DART-DETR: Adaptive Receptive Fields and Dynamic Feature Fusion for Efficient Real-Time Aerial Object Detection

Chenxi Nan
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: 1364734102@qq.com

Zhongsheng Wang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: wzhsh1681@163.com

***Abstract*—Object detection in Unmanned Aerial Vehicle (UAV) scenarios faces significant challenges, including low resolution, a high prevalence of small objects, extreme scale variations, and dense occlusion. To address these issues, this paper proposes DART-DETR, a novel object detection architecture based on adaptive receptive fields and dynamic feature fusion. Specifically, this work introduces the DAFPN, which achieves content-adaptive fusion across multi-scale features via dual-path dynamic weight prediction, a cross-input redistribution mechanism, and lightweight context residual modeling. Furthermore, this model designs the ARMix module, which integrates learnable multi-kernel depthwise convolution with CGLU to enhance spatial-channel modeling capabilities. Extensive experiments on the VisDrone 2019 benchmark demonstrate that our DAFPN-RT-DETR achieves 50.0% mAP50 and 30.9% mAP50-95 with only 15.4M parameters and 59.7 GFLOPs. This work establishes a new state-of-the-art trade-off between efficiency and accuracy, providing an effective and generalizable solution for UAV-based small object detection.**

Keywords-Object Detection; Unmanned Aerial Vehicle; Small Object Detection; Transformer-based Detector

I. INTRODUCTION

Unmanned Aerial Vehicle (UAV) aerial imagery, characterized by its broad bird's-eye view, has demonstrated significant potential in applications such as disaster emergency response, precision agriculture monitoring, and smart city management. However, object detection in aerial scenarios presents severe challenges, including extreme object scale variations, mutual occlusion caused by dense arrangements, and complex ground background interference.

Existing real-time object detection solutions are primarily dominated by CNN-based and Transformer-based architectures. Two-stage detectors, represented by Faster R-CNN [1], achieve high-precision localization via the RPN. However, when handling dense small objects, the computational cost of RPN generation and NMS increases exponentially. Furthermore, the computational redundancy between the two stages makes it difficult to meet real-time requirements. Subsequent one-stage detectors improved inference speed by predicting bounding boxes and categories in a single forward pass. However, CNN-based architectures are limited by local receptive fields, resulting in weak global context modeling capabilities in complex backgrounds.

DETR introduced Hungarian matching to eliminate the constraints of anchors and NMS. Although variants like Deformable-DETR [2] and DINO [3] alleviated issues of slow convergence and poor small object detection via sparse attention mechanisms, their high computational burden remains unsuitable for edge deployment. RT-DETR [4], as the first real-time end-to-end detector, designed a hybrid encoder to decouple multi-scale feature processing, achieving an excellent speed-accuracy trade-off. However, directly applying RT-DETR to UAV scenarios faces limitations: the static nature of feature fusion lacks adaptive weight assignment mechanisms for varying image content. Additionally, the encoder suffers from limited perception capabilities, failing to adaptively capture multi-scale local patterns, and still exhibits significant parameter redundancy on edge devices.

In this paper, our work proposes DART-DETR,

an end-to-end object detector specifically designed for UAV edge computing scenarios. This paper utilizes the ARMix module to reconstruct the backbone network, innovatively constructing a parallel multi-scale perception topology. It employs efficient depthwise convolutions to build multi-path parallel branches, dynamically aggregating spatial context information from different receptive fields via a learnable soft-weighting mechanism. To address the bottleneck of cross-scale information flow, the model proposes the DAFPN architecture, which introduces a bidirectional topology allowing for full interaction between high-level semantics and low-level details across multiple iterations.

Experimental results demonstrate that DART-DETR achieves a breakthrough in both accuracy and efficiency. On the VisDrone2019 benchmark, compared to RT-DETR-R18, our model achieves a 2.71% increase in mAP50 and a 1.9% increase in mAP50-95, while reducing the number of parameters by 22.4%.

The main contributions of this paper are summarized as follows:

- This paper proposes the ARMix module, which synergistically integrates multi-scale re-parameterized convolution with gating mechanisms. This design significantly enhances the model's ability to extract local details while substantially reducing computational complexity.
- This study designs the DAFPN architecture, which optimizes cross-scale information fusion through dynamic weight redistribution and bidirectional feature flow, effectively addressing the challenge of extreme scale variation in aerial imagery.
- DART-DETR is the first lightweight real-time detector to combine re-parameterization concepts with dynamic feature fusion, establishing a new speed-accuracy benchmark for UAV aerial scenarios.

II. Related Work

Compared to conventional ground-view detection tasks, aerial imagery typically contains a vast number of extremely small objects with minimal pixel coverage. These objects possess limited discriminative features, such as textures and edges, rendering local features extracted by traditional convolutions difficult to distinguish. Furthermore, the dense distribution of objects in aerial scenarios often leads to small targets being overwhelmed by background noise or the textures of adjacent objects. Additionally, UAV platforms are generally constrained by onboard computational resources, necessitating models that balance inference speed, accuracy, and stability within a limited computational budget.

The first category comprises CNN-based models. Girshick et al. pioneered R-CNN [5], which significantly improved detection accuracy by feeding candidate regions into a CNN classifier. However, its multi-stage pipeline incurs substantial computational costs. Subsequently, Fast R-CNN improved inference speed by sharing feature maps to reduce redundant computation, yet it still relied on Selective Search for region proposal, preventing true end-to-end training. To address this, Ren et al. proposed Faster R-CNN, introducing the RPN to replace traditional generation methods, thereby fully integrating detection into a unified framework. While this model excelled in accuracy, localization stability, and training controllability, outperforming contemporaneous one-stage detectors for a considerable period, its reliance on high-resolution features limits its utility in real-time applications. In contrast, the YOLO series, proposed by Redmon et al., reframed detection as an end-to-end regression problem. Early versions like YOLOv1 [6] to YOLOv3 [7] utilized anchor-based mechanisms to maximize speed, although the predefined anchors introduced additional hyperparameters and dense regression computational burdens. Subsequent versions, such as YOLOv5 [8], YOLOX [9], and YOLOv8 [10], progressively shifted toward anchor-free architectures, predicting object locations via center points or keypoints to achieve a superior speed-accuracy trade-off. Nevertheless, these real-time

detectors typically generate numerous overlapping candidate boxes, necessitating NMS for post-processing. In the dense small object scenarios typical of UAV imagery, NMS introduces extra computational overhead and risks suppressing adjacent small targets, thereby reducing overall inference speed and recall.

The second category consists of end-to-end models. Following the advancement of Transformers in the computer vision domain, Carion et al. proposed DETR, the first Transformer-based end-to-end object detector. By introducing the Hungarian matching algorithm, this method achieves one-to-one assignment between predicted and ground-truth boxes, effectively eliminating the need for anchor design and NMS post-processing. However, DETR suffers from slow training convergence and extremely high computational costs. Consequently, Zhu et al. proposed Deformable DETR, which replaces the global dense attention of the original DETR with sparse deformable attention, enabling the model to adaptively attend to key regions across multiple sampling points. Nevertheless, it still requires a lengthy training phase to achieve optimal performance. Building on this, Chen et al. proposed Group DETR [11], which achieves a hierarchical attention structure characterized by "dense intra-group and sparse inter-group" connections through the grouping of feature channels. While this method reduces computational complexity and enhances feature representation capabilities, its inference speed remains constrained in real-time scenarios due to the complexity of its structural design. Furthermore, Liu et al. proposed DAB-DETR [12], which introduces learnable dynamic anchor boxes into DETR. This design allows each query to carry spatial positional information from the early stages of training, guiding the model to focus more on the object center and shape information. However, the training process still requires a relatively long duration.

Li et al. proposed DN-DETR [13], introducing the concept of "denoising" into DETR for the first time. By simultaneously feeding noisy labels and ground truth labels during training, it stabilizes training targets and enhances model robustness against perturbations. However, the construction of denoising samples requires additional design, increasing the complexity of the overall framework. Building on this, Zhang et al. proposed DINO, which synergistically combines denoising training, Hybrid Query Selection, and a "look-forward-twice" strategy. This approach successfully achieves faster convergence and superior performance in small object detection; however, its attention mechanisms and decoder structure still impose a heavy computational burden. Targeting real-time scenarios, Pan et al. proposed RT-DETR, achieving significant breakthroughs in both speed and accuracy by constructing an efficient hybrid encoder and uncertainty-minimal query selection. RT-DETR possesses the advantage of flexible speed tuning to adapt to different task requirements. Nevertheless, its encoder remains dependent on a fixed-scale attention structure, lacking the ability to adaptively prioritize computational resources to the most discriminative local regions based on image content. Consequently, there remains room for improvement in complex backgrounds or extremely small object detection tasks.

DART-DETR performs system-level optimization focused on reducing computational costs and improving feature fusion strategies. By introducing an adaptive scale-perception mechanism and a lightweight feature enhancement structure, the model attains stronger small object perception capabilities and higher comprehensive accuracy while maintaining real-time performance.

III. DART-DETR ARCHITECTURE

The proposed DART-DETR architecture comprises the MS-ARMixNet for hierarchical feature learning and the DAFPN multi-scale adaptive fusion head. The overall structure of the improved DART-DETR is illustrated in Figure 1.

Specifically, the model feeds the multi-scale features {S3,S4,S5} from the last three stages of the backbone into an efficient hybrid encoder. This encoder transforms these features into a sequence of image features through intra-scale feature interaction and cross-scale feature fusion. Subsequently, based on the uncertainty-minimal

query selection strategy, a fixed number of features are selected from the encoder output to serve as initial object queries for the decoder. Finally, the decoder with auxiliary prediction heads iteratively optimizes these object queries to generate class categories and bounding boxes.

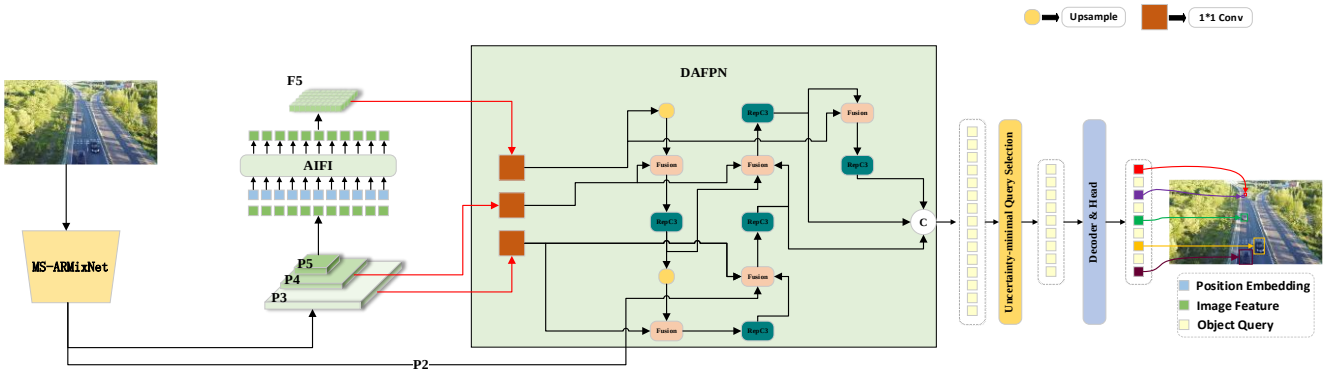
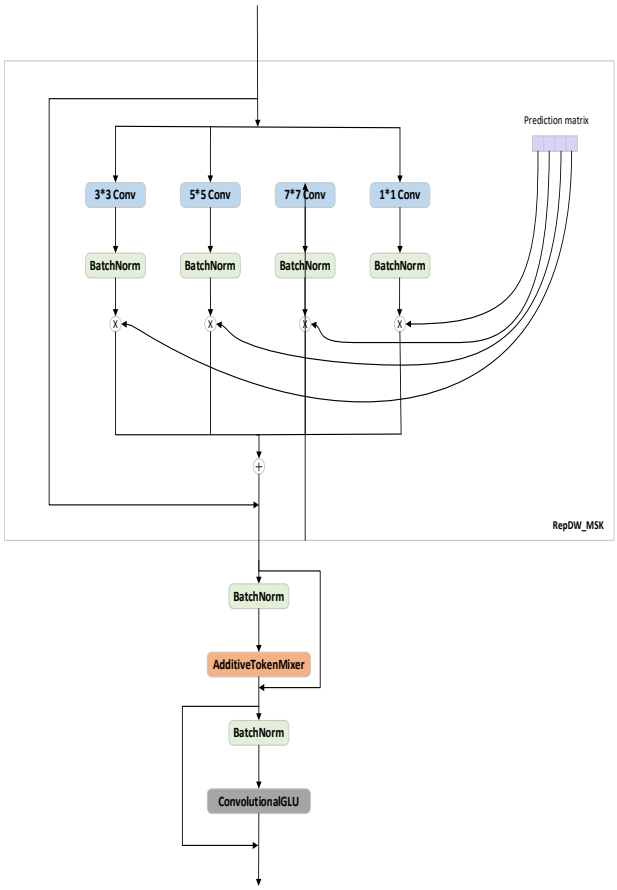


Figure 1. Overview of RT-DETR.

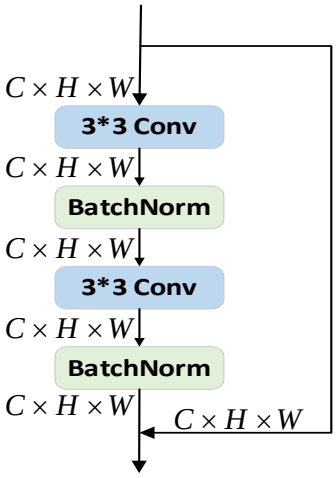
A. ARMix Bottleneck

In Transformer-based detectors, multi-scale features extracted by the backbone are typically fed directly into the encoder for global feature interaction. However, low-level features lack explicit semantic information. Incorporating them into global attention calculations alongside high-level features introduces substantial noise and results in significant computational redundancy. This issue is particularly pronounced in UAV aerial imagery scenarios characterized by dense small objects and complex textures, where the reliance on local structural information far outweighs that on global context. Consequently, premature execution of global self-attention is not only inefficient but also diverts the encoder's focus from key regions. To address this bottleneck, the proposed model enhances both local and global feature modeling capabilities within the backbone stage. While the backbone network of YOLOv8 can help reduce the computational burden on subsequent Transformer modules, its original C2f module relies on convolutions with fixed receptive fields and static fusion mechanisms. It lacks the capacity for multi-scale structural modeling and cross-spatial long-range dependency representation, limiting feature expressiveness. Therefore, the framework proposes MS-ARMixNet, a backbone architecture analogous to YOLOv8, designed to enhance the semantic perception capabilities of the backbone without significantly increasing the computational budget, thereby mitigating the encoder's computational bottleneck at the source.

This study designs an efficient hybrid feature module, named ARMix, to replace the C2f Bottleneck, as illustrated in Figure 2.



(a) ARMix.



(b) C2f Bottleneck

Figure 2. Difference between C2f Bottleneck and ARMix

The ARMix module integrates a Multi-branch Depthwise Convolution Perception module, an Additive Token Mixing module, and a CGLU-based Channel Feed-Forward Network. Within this architecture, this work first employs four depthwise convolutional branches with distinct scales to extract local multi-scale features from the input feature X, where the output of each branch is independently normalized as $B_k = BN(DW_k(X))$. Subsequently, this study introduces a set of learnable static weights α to derive normalized coefficients $W_k = \exp(\alpha_k) / \Sigma_j \exp(\alpha_j)$ via a Softmax function, which are utilized to perform weighted fusion across the multi-branch features, yielding

$$L = \mathrm{Proj}\left(\mathrm{ReLU}\left(\sum_k w_k B_k \right) \right) \quad (1)$$

Where Proj denotes a channel-wise 1×1 projection.

Next, the local feature L is flattened along the spatial dimension into a sequence representation $T = \mathrm{Flatten}(L)$ and fed into the Additive Token Mixer $\mathcal{M}(\cdot)$ to facilitate information interaction across spatial positions, resulting in the mixed representation $T = \mathcal{M}(T)$. Finally, the sequence is restored to the feature map format via a reshape operation:

$$F = \mathrm{Reshape}(\tilde{T}) \in \mathbb{R}^{C \times H \times W} \quad (2)$$

Subsequently, the mixed feature F is fed into the CGLU , which serves as the FFN, formulated as:

$$G = \mathrm{CGLU}(\mathrm{Norm}(F)) \quad (3)$$

Finally, ARMix aggregates the features from these three stages via residual connections, incorporating an optional DropPath (denoted as $D(\cdot)$) to enhance training stability. Consequently, the overall output is defined as:

$$Y = X + L + D(F) + D(\mathrm{CGLU}(\mathrm{Norm}(X + L + D(F)))) \quad (4)$$

These modules function synergistically to efficiently achieve local-global information fusion within the backbone. The Multi-branch Depthwise Convolution Perception module extracts local structural features at varying scales via four convolutional branches, utilizing Softmax-based learnable weights for additive fusion. Following this, the Additive Token Mixer replaces traditional self-attention, achieving cross-spatial long-range dependency modeling with linear complexity. This approach offers reduced computational overhead and improved gradient stability, enabling partial global semantic modeling at the backbone stage. Finally, the CGLU enhances channel-wise non-linear expressive capability; its gating mechanism selectively emphasizes semantically relevant channels while suppressing redundant features, thereby rendering the output features more discriminative.

B. DAFPN

Existing feature pyramid architectures predominantly adopt static fusion strategies, assigning fixed weights to features at different scales regardless of the input content. This fusion process lacks adaptive mechanisms at the spatial-channel level, failing to account for the significant variations in feature effectiveness caused by

diverse object sizes and texture complexities. Consequently, static fusion makes the model susceptible to high-level semantics overshadowing low-level details or low-level noise propagating to higher levels, thereby compromising the overall quality of feature representation.

To address these limitations, this paper proposes the Dynamic Adaptive Feature Pyramid Network (DAFPN), which achieves "on-demand fusion" of multi-scale features via a dynamic three-branch mechanism. Specifically, the Dynamic Weight Generator within the model adaptively predicts fusion weights for each level based on the global descriptors of the input features. To further capture non-linear inter-level correlations, this model introduces a Cross-Input Redistribution Mechanism that fine-tunes the initial weights using a learnable coefficient α, thereby enhancing feature robustness in complex scenarios. The weight generation process can be formally expressed as:

$$W_i = \mathrm{Softmax}\left(\mathcal{F}_{main}(\mathbf{P}) + \tanh(\alpha)\cdot \mathcal{F}_{realloc}(\mathbf{P})\right) \quad (5)$$

Specifically, $\mathbf{P}$ denotes the aggregated vector obtained via global pooling of multi-scale features. $\mathcal{F}_{main}$ and $\mathcal{F}_{realloc}$ represent two parallel convolutional mapping functions, functioning as the primary weight generator and the redistribution corrector, respectively. Furthermore, to mitigate the loss of spatial information induced by pooling operations, DAFPN incorporates a lightweight context residual path. This path utilizes depthwise separable convolutions to extract local spatial context and employs a joint gating mechanism to perform element-wise modulation on the fused features. Consequently, the final output is formulated as:

$$Y = \left(\sum_{i=1}^{N} W_i \odot F_i + \tanh(\beta)\cdot C\left(\sum F_i\right)\right) \otimes \sigma(G_{gate}) \quad (6)$$

Where $\odot$ denotes element-wise multiplication, $C(\cdot)$ represents the context extraction operation, $\sigma(\cdot)$ is the Sigmoid activation function, and β is a learnable parameter controlling the residual intensity.

The dynamic weight prediction and cross-input redistribution mechanisms within DAFPN effectively model scale dependencies, significantly mitigating feature conflicts and information redundancy often encountered with traditional fusion strategies in scenarios featuring dense small objects or complex background textures. Furthermore, the enhancement path, integrated with lightweight context residuals, significantly improves feature discriminability in fine-grained regions while maintaining controllable inference overhead.

C. Evaluation Metrics

In terms of detection accuracy, this study employ *mAP* for evaluation. *mAP* represents the mean of the *AP* across all categories, providing a comprehensive reflection of the model's detection capabilities. Specifically, mAP_{50} denotes the *mAP* calculated at an IoU threshold of 0.5, primarily measuring whether the model's bounding box localization achieves a baseline level of accuracy. Furthermore, $mAP_{50\text{-}95}$ represents the average *mAP* computed over 10 different IoU thresholds ranging from 0.5 to 0.95 (with a step size of 0.05). This metric imposes more stringent requirements on localization precision and serves as the core indicator for assessing the model's overall robustness. The calculation formula is defined as follows:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \quad (7)$$

$$mAP_{50-95} = \frac{1}{10}\sum_{j=0}^{9} mAP_{0.5+0.05j} \quad (8)$$

Where N represents the total number of categories, and AP_i denotes the Average Precision for the i-th category.

Furthermore, Precision measures the proportion of true positive predictions among all

positive predictions generated by the model, calculated as:

$$\frac{TP}{TP+FP} \tag{9}$$

Where *TP* denotes the number of correctly detected objects, and *FP* denotes the number of background regions incorrectly detected as objects. Higher Precision indicates a lower false detection rate, implying greater prediction accuracy.

Recall evaluates the model's ability to detect all ground-truth objects and is calculated as:

$$\frac{TP}{TP+FN} \tag{10}$$

Where *FN* represents the number of missed ground-truth objects. A higher Recall indicates that the model identifies a greater number of actual objects, demonstrating stronger coverage capability. Since a trade-off typically exists between Precision and Recall, both metrics must be considered synergistically in practical tasks to accurately characterize model performance across different scenarios.

To verify the lightweight nature of the model, this model employs Params and GFLOPs as evaluation metrics. Params measures the model size and memory usage, reflecting spatial complexity. GFLOPs represents the computational cost during image processing, reflecting time complexity and theoretical inference speed. Lower values for these two metrics indicate higher model efficiency, facilitating deployment on resource-constrained edge devices.

D. Algorithm Implementation

The forward propagation algorithm within the model primarily encompasses backbone feature extraction, multi-scale feature fusion, and final object detection prediction. By facilitating the interaction and enhancement of multi-scale features, this process significantly improves the model's capability for object localization and classification. The specific execution flow of the algorithm is detailed in Algorithm 1.

This algorithm outlines the forward propagation process of the DART-DETR object detector. It encompasses backbone parameter initialization and multi-scale feature extraction (Lines 1-2), ARMix-based feature enhancement (Lines 3-5), global context modeling and feature projection (Lines 6-7), bidirectional feature fusion via the DAFPN module (Lines 8-12), and finally, Transformer decoder-based query generation followed by the filtering and output of the final detection results (Lines 13-15).

Algorithm 1: Forward Process of DART-DETR Object Detector

Input: Input image $\mathrm{I} \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3}$, confidence threshold τ
Output: Final set of object detections D_{final}.

1: Initialize backbone parameters $\theta_{backbone}$.
2: Extract multi-scale features: $\{\mathrm{F}_2, \mathrm{F}_3, \mathrm{F}_4, \mathrm{F}_5\} \leftarrow \text{MS-ARMixNet}(\mathrm{I})$.
3: for $\mathrm{k} \in \{3,4,5\}$ do
4: $\mathrm{P}_k \leftarrow \text{C2f\_with\_ARMix}(\mathrm{F}_k)$
5: End for
6: $C_5 = \mathrm{Conv}_{1\times1}\left(\mathrm{AIFI}\left(\mathrm{Conv}_{1\times1}(P_5)\right)\right)$
7: $\mathrm{C}_4 \leftarrow \mathrm{Conv}_{1\times1}(\mathrm{P}_4); \mathrm{C}_3 \leftarrow \mathrm{Conv}_{1\times1}(\mathrm{P}_3)$
8: $\mathrm{P}_4^{\mathrm{td}} \leftarrow \mathrm{Fusion}_{\mathrm{DAFPN}}(\mathrm{Upsample}(\mathrm{C}_5), \mathrm{C}_4)$
9: $\mathrm{P}_3^{\mathrm{fuse}} \leftarrow \mathrm{Fusion}_{\mathrm{DAFPN}}(\mathrm{Upsample}(\mathrm{P}_4^{\mathrm{td}}), \mathrm{C}_3)$
10: $\mathrm{P}_3^{\mathrm{td}} \leftarrow \mathrm{Fusion}_{\mathrm{DAFPN}}\left(\mathrm{P}_3^{\mathrm{fuse}}, \mathrm{C}_3, \mathrm{Conv}(\mathrm{F}_2)\right)$
11: $\mathrm{P}_4^{\mathrm{out}} \leftarrow \mathrm{Fusion}_{\mathrm{DAFPN}}\left(\mathrm{Downsample}\left(\mathrm{P}_3^{\mathrm{td}}\right), \mathrm{P}_4^{\mathrm{td}}, \mathrm{C}_4\right)$
12: $\mathrm{P}_5^{\mathrm{out}} \leftarrow \mathrm{Fusion}_{\mathrm{DAFPN}}(\mathrm{Downsample}(\mathrm{P}_4^{\mathrm{out}}), C_5)$
13: $D_{final} \leftarrow \mathrm{RTDETRDecoder}\left(P_3^{td}, P_4^{out}, P_5^{out}\right)$
14: Return D_{final}

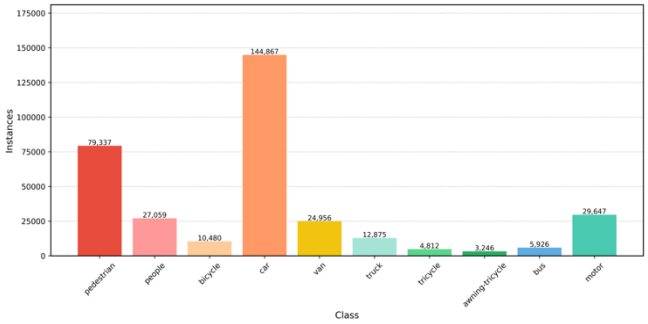
IV. EXPERIMENTS

Following an introduction to the VisDrone 2019 dataset, the experiments demonstrate a comparison between DART-DETR and mainstream models on this benchmark. Subsequently, detailed ablation studies are conducted on the architecture, accompanied by in-depth analysis and qualitative results. Finally, this paper elucidates the distinctions and parameter variations in the training process of DART-DETR.

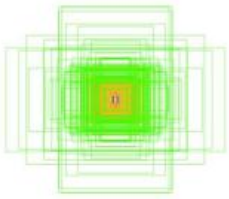
E. Experimental Setup

1) Datasets:

This work conduct experiments on the VisDrone 2019 dataset. Composed of images captured by UAVs, this dataset presents significant challenges such as uneven illumination and object occlusion, covering 10 object categories including pedestrians, vehicles, and buses. Specifically, it comprises 6,471 images for training and 548 images for validation. An analysis of the training set images is illustrated in Figure 3.



(a) Class Distribution of the Training Set



(b) Bounding Box Scale Visualization of the Training Set

Figure 3. Training Set Statistics and Bounding Box Scale Visualization.

As illustrated in Figure 3(a), the class distribution within the training set exhibits significant imbalance. Figure 3(b) depicts the superimposed distribution of all annotated bounding boxes from the VisDrone 2019 training set within a normalized coordinate system. Each rectangle represents the normalized position and dimensions of a target's bounding box, visualized in a semi-transparent manner. The bounding boxes exhibit a highly dense overlap in the central region (transitioning gradually from green to yellow and orange), indicating that a vast number of targets are concentrated in the image center; these typically correspond to numerous small or medium-sized objects such as pedestrians and vehicles. Conversely, the larger, light-green rectangles in the peripheral regions correspond to less frequent large objects. The overall distribution displays an approximately symmetric morphology, suggesting that the target aspect ratios are relatively stable and the data annotations are free from significant bias or anomalies. Intuitively, it is evident that the majority of target boxes are concentrated within the small-scale range.

2) Experimental Configuration

The hardware configuration of the experimental platform is detailed in Table 1.

TABLE I. HARDWARE CONFIGURATION

Description	model
CPU	12 vCPU Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz
GPU	NVIDIA GeForce RTX 3080 Ti
Language	Python 3.10
Pytorch	2.0.0
CUDA	11.8

The detailed hyperparameter settings employed during training are presented in Table 2. To ensure fair comparison, all experiments utilize consistent initialization strategies and data augmentation protocols.

TABLE II. HYPERPARAMETER CONFIGURATION

Hyperparameter	Value
Input size	640×640
Batch size	4
Training epochs	200
Optimizer	AdamW
Initial learning rate	0.0001
Learning rate factor	1.0
Momentum	0.9
Warmup steps	2000

F. Comparison Experiments

This paper compares DART-DETR with representative two-stage and one-stage detectors, as presented in Table 3.

Comparative experiments demonstrate that DART-DETR exhibits significant advantages over traditional two-stage methods. For instance, Faster R-CNN and ARFP achieve mAP_{50} scores of 31.2% and 33.7% on VisDrone2019, respectively. In contrast, DART-DETR reaches 50.0%, yielding improvements of 18.8% and 16.3% in mAP_{50}.

Furthermore, DART-DETR requires significantly fewer parameters and FLOPs compared to these models, highlighting the advantages of the end-to-end framework in terms of model efficiency and feature redundancy control.

Benefiting from the dynamic cross-scale fusion of DAFPN and the enhanced local-global hybrid modeling capabilities of the improved ARMix, DART-DETR demonstrates superior competitiveness in terms of mAP_{50} and $mAP_{50\text{-}95}$ compared to one-stage detectors such as YOLOv12-M, HIC-YOLOv5, and MSFE-YOLO-L. This advantage is particularly pronounced in scenarios involving dense small objects, where our model exhibits greater stability.

Compared with mainstream end-to-end Transformer detectors, DART-DETR achieves a superior trade-off between lightweight design and accuracy. For example, while RT-DETR-R50 achieves 50.7% mAP_{50} and 30.8% $mAP_{50\text{-}95}$, DART-DETR attains comparable accuracy with significantly reduced resource consumption, utilizing only 15.4M parameters and 59.7 GFLOPs.

Moreover, compared to Deformable DETR and DETR, DART-DETR achieves precision improvements of 6.8% and 10.0%, respectively, while maintaining substantially lower complexity. This effectively addresses the limitations of the DETR series in small object detection scenarios. During the inference phase, DART-DETR fully retains the end-to-end advantages of the DETR family, eliminating the need for additional candidate box generation and NMS. This results in a more concise overall architecture and a highly efficient inference workflow.

TABLE III. COMPARISON OF DART-DETR WITH STATE-OF-THE-ART METHODS ON THE VISDRONE 2019 VALIDATION SET. FOR EACH METRIC, THE BEST RESULTS ARE HIGHLIGHTED IN BOLD RED AND THE SECOND-BEST RESULTS IN BOLD BLUE

Model		Img size	mAP50(%)	mAP50-95 (%)	Params(M)	FLOPs(G)
Two-stage methods	Faster R-CNN	640*640	31.2	18.3	41.2	127.0
	ARFP[14]	1333*800	33.7	20.9	42.7	193.8
	EMA[15]	640*640	49.4	30.1	91.2	-
One-stage methods	YOLOv12-M[16]	640*640	43.5	26.0	20.2	67.5
	HIC-YOLOv5[17]	640*640	44.6	26.2	10.5	31.2
	MSFE-YOLO-L[18]	640*640	47.0	29.2	41.6	160.2
	YOLO-DCTI[19]	1024*1024	49.7	27.6	37.7	-
End-to-end methods	DETR	1333*750	40.0	24.2	40.0	187.1
	Deformable DETR	1333*800	43.2	27.5	40.2	172.5
	RT-DETR-R18	640*640	47.2	29.0	20.0	57.0
	RT-DETR-R50	640*640	50.7	30.8	41.8	133.2
Ours	DART-DETR	640*640	50.0	30.9	15.4	59.7

G. Ablation Experiments

This paper conducts systematic ablation studies based on RT-DETR on the VisDrone 2019 validation set to validate the effectiveness of the proposed modules. The results are presented in Table 3:

The baseline model yields 47.2% and 29.0% in terms of mAP_{50} and $mAP_{50\text{-}95}$, respectively. Upon this foundation, introducing solely DAFPN boosts the mAP_{50} to 49.2%, representing a 2.0% gain over the baseline, while $mAP_{50\text{-}95}$ improves by 1.4%. This indicates that the dynamic scale-aware feature fusion structure effectively enhances multi-scale semantic representation capabilities.

TABLE IV. ABLATION RESULTS BASED ON THE RT-DETR BASELINE ON THE VISDRONE 2019 DATASET. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD.

MS-ARMixNet	DAFPN	mAP50	mAP50-95
-	-	47.2	29.0
-	√	49.2	30.4
√	√	50.0	30.9

When integrating both MS-ARMixNet and DAFPN, the model performance is further elevated: mAP_{50} reaches 50.0%, achieving an overall improvement of 2.8% over the baseline, and $mAP_{50\text{-}95}$ rises to 30.9%, an increase of 1.9%. These results strongly demonstrate the complementarity between the enhanced local-global feature modeling capability of MS-ARMixNet and the dynamic multi-scale fusion

mechanism of DAFPN. Their synergistic effect significantly improves detection performance for small and dense objects in UAV aerial photography scenarios.

H. Analysis of Training Results

1) confusion_matrix_normalized

The confusion matrix illustrates the fine-grained classification performance of DART-DETR across the 10 object categories on the VisDrone 2019 dataset, as shown in Figure 4:

Figure 4. Confusion_matrix_normalized

Overall, the model exhibits high recognition stability across the majority of categories. Notably, the high diagonal values for categories such as car, bus, and motor indicate the model's robust discriminative capability for these relatively large and structurally regular traffic objects.

For pedestrian and people categories, the model also maintains high recognition accuracy. Although a subset of samples is misclassified as background or confused with each other, this trend is consistent with the characteristics of the VisDrone 2019 dataset, which features small object sizes and severe occlusions for pedestrians. In extremely dense scenarios, influenced by occlusion, motion blur, and long-distance viewpoints, small or structurally similar categories like awning-tricycle and bicycle exhibit higher misclassification rates. They are particularly prone to being confused with background or visually similar categories such as tricycle and motor. The confusion matrix suggests that while DART-DETR maintains stable precision across most categories, there remains room for further optimization regarding the fine-grained classification of small objects.

2) Training and Validation Dynamics of the Detect

The convergence curves of the training process are shown in Figure 5:

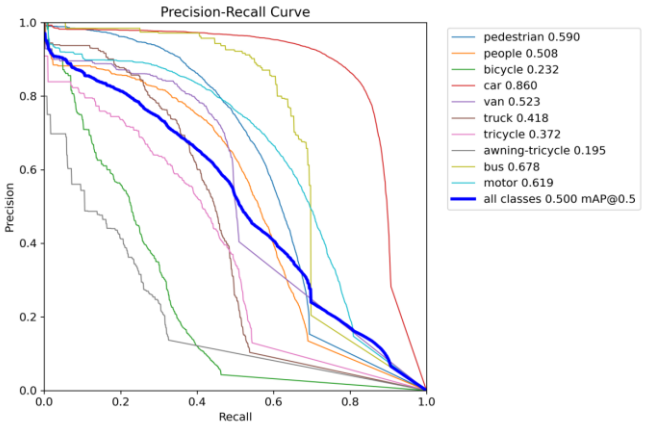


Figure 5. Training and validation curves of the model on the VisDrone2019-DET dataset, including loss convergence (GIoU, classification, and objectness losses) and evaluation metrics (precision, recall, mAP_{50}, and $mAP_{50\text{-}95}$).

As observed from the training curves, DART-DETR exhibits excellent convergence and stability throughout the 200-epoch training process. Regarding the localization-related losses, namely GIoU Loss and L1 Loss, the model demonstrates a rapid descent during the initial training phase, enters a stable convergence region after 20–30 epochs, and ultimately converges to a relatively low level. The loss curves of the validation set are highly consistent with those of the training set, showing no significant overfitting or oscillation phenomena. This indicates that the model's structural design and regularization mechanisms have effectively enhanced training stability. Although the Cls Loss exhibits high fluctuation in the initial stage, it gradually decreases and maintains a plateau trend as training progresses. This suggests that DART-DETR's feature fusion module and decoding architecture are effectively learning inter-class discriminability, thereby continuously strengthening the model's classification capability.

In terms of the trends in accuracy metrics, Precision, Recall, mAP50, and mAP50-95 all display a continuous monotonic increase. Notably, mAP50 stabilizes after approximately 150 epochs, reaching a final value of 0.50. Precision also steadily rises to approximately 0.58, with Recall approaching 0.50, demonstrating that the model effectively reduces the missed detection rate while minimizing false detections, resulting in more reliable prediction outcomes. The overall process is free from gradient explosion or performance degradation phenomena, highlighting the optimization advantages of the end-to-end Transformer framework combined with the improved feature fusion module.

In conclusion, the training curves demonstrate that DART-DETR possesses excellent trainability and convergence properties. The stable decrease in loss and the upward trend in metrics illustrate the model's strong adaptability to complex scenarios.

3) PR_curve

As shown in Figure 6, the PR curves demonstrate the detection performance of DART-DETR across different categories.

Figure 6. PR_curve

The model achieves an mAP50 of 50.0%, exhibiting distinct performance variations across different categories.

For vehicle categories such as car, bus, and motor, the PR curves demonstrate large AUC with a smooth descending trend. This indicates that the model possesses robust detection stability for these targets, maintaining high precision and recall under both high and low confidence threshold conditions.

regarding moderate-difficulty categories like pedestrian, people, and van, the AP values are 0.590, 0.508, and 0.523, respectively. The trajectories of their PR curves are relatively balanced, suggesting that the model remains stable when handling dense crowds and visually similar vehicle types, although it is still susceptible to the effects of occlusion and target scale variations.

In contrast, small objects and structurally similar categories, such as bicycle and awning-tricycle, exhibit a sharp decline in their PR curves and lower AP scores. This performance drop is primarily attributed to the small scale, blurred edges, and irregular shapes of these targets, as well as their tendency to be confused with the background or other tricycle variants. This phenomenon aligns with the misclassification patterns observed in the confusion matrix, confirming that small objects and fine-grained categories remain the most challenging aspects of object detection from a UAV perspective.

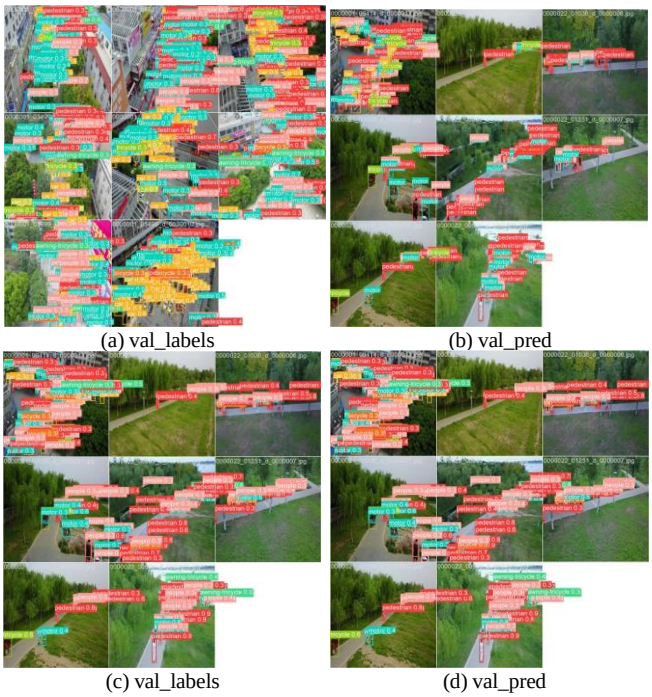
In conclusion, these results further validate the effectiveness of the proposed model and demonstrate the significant role of the adopted cross-scale feature fusion mechanism in enhancing detection capabilities within complex scenes.

4) Visualization of Ground-Truth Labels and Predicted Bounding Boxes

During the training process, the Ultralytics framework automatically generates visualization results for both the training and validation phases, as shown in Figure 7.

The figure above illustrates a comparison between the ground truth annotations and the model's prediction results on a sample batch from the validation set. From an overall perspective, the model demonstrates the capability to accurately identify the majority of common object categories within complex urban environments, including pedestrian, people, car, motor, and bus. The substantial overlap between the predicted bounding boxes and the ground truth labels indicates that the model possesses strong localization capabilities and class discriminability for primary targets.

In dense street scenes, the model maintains robust detection stability, exhibiting particularly outstanding performance on vehicle categories (e.g., car, bus, motor), where the predictions show high consistency with the ground truth and minimal missed detections. Simultaneously, regarding pedestrian and people, the model delivers relatively stable detection results, although sporadic missed detections or lower confidence scores are observed in regions with partial occlusion or at long distances.



(a) val_labels (b) val_pred

(c) val_labels (d) val_pred

Figure 7. Visualization results.val_labels denotes the ground truth bounding boxes, and val_pred represents the predicted bounding boxes on the validation set.

Regarding complex categories with morphological similarities, such as tricycle and awning-tricycle, the model achieves a considerable number of correct predictions, yet some minor confusion persists. This observation aligns with

the analyses of the confusion matrix and PR curves discussed earlier, confirming that the small scale and visual similarity of these targets pose challenges to the model.

In conclusion, the high consistency between the predictions and the ground truth demonstrates the model's excellent generalization capability and detection reliability, enabling stable object recognition performance amidst dense traffic and intricate backgrounds.

V. Conclusions

In this paper, the model proposes DART-DETR, an end-to-end detector tailored for real-time object detection in UAV imagery, which successfully enhances both the applicability and lightweight performance of Transformer-based detectors in real-time scenarios. Specifically, this study integrates the MS-ARMixNet Backbone into DART-DETR and introduce the ARMix module to achieve hybrid local-global modeling, thereby strengthening semantic and detailed representations without significantly increasing the computational burden. Furthermore, this paper proposes the DAFPN to adaptively perform dynamic weighted fusion on multi-scale features. Although the GFLOPs of DART-DETR increased by only approximately 4.7% compared to RT-DETR-R18, the model achieved significant gains in detection performance: mAP50 improved from 47.2% to 50.0%, and mAP50increased from 29.0% to 30.9%. Simultaneously, the number of model parameters was reduced by approximately 23%, fulfilling the design objective of achieving higher detection performance with a smaller model size. The proposed model provides an effective technical pathway for real-time detectors in resource-constrained aerial photography scenarios and holds promising prospects for engineering applications.

References

[1] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(6): 1137-1149.

[2] Zhu X, Su W, Lu L, et al. Deformable detr: Deformable transformers for end-to-end object detection[J]. arXiv preprint arXiv:2010.04159, 2020.

[3] Zhang H, Li F, Liu S, et al. Dino: Detr with improved denoising anchor boxes for end-to-end object detection[J]. arXiv preprint arXiv:2203.03605, 2022.

[4] Zhao Y, Lv W, Xu S, et al. Detrs beat yolos on real-time object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024: 16965-16974.

[5] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[6] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[7] Farhadi A, Redmon J. Yolov3: An incremental improvement[C]//Computer vision and pattern recognition. Berlin/Heidelberg, Germany: Springer, 2018, 1804: 1-6.

[8] Khanam R, Hussain M. What is YOLOv5: A deep look into the internal features of the popular object detector[J]. arXiv preprint arXiv:2407.20892, 2024.

[9] Ge Z, Liu S, Wang F, et al. Yolox: Exceeding yolo series in 2021[J]. arXiv preprint arXiv:2107.08430, 2021.

[10] Hussain M, Nadeem M A, Khan A M. YOLOv8: an in-depth exploration of the internal features of the next-generation object detector[J]. arXiv preprint arXiv:2408.15857, 2024.

[11] Chen Q, Chen X, Wang J, et al. Group detr: Fast detr training with group-wise one-to-many assignment[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2023: 6633-6642.

[12] Liu S, Li F, Zhang H, et al. Dab-detr: Dynamic anchor boxes are better queries for detr[J]. arXiv preprint arXiv:2201.12329, 2022.

[13] Li F, Zhang H, Liu S, et al. Dn-detr: Accelerate detr training by introducing query denoising[C]// Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 13619-13627.

[14] Wang J, Yu J, He Z. ARFP: A novel adaptive recursive feature pyramid for object detection in aerial images[J]. Applied Intelligence, 2022, 52(11): 12844-12859.

[15] Ouyang D, He S, Zhang G, et al. Efficient multi-scale attention module with cross-spatial learning[C]//ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2023: 1-5.

[16] Tian Y, Ye Q, Doermann D. Yolov12: Attention-centric real-time object detectors[J]. arXiv preprint arXiv:2502.12524, 2025.

[17] Tang S, Zhang S, Fang Y. HIC-YOLOv5: Improved YOLOv5 for small object detection[C]//2024 IEEE international conference on robotics and automation (ICRA). IEEE, 2024: 6614-6619.

[18] Qi S, Song X, Shang T, et al. Msfe-yolo: An improved yolov8 network for object detection on drone view[J]. IEEE Geoscience and Remote Sensing Letters, 2024.

[19] Min L, Fan Z, Lv Q, et al. YOLO-DCTI: small object detection in remote sensing base on contextual transformer enhancement [J]. Remote Sensing, 2023, 15(16):3970.