ISSN: 2664-9640

YOLO-GC: A Lightweight Model for Remote Sensing Image Object Detection

Zhongren Liang[a]

School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China

[a]2936721262@qq.com

Abstract

Remote sensing images are crucial national strategic resources. However, existing high-performance object detection models for remote sensing images typically suffer from high computational complexity and large parameter sizes. To reduce computational demands while maintaining detection accuracy for multi-scale targets, this study proposes YOLO-GC, a lightweight object detection model for remote sensing images. First, the conventional Conv modules in the YOLOv8 backbone network are replaced with GhostConv, and the C2f module is enhanced into a C2f-GhostConv structure integrating Ghost operations, significantly reducing model complexity. Meanwhile, a CBAM attention layer is incorporated into the backbone network, effectively improving feature extraction capability without increasing computational overhead. Experimental results on the NWPU VHR-10 remote sensing object detection dataset demonstrate that YOLO-GC achieves a computational complexity of 21.4 GFLOPs, reducing the baseline YOLOv8s' complexity by 7.4 GFLOPs, while maintaining a mAP@50 of 0.922-only 0.011 lower than the baseline. YOLO-GC achieves a superior balance between accuracy and efficiency, significantly enhancing its potential for deployment on resource-constrained edge platforms.

Keywords

Remote Sensing Images; Object Detection; YOLOv8; Lightweight; GhostConv; CBAM.

1. Introduction

As a core component of national aerospace information infrastructure, remote sensing images play an irreplaceable strategic role in critical domains such as national security surveillance, resource and environmental monitoring, disaster emergency response, and smart city construction-all of which are vital to national security and socio-economic development [1]. With the rapid advancement of high-resolution Earth observation technology, the information contained in massive remote sensing data has become increasingly rich. Accurately and efficiently extracting key target information, such as military installations, vehicles, and infrastructure, has emerged as a crucial prerequisite for intelligent decision-making in these fields [2].

Traditional remote sensing object detection primarily relies on two technical approaches: (1) Manual interpretation methods, which are inefficient and highly subjective, struggle to meet the demands of the big data era; (2) Machine learning-based methods employing handcrafted feature extraction typically use sliding windows to traverse the entire image, followed by feature extraction for each window (e.g., Haar features, Scale-Invariant Feature Transform (SIFT), and Histogram of Oriented Gradients (HOG)) [3][4], and finally classification via classifiers such as SVM [5] or random forest [6]. Although these methods achieve partial automation, the exhaustive sliding window approach leads to excessively high time complexity,

ISSN: 2664-9640

while designing highly robust features remains challenging, resulting in significant accuracy limitations [2].

With breakthroughs in deep learning, convolutional neural network (CNN)-based object detection algorithms have demonstrated remarkable advantages. Among them, two-stage detectors reduce redundant sliding window computations by generating region proposals [7], followed by CNN-based feature extraction and classification. Representative models include R-CNN [8], Fast R-CNN [9], and Faster R-CNN [10]. While such region proposal mechanisms achieve high accuracy, the candidate region generation introduces substantial computational overhead and slow inference speeds, making real-time processing difficult. In contrast, single-stage detectors adopt an end-to-end architecture, eliminating the need for a region proposal stage and directly predicting target class probabilities and bounding box coordinates. Compared to two-stage detectors, they offer faster detection speeds, with representative models including the YOLO series [11], SSD [12], and CenterNet [13].

Current research on remote sensing object detection primarily focuses on improving accuracy by incorporating attention mechanisms or complex feature fusion modules. For instance, Ju Moran et al. [14] enhanced YOLOv3 by upsampling and concatenating feature maps while adding residual modules, significantly improving small object detection with a 6.55% increase in mAP-albeit at the cost of higher computational load. Wang Xikun et al. [15] augmented YOLOv3-Tiny with feature mapping and residual connection modules, increasing ship detection accuracy by 9.44% but reducing speed by 1.5 FPS. Tang Jianyu et al. [16] improved YOLOv5 by enlarging feature map resolution and quantity while integrating the CBAM attention mechanism, boosting mAP by 8.06% but neglecting model complexity considerations. Zhang Shaowen et al. [17] enhanced YOLOX-S with CBAM, a weighted multi-receptive field spatial pyramid pooling module, and a cross-layer attention fusion module, achieving a 5.1% APs improvement but increasing parameters by 1.01M. Yan Junhua et al. [18] proposed CC-YOLO, which incorporated a Coordinate Attention (CA) mechanism and cross-level channel feature fusion to achieve 94.6% AP50.

However, remote sensing object detection still faces several challenges: wide variations in target scales, dense arrangements, complex backgrounds, susceptibility to small object information loss, viewpoint changes, cloud occlusion, and illumination variations [19][20], all of which impose stringent demands on model robustness and generalization. Most critically, the prevailing "accuracy at the cost of computation" strategy in existing research contradicts the core requirements of edge devices-low power consumption, compact size, and real-time performance-further exacerbating model bloat.

To address these challenges, this study proposes YOLO-GC, a lightweight remote sensing object detection model tailored for edge computing. Based on YOLOv8s, it achieves an optimal balance between accuracy and efficiency through two key optimizations: (1) Ghost Module Integration: The backbone network incorporates GhostNet's GhostConv to replace standard convolutions, and the C2f module is upgraded to C2f_GhostConv, significantly reducing computational complexity and enhancing deployability. (2) CBAM Attention Mechanism: An efficient multi-scale CBAM attention layer is embedded into the backbone network to strengthen feature focus on small and densely arranged targets in complex backgrounds. Compared to existing approaches, the proposed model demonstrates comprehensive advantages in accuracy retention, computational efficiency, and edge deployment, providing technical support for real-time onboard/airborne intelligent processing.

ISSN: 2664-9640

2. Materials and Methods

2.1. YOLOv8 Deep Learning Model

The YOLOv8 model was released by Ultralytics in 2023 as an improved version of its predecessors including YOLOv5 and YOLOv7 [21][22], and serves as the baseline model in this study. The network architecture of YOLOv8 primarily consists of three components: the Backbone, Neck, and Head, as shown in Figure 1.

The Backbone is responsible for feature extraction, progressively extracting multi-scale semantic features from raw images to construct fundamental feature maps, serving as the core module of the model. The Neck performs feature fusion, integrating multi-scale feature maps from the Backbone to enhance the interaction between semantic information and spatial details. The Head makes prediction decisions, executing object localization and classification tasks based on the fused features.

The Backbone mainly contains Conv, C2f, and SPPF modules. The Conv module is a composite convolutional block incorporating batch normalization and activation functions. The C2f module represents a core innovation of YOLOv8 compared to previous versions, significantly improving feature reuse efficiency through multi-branch cross-layer connections and enhanced gradient flow. It divides feature maps into a main branch and a secondary branch, where the secondary branch directly transmits to preserve original information while the main branch extracts deep features through multiple Bottleneck layers, finally concatenating all branches to enhance multi-scale perception. The SPPF module performs three consecutive 5×5 max pooling operations to capture multi-scale features.

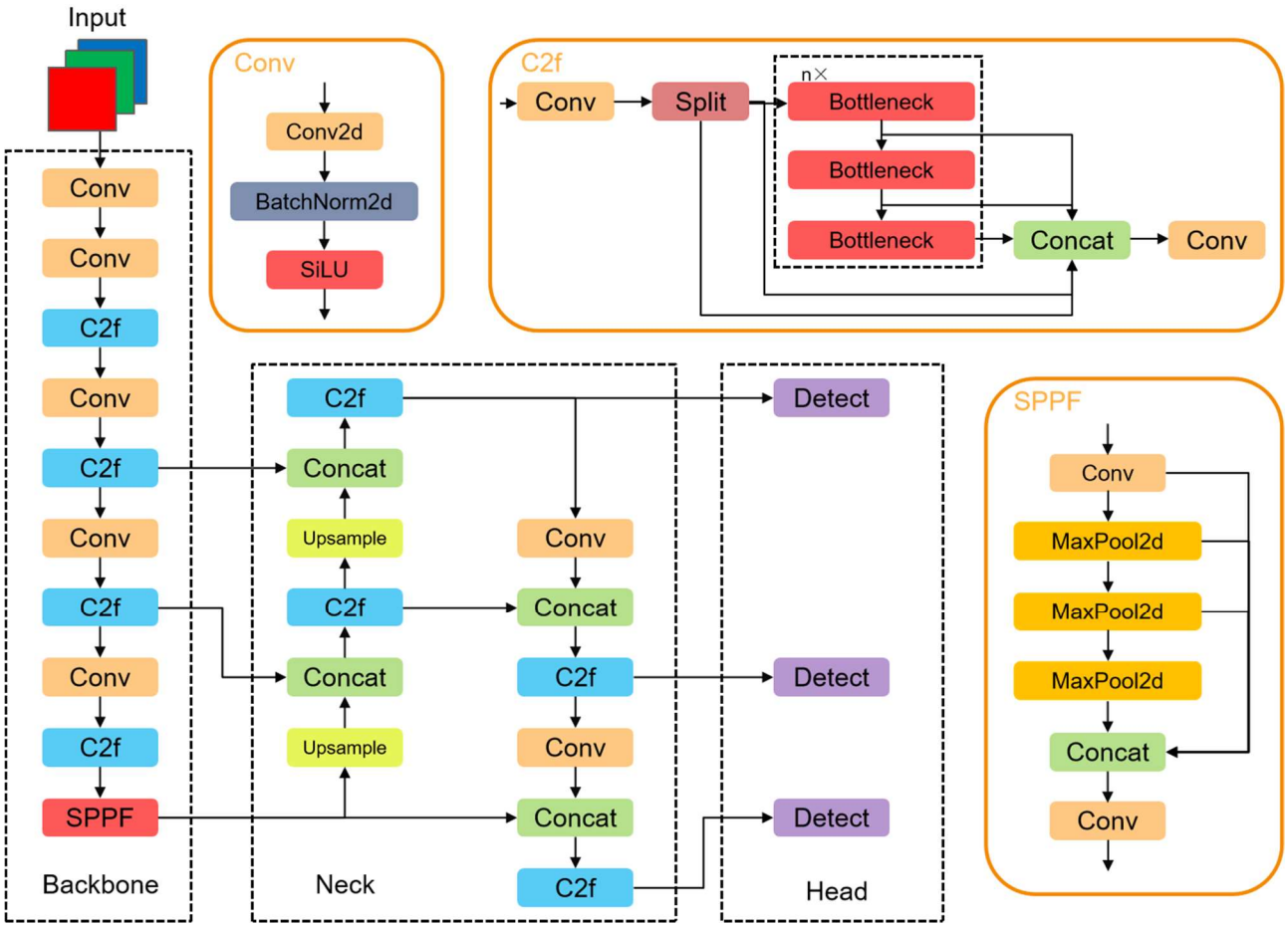

Figure 1. Network architecture of the YOLOv8 model.

ISSN: 2664-9640

The Neck employs Path Aggregation Network (PAN) and Feature Pyramid Network (FPN) structures to conduct multiple upsampling, downsampling, and concatenation operations, further fusing the multi-scale features extracted by the Backbone and enhancing the model's detection capability for targets of different sizes.

For the Head component, YOLOv8 adopts a decoupled head instead of YOLOv5's coupled head, enabling classification and regression tasks to be handled by separate branches. Additionally, it utilizes anchor-free detection instead of the previous anchor-based approach, both of which contribute to improved detection speed and accuracy.

2.2. YOLO-GC Model Architecture

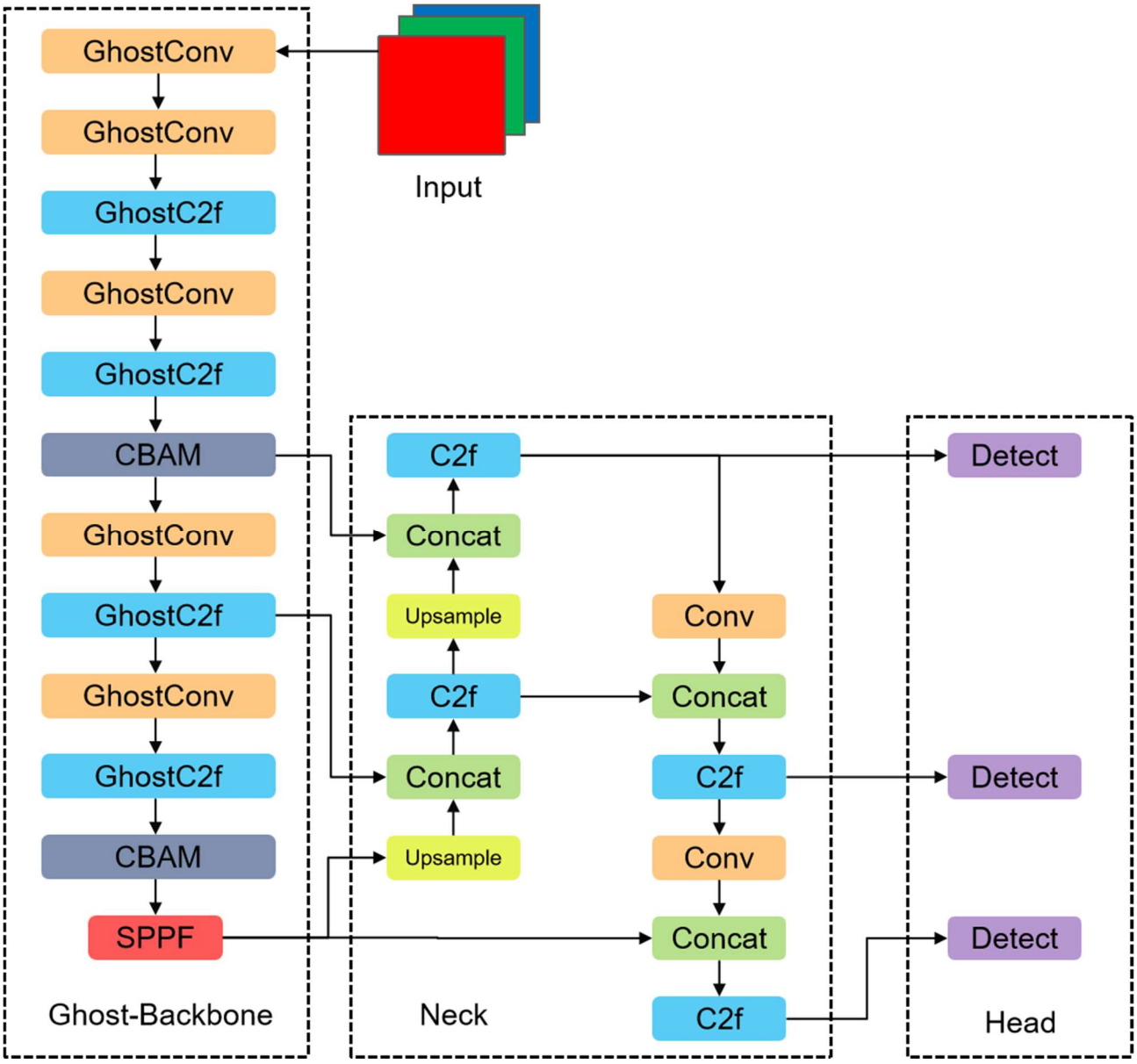

Figure 2. Network architecture of the YOLO-GC model.

Compared with general object detection tasks, remote sensing image object detection exhibits the following distinctive characteristics: 1) Extreme scale variations among detection targets. The size difference between objects like cars and football fields can span several orders of magnitude, yet they may appear in the same remote sensing image, while the same object may vary in size across different images due to varying capture altitudes. 2) Frequent illumination changes and occlusions. Affected by natural conditions, the lighting environment cannot be

ISSN: 2664-9640

consistent across all remote sensing images, and occlusions caused by clouds and buildings cannot be ignored. These challenges impose stringent requirements on model robustness and detection accuracy. On the other hand, remote sensing image acquisition and processing are typically performed by drones and satellites, which inherently have limited computational power, placing significant demands on model lightweighting.

To address these challenges, this study proposes YOLO-GC, a lightweight yet high-accuracy remote sensing image object detection model based on YOLOv8, with the following improvements to the YOLOv8 backbone network:

(1) To achieve model lightweighting, the Ghost module from GhostNet is introduced, replacing the original Conv and C2f modules with GhostConv and C2f_GhostConv modules respectively, significantly reducing computational complexity while maintaining high accuracy.

(2) To further enhance the model's capability for effective feature extraction and improve robustness, the CBAM attention mechanism is incorporated, mitigating the impacts of scale variations, illumination changes, and occlusions, thereby further improving accuracy.

The architecture of the YOLO-GC model is shown in Figure 2.

2.2.1. Ghost-Backbone Improvement

In the field of real-time remote sensing image object detection, computational complexity is a critical factor when deploying models on edge computing platforms with limited computational resources. Lower computational complexity enhances the deployability of models on such platforms. To reduce computational complexity while maintaining high accuracy, this study introduces the Ghost module from GhostNet [23], constructing a Ghost-Backbone composed of GhostConv and C2f_GhostConv modules to replace the original backbone network.

GhostNet, released by Huawei Noah's Ark Lab in 2019, is a lightweight network where the Ghost module serves as its core component. The key idea is to reduce computational load by generating ghost feature maps through cheap operations. Specifically, the input feature maps first undergo a convolution operation to produce a small number of intrinsic feature maps. Then, each intrinsic feature map undergoes linear cheap operations to generate multiple ghost feature maps. Finally, the intrinsic and ghost feature maps are concatenated to form the output feature maps. This approach splits the original single convolution into two stages, both with significantly reduced computational load, thereby greatly decreasing the model's overall computational complexity.

In the Ghost-Backbone improvement, the Ghost module concept is first applied to replace all Conv modules with GhostConv modules. The process is illustrated in Figure 3, and the structure is shown in Figure 4. For an input $X \in \mathbb{R}^{c\times h\times w}$ and output $Y \in \mathbb{R}^{h'\times w'\times n}$, where c is the number of input channels, h and w are the image dimensions, and n is the number of output channels, the first convolution operation on the input can be expressed as:

$$Y' = X * f' + b, \tag{1}$$

where $Y' \in \mathbb{R}^{h'\times w'\times m}$, $f' \in \mathbb{R}^{c\times k\times k\times m}$, m is the number of intrinsic feature maps (m≤n), k is the kernel size, and b is the bias term. This yields m intrinsic feature maps. The remaining feature maps in the output channels are considered ghost feature maps of these intrinsic maps and can be obtained through linear cheap operations:

$$y_{ij} = \Phi_{i,j}(y_i'), \forall i = 1,\ldots,m, \qquad j = 1,\ldots,s-1, \tag{2}$$

ISSN: 2664-9640

where y_i' is the i-th intrinsic feature map in Y'; $\Phi_{i,j}$ is the linear operation (here, a 5×5 convolution) that generates the j-th ghost feature map from y_i'; and s is the compression ratio (input channels/intrinsic feature maps). The final output feature maps are obtained by concatenating intrinsic and ghost feature maps:

$$Y = [y_1', \ldots, y_m', y_{11}, y_{12}, \ldots, y_{m(s-1)}], \tag{3}$$

Using the speed-up ratio r_s as a metric, we can derive that conventional convolution requires s times more computation than the Ghost module:

$$\begin{aligned} r_s &= \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d} \\ &= \frac{c \cdot k \cdot k}{\frac{1}{s} \cdot c \cdot k \cdot k + \frac{s-1}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s, \end{aligned} \tag{4}$$

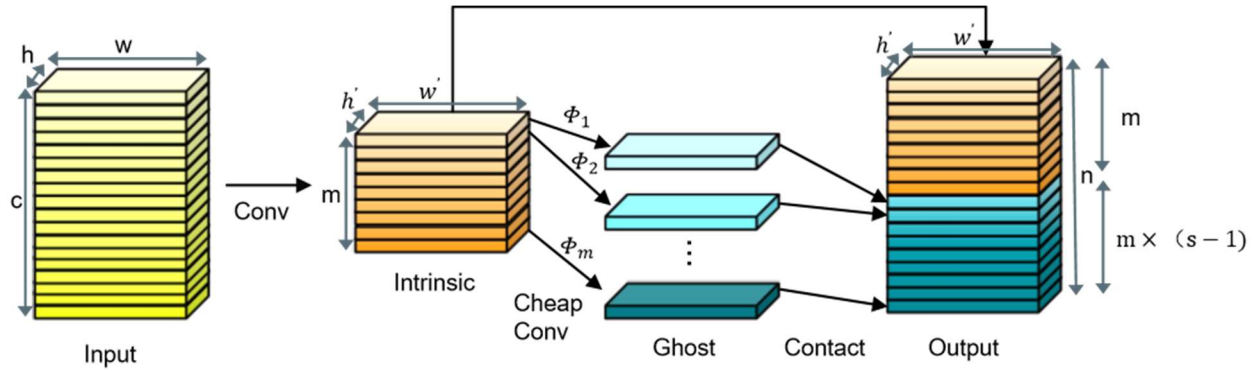

Figure 3. Flowchart of Ghost Conv.

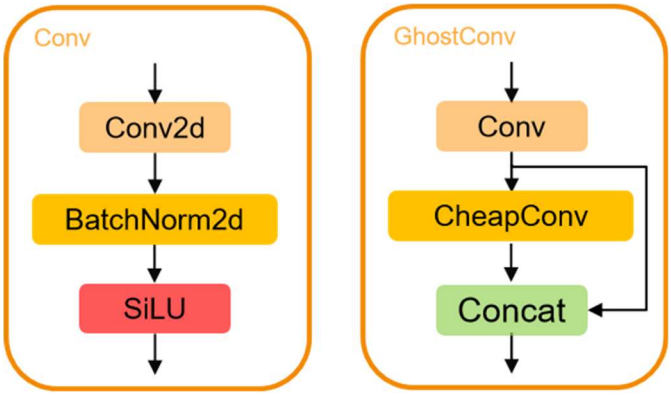

Figure 4. Structure comparison between Conv and GhostConv.

Additionally, all C2f modules are replaced with C2f_GhostConv modules. The original C2f contains multiple convolution operations, particularly in its Bottleneck modules. Therefore, C2f_GhostConv replaces all Bottleneck modules with GhostConv to further reduce computation. To compensate for potential accuracy loss due to cheap operations and the absence of Bottleneck structures, the convolution operations in GhostConv are replaced with DynamicConv, which captures richer texture details through multi-expert combinations. Although this increases parameters, it enhances model expressiveness with negligible additional computational overhead. The structure is shown in Figure 5.

ISSN: 2664-9640

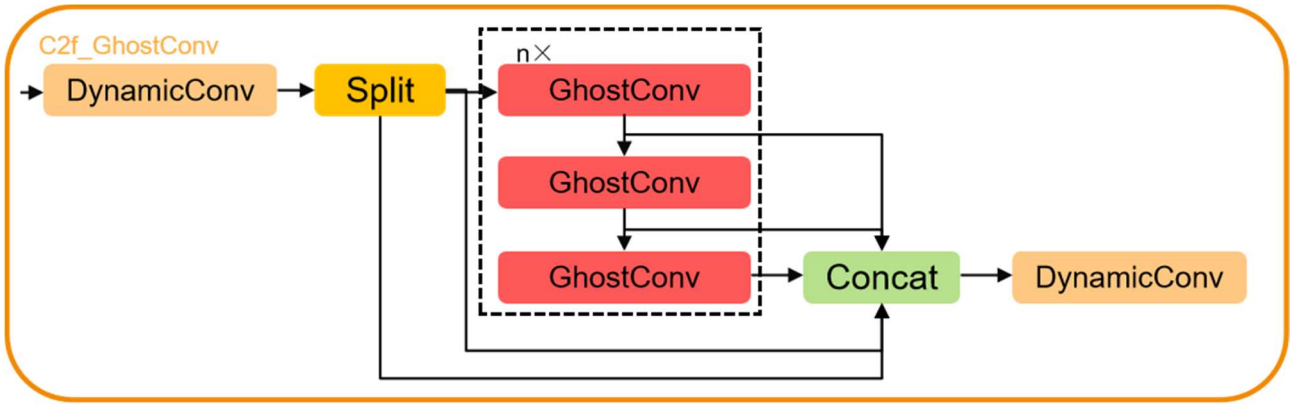

Figure 5. Structure of C2f_GhostConv.

2.2.2. CBAM Attention Mechanism

While the Ghost module's cheap operations and the structural simplifications in C2f_GhostConv significantly reduce computational load, they also lead to some accuracy degradation. Additionally, remote sensing images present unique challenges: extreme scale variations between targets make feature extraction difficult; unavoidable obstructions from clouds and buildings often cause missed detections; and high similarity between certain targets increases false detection rates. These factors demand enhanced feature extraction capabilities from the model.

To improve the model's representational capacity with minimal computational overhead, we introduce the Convolutional Block Attention Module (CBAM) proposed by Sanghyun Woo et al. [24] in 2018. CBAM is a lightweight yet efficient cross-spatial and cross-channel attention mechanism that adaptively learns the importance of each channel and spatial position through coordinated channel and spatial attention modules, thereby enhancing feature map expressiveness and discriminative power. The structure is shown in Figure 6. For an input feature map $F \in \mathbb{R}^{c \times h \times w}$, channel attention map $M_c \in \mathbb{R}^{c \times 1 \times 1}$, and spatial attention map $M_s \in \mathbb{R}^{1 \times h \times w}$, the processing flow can be summarized as:

$$\begin{aligned} F' &= M_c(F) \otimes F, \\ F'' &= M_s(F') \otimes F', \end{aligned} \tag{5}$$

where $\otimes$ denotes element-wise multiplication, and F'' is the final output feature map.

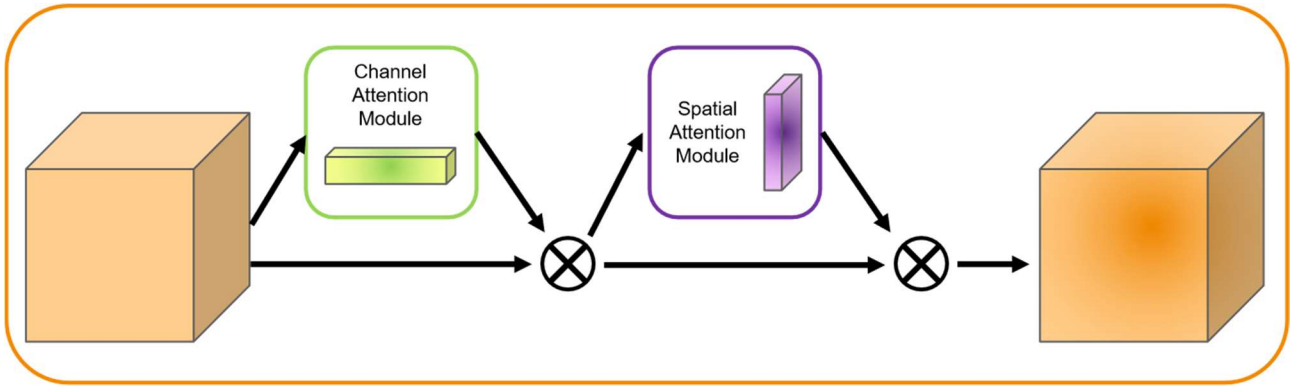

Figure 6. CBAM structure diagram.

The channel attention module assigns different weights to each channel, as different channels represent distinct feature detectors. We aim to enhance task-relevant channels (e.g., those

ISSN: 2664-9640

capturing target textures) while suppressing redundant ones (e.g., background noise). Beyond traditional global average pooling, the module incorporates global max pooling to aggregate spatial features for channel weight evaluation. Both results are fed into a shared multilayer perceptron (MLP) with one hidden layer, then combined through element-wise addition to produce the channel attention (Figure 7):

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ = \sigma(W1(W0(F^c_{avg})) + W1(W0(F^c_{max}))), \quad (6)$$

where σ is the Sigmoid activation function, $W0 \epsilon \mathbb{R}^{c/r \times c}$ and $1 \epsilon \mathbb{R}^{c \times c/r}$ are shared MLP weights, and ReLU activation follows $W0$.

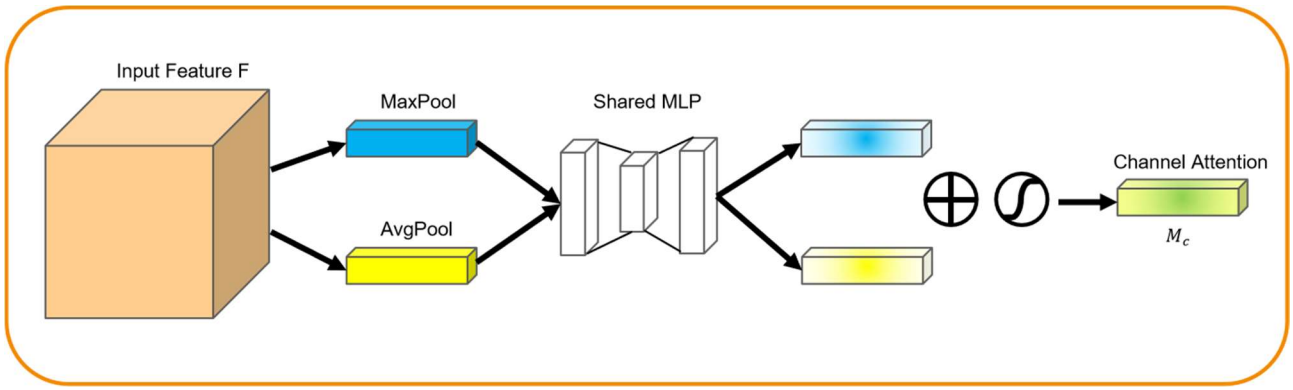

Figure 7. Channel attention module structure.

The spatial attention module complements channel attention by identifying the most informative spatial locations. It applies average and max pooling along the channel dimension to generate two 1×h×w aggregated maps, which are concatenated and processed by a 7×7 convolution followed by Sigmoid activation (Figure 8):

$$M_s(F) = \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \\ = \sigma(f^{7\times7}([F^s_{avg}; F^s_{max}])), \quad (7)$$

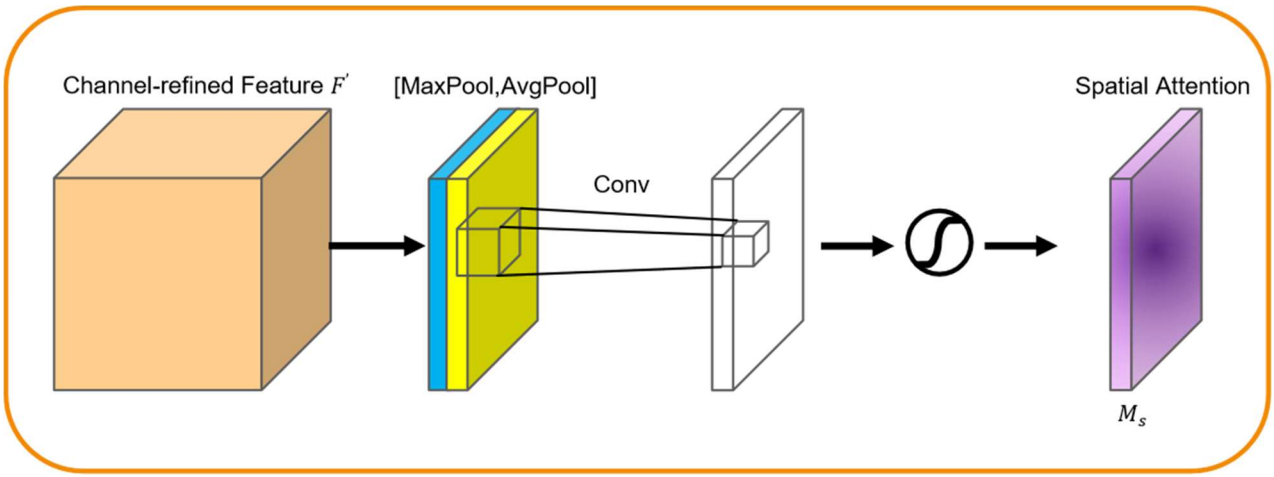

Figure 8. Spatial attention module structure.

ISSN: 2664-9640

In our implementation, complete CBAM layers are inserted at the end of Stage 3 and Stage 5 in the backbone network (corresponding to 80×80 and 20×20 resolutions) to respectively enhance feature extraction for small and large targets, mitigating accuracy loss from scale variations and occlusions.

2.3. Model Training

2.3.1. Experimental Setup

The hardware platform for this experiment utilizes an NVIDIA GeForce RTX 3080 GPU with 10GB VRAM and 24GB system memory. The experiments were conducted on Ubuntu 20.04 operating system with the following software versions: Python 3.11, PyTorch 2.3.0+cu118, torchvision 0.18.0+cu118, CUDA 11.8, and Ultralytics 8.2.82. The model was trained for 200 epochs with a batch size of 16, while other parameters remained at their default settings.

2.3.2. Dataset

This study employs the NWPU VHR-10 dataset, a high-resolution remote sensing object detection dataset published by Northwestern Polytechnical University. The dataset consists of 800 high-resolution remote sensing images, including 650 positive samples (with annotated targets) and 150 background images (without targets). It contains ten target categories: Airplane, Ship, Storage Tank, Baseball Diamond, Tennis Court, Basketball Court, Ground Track Field, Harbor, Bridge, and Vehicle. Sample images are shown in Figure 9. The positive samples were randomly divided into training and validation sets at an 8:2 ratio, while all negative samples were included in the training set.

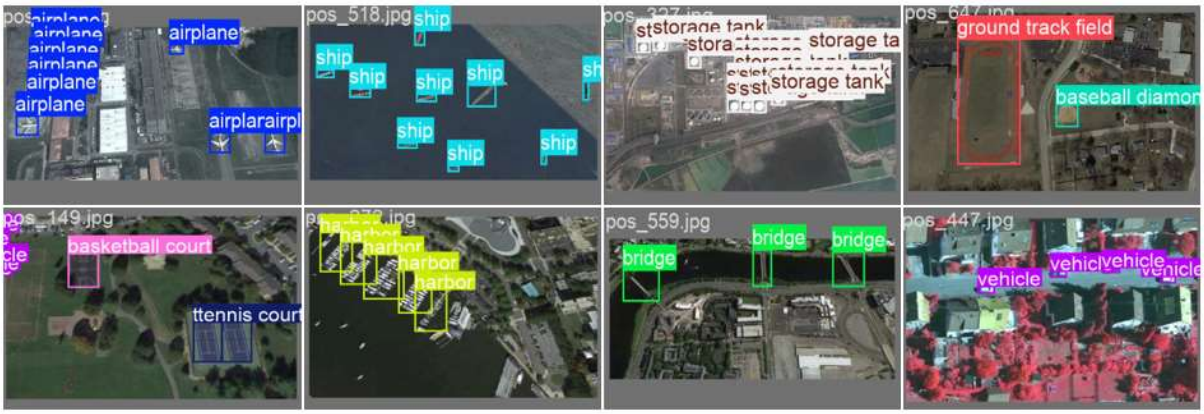

Figure 9. Example images from NWPU VHR-10 dataset.

2.3.3. Evaluation Metrics

The experiment adopts precision (P), recall (R), and mean average precision (mAP) to evaluate model accuracy, while using giga floating-point operations (GFLOPs) to assess model lightweightness. The formulas for precision, recall, and mAP are as follows:

$$P = \frac{TP}{TP + FP}, \tag{8}$$

$$R = \frac{TP}{TP + FN}, \tag{9}$$

ISSN: 2664-9640

$$mAP = \frac{1}{k}\sum_{i=1}^{k} AP_i, \tag{10}$$

where precision measures the proportion of correctly predicted positive samples among all predicted positive samples; recall measures the proportion of correctly identified positive samples among all actual positive samples; and mAP represents the average of precision values across all categories. TP denotes true positives (correctly predicted positive samples), FP denotes false positives (negative samples incorrectly predicted as positive), and FN denotes false negatives (positive samples incorrectly predicted as negative). AP_i represents the area under the precision-recall curve. For mAP evaluation, we specifically use mAP@50 and mAP@50-95: the former calculates mAP at an IoU (Intersection over Union) threshold of 0.5, while the latter averages mAP values across 10 IoU thresholds from 0.5 to 0.95 (with a step size of 0.05).

3. Results and Analysis

3.1. Comparison of Different Lightweight Networks

To demonstrate the superior performance and rationale of introducing the Ghost module from GhostNet, we compared it with other lightweight networks, including ShuffleNetV2 [25], MobileNeXt [26], and the original GhostNet [23]. These networks were used to completely replace the backbone of YOLOv8, and all models were trained and tested on the same NWPU VHR-10 dataset. The comparative results are shown in Table 1:

Table 1. Performance comparison of different lightweight networks.

Model	GFLOPs	P	R	mAP@50	mAP@50-95
YOLOv8s	28.8	0.939	0.886	0.933	0.638
YOLOv8s-GhostModule	21.3	0.934	0.853	0.913	0.591
YOLOv8s-ShuffleNet	16.2	0.866	0.82	0.863	0.536
YOLOv8s-MobileNeXt	18.9	0.894	0.827	0.883	0.547
YOLOv8s-GhostNet	18.7	0.914	0.811	0.9	0.571

As shown in Table 1, all lightweight networks effectively reduced computational complexity, but detection accuracy decreased to varying degrees. Among them, YOLOv8s-GhostModule (our proposed method) achieved the best overall performance. By retaining part of YOLOv8's backbone structure and applying the Ghost module's cheap operations, it reduced GFLOPs by 7.5 while only decreasing precision (P) by 0.005 and mAP@50 by 0.02, with other metrics also performing well. In contrast, the other three models reduced GFLOPs more significantly but suffered from excessive accuracy degradation due to complete backbone modifications. For instance, mAP@50 decreased by 0.07, 0.05, and 0.33 for ShuffleNetV2, MobileNeXt, and GhostNet, respectively, failing to meet the accuracy requirements for remote sensing object detection. Specifically: (1) YOLOv8s-ShuffleNet: Channel shuffling may disrupt spatial continuity in feature maps, weakening sparse texture features of small objects and leading to feature loss. (2) YOLOv8s-MobileNeXt: Its inverted residual structure repeatedly compresses channel dimensions, increasing noise and potentially causing gradient decay in occlusion scenarios (e.g., cloud cover). (3) YOLOv8s-GhostNet: Excessive use of Ghost modules reduced effective feature channels, resulting in insufficient feature representation and lower accuracy.

In summary, YOLOv8s-GhostModule is the most effective lightweight modification, achieving an optimal balance between computational efficiency and accuracy.

ISSN: 2664-9640

3.2. Ablation Study

Building upon the lightweight improvements, we further enhanced the model with the CBAM attention mechanism. To evaluate the impact of the Ghost module and CBAM, we conducted ablation experiments on the NWPU VHR-10 dataset under identical parameter settings. The results are presented in Table 2:

Table 2. Ablation study results.

Model	GFLOPs	P	R	mAP50	mAP50-95
YOLOv8s	28.8	0.939	0.886	0.933	0.638
YOLOv8s-GhostModule	21.3	0.934	0.853	0.913	0.591
YOLO-GC	21.4	0.935	0.852	0.922	0.596

As shown in Table 2, after implementing the lightweight improvement with the Ghost module, the model's GFLOPs decreased by 7.5, while precision (P), recall (R), mAP@50, and mAP@50-95 decreased by 0.005, 0.033, 0.02, and 0.47, respectively. This indicates that the Ghost module significantly reduces computational complexity while causing only a minor accuracy drop. The slight performance degradation occurs because the module replaces some effective feature maps with redundant ghost features, thereby reducing the model's representational capacity. However, given the substantial reduction in computation, this accuracy trade-off is acceptable.

When the CBAM attention mechanism is further added to YOLOv8s-GhostModule, the model's GFLOPs increase by only 0.1, recall decreases by 0.001, while precision, mAP@50, and mAP@50-95 improve by 0.001, 0.009, and 0.005, respectively. Overall, CBAM successfully enhances accuracy with negligible computational overhead. This improvement is attributed to the CBAM layers integrated into Stage 3 and Stage 5 of the backbone, which strengthen the model's ability to extract multi-scale target features. For example, ship and bridge share similar shapes and backgrounds but differ in scale. With CBAM, the model's discrimination capability improves significantly-mAP@50 for ship increases from 0.858 to 0.891, and for bridge, from 0.816 to 0.915.

In summary, the Ghost and CBAM modules complement each other: the former reduces computation, while the latter recovers lost accuracy. Compared to the baseline, YOLO-GC achieves a 25.7% reduction in computational cost with only a 1.1% drop in mAP@50, effectively meeting the improvement objectives.

3.3. Detection Performance Comparison

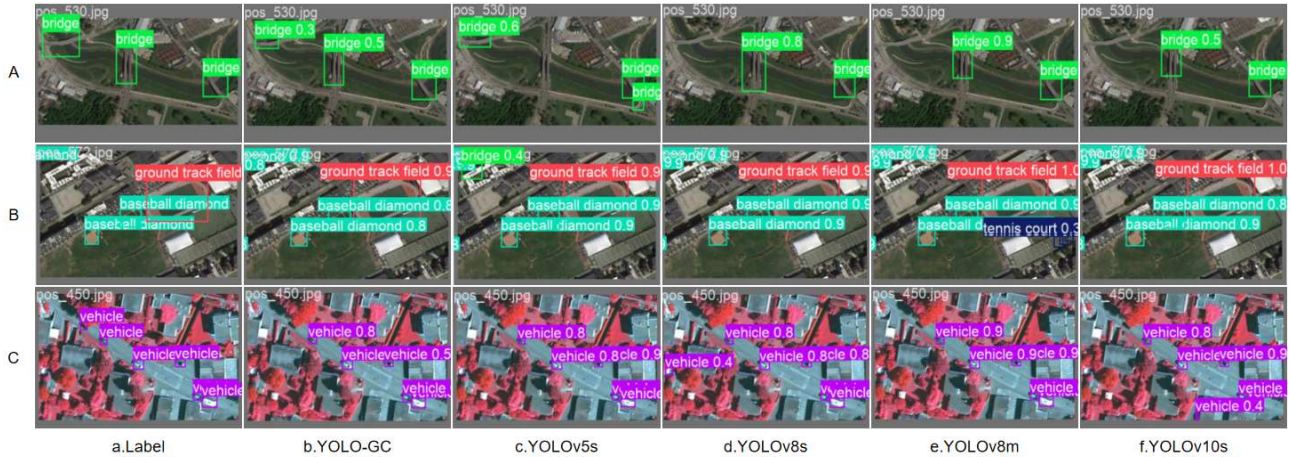

Figure 10. Visual comparison of detection results across different models.

ISSN: 2664-9640

To validate the effectiveness of YOLO-GC compared to other mainstream models in remote sensing object detection, we conducted comparative experiments on the NWPU VHR-10 dataset using identical parameters with the following models: YOLOv5s, YOLOv8s, YOLOv10s, and YOLOv8m. Partial detection results are presented in Figure 10.

Analysis of Figure 10 reveals that for Image A (dominated by bridges), the YOLO-GC model demonstrates superior detection performance, correctly identifying all targets, while other models exhibit varying degrees of missed detections and false positives. Specifically, YOLOv5s failed to detect the central target while misclassifying a road segment as a bridge, and YOLOv8s, YOLOv8m, and YOLOv10s all missed the target in the upper-left region of the image. In Image B, YOLO-GC again achieved perfect detection, whereas YOLOv5s produced one false positive by misidentifying a building as a bridge, and YOLOv10s generated three false positives by classifying buildings as tennis courts. For Image C, while YOLO-GC had one false positive (a particularly challenging case where the target was obscured beneath trees and could easily be mistaken for shadows), it's noteworthy that none of the other models successfully detected this target. Additionally, both YOLOv8s and YOLOv10s produced false positives by misclassifying buildings or shadows as vehicles.

These instances of missed detections and false positives underscore the inherent challenges in feature extraction from aerial-view remote sensing images, particularly for small targets that are highly susceptible to interference from complex backgrounds, occlusions, and shadows. Remarkably, the YOLO-GC model not only achieves significant computational efficiency gains but also maintains comparable or superior detection performance to other models. These results convincingly demonstrate both the effectiveness of the attention mechanism and the success of the model's lightweight design.

4. Conclusion

To address the challenges of high computational complexity and difficulties caused by occlusions, shadows, and scale variations in remote sensing object detection, this study proposes YOLO-GC, a lightweight object detection model based on improved YOLOv8. First, to reduce computational load, we introduced the Ghost module and its ghost feature concept from GhostNet, replacing Conv with GhostConv and C2f with C2f_GhostConv in the backbone network. These modifications significantly decreased computational requirements through cheap operations while maintaining satisfactory accuracy. Subsequently, to mitigate issues like scale variation and further improve accuracy, we incorporated the CBAM attention mechanism by adding CBAM layers at the end of Stage 3 and Stage 5 in the backbone network. The channel and spatial attention mechanisms in CBAM effectively enhanced feature extraction, boosting model accuracy with negligible computational overhead.

Experimental results on the NWPU VHR-10 dataset demonstrate that compared to the baseline YOLOv8s, YOLO-GC reduces GFLOPs by 7.4 while only decreasing precision (P), recall (R), mAP50, and mAP50-95 by 0.004, 0.034, 0.011, and 0.042, respectively. These results validate that the improved model substantially reduces computational demands while maintaining detection accuracy, making it suitable for deployment on resource-constrained edge platforms.

However, the improved model still faces certain limitations: although computational requirements are reduced, its accuracy does not comprehensively surpass comparable models. For some targets, even the attention mechanism cannot fully compensate for the loss of effective features caused by ghost features, indicating room for further accuracy improvement. Since this study only employed CBAM as the attention mechanism, future research could explore combining multiple attention mechanisms to potentially achieve better performance.

ISSN: 2664-9640

References

[1] Tang W J. Research Status of Remote Sensing Image Classification Based on Deep Learning[J]. Computer Science and Application, 2024, 14(8): 101-109. DOI: 10.12677/csa.2024.148179

[2] Liu X B, Liu P, Cai Z H, Qiao Y L, Wang L, Wang M. Advances in Optical Remote Sensing Image Object Detection Based on Deep Learning[J]. Acta Automatica Sinica, 2021, 47(9): 2078-2089. DOI: 10.16383/j.aas.c190455

[3] Cao X M. Vehicle Detection Based on Multi-Image Feature Pyramid[D]. Beijing Jiaotong University, 2016. DOI:10.7666/d.Y3124193.

[4] Zhang G M, Zhang S, Chu J. A Novel Target Detection Method Based on Local Contour Features[J]. Acta Automatica Sinica, 2014, 40(10): 2346-2355. DOI:10.3724/SP.J.1004.2014.02346.

[5] Lu, D,Weng, Q,SanchezHernandez, Carolina,et. Support vector machines in remote sensing: A review[J]. 2011.

[6] Mariana,Belgiu,Lucian,et al.Random forest in remote sensing: A review of applications and future directions[J].Isprs Journal of Photogrammetry & Remote Sensing, 2016.DOI:10.1016/j.isprsjprs.2016.01.011.

[7] Felzenszwalb P F , Mcallester D A , Ramanan D .A discriminatively trained, multiscale, deformable part model[C]//2008 IEEE Conference on Computer Vision and Pattern Recognition.IEEE, 2008.DOI:10.1109/CVPR.2008.4587597.

[8] Girshick R , Donahue J , Darrell T ,et al.Region-Based Convolutional Networks for Accurate Object Detection and Segmentation[J].IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 38(1):142-158.DOI:10.1109/TPAMI.2015.2437384.

[9] Girshick R .Fast R-CNN[J].Computer Science, 2015.DOI:10.1109/ICCV.2015.169.

[10] Ren S , He K , Girshick R ,et al.Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J].IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.DOI:10.1109/TPAMI.2016.2577031.

[11] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788

[12] Wei L , Dragomir A , Dumitru E ,et al.SSD: Single Shot MultiBox Detector[J].Springer, Cham, 2016.DOI:10.1007/978-3-319-46448-0_2.

[13] Duan K , Bai S , Xie L ,et al.CenterNet: Keypoint Triplets for Object Detection[J].University of Chinese Academy of Sciences; University of Oxford; Huawei Noah's Ark Lab; Huawei Noah’s Ark Lab[2025-07-09].DOI:10.1109/ICCV.2019.00667.

[14] Ju M R, Luo H B, Wang Z B, et al. Improved YOLO V3 Algorithm and Its Application in Small Target Detection[J]. Acta Optica Sinica, 2019, 39(7): 0715004. DOI:10.3788/AOS201939.0715004.

[15] Wang X K, Jiang H X, Lin K Y. Ship Detection in Remote Sensing Images Based on Improved YOLO Algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2020, 46(6): 8. DOI:10.13700/j.bh.1001-5965.2019.0394.

[16] Tang J Y, Tang C H. Remote Sensing Image Object Detection Algorithm Based on Rotated Bounding Box and Attention Mechanism[J]. Electronic Measurement Technology, 2021, 44(13): 7. DOI:10.19651/j.cnki.emt.2106740.

[17] Zhang S W, Shi W Y, Zhang S Q, et al. Small Target Detection in Remote Sensing Images Based on Weighted Receptive Field and Cross-Layer Fusion[J]. Electronic Measurement Technology, 2023, 46(18): 129-138.

[18] Yan J H, Zhang K, Shi T J, et al. Weak and Small Ground Target Detection in Remote Sensing Images by Fusing Multi-Level Features[J]. Chinese Journal of Scientific Instrument, 2022(3): 221-229.

[19] Karim S , Zhang Y , Yin S ,et al.A brief review and challenges of object detection in optical remote sensing imagery[J].Multiagent and Grid Systems, 2020, 16(3):227-243.DOI:10.3233/MGS-200330.

[20] Han,Junwei,Cheng,et al.A survey on object detection in optical remote sensing images[J].ISPRS journal of photogrammetry and remote sensing, 2016, 117(Jul.):11-28.

ISSN: 2664-9640

[21] Ma C W, Zhang H, Ma X M, et al. Lightweight Wheat Disease Detection Method Based on Improved YOLOv8[J]. Transactions of the Chinese Society of Agricultural Engineering, 2024, 40(5). DOI:10.11975/j.issn.1002-6819.202309211.

[22] Varghese R , Sambath M .YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness[C]//2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS).0[2025-07-09].DOI:10.1109/ADICS58448.2024.10533619.

[23] Han K , Wang Y , Tian Q ,et al.GhostNet: More Features From Cheap Operations[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).IEEE, 2020.DOI:10.1109/CVPR42600.2020.00165.

[24] Woo S , Park J , Lee J Y ,et al.CBAM: Convolutional Block Attention Module[J].Springer, Cham, 2018.DOI:10.1007/978-3-030-01234-2_1.

[25] Ma N , Zhang X , Zheng H T ,et al.ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design[J].Springer, Cham, 2018.DOI:10.1007/978-3-030-01264-9_8.

[26] Daquan Z , Hou Q , Chen Y ,et al.Rethinking Bottleneck Structure for Efficient Mobile Network Design[J]. 2020.DOI:10.1007/978-3-030-58580-8_40.