Deep Reinforcement Learning-Based Target Detection and Autonomous Obstacle Avoidance Control for UAV

Like Zhao*, Hao Liu, Guangmin Gu, Fei Wan, Yanyang Feng
Inner Mongolia Power (Group) Co, Ltd, Nei Mongol 017000, China

***Abstract*—To address the challenges faced by distribution network monitoring systems—such as significant variations in anomaly scale, frequent missed and false detections of small-scale faults, and the need for real-time operational control—this paper proposes a lightweight multi-scale feature fusion detection network combined with a deep reinforcement learning-based autonomous control strategy, forming an end-to-end intelligent perception and decision-making system for distribution networks. To enhance detection accuracy and computational efficiency, a lightweight feature fusion network (Grid_RepGFPN) is designed, and a novel feature fusion module (DBB_GELAN) is proposed, which significantly reduces model parameters and computational cost while improving detection performance. Additionally, a feature extraction module (FTA_C2f) is constructed using partial convolution (PConv) and triplet attention mechanisms, combined with the ADown downsampling structure to improve the model's capability to capture spatial and electrical measurement details. The programmable gradient information (PGI) strategy of YOLOv9 is further optimized by introducing a context-guided reversible architecture and a Grid_PGI method with additional detection heads, thereby enhancing deep supervision stability and reducing semantic information loss. Based on the detection model, a real-time operational control strategy is developed using deep reinforcement learning, enabling autonomous fault response, load adjustment, and network optimization through a state–action–feedback optimization loop. Experimental results on multiple distribution network simulation platforms demonstrate that the proposed LMGrid-YOLOv8 model outperforms YOLOv8s, with improvements of 4.2%, 3.9%, 5.1%, and 3.0% in precision, recall, mAP@0.5, and mAP@0.5:0.95, respectively, while reducing parameters by 63.9% and increasing computation by only 0.4 GFLOPs, achieving a favorable balance between performance and resource consumption. Inference experiments on edge computing platforms confirm that the proposed model maintains high detection accuracy under real-time constraints, demonstrating strong applicability to real-time distribution network monitoring. Furthermore, class activation map-based visual analysis reveals the model's superior capabilities in detecting small-scale faults and processing high-resolution network measurement regions.**

Keywords*—Target detection; multi-scale; lightweight; YOLOv8; autonomous obstacle avoidance

I. Introduction

Low-altitude unmanned aerial To address the challenges faced by distribution network monitoring systems—such as significant variations in anomaly scale, frequent missed and false detections of small-scale faults, and the need for real-time operational control—this paper proposes a lightweight multi-scale feature fusion detection network combined with a deep reinforcement learning-based autonomous control strategy, forming an end-to-end intelligent perception and decision-making system for distribution networks. To enhance detection accuracy and computational efficiency, a lightweight feature fusion network (Grid_RepGFPN) is designed, and a novel feature fusion module (DBB_GELAN) is proposed, which significantly reduces model parameters and computational cost while improving detection performance. Additionally, a feature extraction module (FTA_C2f) is constructed using partial convolution (PConv) and triplet attention mechanisms, combined with the ADown downsampling structure to improve the model's capability to capture spatial and electrical measurement details. The programmable gradient information (PGI) strategy of YOLOv9 is further optimized by introducing a context-guided reversible architecture and a Grid_PGI method with additional detection heads, thereby enhancing deep supervision stability and reducing semantic information loss. Based on the detection model, a real-time operational control strategy is developed using deep reinforcement learning, enabling autonomous fault response, load adjustment, and network optimization through a state–action–feedback optimization loop. Experimental results on multiple distribution network simulation platforms demonstrate that the proposed LMGrid-YOLOv8 model outperforms YOLOv8s, with improvements of 4.2%, 3.9%, 5.1%, and 3.0% in precision, recall, mAP@0.5, and mAP@0.5:0.95, respectively, while reducing parameters by 63.9% and increasing computation by only 0.4 GFLOPs, achieving a favorable balance between performance and resource consumption. Inference experiments on edge computing platforms confirm that the proposed model maintains high detection accuracy under real-time constraints, demonstrating strong applicability to real-time distribution network monitoring. Furthermore, class activation map-based visual analysis reveals the model's superior capabilities in detecting small-scale faults and processing high-resolution network measurement regions. (UAVs), characterized by their wide field of view, high maneuverability, and minimal geographical constraints, have been widely applied in various fields such as management, power inspection, remote sensing, geospatial surveying, emergency rescue, and agricultural monitoring [1–4]. For example, in the "UAVs-road-cloud" integrated intelligent transportation architecture, real-time detection and recognition of targets based on UAV aerial

imagery provides essential foundational data and decision support for flow monitoring, conflict prediction, and collision risk warning, making it a key technology for urban intelligent connected UAVs safety monitoring platforms [5]. However, UAV-based visual target detection in complex road scenarios still faces several challenges. On one hand, UAV imagery often features small targets, large scenes, diverse scales, and occlusions, making accurate detection of specific objects difficult. Existing deep neural network-based detection algorithms have shown satisfactory performance for medium and large-scale targets [6–8], but still fall short in handling small targets in UAV scenes due to limitations in data preprocessing, backbone feature extraction, and high-level feature adaptation, resulting in high rates of missed and false detections. On the other hand, the strong real-time requirements of urban management demand high deployment efficiency and inference speed, while traditional deep learning models tend to suffer from slow inference and low deployment efficiency. Although lightweight networks such as GCL-YOLO [9] and SF-YOLOv5 [10] have been proposed, tests on multiple datasets show that their accuracy and performance gains are limited compared to baseline models.

In terms of small object detection, several recent studies have made progress. Lei Bangjun et al. [11] proposed an improved YOLOv8s algorithm by enhancing the detection head and fusing shallow and deep features to boost the perception and capture of small targets, constructing novel F_C2f_EMA and SM_SPPCSPC modules. Pan Wei et al. [12] introduced an improved YOLOv8s model integrating multiple attention mechanisms, including receptive field attention convolution and CBAM [13], as well as separable convolution-based attention for the pyramid pooling layer to enhance cross-layer feature interaction. In lightweight visual detection, many efforts have also emerged. Wang et al. [14] proposed UAV-YOLOv8, incorporating the BiFormer [15] attention mechanism into the backbone, designing the FFNB lightweight feature module, and introducing new detection scales based on this module and PAFPN. Li et al. [16] addressed the common problem of missed and false detections of small targets in aerial images by improving the Neck with Bi-FPN and replacing part of the C2f modules with GhostBlockV2 based on GhostConv [17], thereby suppressing information loss during long-distance feature propagation and significantly reducing parameter count. Li Zixuan et al. [18] proposed the F-GFPN fusion module, enhancing feature interaction via jump and cross-scale connections based on Efficient-RepGFPN [19], effectively improving detection performance for small targets like rebar tie points.

Although these methods have optimized the backbone and neck networks for aerial object detection to some extent, they still struggle to balance detection performance and resource consumption. Feature fusion quality in the neck remains inadequate, and information loss during network propagation persists. This paper proposes a lightweight multi-scale object detection algorithm, LMUAV-YOLOv8, by improving the YOLOv8 baseline model and optimizing the programmable gradient information (PGI) strategy [20]. First, a lightweight and efficient feature fusion network (UAV_RepGFPN) is designed, which retains more shallow features through optimized fusion paths, generates additional shallow feature maps using ghost convolution, and enriches the feature space with the DBB_GELAN module in the deep network. Second, a new feature extraction module (FTA_C2f) is constructed using partial convolution (PConv) [21] and triplet attention (TA) [22], combined with the ADown [20] downsampling module to enhance the deep network's spatial feature extraction capability. Then, a context-guided reversible architecture [23] is introduced and optimized within the PGI auxiliary branch, generating an additional P2 detection head while removing the P5 head to avoid semantic information loss caused by multi-path feature aggregation in traditional deep supervision. This improves detection accuracy during inference without increasing parameter count or computational cost.

Ablation and comparison experiments on the VisDrone2019 test set verify the effectiveness and superiority of the proposed algorithm. Moreover, inference results on the NVIDIA Jetson Xavier NX embedded platform show that, compared to the baseline model, the proposed method achieves higher detection accuracy while meeting real-time requirements, demonstrating strong applicability in UAV-based real-time detection scenarios. Finally, class activation map-based visualizations are used to analyze the model's decision-making process during inference, providing insights into the underlying mechanism of the proposed algorithm.

II. Multiscale YOLOv8 Improved Model for Lightweighting

The final improved network model is referred to as LMUAV-YOLOv8 in this paper, and its overall architecture is illustrated in Fig. 1. The improved model reconstructs the feature extraction and feature fusion networks and incorporates a context-guided auxiliary reversible branch.

A. Cost of Track Length

In the YOLO series of object detection models, shallow feature maps contain rich high-resolution details, which are particularly beneficial for detecting densely distributed small-scale objects. To address the low detection accuracy of PANet and the high latency issues in GFPN-based models, Xu et al. proposed Efficient-RepGFPN, which facilitates comprehensive information exchange between high-level semantic features and low-level spatial detail maps through an improved queen-fusion mechanism and an enhanced CSPNet [24]. The modified CSPNet incorporates a re-parameterization mechanism and the connection design from the Efficient Layer Aggregation Network (ELAN) [25].

YOLOv8, as a lightweight object detection network, emphasizes reducing computational cost and parameter count while maintaining high detection accuracy. The design philosophy of Efficient-RepGFPN aligns with this goal by enabling efficient feature extraction and fusion without introducing excessive computational overhead. Ablation experiment results demonstrate that incorporating Efficient-RepGFPN leads to a reduction in both parameter count and FLOPs, with an improvement in accuracy, further enhancing the neck network's capability for multi-scale feature fusion. Compared to PANet, Efficient-RepGFPN also offers greater

optimization potential, confirming its effectiveness in multi-scale fusion within YOLOv8.

Moreover, YOLOv8 requires strong feature extraction capabilities to accurately detect both small and large objects across various scales. Efficient-RepGFPN achieves this through efficient path enhancement and feature fusion mechanisms that extract and integrate multi-scale features without increasing model complexity, which is critical for improving YOLOv8's performance in complex scenes.

Lastly, YOLOv8 is designed with modularity in mind, allowing for easy adaptation and expansion according to different application requirements. Efficient-RepGFPN is also a modular design, allowing selective inclusion or removal of sub-modules as needed. This flexibility enables Efficient-RepGFPN to be adapted to various YOLOv8 versions or configurations, offering broader application potential across different scenarios.

However, since Efficient-RepGFPN is a general-purpose network aimed at detecting standard object types, it is not specifically optimized to exploit the rich information in shallow feature maps, which still maintain high resolution and may not be well-suited for general object detection tasks. To address this limitation, this paper proposes an improved feature fusion strategy based on the Efficient-RepGFPN architecture, named UAV_RepGFPN (as shown in Fig. 2). UAV_RepGFPN integrates Efficient-RepGFPN into the neck of the YOLOv8 architecture and applies lightweight multi-scale enhancements. The proposed network enhances the fusion of multi-scale information, thereby improving small object detection performance and increasing the efficiency of spatial information retention. Additionally, the network structure is lightweighted to minimize computational overhead while reducing the loss of high-resolution features. The main improvements of this architecture are detailed.

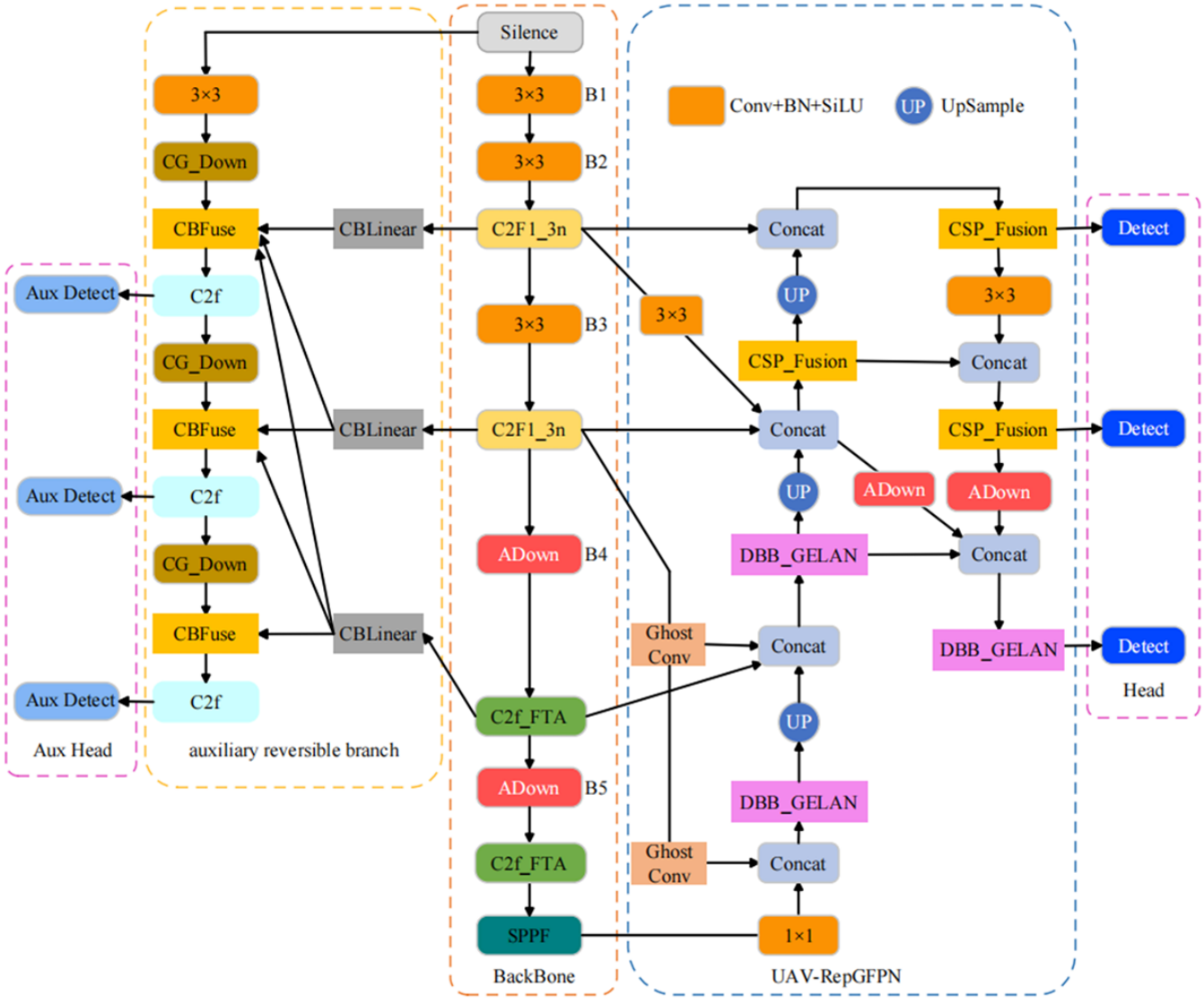


Fig. 1. LMUAV-YOLOv8 algorithm structure.

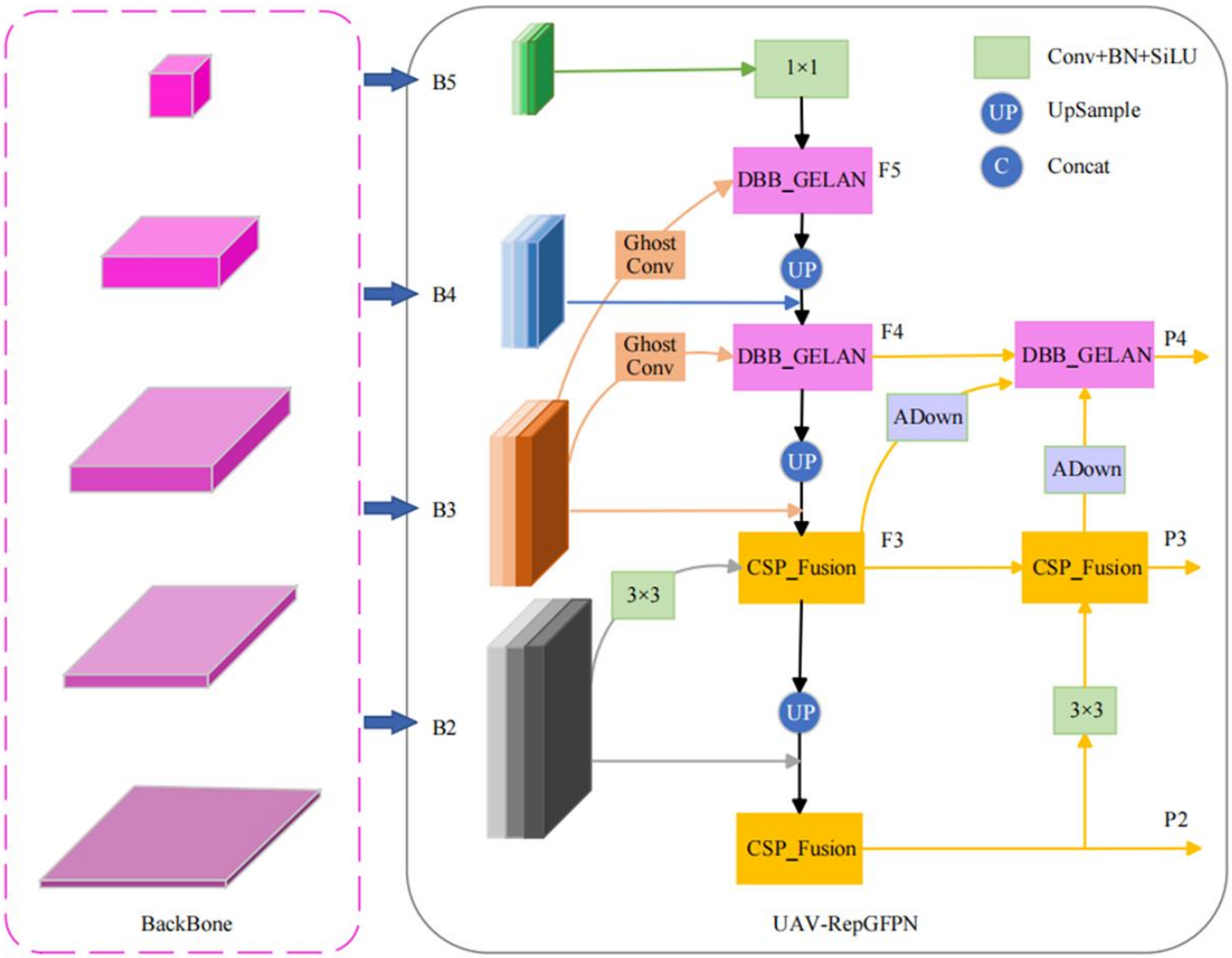


Fig. 2. UAV_RepGFPN algorithm structure.

The original detection head architecture of Efficient-RepGFPN includes three output layers, each corresponding to large, medium, and small objects, with strides of 32, 16, and 8, respectively. Assuming an input resolution of 640×640 pixels, these output layers generate feature maps of sizes 20×20, 40×40, and 80×80, respectively. In the 80×80 feature map, each pixel corresponds to an 8×8 region in the original image. However, this resolution may be insufficient for accurately detecting very small objects. A 160×160 feature map would be more suitable for such targets. Therefore, based on Efficient-RepGFPN, this paper adds a new output layer with a downsampling rate of 4 (i.e., stride 4) as a small-object detection head to enhance the model's ability to extract features from small targets and improve multi-scale feature fusion.

As shown in Fig. 3, an additional 160×160 small-object detection layer (P2) is introduced in the Head section of Efficient-RepGFPN. The B3 feature map from the backbone, a downsampled version of the B2 feature map, and an upsampled version of the F4 feature map are combined as the input for the F3 layer. After processing, this layer generates feature maps rich in small-object information. The output from the F3 layer is then upsampled and concatenated with the B2 layer's 160×160 feature map along the channel dimension. This enhances the representational capacity of the fused 160×160 feature map for small objects and increases the network's sensitivity to small targets. Finally, the CSP_Fusion module outputs the P2 detection head specifically for small-object detection. Meanwhile, during the top-down information flow in PANet, P2 can transmit location information to feature layers at other scales, thereby improving multi-scale feature fusion and enhancing small-object detection accuracy.

In addition, considering the limited presence of large-scale objects in aerial imagery, the detection head P5 with a downsampling rate of 32 and its corresponding layers in the Neck are removed, as indicated in gray in Fig. 3. According to the ablation experiments on the proposed feature fusion network, removing the P5 detection head results in no change in mAP\@0.5, while reducing the model's parameter count and computational cost by 13% and 4%, respectively. Therefore, eliminating the P5 head helps reduce model complexity without compromising performance. Since low-altitude UAV imagery typically contains densely distributed

small objects and relatively few large, easily identifiable objects, the removal of the P5 detection layer does not negatively impact detection performance and does not conflict with the addition of the P2 detection head.

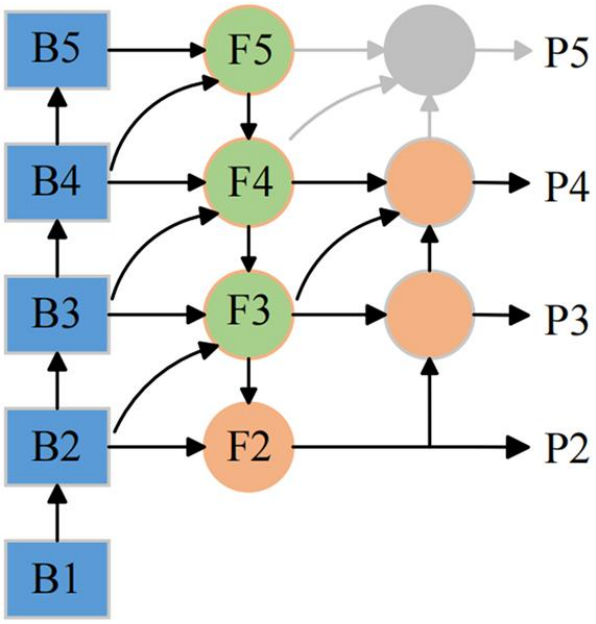


Fig. 3. Improved detection head.

"Queen-fusion" is a novel inter-feature-layer communication strategy that enables cross-scale feature fusion through 3×3 convolutions. It not only receives input from the previous layer but also aggregates feature inputs from neighboring layers in all directions. This design minimizes information loss during the feature fusion process.

In Efficient-RepGFPN, the F5 layer typically receives inputs from the B5 feature map processed by a 1×1 convolution and the B4 feature map after downsampling. However, considering that the shallow B3 layer contains more spatial and positional information than B4, this paper replaces the downsampled B4 with a twice-downsampled B3 in the F5 layer. This modified B3 input is then concatenated and fused with the 1×1 convolved B5 feature map, as indicated by the red lines in Fig. 4. This approach not only enhances the model's detection accuracy for small objects but also effectively reduces the number of parameters in the model.

Ghost Convolution (GhostConv), proposed by Han et al. [17], is an efficient convolutional operation that decomposes the features generated by traditional costly convolutions into primary features and redundant features. The redundant features, also known as ghost feature maps, are derived from the primary features through a series of inexpensive linear transformations. This approach significantly reduces both the number of parameters and computational overhead while preserving the spatial information of the original feature maps.

As illustrated in Fig. 5, the feature extraction process in GhostConv begins by applying a 1×1 convolution to an input feature map of size $H\times W\times C$, resulting in a primary feature map of size $H\times W\times C/2$. Next, a 5×5 depthwise separable convolution is applied to expand the receptive field and generate ghost feature maps of size $C/2$. Notably, depthwise convolution processes each input channel separately with its own filter, rather than applying all filters across all channels. Finally, the ghost feature maps are concatenated with the primary feature map to reconstruct and reuse the original features. This process effectively mitigates the loss of spatial and positional information, which is crucial for small object detection in feature fusion networks.

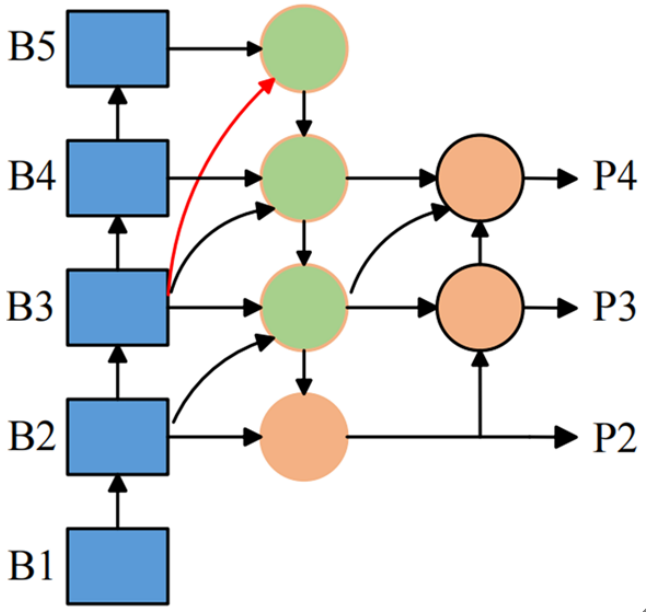


Fig. 4. Improved feature fusion paths.

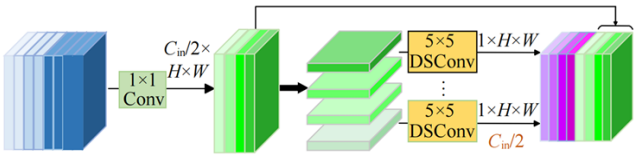


Fig. 5. Structure of GhostConv.

Inspired by this mechanism, and to address the degradation of shallow features—beneficial for small object detection—caused by downsampling in deeper layers, this paper replaces the traditional downsampling modules in the F4 and F5 layers of the Efficient-RepGFPN with GhostConv. By generating more ghost features from shallow features through inexpensive linear operations, the model achieves improved detection accuracy for small objects. Ablation experiments show that after replacing the traditional downsampling modules in F4 and F5 with GhostConv, the model exhibits enhancements in mAP, parameter efficiency, and computational cost. These results indicate that the linear operations of GhostConv help the deep network retain shallow feature information more effectively, making it a practical choice for the current application scenario.

The Generalized Efficient Layer Aggregation Network (GELAN) is a lightweight feature fusion module in YOLOv9 that combines the gradient path planning module CSPNet with ELAN. The GELAN module fuses features from different layers, enabling the network to better capture multi-scale information of targets. At the same time, it is designed with

computational efficiency in mind, ensuring enhanced feature fusion capability while reducing computational cost. This module offers a favorable balance between accuracy and computational overhead.

The Diverse Branch Block (DBB), proposed by Ding et al. [26], aims to improve the performance of CNNs by introducing a multi-branch architecture, while ensuring no additional inference time through structural re-parameterization, as illustrated in Fig. 6. The DBB module allows the use of different kernel sizes (1×1, K×K, and average pooling) within the same convolutional layer. It increases feature space richness through multi-branch paths. This operation, featuring different receptive fields and complexity paths, enriches the feature space and is particularly important for feature fusion networks that need to extract and aggregate hierarchical information.

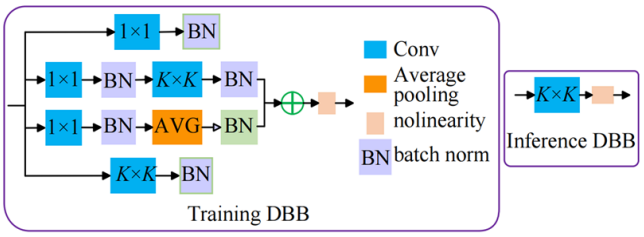


Fig. 6. Structure of diverse branch block.

In this paper, DBB is adopted as the convolutional block within the GELAN structure, forming a new multi-branch-based efficient layer aggregation module called DBB_GELAN, as shown in Fig. 7. The feature fusion network primarily integrates shallow spatial information with deep semantic information, and in small object detection, shallow features are more critical. As shown in Fig. 2, UAV_RepGFPN employs DBB_GELAN as the feature fusion module for the neck network layers F4, F5, and P4. This not only introduces a multi-branch structure into the original mechanism, enriching the feature space and enhancing feature fusion, but also effectively reduces the number of parameters and computational cost.

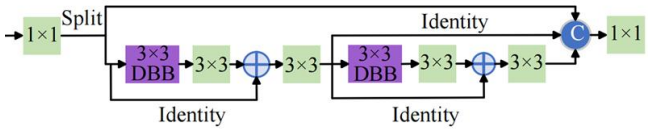


Fig. 7. Structure of DBB_GELAN.

B. Backbone Network

ADown is a lightweight downsampling module proposed by Wang et al. [20]. This module replaces traditional convolutional downsampling with pooling operations and uses a branching design to ensure the final feature map contains both the original feature information and additional information obtained through different processing paths, thereby enhancing the model's feature representation capability. As shown in Fig. 8, the branches containing MaxPooling and a CBS module with a 3×3 convolution kernel are referred to as the 1×1 branch and 3×3 branch, respectively. First, the input feature map X undergoes average pooling with a 2×2 kernel, stride 1, and no padding to reduce the feature map size. The pooled feature map is then split into two parts: X_1 and X_2. The X_1 feature map passes through the 3×3 branch, adjusting the number of channels to $C_{out}/2$, where C_{out} is the number of output channels. The X_2 feature map passes through the 1×1 branch and is max-pooled to achieve lightweight downsampling while preserving key information. Finally, the outputs of the two branches are concatenated along the channel dimension. In this paper, the lightweight downsampling module ADown is applied to the B4 and B5 layers of the backbone network as well as the deep downsampling modules in the feature fusion network. Ablation experiments show that this improvement effectively reduces the model's parameter count and computational cost while improving accuracy.

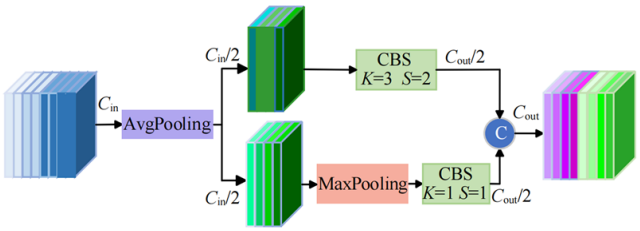


Fig. 8. Structure of ADown.

The C2f module integrates high-level features and contextual information using residual connections to obtain richer gradient flow; however, it also introduces additional computational cost. Moreover, as the number of layers in the feature extraction network increases, while the network enhances its understanding of complex semantic information, the C2f module tends to weaken the network's ability to capture spatial positional information—a phenomenon particularly evident in small object detection tasks. Finally, because the C2f module does not establish interdependencies between channels or spatial positions, it remains insufficient in addressing challenges such as target scale variation and complex backgrounds in UAV aerial images. To overcome these issues, this paper proposes a new feature fusion module named FTA_C2f (see Fig. 9).

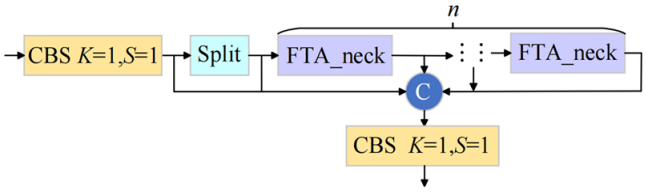


Fig. 9. Structure of FTA_C2f.

The Faster Block in FasterNet [21] is a lightweight feature extraction module with several advantages for improving small object detection performance. First, in the traditional convolution operation within C2f, the convolution kernel is applied to all channels of the input feature map. However, in Faster Block, Partial Convolution (PConv) selects only a portion of consecutive channels for convolution to extract features. This characteristic helps preserve more complete high-resolution information from shallow feature maps, which is beneficial for detecting densely distributed small objects. Second, PConv significantly reduces computation and

memory access by performing regular convolution on only the front or rear consecutive channels while leaving the rest untouched. For example, when the partial convolution ratio is 1/4, the computation cost is only 1/16 of that of regular convolution. Similarly, memory access is reduced to one-fourth of that of regular convolution. Inspired by the characteristics of Faster Block, this paper designs the FTA_neck module shown in Fig. 10. FTA_neck is similar to Faster Block but differs by incorporating a Triplet Attention (TA) mechanism to handle densely packed small objects in complex backgrounds.

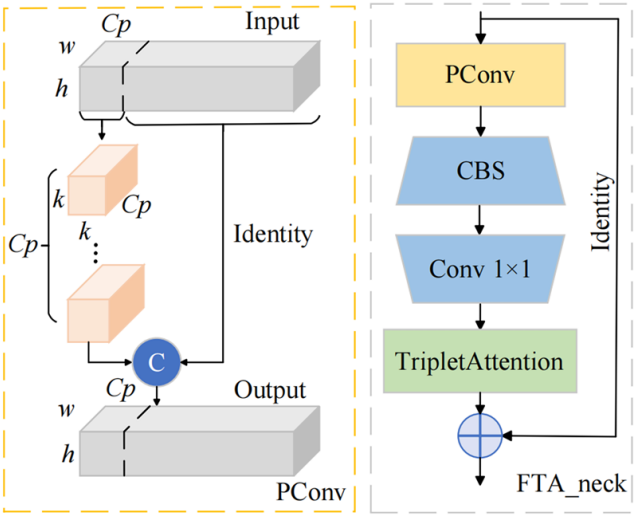


Fig. 10. Structure of FTA_neck.

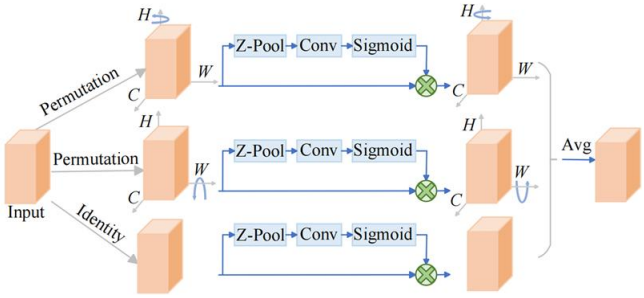


Fig. 11. Structure of TripletAttention.

As shown in Fig. 11, Triplet Attention (TA) adopts a three-branch parallel design. The first two branches compute attention weights along the channel dimension (C) and the spatial dimensions (H and W) respectively, achieved through rotation operations and residual transformations with low computational cost. The bottom branch is similar to CBAM and is used to capture spatial dependencies (H and W). After processing all three branches, the output tensors of each branch are aggregated by simple averaging to form the final triplet attention output. The introduction of TA enables FTA_neck to efficiently encode both channel and spatial information with minimal computational overhead. By capturing the relationships between spatial and channel dimensions, FTA_neck enhances detailed representation, which is critical for small object detection. By integrating spatial and channel dimensions, the network gains better contextual awareness, helping to accurately identify and localize small targets. This results in more accurate and robust detection of small objects across various scenarios.

In FTA_neck, the PConv module performs convolution on one-quarter of the input channels while the remaining channels pass through an identity mapping. The features extracted by PConv then go through a CBS module, which doubles the number of channels in the feature map. This is followed by a 1×1 convolution to reduce the channel count back to match the input's number of channels, ensuring consistency. The feature map then passes through the Triplet Attention module. Finally, the two parts of the feature map are concatenated via a Concat operation. By combining PConv and Triplet Attention, FTA_C2f achieves fusion of channel and spatial dimension information while avoiding significant loss of spatial information, thus balancing computational cost and feature extraction capability.

Replacing the C2f modules in the B4 and B5 layers of the backbone network with FTA_C2f allows for semantic information extraction while preventing excessive spatial information loss caused by the original feature extraction mechanism. This replacement also reduces parameter count and computational cost, enabling closer interaction and encoding of channel and spatial information while maintaining a low computational burden. Additionally, this paper finds that halving the number of channels in the FTA_C2f module at the B5 layer significantly reduces parameters and computation without degrading detection performance.

C. Context-Based Programmable Gradient Information

To address the problem of information loss in deep networks, Wang et al. [20] proposed Programmable Gradient Information (PGI). PGI coordinates the propagation of gradient information across different semantic levels through auxiliary reversible branches, ensuring that deep features retain task-relevant key information without adding excessive parameter overhead. The design of these auxiliary reversible branches avoids the semantic information loss that can occur in traditional deep supervision due to multi-path feature integration. Since the auxiliary reversible branches can be removed during model inference, they do not increase the computational burden during the inference process. The improved model incorporates PGI-based auxiliary reversible branches by adding auxiliary reversible branches specifically for small object detection and generating three additional auxiliary detection heads (AuxP2, AuxP3, AuxP4), proposing the UAV_PGI programmable gradient method, as shown in Fig. 12. YOLOv9 proposed using GELAN [20] as the reversible function in the PGI auxiliary reversible branch architecture, but directly using GELAN as the reversible function in this model did not yield good results and instead reduced detection accuracy. To address this issue, a new context-guided reversible architecture was designed.

The Context Guided Block (CG_Block) is a key component of CGNet [23], inspired by the human visual system's reliance on contextual information to understand scenes. As shown in Fig. 13, CG_Block includes a local feature extractor ($f_{loc}(*)$), a surrounding context extractor ($f_{sur}(*)$), a joint feature extractor ($f_{joi}(*)$), and a global context extractor ($f_{glo}(*)$). $f_{loc}(*)$ uses a 3×3 depthwise convolution to

extract local features, $f_{sur}(*)$ uses a 3×3 dilated convolution with a dilation rate of 2 to capture surrounding context, $f_{joi}(*)$ combines and processes these two types of features, and $f_{glo}(*)$ obtains global context through global average pooling followed by a multilayer perceptron, weighting the output of $f_{joi}(*)$ to refine the joint features. CG_Block employs residual learning processes such as local residual learning (LRL) and global residual learning (GRL) to enhance complex feature learning and gradient backpropagation. GRL connects the input to $f_{glo}(*)$ to enable higher information flow. The design of CG_Block leverages multi-level contextual information, capturing deep semantic features while preserving shallow spatial information. Using channel-wise convolutions and residual learning, CG_Block significantly reduces parameter count while maintaining good accuracy.

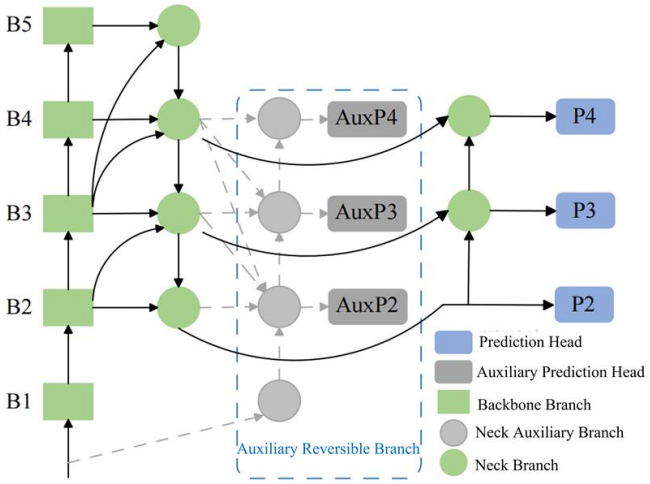


Fig. 12. UAV_PGI structure.

Based on this, this paper proposes the context-guided CG_Down module. As shown in Fig. 14, CG_Down reduces spatial dimensions using a 3×3 convolution with stride 2 while increasing the number of channels. Then, $f_{loc}(*)$ and $f_{sur}(*)$ extract local and contextual information respectively, which are fused by $f_{joi}(*)$. A 1×1 convolution then reduces the number of channels to obtain the joint features. Finally, $f_{glo}(*)$ generates a weighting vector applied to the joint features. The CG_Down module is introduced here as the downsampling module for the auxiliary branch network.

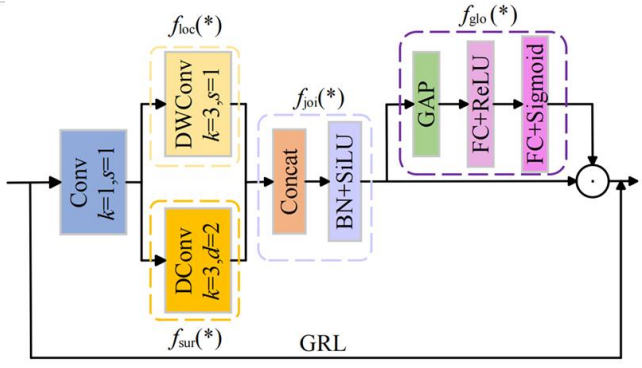


Fig. 13. Structure of context guided block.

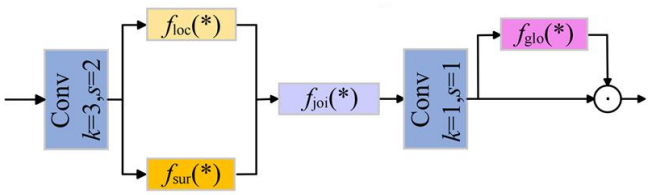


Fig. 14. Structure of CG_Down.

III. Experimental Results and Analysis

A. Data Set

To evaluate the performance of the proposed model in detecting small objects, the VisDrone2019 dataset [27] was selected as the experimental dataset for detailed comparative and ablation studies. This dataset contains 10,209 static aerial images, featuring over 2.6 million object bounding boxes covering common categories such as pedestrians, cars, and bicycles. A notable characteristic of this dataset is the predominance of small objects and class imbalance, providing ideal testing conditions for small-scale object detection.

B. Experimental Environment and Strategy

The model training was conducted using an NVIDIA A40-PCIE-48GB GPU, running on Ubuntu 20.04, with PyTorch version 1.13.1, Python version 3.8, CUDA version 12.4, and YOLOv8s dependent library Ultralytics version 8.0.202. The network was trained for 300 iterations. During training, the optimizer used was Stochastic Gradient Descent (SGD) with a batch size of 4. The initial learning rate was set to 0.01, gradually decreasing to 1% of the initial learning rate by the end of training, and the weight decay coefficient was set to 0.0005. A warm-up period of 5 epochs was applied, adopting the default data augmentation strategy from YOLOv8. Data augmentation was disabled in the last 10 epochs, and early stopping was set to 50 epochs. All input images were resized to 640×640 pixels. Inner-WIoU was used as the loss function, with hyperparameters δ, α, and *ratio* set to 2.7, 1.7, and 1.2 respectively. To ensure fairness of experimental data, no official pretrained weights were loaded during training for any network in the experiments. Considering the performance requirements for drone detection scenarios and the need to balance detection performance with resource consumption, YOLOv8s was selected as the baseline model.

To verify the applicability of the proposed algorithm for real-time UAV target detection scenarios, comparative experiments on detection accuracy and speed were conducted on an embedded platform based on the NVIDIA Jetson Xavier NX (16GB). The main hardware configuration of the embedded platform includes the Jetson Xavier NX Developer Kit, featuring a 384-core NVIDIA Volta GPU (with 48 Tensor cores), a 6-core NVIDIA Carmel ARMv8.2 CPU running at 1.9 GHz, and 16GB of memory. The NVIDIA JetPack SDK version is 5.1.3. The primary software environment consists of Ubuntu 20.04 focal as the operating system, Python version 3.8.10, CUDA version 11.4, TensorRT version 8.5.2.2, PyTorch version 2.1.0, Torchvision version 0.16.2, and Onnxruntime-GPU version 1.17.0.

C. Description of Indicators and Parameters

To intuitively and effectively demonstrate the performance improvements made to YOLOv8 in this study, the following evaluation metrics were selected: Precision (*P*), Recall (*R*), Mean Average Precision (mAP), Model Parameter Size (Params), and Total Floating Point Operations (FLOPs). Among these metrics, Params represents the total number of parameters contained in the model, while FLOPs assess the computational complexity of the model. Precision evaluates the accuracy of the prediction results, and Recall measures the model's ability to correctly identify true positive samples.

D. Comparison Experiment

To demonstrate the advancement of the proposed model, comparisons were conducted on the VisDrone2019 test set between the proposed model and various sizes of YOLOv8 (YOLOv8n, YOLOv8s, YOLOv8m), the YOLOv9c model, as well as algorithms reported in the literature. The experimental results are shown in Table I. The improved models of different sizes achieved significant improvements in precision, recall, and mAP@0.5 metrics. The proposed improved model notably enhanced detection accuracy while significantly reducing model size and computational complexity. Specifically, compared to YOLOv8s, LMUAV-YOLOv8s improved precision, recall, mAP@0.5, and mAP@0.5:0.95 by 4.8, 4.5, 5.3, and 3.3 percentage points respectively, while reducing parameter count by 63.9% and increasing computation by only 0.4 GFLOPs.

Compared to YOLOv8m, LMUAV-YOLOv8s improved precision, recall, mAP@0.5, and mAP@0.5:0.95 by 0.5, 2.5, 2.3, and 1.1 percentage points, respectively, with reductions of 77.3% and 59.4% in parameters and computation. Compared to high-precision UAV aerial target detection algorithms [12,14], LMUAV-YOLOv8s shows significant advantages in accuracy, model size, and computational complexity. Although lightweight UAV aerial target detection algorithms [28] have smaller model size and computation compared to LMUAV-YOLOv8s, they suffer from excessive accuracy loss.

The comparative experimental results indicate that LMUAV-YOLOv8 offers significant advantages in both detection accuracy and lightweight design compared to mainstream target detection algorithms and the latest aerial target detection methods.

TABLE I. COMPARATIVE EXPERIMENTAL RESULTS OF MODELS

Model	P/%	R/%	mAP@0.5/%	mAP@0.5：0.95/%	Model Size/MB	Number of Parameters/10^6	Computation /10^9
YOLOv8n	38.8	29.6	27.3	15.1	6.20	3.00	8.1
LMUAV_YOLOv8n	41.3	33.0	31.1	17.5	2.79	1.16	10.2
YOLOv8s	45.3	34.5	32.9	18.6	22.50	11.13	28.5
LMUAV_YOLOv8s	50.1	39.0	38.2	21.9	8.41	4.01	28.9
YOLOv8m	49.6	36.5	35.9	20.8	52.00	25.84	78.7
YOLOv9-c	50.2	38.5	37.0	22.1	49.10	25.30	102.1
Reference [12]	48.9	38.5	37.2	21.2	23.08	11.85	38.5
Reference [14]	46.3	36.8	35.1	20.0	20.87	10.70	35.1
Reference [28]	46.8	35.5	34.2	19.6	12.65	6.49	24.8

To verify the detection performance of the proposed model on various targets, comparisons were made between the proposed model, the baseline YOLOv8s, and YOLOv9c in terms of classification. The experimental results are shown in Table II, detailing the precision, recall, and average precision (AP) for each target category, as well as the overall mAP@0.5 across all categories. Compared to YOLOv8s, the proposed model significantly improved the AP values for each category. Notably, for small targets such as pedestrians, crowds, tricycles, cars, and motorcycles, the AP values increased by 8.1, 8.9, 8.0, 5.0, and 5.5 percentage points, respectively. Specifically, the recall rates for pedestrians, crowds, tricycles, cars, and bicycles improved significantly by 7.1, 7.7, 8.2, 4.0, and 3.8 percentage points, indicating that the proposed model greatly reduced false negatives and demonstrated higher reliability in densely populated pedestrian and UAVs scenes. Furthermore, detection precision across categories also showed varying degrees of significant improvement, with bicycles, crowds, tricycles, and motorcycles increasing by 11.1, 7.3, 4.8, and 6.9 percentage points respectively, demonstrating that the improved model is more accurate in identifying targets and significantly reduces false positives.

These experimental results indicate that LMUAV_YOLOv8s not only significantly improves small target detection accuracy but also advances detection performance for medium and large targets. Compared to YOLOv9c, the improved model exhibits higher performance in small target detection, with AP values for pedestrians, crowds, and cars exceeding YOLOv9c by 4.1, 7.6, and 1.8 percentage points respectively, and mAP@0.5 exceeding YOLOv9c by 1.2 percentage points. However, due to YOLOv9c's higher computational cost and parameter count compared to the proposed model, YOLOv9c achieves slightly better AP values for trucks and buses. Therefore, YOLOv9c's balance between detection performance and resource consumption is suboptimal, making it less suitable for resource-constrained embedded devices.

To validate the effectiveness of the proposed feature fusion method, this paper compares UAV_RepGFPN with other advanced feature fusion techniques. The comparison results are shown in Table III. PAFPN is a feature fusion technique used in YOLOv8 that combines multi-scale feature information through lateral connections and a pyramid hierarchical structure to achieve more comprehensive and

multi-scale feature representation. The author in [11] describes the use of PAFPN in YOLOv8, which includes the P2 detection head but removes the P5 detection head. Reference [28] discusses the commonly used BiFPN implementation in YOLOv8, which includes skip connections added at the P3 and P4 layers. Reference [29] presents the general use of Efficient-RepGFPN in YOLOv8 with the addition of a small object detection head.

Compared with the above feature fusion methods, the proposed UAV_RepGFPN demonstrates significant advantages in performance metrics, model size, and computational cost. Compared to the optimized PAFPN method in reference [11], UAV_RepGFPN improves precision, recall, mAP@0.5, and mAP@0.5:0.95 by 0.5, 0.5, 0.8, and 0.4 percentage points respectively. Additionally, the computational weight decreases by 7.0×10^8 FLOPs, while the model size remains nearly unchanged, differing by only 0.07 MB.

To verify the impact of the Triplet Attention module added to the Faster_Block on model performance, this study combines Faster_Block with different attention mechanisms based on the improved model for comparison. The experimental results are shown in Table IV. Compared to several other attention mechanisms, Triplet Attention, as a lightweight spatial-channel attention mechanism, achieves the highest performance improvement while causing almost no increase in model parameters and computational cost.

TABLE II. COMPARATIVE EXPERIMENTAL RESULTS OF DETECTION ACCURACY FOR EACH CATEGORY

Model	Result	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning-Tricycle	Bus	Motor	all
YOLOv8s	P	50.7	50.1	22.2	66.7	44.4	44.7	25.1	40.5	67.0	41.3	45.3
	R	25.9	12.0	12.0	72.9	40.6	41.0	26.0	22.7	56.0	36.0	34.5
	mAP@0.5	27.9	15.6	9.0	72.2	38.5	39.8	17.5	19.3	58.6	30.3	32.9
YOLOv9c	P	56.4	52.2	32.3	72.2	48.7	51.6	28.5	38.4	75.0	47.8	50.3
	R	28.9	13.7	15.2	75.4	43.3	49.1	35.3	25.0	55.9	39.8	38.2
	mAP@0.5	31.9	16.9	14.0	75.4	41.3	**47.2**	23.5	19.5	**62.6**	**36.2**	36.9
LMUAV_YOLOv8s	P	55.2	57.4	33.3	70.5	47.7	47.7	29.9	42.4	69.2	48.2	50.1
	R	33.0	19.7	15.8	76.9	44.3	46.7	34.2	23.9	56.3	39.1	39.0
	mAP@0.5	**36.0**	**24.5**	**14.0**	77.2	**43.4**	44.0	**24.5**	**21.9**	60.8	35.8	**38.2**

TABLE III. COMPARATIVE EXPERIMENTAL RESULTS OF DIFFERENT FEATURE FUSION METHODS

Model	P/%	R/%	mAP@0.5/%	mAP@0.5:0.95/%	Model Size/MB	Computation /10^9	Number of Parameters/10^6
PAFPN	45.3	34.5	32.9	18.6	22.50	28.5	11.13
Reference [11]	48.0	37.6	36.6	20.8	15.20	34.6	**7.40**
Reference [28]	46.8	36.0	34.8	20.0	20.40	34.5	9.66
Reference [29]	48.1	37.6	36.7	21.0	20.30	37.8	10.30
UAV_RepGFPN	**48.5**	**38.1**	**37.1**	**21.2**	15.27	**33.9**	7.62

TABLE IV. COMPARATIVE EXPERIMENTAL RESULTS OF DIFFERENT ATTENTION MECHANISMS

Model	P/%	R/%	mAP@0.5/%	mAP@0.5:0.95/%	Number of Parameters/10^6	Model Size/MB	Computation /10^9
Faster_Block	48.9	37.8	37.4	21.4	4.01	8.39	28.9
+SE	48.8	38.2	37.4	21.4	4.01	8.41	28.9
+EMA	48.3	38.1	37.2	21.2	4.02	8.41	29.1
+SimAM	48.5	38.3	37.5	21.4	4.01	8.39	28.9
+MLCA	48.4	37.8	37.4	21.5	4.01	8.39	28.9
+CPCA	48.9	38.1	37.3	21.4	4.13	8.63	29.4
+TA	**49.2**	**38.4**	**37.6**	**21.5**	**4.01**	**8.41**	**28.9**
+BiLevelRouting	49.2	37.9	37.4	21.3	4.13	8.78	29.4

E. Ablation Experiment

To verify the effectiveness of each improvement method, YOLOv8s was used as the baseline model, and ablation experiments were conducted on each improved module based on the baseline. The modules UAV_RepGFPN, ADown, FTA_C2f, and UAV-PGI were added sequentially, with the resulting networks denoted as Model_A, Model_B, Model_C, Model_D, Model_E, and Model_F. The experimental results are shown in Table V.

TABLE V. ABLATION EXPERIMENT OF LMUAV-YOLOV8S

Model	UAV _RepGFPN	ADown	FTA_C2f	UAV_PGI	Number of participants/10^6	Calculations/10^9	P/%	R/%	mAP@0.5/%	mAP@0.5/%: 0.95/%
YOLOv8s	×	×	×	×	11.13	28.5	45.3	34.5	32.9	18.6
Model_A	√	×	×	×	7.62	33.9	48.5	38.1	37.1	21.2
Model_B	×	√	√	×	7.73	24.2	45.9	34.9	33.6	19.1
Model_C	√	√	×	×	6.35	31.9	49.0	38.2	37.4	21.4
Model_D	√	×	√	×	5.29	31.0	48.9	38.3	37.6	21.6
Model_E	√	√	√	×	4.01	28.9	49.2	38.4	37.6	21.5
Model_F	√	√	√	√	4.01	28.9	50.1	39.0	38.2	21.9

1) As shown in Model_A, by designing the lightweight multi-scale feature fusion network UAV_RepGFPN, the network performance was significantly improved, with mAP@0.5 increasing by 4.2 percentage points; at the same time, through optimization of the feature fusion paths, downsampling modules, and feature fusion modules, the parameter count was reduced by 31.5%. UAV_RepGFPN enhances the network's ability to capture spatial location information, leading to better extraction of small object features.

2) As shown in Model_C, using the ADown downsampling module in the network's high downsampling layers, which replaces convolution with average pooling and max pooling operations, reduces the model's parameter count and computational load by 16.6% and 5.8% respectively, while accuracy and mAP@0.5 improved by 0.5 and 0.3 percentage points.

3) As shown in Models D and E, replacing traditional convolutions with PConv in the backbone's C2f modules results in substantial reductions in parameters and computation; adding the triplet attention mechanism to the C2f bottleneck further improves model performance with negligible impact on complexity. FTA_C2f improves feature extraction efficiency while reducing parameters and computation by 43.8% and 14.7%, respectively.

4) As shown in Model_F, the introduction of UAV_PGI can improve the model's accuracy and mAP@0.5 by 0.9 and 0.6 percentage points, respectively, without adding any computation or parameters during inference.

The experiments of adding each proposed module one by one demonstrate that every improvement contributes to performance enhancement. This series of optimizations not only significantly boosts detection accuracy but also effectively reduces model complexity, making the model more suitable for resource-constrained UAV object detection scenarios.

To verify the effectiveness of the improvements made to the feature fusion network, PAFPN was used as the baseline model, with Efficient-GFPN as the base structure for the neck network. Then, the P2 detection head was added, the P5 detection head was removed, and improvements were made to Queen Fusion, GhostConv, and DBB_GELAN, resulting in the creation of Neck_A, Neck_B, Neck_C, Neck_D, Neck_E, Neck_F, and Neck_G for ablation experiments on the feature fusion network. The experimental results are shown in Table VI.

TABLE VI. ABLATION EXPERIMENT OF UAV_REPGFPN

Model	Efficient-GFPN	Add P2	Remove P5	Improvement of Queen fusion	GhostConv	DBB_GELAN	Number of participants/10^6	Calculations/10^9	mAP@0.5/%
YOLOv8s	×	×	×	×	×	×	11.13	28.5	32.9
Neck_A	√	×	×	×	×	×	10.40	27.9	33.0
Neck_B	√	√	×	×	×	×	10.30	37.8	36.7
Neck_C	√	√	√	×	×	×	8.87	36.1	36.7
Neck_D	×	√	√	×	×	×	7.40	34.1	36.1
Neck_E	√	√	√	√	×	×	8.54	36.1	36.7
Neck_F	√	√	√	√	√	×	8.32	35.6	36.9
Neck_G	√	√	√	√	√	√	7.62	33.9	37.1

5) Based on YOLOv8s, model Neck_A introduced Efficient-GFPN as the feature fusion network, reducing the model's parameter count and computation by 6.5% and 2.1%, respectively, while mAP increased by 0.1 percentage points. This indicates that the Efficient-GFPN module can reduce model complexity while improving detection performance, demonstrating good applicability of Efficient-RepGFPN in YOLOv8.

1) Model Neck_B added the P2 detection head on top of Neck_A. Although the parameter count and computation increased, mAP significantly improved from 33.3% to 36.7%. This shows that adding P2 effectively enhances the model's

multi-scale feature extraction ability, thus improving detection accuracy.

2) Removing the P5 detection layer in Neck_C led to reductions in parameters and computation by 13.8% and 4.5%, respectively, while mAP remained unchanged at 36.7%. This suggests that P5 has a minor role in this model, and its removal reduces complexity without sacrificing accuracy.

3) After improving Queen Fusion, even though parameters and computation decreased, mAP remained at 36.7%. This indicates that the improved Queen Fusion module maintains detection accuracy while reducing computational resource consumption.

4) Introducing GhostConv based on the improved Queen Fusion further reduced parameters and computation, while mAP increased to 36.9%, demonstrating the effectiveness of this improvement.

5) Adding DBB_GELAN on top of Neck_F reduced parameters and computation by 8.4% and 4.7%, respectively, and increased mAP by 0.2 percentage points. This shows that the DBB_GELAN module can significantly enhance model efficiency and accuracy.

Each step of the improvements targeted different aspects of the model, ultimately achieving the goal of reducing parameters and computation while improving detection accuracy. Particularly, the combination of adding P2, removing P5, and introducing DBB_GELAN significantly improved model efficiency while maintaining or enhancing accuracy.

F. Visual Analysis

To visually demonstrate the detection performance of the proposed model, comparative experiments were conducted across four dimensions: PR curves, confusion matrices, inference result images, and saliency maps.

Fig. 15 shows the PR curves of YOLOv8s and LMUAV-YOLOv8s on the VisDrone2019 dataset. From the average precision of the PR curves, the LMUAV-YOLOv8s model overall outperforms YOLOv8s, with a larger area under the curve, indicating higher precision at various recall rates.

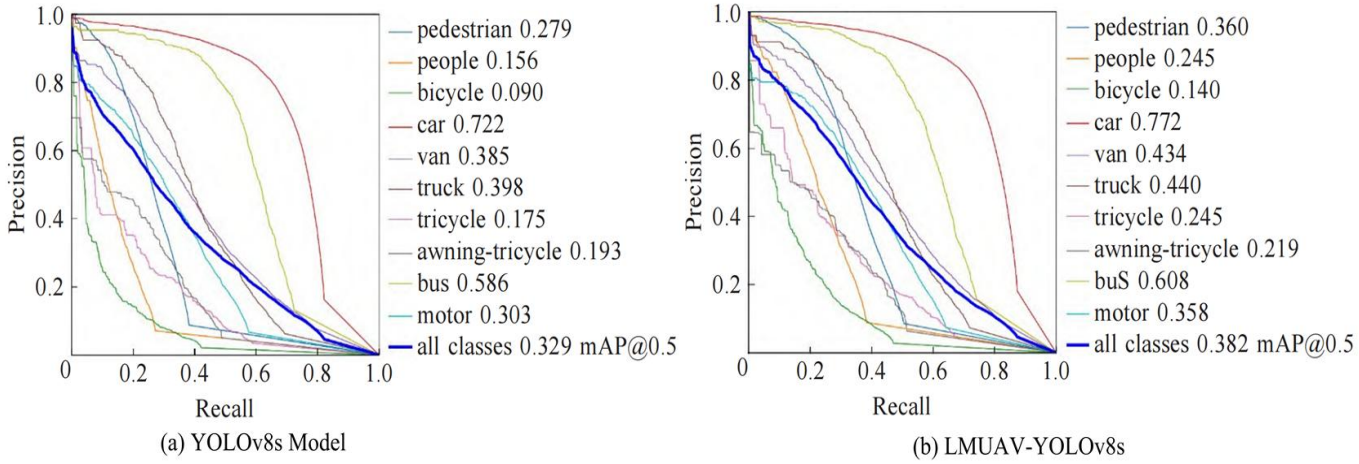


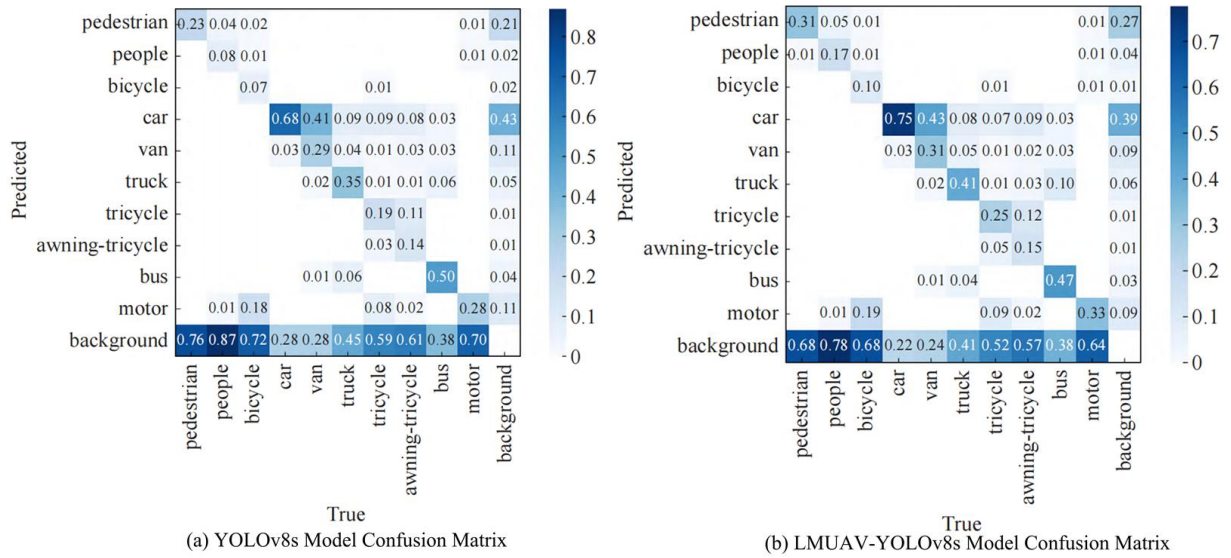
Fig. 15. PR curve.

Fig. 16 presents the confusion matrices of YOLOv8s and LMUAV-YOLOv8s on the VisDrone2019 dataset. Fig. 16(a) is the confusion matrix of YOLOv8s, while Fig. 16(b) is that of LMUAV-YOLOv8s. The diagonal values in the confusion matrix of LMUAV-YOLOv8s are generally higher than those of YOLOv8s, and the values in the last row (background) are generally lower, indicating that LMUAV-YOLOv8s effectively reduces false positives and false negatives across categories. However, for categories with less distinctive features such as pedestrians, bicycles, tricycles, and motorcycles, although the improved model reduces false positive and false negative rates, the proportion of correctly detected instances remains relatively low.

To intuitively compare the performance of the models, four challenging scenarios were selected: complex background scenes, densely distributed small target scenes, occluded small target scenes, and low-light scenes.

For a direct comparison of the detection performance of these three algorithms, a quantitative analysis was conducted on the inference results, counting the detection performance for specific target categories (i.e. the ground truth and the number of correctly detected targets by each model) under different scenarios.

- stronger detection capability for distant UAVs and densely packed pedestrians from high-altitude views;
- more accurate recognition and classification of visually similar objects, such as cars and vans, bicycles and motorcycles, across multiple scenarios;
- stronger detection ability for partially occluded objects.



(a) YOLOv8s Model Confusion Matrix (b) LMUAV-YOLOv8s Model Confusion Matrix

Fig. 16. Confusion matrix.

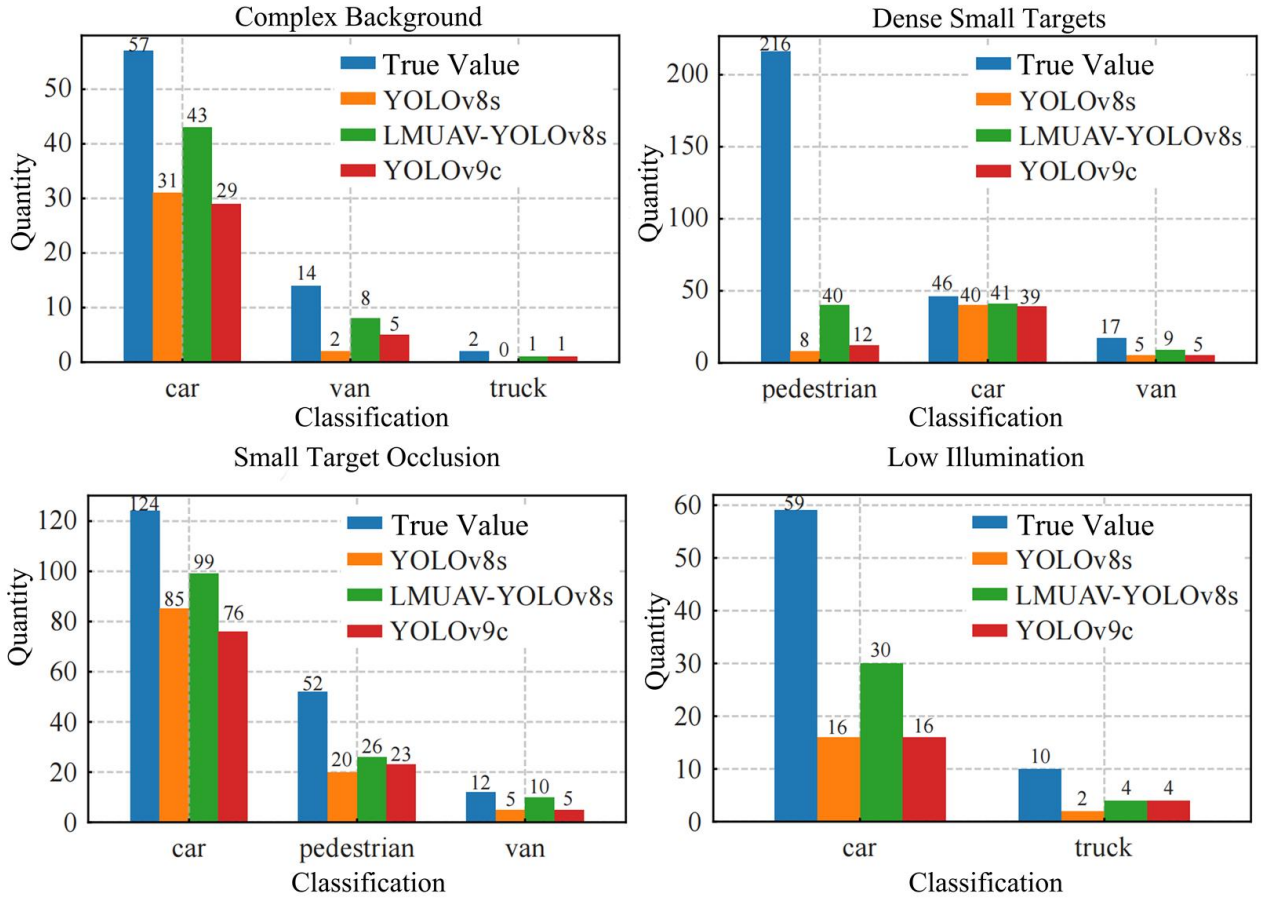


Fig. 17. Performance comparison of object detection models in multiple scenarios.

G. Model Deployment Experimental Results and Analysis

To validate the effectiveness of the proposed algorithm on a UAV embedded platform, comparative experiments on detection accuracy and speed were conducted on the NVIDIA Jetson Xavier NX platform using the VisDrone2019 dataset. The proposed method was compared against six state-of-the-art models (YOLOv8, YOLOv9, YOLOv10, UAV-YOLOv8, HV-YOLOv8 [31], Aero-YOLO [32], and the method in [12]). Additionally, the impact of TensorRT precision calibration modes on detection performance was evaluated. The evaluation metrics for accuracy included precision, recall, and mean average precision (mAP), while detection latency

was used to assess detection speed. Detection latency refers to the sum of image preprocessing delay, inference delay, and non-maximum suppression delay. Latency was measured on the embedded platform by running the models on the test dataset with a batch size of 1, calculating the average detection latency. The evaluation was conducted using single-precision floating point (FP32), half-precision floating point (FP16), and 8-bit integer (INT8) computation modes. A batch size of 1 was chosen because the goal is real-time image processing, matching the maximum image acquisition rate of the camera. The input image size used was 384×640. The models trained on a high-performance computing platform were deployed to the Jetson Xavier NX development board. TensorRT acceleration was utilized to convert the PyTorch-trained UAV detection model's .pt weight files into .onnx intermediate files. Then, the .onnx files were used to build inference engine (.engine) files for accelerated inference, allowing the model to run efficiently on the embedded platform.

To further validate the effectiveness of the improved algorithm, this section evaluates the proposed method on the VisDrone2019 dataset and conducts comparative experiments with other state-of-the-art (SOTA) object detection models under the same experimental conditions. The detailed comparison results are shown in Fig. 17. As seen in Table VII, after FP32 (single precision) and FP16 (half precision) calibration, the model size of LMUAV-YOLOv8s is reduced to 21.4 MB and 10.3 MB, respectively. Compared with the algorithm proposed in [12], which has similar detection accuracy, the improved algorithm reduces the model size by 64.6% and 59.3% under the two quantization methods. Compared with the lightweight algorithm YOLO-Aero, the improved algorithm reduces the model size by 42.6% and 31.7% under the two quantization schemes. Thanks to the use of lightweight multi-scale feature fusion, efficient feature extraction networks, and context-guided programmable gradient information, LMUAV-YOLOv8s achieves significantly smaller model sizes under both precision calibrations compared to other SOTA models, accelerating model download speeds, reducing storage requirements, and lowering memory usage during deployment—making it especially suitable for embedded devices.

Without TensorRT acceleration and after single-precision calibration with TensorRT acceleration, the detection latency of all models at the small (s) scale in Table VII exceeds 33 ms, failing to meet real-time detection latency requirements. After half-precision calibration, LMUAV-YOLOv8n and LMUAV-YOLOv8s achieve detection latencies of 25.3 ms and 28.2 ms, respectively, satisfying real-time detection latency demands. Specifically, compared to YOLOv9s, UAV-YOLOv8s, and the method in [14], LMUAV-YOLOv8s reduces detection latency by 1.3%, 21.5%, and 64.6%, respectively, while improving detection accuracy (mAP@0.5) by 4.6, 3.3, and 0.9 percentage points. Compared to lightweight algorithms (YOLOv8s, HV-YOLOv8s, Aero-YOLO), although LMUAV-YOLOv8s shows increased detection latency by 6 ms, 3 ms, and 2.3 ms, it achieves higher detection accuracy (mAP@0.5) improvements of 5.2, 5.6, and 6.9 percentage points, respectively, demonstrating superior overall performance. Therefore, after TensorRT acceleration, the LMUAV-YOLOv8 model achieves a better balance between detection accuracy and latency.

TABLE VII. MODEL TESTING RESULTS ON JETSON XAVIER NX PLATFORM

Model	Accuracy Calibration	Model size/MB	P/%	R/%	mAP@0.5%	mAP@0.5%: 0.95/%	Detection delay/ms
YOLOv8n	FP32	17.3	39.4	30.0	27.5	15.2	20.9
	FP16	7.5	39.4	30.0	27.5	15.2	**13.9**
LMUAV-YOLOv8n	FP32	9.6	42.3	31.6	30.3	17.0	30.9
	FP16	**5.5**	42.1	31.7	30.3	17.0	25.3
YOLOv8s	FP32	60.8	45.8	34.4	32.9	18.6	49.1
	FP16	22.9	45.6	34.4	32.8	18.5	22.2
LMUAV-YOLOv8s	FP32	21.4	**50.1**	**37.9**	**38.1**	**21.8**	53.9
	FP16	10.3	49.9	37.9	38.0	21.7	28.2
YOLOv9s	FP32	44.3	46.8	34.7	33.4	19.0	55.2
	FP16	16.9	46.7	34.6	33.4	18.9	29.5
YOLOv10s	FP32	34.7	45.3	35.1	33.5	18.6	43.4
	FP16	16.0	44.8	35.0	33.3	18.5	20.5
UAV-YOLOv8[14]	FP32	57.6	45.9	36.7	34.8	19.9	71.8
	FP16	22.9	46.2	36.5	34.7	19.8	36.7
HV-YOLOv8[30]	FP32	53.8	45.5	34.2	32.4	18.2	51.2
	FP16	20.6	45.41	34.1	32.4	18.2	25.2
YOLO-Aero[31]	FP32	37.3	43.1	33.2	31.1	17.4	48.4
	FP16	15.1	43.0	33.1	31.1	17.3	25.9
[12]	FP32	60.5	49.0	38.5	37.1	21.3	117.8
	FP16	25.3	49.0	38.4	37.1	21.2	81.4

In summary, the proposed LMUAV-YOLOv8 model demonstrates excellent detection performance on embedded platforms, indicating strong suitability for real-time UAV target detection scenarios.

IV. Model Interpretation

Deep learning models contain a large number of parameters that are difficult to interpret and involve complex nonlinear operations, making their interpretability a significant challenge. To address this challenge, researchers have proposed various saliency map generation methods to explain the behavior of deep neural networks. Saliency maps highlight important regions in the input image, helping to interpret the model's decision-making process. Among these methods, Class Activation Map (CAM) approaches are widely used due to their speed and lack of need for manual guidance. These methods generate saliency maps corresponding to the input image by utilizing the model's feature maps or gradient information, emphasizing regions most important for the model's prediction.

This work employs the High-Resolution Class Activation Mapping (HiResCAM) method to generate saliency maps. First, HiResCAM backpropagates the confidence scores of the model's output classes and bounding box regressions to obtain the gradient value of each pixel. The gradient values reflect the model's attention to different regions of the input image during classification and localization decisions. In the heatmaps generated by HiResCAM, pixels with high gradients are shown in deep red, indicating regions closely related to object recognition and localization; pixels with low gradients are shown in deep blue, indicating regions less related to these tasks.

To further verify the effectiveness of the proposed modules and explain their mechanisms, the baseline model YOLOv8s and three models from the ablation experiments (Experiments A, E, and F) were selected to generate saliency maps across four scenarios. Compared to the baseline model, the saliency maps of ablation experiment A show three changes: the red regions become larger and more concentrated, new small blue regions appear or small blue regions turn red, and the red regions of cars occluded by trees gradually become larger and more focused.

1) The enlargement and concentration of red regions indicate that the introduction of the UAV_RepGFPN network enables more efficient extraction and fusion of multi-scale features, making the feature information of target regions more complete and clear, thereby improving detection confidence.

2) The deepening of red regions on small objects indicates increased detection confidence for small targets, while newly appearing small blue regions suggest the model can detect more low-confidence small targets, demonstrating that the model retains high-resolution features over a larger area and is more effective in handling tasks with high detail and resolution requirements.

3) The expansion and concentration of red regions on occluded objects indicate that the UAV_RepGFPN network helps the model recover occluded parts of the features and increases confidence in those features when handling partially occluded targets.

Ablation experiment E (which introduces lightweight modules) maintains the additional blue and light red regions observed in ablation experiment A across various scenarios, while the red regions become more concentrated. This indicates that after lightweight processing and the introduction of the Triplet Attention mechanism, the model still maintains the ability to detect small and low-confidence objects, while further improving detection accuracy in high-confidence regions. Specifically, the FTA_C2f module introduced in experiment E performs convolution operations on non-masked regions, preserving more high-resolution features needed for small object detection. This allows the model to better detect fine targets, even within lower-confidence regions. Furthermore, the FTA_C2f module jointly encodes spatial and channel information, significantly enhancing the model's spatial position capturing ability and increasing attention to important feature regions, resulting in more concentrated red areas. These improvements collectively enable the model to detect accurately within high-confidence regions while maintaining sensitivity to small objects over a broader area.

Compared to ablation experiment E, the heatmaps generated by ablation experiment F show even more focused red regions and further improvements in reducing small object misses, which is closely related to UAV_PGI. The context-guided auxiliary branch produces more reliable gradients by reducing information loss, ensuring accurate parameter updates during training and improving detection confidence. This is reflected in the heatmaps as more focused red regions and the presence of additional small blue or red areas.

V. Conclusion

In this study, to address the issues of missed and false detections of small-scale faults and anomalies in complex distribution network environments, a lightweight multi-scale feature fusion detection network named LMGrid-YOLOv8 is proposed and embedded into a deep reinforcement learning framework to enable distribution network-oriented fault detection and autonomous control. Specifically:

- A lightweight multi-scale feature fusion structure Grid_RepGFPN tailored for distribution network monitoring data is designed to effectively enhance the fusion efficiency of electrical and spatial information;
- The introduction of FTA_neck and ADown modules improves the spatial feature extraction capability of the deep backbone network while maintaining lightweight characteristics and reducing loss of high-resolution measurements;
- A context-guided reversible branch structure Grid_PGI is designed to alleviate the information bottleneck issue, improving both training efficiency and feature representation capability.

Comparative experiments show that LMGrid-YOLOv8 significantly improves detection accuracy for small or subtle anomalies, such as partial discharges, line overloads, and

minor voltage fluctuations, while keeping the number of parameters and computational cost low, with average precision (AP) and recall for these events outperforming mainstream detection networks. Embedded deployment experiments demonstrate that the model achieves an inference speed of 28.2 ms/frame and a detection accuracy of 38.0% on an edge computing platform, meeting the real-time monitoring requirements of distribution networks and achieving a good balance between speed and accuracy. Class activation maps further confirm the model's ability to focus on critical fault points and capture long-range correlations, enhancing its detection performance in complex network scenarios. Additionally, by integrating the detection module with a deep reinforcement learning-based control strategy, real-time fault response and load adjustment can be realized, demonstrating strong system integration and practical value. Nevertheless, there is still room for improvement in detecting extremely subtle or transient anomalies. Future work will focus on further optimizing the model structure to continuously enhance detection accuracy and the robustness of the reinforcement learning control strategy under constrained computational resources.

ACKNOWLEDGMENT

This work was supported in part by the Inner Mongolia Electric Power Group (Limited) Company Science and Technology Project Funding (Document Number: Nei Dian Sheng (2022) No. 6).

REFERENCES

[1] XU H Z, GU X N. Research on optimization of UAV traffic small target image detection algorithm. Computer Engineering and Applictions, 2024, 60(21): 194-204.

[2] SHENG S, DUAN X H, HU W K, et al. Dynamic-YOLOX: detection model for apple leaf disease in complex background. Journal of Frontiers of Computer Science and Technology, 2024, 18(8): 2118-2129.

[3] XU Y P, XIE Y Q, YU R, et al. Integrated perception-communication-logistics multi-objective. Journal on Communications, 2024, 45(4): 1-12.

[4] SUN S, MO B, XU J, et al. Multi-YOLOv8: an infrared moving small object detection model based on YOLOv8 for air vehicle. Neurocomputing, 2024, 588.

[5] Feng, Pengbin, and Xuhui Peng. "A note on Monge–Kantorovich problem." Statistics & Probability Letters 84 (2014): 204-211.

[6] XIAO B, NGUYEN M, YAN W Q. Fruit ripeness identification using YOLOv8 model. Multimedia Tools and Applications, 2024, 83(9): 28039-28056.

[7] SUN Y, ZHANG Y, WANG H, et al. SES-YOLOv8n: automatic driving object detection algorithm based on improved YOLOv8. Signal, Image and Video Processing, 2024, 18: 3983-3992.

[8] YAN H N, LYU F, FENG Y A. Feature-level adaptive enhancement for UAV target detection algorithm. Journal of Frontiers of Computer Science and Technology, 2024, 18(6): 1566-1578.

[9] CAO J, BAOW, SHANGH, et al. GCL-YOLO: a Ghost Conv-based lightweight YOLO network for UAV small object detection. Remote Sensing, 2023, 15(20): 4932.

[10] Zhang, Zhilin, Naveed Ahmed Saleem Janvekar, Pengbin Feng, and Nitika BHASKAR. "Graph-based detection of abusive computational nodes." U.S. Patent 12,223,056, issued February 11, 2025.

[11] Zhou, Yu, Hao Xia, Dahui Yu, Jiaoyang Cheng, and Jichun Li. "Outlier detection method based on high-density iteration." Information Sciences 662 (2024): 120286.

[12] PAN W, WEI C, QIAN C Y, et al. Improved YOLOv8s model for small object detection rom perspective of drones. Computer Engineering and Applictions, 2024, 60(9): 142-150.

[13] H. Liu, Z. Zhang, R. Song, Z. Shu, J. Wang, H. Tian, Y. Song, W. Chen, Pattern Recognition Method for Detecting Partial Discharge in Oil-paper Insulation Equipment using Optical F-P Sensor Array based on KAN-CNN Algorithm, Journal Lightwave Technology, (2025) 43(12) 6004–6012.

[14] WANG G, CHEN Y, AN P, et al. UAV-YOLOv8: a small object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. Sensors, 2023, 23(16): 7190.

[15] Lu, Wanglong, Jikai Wang, Tao Wang, Kaihao Zhang, Xianta Jiang, and Hanli Zhao. "Visual style prompt learning using diffusion models for blind face restoration." Pattern Recognition 161 (2025): 111312.

[16] LI Y, FAN Q, HUANG H, et al. Amodified YOLOv8 detection network for UAV aerial image recognition. Drones, 2023, 7(5): 304.

[17] HAN K, WANG Y, TIAN Q, et al. GhostNet: more features from cheap operations. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 1580-1589.

[18] H. Liu, Z. Zhang, H. Tian, Y. Song, J. Wang, Z. Shu, W. Chen, Comparison of Different Coupling Types of Fiber Optic Fabry-Perot Ultrasonic Sensing for Detecting Partial Discharge Faults in Oil-Paper Insulated Equipment, IEEE Transactions on Instrumentation and Measurement. (2024), 73, 9519612.

[19] XU X, JIANG Y, CHEN W, et al. Damo-YOLO: a report on real-time object detection design. arXiv:2211.15444, 2022.

[20] WANG C Y, YEH I H, LIAO H Y M. YOLOv9: learning what you want to learn using programmable gradient Information. arXiv:2402.13616, 2024.

[21] CHEN J, KAO S, HE H, et al. Run, don't walk: chasing higher FLOPs for faster neural networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 12021-12031.

[22] MIRSA D, NALAMADA T, ARASANIPALAI A U, et al. Rotate to attend: convolutional triplet attention module. Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2021: 3138-3147.

[23] Yao, Shanliang, Runwei Guan, Zhaodong Wu, Yi Ni, Zile Huang, Ryan Wen Liu, Yong Yue et al. "Waterscenes: A multi-task 4d radar-camera fusion dataset and benchmarks for autonomous driving on water surfaces." IEEE Transactions on Intelligent Transportation Systems 25, no. 11 (2024): 16584-16598.

[24] WANG C Y, LIAO H Y M, WU YH, et al. CSPNet: a new backbone that can enhance learning capability of CNN. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020: 390-391.

[25] H. Liu, T.H. Yang, Z.X. Zhang, H.Y. Tian, Y.X. Song, Q.X. Sun, W. Wang, Y.J. Geng, W.G. Chen, Ultrasonic localization method based on Chan-WLS algorithm for detecting power transformer partial discharge faults by fibre optic F-P sensing array. High Voltage, 9(6), (2024) 1234-1245.

[26] DING X, ZHANG X, HAN J, et al. Diverse branch block: building a convolution as an inception- like unit. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021: 10886-10895.

[27] DU D W, ZHU P F, WEN L Y, et al. VisDrone-DET2019: the vision meets drone object detection in image challenge results. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, 2019: 213-226.

[28] CHENG H X, QIAO QY, LUO X L, et al. Object detection algorithm for UAV aerial image based on improvedYOLOv8. Radio Engineering, 2024, 54(4): 871-881.

[29] WANG A J, YUAN J L, ZHU Y J. Drum roller surface defect detection algorithm based on improved YOLOv8s. Journal of Zhejiang University (Engineering Science), 2024, 58(2): 370-380.

[30] Mengjie Zhang, Xiaolin Li. Understanding the relationship between coopetition and startups' resilience: the role of entrepreneurial ecosystem and dynamic exchange capability. Journal of Business & Industrial Marketing, 40(2), 2025, 527-542.

[31] SHAO Y, YANG Z, LI Z, et al. Aero-YOLO: an efficient vehicle and pedestrian detection algorithm based on unmanned aerial imagery. Electronics, 2024, 13(7): 1190.