






CGBi_YOLO: Lightweight Land Target Detection Network

Ruiyang Wang¹, Siyu Lu¹, Jiawei Tian¹ , Lirong Yin^{2,*} , Lei Wang² , Xiaobing Chen³ 
and Wenfeng Zheng^{1,*} 

¹ School of Automation, University of Electronic Science and Technology of China, Chengdu 610054, China; ruiyang.wang@std.uestc.edu.cn (R.W.); siyu.lu@std.uestc.edu.cn (S.L.); tianjiawei@hanyang.ac.kr (J.T.)

² Department of Geography and Anthropology, Louisiana State University, Baton Rouge, LA 70803, USA; leiwang@lsu.edu

³ School of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803, USA; xchen87@lsu.edu

* Correspondence: lyin5@lsu.edu (L.Y.); winfirms@ieee.org (W.Z.)

Abstract: Object detection algorithms for optical remote sensing images often face challenges in computational efficiency, particularly when detecting small and densely packed targets. This paper introduces CGBi_YOLO, a novel lightweight land target detection network designed to optimize computational resource utilization while maintaining detection capabilities for small-scale targets. Our approach incorporates an innovative lightweight optimization strategy featuring a new lightweight backbone feature extraction network: CSPGhostNet. This model significantly enhances the detection ability of small objects within optical remote sensing images without increasing computational demands. The efficacy of the proposed model is validated through rigorous experimentation on the DOTA dataset. Compared to the baseline model, CGBi_YOLO achieves a 30% reduction in parameters and a 36% increase in inference speed. The model demonstrates exceptional performance in handling small and densely packed targets within optical remote sensing images, showcasing its potential for real-world applications in fields such as environmental monitoring, urban planning, and disaster management.

Keywords: remote sensing image; object detection; land target detection; deep learning; CGBi_YOLO; lightweight transformation



Citation: Wang, R.; Lu, S.; Tian, J.; Yin, L.; Wang, L.; Chen, X.; Zheng, W. CGBi_YOLO: Lightweight Land Target Detection Network. *Land* **2024**, *13*, 2060. <https://doi.org/10.3390/land13122060>

Academic Editor: Emiliano Carmona

Received: 8 October 2024

Revised: 26 November 2024

Accepted: 27 November 2024

Published: 30 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the realm of object detection, the analysis of optical remote sensing images represents a fundamental research task. These images, capturing the optical wavelength band within the visible light spectrum, offer intuitive, high-resolution data rich in feature information [1]. The wealth of information contained in this data holds significant potential across various societal domains. The evolution of optical remote sensing image target detection can be broadly categorized into two distinct eras: the pre-2014 era dominated by traditional methods and the post-2014 era characterized by deep learning-based approaches [2].

Initially, optical remote sensing image target detection relied heavily on manual annotation. This approach, while precise, was time-consuming and labor-intensive and struggled to meet real-time demands. Its highly targeted nature also limited its generalization capabilities [3]. The advent of advanced computing equipment has ushered in a new era where deep neural network-based target detection methods have gained prominence in related fields [4–6]. These methods enhance object detection in natural scenes through targeted neural networks that learn feature information and perform reasoning. However, the transition of these neural networks to optical remote sensing images is not without challenges, given the inherent differences between optical remote sensing images and natural scenes. Deep learning-based object detection methodologies can be broadly classified into two categories [7]: two-stage detection processes, which conceptualize the generation

of detection frames as a range adjustment process, and single-stage detection processes, which generate detection frames in a single pass.

The R-CNN network, proposed by Girshick et al. marked the first concrete implementation of the two-stage detection method and pioneered the use of convolutional neural networks for object detection [8]. This method achieves object detection through a four-step process: calculation of candidate regions, feature information extraction via convolutional neural networks, target classification, and bounding box regression. While this approach significantly improved detection performance compared to traditional algorithms, it also substantially increased computational costs.

Subsequent developments saw the introduction of SPP-Net by He et al. in 2014 [9], which addressed the fixed resolution input image requirement of R-CNN networks. However, SPP-Net still faced challenges with computational intensity. In 2015, Girshick et al. proposed Fast-RCNN [10], an enhancement to R-CNN based on the ROI pooling method, which improved inference speed. The same year saw the introduction of Faster-RCNN by Shaoqing Ren et al. [11], the first end-to-end network in deep learning history, with detection speeds approaching real-time requirements.

Concurrently with Faster-RCNN's debut, R. Joseph et al. developed YOLO [12], the first single-stage target detection algorithm. The initial version of the YOLO series achieved a detection speed of 155 FPS (frame per second), fully satisfying real-time demands. In 2016, W. Liu et al. proposed the Single Shot MultiBox Detector (SSD) [13], marking the beginning of the deep convolution-based single-stage detection network era. The key distinction between SSD and YOLO lies in their approach to multi-scale detection: YOLO performs detection at different feature levels of the network, while SSD focuses on the highest-level features.

The YOLO series saw further advancements with the open-sourcing of YOLOV3 by R. Joseph et al. in 2018 [14]. This iteration introduced feature fusion structures to the YOLO series, drawing inspiration from Feature Pyramid Networks (FPN) [15]. Subsequently, Bochkovskiy A. et al. updated the YOLO algorithm to its fourth version, YOLOV4 [16], exploring various optimization methods. The YOLO network architecture is now typically divided into four functional components: network input structure, backbone feature extraction network structure, neck structure, and detection head structure.

As deep learning-based target detection techniques advance, the field of optical remote sensing image analysis is increasingly adopting single-stage detection algorithms. A prevalent approach involves utilizing automated clustering algorithms for dataset analysis, followed by the development of adaptive distance calculation formulas to derive more meaningful intersection ratios. This methodology draws inspiration from the YOLO series architecture, aiming to enhance adaptability. Concepts from Densely Connected Convolutional Networks (DenseNet) [17] are employed, integrating dense connection layers with residual block structures. This combination strengthens the network's capacity to extract informative features. Additionally, the Neck network, responsible for feature fusion, extensively utilizes various feature pyramid structures. These optimization techniques demonstrate superior performance compared to traditional methods.

In 2018, Xu Y et al. focused on improving the feature fusion structure [18]. Ghorbani F et al. introduced a processing method for differentiated samples and their background changes using the PIIFD characterization operator [19], demonstrating enhanced performance in optical remote sensing image target detection tasks compared to traditional methods. In 2020, Cao C et al. proposed a deep learning-based ship detection method [20], essentially a series of YOLO-based algorithms.

However, challenges persist in disseminating low-level semantic information when dealing with small-sized target objects. In optical remote sensing images, substantial variations in appearance and shape complicate the predefinition of anchor frames, potentially leading to missed targets. Current approaches to enhancing potential target detection often involve increasing the number of anchor frames [21–23], but this strategy comes with increased computational costs.

To address these challenges and further enhance computational efficiency in the feature extraction process, we build upon established models and introduce lightweight optimization strategies. We propose a novel lightweight backbone feature extraction network, CSPGhostNet, which we integrate into an existing architecture to form a new target detection model: CGBi_YOLO.

This model significantly improves the detection capability of small objects in optical remote sensing images through careful optimization and an innovative combination of CSPGhostNet components. The basic unit modules within CSPGhostNet are designed for reuse through simple linear transformations and truncated gradients, generating high-quality feature maps with low computational costs. Our model strikes an optimal balance between accuracy and computational efficiency while controlling computational overhead.

The proposed CGBi_YOLO model is validated through rigorous experiments on an augmented dataset, demonstrating improved detection capabilities for small-scale targets in optical remote sensing images without significantly increasing computational complexity.

2. Methods

To enhance the detection of small targets in optical remote sensing images, we chose YOLOv4 as the base model for this study. Although newer versions of YOLO, such as YOLOv5 and YOLOv8, are widely used, YOLOv4 still offers significant advantages in small target detection.

YOLOv4 strikes a good balance between accuracy and computational efficiency, making it particularly suitable for resource-constrained applications that require real-time processing. Small and densely packed targets in optical remote sensing images present unique challenges, and YOLOv4's architecture is well suited to address these issues effectively.

2.1. YOLOV4_CSPBi

YOLOV4_CSPBi [24] is a model specifically engineered to enhance the detection of small land targets in remote sensing images. It incorporates a weight-based, bidirectional, and multi-scale mechanism for effective feature fusion, enabling efficient reasoning about objects of various sizes, with a particular focus on small land objects. This model refines the channel division approach compared to the conventional cross-stage part network (CSPNet) [25], integrating this improved structure into the neck segment of the YOLO network to bolster its learning capabilities and augment small land object recognition in remote sensing images.

The YOLOV4_CSPBi architecture removes the pyramid fusion structure typically used for large target detection in traditional BiFPN, reallocating the computation for large object detection to the efficient feature fusion part for small object detection. Figure 1 illustrates the network architecture.

2.2. Improved Ghost Feature Extraction Unit: CSPGhost

Sometimes, to ensure that the extracted features accurately capture the key characteristics of the original sample, the extracted information may contain a significant amount of redundancy. This results in higher computational complexity and reduced model performance.

Figure 2 shows an example of partial feature map visualization after the first feature extraction of the YOLOV4_CSPBi network. Among these images, some of them are similar, which exist such as each other's "ghosts" and are redundant with each other, as described by Kai Han in [26]. However, the remarkable achievements of deep neural networks may be achieved precisely because of the existence of abundant redundant feature maps. Therefore, in the construction of deep neural networks, it should be inclined to accept this redundancy rather than try to avoid its existence. Meanwhile, this tendency to obtain multi-redundant feature information should be achieved at a lower computational cost rather than by unlimitedly expanding the structural width or depth of deep neural networks.

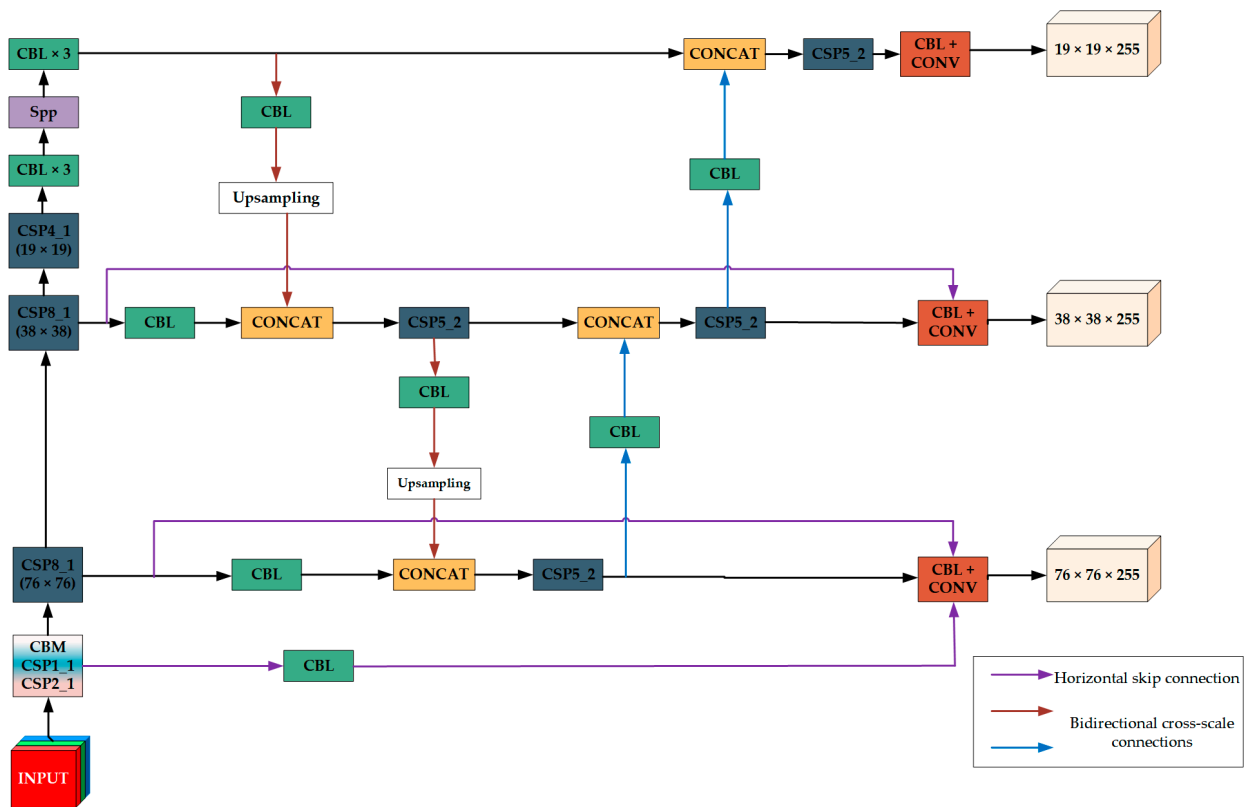


Figure 1. YOLOV4_CSPBi network structure [24].

This is also the research significance of the GhostNet network [26]. Inspired by GhostNet, we will further reduce the computation complexity of the target detection network in this paper.

To reduce the computational cost as much as possible and speed up the reasoning process of the network, we propose an improved feature extraction unit, CSPGhost, which is by GhostNet.

GhostNet reduces the complexity of the deep convolutional structure through a simple linear transformation structure, as depicted in Figure 3. For the input sample information $X \in \mathbb{R}^{c \times h \times w}$ required by the network (where c represents the input sample channel number, h and w stand for the height and width of the input sample), any convolution calculation can be formally represented by Equation (1):

$$Y = X \otimes f + b \tag{1}$$

where \otimes means the convolution operation, b means the bias term of this layer of the network, and $f \in \mathbb{R}^{c \times k \times k \times n}$ means the convolution kernel of this layer of the network. The number of FLOPs required in the convolution calculation process can be obtained by the following Equation (2).

$$FLOPs_{CONV} = n \times h' \times w' \times c \times k \times k \tag{2}$$

where c is the channel number of the convolution kernel, and this value should be kept with the input sample channel number. k and n are the size and number of convolution kernels. The output result of the convolution of this layer $Y \in \mathbb{R}^{h' \times w' \times n}$ can be calculated, where h' , w' and n represent the height and width of the output feature map and channel number, respectively.

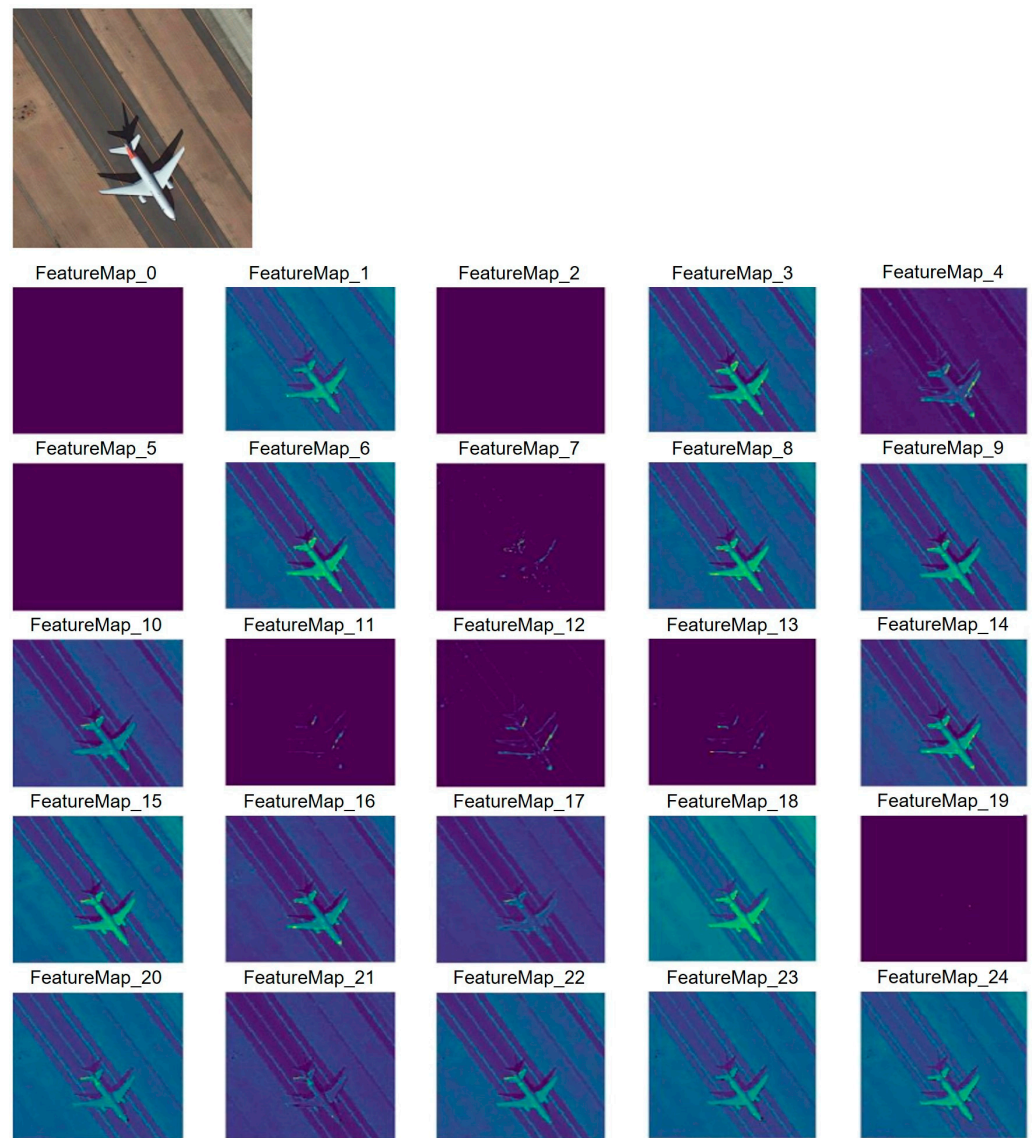


Figure 2. Visualization of partial feature maps after the first feature extraction of the YOLOV4Bi network.

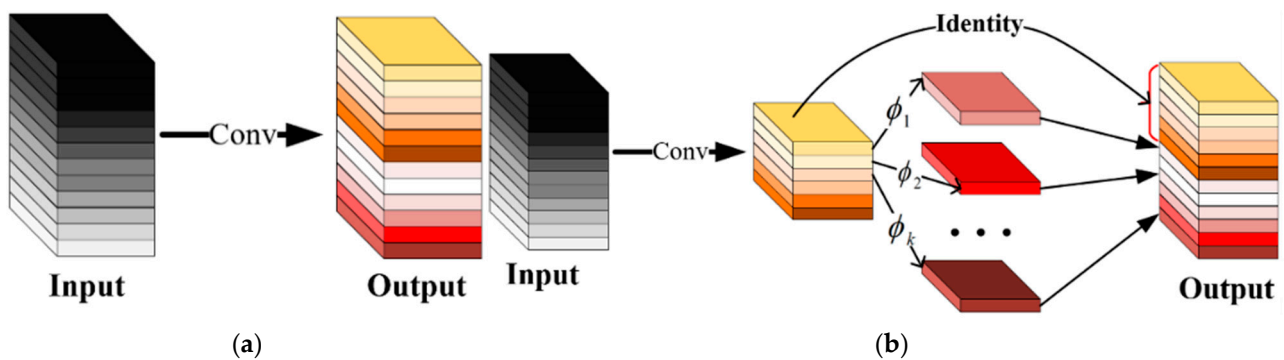


Figure 3. Structure comparison of traditional convolution (a) and ghost convolution (b).

From Equation (2), many convolution stacks in deep convolutional neural networks will generate huge FLOPs. Therefore, to optimize the number of parameters contained in f and b is necessary to minimize the use of complex convolutions. As shown in Figure 3, the output feature map after convolution calculation contains many redundant feature maps, so it is actually unnecessary to obtain all feature maps one by one through such a

computationally expensive convolution operation. The mutually redundant “Ghost maps” can be obtained using some linear changes with less computational complexity. Some basic feature maps can be obtained through a small amount of original convolution calculations. Then, by using these basic feature maps, a complete redundant feature map can be obtained through a linear transformation with low computational cost.

Specifically, m original feature maps $Y' \in \mathbb{R}^{h' \times w' \times m}$ can be obtained by the first convolution, where Y' is calculated by the following Equation (3).

$$Y' = X \otimes f' \quad (3)$$

where $f' \in \mathbb{R}^{c \times k \times k \times m}$ is the convolution kernel. For the convenience of explaining the principle of the method, the bias term b is omitted here, and the hyperparameters of other convolution operations are the same as those in ordinary convolution. To further obtain the required n feature maps, it is only necessary to use a low-computationally-cost linear transformation operation on each original feature map in Y' to generate s redundant feature maps. The specific operation is completed by the Equation (4).

$$y_{ij} = \Phi_{y,j}(y'_i), \forall i = 1, \dots, m, j = 1, \dots, s, \quad (4)$$

where y'_i means the number of sub-images in the m original feature maps, and $\Phi_{y,j}$ is a function of y'_i generating redundant feature map y_{ij} .

The redundant feature map y_{ij} and y'_i has a many-to-one relationship, that is, the same original feature map can generate multiple redundant feature maps, and the last one $\Phi_{y,j}$ is used to add an identity map connection to preserve the original map. The information of the map, the identity map in Figure 3b, through this low-computational operation, n feature maps containing redundancy can be obtained as the output of the ghost module.

CSPGhost proposed here performs channel shunting on the convolutional part of the traditional ghost network structure, which ensures that the ghost network does not contain repeated gradient information when updating the gradient information of the weight, and this channel shunting method based on the CSP idea does need to update parameters while optimizing the overall efficiency of the network in the gradient information transfer. The improved CSPGhost structure adopts depthwise separable convolution in generating redundant feature maps. Its formalized expression is shown in the following Equation (5).

$$y_{ij} = \Phi_{i,j}^{DC}(y'_i) \oplus \Phi_{i,j}^{DC}(y''_i), \forall i = 1, \dots, m, j = 1, \dots, s, \quad (5)$$

where \oplus represents the depthwise separable convolution operation on the feature information divided by the CSP structure and y_{ij} represents the different feature information after the CSP structure. After completing the depthwise separable convolution operation to generate the redundant feature map, the CSPGhost structure performs identity mapping on the original condensed feature map obtained after the CSP convolution calculation and directly superimposes the redundant feature map as the output feature of a CSPGhost basic unit. Figure 4 is a schematic structural diagram of a CSPGhost basic unit and CSPGhost unit.

The calculation process of CSPGhost redundant feature map generation, as shown in Equation (5), obviously reduces the number of parameters of traditional convolutional networks and reduces the reuse rate during gradient propagation. Specifically, the basic unit of CSPGhost contains a feature-concentrated constant equal mapping and $m \times (s - 1) = n/s \times (s - 1)$ linear operations. Assuming that the kernel size of each linear depth separable structure is all $d \times d$, the achievable theoretical speed improvement ratio can be calculated by the following Equation (6).

$$\begin{aligned}
 r_s &= \frac{n \times h' \times w' \times c \times k^2}{\frac{n}{s} \times h' \times w' \times c \times k^2 + (s-1) \times \frac{n}{s} \times h' \times w' \times d^2} \\
 &= \frac{c \times k^2}{\frac{1}{s} \times c \times k^2 + \frac{s-1}{s} \times d^2} \\
 &\approx \frac{s \times c}{s+c-1} \\
 &\approx s
 \end{aligned}
 \tag{6}$$

where the parameters $d \times d$ and $s \times s$ represent the kernel size of traditional convolution operations and linear depthwise separable operations, basically comparable in magnitude, but $s \ll c$. Therefore, the parameters of the improved CSPGhost basic unit can achieve the compression ratio shown in the following Equation (7) compared with the traditional convolution operation.

$$\begin{aligned}
 r_c &= \frac{n \times c \times k^2}{\frac{n}{s} \times c \times k^2 + (s-1) \times \frac{n}{s} \times d^2} \\
 &\approx \frac{s \times c}{s+c-1} \\
 &\approx s
 \end{aligned}
 \tag{7}$$

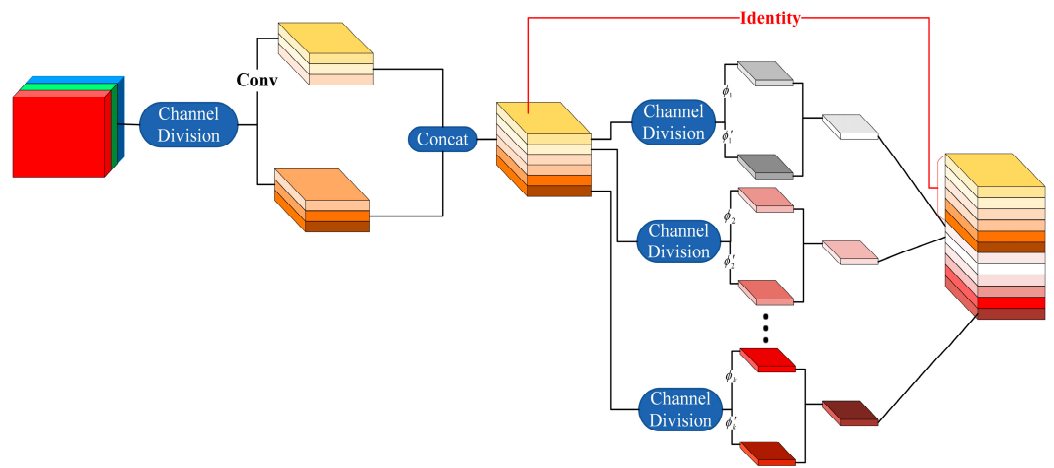


Figure 4. Improved feature extraction unit CSPGhost unit structure.

2.3. Improved Lightweight Backbone Network: CSPGhostNet

With the CSPGhost unit component described above, we introduce two modules for the backbone feature extraction structure: CSPGhost_1 and CSPGhost_2, shown in Figure 5.

The difference between these two modules is that CSPGhost_1 first expands the number of channels through the first CSPGhost unit, extracts feature information, and subsequently reduces the channel number through the second CSPGhost unit to achieve the purpose of condensing the feature map. It can be seen that CSPGhost_1 only has the ability to scale channels, while CSPGhost_2 has the ability to simultaneously scale channels, width, and height information by adding a new layer of depthwise separable convolutional networks.

In addition, it can also be seen that the overall structure of the CSPGhost module is similar to the structure of the residual network, and the addition of features is achieved through a residual short-circuit edge. The problem of gradient loss exists, but CSPGhost has made a certain improvement on this basis and modified the calculation process of features in the traditional residual network from “compression, extraction, expansion” to “expansion, extraction, compression” to prevent the difficulty of extracting effective information after feature compression, and it also ensures that the modification of the dimension does not affect the operation result of the activation function. This idea of inverse residual processing can also be found in related lightweight networks, such as MobileNetV2 [27].

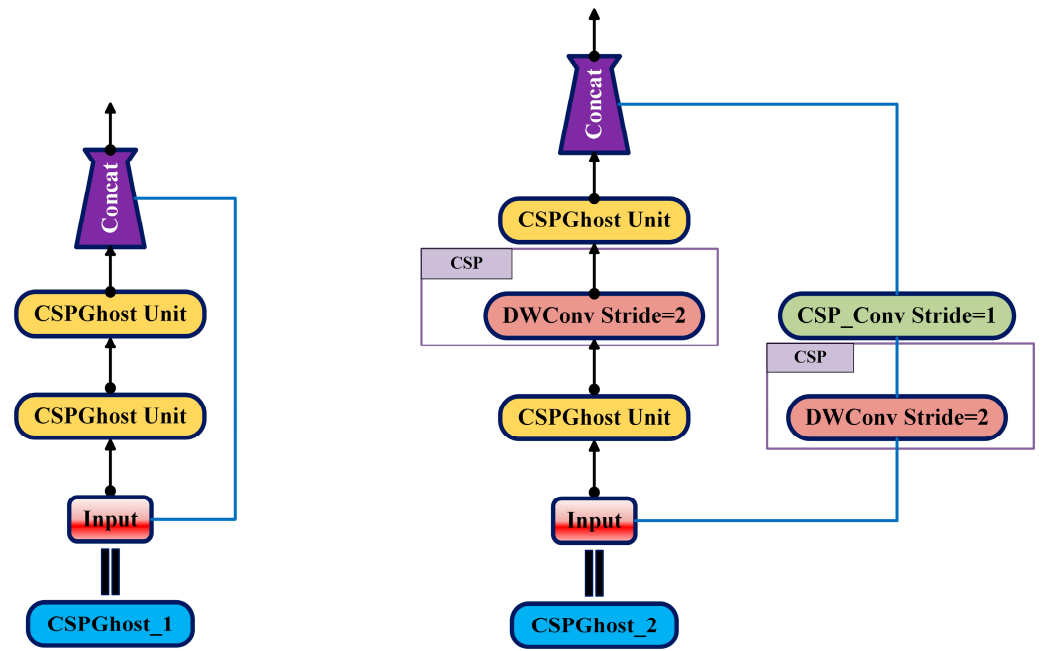


Figure 5. Two basic CSPGhost units for a lightweight backbone feature extraction network.

By stacking the CSPGhost_1 and CSPGhost_2 structures, we propose a new lightweight backbone feature extraction network: CSPGhostNet. By alternately using the two basic structures, the feature information of each level of the original sample is continuously extracted, and three effective features for the YOLO Neck part are generated.

2.4. Lightweight Target Detection Network: CGBi_YOLO

Based on the above improved CSPGhost bottleneck and combined with the YOLOV4_CSPBi target detection network model, a more lightweight target detection network model, CGBi_YOLO, is proposed. Figure 6 shows the architecture of the lightweight detection model.

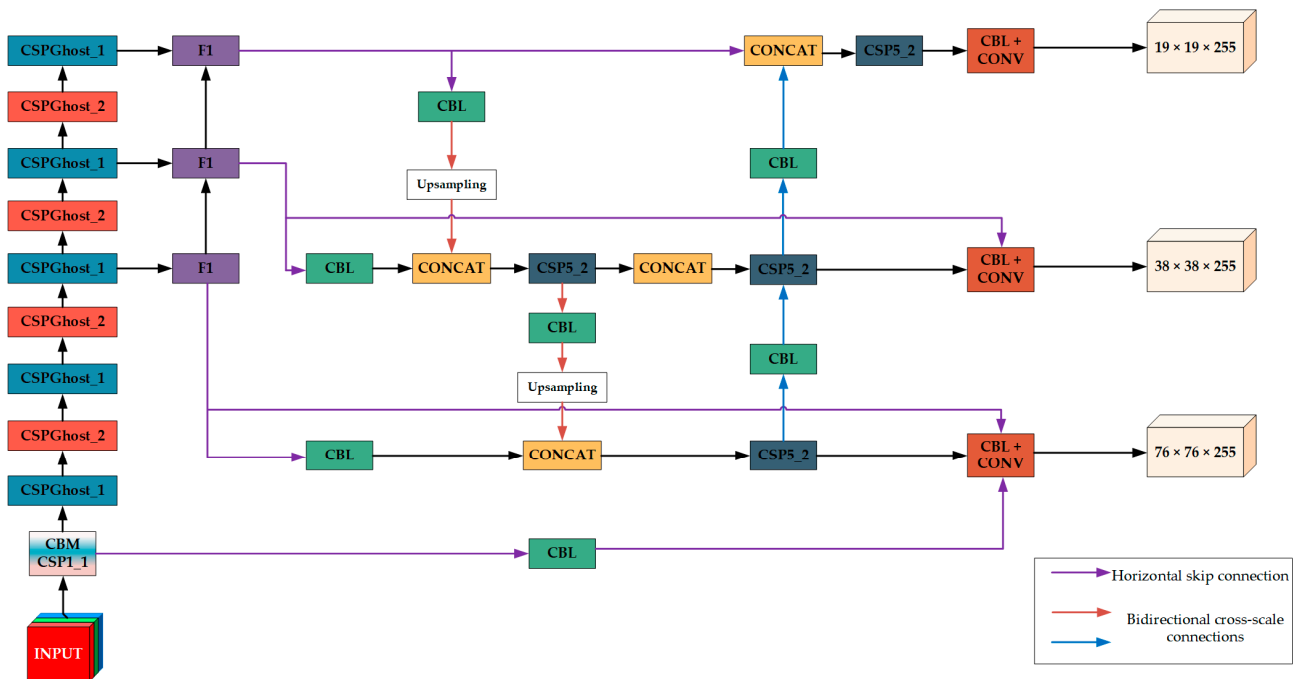


Figure 6. The architecture of the CGBi_YOLO network.

CGBi_YOLO is based on the framework of the original YOLOV4_CSPBi network, replacing the backbone feature-extracting structure CSPDarkNet with CSPGhostNet. It first performs the CSP1 operation on the original input sample image to obtain the original feature map. Subsequently, three different stages of feature extractors are used to generate an effective YOLO Neck feature map. These feature extraction structures deepen the channel depth of the feature map layer by layer through the combination of different CSPGhost modules.

By replacing the backbone feature extraction structure of the YOLOV4_CSPBi network, the dimension of the convolution operation in the backbone network is reduced, thereby improving the network inference speed. At the same time, based on the advantages of YOLOV4_CSPBi in bidirectional feature fusion, CGBi_YOLO does not significantly reduce network performance while reducing computational costs. Moreover, in the backbone feature extraction network structure of CGBi_YOLO, the computational cost of each basic computing unit compared to the basic unit of CSPDarkNet in YOLOV4_CSPBi is reduced by about 30–50%, so the overall computing cost will be greatly reduced.

3. Data Processing

3.1. DOTA Dataset

This study employs the DOTA dataset [28], specifically designed for object detection tasks in visible light remote sensing images.

Figure 7 illustrates an annotation example from the DOTA dataset, which includes common small-scale objects in remote sensing images, such as airplanes, vehicles, ships, and sports fields. The bounding boxes in the figure are in random colors to represent different object categories.



Figure 7. An annotated image example from the DOTA dataset.

Table 1 presents the detailed parameters of the DOTA dataset.

Table 1. The information and characteristics of the DOTA Dataset.

Name	DOTA
Creator	Wuhan University (Wuhan, China)
Categories	15
The number of tagged images	2806
The number of real object detection tasks	188,282
Application scenarios	Target detection of visible light remote sensing images
Size	800 × 800 pixels~2000 × 2000 pixels
Data sources	China Resources Satellite Data and Application Center (Beijing, China) (GF-2 and JL-1) CycloMedia B.V. (Zaltbommel, The Netherlands) (Google Earth and Optical remote sensing images)

3.2. Data Processing and Augmentation

If the feature information of the training sample is too small, minor image alterations can significantly impact prediction capabilities. Dataset augmentation is an effective solution to solve this problem and improve model performance. This study explores several mainstream image data enhancement techniques, such as the slide overlapping area image cutting, pixel-level enhancement, and area random erase, and the adaptive improvement and combination of the characteristics of visible light remote sensing images were carried out.

3.2.1. Slide Overlapping Area Image Cutting

The image size of the DOTA dataset varies in size, and the maximum size can reach 20,000 × 20,000 pixels, making the model difficult to train. To address this, we segmented the original DOTA image files into 832 × 832 pixel sections. Since direct cutting will cause the loss of label frame information, we incorporate the concept of sliding windows to achieve image overlap at uniform intervals. The original image is cut using 50% of the step size, and the image file without annotation information after cutting is removed. At last, as shown in Figure 8, 55,992 optical remote sensing images are obtained for the model training.

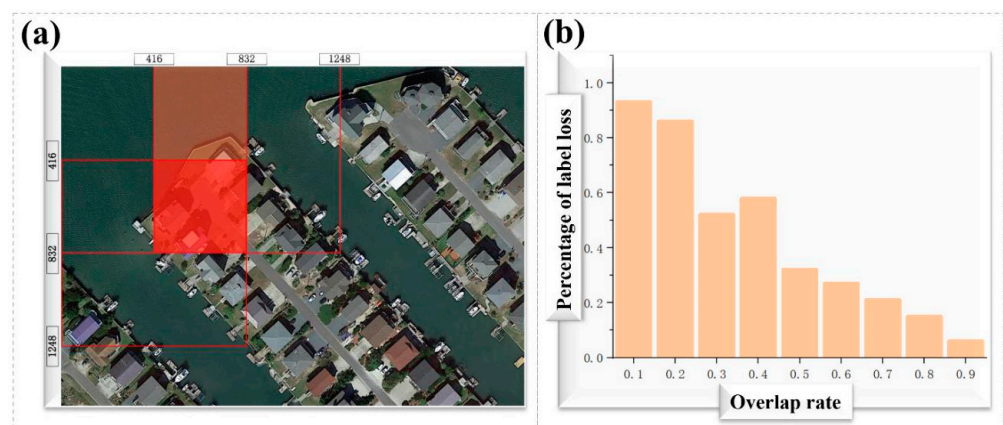


Figure 8. Diagram of the sliding overlapping area image cutting method. (a) An example of segmentation is when the overlapping area ratio is 0.5. (b) The proportion of label loss is due to different overlapping area ratio cuts.

3.2.2. Enhancement on Pixel Level

Multiple enhancement methods were applied to the original images, taking into account the characteristics of remote sensing imagery. The pixel-level enhancement methods used are shown in Figure 9.

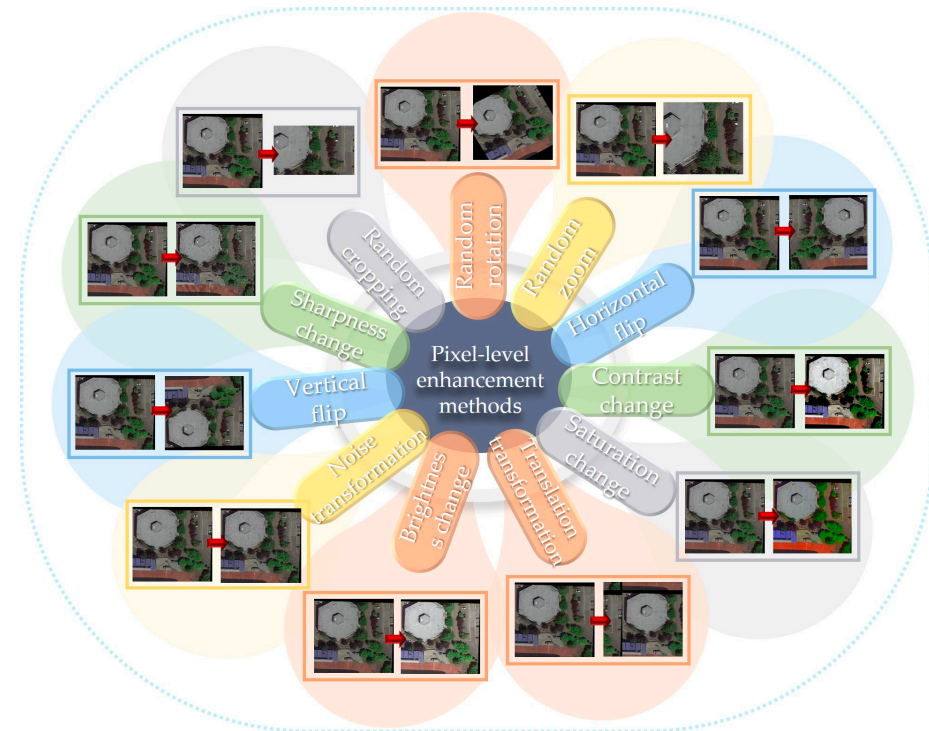


Figure 9. The pixel-level enhancement method has been used in this study and the samples of them.

All enhancement methods utilized the relevant API interfaces and default parameter settings provided by the TensorFlow2 framework. In this study, approximately 60% of the original samples were randomly selected, and three random enhancement methods were applied to them.

3.2.3. Area Random Erase Methods

The region random erase method improves generalization ability and robustness by simulating the occlusion of the target instance. However, in specific operations, attention should be paid to controlling the size and density of the occlusion area to prevent the target instance from being completely occluded or not blocked at all. The GridMask [29] method achieves balance by evenly distributing occlusion areas and adjusting parameters. If there is a target instance below 60 pixels in the example image, use the Mosaic method that extends from the idea of Cutmix [30] instead. The two area random erase methods are shown in Figure 10.

Table 2 shows the processed dataset information and characteristics.

Table 2. The information and characteristics of the processed dataset.

Item	Description
Number of Samples	87,382
Size	832 × 832
Target Frames	334,585
Format	Pascal VOC label
File Format	xml
Syntax Specification	xml
Annotation Information	Filename, Size, Object_Name, Pose, Truncated, Difficulty

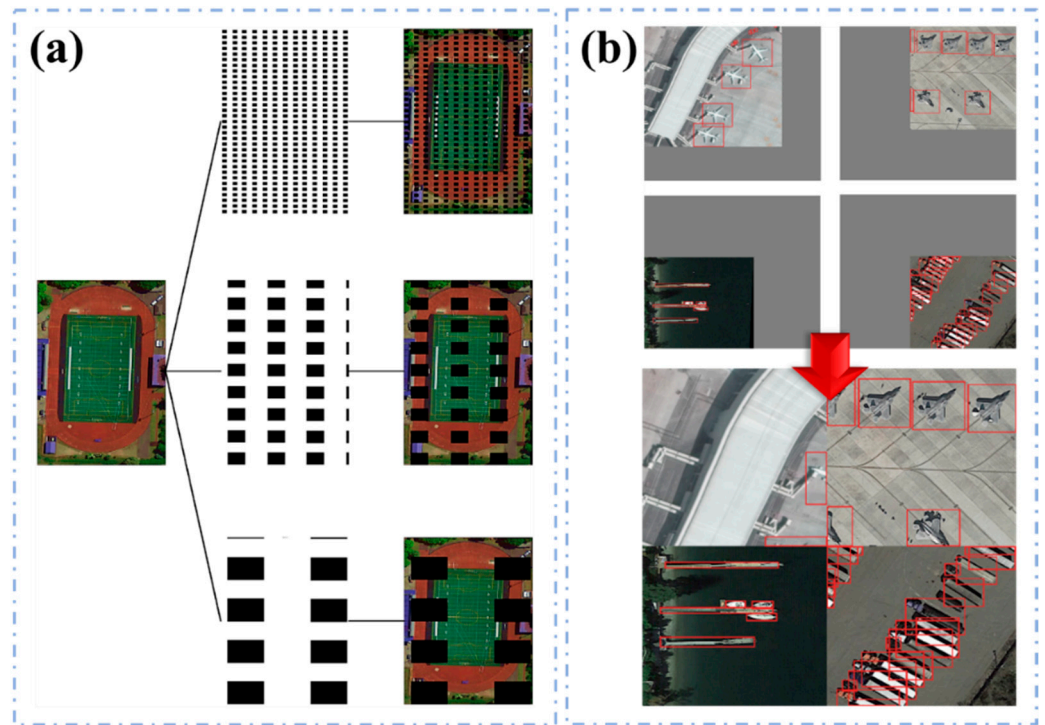


Figure 10. Area random erase methods. (a) Example of applying the GridMask region random erase method. (b) Example of the Mosaic stitching method.

4. Experiments and Results

4.1. Performance Evaluation Metrics

This study employs standard performance metrics for target detection models, as outlined in Table 3. The primary evaluation metric, mAP@0.5, is derived from precision and recall rates.

Table 3. Common Performance Metrics.

True Label	Prediction Results	Common Performance Metrics
True	Positive	TP
True	Negative	TN
False	Negative	FN
False	Positive	FP

Recall, defined as the ratio of correctly identified targets to the total number of relevant targets in the test dataset, is computed using Equation (8):

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

Precision, representing the proportion of correctly detected objects among all detections, is calculated via Equation (9):

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

The *Precision* and *Recall* values are represented by two tuples. The area above the two-dimensional coordinate axis enclosed by all the two-tuples is calculated for each category under a predefined IoU threshold. These areas correspond to the Average Precision (*AP*)

values, which are averaged to obtain the mean Average Precision (mAP) index value. Equations (10) and (11) provide the mathematical definition of AP and mAP .

$$AP = \int_0^1 P(y)dy \quad (10)$$

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (11)$$

4.2. Results and Analysis

4.2.1. Detection Performance of CGBi_YOLO

Initial attempts to train CGBi_YOLO's backbone network directly on the DOTA dataset revealed some challenges, including pronounced oscillations and prolonged convergence during early training phases, ultimately leading to suboptimal model performance. To mitigate these issues, we implemented a two-stage training strategy. We pre-trained the backbone feature extraction network on the VOC dataset at the beginning, and then completed fine-tuning on DOTA. This approach leverages transfer learning principles to establish a more robust initial feature representation, potentially accelerating convergence and enhancing final model performance.

Our training protocol, therefore, encompasses two distinct phases: backbone network pre-training and subsequent fine-tuning. Table 4 delineates the specific parameter settings for each stage, including key hyperparameters such as learning rate, batch size, and epoch count. This bifurcated approach allows for more nuanced optimization, tailoring the learning process to the unique characteristics of each dataset and the evolving needs of the model during different training phases.

Table 4. Parameter settings in the pretraining stage and fine-tuning stage.

Setting	Pretraining	Fine-Tuning
Epoch	50	150
Learning Rate	10^{-3}	10^{-4}
Batch Size	16	8
Early Stop	Yes	No
Optimizer		Adam
CSPGhost Unit Size	3	5

The experiment conducted evaluates the performance of three networks: the traditional YOLOV4, the YOLOV4_CSPBi, and the lightweight CGBi_YOLO proposed in this paper. Table 5 shows the CGBi_YOLO network effectively maintains performance comparable to that of YOLOV4_CSPBi while also reducing computational demands. Notably, CGBi_YOLO achieves performance on par with or slightly below YOLOV4_CSPBi across various categories, indicating its efficacy in balancing efficiency and accuracy.

To further evaluate the efficiency of our proposed model, we conducted a comparative analysis of the model size and inference speed across the three networks: YOLOV4, YOLOV4_CSPBi, and CGBi_YOLO. This comparison provides crucial insights into the computational efficiency and practical applicability of each model. Table 6 presents a comprehensive overview of these performance metrics, allowing for a direct assessment of the trade-offs between model complexity and operational speed.

It can be seen from a comprehensive comparison of the sizes of Weights, FPS, and mAP that CGBi_YOLO completely surpasses YOLOV4 and YOLOV4_CSPBi in terms of computation complexity and inference speed. The performance in mAP is basically the same as that of YOLOV4_CSPBi. CGBi_YOLO can ensure that the mAP is only reduced by about 0.6% when the number of parameters is compressed to only 70% of YOLOV4_CSPBi, and its inference speed is 15% and 36% faster than that of YOLOV4 and YOLOV4_CSPBi,

respectively. This evidently proves the effectiveness of the proposed lightweight backbone feature extraction.

Table 5. Comparison of detection performance of CGBi_YOLO.

Target Category	YOLOV4	YOLOV4_CSPBi	CGBi_YOLO
Basketball-Court	81.69	81.2	81.77
Storage-Tank	70.73	72.88	73.16
Soccer-Ball-Field	61.94	62.39	61.84
Roundabout	60.11	63.03	63.15
Harbor	71.08	77.71	76.82
Swimming-Pool	68.27	76.25	75.47
Helicopter	48.81	55.44	55.03
Tennis-Court	86.71	87.96	87.31
Plane	84.92	88.47	87.64
Baseball-Diamond	79.58	80.17	78.82
Bridge	46.62	48.73	46.66
Ground-Track-Field	71.78	75.51	75.17
Small-Vehicle	70.67	73.38	71.18
Large-Vehicle	63.29	69.57	68.96
Ship	77.37	79.42	79.14
TOTAL_mAP@0.5	69.57	72.8	72.14

Table 6. Comparison of network model parameters and inference speed of CGBi_YOLO.

Contrast Parameters	YOLOV4	YOLOV4_CSPBi	CGBi_YOLO
Weights (M)	64.50	66.37	47.21
FPS	45	42	52
TOTAL_mAP@0.5	69.57	72.8	72.14

4.2.2. Ablation Study on Compression Ratio and Convolution Size

There are two important parameters in the backbone feature extraction, namely compression ratio, which is the channel compression ratio of the input feature, and kernel size, which is the size of the depth-wise separable convolution. The ablation studies of the two parameters are carried out separately to analyze the impact of different parameter settings on the model performance.

Our initial investigation focused on optimizing the kernel size of the depth-wise separable convolution. Based on the widely accepted principle that halving the channel dimension of input features generally preserves essential information while achieving effective compression, we fixed the compression ratio at 2 for this experiment. We evaluated four distinct kernel size configurations: 1×1 , 3×3 , 5×5 , and 7×7 . This systematic approach allowed us to assess the impact of varying kernel dimensions on model performance and efficiency. Table 7 presents a comprehensive summary of our experimental findings, elucidating the relationship between kernel size and key performance metrics.

Table 7. Performance on kernel size parameter.

Kernel Size	Weights (M)	mAP	FPS
CGBi_YOLO_1 × 1	21.3	48.33	57
CGBi_YOLO_3 × 3	47.13	71.68	54
CGBi_YOLO_5 × 5	47.48	71.94	52
CGBi_YOLO_7 × 7	47.62	57.71	49

It can be seen that the CGBi_YOLO network can show better performance when the operation kernel size is set to 3×3 or 5×5 . There is no meaningful spatial semantic information that can be extracted from the original feature map for the operation kernel

size of 1×1 ; although its computational cost is the lowest, the mAP score is up to 32% lower than the other three experiments. If the value is set to 7×7 , it will also cause a sudden drop in network performance, as shown in Figure 11.

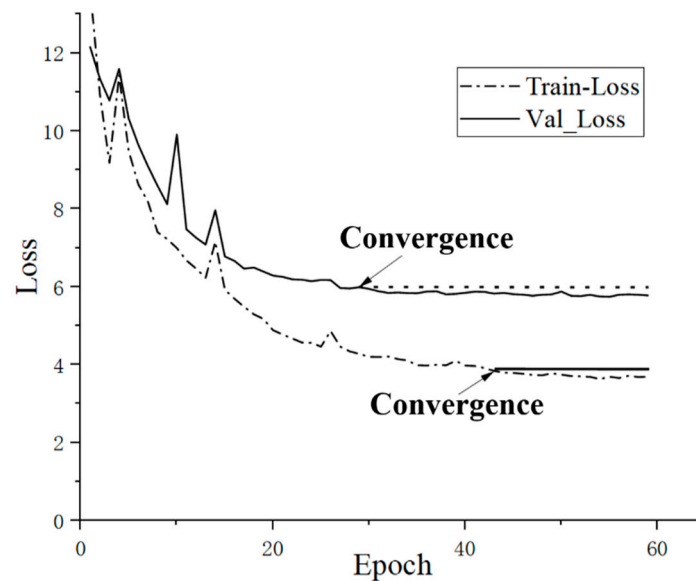


Figure 11. Overfitting of CGBi_YOLO (with kernel size: 7×7).

When using a kernel size of 7×7 , the network has obvious overfitting during the training process, which is also the reason for the sudden drop in network performance. Therefore, both 3×3 and 5×5 kernel size settings are used in the CGBi_YOLO network structure described above.

After determining the size of the kernel size parameter, an ablation study was performed on the optimal setting of the compression ratio parameter within the range of. From Table 8, the compression ratio is directly related to the parameters of the network and its computational cost. A larger compression ratio brings a more extreme compression ratio and acceleration ratio, but as expected, the detection accuracy will be greatly reduced. Therefore, we set the optimal compression ratio to be 2.

Table 8. Ablation study on the compression ratio parameter.

Compression Ratio	Weights (M)	mAP	FPS
2	47.13	71.68	54
3	39.67	63.84	63
4	29.36	50.61	78
5	18.92	37.22	102

4.2.3. Ablation Study on Backbones

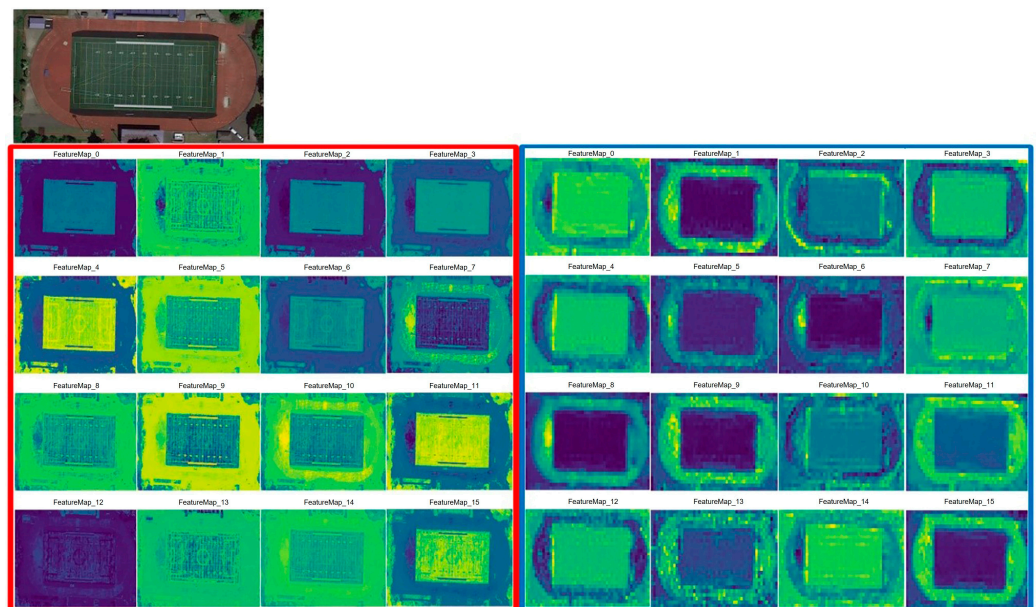
To further verify the performance improvement brought by the CSPGhostNet backbone feature extraction network proposed in this paper, ablation studies based on different backbone feature extraction networks were set up. We take YOLOV4 as the base model.

Table 9 presents a comparative analysis of various lightweight network versions. The results demonstrate that employing CSPGhostNet as the backbone feature extractor yields a model with reduced parameters and enhanced inference speed while maintaining mAP performance. These findings validate the efficacy of our proposed lightweight architecture.

Table 9. Detection performance with various backbones.

Object Detection Framework	Backbone Feature Extraction Network	Weights (M)	mAP	FPS
YOLOV4 [16]	MobileNetV1 [31]	48.29	72.53	52
	MobileNetV2 [27]	46.74	68.29	54
	MobileNetV3 [32]	47.83	71.96	53
	ShuffleNetV2 [33]	45.16	66.58	53
	CGBi_YOLO	47.21	72.14	53

Since the CSPGhost structure greatly reduces the parameters of the network, whether its ability to extract feature information will be greatly weakened needs to be further explored. For this reason, we visualize the network after the initial convolution and the first time calculated by the CSPGhost structure. Figure 12 illustrates the feature maps generated by CGBi_YOLO's initial convolution (red border) and the CSPGhost structure (blue border).

**Figure 12.** Visualization after the first convolution of CGBi_YOLO (red border) and the first calculation of CSPGhost Unit (blue border).

The visualization reveals the preservation of anticipated redundant feature maps. Notably, the CSPGhost-derived features exhibit more abstract semantic content compared to those from the initial convolution. This visual evidence supports the CSPGhost structure's capability to efficiently capture effective redundant feature information, despite its low computational cost.

5. Discussion

This study focuses on advancing target detection in optical remote sensing imagery through innovations in YOLO-based network architectures. Our experimental outcomes corroborate the efficacy of the proposed methodologies. The following discussion examines the broader implications of our findings and identifies areas for future investigation.

1. Lightweight Network Improvement:

One notable achievement of this study is the successful reduction of the overall computational burden of the network while preserving its performance. While our focus is primarily centered on optimizing the backbone network, we acknowledge the need to address redundant calculations in the neck part. The complex fusion mechanism akin to

PANet has shown substantial enhancement in prediction performance; however, it carries redundancy in computation. Our future efforts will be dedicated to achieving a lightweight enhancement for this component of the network, which is pivotal to its efficiency.

2. Contextual Understanding and Reasoning:

An imperative area for future research involves advancing the incorporation of contextual understanding and reasoning in target detection methodologies. While existing methods have incorporated some level of contextual and global information, they predominantly remain grounded in visual features. To achieve more comprehensive and interpretable results, the integration of high-level semantic knowledge should be explored. This could enhance the model's capability to comprehend and infer from the broader image context, thus leading to improved interpretability.

3. Semantic Interpretation:

A significant challenge in contemporary target detection methodologies is the insufficient integration of high-level semantic knowledge, which impedes model interpretability. Addressing this limitation presents a crucial avenue for enhancing the practical applicability of our approach. Future research should focus on incorporating advanced semantic reasoning capabilities into the network architecture. Such enhancements could potentially enable the model to process and interpret contextually rich information, thereby expanding its utility across diverse domains. By bridging the gap between low-level feature detection and high-level semantic understanding, we anticipate opening new directions in the intelligent analysis of remote sensing imagery.

6. Conclusions

This study presents a novel approach to enhance target detection in optical remote sensing imagery through modifications to YOLO-series network models. Our proposed CGBi_YOLO architecture demonstrates significant improvements in both performance and efficiency, as validated by comparative experiments. Notably, CGBi_YOLO achieves a 30% reduction in parameters compared to YOLOV4_CSPBi, with only a marginal 0.6% decrease in mAP. Moreover, it exhibits accelerated inference speeds, surpassing YOLOV4 and YOLOV4_CSPBi by 15% and 36%, respectively. These results underscore the efficacy of our lightweight backbone feature extraction network, outperforming comparable models in its class. The experimental outcomes conclusively validate the effectiveness of the CGBi_YOLO methodology in achieving its intended objectives of improved efficiency without significant performance compromise.

The novel lightweight backbone network based on CSPGhost, combined with the YOLOV4_CSPBi model, has showcased a robust potential for practical deployment. Despite these achievements, we acknowledge certain shortcomings that beckon further research. The pursuit of a more streamlined computation strategy in the neck part, alongside a deeper integration of contextual reasoning, stands as a critical future direction. By augmenting the model's ability to comprehend and interpret the broader image context, we aim to elevate the semantic understanding and utility of our approach. This study contributes to the advancement of target detection methodologies, particularly in the realm of optical remote sensing images, while simultaneously highlighting areas for continued investigation and improvement. We anticipate that our findings will motivate and guide subsequent research endeavors in pursuit of even more refined and interpretable target detection systems.

Author Contributions: Conceptualization, L.Y., W.Z. and L.W.; methodology, R.W. and X.C.; software, X.C., R.W. and J.T.; formal analysis, S.L. and L.Y.; data curation, R.W. and J.T.; writing—original draft preparation, R.W., L.Y., S.L. and W.Z.; writing—review and editing, L.Y., L.W. and W.Z.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Support by Sichuan Science and Technology Program (2023YFH0004).

Data Availability Statement: The DOTA dataset, which serves as a valuable resource for validating the results obtained in this study, is publicly accessible at the following location: <https://captain-whu.github.io/DOTA/index.html> (accessed on 26 November 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Zhou, X.; Shen, K.; Weng, L.; Cong, R.; Zheng, B.; Zhang, J.; Yan, C. Edge-Guided Recurrent Positioning Network for Salient Object Detection in Optical Remote Sensing Images. *IEEE Trans. Cybern.* **2023**, *53*, 539–552. [CrossRef] [PubMed]
- Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
- Zhang, K.; Shen, H. Multi-stage feature enhancement pyramid network for detecting objects in optical remote sensing images. *Remote Sens.* **2022**, *14*, 579. [CrossRef]
- Wu, S.; Liu, Y.; Liu, S.; Wang, D.; Yu, L.; Ren, Y. Change detection enhanced by spatial-temporal association for bare soil land using remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *17*, 150–161. [CrossRef]
- Li, L.; Wang, L.; Du, A.; Li, Y. LRDE-Net: Large receptive field and image difference enhancement network for remote sensing images change detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *17*, 162–174. [CrossRef]
- Xie, X.; You, Z.-H.; Chen, S.-B.; Huang, L.-L.; Tang, J.; Luo, B. Feature enhancement and alignment for oriented object detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *17*, 778–787. [CrossRef]
- Tuia, D.; Persello, C.; Bruzzone, L. Recent advances in domain adaptation for the classification of remote sensing data. *arXiv* **2021**, arXiv:2104.07778. [CrossRef]
- Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
- He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
- Girshick, R. Fast r-cnn. *arXiv* **2015**, arXiv:1504.08083.
- Ren, S. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497. [CrossRef] [PubMed]
- Redmon, J. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; pp. 21–37.
- Redmon, J. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
- Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
- Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
- Xu, Y.; Zhu, M.; Xin, P.; Li, S.; Qi, M.; Ma, S. Rapid airplane detection in remote sensing images based on multilayer feature fusion in fully convolutional neural networks. *Sensors* **2018**, *18*, 2335. [CrossRef] [PubMed]
- Ghorbani, F.; Ebadi, H.; Sedaghat, A. Geospatial target detection from high-resolution remote-sensing images based on PIIFD descriptor and salient regions. *J. Indian Soc. Remote Sens.* **2019**, *47*, 879–891. [CrossRef]
- Cao, C.; Wu, J.; Zeng, X.; Feng, Z.; Wang, T.; Yan, X.; Wu, Z.; Wu, Q.; Huang, Z. Research on airplane and ship detection of aerial remote sensing images based on convolutional neural network. *Sensors* **2020**, *20*, 4696. [CrossRef]
- Xie, S.; Zhou, M.; Wang, C.; Huang, S. CSPPartial-YOLO: A lightweight YOLO-based method for typical objects detection in remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *17*, 388–399. [CrossRef]
- Wang, J.; Li, X.; Zhou, L.; Chen, J.; He, Z.; Guo, L.; Liu, J. Adaptive receptive field enhancement network based on attention mechanism for detecting the small target in the aerial image. *IEEE Trans. Geosci. Remote Sens.* **2023**, *62*, 5600118. [CrossRef]
- Qin, R.; Liu, Q.; Gao, G.; Huang, D.; Wang, Y. MRDet: A multihead network for accurate rotated object detection in aerial images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5608412. [CrossRef]
- Yin, L.; Wang, L.; Li, J.; Lu, S.; Tian, J.; Yin, Z.; Liu, S.; Zheng, W. YOLOV4_CSPBi: Enhanced land target detection model. *Land* **2023**, *12*, 1813. [CrossRef]
- Vicente, S.; Carreira, J.; Agapito, L.; Batista, J. Reconstructing pascal voc. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 41–48.
- Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.

27. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 4510–4520.
28. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
29. Chen, P.; Liu, S.; Zhao, H.; Wang, X.; Jia, J. Gridmask data augmentation. *arXiv* **2020**, arXiv:2001.04086.
30. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6023–6032.
31. Howard, A.G. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
32. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
33. Ma, N.; Zhang, X.; Zheng, H. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerland, 2018.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.