Sensors and Materials, Vol. 38, No. 1 (2026) 477–494
MYU Tokyo

S & M 4311

C2 Block + Parallel Spatial Attention Module-Ghost Convolution-Feature Diffusion Pyramid Network-You Only Look Once (YOLO)-v11n: An Efficient and Real-time Small Object Detection Algorithm Based on YOLOV11n

Yu Fan,[1*] Junchao Lin,[1] Chinta Chen,[1] Mingkun Xu,[1] and Cheng-Fu Yang[2,3**]

[1]School of Electronic and Electrical Engineering, Zhaoqing University, Zhaoqing 526061, China
[2]Department of Chemical and Materials Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan
[3]Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

(Received December 7, 2025; accepted January 7, 2026)

Keywords: YOLOv11 algorithm, deep learning, feature extraction, attention mechanism, small target detection

Small object detection plays a critical role in applications such as security surveillance, autonomous driving, and remote sensing. However, conventional detection methods often struggle with high annotation costs, low resolution, and heavy computational requirements. To address these challenges, we propose CGF-YOLOv11n, which is the abbreviation of C2 block + parallel spatial attention module (C2PLUS)-Ghost Convolution (GhostConv)-Feature Diffusion Pyramid Network (FDPN)-You Only Look Once (YOLO)v11n, an efficient and real-time small object detection algorithm built upon the YOLOv11n framework. First, we introduce the C2PLUS module, which effectively enhances fine-grained feature extraction for small targets. Second, we design a plug-and-play Ghost-Residual Field-Aware Convolution module to strengthen the feature extraction capability of the backbone network. Finally, the FDPN module is incorporated to promote the balanced fusion between semantic features and spatial information. Experimental results on the VisDrone2019 dataset demonstrate that the proposed method achieves improvements of 3.5 and 3.1% in *mAP*@0.5 on the validation and test sets, respectively, outperforming the baseline YOLOv11n model. In addition, CGF-YOLOv11n achieves 34 frames per second on the Orange Pi 5 platform, confirming its suitability for real-time deployment and advancing the performance of small object detection systems. The related implementation details, including code and datasets, are available through the authors' public project repository. In this study, we primarily contribute an efficient modular enhancement strategy for real-time small object detection by integrating C2PLUS, Ghost-based convolution, and FDPN into a lightweight YOLOv11n framework. While the proposed CGF-YOLOv11n demonstrates notable accuracy gains and real-time performance on an embedded platform, the current evaluation is limited to a single aerial benchmark dataset and does not fully explore robustness under extremely dense scenes or severe resolution degradation. Future work will

*Corresponding author: e-mail: fy@zqu.edu.cn
**Corresponding author: e-mail: cfyang@nuk.edu.tw
https://doi.org/10.18494/SAM6112

ISSN 0914-4935
https://myukk.org/

focus on extending validation to more diverse datasets, improving generalization in complex real-world environments, and further optimizing the model for ultralow-power edge devices.

1. Introduction

With the widespread adoption of computer vision (CV) technologies in security surveillance, autonomous driving, and remote sensing, the demand for high-performance object detection algorithms continues to increase. Among existing approaches, the You Only Look Once (YOLO) family has become a major research focus in both academia and industry because of its favorable balance between detection accuracy and computational efficiency.[1] Despite these advantages, traditional YOLO models still encounter substantial challenges when detecting small objects. Owing to factors such as low resolution, blurred texture details, and susceptibility to background interference, the rates of missed and false detections for small targets–typically defined as objects occupying fewer than 32 × 32 pixels in an image–remain significantly higher than those for medium or large objects. For example, in traffic surveillance, pedestrians or distant vehicles often occupy only a tiny portion of the frame, while in remote sensing imagery, targets such as ships or vehicles frequently appear as densely distributed small pixel clusters.

Although the YOLO architecture incorporates multi-scale prediction to capture objects of various sizes, the deep network's aggressive downsampling inevitably weakens the feature representations of small targets. Moreover, shallow layers retain detailed textures but lack high-level semantic information, resulting in suboptimal feature fusion and further complicating detection.[2] To overcome these issues, recent studies have explored feature enhancement and context modeling strategies, for instance, optimizing feature pyramids (e.g., BiFPN), or integrating an attention mechanism such as Squeeze-and-Excitation (SE) or Convolutional Block Attention Module (CBAM) to strengthen informative feature regions.[3–5] However, these approaches still suffer from limitations, including the insufficient preservation of fine-grained details, increased computational overhead, and limited robustness of data augmentation methods.[6–9] Such technical bottlenecks severely hinder the practical deployment of YOLO-based models in key application scenarios including smart cities, unmanned aerial vehicle (UAV) inspection, and medical image analysis. For instance, failure to detect small objects in remote sensing imagery may compromise disaster monitoring accuracy, while missed or incorrect detections in traffic environments can lead to erroneous decision-making in autonomous driving systems.

Improving YOLO's capability in small object detection holds substantial theoretical and practical value. Theoretically, the design of lightweight feature enhancement modules, multi-granularity context-awareness mechanisms, and dynamic data augmentation strategies promotes advancements in feature representation, semantic reasoning, and sample balancing within object detection frameworks. Practically, enhancing the robustness of detection models in complex environments is critical for ensuring the reliability of systems such as autonomous driving and intelligent surveillance, while also guiding the deployment of lightweight models on edge-computing devices. Ultimately, these improvements support the broader integration of computer vision technologies into smart industry, public security, and medical diagnostics, yielding

significant societal and economic benefits. Since the introduction of the YOLO series, research on object detection, particularly the detection of tiny objects, has expanded rapidly worldwide. YOLOv1 pioneered the idea of reframing object detection as a regression problem, enabling end-to-end real-time detection. However, its performance on small objects remained limited. YOLOv2 subsequently incorporated multi-scale training and high-resolution classifiers, offering partial improvements in small object detection, yet considerable challenges persist when targets appear under complex imaging conditions.[10]

As the YOLO architecture has evolved, it has achieved greater representational capacity, improved computational efficiency on CPUs and embedded devices, and enhanced adaptability across diverse CV tasks.[11] The most recent iteration, YOLOv11, represents a major leap forward in real-time detection, delivering notable improvements in speed, efficiency, and accuracy through refined architecture and training techniques. Building upon YOLOv11n, in this study, we propose several architectural enhancements and validate their effectiveness on the VisDrone2019 dataset, and the contributions of this work are summarized as follows.

(1) C2 block + Parallel Spatial Attention (C2PSA) module: Inspired by transformer-based designs, the C2PLUS module, which serves as an advanced refinement of the original C2 block + C2PSA module in the YOLOv11 architecture, is an enhanced feature extraction block developed as part of YOLOv11 improvement research. Experiments demonstrate that this module substantially improves detection accuracy on VisDrone2019 without compromising inference speed.[12]
(2) Ghost Convolution (GhostConv) module: By integrating the concepts of GhostConv and Residual Field-Aware Convolution RFAConv,[13,14] we propose GRFAConv, a plug-and-play convolutional module that enhances backbone feature extraction. By focusing on spatial structures within the receptive field, GRFAConv mitigates the inherent limitations of convolutional kernel parameter sharing.
(3) Feature Diffusion Pyramid Network (FDPN) Neck module: A new neck architecture, the Balanced Spatial and Semantic Information FDPN, is developed to process multi-scale features extracted from the backbone. FDPN effectively balances spatial details and semantic cues, improving overall feature fusion quality.

To address this issue, we propose C2PLUS-GRFAConv-FDPN (CGF)-YOLOv11n, an enhanced version of YOLOv11n that integrates three key modules, namely, C2PLUS, GRFAConv, and FDPN, to strengthen fine-grained feature extraction, expand receptive fields, and improve semantic–spatial fusion. Through these advancements, in this study, we present an efficient and lightweight small object detection framework that achieves an optimal balance among accuracy, real-time performance, and model compactness. The proposed enhancements not only address long-standing limitations in feature preservation and semantic–spatial fusion within YOLO-based architectures but also provide a scalable solution suitable for deployment on edge devices and resource-constrained platforms. By substantially improving detection robustness in challenging environments, in this work, we lay a solid foundation for future research on real-time, high-precision perception systems and promote the broader adoption of small object detection technologies in practical applications.

2. Methodology

The YOLO family of object detection algorithms marked a major breakthrough by integrating class prediction and bounding box regression into a unified end-to-end neural network. This streamlined design removed the dependence on multi-stage processing pipelines, thereby enabling real-time inference while maintaining competitive accuracy compared with traditional detection frameworks.[15] Building on this paradigm, YOLOv11n extends and refines the architectural principles of YOLOv8 through structural innovations and parameter optimization, further enhancing its effectiveness in object detection tasks. The model incorporates advanced feature extraction components to capture fine-grained visual cues and significantly improves processing efficiency for real-time applications. The overall architecture of YOLOv11n is illustrated in Fig. 1. Compared with YOLOv8, YOLOv11n introduces several structural modifications. As shown in Fig. 1, the original C2f module is replaced with the C3k2 block, where the reduced convolutional kernel size (denoted by "k2") accelerates computation while preserving representational capacity. Additionally, the new C2PSA module is incorporated to enhance detection robustness for objects of various scales and spatial distributions. To further improve computational efficiency, two depthwise convolution (DWConv) layers are added to the decoupled detection head, which substantially reduces both parameter count and computational load.

Despite these improvements, detecting small objects remains challenging owing to insufficient feature preservation and limited receptive field adaptation. First, the C2PLUS module is introduced into the backbone to replace the original C2PSA module. Designed to

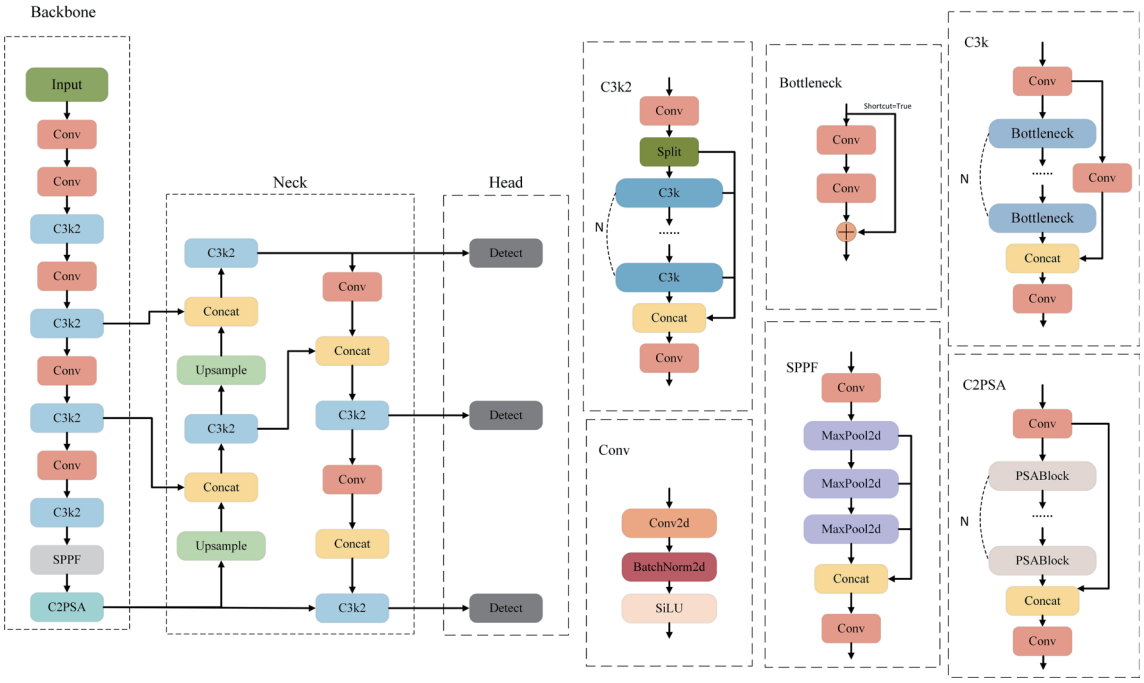


Fig. 1. (Color online) YOLOv11n network structure.

capture multi-granularity contextual information more effectively, C2PLUS enhances the network's ability to process small object features without degrading inference speed. Second, to further expand the receptive field and refine feature extraction within the backbone, we develop the GRFAConv module, an improved convolutional unit that replaces the two standard Conv layers and the Conv layer inside the C3k2 block. GRFAConv concentrates on spatial relationships within the receptive field and alleviates limitations associated with traditional convolution kernel parameter sharing. Finally, the Balanced Spatial and Semantic Information FDPN is introduced at the neck stage to facilitate the balanced fusion of shallow spatial information and deep semantic cues, thereby improving multi-scale representation consistency. The overall architecture of CGF-YOLOv11n is presented in Fig. 2.

To address the low detection accuracy of tiny objects in conventional object detection networks, many approaches incorporate self-attention mechanisms. Although traditional self-attention offers a large effective receptive field, it often overlooks channel-wise similarity. Conversely, attention mechanisms in the classical Convolutional Neural Network (CNN) exhibit limited receptive fields. For instance, popular channel-attention modules such as SE and Efficient Channel Attention rely on global average pooling to aggregate spatial information and then generate channel-wise weights based on similarity, which are multiplied with the original

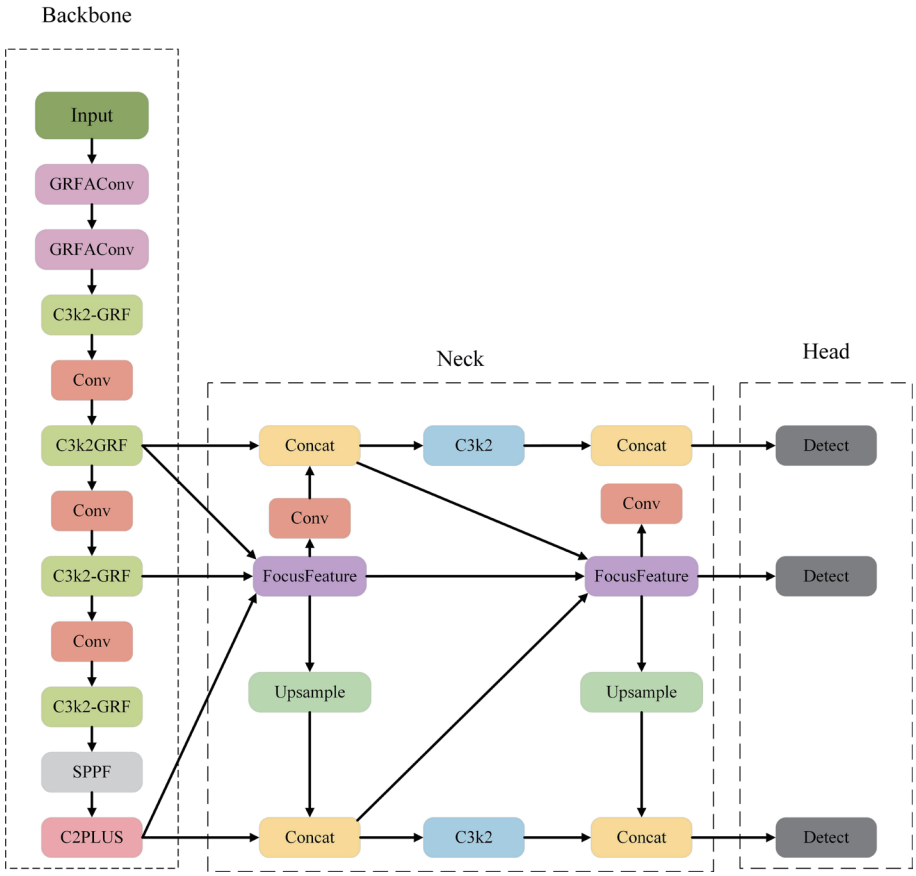


Fig. 2. (Color online) CGF-YOLOv11n network structure.

feature map to enhance important channels.[16] While such methods improve detection performance, their ability to capture fine-grained contextual relationships remains insufficient for small object feature representation. Motivated by these limitations, in this study, we propose a novel C2PLUS module to replace the PSABlock in the original C2PSA module. As illustrated in Fig. 3, the C2PLUS module enhances feature modeling by jointly considering the original input and inter-channel similarity, and then applying transformer-based processing to strengthen global contextual interactions.

Specifically, the redesigned block replaces the standard Attention mechanism with an AttentionPLUS structure. In this design, the feature map processed by the Multi-Path Aggregation (MPA) module is assigned as the key feature K, while the original input feature map serves as the query Q. Within the MPA module, the feature map is subjected to pooling and average-pooling operations along both horizontal and vertical directions. The aggregated results are then summed to obtain direction-aware structural information. Inspired by the SE attention mechanism, two 1 × 1 convolution layers are introduced for channel compression and expansion, enabling more effective channel-wise feature fusion. The fused representation is passed through a Sigmoid activation function to obtain the final attention weights, which combine global contextual cues with channel interaction. This process yields an enhanced feature representation that is better suited for capturing small object details. The computation procedure is summarized in Eq. (1), where X denotes the input feature map of the MPA module and the output of the MPA module is denoted as Y.

$$Y = X \times \mathrm{Sigmoid}\left(\mathrm{Conv}^{(1\times1)}\left(\mathrm{Conv}^{(1\times1)}\left(\mathrm{XAvgPool}(X) + \mathrm{YAvgPool}(Y)\right)\right)\right) \tag{1}$$

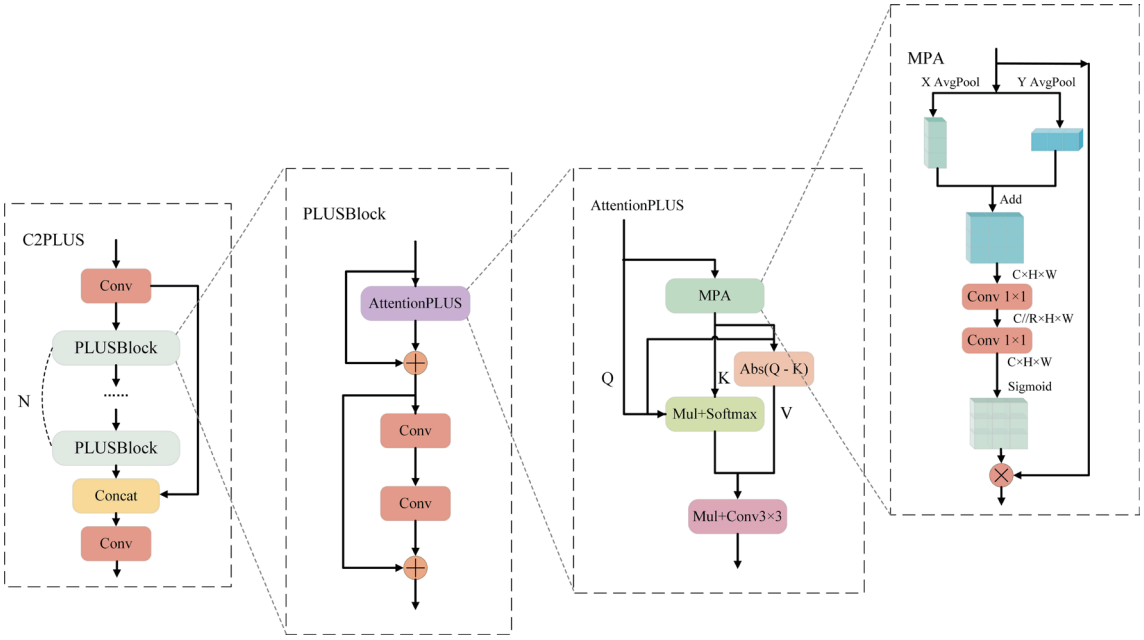


Fig. 3. (Color online) C2PLUS module structure diagram.

To model spatial similarity within the feature map, the value component V is defined as the absolute difference between Q and K, thereby strengthening the correlation between features before and after transformation. The computation of Q, K, and V is summarized in Eq. (2).

$$Q = X, K = Y, V = \mathrm{Abs}(Q - K) \tag{2}$$

After obtaining Q and K, a dot-product operation is performed to generate an intermediate correlation matrix. To prevent excessively large dot-product magnitudes and to avoid gradient vanishing after activation, we introduce a scaling factor $G = 1/\sqrt{d_k}$ in the dot-product attention. Here, d_k denotes the dimensionality of the key vector K (i.e., the channel/embedding size). This normalization stabilizes the SoftMax input distribution and improves training stability. The resulting matrix is then passed through the SoftMax function to produce the attention map, which reflects the similarity strength between different spatial regions of the feature map. Higher response values indicate stronger positive associations. The attention map is subsequently multiplied with the value representation to obtain the refined feature map, as summarized in Eq. 3(a). Finally, the output of this computation represents the enhanced representation produced by the AttentionPLUS module. The full process is described in Eq. 3(b), where Z denotes the final output feature of AttentionPLUS.

$$\begin{aligned} &\text{(a) } G = \frac{1}{\sqrt{d_k}} \\ &\text{(b) } Z = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \end{aligned} \tag{3}$$

The classic RFAConv module is designed from the perspective of spatial attention, utilizing partial pooling and 3 × 3 convolution operations to emphasize the spatial characteristics of the receptive field. While effective in expanding contextual perception, this approach may introduce redundancy within the feature maps. To overcome this limitation, the proposed model integrates the strengths of RFAConv and GhostConv to develop a lightweight, plug-and-play convolutional unit named GRFAConv. The structural layout of the module is illustrated in Fig. 4.

Let the input feature map be $X \in R^{C \times H \times W}$. To avoid redundant feature extraction, a 1 × 1 convolution is first applied to adjust the number of output channels. These channels are then evenly divided into two parts using a Split operation. The first branch aggregates global contextual information through average pooling within each receptive field. A subsequent 1 × 1 group convolution further enhances feature interaction, followed by a SoftMax operation that assigns importance weights to different spatial positions. In parallel, the module employs a 3 × 3 grouped convolution to capture enhanced local contextual cues while maintaining computational efficiency. The grouped structure significantly reduces the number of parameters, whereas the 3 × 3 kernels ensure sufficient contextual extraction. A Rectified Linear Unit (ReLU) activation is applied to enforce unilateral suppression and stabilize training. The outputs of the two weighted branches are then fused through element-wise multiplication. After reshaping, a 3 × 3

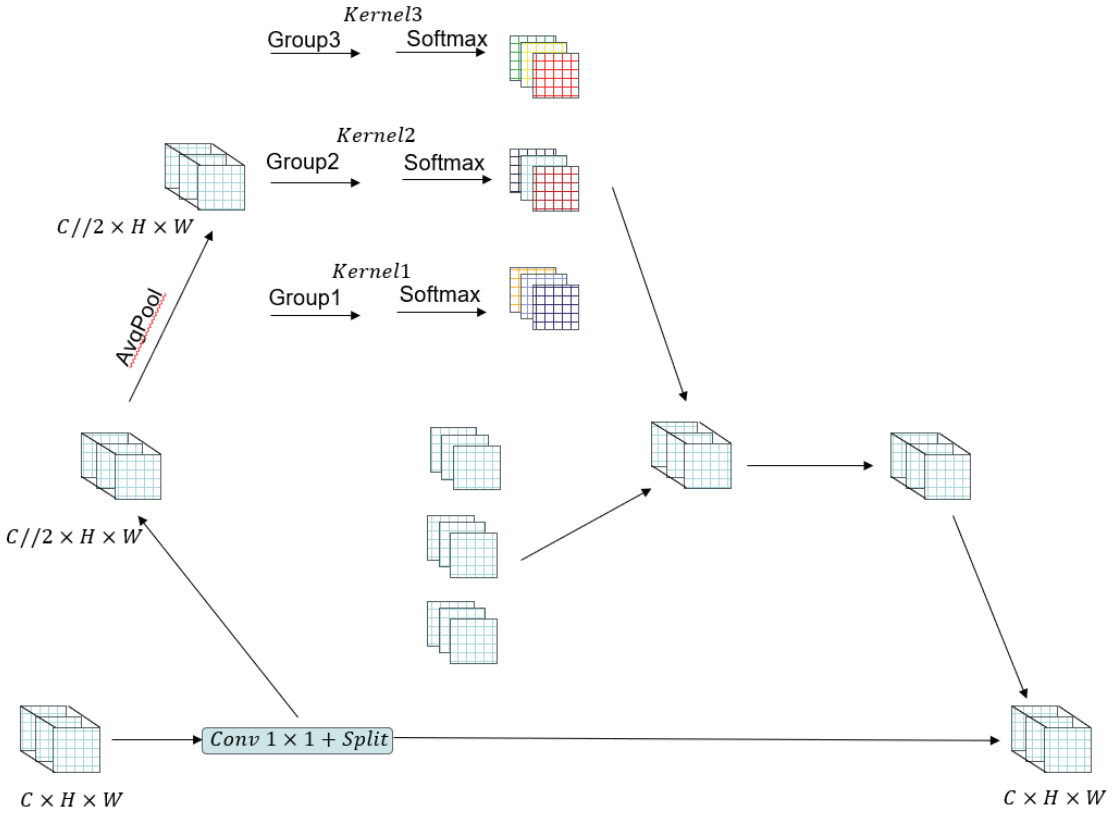


Fig. 4. (Color online) GRFAConv convolution structure diagram.

convolution is used for downsampling. Finally, the processed feature maps are concatenated with the previously divided feature channels along the channel dimension to form the final output. The overall GRFAConv computation process is formally described in Eqs. (4)–(6).

$$X_1, X_2 = \mathrm{Spilt}\left(\mathrm{Conv}^{1\times1}(X)\right) \tag{4}$$

$$F_1 = \mathrm{Softmax}\left(g^{1\times1}\left(\mathrm{AvgPool}(X_1)\right) \times \mathrm{Relu}\left(\mathrm{Norm}\left(g^{3\times3}(X_1)\right)\right)\right) \tag{5}$$

$$F_2 = X_2 \text{ and } F = \mathrm{Concat}\left(\mathrm{Conv}^{3\times3}\left(\mathrm{Adjust}(F_1)\right), F_2\right) \tag{6}$$

In Eq. (5), $g^{i\times i}$ denotes a multilayer convolution with a kernel size of $i \times i$ representing the normalization operation and X is the input feature map. The final output is obtained by concatenating the attention maps F_1 and F_2 along the channel dimension. Unlike CBAM and Coordinate Attention, which generate global or channel-level attention maps, the proposed GRFAConv module produces attention maps for each individual receptive-field region, enabling more fine-grained spatial modeling. Traditional convolution operations limit CNN performance because they rely on shared kernel parameters, making them insensitive to positional variations

within the receptive field. This restricts the network's ability to adapt to subtle spatial changes, an issue that is especially detrimental in small object detection. GRFAConv overcomes this limitation by emphasizing the spatial characteristics within the receptive field and assigning differentiated importance to features at different spatial positions within the sliding window. Furthermore, by incorporating the grouped processing strategy used in GhostConv, GRFAConv significantly reduces parameter overhead and computational cost while preserving representational richness. This combination of receptive-field-aware attention modeling and lightweight convolutional design enables GRFAConv to enhance context perception efficiently, making it well suited for real-time small object detection tasks.

In the feature fusion stage, traditional PANet structures require multiple rounds of upsampling and downsampling to merge features across scales.[17] However, such repetitive spatial transformations inevitably lead to semantic information loss. As a result, deep layers often fail to retain the semantic cues necessary for identifying small objects, while shallow layers lack sufficient contextual information. When the features of small targets are weakened or lost during fusion, detection performance deteriorates significantly. To alleviate these issues, in this study, we introduce a novel feature fusion network termed FDPN. The proposed FDPN leverages the FocusFeature module, which integrates multi-scale features from adjacent upper, lower, and same-level layers, thereby compensating for semantic degradation during fusion. Through a feature diffusion mechanism, each scale receives richer contextual information, effectively enhancing feature completeness. As illustrated in Fig. 5, the FDPN workflow proceeds as follows. First, FocusFeature aggregates the rich semantic representations from layers B3, B4, and B5 to generate the fused feature layer P4. Next, P4 serves as the central diffusion source, propagating contextualized information upward to P3 and downward to P5.

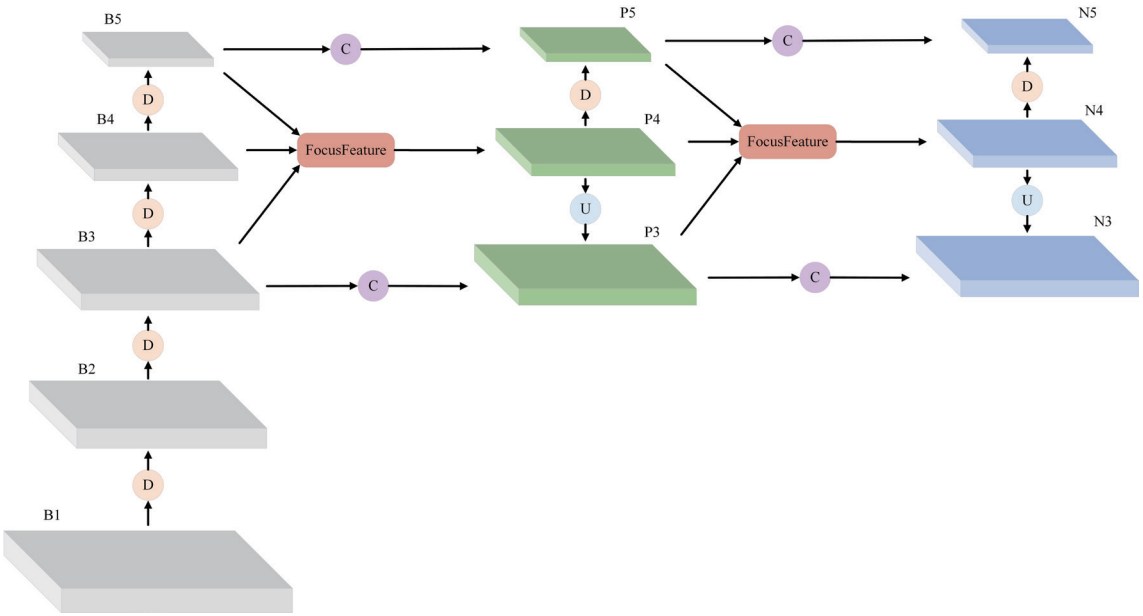


Fig. 5. (Color online) FDPN structure diagram.

The diffusion mechanism includes one upsampling and one downsampling operation for each propagation direction. Unlike traditional repeated sampling operations that risk feature loss, this design preserves more stable semantic consistency across scales.

Within FDPN, the FocusFeature module is responsible for cross-scale feature focusing and integration. By accepting inputs from three different resolutions and processing them through parallel convolutional paths, FocusFeature extracts richer semantic and contextual cues. Its architectural design is shown in Fig. 6. As depicted in Fig. 6, layer B3 is downsampled using an ADown convolution, layer B4 undergoes channel adjustment via a 1 × 1 convolution, and layer B5 is upsampled.[18] The outputs from these three branches are then concatenated along the channel dimension. To fully extract hierarchical features, three depthwise-separable convolutions with different kernel sizes are applied in parallel. Finally, a classic residual structure is used to stabilize training and enhance representation. The computation process of FocusFeature is defined by Eqs. (7) and (8), where Z denotes the module's final output.

$$Y = \mathrm{Concat}\left(\mathrm{ADown}^{3\times3}\left(B_3\right), \mathrm{Conv}^{1\times1}\left(B_4\right), \mathrm{Upsample}\left(B_5\right)\right) \tag{7}$$

$$Z = Y + \mathrm{Conv}^{1\times1}\left(\mathrm{DwConv}^{3\times3}\left(Y\right) + \mathrm{DwConv}^{5\times5}\left(Y\right) + \mathrm{DwConv}^{7\times7}\left(Y\right)\right) \tag{8}$$

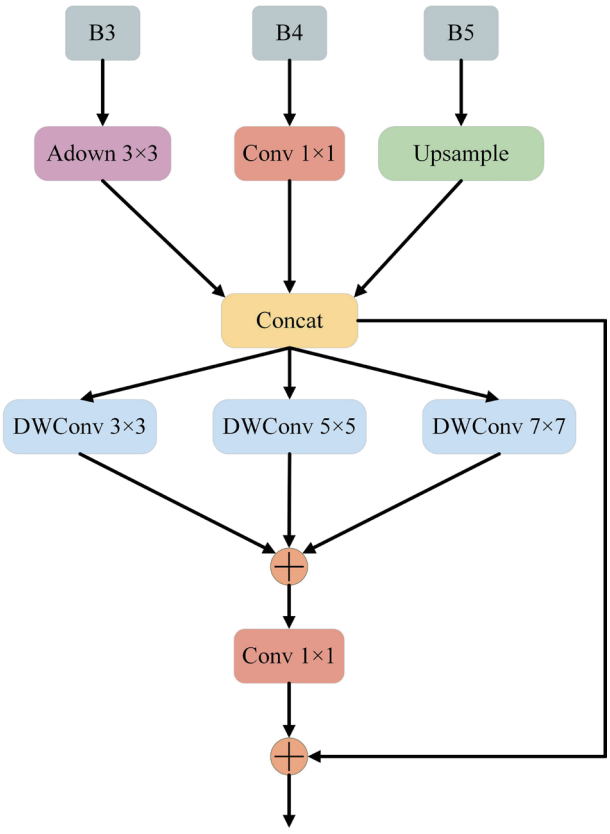


Fig. 6. (Color online) FocusFeature structure diagram.

3. Experiment Procedure

The experiments were conducted using the public VisDrone2019 dataset, collected by the research team at Tianjin University. This dataset contains 8599 static aerial images, including 6471 for training, 548 for validation, and 1580 for testing, all captured by drones operating at medium to high altitudes. VisDrone2019 provides highly diverse and challenging visual conditions essential for evaluating small object detection models. The dataset spans 14 cities and includes urban, suburban, and rural environments distributed across thousands of kilometers, introducing significant variability in illumination, background clutter, occlusion, and perspective distortion. The target categories such as bicycles, cars, trucks, buses, and pedestrians often appear as extremely small pixel regions owing to camera altitude and oblique viewing angles. Furthermore, scene density ranges from sparsely populated areas to heavily congested traffic zones, making VisDrone2019 a rigorous benchmark for assessing both detection accuracy and robustness in real-world UAV applications. All experiments were implemented on an NVIDIA RTX 4090 GPU running Linux, equipped with 60 GB of system memory. The training framework was based on Python 3.8.10, CUDA 11.3, and PyTorch 2.0.0. To ensure fairness and reproducibility across experiments, all models were trained under identical hyperparameter configurations.

The input resolution was set to 640 × 640, and each model was trained for 300 epochs with a batch size of 8, using 8 workers for data loading. The learning rate was initialized at 0.01, with a final learning rate factor (lrf) of 0.01, a momentum of 0.937, and a weight decay of 0.0005. The selected experimental configuration reflects a balance between computational feasibility and model performance. A resolution of 640 × 640 is commonly adopted in small object detection studies because it preserves fine-grained spatial details while maintaining acceptable training speed. Similarly, a batch size of 8 is well suited for high-resolution imagery, preventing GPU memory overflow while ensuring stable gradient updates. Maintaining consistent hyperparameters across all trials allows performance differences to be attributed directly to architectural innovations such as C2PLUS, GRFAConv, and FDPN rather than variations in training settings. Overall, this experimental setup provides a rigorous and controlled environment for evaluating the effectiveness of CGF-YOLOv11n on small object detection under realistic UAV scenarios.

The evaluation metrics used in this study include Precision, Recall, and mean Average Precision (*mAP*). In addition to accuracy-related indicators, it is essential to consider model complexity and computational efficiency, as these directly affect real-time performance, particularly in UAV-based small object detection tasks. Therefore, Floating Point Operations (GFLOPs) and the number of model parameters are also reported to comprehensively assess model efficiency. The formulas for the evaluation metrics are provided in Eqs. (9) and (10), where N denotes the total number of object categories in the dataset.

$$Precision = \frac{TP}{TP + FP} \text{ and } Recall = \frac{TP}{TP + FN} \tag{9}$$

$$AP = \int Precision\,and\,Recall \text{ and } mAP = \frac{1}{N}\sum_{1}^{N} AP_i \tag{10}$$

Precision measures the proportion of correctly predicted positive samples, where True Positives (*TP*) represent accurately identified objects, and False Positives (*FP*) correspond to incorrectly detected targets. Recall reflects the proportion of actual positive samples that are successfully detected by the model, with False Negatives (*FN*) indicating true objects that the model fails to identify. For overall detection quality, *mAP* is computed by averaging the *AP* scores across all categories, where AP_i denotes the *AP* value associated with class *i*. Including both accuracy-based and efficiency-oriented metrics is crucial for a fair and meaningful comparison. Small object detection models often face a trade-off: improving accuracy may increase computational cost, while reducing model size may degrade detection performance. By jointly evaluating *mAP*, Precision, Recall, GFLOPs, and parameter count, in this study, we provide a balanced perspective on how the proposed CGF-YOLOv11n architecture enhances small object detection without compromising real-time capability. This multifaceted evaluation approach ensures that improvements are not limited to accuracy alone but extend to computational practicality, an essential requirement for real-world deployment on embedded and edge-computing platforms.

4. Results and Discussion

To evaluate the effectiveness of the enhanced algorithmic modules proposed in this study, ultralytics-YOLOv11n was selected as the baseline model. A comprehensive ablation study was conducted using key metrics including GFLOPs, the number of parameters, *mAP*@0.5, Recall, and Precision. Multiple combinations of the proposed modules were tested to assess their individual and joint contributions. The statistical results of all ablation configurations are summarized in Table 1, where A, B, and C denote the progressive improved versions derived from the baseline. Model A incorporates the C2PLUS module into the baseline architecture. Despite introducing only negligible increases in computational complexity and parameter count, Model A demonstrates a substantial improvement in detection accuracy across both the validation and test sets. This confirms that C2PLUS significantly enhances feature extraction for small objects while maintaining lightweight characteristics, providing a strong foundation for subsequent architectural improvements. Model B extends Model A by integrating the proposed GRFAConv convolution module. Although GRFAConv incurs a slightly higher computational cost than a standard convolution, it yields notable performance gains.

Table 1
Ablation tests using the VisDrone2019 dataset as a validation set.

Model	C2PLUS	GRFAConv	FDPN	*Precision*	*Recall*	*mAP*@0.5	Parameters/M	GFLOPs
YOLOv11n				44.5	33.6	33.4	2.584	6.4
A	✓			45.1	35.1	34.3	2.573	6.4
B	✓	✓		46.0	36.2	35.8	2.594	6.6
C	✓	✓	✓	46.7	37.0	36.9	2.745	7.2

On the validation set, Model B achieves improvements of 2.1% in Precision, 1.1% in Recall, and 1.5% in *mAP*@0.5 relative to Model A. These results demonstrate that GRFAConv effectively strengthens receptive-field modeling and spatial feature interaction, leading to a more discriminative feature representation. Model C further enhances Model B by replacing the original YOLOv11n FPN with the proposed FDPN structure. The FDPN introduces only minimal increases in parameters and GFLOPs, yet it significantly improves multi-scale feature fusion. As shown in Tables 1 and 2, Model C achieves an additional 0.8% increase in *mAP*@0.5 on the test set and 1.1% on the validation set, indicating that FDPN effectively mitigates semantic loss during feature integration and enhances the stability of small object detection. The final improved model, integrating all three modules, is named CGF-YOLOv11n. Subsequent evaluations on embedded hardware platforms further confirm its suitability for UAV-based and perspective-view detection tasks, demonstrating both enhanced accuracy and practical deployability. Overall, the ablation results validate that each proposed module contributes meaningful performance improvements while maintaining computational efficiency. The consistent gains across both validation and test sets on the VisDrone2019 dataset confirm the robustness and effectiveness of the CGF-YOLOv11n architecture.

To further validate the detection performance of the enhanced model, a comparative study was conducted using several mainstream object detection algorithms, including YOLOv3-tiny,[19] YOLOv5n,[20] YOLOv7,[21] YOLOv7-tiny,[22,23] YOLOv8n,[24] YOLOv10,[25] YOLOv11n,[26] YOLOv12,[27] YOLOX-Tiny,[28,29] as well as the benchmark YOLOv11n model. Table 3 presents the performance comparison between the original YOLOv11n and the proposed CGF-YOLOv11n on the VisDrone2019 test set. Experimental results show that although CGF-YOLOv11n introduces slight increases in parameters and computational load, it achieves a

Table 2
Ablation tests using the VisDrone2019 dataset as test sets.

Model	C2PLUS	GRFAConv	FDPN	*Precision*	*Recall*	*mAP*@0.5	Parameters/M	GFLOPs
YOLOv11n				36.5	28.6	27.0	2.584	6.4
A	✓			37.3	29.1	27.7	2.573	6.4
B	✓	✓		38.0	30.2	29.3	2.594	6.6
C	✓	✓	✓	39.7	30.5	30.1	2.745	7.2

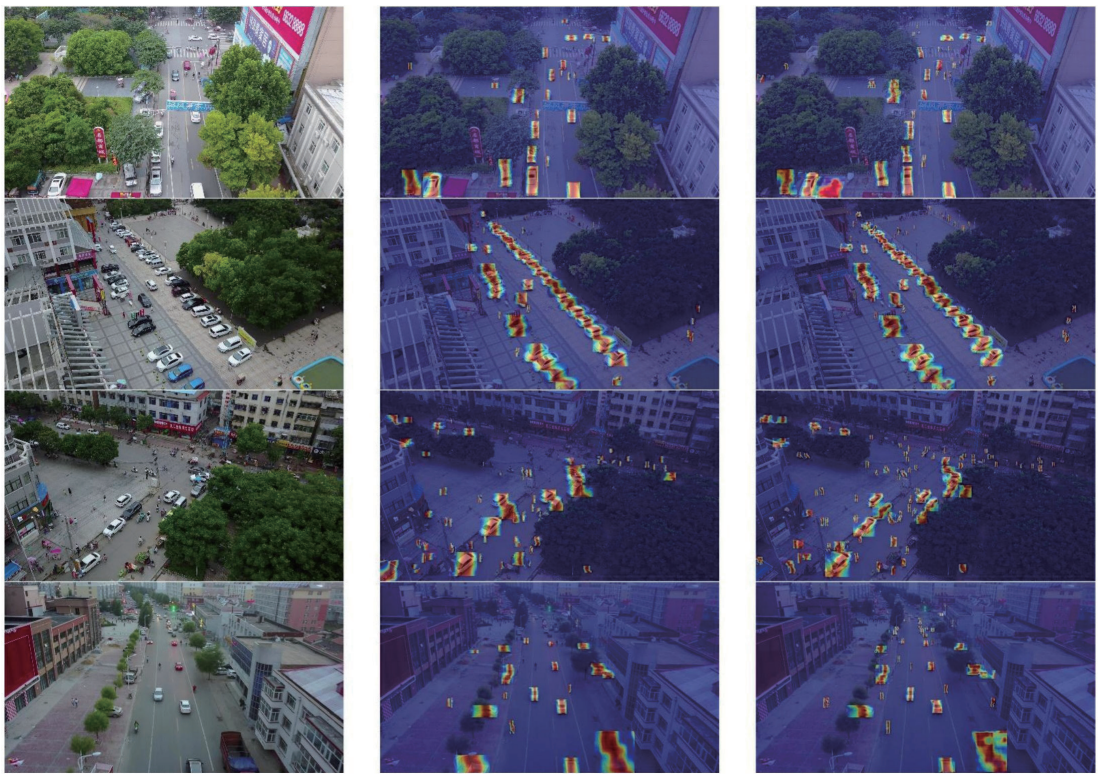
Table 3
VisDrone2019 dataset's object detection outcomes using several algorithms.

Model	Parameters/M	GFLOPs	*mAP*@0.5
YOLOv3-tiny	12.2	19.0	23.1
YOLOv5n	1.9	4.5	24.5
YOLOv7-tiny	5.9	13.2	25.3
YOLOv8n	3.0	8.1	25.9
YOLOv10n	2.3	6.5	26.1
YOLOv11n	2.6	6.4	27.0
YOLOv12n	2.6	6.3	25.9
YOLOX-Tiny	5.0	7.6	27.8
CGF-YOLOv11n	2.7	7.2	30.1

significant improvement in detection accuracy, outperforming all compared methods. This demonstrates that the proposed architecture effectively enhances small object detection while maintaining a lightweight design suitable for real-time applications.

The performance gains can be attributed to the contributions of the three improved modules. The C2PLUS module enhances fine-grained feature extraction with minimal computational overhead, producing richer feature representations essential for small object detection. The plug-and-play GRFAConv module reduces redundant information and accelerates network convergence, enabling the model to focus more effectively on critical target regions. Additionally, the FDPN structure diffuses semantically enriched features across multiple scales, substantially improving the model's capability to detect small targets under varying perspectives, an important advantage for UAV-based detection tasks. Overall, the enhanced CGF-YOLOv11n model delivers superior performance compared with existing lightweight and standard detection architectures. Its improvements in accuracy, robustness, and multi-scale feature representation confirm its effectiveness for small object detection in complex aerial scenarios.

To intuitively evaluate the effectiveness of the proposed model, heatmap visualization was first employed to illustrate the distribution of attention across different image regions. In such visualization, warmer colors (redder regions) indicate a stronger contribution to classification, whereas cooler colors (bluer regions) indicate a weaker contribution. Figure 7 shows a

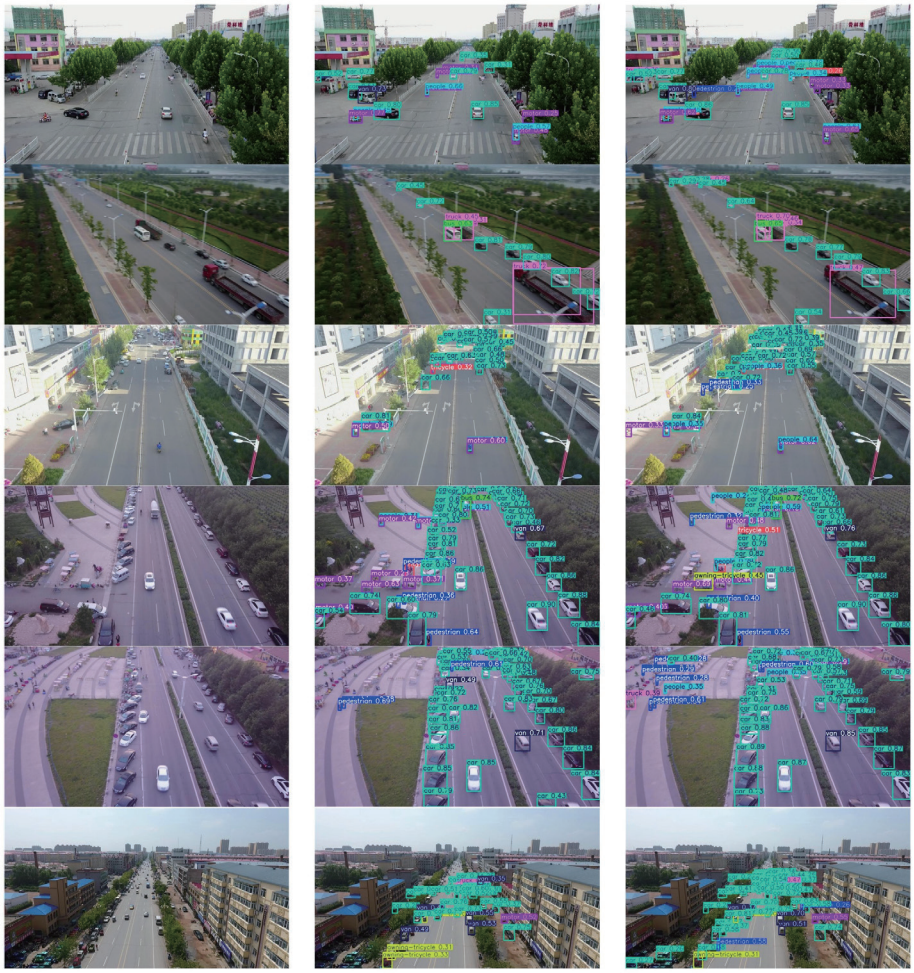


(a) (b) (c)

Fig. 7. (Color online) Heatmap comparison.

comparative heatmap visualization of different models; Fig. 7(a) corresponds to the original image, whereas Figs. 7(b) and 7(c) show the heatmaps generated by the baseline YOLOv11n and the proposed CGF-YOLOv11n, respectively. Both YOLOv11n and CGF-YOLOv11n are able to focus on the primary target regions; however, CGF-YOLOv11n exhibits significantly stronger and more concentrated attention on small objects. In the first scene, YOLOv11n only attends to a limited number of vehicles. In the second and third scenes, the baseline model mainly focuses on relatively larger objects in the near field, whereas the proposed model is able to attend to small targets located in distant regions. In the fourth scene, even under low-light conditions, the improved model successfully identifies a greater number of small objects at longer distances. These comparisons demonstrate that CGF-YOLOv11n achieves superior detection capability compared with the baseline model, highlighting the effectiveness of the proposed architectural enhancements in strengthening feature extraction and sensitivity to small target regions.

Next, Fig. 8 presents detection results of YOLOv11n and CGF-YOLOv11n across various challenging environments to further demonstrate the advantages of the enhanced model. Figure



(a) (b) (c)

Fig. 8. (Color online) Comparison of detection outcomes across several situations.

8 also shows the detection results of the baseline YOLOv11n and the proposed CGF-YOLOv11n on six representative scenarios from the VisDrone2019 dataset, where Fig. 8(a) corresponds to the original images, Fig. 8(b) shows the results of YOLOv11n, and Fig. 8(c) shows the results of CGF-YOLOv11n. In the first row of images, where numerous small objects such as motorcycles and pedestrians appear, CGF-YOLOv11n detects substantially more targets than YOLOv11n, thereby reducing missed detections. In the first three scenarios, the proposed model successfully detects low-angle vehicles, pedestrians, and trucks, whereas the baseline model struggles with these targets. In the fourth and fifth scenarios, CGF-YOLOv11n is able to identify smaller pedestrians and trucks located on the left side of the images, whereas in the sixth scenario, it detects an even smaller vehicle in the lower-left region. By contrast, YOLOv11n shows limited capability in detecting such small-scale targets. Moreover, compared with YOLOv11n, CGF-YOLOv11n achieves comparable confidence levels when detecting large and medium-sized objects, demonstrating that the proposed improvements do not compromise robustness on larger targets. These results highlight the strong potential of CGF-YOLOv11n for high-precision applications across a wide range of small-object-dominated scenarios.

In the third example of the third column, the enhanced model successfully identifies a motorcycle located under intense illumination on the far left, which YOLOv11n fails to detect. Similarly, in the fourth example, only a partially visible tractor appears at the left edge of the image. YOLOv11n misses this object entirely, whereas CGF-YOLOv11n correctly identifies it, demonstrating superior robustness in scenarios involving occlusion and partial visibility. Overall, the visualization results clearly show that CGF-YOLOv11n exhibits stronger generalization ability and improved adaptability across diverse environments, including scenes with large numbers of small objects, complex lighting variations, and cluttered backgrounds. Compared with YOLOv11n, the enhanced model not only detects significantly smaller targets but also maintains similar confidence levels for large and medium-sized objects, highlighting its robustness. These findings collectively indicate that CGF-YOLOv11n holds substantial potential for high-precision small-object detection in real-world UAV and aerial imaging applications.

5. Conclusions

In this study, we proposed an enhanced small object detection model, CGF-YOLOv11n, to address the challenges of missed and erroneous detections commonly encountered in small target recognition. First, the model integrates transformer-based self-attention with traditional CNN attention mechanisms, strengthening multi-scale feature fusion within the neck structure. Second, a lightweight, plug-and-play convolution module, GRFAConv, was designed by combining the principles of RFAConv and GhostConv, effectively improving receptive-field modeling while maintaining computational efficiency. Finally, a novel diffusion pyramid network, FDPN, was introduced to mitigate feature loss through a feature diffusion process, ensuring more robust semantic propagation across scales. Experimental results on the VisDrone2019 dataset demonstrate that the proposed framework significantly improves small object detection performance and consistently outperforms other benchmark models. Although CGF-YOLOv11n achieves notable gains in accuracy, there remains room for further optimization

in terms of parameter count and computational complexity. Future work will focus on additional lightweight model compression strategies to enhance detection speed without compromising accuracy. Moreover, expanding evaluation to include diverse real-world datasets will help assess and strengthen the generalization capability of the proposed model in practical deployment scenarios.

Acknowledgments

This research was funded by the 2023 Annual Project for Quality Assurance and Teaching Reform of Undergraduate Universities in Guangdong Province and the National-level College Students' Innovation and Entrepreneurship Training Project (No. 202510580014).

References

1 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (Las Vegas, NV, USA) (2016) 779–788.
2 S. Li, S. Li, and G. Liu: Small Micro Comput. Syst. **45** (2024) 2165.
3 S. Woo, J. Park, J. Y. Lee, and I. S. Kweon: CBAM: Convolutional Block Attention Module, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. (Springer, Berlin, 2018).
4 J. Hu, L. Shen, and G. Sun: 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition (Salt Lake City, UT, USA) (2018) 7132–7141.
5 M. Tan, R. Pang, and Q. V. Le: 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (Seattle, WA, USA) (2020) 10778–10787.
6 Y. Hao, W. Luo, Y. Li, B. Zhang, and J. Bei: Proc. SPIE 12508, Int. Sym. Artificial Intelligence and Robotics 2022, 125080H.
7 M. Li, Y. Chen, T. Zhang, and W. Huang: Complex Intell. Syst. **10** (2024) 5459.
8 M. M. Rahman, M. Munir, and R. Marculescu: Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (Seattle, WA, USA) 11769–11779.
9 A. Gomaa and O. M. Saad: Multimed. Tools Appl. **1** (2025) 1.
10 J. Redmon and A. Farhadi: 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (Honolulu, HI, USA) (2017) 6517–6525.
11 Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye: Proc. IEEE **111** (2023) 257.
12 https://accautomation.ca/click-plus-c2-nred-easy-install-for-plc-module/ (accessed March 2025).
13 K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu: 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (Seattle, WA, USA) (2020) 1577–1586.
14 M. Hussain and R. Khannam: Solar **4** (2024) 351.
15 D. Ouyang, S. He, G. Zhang, M. Luo, H. Guo, J. Zhan, and Z. Huang: ICASSP 2023 - 2023 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP) (Rhodes Island, Greece) (2023) 1–5.
16 Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu: 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (Seattle, WA, USA) (2020) 11531–11539.
17 S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia: 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition (Salt Lake City, UT, USA) (2018) 8759–8768.
18 C. Y. Wang, I. H. Yeh, and H. Y. M. Liao: Computer Vision – ECCV 2024. ECCV 2024. Lecture Notes in Computer Science, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds. (Springer, Berlin, 2018) Vol. 15089.
19 L. Fu, Y. Feng, J. Wu, Z. Liu, F. Gao, Y. Majeed, A. Al-Mallahi, Q. Zhang, R. Li, and Y. Cui: Precis. Agric. **22** (2021) 754.
20 https://zenodo.org/records/5563715 (accessed Oct. 2021).
21 C. -Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao: 2023 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (Vancouver, BC, Canada) (2023) 7464–7475.
22 L. Ma, L. Zhao, Z. Wang, J. Zhang, and G. Chen: Agronomy **13** (2023) 1419.
23 J. Feng, Q. An, J. Zhang, S. Zhou, G. Du, and K. Yang: 2024 5th Int. Seminar on Artificial Intelligence, Networking and Information Technology (AINIT) (Nanjing, China) (2024) 2241–2245.

24 Z. Wang, Z. Hua, Y. Wen, S. Zhang, X. Xu, and H. Song: Expert Syst. Appl. **238** (2024) 122212.
25 H. Chen, K. Chen, G. Ding, J. Han, Z. Lin, L. Liu, and A. Wang: Advances in Neural Information Processing Systems 37 (10–15 December 2024, Vancouver, Canada) 107984–108011.
26 C. Wen, Y. Cheng, S. Li, L. Liu, Q. Liang, K. Li, and Y. Huang: Appl. Sci. **15** (2025) 4286.
27 M. Chileshe, M. Nyirenda, and J. Kaoma: Open J. Appl. Sci. **15** (2025) 19.
28 J. Lin, D. Yu, R. Pan, J. Cai, J. Liu, L. Zhang, X. Wen, X. Peng, T. Cernava, S. Oufensou, Q. Migheli, X. Chen, and X. Zhang: Front. Plant Sci. **14** (2023) 1135105.
29 Y. She and H. Li: J. Intell. Manuf. **36** (2025) 2142.