of computational resources. Additionally, AdaptiveDM Loss contributes to the model's robustness in complex scenes by facilitating denoising training, enabling the model to maintain high accuracy even in the presence of noise or partial target occlusion. This, in turn, results in an overall improvement in object detection performance.

4. Experiments

4.1. Datasets

In this study, three publicly available drone-based datasets were utilized: VisDrone2019 [66], DOTA-V1.5 [67] and UAVDT [68]. VisDrone2019 served as the primary benchmark dataset, while DOTA-V1.5 and UAVDT were employed to assess the model's generalization ability across different datasets, thus providing a comprehensive validation of the model's performance and adaptability. These three datasets offer complementary scene characteristics, aiding in the evaluation of the model's object detection capabilities in diverse and challenging environments [69].

The VisDrone2019 dataset was released by the Machine Learning and Data Mining Laboratory at Tianjin University. It consists of high-resolution images captured from a variety of complex environments, covering 10 object categories such as pedestrians, vehicles, and bicycles. The dataset includes 10,290 images in total, all with complete object annotations. Of these, 6471 images were allocated for training, 548 for validation, and 1610 for testing. These images are sourced from urban streets, traffic scenes, and various complex backgrounds, offering a wide range of imaging conditions (e.g., varying lighting, weather, camera altitudes, and viewing angles) and multi-scale object distributions. Notably, despite its large training set (6471 images), the VisDrone2019 dataset remains highly challenging for conventional object detectors due to its intrinsic properties: extreme scale variations (from tiny pedestrians to large vehicles), high object density in crowded scenes (e.g., intersections and markets), and complex background clutter (buildings, trees, shadows). This diversity

presents significant challenges for object detection tasks, including occlusion, overlap, and complex background interference.

On the other hand, the DOTA-V1.5 dataset includes 18 object categories, such as airplanes, ships, roads, and buildings, covering a vast array of geographical areas. Primarily composed of satellite and aerial remote sensing imagery (i.e., acquired from significantly higher altitudes than typical drone-captured images), it is designed for object detection in remote sensing scenarios. The dataset contains 1411 images for training, 458 for validation, and 937 for testing. We included DOTA-V1.5 in our experiments to evaluate the cross-domain generalization capability of DPF-DETR across different imaging altitudes and platforms. Due to the high-altitude acquisition and large geographical coverage, objects in this dataset exhibit a significantly broader scale spectrum than those in typical drone imagery, ranging from tiny vehicles (a few pixels) to massive infrastructure (hundreds of pixels). Consequently, this dataset poses greater challenges with respect to object scale, background complexity, and occlusion between objects. UAVDT is also a dataset specifically constructed for the object detection task from the perspective of drones. It contains dynamic images in various complex scenes such as cities, villages, and transportation hubs. It contains a total of 40,409 images of three categories: cars, trucks, and buses, with 23,829 for training and 16,580 for testing.

The infographic for the VisDrone2019 dataset is shown in Figure 9. The first image provides a dataset statistics summary, outlining essential details such as the total number of images, categories, annotations, and the top three categories by instance count. The second image presents a bar chart illustrating the top 10 categories by instance count, with 'car' being the most prevalent. The third image displays a heatmap showing the spatial distribution of object centers across the dataset, highlighting areas with higher object concentration. The fourth image visualizes the distribution of object areas, revealing that the majority of objects are relatively small. The fifth image presents the aspect ratio distribution of objects, where the majority exhibit narrow aspect ratios. The sixth image shows a scatter plot of object width versus height, indicating a positive correlation between the two attributes. The seventh image is a pie chart categorizing objects into small, medium, and large sizes, with small objects constituting the largest proportion. The eighth image is a boxplot that illustrates the distribution of instances across categories, showing the mean and median values for reference.

To ensure fairness and consistency in the experiments, all images from the VisDrone2019, DOTA-V1.5 and UAVDT datasets were uniformly rescaled to 640 × 640 pixels, which was used as the input size for the model. Since the DOTA-V1.5 dataset does not provide ground truth annotations for the test set, all comparative experiments conducted on this dataset were performed using the validation set.

4.2. Experimental Environment and Parameter Configuration

All experiments in this study were conducted in a unified deep learning environment with the following configuration: the operating system was Ubuntu 22.04, the GPU used was the RTX 4090 (24 GB memory), the CPU was a 16-core Intel® Xeon® Gold 6430, the deep learning framework was PyTorch 2.3.0, and the CUDA version was 12.1. During model training, the input image size was set to 640 × 640, with a batch size of 8 and a total of 160 training epochs. The initial learning rate was set to 0.0008, and the learning rate for the Backbone was 0.0004. The optimizer used was AdamW, with a momentum value of 0.9 and a weight decay coefficient of 1×10^{-4}. For the SOTA methods used in the comparative experiments, all hyperparameters were strictly set according to the configurations reported in their original papers.

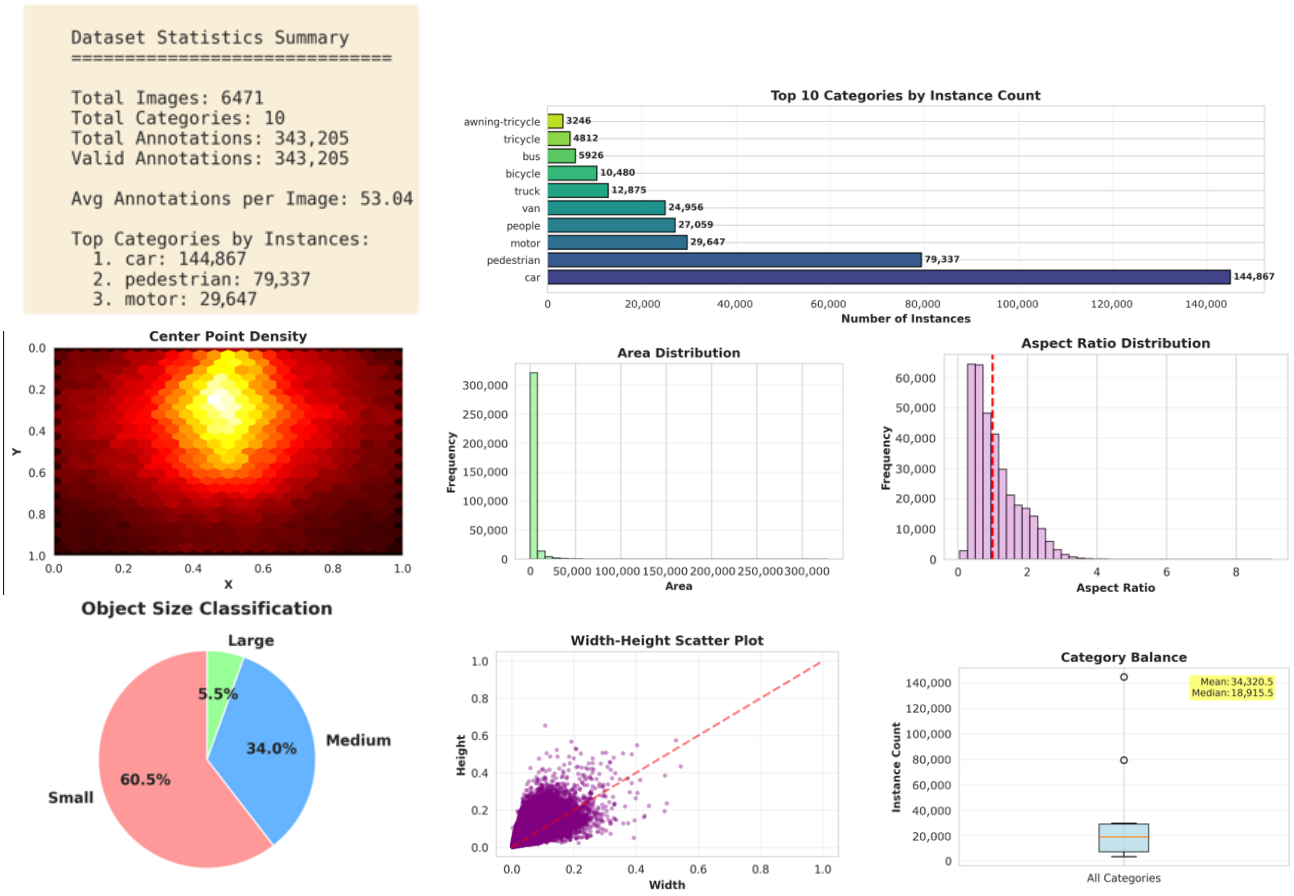


Figure 9. Information statistics of the Visdrone2019 dataset.

4.3. Evaluation Metrics

To comprehensively evaluate the performance of the proposed model, several core evaluation metrics are utilized to assess its effectiveness and efficiency. The primary performance metrics include the mean average precision (mAP) at various intersection over union (IoU) thresholds, with particular emphasis placed on mAP_{50} and $mAP_{50–95}$. In addition, model efficiency is assessed through the total number of model parameters and the floating-point operations per second (GFLOPs). These metrics provide a robust foundation for thoroughly understanding the model's detection performance and computational efficiency.

mAP is an important metric derived from precision (P), recall (R), and average precision (AP), and is calculated as follows:

Precision (P) measures the ratio of true positives (TP) to all predicted positives (TP + FP):

$$P = \frac{TP}{TP + FP}, \tag{28}$$

Recall (R) measures the ratio of true positives (TP) to the total number of actual positives (TP + FN):

$$R = \frac{TP}{TP + FN}, \tag{29}$$

Average Precision (AP) is computed by integrating the precision across different recall levels, as follows:

$$AP = \int_0^1 P(r)\, dr, \tag{30}$$

where $P(r)$ is the precision at a given recall rate r.

To further enhance the performance evaluation, the average precision for different object sizes is also considered: AP_s (average precision for small objects, defined as objects with area $< 32 \times 32$ pixels^2), AP_m (average precision for medium objects, area between 32×32 and 96×96 pixels^2), and AP_l (average precision for large objects, area $\geq 96 \times 96$ pixels^2). These categories are defined based on specific object areas within pixel ranges, following the annotation guidelines of the MS COCO dataset.

There are two variants of mAP computation:

- mAP_{50} computes the average precision at an IoU threshold of 0.5, providing a general evaluation of the model's target detection.
- $mAP_{50–95}$ computes the average precision across IoU thresholds ranging from 0.5 to 0.95 (with a step size of 0.05), providing a more comprehensive evaluation of localization accuracy. The mathematical formulas are as follows:

$$mAP_{50} = \frac{1}{N} \sum_{n=1}^{N} AP_n(\text{IoU} = 0.5), \tag{31}$$

and

$$mAP_{50–95} = \frac{1}{10N} \sum_{n=1}^{N} \sum_{i=0.5}^{0.95} AP_n(\text{IoU} = i), \tag{32}$$

where N is the total number of object categories, and $AP_n(\text{IoU})$ represents the average precision of category n at the given IoU threshold.

4.4. Best Configuration of DSM

To optimize the configuration of DSM, an ablation study was conducted on the VisDrone2019 validation set, with a focus on analyzing the effect of different query factor values on the model's performance. The DPF-DETR model was used as the baseline, and all subsequent ablation experiments were carried out based on the complete DPF-DETR model. The experimental environment and hyperparameter settings were kept consistent with those described in Section 4.2, and the evaluation metrics were measured at a resolution of 640×640. The primary evaluation metrics included mAP_{50}, $mAP_{50–95}$, as well as average precision for different object sizes: small-object average precision (mAP_s), medium-object average precision (mAP_m), and large-object average precision (mAP_l). Since the query factor does not influence the model's parameters or computational load, no comparison was made in terms of these aspects. The experimental results are presented in Table 1.

Five different query factor values were compared in the experiments. The results indicate that the model performs optimally when the query factor is set to 5. It is believed that the advantage of the third group lies in the moderate query factor value, resulting in an optimal number of bounding boxes, which allows for effective target capture while minimizing interference from excessive background noise. In contrast, the query factor setting in the first and second group was insufficient to provide adequate receptive field expansion for distant small targets in high-altitude scenarios, resulting in a 3.0% and 1.9% decrease in mAP_{50} compared to the third group, and a 1.3% and 0.7% decrease in average precision for small targets. The fourth and fifth group introduced excessive redundant context, causing a 1.5% and 2.0% decrease in mAP_{50} compared to the third group, and a decline in the average precision for targets across different resolutions. To determine whether the observed performance differences are statistically significant, we conducted paired t-tests across five independent runs for each query factor setting. The results show that the configuration with a query factor of 5 achieves statistically significant improvements over the other configurations ($p < 0.05$) in terms of mAP_{50} and mAP_s. For instance, the im-

provement over query factor 3 is significant with $p = 0.018$, and that over query factor 7 is significant with $p = 0.024$. These results confirm that the advantage of the query factor of 5 is not merely numerical but statistically reliable.

Table 1. Optimal parameter ablation results of the DSM.

Query Factor	$mAP_{50–95}$	mAP_{50}	mAP_s	mAP_m	mAP_l
3	0.213	0.359	0.102	0.343	0.487
4	0.217	0.370	0.108	0.350	0.496
5	**0.222**	**0.389**	**0.115**	**0.365**	**0.527**
6	0.220	0.374	0.101	0.351	0.509
7	0.215	0.369	0.101	0.349	0.505

The best result is highlighted in red bold.

4.5. Best Configuration for SFIM

A systematic ablation study was conducted on the Selective Feature Integration Module using the VisDrone2019 validation set to determine its optimal configuration. The evaluation metrics included mAP_{50}, mAP_s, model parameter count, and floating-point operations per second (GFLOPs). The experimental results are presented in Table 2.

In the experiments, four sets of equally spaced parallel deep convolution kernel configurations were designed and compared. The results indicated that the third configuration (i.e., convolution kernels of [5, 7, 9, 11]) delivered the best performance. It is believed that the moderate starting value and range of the third configuration enabled it to effectively capture cross-scale spatial patterns and semantic features across various receptive fields. This configuration not only captures detailed target features but also integrates global information effectively over a large range, ensuring the comprehensive use of multi-scale information.

In contrast, the first and second configurations, when processing distant targets or complex backgrounds, employed smaller convolution kernels that were insufficient for capturing adequate global information. In particular, their performance in extracting large-scale background information was weaker, making it difficult to effectively suppress background noise. Consequently, compared to the third configuration ($[5, 7, 9, 11]$), the first configuration ($[1, 3, 5, 7]$) showed a mean decrease of 1.9% in mAP_{50}, and the second ($[3, 5, 7, 9]$) showed a mean decrease of 1.3%. To assess the statistical significance of these differences, we performed five independent runs for each configuration and conducted paired t-tests. The improvement of the third configuration over the first is statistically significant ($p = 0.008$), and that over the second is also significant ($p = 0.031$).

The fourth configuration ($[7, 9, 11, 13]$), although utilizing larger convolution kernels, resulted in wasted computational resources in scenes with small targets or high-resolution details. Additionally, it was less precise in extracting small-scale information compared to the moderate-sized kernels, leading to a mean decrease of 2.3% in small-target detection accuracy (mAP_s) relative to the third configuration. This difference is also statistically significant ($p = 0.019$). These results confirm that the moderate kernel set $[5, 7, 9, 11]$ provides the best trade-off and statistically reliable performance gains.

Table 2. Optimal parameter ablation results of the SFIM.

Convolution Rate	mAP_{50}	mAP_s	Params	GFLOPs
(1, 3, 5, 7)	0.370	0.095	**4.50 M**	**9.60 G**
(3, 5, 7, 9)	0.376	0.102	4.55 M	9.76 G
(5, 7, 9, 11)	**0.389**	**0.115**	4.68 M	9.94 G
(7, 9, 11, 13)	0.368	0.092	4.86 M	10.02 G

The best result is highlighted in red bold.

4.6. Best Configuration for CSFIM

To validate the advantages of the Context-Sensitive Feature Interaction Module (CSFIM) fusion method over concatenation (Concat) and addition (Add) operations, and to analyze the roles of the individual components within the CSFIM, comparative experiments were conducted on the VisDrone2019 validation set. The experimental design included five groups: Group A (using the concatenation operation from the original model), Group B (replacing all concatenation operations with addition), Group C (CSFIM with only the attention mechanism), Group D (CSFIM with only the bi-directional feature enhancement strategy), and Group E (using the complete CSFIM). The experimental results are shown in Table 3.

The results indicate that the CSFIM increased mAP_{50} to 38.9%, surpassing the concatenation operation by 0.7%. This improvement is attributed to the CSFIM's capacity to efficiently integrate heterogeneous feature representations across scales. In contrast, Group B's addition operation, which overlooked the importance of feature weighting, resulted in a 3.7% decrease in mAP_{50}. Despite the lightweight nature of the addition operation, it did not effectively enhance feature fusion quality, ultimately sacrificing accuracy. Group A's concatenation operation, lacking an effective feature selection mechanism, performed worse than CSFIM, further confirming the importance of targeted feature fusion in enhancing detection performance in dense small-object scenes.

Further analysis of the module components revealed that both Group C (only the attention mechanism) and Group D (only the bi-directional feature enhancement strategy) yielded inferior results compared to the complete CSFIM. However, a potential concern is that the superior performance of the complete CSFIM might simply arise from increased parameterization (overparameterization) rather than architectural innovation. To rule out this possibility, we designed an additional control experiment: a parameter-matched version of Group C (denoted Group C+), where we added extra convolutional layers to Group C to match the total number of parameters of the complete CSFIM (Group E), while keeping the attention-only design unchanged. As shown in Table 3, Group C+ achieves only a marginal improvement ($mAP_{50} = 38.2\%$) over Group C (38.0%), which is not statistically significant ($p = 0.34$). In contrast, the complete CSFIM (Group E) achieves a substantially higher mAP_{50} (38.9%) with the same parameter budget, and this improvement over Group C+ is statistically significant ($p = 0.009$). These control results confirm that the performance gain of CSFIM is attributable to its architectural design, specifically the synergy between attention-based recalibration and bidirectional feature interaction rather than to overparameterization.

Statistical analysis of the results in Table 3 reveals that Groups A (Concat), C (attention only), C+ (parameter-matched attention), and D (bidirectional only) yield mAP_{50} values that are not significantly different from each other ($p > 0.05$). Only the complete CSFIM (Group E) achieves a statistically significant improvement over Groups A, C, and C+ ($p < 0.05$). The improvement over Group D is positive but only marginally significant ($p = 0.06$), suggesting that while the combination of attention and bidirectional enhancement provides

additional benefit, the effect size is modest. Nevertheless, the complete CSFIM consistently outperforms all other fusion strategies across multiple runs, and its advantage over simple concatenation or addition is statistically robust. These findings support the conclusion that the synergy between attention-based recalibration and bidirectional feature interaction contributes positively to detection performance, though further optimization may be needed to amplify the gain. The integration of these components allows for the effective exploitation of relationships between features at different scales, thereby enhancing the overall performance of the model.

Table 3. Optimal parameter ablation results of the SFIM.

Fusion Mode	mAP_{50}	mAP_s	Params (M)
A	0.382	0.107	4.62
B	0.352	0.086	**4.60**
C	0.380	0.111	4.62
C+	0.382	0.112	4.68
D	0.385	0.113	4.66
E	**0.389**	**0.115**	4.68

The second row is highlighted in red bold.

4.7. Ablation Study

To further validate the role and importance of each component in the DPF-DETR model, a comprehensive ablation study was conducted on the VisDrone2019 test set. The DSM, MSFFN, CSFIM, and AdaptiveDM Loss were incrementally added to the baseline model, DPF-DETR, and the performance of different model combinations was compared. The experimental setup was consistent with the descriptions provided in Section 4.2. The experimental results are presented in Table 4.

Table 4. Ablation experiments based on the D-Fine baseline on the VisDrone2019 test set.

DSM	MSFFN	CSFIM	AdaptiveDM Loss	mAP_{50}	$mAP_{50–95}$	Params (M)	GFLOPs (G)
-	-	-	-	0.334	0.183	**3.73**	7.12
✓	-	-	-	0.343	0.190	3.97	**7.07**
-	✓	-	-	0.344	0.192	4.41	9.97
-	-	✓	-	0.337	0.185	3.76	7.14
-	-	-	✓	0.338	0.186	**3.73**	7.12
✓	✓	-	-	0.352	0.201	4.65	9.91
✓	✓	✓	-	0.355	0.204	4.68	9.94
✓	✓	✓	✓	**0.359**	**0.207**	4.68	9.94

The best result is highlighted in red bold.

The ablation study results confirm the effectiveness of the proposed modules. The complete DPF-DETR model led to a 2.5% improvement in mAP_{50} and a 2.4% improvement in $mAP_{50–95}$, while the model parameters increased by only 25.4% and the computational load by 39.5%.

Notably, the DSM module, while maintaining a complexity comparable to the original DPF-DETR decoder, resulted in a minor increase in parameters from 3.73 M to 3.97 M upon replacing the TransformerDecoder in DPF-DETR. The FLOPs decreased slightly from 7.12 G to 7.07 G, yet mAP_{50} increased by 0.9%. This enhancement is attributed to the Density-aware Query Selection method, which introduces the density map module to explicitly model the spatial distribution of targets, providing effective spatial prior information, particularly for detecting dense targets. This method utilizes multi-scale parallel deep

convolutions to capture target features at different scales, extracting fine-grained density information while aggregating contextual information, thus improving performance in both sparse and dense regions. Max-pooling operations suppress noise and enhance robustness, while bilinear upsampling ensures pixel-level alignment of the density map with the input image, optimizing target localization. Additionally, the normalized density map serves as spatial attention weights and confidence priors, further improving detection accuracy and localization.

In the MSFFN experiment, the redesigned feature pyramid combined with the selective feature integration module significantly enhanced model performance. MSFFN increased mAP_{50} to 35.2% (a 0.9% improvement). The introduction of the Selective Feature Integration Module (SFIM) and the feature diffusion mechanism greatly improved multi-scale feature fusion capabilities, optimizing the information integration process. The network effectively captures features at various scales through a combination of convolution kernels, retaining local details while incorporating global semantic information, thereby improving the model's adaptability to complex scenes.

Moreover, the inclusion of the CSFIM (on top of MSFFN) raised the mean mAP_{50} from 35.2% to 35.5%. However, a paired t-test over five independent runs indicates that this 0.3% absolute increase is not statistically significant ($p = 0.18$). Therefore, the isolated contribution of CSFIM alone should be interpreted with caution. Nevertheless, as shown in our detailed module analysis (Section 4.6, Table 3), the complete CSFIM achieves statistically significant improvements over simpler fusion strategies (e.g., Concat or Attention-only) when controlling for parameter count. Moreover, in the full DPF-DETR model (all modules combined), CSFIM contributes to a cumulative gain, and its benefits are more pronounced in small-target detection (mAP_s) and in dense scenes, where the bidirectional feature interaction helps resolve occlusions and scale variations. This improvement stems from CSFIM's effective combination of low-level spatial features and high-level semantic cues, significantly enhancing the multi-scale feature fusion process, particularly in small target detection tasks in complex environments. In this module, low-level spatial features are seamlessly combined with high-level semantic information (such as refining target boundaries) through the attention mechanism, while high-level features retrieve essential spatial information from low-level details. This bidirectional enhancement strategy mitigates semantic conflicts in multi-scale fusion and optimizes the feature integration process, substantially improving small target detection accuracy in cluttered backgrounds (e.g., dense crowds). Overall, CSFIM strengthens the contextual correlation between features, enhancing the detection accuracy of small targets in challenging scenarios.

Additionally, the incorporation of AdaptiveDM Loss (on top of the other three modules) resulted in mean improvements of 0.4% in mAP_{50} and 0.3% in $mAP_{50–95}$. However, paired t-tests over five independent runs indicate that these improvements are not statistically significant ($p = 0.21$ for mAP_{50}, $p = 0.34$ for $mAP_{50–95}$).

The primary purpose of AdaptiveDM Loss is to improve the quality of density map predictions, which are essential for the Density Sensing Mechanism (DSM). Without AdaptiveDM Loss, the Mean Absolute Error (MAE) of the predicted density map is 0.087; with AdaptiveDM Loss, it decreases to 0.071, indicating a meaningful enhancement in density estimation. Furthermore, when evaluating on dense scenes (images containing more than 50 objects), the gain in mAP_{50} reaches 0.8% ($p = 0.04$), confirming that the loss function is particularly beneficial in high-density scenarios. These results suggest that AdaptiveDM Loss fulfills its intended role of refining density map supervision, and its benefits are best realized in conjunction with DSM and in dense target environments. This improvement is attributed to the AdaptiveDM Loss's ability to adjust weights adaptively, addressing overestimation and underestimation errors. It introduces precision- and recall-based weighting,

applying distinct weights to overestimated and underestimated regions, thereby balancing the contribution of various error types to the loss. The loss function also emphasizes regions with larger errors in the density map through weighted squared errors, enhancing model performance in critical areas and ultimately yielding more accurate density distribution predictions, especially in dense target detection tasks.

The ablation study results indicate that all four modules must function collaboratively to achieve optimal performance, effectively validating the importance of the density-aware module, feature focus and diffusion mechanism, context-sensitive feature interaction module, and AdaptiveDM Loss in remote sensing object detection.

4.8. Comparison Experiment

To comprehensively evaluate the performance of DPF-DETR across varying model complexities and between specialized and general models, experiments were conducted on the VisDrone2019 test set (Table 5) and the DOTA-V1.5 validation set (Table 6). In these experiments, DPF-DETR was compared with multiple benchmark models and state-of-the-art (SOTA) models, using metrics such as mean average precision (mAP), parameter count, and floating-point operations per second (GFLOPs). One-stage models rely on anchor-based mechanisms to generate predictions and depend on post-processing, whereas end-to-end models directly output predictions through set prediction and bipartite graph matching, eliminating the need for post-processing. These two model types differ fundamentally in their architectures. To maintain clarity, the "Input shape" column in the tables specifies the input size settings, with all models retaining the original input sizes reported in their respective papers.

As presented in Table 5, DPF-DETR achieves the highest $mAP_{50–95}$ and mAP_{50} on the VisDrone2019 test set. In terms of small-object detection accuracy, it performs slightly worse than the state-of-the-art end-to-end model RT-DETR-R18, but its parameter count is only 23.4% of that model's. Despite having more parameters than our baseline model D-Fine-N, DPF-DETR outperforms the baseline in both computational efficiency and accuracy. With a parameter count of 4.68 M and a computational load of 9.94 G FLOPs, DPF-DETR is significantly more efficient than most comparison models, achieving a balance between accuracy and efficiency while maintaining a low computational cost.

Table 5. On the VisDrone 2019 test set, DPF-DETR is compared with the benchmark and SOTA models. For each metric, the best result is highlighted in bold red, and the second-best result is highlighted in bold blue.

Model	Input	mAP_{50}	$mAP_{50–95}$	mAP_s	mAP_m	mAP_l	Params	GFLOPs
One-Stage Object Detector								
TOOD-R50 [70]	(768, 1344)	0.339	0.204	0.102	**0.317**	0.403	32.04 M	199 G
RetinaNet-R50 [71]	(768, 1344)	0.276	0.164	0.060	0.274	0.427	36.52 M	210 G
GFL [72]	(768, 1344)	0.321	0.193	0.094	0.300	0.409	32.28 M	206 G
ATSS-R50 [73]	(768, 1344)	0.338	0.204	0.100	**0.317**	**0.485**	38.91 M	110 G
RTMDet-Tiny [74]	(640, 640)	0.312	0.184	0.077	0.288	0.445	4.88 M	**8.03 G**
YOLOV8-M [24]	(640, 640)	0.332	0.190	0.090	0.294	0.417	25.85 M	78.7 G
YOLOV10-M [5]	(640, 640)	0.345	0.195	0.097	0.300	0.414	15.32 M	58.9 G
YOLOV11-M [6]	(640, 640)	0.350	0.203	0.098	**0.312**	0.413	20.04 M	67.7 G
YOLOV12-M [7]	(640, 640)	0.336	0.192	0.094	0.298	0.386	20.11 M	67.2 G
YOLOV26-M [75]	(640, 640)	0.332	0.186	0.096	0.281	0.361	20.36 M	67.9 G
FBRT-YOLO-M [76]	(640, 640)	0.344	0.196	0.094	0.309	0.421	7.36 M	58.7 G
Two-Stage Object Detector								
Faster-RCNN-R50 [3]	(768, 1344)	0.329	0.194	0.095	0.309	0.429	41.39 M	208 G
Cascade-RCNN-R50 [77]	(768, 1344)	0.326	0.197	0.099	0.309	0.406	69.29 M	236 G

Table 5. *Cont.*

Model	Input	mAP_{50}	$mAP_{50–95}$	mAP_s	mAP_m	mAP_l	Params	GFLOPs
End-to-end Object Detector								
DETR [12]	(750, 1333)	0.333	0.184	0.081	0.287	0.446	41.57 M	96.5 G
Deformable-DETR [44]	(750, 1333)	0.343	0.194	0.095	0.296	0.457	40.10 M	193 G
Conditional-DETR-R50 [78]	(750, 1333)	0.338	0.189	0.086	0.291	0.452	43.45 M	101 G
RT-DETR-R18 [27]	(640, 640)	0.333	0.185	**0.113**	0.275	0.423	20 M	60 G
D-Fine-N [30] (Baseline)	(640, 640)	0.334	0.183	0.093	0.270	0.442	**3.73 M**	**7.12 G**
DEIM-D-Fine-N [43]	(640, 640)	0.322	0.177	0.090	0.262	0.376	**3.73 M**	**7.12 G**
DEIMV2-S [45]	(640, 640)	0.357	0.204	0.107	0.299	0.452	9.67 M	25.39 G
Ours								
DPF-DETR	(640, 640)	**0.359**	**0.207**	**0.108**	0.309	**0.462**	**4.68 M**	9.9 G

Furthermore, a systematic performance evaluation was conducted on nine models, including DPF-DETR, using the VisDrone2019 test set, with the results visualized in Figure 10. Each data point in the figure represents a model, where the x-axis denotes the model's parameter count and the y-axis shows the detection accuracy at mAP_{50}. The data points are color-coded to distinguish between one-stage, two-stage, and end-to-end detectors. The DPF-DETR model, represented by the red circle, exhibits an exceptional balance between model size and detection performance, achieving the highest mAP_{50} while maintaining a relatively low parameter count of 4.68 M. This highlights its capability to optimize both accuracy and computational efficiency when compared to other models.

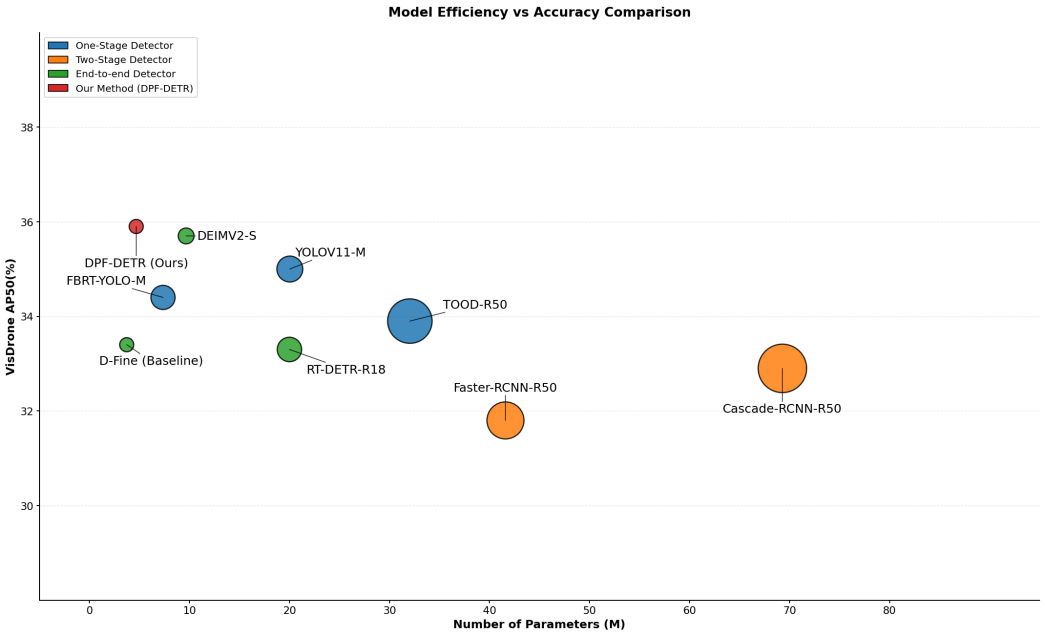


Figure 10. Comparison chart of different models on the VisDrone2019 test set.

The experimental results on the DOTA-V1.5 validation set further demonstrate the effectiveness of DPF-DETR. Table 6 presents supplementary comparison models selected based on similar complexity levels (parameters and FLOPs) to DPF-DETR, ensuring a fair accuracy comparison. DPF-DETR achieves a mAP50 of 69.7%, which is the highest among all compared models with comparable or larger parameter budgets. Specifically, it outperforms the baseline D-Fine-N by 4.7% in mAP50 (69.7% vs. 65.0%) while using only 0.95M additional parameters (4.68M vs. 3.73M) and a modest increase in computational cost (9.94 GFLOPs vs. 7.12 GFLOPs). In terms of mAP50–95, DPF-DETR attains 43.8%, which is slightly lower than HSF-DETR (44.7%) but surpasses all other methods. Considering its

substantially lower parameter count and FLOPs compared to HSF-DETR, DPF-DETR offers a favorable trade-off between accuracy and efficiency. These results confirm that DPF-DETR generalizes well to aerial remote sensing images with different object distributions and scale characteristics.

Table 6. On the DOTA-v1.5 validation dataset, DPF-DETR is compared with the benchmark and SOTA models. For each metric, the best result is highlighted in bold red, and the second-best result is highlighted in bold blue.

Model	mAP_{50}	$mAP_{50–95}$
D-Fine-N [30] (Baseline)	0.650	0.412
RT-DETR-R18 [27]	0.646	0.409
HSF-DETR [17]	0.693	0.447
Faster R-CNN [3]	0.665	0.399
YOLOV8-S [24]	0.657	0.402
YOLOV10-M [5]	0.661	0.421
YOLOV11-M [6]	0.667	0.424
YOLOV12-M [7]	0.671	0.428
AG-YOLO [79]	**0.728**	**0.466**
	Ours	
DPF-DETR	**0.729**	**0.458**

To further evaluate the generalization capability of DPF-DETR across different drone-based scenarios, we additionally conduct experiments on the UAVDT dataset, which features dynamic urban and suburban scenes with vehicle-centric categories. As reported in Table 7, DPF-DETR consistently outperforms all compared methods, achieving the highest mAP_{50} (40.8%) and $mAP_{50–95}$ (26.1%). Notably, it surpasses the recent VRF-DETR by 0.5% in mAP_{50} and 0.2% in $mAP_{50–95}$, while also exceeding the baseline D-Fine-N by 2.3% and 2.0%, respectively. These results confirm that the proposed density-aware query selection and multi-scale feature fusion modules are not over-fitted to a single dataset but generalize well to other aerial benchmarks with different object distributions and scene complexities.

Table 7. On the UAVDT test dataset, DPF-DETR is compared with the benchmark and SOTA models. For each metric, the best result is highlighted in bold red, and the second-best result is highlighted in bold blue.

Model	mAP_{50}	$mAP_{50–95}$
D-Fine-N [30] (Baseline)	0.385	0.241
RT-DETR-R18 [27]	0.381	0.233
Deformable-DETR [44]	0.284	0.171
DETR [12]	0.365	0.221
Faster R-CNN [3]	0.307	0.169
Cascade R-CNN [77]	0.305	0.171
YOLOV8-M [24]	0.310	0.188
YOLOV5-M [25]	0.298	0.172
FBRT-YOLO-M [76]	0.311	0.184
VRF-DETR	**0.403**	**0.259**
	Ours	
DPF-DETR	**0.408**	**0.261**

Furthermore, in the real-time performance evaluation, the inference efficiency of the model is influenced by various factors, such as hardware configuration, running environment, batch size, and code optimization. To ensure fairness and reproducibility in the comparative experiments, tests were conducted on the same hardware platform (configuration described in Section 4.2) using the same dataset (VisDrone2019 test set)

under simulated resource-limited conditions (with the batch size set to 1). The baseline model D-Fine-N and RT-DETR-R18 were compared with DPF-DETR, all using an input image size of 640 × 640. In these experiments, inference time and frames per second (FPS) were utilized as evaluation metrics to assess the model's real-time response capability during continuous inference tasks. The results are presented in Table 8.

Table 8. Compare the real-time performance of DPF-DETR, RT-DETR, and D-Fine on the 640 × 640 VisDrone2019 test set.

Batch Size	Model	Inference Time (ms)	FPS
1	RT-DETR-R18	1.12	889.75
	D-Fine-N	1.08	924.63
	DPF-DETR (Ours)	1.84	543.48

As presented in Table 8, DPF-DETR demonstrated exceptional real-time performance on the VisDrone2019 test set. The model achieved an inference time of 1.84 ms and a frames per second (FPS) rate of 543.48. Although these results were slightly lower than those of RT-DETR-R18 and D-Fine-N, DPF-DETR surpassed the critical 60 FPS threshold, which is essential for human perception, thus fulfilling the real-time requirements for most practical scenarios. This performance was accomplished with a low computational load of 9.94 G FLOPs, indicating that both inference speed and detection accuracy were effectively optimized. Consequently, DPF-DETR exhibits significant potential for deployment in real-time object detection applications.

4.9. Visual Analysis

To facilitate a visual comparison of the detection performance between DPF-DETR and the baseline model DFine-N, several images from the VisDrone2019 test set were selected, and inference was performed using both models. These images, exhibiting varying heights, angles, lighting conditions, and occlusions, offer a comprehensive demonstration of model performance across different scenarios. Figure 11 displays the detection results of both DFine-N and DPF-DETR, with a comparison to the original annotations from the dataset.

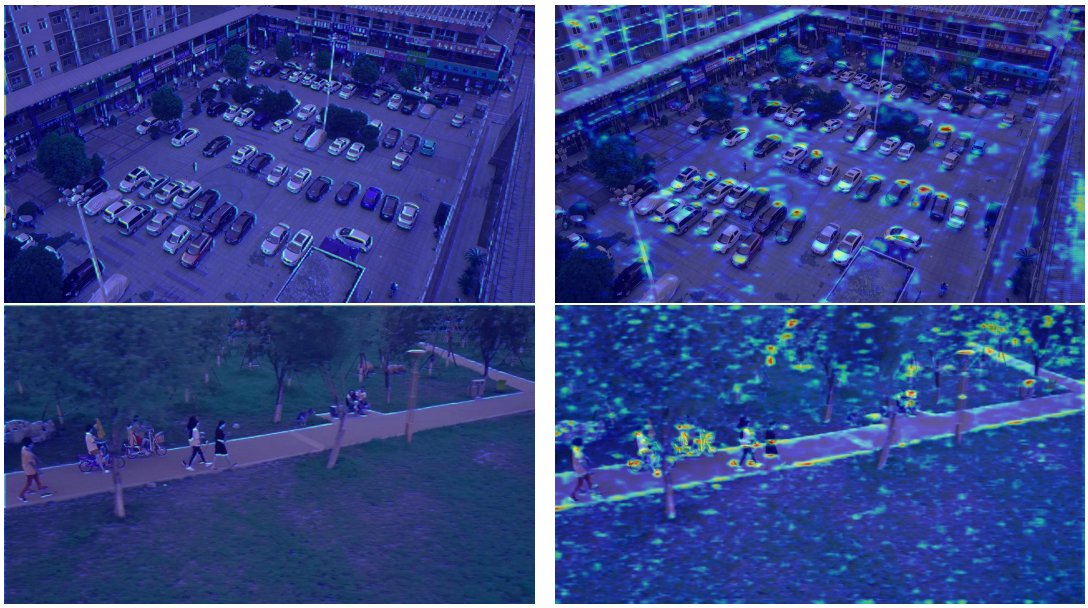
The experimental results demonstrate that, in addressing the challenges posed by large target scale variations and complex feature extraction in drone-captured images, the proposed DPF-DETR model outperforms the baseline model D-Fine-N by clear margins. As reported in Table 5, DPF-DETR achieves a 2.5% higher mAP_{50}, a 2.4% higher $mAP_{50–95}$, and a 1.5% higher mAP_s on the VisDrone2019 test set. These improvements are statistically significant based on paired t-tests over five independent runs ($p < 0.05$ for all three metrics). This quantitative evidence indicates that DPF-DETR possesses stronger feature capturing and multi-scale representation capabilities. To visually illustrate how the model allocates attention during the detection process, dense small target images from the VisDrone2019 dataset were selected, and the attention heatmaps of both DPF-DETR and DFine-N were compared. Figure 12 presents the attention distributions of the two models.

The experimental results indicate that DPF-DETR significantly enhances its attention to dense small target regions, particularly in dense crowd scenarios. It can be observed that the attention distribution of DPF-DETR is more refined, and the object boundaries are clearer, especially in deeper feature layers (e.g., P5). This effect is particularly pronounced, suggesting that the proposed module effectively improves the model's ability to retain more local information. This validates that the hybrid architecture employed better balances the integration of detailed information with high-level semantic features.



(**a**) Original annotations in the dataset (**b**) The detection results of D-Fine (**c**) The detection results of DPF-DETR

Figure 11. Visualization of detection results on the VisDrone dataset. Each row shows a different perspective and height.



(**a**) The Heatmap of D-Fine (**b**) The Heatmap of DPF-DETR

Figure 12. Comparison of visual heat maps in scenarios with confusing objects.

5. Discussion

In the preceding section, the performance and strengths of the DPF-DETR model were critically analyzed in comparison to existing methods for UAV-based object detection. The experimental results obtained from the VisDrone2019, DOTA-V1.5 and UAVDT datasets

validate the effectiveness of DPF-DETR in addressing the challenges associated with high-density targets, complex backgrounds, and large variations in object scale.

A key contribution of DPF-DETR lies in the innovative implementation of the Density-Aware Query Selection (DSM), which dynamically adjusts the number of queries based on the spatial density of objects within the scene. This adaptive mechanism improves the model's focus on dense regions while reducing unnecessary computational costs in sparser areas. This approach not only enhances detection accuracy, particularly for small targets, but also ensures the efficient utilization of computational resources.

The introduction of the Adaptive Density Map Loss (AdaptiveDM Loss) further augments the model's performance by directing attention to regions with high errors in the density map, thereby improving localization accuracy and object detection precision in dense and complex scenarios. This capability proves essential for handling common challenges in aerial imagery, such as occlusions and overlapping objects.

In addition, the Multi-Scale Feature Fusion Network (MSFFN) and the Context-Sensitive Feature Interaction Module (CSFIM) have significantly improved the model's ability to integrate features from different scales. MSFFN enhances multi-scale feature extraction by ensuring optimal fusion of both low-level spatial details and high-level semantic information, thereby increasing the model's capacity to detect objects of various sizes. CSFIM further refines this process by guiding the fusion with contextual information, thereby reducing semantic discrepancies between features of different scales, which is critical for dense target detection.

The current experiments were conducted on raw perspective images, which provide neither orthorectified results nor pose information. In principle, orthorectification reduces perspective distortion and homogenizes ground sampling distance, which would decrease intra-image scale variation and potentially make the density map learning more regular. Therefore, we hypothesize that DPF-DETR could maintain or even slightly improve its performance on high-quality orthophotos, though this requires dedicated verification. Moreover, aerotriangulation provides accurate exterior orientation parameters. These pose data could be incorporated as additional positional encodings into the Transformer decoder, enabling the model to exploit geometric constraints for better handling of occlusions and severe perspective distortions. While these extensions are beyond the scope of this work, they represent promising future directions to further enhance the robustness of density-aware detection in geometrically complex UAV scenarios.

While DPF-DETR demonstrates outstanding performance, the analysis identifies areas for potential enhancement. Specifically, further improvements in the model's handling of small targets in highly occluded or cluttered environments are needed. Incorporating spatiotemporal dynamics could also enhance robustness, particularly in scenarios where objects are in motion, such as in real-time UAV applications. Additionally, optimizations in query selection and computational efficiency could facilitate the deployment of DPF-DETR in resource-constrained environments, such as edge devices.

While this study focuses on UAV imagery using optical data, the integration of LiDAR and SLAM could further enhance the model's capability, especially for applications requiring detailed 3D spatial information. Future work will explore the potential of incorporating LiDAR data and SLAM techniques to improve detection accuracy and robustness in more complex environments.

6. Conclusions

This study introduces DPF-DETR, a novel approach to object detection in UAV imagery, designed to tackle significant challenges such as large-scale variations, complex backgrounds, and dense targets. Through the innovative integration of Density-Aware

Query Selection (DSM), Adaptive Density Map Loss (AdaptiveDM Loss), Multi-Scale Feature Fusion Network (MSFFN), and Context-Sensitive Feature Interaction Module (CSFIM), DPF-DETR exhibits exceptional performance in both detection accuracy and computational efficiency.

The primary strength of DPF-DETR lies in its ability to dynamically adjust the number of queries based on the spatial density of objects within the scene. This adaptive mechanism not only improves the model's ability to handle dense and small objects but also ensures efficient allocation of computational resources, making the model suitable for real-time applications. By incorporating multi-scale feature fusion, DPF-DETR captures fine-grained details while also accounting for broader contextual information, thereby enhancing robustness in complex environments.

Furthermore, the inclusion of AdaptiveDM Loss strengthens the model's localization capabilities by prioritizing regions where predictions are less accurate, resulting in improved overall detection performance. This is particularly advantageous in scenarios characterized by complex background clutter and object occlusions, which are common in aerial imagery.

In this study, the proposed method is specifically designed and applied to UAV imagery due to the unique challenges posed by drone-captured images. UAVs typically operate at lower altitudes, providing high-resolution imagery with significant variations in target scale and complex backgrounds, which are particularly difficult to handle in object detection tasks. Additionally, UAV imagery often involves dense distributions of objects in dynamic environments, which requires specialized handling and adaptive strategies, such as our Density Sensing Mechanism (DSM) and Multi-Scale Feature Fusion Network (MSFFN).

While traditional airplane-derived and ground-based imagery present their own set of challenges, such as larger flight altitudes and different spatial resolutions, the current approach primarily addresses the specific needs of UAV-based detection. However, the methods developed here are not limited to UAV imagery alone; they are adaptable and can be extended to other remote sensing data sources in the future. Further research will explore how our proposed strategies perform on airplane-derived and ground-based imagery, with the goal of broadening their applicability across various remote sensing applications.

In conclusion, DPF-DETR achieves quantifiable improvements in UAV-based object detection. On the VisDrone2019 test set, it outperforms the baseline D-Fine-N by 2.5% in mAP_{50} and by 2.4% in $mAP_{50–95}$, while using only 4.68 M parameters. On DOTA-V1.5, it attains 69.7% mAP_{50}, the highest among compared models with similar or larger parameter budgets. On the UAVDT dataset, DPF-DETR achieves 40.8% mAP_{50} and 26.1% $mAP_{50–95}$, surpassing the baseline D-Fine-N by 2.3% and 2.0%, respectively. These consistent results across three challenging drone-based datasets demonstrate a favorable balance between detection accuracy and computational efficiency. Future research will focus on refining the model's performance in more challenging environments and enabling deployment on platforms with limited computational resources. With its robust feature integration and dynamic query mechanisms, DPF-DETR shows promise for real-world UAV applications.

Author Contributions: Conceptualization, S.L.; methodology, S.L., Y.W., S.Z.; software, S.L.; validation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, Z.L., X.W., Y.W., S,Z.; visualization, S.L.; supervision, Z.L.; funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Joint Funds of the National Natural Science Foundation of China, (Ye Qisun Science Foundation) under Grant U2341223.

Data Availability Statement: The datasets used in this study are public datasets and can be downloaded from the official website of the corresponding datasets.

Acknowledgments: We sincerely thank the editor and anonymous reviewers for their valuable feedback and constructive suggestions, which have significantly enhanced the quality of this manuscript.

Conflicts of Interest: All the authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

1. Khelifi, L.; Mignotte, M. Deep Learning for Change Detection in Remote Sensing Images: Comprehensive Review and Meta-Analysis. *IEEE Access* **2020**, *8*, 126385–126400. [CrossRef]
2. Zhang, X.; Zhang, T.; Wang, G.; Zhu, P.; Tang, X.; Jia, X.; Jiao, L. Remote Sensing Object Detection Meets Deep Learning: A Metareview of Challenges and Advances. *IEEE Geosci. Remote Sens. Mag.* **2023**, *11*, 8–44. [CrossRef]
3. 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]
4. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430. [CrossRef]
5. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
6. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725. [CrossRef]
7. Tian, Y.; Ye, Q.; Doermann, D. YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv* **2025**, arXiv:2502.12524.
8. Zhang, H.; Liu, K.; Gan, Z.; Zhu, G.-N. UAV-DETR: Efficient End-to-End Object Detection for Unmanned Aerial Vehicle Imagery. *arXiv* **2024**, arXiv:2501.01855.
9. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo Algorithm Developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
10. Zhang, Y.; Lu, T. RecFRCN: Few-Shot Object Detection with Recalibrated Faster R-CNN. *IEEE Access* **2023**, *11*, 121109–121117. [CrossRef]
11. Long, Y.; Di, L.; Fu, Y.; Zhao, S.; Bai, X. An Improved RT-DETR-Based Object Detection Algorithm for UAV Aerial Photography. *Int. J. Adv. Netw. Monit. Control. (IJANMC)* **2025**, *21*, 59–69. [CrossRef]
12. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision*; Springer International Publishing: Cham, Switzerland, 2020.
13. Li, C.; Yang, T.; Zhu, S.; Chen, C.; Guan, S. Density Map Guided Object Detection in Aerial Images. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 737–746.
14. Zhang, S.; Wu, Z.; Zhao, H.; Liu, Z.; Chen, L.; Luo, Y.; Zhang, L.; Cui, Z.; Ke, Z.; Du, B. DSA-SRGS: Super-Resolution Gaussian Splatting for Dynamic Sparse-View DSA Reconstruction. *arXiv* **2026**, arXiv:2603.04770.
15. Duan, C.; Wei, Z.; Zhang, C.; Qu, S.; Wang, H. Coarse-Grained Density Map Guided Object Detection in Aerial Images. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 2789–2798.
16. Gao, T.; Li, Z.; Wen, Y.; Chen, T.; Niu, Q.; Liu, Z. Attention-Free Global Multiscale Fusion Network for Remote Sensing Object Detection. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–14. [CrossRef]
17. Mao, Y.; Zhang, H.; Li, R.; Zhu, F.; Sun, R.; Ji, P. HSF-DETR: Hyper Scale Fusion Detection Transformer for Multi-Perspective UAV Object Detection. *Remote Sens.* **2025**, *17*, 1997. [CrossRef]
18. Cheng, G.; Si, Y.; Hong, H.; Yao, X.; Guo, L. Cross-Scale Feature Fusion for Object Detection in Optical Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 431–435. [CrossRef]
19. Li, B.; Xiao, C.; Wang, L.; Wang, Y.; Lin, Z.; Li, M.; An, W.; Guo, Y. Dense Nested Attention Network for Infrared Small Target Detection. *IEEE Trans. Image Process.* **2023**, *32*, 1745–1758. [CrossRef]
20. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2003.
21. Hua, W.; Chen, Q. A survey of small object detection based on deep learning in aerial images. *Artif. Intell. Rev.* **2025**, *58*, 162. [CrossRef]
22. 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 Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
23. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
24. Jocher, G.; Chaurasia, A.; Qiu, J. YOLOv8 by Ultralytics. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 8 March 2026).

25. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Wong, C.; Montes, D.; Wang, Z.; et al. YOLOv5 by Ultralytics. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 8 March 2026).
26. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE/CVF, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
27. 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.
28. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. *arXiv* **2022**, arXiv:2203.03605.
29. Drone-YOLO: An Efficient Neural Network Method for Target Detection in Drone Images. *Drones* **2023**, *7*, 526. [CrossRef]
30. Peng, Y.; Li, H.; Wu, P.; Zhang, Y.; Sun, X.; Wu, F. D-FINE: Redefine Regression Task in DETRs as Fine-Grained Distribution Refinement. *arXiv* **2024**, arXiv:2410.13842.
31. Huang, Y.X.; Liu, H.I.; Shuai, H.H.; Cheng, W.H. DQ-DETR: DETR with Dynamic Query for Tiny Object Detection. *arXiv* **2024**, arXiv:2404.03507. [CrossRef]
32. Gao, R. Rethinking Dilated Convolution for Real-Time Semantic Segmentation. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 17–24 June 2023; pp. 4675–4684.
33. Wu, H.; Huang, P.; Zhang, M.; Tang, W.; Yu, X. CMTFNet: CNN and Multiscale Transformer Fusion Network for Remote-Sensing Image Semantic Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–12. [CrossRef]
34. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
35. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
36. Liu, W.; Shi, L.; An, G. An Efficient Aerial Image Detection with Variable Receptive Fields. *Remote Sens.* **2025**, *17*, 2672. [CrossRef]
37. Liu, C.; Gao, G.; Huang, Z.; Hu, Z.; Liu, Q.; Wang, Y. YOLC: You Only Look Clusters for Tiny Object Detection in Aerial Images. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 13863–13875. [CrossRef]
38. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
39. Du, S.; Liang, X.; Wu, K.; Tian, Y.; Liu, Y.; Jian, L. Multiple Scales Fusion and Query Matching Stabilization for Detection with Transformer. *Eng. Appl. Artif. Intell.* **2025**, *144*, 110047. [CrossRef]
40. Wang, J.; Li, Z.; Luo, D.; Zhou, Y.; Xue, Q. Fast Federated Learning Via Imprecise Correction in Unreliable Wireless Networks. *IEEE Trans. Mob. Comput.* **2026**, *17*, 10.1109.
41. Zhao, C.; Zhang, Y.; Chen, Y. Density-based Object Detection in Crowded Scenes. *arXiv* **2025**, arXiv:2504.09819. [CrossRef]
42. Chen, C.; Zeng, W.; Zhang, X.; Zhou, Y.; Yu, J.; Chang, Y.; Wang, F. RSDS: A Specialized Loss Calculation Method for Dense Small Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–17. [CrossRef]
43. DEIM: DETR with Improved Matching for Fast Convergence. Available online: http://ieeexplore-ieee-org-s.vpn.bistu.edu.cn:8118/document/11092982 (accessed on 8 March 2026).
44. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv* **2021**, arXiv:2010.04159.
45. Huang, S.; Hou, Y.; Liu, L.; Yu, X.; Shen, X. Real-Time Object Detection Meets DINOv3. *arXiv* **2025**, arXiv:2509.20787. [CrossRef]
46. Zhang, J.; Zhang, X.; Tang, X.; Chen, P.; Jiao, L. Sketch-Based Region Adaptive Sparse Unmixing Applied to Hyperspectral Image. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8840–8856. [CrossRef]
47. Hadji, I.; Wildes, R.P. What Do We Understand About Convolutional Networks? *arXiv* **2018**, arXiv:1803.08834.
48. Shahadat, N.; Maida, A.S. Analyzing Parameter-Efficient Convolutional Neural Network Architectures for Visual Classification. *Sensors* **2025**, *25*, 7663. [CrossRef]
49. Khan, S.H.; Iqbal, R. A Comprehensive Survey on Architectural Advances in Deep CNNs: Challenges, Applications, and Emerging Research Directions. *arXiv* **2025**, arXiv:2503.16546. [CrossRef]
50. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 1314–1324.
51. Bai, C.; Bai, X.; Wu, K.; Ye, Y. Adaptive condition-aware high-dimensional decoupling remote sensing image object detection algorithm. *Sci. Rep.* **2024**, *14*, 20090. [CrossRef]
52. Wang, C.Y.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616. [CrossRef]

53. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
54. Ioannou, Y.; Thai, C.; Trang, M.X.; Anh, S.L. Training CNNs with Frozen Pointwise Convolutions. *J. Mach. Learn. Res.* **2025**, *26*, 1–35.
55. Liu, S.; Huang, D.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1911.09516. [CrossRef]
56. Chen, Y.; Zhang, C.; Chen, B.; Huang, Y.; Sun, Y.; Wang, C.; Fu, X.; Dai, Y.; Qin, F.; Peng, Y.; et al. Accurate Leukocyte Detection Based on Deformable-DETR and Multi-Level Feature Fusion for Aiding Diagnosis of Blood Diseases. *Comput. Biol. Med.* **2024**, *170*, 107917. [CrossRef] [PubMed]
57. Deng, L.; Yang, M.; Li, T.; He, Y.; Wang, C. RFBNet: Deep Multimodal Networks with Residual Fusion Blocks for RGB-D Semantic Segmentation. *arXiv* **2019**, arXiv:1907.00135. [CrossRef]
58. Yang, Z.; Sun, Z. Review of Multimodal Medical Image Fusion. *Chin. J. Med Phys.* **2026**, *2026*, 14–27. [CrossRef]
59. Wang, J.; Ge, X.; Shu, W.; He, Z.; Qiu, X. Attention Layers Add Into Low-Dimensional Residual Subspaces. *arXiv* **2025**, arXiv:2508.16929. [CrossRef]
60. 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), Munich, Germany, 8–14 September 2018; pp. 3–19.
61. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
62. Chien, T.; Trang, M.X.; Le, A. Towards robust visual recognition for smart cities and remote sensing: A survey of regression losses in rotated object detection (综述旋转目标检测中回归损失函数的核心挑战，包括角度周期性、边缘模糊性和度量不一致性). *CTU J. Innov. Sustain. Dev.* **2025**, *17*, 64–74. [CrossRef]
63. Janocha, K.; Czarnecki, W.M. On Loss Functions for Deep Neural Networks in Classification. *arXiv* **2017**, arXiv:1702.05659. [CrossRef]
64. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
65. Liu, Y.; Wang, F.; Chen, W.; Zhang, L. A Survey of Loss Functions in Deep Learning (重新分类损失函数为回归损失、分类损失和度量损失三类，提出新的分类体系). *Mathematics* **2025**, *13*, 2417. [CrossRef]
66. 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]
67. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983. [CrossRef]
68. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 370–386.
69. Qi, S.; Song, X.; Shang, T.; Hu, X.; Han, K. MSFE-YOLO: An Improved YOLOv8 Network for Object Detection on Drone View. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 1–5. [CrossRef]
70. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-Aligned One-Stage Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 3510–3519.
71. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
72. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; Volume 33, pp. 21002–21012.
73. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9759–9768.
74. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. *arXiv* **2022**, arXiv:2212.07784. [CrossRef]
75. Sapkota, R.; Cheppally, R.H.; Sharda, A.; Karkee, M. YOLO26: Key Architectural Enhancements and Performance Benchmarking for Real-Time Object Detection (首次系统介绍YOLOv26的架构创新，包括DFL移除、无NMS推理、ProgLoss、STAL和MuSGD优化器). *arXiv* **2025**, arXiv:2509.25164.
76. Xiao, Y.; Xu, T.; Xin, Y.; Li, J. FBRT-YOLO: Faster and Better for Real-Time Aerial Image Detection. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 8673–8681. [CrossRef]

77. 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, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
78. Meng, D.; Chen, X.; Fan, Z.; Zeng, G.; Li, H.; Yuan, Y.; Sun, L.; Wang, J. Conditional DETR for Fast Training Convergence. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 3651–3660.
79. Wang, X.; Han, C.; Huang, L.; Nie, T.; Liu, X.; Liu, H.; Li, M. AG-Yolo: Attention-Guided Yolo for Efficient Remote Sensing Oriented Object Detection. *Remote Sens.* **2025**, *17*, 1027. [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.