RemDet: Rethinking Efficient Model Design for UAV Object Detection

Chen Li[1,2], **Rui Zhao**[1,2], **Zeyu Wang**[1,2], **Huiying Xu**[1,2*], **Xinzhong Zhu**[1,2*]

[1]School of Computer Science and Technology, Zhejiang Normal University, Zhejiang, 321004, China
[2]Research Institute of Hangzhou Artificial Intelligence, Zhejiang Normal University, Zhejiang, 311231, China
{lilsodachen,zhaorui,14797857499,xhy,zxz}@zjnu.edu.cn

Abstract

Object detection in Unmanned Aerial Vehicle (UAV) images has emerged as a focal area of research, which presents two significant challenges: i) objects are typically small and dense within vast images; ii) computational resource constraints render most models unsuitable for real-time deployment. Current real-time object detectors are not optimized for UAV images, and complex methods designed for small object detection often lack real-time capabilities. To address these challenges, we propose a novel detector, RemDet (Reparameter efficient multiplication Detector). Our contributions are as follows: 1) Rethinking the challenges of existing detectors for small and dense UAV images, and proposing information loss as a design guideline for efficient models. 2) We introduce the ChannelC2f module to enhance small object detection performance, demonstrating that high-dimensional representations can effectively mitigate information loss. 3) We design the GatedFFN module to provide not only strong performance but also low latency, effectively addressing the challenges of real-time detection. Our research reveals that GatedFFN, through the use of multiplication, is more cost-effective than feed-forward networks for high-dimensional representation. 4) We propose the CED module, which combines the advantages of ViT and CNN downsampling to effectively reduce information loss. It specifically enhances context information for small and dense objects. Extensive experiments on large UAV datasets, Visdrone and UAVDT, validate the real-time efficiency and superior performance of our methods. On the challenging UAV dataset VisDrone, our methods not only provided state-of-the-art results, improving detection by more than **3.4**%, but also achieve **110** FPS on a single 4090.

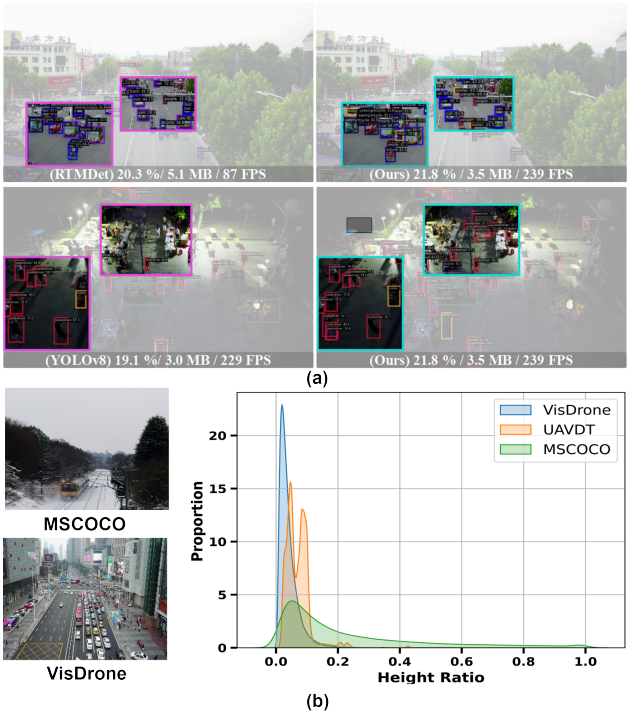


Figure 1: (a) Visualization of real-time detector results on VisDrone, white characters below the image represent mAP, model size, and FPS. (b) Kernel density analysis of detection object width on UAV and COCO datasets.

Introduction

Recent years have witnessed significant progress in object detection techniques, including the success of general detectors like Faster R-CNN (Girshick 2015), YOLO (Redmon et al. 2016; Redmon and Farhadi 2017), and DETR (Carion et al. 2020). Additionally, researchers have explored lightweight and efficient architectures tailored specifically for object detection. Despite these advancements, Unmanned Aerial Vehicle (UAV) images present unique challenges due to small and dense objects. Object detection in UAV images is a critical research area with applications in surveillance, disaster management, and environmental monitoring. Captured from an aerial perspective, UAV datasets exhibit a higher prevalence and density of small objects compared to traditional datasets (as illustrated in Figure 1 (b)). For instance, while the MSCOCO (Lin et al. 2014) dataset contains an average of 7 objects per image, the VisDrone (Zhu et al. 2021) dataset contains an average of 53 objects.

To address the detection challenges posed by small and dense objects, Region of Interest (RoI) methods are widely adopted. (Ding et al. 2018) and (Ashraf, Sultani, and Shah 2021) can amplify or prioritize the selection of regions of interest to enhance object visibility and distinguishability. Besides, current mainstream UAV detectors tend to employ

*Corresponding Author.

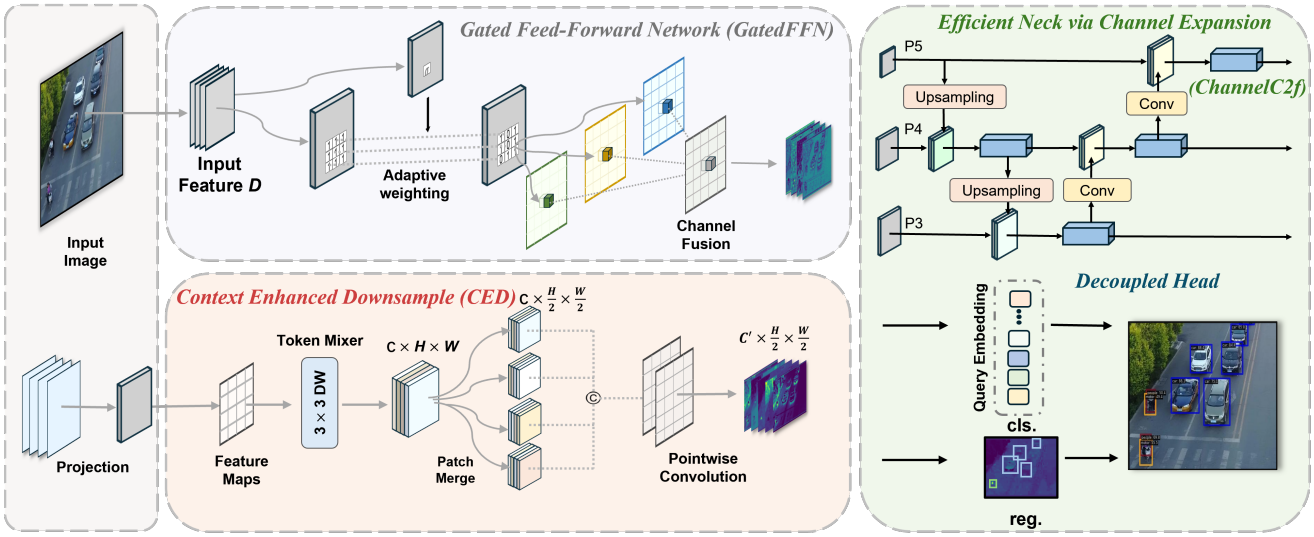


Figure 2: Overview of the proposed RemDet. Our method consists of three components: GatedFFN achieves adaptive weighting through multiplication, followed by channel fusion. CED utilizes patch merge operations to concatenate channels. ChannelC2f and decoupled heads are employed for prediction.

density cropping methods (Meethal, Granger, and Pedersoli 2023), achieved through heavily handcrafted designs, to achieve background suppression or enhance fusion.

On the contrary, lightweight models have made significant progress in the field of generic object detection. By leveraging techniques such as depthwise separable convolutions (Howard et al. 2019), reparameterization (Ding et al. 2021), and pruning (Ye et al. 2023), they exhibit outstanding performance in object detection tasks. However, research specifically focused on lightweight models for UAV object detection remains relatively scarce. QueryDet (Yang, Huang, and Wang 2022) and CEASC (Du et al. 2023) use sparse convolutions in their detection heads to reduce model weights, which lowers computational requirements. However, they still rely on complex handcrafted designs and lack hardware optimization, hindering real-time efficiency.

This raises a question: Is it possible to balance the efficiency and accuracy of UAV detection through device-friendly operations rather than heavily handcrafted designs?

To address the challenges of detecting small and dense objects as well as achieving high-speed inference, in this paper, we propose RemDet (Reparameter efficient multiplication Detector), a one-stage anchor-free detector designed for real-time UAV object detection. Our approach rethinks the design of UAV detectors with the overarching goal of reducing information loss. Specifically, for small object detection, we introduce ChannelC2f and Context Enhanced Downsample (CED). The former extends the C2f with additional channels, providing a simple yet effective enhancement for small object detection. The latter combines the advantages of lightweight detectors (Sandler et al. 2018) and ViT (Dosovitskiy et al. 2020) downsampling, effectively enhancing contextual information and reducing information loss. To meet the most demanding real-time detection requirements, we introduce GatedFFN. This model employs cost-efficient operations to achieve high-dimensional representation. Additionally, GatedFFN reparameterizes two convolutions, effectively balancing performance and speed. The maximum version, RemDet-X, trained on high-resolution UAV images, achieves an mAP of 40%, with inference latency as low as 9 ms on a single 4090, achieving 110 FPS.

The main contributions of our research include:

1. We rethink the design of UAV detectors, discarding complex handcrafted designs. By exploring information loss, we effectively enhanced small object detection using a simplest structure.
2. Following the principle of reducing information loss, our research revealed that high-dimensional representation alone can reduce information loss and enhance small object performance. We validate our analysis through empirical results (see Figure 4 (b)), theoretical exploration (in Section 3.2), and visual representations (Figure 4 (a)).
3. To address the challenging real-time requirements, where complex designs and multi-feature fusion are impractical for accuracy improvement, our study reveals that multiplication, rather than feedforward networks, serves as a cost-effective and simpler high-dimensional representation. Our designs, based on this insight, reduces information loss while maintaining low latency.

Related Work

Object Detection for UAV images Unlike general object detection, UAV object detection has always focused on designing methods that transition from coarse to fine granularity. Addressing the non-uniform distribution of small objects in images, ClusDet (Yang et al. 2019) utilized a clustering-based scale estimation method, effectively enhancing small object detection. UFPMP-Det (Huang, Chen, and Huang 2022) first merged sub-regions provided by a coarse detector through

clustering to suppress the background, then packaged the results into a mosaic for single inference. AMRNET (Wei et al. 2020) significantly expanded the coarse-to-fine framework through two specially designed modules. CZDet (Meethal, Granger, and Pedersoli 2023), based on density cropping method, first detected density-cropped regions and basic category objects during inference, then inputs them into the second stage of inference. Additionally, YOLC (Liu et al. 2024) adaptively searched for clustered regions based on CenterNet (Duan et al. 2019), adjusted them to appropriate scales, and improves the loss function to enhance performance. However, most of these works are designed for detection heads or feature fusion layers, neglecting the information loss during the backbone stage. Above all, their real-time performance is also hindered by heavily handcrafted design.

Real-time detection of UAV images In real-time UAV detection, one-stage detectors like YOLO (Jocher 2020; Li et al. 2022; Wang, Bochkovskiy, and Liao 2023) are widely used. The YOLO series has consistently aimed for real-time object detection, showcasing strong vitality through continuous updates and iterations. YOLOv8 (Jocher, Chaurasia, and Qiu 2023), in particular, improved real-time performance with its simple and effective C2f and decoupled head. However, on UAV images, the efficient extraction modules designed for these detectors often perform poorly due to background interference, as the objects to be detected are small and dense. Designing modules solely to enhance small object detection often fails to balance real-time performance.

Our work focuses on achieving a balance between small object detection and real-time performance, using more hardware-friendly designs rather than heavily handcrafted designs for UAV detection.

Method

Exploring Designs for Efficient Models

In order to further enhance model performance in complex scenarios, researchers have begun investigating factors that affect detection performance, with an increasing focus on the information bottleneck theory.

Principles for Hidden Layer Design We rethink the information bottleneck definition to gain design insights. Simply put, the input variable is defined as X and the output variable as Y. The hierarchical structure of a DNN forms a Markov chain, which can approximately represent all relationships and data. Each layer of the DNN relies solely on the input data from the previous layer, meaning that if any layer loses information about Y, it cannot be recovered in deeper layers.

We define the mutual information within the layer as $I(X;Y)$ and describe this process concisely using mathematics. The input variable X has high resolution and low dimensionality, representing the lower-level representation of the data; whereas Y, as the prediction, has high dimensionality and low resolution. This implies that the neural network essentially performs data compression throughout the process. In statistics, we denote the X related to the prediction of Y as X'. Under the constraint $I(X';Y)$, finding X' can be expressed as the minimization of the Lagrangian function.

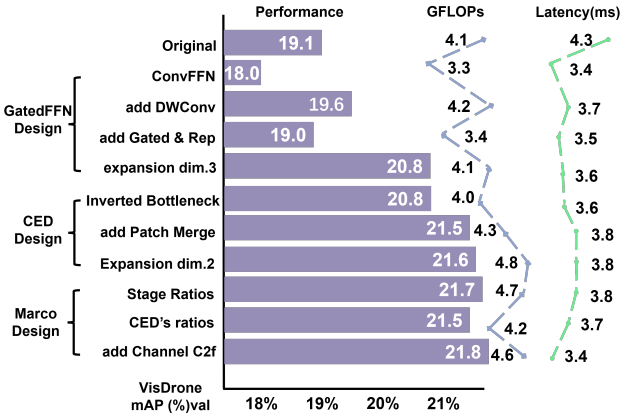


Figure 3: RemDet design process evaluated using FLOP, latency, and mAP.

$I(X;Y|X')$ represents the information between X and Y that is not captured by X', and by replacing the constraint with it, the formula can be equivalently expressed as:

$$L_{p(x'|x)} = I(X;X') + \beta I(X;Y|X') \tag{1}$$

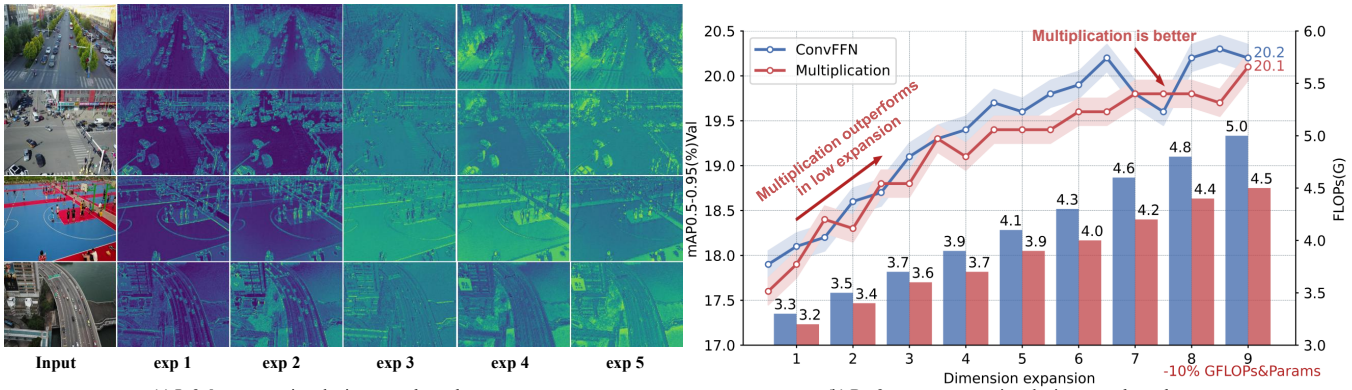
Where $\beta \geq 0$. We further explain its optimization objective: β represents the relaxation variable balancing complexity $I(X;X')$ and irrelevance $I(X;Y|X')$. It can be observed that when irrelevance is 0 and $I(X;X')$ is minimized, the Lagrangian function achieves its minimum value. This implies that X' also reaches its minimum value. In DNN, this indicates that X' should be as simple as possible, i.e., **finding the minimal information from X that satisfies the conditions.**

Due to the complexity of the data X, it is difficult to find the minimal sufficient statistic. When using Y' for prediction, we have $I(X;X') \geq I(Y;Y')$. More generally, for any neural network, it can be expressed as:

$$I(Y;X) \geq I(Y;h_i) \geq I(Y;h_{i+1}) \geq I(Y;Y') \tag{2}$$

Where h_i represents the intermediate information. The above equation holds with equality only if each layer is a sufficient statistic of its input. Therefore, the goal of each layer is to optimally **capture all information relevant to the output in its input and discard all irrelevant parts.**

Dimension Expansion Design Principle In (Tishby and Zaslavsky 2015), the importance of designing compact DNN is emphasized, focusing on reducing the number of layers and minimizing the units per layer. YOLOv9 (Wang, Yeh, and Liao 2024) employs a feature-sharing approach to enable each layer to acquire relevant information from the preceding layer. While theoretically effective, this method incurs high training costs, posing significant challenges to model training. Conversely, other efficient structures in (Hollard et al. 2024), guided by the information bottleneck principle, utilize a simple inverted bottleneck structure to enhance inter-layer information interaction. However, this design lacks detailed explanation. In (Han et al. 2021), a search on channel dimensions was conducted, resulting in the design principle of



(a) **Info loss** comparison by increase *channel expansion*

(b) **Performance** comparison by increase *channel expansion*

Figure 4: (a) Visualization of model features as channel expansion increases. (b) Comparison of Multiplication and ConvFFN.

maintaining constant channel dimensions within layers and linearly increasing them between stages.

In summary, the information bottleneck theory guides us in enhancing the mapping relationship between X and Y. The core objective is to create a cleaner and more compact structure to **achieve simpler intra-layer mappings**. Additionally, the design between layers must maintain consistent input and output dimensions. This brings us to the next design goal: **enhancing intra-layer information**.

How to Design Efficient Modules?

Design for Enhanced Information Interaction After establishing inter-layer design guidelines, our focus shifts to enhance intra-layer information interaction. (Shwartz-Ziv and Tishby 2017) delve into the essence of hidden layers, emphasizing their role in learning from input X while compressing Y information without labels. Our experimental investigations center on a Multilayer Perceptron (MLP), a common choice for dimension expansion when learning from X.

To assess the impact of this operation on UAV detection, we introduce ConvFFN, which employs only two 1×1 convolutions as its backbone, with scaled hidden layer dimensions. The detailed design of ConvFFN is provided in the appendix. Surprisingly, with a channel expansion set to 3, ConvFFN performs comparably to a baseline (Jocher, Chaurasia, and Qiu 2023) that includes dense computations and residual connections, while reducing parameters and computation by approximately 10%. This phenomenon underscores the heightened optimization benefits of minimizing information loss in UAV datasets, where smaller objects prevail compared to general datasets. Visualizing features across different channel expansions (Figure 4 (a)), we observe a significant amplification in feature weights as hidden dimensions increase. This amplification is indicative of enhanced modeling capabilities. Under consistent input-output dimensions, the model more effectively accomplishes the tasks of 'learning' and 'compression'. We deem that higher representations enhance inter-layer interaction and are a form of "learning".

Multiplication Resulting in Higher Representations In deep learning, the MLP serves as a simple and common modeling approach, often used for processing input-output data. However, for pixel-sparse images with small correlations, the operations of MLP can become redundant. Therefore, we need a more efficient method for mapping to high-dimensional.

Gated Linear Units (GLU) (Dauphin et al. 2017) in natural language processing has been considered an alternative to recurrent neural networks (RNNs). We specifically focus on the gating part within GLU, which involves element-wise multiplication.

We analyze the dimension expansion of MLP and GLU from a mathematical perspective. To simplify the analysis, we just consider the case of one-output channel transformation, single-element input. To align the two methods, we assume that the input element $x \in \mathbb{R}^{d\times1}$ and map it to $w_1^T x$ and $w_2^T x$. These two elements are then combined in an MLP. Specifically, the representation of MLP is

$$\begin{aligned} w_0^T x =& w_1^T x + w_2^T x \\ =& (\sum_{i=1}^{d} w_1^i x^i) + (\sum_{j=1}^{d} w_2^j x^j) \end{aligned} \tag{3}$$

Where $w_0 \in \mathbb{R}^{d\times2}$. Represent the multiplication in the same way:

$$\begin{aligned} & w_1^T x * w_2^T x \\ =& (\sum_{i=1}^{d} w_1^i x^i) * (\sum_{j=1}^{d} w_2^j x^j) \\ =& \sum_{i=1}^{d}\sum_{j=1}^{d} w_1^i w_2^j x^i x^j \end{aligned} \tag{4}$$

where $w_1, w_2 \in \mathbb{R}^{d\times1}$. By computing the polynomial sum, it obtain $\frac{(d+1)d}{2}$ distinct terms. From a parameter perspective, multiplication incurs no additional computational cost. Moreover, given that $d \gg 2$, we observe that $\frac{d^2+d}{2} \geq 2d$, indicating higher dimensionality after element-wise multiplication. However, during multiplication, $w_2^T x$ is discarded but implicitly included in the output. As a result, the dimension of the output is halved compared to the original. As the

dimension increases, the impact of $w_2^T x$ becomes more pronounced. This shows that when hidden layers are expanded in low dimensions, The closer the performance of multiplication will be to MLP. We conducted a series of experiments, as shown in Figure 4 (b). The results not only support our conjecture but also reveal that as the dimensionality extension increases, the gains in accuracy from higher dimensions for both methods become closer to each other. Further, the computational demand for multiplication is lower, allowing us to compensate for the implicit dimension loss resulting from polynomial addition by increasing dimension expansion. Notably, Figure 4 (b) illustrates that the computational cost of channel-expanded multiplication (to 9) is comparable to that of MLP (with 7), yet the mAP improves by 0.3%.

We also observe that the multiplication resembles the form of a kernel function. The kernel function is defined as $K(x, z) = \phi(x) \cdot \phi(z)$, where $\cdot$ is expressed as an inner product. In fact, we perform element-wise polynomial multiplication in formula 4, with $w_2^T x$ serving as the mapping function ($\phi(z)$) to increase the result's dimensionality.

Consequently, **we now adopt the multiplication as our primary design approach.**

Module Design Our model design, as depicted in Figure 5, provides a clear exposition of our approach. Based on C2f, we employ a double-branch multiplication, eliminating the Bottleneck structure, and utilize 1×1 and 3×3 depthwise convolutions as reparameterized convolutions in the main branch. Notably, (Wang et al. 2020) emphasizes that direct channel compression may compromise expressive capacity. To address this concern, we set the channel expansion factor to 3, enhancing inter-layer information. Finally, a 1×1 convolutional layer is positioned at the end of the model to compress in-layer information for efficient output. Collectively, these design choices constitute what we refer to as a lightweight structure—the GatedFFN, see Figure 5.

In the Neck layer, we merely scale the C2f (see in Figure 5) channels creating a structure known as ChannelC2f. Specifically, we increase the overall channel expansion from 0.5 to 1.0 and reduce the Bottleneck's expansion ratio from 1 to 0.25, thereby minimizing dense computations. Thus, we enhance intra-layer information solely by adjusting channel expansions.

Context Enhanced Downsample Module

In neural networks, downsampling modules are used to reduce the resolution of feature maps. As far as we know, deepening the downsampling module is an efficient way to mitigate information loss resulting from resolution reduction. For this purpose, EfficientViT (Liu et al. 2023) and RepViT (Wang et al. 2024) deepen the module and incorporate an additional FFN at the end for information compression. In contrast, lightweight CNNs employ simpler modules, such as 3×3 convolutions with a stride of 2 for downsampling, which is extremely fast. However, due to insufficient network depth, concerns arise regarding information loss and performance degradation. In ViT (Dosovitskiy et al. 2020), downsampling is typically achieved using Patch Merge layers, effectively increasing the channel expansion of layers to avoid information loss. Additionally, Convnext (Liu et al. 2022) explores how to adapt CNNs using ViT designs in detail. However, due to the difficulty of Patch Merge in gaining an advantage over convolutions in classification tasks, exploration of downsampling layers was abandoned. This raises a question: Can combining depthwise separable convolutions and Patch Merge improve performance on UAV images?

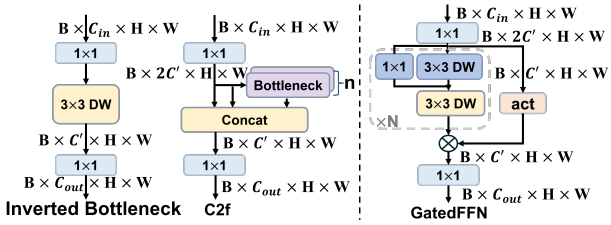


Figure 5: Structure of the IB, C2f, and proposed GatedFFN.

To address this issue, we adopted the Inverted Bottleneck and changed the stride to 1 while setting the input dimension expansion to 1. It is worth noting that the performance of depthwise separable convolutions is limited by the loss of channel information during the split convolution operation. To enhance information capture, we inserted Patch Merge after the depthwise convolution, allowing the subsequent pointwise convolution to obtain richer information. Detailed design specifications can be found in the appendix. We named the entire module the Context Enhanced Downsample (CED). The CED module effectively mitigates information loss while maintaining high-speed inference.

Experiment

Experimental Setup

Datasets To evaluate our method, we conduct UAV detection experiment on the VisDrone (Zhu et al. 2021) and UAVDT (Du et al. 2018), and also included the MSCOCO (Lin et al. 2014) dataset as an additional benchmark. VisDrone comprises 8,599 aerial images across 10 categories, with 6,471 images for training and 548 images for validation, all at a resolution of 2,000×1,500 pixels. Since the evaluation server is currently closed, we followed related works and used the validation set for performance evaluation. MSCOCO contains over 330,000 images with multiple annotations across 80 categories. UAVDT includes 23,258 training images and 15,069 testing images, with a resolution of 1,024×540 pixels across 3 classes.

Evaluation Measures The metric we use to evaluate and compare the performance of various methods is the COCO-style Average Precision (AP). Additionally, we report the average precision for small, medium, and large objects to assess our method's performance in detecting small objects. Efficiency is represented using GFLOPs and latency.

Implementation Details Using PyTorch and MMDetection, we trained one-stage models from scratch on the VisDrone and UAVDT datasets for 300 epochs, with a learning rate of 1e-2, and applied data augmentation techniques such

Model	imgsz	test	AP_{95}^{val}	AP_{50}^{val}	AP_{75}^{val}	AP_{s}^{val}	AP_{m}^{val}	AP_{l}^{val}	Latency(ms)	FLOPs(G)
YOLOv6-v3.0-N	640	o	19.0	32.8	18.7	9.9	29.0	41.3	11.9	3.9
YOLOv8-N	640	o	19.1	33.0	18.9	10.6	28.9	38.3	4.3	4.1
YOLOv7-Tiny	640	o	19.4	35.1	18.5	10.5	29.1	41.0	3.8	4.2
RTMDet-Tiny	640	o	20.3	33.5	21.2	10.2	32.9	**47.1**	13.2	5.1
RemDet-Tiny	640	o	**21.8**	**37.1**	**21.9**	**12.7**	**33.0**	44.5	3.4	4.6
QueryDet	800	o	19.6	35.7	19.0	-	-	-	288	-
RetinaNet	800	o	20.2	36.9	19.5	-	-	-	14.7	210
Faster-RCNN	800	o	21.4	40.7	19.9	11.7	33.9	54.7	21.2	285
RTMDet-L	640	o	23.7	37.4	25.5	12.5	38.7	50.4	13.9	50.4
CenterNet	800	o	27.8	47.9	27.6	21.3	42.1	49.8	95.2	1855
HRDNet	1333	o	28.3	49.3	28.2	-	-	-	-	421
GFLV1	1333	o	28.4	50.0	27.8	-	-	-	525	-
CEASC	1333	o	28.7	**50.7**	28.4	-	-	-	43.8	150
RemDet-L	640	o	29.3	47.4	30.3	18.7	43.4	55.8	7.1	67.4
RemDet-X	640	o	**29.9**	48.3	**31.0**	**19.5**	**44.1**	**58.6**	8.9	114
ClusDet	1000	o+ca	26.7	50.6	24.7	17.6	38.9	51.4	273	-
DMNet	1500	o+ca	28.2	47.6	28.9	19.9	39.6	55.8	290	-
CDMNet	1000	ca	29.2	49.5	29.8	20.8	40.7	41.6	-	-
GLASN	600	o+ca	30.7	55.4	30.0	-	-	-	-	-
AMRNet	1500	o+aug	31.7	-	-	23.0	43.4	**58.1**	-	-
YOLC	1024	o+ca	31.8	55.0	31.7	24.7	42.3	45.0	441	151
CZDet	1200	o+dc	33.2	58.3	33.2	26.0	42.6	43.4	-	-
UFPMP-Det	1333	o+ca	36.6	**62.4**	36.7	-	-	-	152	205
RemDet-X	1024	o+ca	**40.0**	61.9	**42.8**	**30.4**	**52.5**	54.6	9.0	182

Table 1: Comparison in terms of AP (%), Latency, and FLOPs on VisDrone. o, ca, aug respectively stand for the original validation set, cluster-aware cropped images, and augmented images. "-" indicates that the result is not reported.

as mixup and Mosaic. For two-stage models, we utilized pretrained backbone networks. On MSCOCO, we kept the same parameters, except for a momentum of 0.937, a weight decay of 5e-4, and a learning rate decay of 1e-2 every 10 epochs. The input size for the YOLO series models was 640×640, while for other models it was 1,333×800. All experiments were conducted on 8 NVIDIA RTX 4090 GPUs, with inference performed on a single 4090 GPU.

Comparison with SOTA on UAV Datasets

Comparison Results on VisDrone Our proposed model demonstrates significant improvements over existing models in terms of the key evaluation metric, mean Average Precision (mAP), on the VisDrone dataset. Additionally, to emphasize real-time performance, we compare our model with real-time general object detectors. Notably, the field of real-time lightweight UAV detection lacks comprehensive research. Our smallest model, RemDet-Tiny, achieves outstanding performance with an inference speed of 3.4 ms, surpassing the baseline (Jocher, Chaurasia, and Qiu 2023) by 2.7%.

When trained on the original validation set, as shown in Table 1, our model outperforms the previous state-of-the-art lightweight model, CEASC (Du et al. 2023), by 0.8%, while reducing computational complexity by 35%. Compared to QueryDet (Yang, Huang, and Wang 2022), our model achieves a 9.6% improvement. Furthermore, after incorporating the Cluster-Aware Crops method, our model aligns input image sizes with YOLC, achieving the best performance to date with an mAP of 40%. This represents an 8.2% improvement over YOLC and a 6.8% improvement over CZDet. Remarkably, on a single 4090 GPU (without any model acceleration techniques), our model achieves a 9 ms inference speed, underscoring the effectiveness of our detector design.

However, our model shows a slight disadvantage in detecting large objects compared to other models. We attribute this to the intricate and heavily handcrafted designs used in competing methods, which can effectively detect large objects in UAV images. In contrast, our simpler design prioritizes high-speed inference capability and strong generalization.

Model	AP_{95}^{val}	AP_{50}^{val}	AP_{75}^{val}	AP_{s}^{val}	AP_{m}^{val}	AP_{l}^{val}
R-FCN	7.0	17.5	3.9	4.4	14.7	12.1
FRCNN+FPN	11.0	23.4	8.4	8.1	20.2	26.5
CenterNet	13.2	26.7	11.8	7.8	26.6	13.9
ClusDet	13.7	26.5	12.5	9.1	25.1	31.2
DMNet	14.7	24.6	16.3	9.3	26.2	35.2
CDMNet	16.8	29.1	18.5	11.9	29.0	15.7
GLSAN	17.0	28.1	18.8	-	-	-
CEASC	17.1	30.9	17.8	-	-	-
AMRNet	18.2	30.4	19.8	10.3	31.3	33.5
YOLC	19.3	30.9	20.1	10.9	**32.2**	**35.5**
RemDet-L†	**20.6**	**34.5**	**22.1**	**13.9**	31.4	30.3

Table 2: Comparison in terms of AP (%) on UAVDT. † represents the use of cluster-aware cropped method.

Comparison Results on UAVDT Based on our experimental results using the UAVDT dataset (as shown in Table 2), we have demonstrated that the proposed method outperforms the current state-of-the-art model, YOLC, achieving the highest performance (20.6%). Additionally, compared to other methods, our approach significantly improves the accuracy of small object detection by 3%, although it performs worse

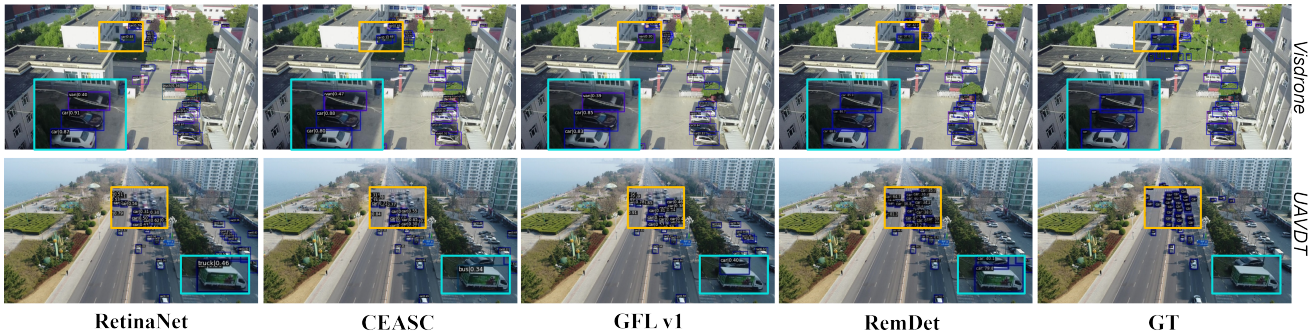


Figure 6: Visualization of the results of RetinaNet, CEASC, GFL and RemDet on the VisDrone and UAVDT datasets.

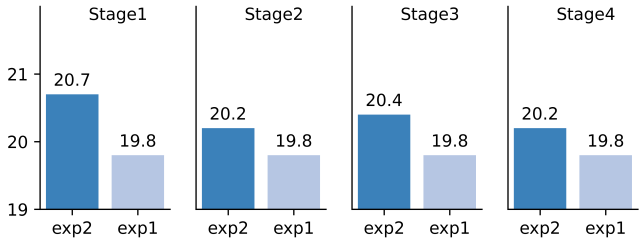


Figure 7: Ablation for each stage using different expansions.

for large objects. Furthermore, our methods can be combined with these handcrafted designs to obtain performance gains, even though it may introduce additional inference overhead.

Ablation Study

Ablation of Overall Design In our research, the block ratios for each stage are (3:6:6:3). This ratio was determined through neural architecture search (NAS) applied to the MSCOCO dataset. However, when dealing with UAV images, we aim to find the optimal block ratio by removing redundant modules for lightweight models.

To achieve this, we initially set all stage blocks to 3 and progressively increased them to 6. The experimental results are detailed in Appendix. Finally, We adopted a (3:3:6:3) block ratio, eliminating unnecessary blocks. In our CED module, the original channel expansion ratio was 1. Increasing it to 2 would enhance model performance, but at the cost of higher computation and latency. Thus, we aimed for a balance to maximize channel information expansion. Notably, channel expansion significantly improved stage 1 performance (Figure 7), leading us to remove it from other stages. This operation reduced model inference time to 3.7 ms, with only a 0.1% performance drop. For detailed exploration of ours models, refer to Figure 3.

More Ablation Experiments To validate our methods, we combine RemDet's backbone with various detectors, as shown in Table 3. Surprisingly, when combined with other detectors, our method consistently improves detection accuracy. Furthermore, we trained RemDet on the MSCOCO dataset (as shown in Table 4). In comparison to all minimal versions of generic detectors, our approach provides the most SOTA

Base Detector	Backbone	AP^{val}_{95}	AP^{val}_{50}	AP^{val}_{s}	AP^{val}_{m}	AP^{val}_{l}
YOLOv5-S	CSPDarknet	18.6	32.9	10.5	27.9	38.5
	RemDet	**20.7**	**36.2**	**12.1**	**31.1**	**40.7**
RTMDet	CSPNeXt	21.8	35.2	11.5	35.2	49.5
	RemDet	**22.8**	**35.8**	11.4	**35.5**	48.8
YOLOv8-S	CSPDarknet	23.1	38.7	13.9	34.8	42.3
	RemDet	**24.6**	**41.3**	**15.0**	**36.5**	**46.7**
FasterRcnn (1×)	ResNet50	16.3	29.5	9.1	25.6	27.3
	RemDet	**19.0**	**34.2**	**11.6**	**29.1**	**30.6**
RetinaNet (100e)	ResNet50	8.0	14.6	3.3	14.3	18.4
	RemDet	**15.9**	**27.7**	**7.1**	**26.8**	**35.0**
DyHead (2×)	ResNet50	13.8	24.6	7.1	22.5	25.5
	RemDet	**17.2**	**29.1**	**9.6**	**27.5**	**31.7**

Table 3: Comparing AP(%) and GFLOPs/Param of various detectors on VisDrone with our approach.

Model	AP^{val}_{95}	AP^{val}_{50}	AP^{val}_{s}	AP^{val}_{m}	AP^{val}_{l}	Param	FLOPs
YOLOv6-3.0-N	36.2	51.6	16.8	40.2	52.6	4.3M	5.5G
YOLOv8-N	37.3	52.6	18.8	41.0	53.5	3.0M	4.1G
YOLOv7-Tiny	37.5	55.8	19.9	41.1	50.8	5.2M	6.2G
RemDet-Tiny	**39.5**	**55.8**	**21.0**	**43.9**	**54.0**	3.2M	4.6G
YOLO-MS-XS	43.1	60.1	24.0	47.8	59.1	4.5M	8.7G
YOLOv6-v3.0-S	43.7	60.8	23.6	48.7	59.8	17.2M	21.9G
YOLOv8-S	44.9	61.8	26.0	49.9	**61.0**	11.1M	14.3G
RemDet-S	**45.5**	**62.8**	**27.8**	**50.5**	60.0	11.9M	16.0G

Table 4: Comparison of AP(%) and params/FLOPs with the real-time approaches on MSCOCO.

results, surpassing our baseline by 2.3% on the COCO. We believe that our sufficiently simple structure contributes to powerful generalization, while also significantly enhancing small object detection accuracy on generic datasets.

Conclusion

In this paper, we present RemDet, a novel UAV object detector with a focus on small and dense objects in UAV imagery. Our approach involves designing modules that enhance inter-layer information while mitigating information loss. To meet real-time requirements, we explore cost-efficient multiplication operations, adopt reparameterization, and remove unnecessary components. Our experiments show that RemDet achieves state-of-the-art results and high-speed inference. However, further enhancements are needed for large objects. We hope this work inspires future research in UAV detectors.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62376252); Key Project of Natural Science Foundation of Zhejiang Province (LZ22F030003); Zhejiang Province Leading Geese Plan(2024C02G1123882).

References

Ashraf, M. W.; Sultani, W.; and Shah, M. 2021. Dogfight: Detecting drones from drones videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7067–7076.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, 213–229. Springer.

Dauphin, Y. N.; Fan, A.; Auli, M.; and Grangier, D. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, 933–941. PMLR.

Ding, J.; Xue, N.; Long, Y.; Xia, G.-S.; and Lu, Q. 2018. Learning RoI transformer for detecting oriented objects in aerial images. *arXiv preprint arXiv:1812.00155*.

Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; and Sun, J. 2021. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13733–13742.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Du, B.; Huang, Y.; Chen, J.; and Huang, D. 2023. Adaptive sparse convolutional networks with global context enhancement for faster object detection on drone images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13435–13444.

Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; and Tian, Q. 2018. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, 370–386.

Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; and Tian, Q. 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6569–6578.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Han, D.; Yun, S.; Heo, B.; and Yoo, Y. 2021. Rethinking channel dimensions for efficient model design. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 732–741.

Hollard, L.; Mohimont, L.; Gaveau, N.; and Steffenel, L.-A. 2024. LeYOLO, New Scalable and Efficient CNN Architecture for Object Detection. *arXiv e-prints*, arXiv:2406.14239.

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1314–1324.

Huang, Y.; Chen, J.; and Huang, D. 2022. UFPMP-Det: Toward accurate and efficient object detection on drone imagery. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 1026–1033.

Jocher, G. 2020. YOLOv5 by Ultralytics.

Jocher, G.; Chaurasia, A.; and Qiu, J. 2023. Ultralytics YOLO.

Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. 2022. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, C.; Gao, G.; Huang, Z.; Hu, Z.; Liu, Q.; and Wang, Y. 2024. YOLC: You Only Look Clusters for Tiny Object Detection in Aerial Images. *IEEE Transactions on Intelligent Transportation Systems*.

Liu, X.; Peng, H.; Zheng, N.; Yang, Y.; Hu, H.; and Yuan, Y. 2023. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14420–14430.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11976–11986.

Meethal, A.; Granger, E.; and Pedersoli, M. 2023. Cascaded Zoom-in Detector for High Resolution Aerial Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2045–2054.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Redmon, J.; and Farhadi, A. 2017. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263–7271.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Shwartz-Ziv, R.; and Tishby, N. 2017. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

Tishby, N.; and Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, 1–5. IEEE.

Wang, A.; Chen, H.; Lin, Z.; Han, J.; and Ding, G. 2024. Repvit: Revisiting mobile cnn from vit perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15909–15920.

Wang, C.-Y.; Bochkovskiy, A.; and Liao, H.-Y. M. 2023. 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*, 7464–7475.

Wang, C.-Y.; Yeh, I.-H.; and Liao, H.-Y. M. 2024. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv preprint arXiv:2402.13616*.

Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; and Hu, Q. 2020. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11534–11542.

Wei, Z.; Duan, C.; Song, X.; Tian, Y.; and Wang, H. 2020. Amrnet: Chips augmentation in aerial images object detection. *arXiv preprint arXiv:2009.07168*.

Yang, C.; Huang, Z.; and Wang, N. 2022. QueryDet: Cascaded sparse query for accelerating high-resolution small object detection. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 13668–13677.

Yang, F.; Fan, H.; Chu, P.; Blasch, E.; and Ling, H. 2019. Clustered object detection in aerial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8311–8320.

Ye, H.; Zhang, B.; Chen, T.; Fan, J.; and Wang, B. 2023. Performance-aware approximation of global channel pruning for multitask cnns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 10267–10284.

Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; and Ling, H. 2021. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7380–7399.