CSPPartial-YOLO: A Lightweight YOLO-Based Method for Typical Objects Detection in Remote Sensing Images

Siyu Xie, Mei Zhou, Chunle Wang, and Shisheng Huang

***Abstract*—Detecting and recognizing objects are crucial steps in interpreting remote sensing images. At present, deep learning methods are predominantly employed for detecting objects in remote sensing images, necessitating a significant number of floating-point computations. However, low computing power and small storage in computing devices are hard to afford the large model parameter quantity and high computing complexity. To address these constraints, this article presents a lightweight detection model called CSPPartial-YOLO. This model introduces the partial hybrid dilated convolution (PHDC) Block module that combines hybrid dilated convolutions and partial convolutions to increase the receptive field at a low computational cost. By using the PHDC Block within the model design framework of cross-stage partial connection, we construct CSPPartialStage that reduces computational burden without compromising accuracy. Coordinate attention module is also employed in CSPPartialStage to aggregate position information and improve the detection of small objects with complex distributions in remote sensing images. A backbone and neck are developed with CSPPartialStage, and the rotation head of the PPYOLOE-R model adapts to objects of multiple orientations in remote sensing images. Empirical experiments using the dataset for object deTection in aerial images (DOTA) dataset and a large-scale small object detection dAtaset (SODA-A) dataset indicate that our method is faster and resource efficient than the baseline model (PPYOLOE-R), while achieving higher accuracy. Furthermore, comparisons with current state-of-the-art YOLO series detectors show our proposed model's competitiveness in terms of speed and accuracy. Moreover, compared to mainstream lightweight networks, our model exhibits better hardware adaptability, with lower inference latency and higher detection accuracy.**

***Index Terms*—Deep learning, object detection, partial convolution, remote sensing image.**

I. Introduction

REMOTE sensing images possess a broad range of applications, including Traffic Monitoring [1], Maritime Rescue [2], and Aviation Control [3]. The development of deep learning technology has resulted in more intelligent and efficient analysis of remote sensing images, decreasing the reliance on manual work. Object recognition and detection are fundamental tasks in computer vision and are core components of the analysis of remote sensing images.

Deep learning-based object detectors can be categorized into two groups: 2-stage detectors, including R-CNN [4], Mask R-CNN [5], Faster-RCNN [6], and others. In 2014, Ross et al. proposed R-CNN as the first two-stage object detection algorithm. This method first utilizes selective search to extract candidate frames, then feeds them through a convolutional neural network to extract target characteristics, and performs support vector machine classification and frame calibration on the target characteristics. On the other hand, single-stage detectors, including YOLO [7] and SSD [8], treat the detection process as regression, eschewing the region proposal stage to reduce computation time, thus achieving faster detection. Joseph et al. introduced YOLO in 2015, dividing the image into a grid and providing predicted bounding boxes in each division. Finally, redundant predicted boxes were removed using the nonmaximum suppression (NMS) method. YOLOv2 [9] expands the detection dataset through joint training, YOLOv3 [10] uses Darknet53 as a backbone network to boost detection performance, and YOLOv4 [11] utilizes CIoU loss for predictive frame filtering to improve the model's convergence. YOLOv5 [12] uses the feature pyramid network (FPN) and pixel aggregation network (PAN) structure in the neck network, achieving superior speed with equivalent precision to YOLOv4 due to its lighter model size. YOLO-X [13] reintroduced the anchor-free technique to the YOLO series, proposing the SimOTA label assignment method and decoupled detection head to separate classification and location issues, thereby producing higher quality predicted bounding boxes. PPYOLOe [14] introduced advanced technologies such as reparameterization, redesigns the backbone network, and achieves a good balance between speed and accuracy on the MS COCO dataset. In addition, the PPYOLOE-R [15] model is more suitable for multidirectional object distribution in remote sensing images by designing a detection head for rotating boxes and angle loss.

In spite of achieving good results on general datasets, object detectors face challenges including large parameter volume and high computational limits and limited storage space required for the surveillance applications in real time. Although the use of

Manuscript received 24 July 2023; revised 5 October 2023; accepted 24 October 2023. Date of publication 1 November 2023; date of current version 23 November 2023. *(Corresponding author: Mei Zhou.)*

Siyu Xie is with the Department of Space Microwave Remote Sensing System, Aerospace Information Research Institute, Chinese Academy of Science, Beijing 100190, China, and also with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: xiesiyu21@mails.ucas.ac.cn).

Mei Zhou and Chunle Wang are with the Department of Space Microwave Remote Sensing System, Aerospace Information Research Institute, Chinese Academy of Science, Beijing 100190, China (e-mail: zhoumei@aircas.ac.cn; clwang@mail.ie.ac.cn).

Shisheng Huang is with the Beijing Institute of Tracking and Telecommunications Technology, Beijing 100094, China (e-mail: huangss@nudt.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3329235

deep and wide pruning can decrease the model size as seen with YOLOv5 model versions like L, M, S, and N, simple pruning of the model depth and feature map channel numbers can weaken model representation ability, which results in performance degradation. While objects in remote sensing aerial perspective have small size, the model can easily lose important features during the process of downsampling, therefore extracting adequate features for accurate detection becomes difficult.

In order to address the real-time processing issue of object detection in remote sensing image interpretation, many scholars have designed lightweight object detection models based on the characteristics of remote sensing images. Guo et al. [16] used depthwise separable convolution to replace standard convolution, reducing the model's parameter volume. They also proposed the ACON activation function, which effectively avoids neuronal death in large gradient propagation. In addition, they introduced the DSASFF module, which effectively aggregates the target properties at different scales that are neglected during feature fusion. Cui et al. [17] introduced prior knowledge of the Laplacian operator into the Bottleneck and added a sharp value transition based on the original tensor to enhance the low-level feature tensor that contains small target contours. They concurrently decreased the parameter volume and computational complexity of the model by employing multiple small convolutional kernels in place of larger ones. Zhang et al. [18] used the ShuffleV2 module to construct a lightweight FPN network that fully fuses shallow and deep features to generate an abundant fused feature map with rich object position information, thereby improving the ability to locate targets of the original model. Lyu et al. [19] took inspiration from Liu's [20] utilization of large kernel convolution to enhance the detector's performance. However, in order to balance efficiency, they employed depthwise separable convolution. The RTMDet model they designed achieved a good balance between parameters and accuracy.

In comparison with the aforementioned methods, this article focuses on the redundant feature maps in the process of model inference. Inspired by [21], this article uses pointwise convolution and partially connected layers to construct module stages and improve the PPYOLOE-R model, thereby proposing a lightweight and efficient object detector called CSPPartial-YOLO. Specifically, the primary contributions of this research are as follows:

1) We present the partial hybrid dilated convolution (PHDC) Block module, which combines partial convolution and pointwise convolution to fully utilize the redundancy of the feature map and reduce the model's parameters and burden on computation. In addition, hybrid dilated convolutions are used in the partial convolution to reduce the computational burden on large sized convolution kernels as well as to enlarge the receptive field to accurately extract small targets in complex background of remote sensing images and improve the problem of long-range dependency.
2) The CSPPartialStage is constructed by integrating the PHDC Block with partial convolution and CSPNet to decrease computing complexity while simultaneously preserving comparable precision. At the end of the CSPPartialStage, a coordinate attention (CA) module is appended to enhance the module's object representation capability. A new backbone and neck were established using the CSPPartialStage, and a lightweight and efficient remote sensing image rotation box detector named CSPPartial-YOLO was developed based on the PPYOLOE-R detection head.
3) Undertaken experimental studies on typical objects in remote sensing images in the dataset for object deTection in aerial images (DOTA) dataset and a large-scale small object detection dAtaset (SODA-A) dataset. In terms of accuracy, the proposed model was compared with common YOLO models YOLOv8 YOLO X and RTMDet. The proposed model demonstrates superior results in terms of both volume and speed. In addition, the backbone of the proposed method exhibits dual advantages of both accuracy and speed when compared with common lightweight backbones such as MobileNet v3, ShuffleNet v2, and GhostNet.

The rest of this article is organized as follows. Section II provides a review of relevant research on lightweight backbone networks and mechanisms of attention. Section III presents a detailed description of the proposed model. Section IV presents the experimental details, including the findings and discussions from both ablation experiments and comparative experiments. Finally, Section V of this article summarizes the key findings and presents the conclusion of the study.

II. Related Work

A. Llightweight Backbone Networks

In recent years, deep neural networks have been advancing toward deeper and larger models, achieving continuous accuracy improvement across multiple benchmark datasets. However, a high number of parameters and computations pose a challenge for model applications. Researchers have explored lightweight backbone networks in an attempt to reduce the number of model parameters and computations, while still maintaining similar accuracy. In 2017, Howard et al. [22] introduced depthwise separable convolution (DSC) in MobileNet V1, decomposing standard convolution into depthwise convolution and pointwise convolution to effectively reduce the number of parameters and computations in convolutional layers. That same year, ShuffleNet V1 [23] used group convolutions to reduce computation and employed the ChannelShuffle operation to enhance the interchannel information flow, resulting in better performance compared to MobileNet V1. In 2018, ShuffleNet V2 [24] proposed four guidelines to optimize the model, further increasing the inference speed. In 2020, Han et al. [25] discovered the redundancy in feature maps via experiments and proposed GhostNet, which employed cheap operations to replace standard convolutional layers, generating additional feature maps while reducing the calculation cost. In 2023, Chen et al. [21] proposed the Partial Convolutional Module, which employs a combination of partial convolution and point convolution to reduce the computational cost while addressing feature redundancy. Building on the partial convolutional module, this study employs hybrid dilated convolution to expand the receptive field and enhance the module's feature extraction ability for small objects.

B. Attention Mechanism

Attention is a cognitive mechanism that imitates human ability to selectively focus on specific information and amplify key details to grasp the essence of data. Deep learning models employ attention mechanisms to improve their performance. Visual attention mechanisms in deep learning are classified into channel attention mechanisms and spatial attention mechanisms. The squeeze-and-excitation (SE) [26] block is a well-known module that performs dynamic attention on channel features. It utilizes global average pooling to compress the channels into a single value, which is then subject to nonlinear transformations via a fully connected network before being multiplied with the input channel vector as weights. ECA [27] reduces model redundancy and captures channel interactions by removing the fully connected layer and leveraging 1-D convolutional layers. Both SE and ECA apply attention mechanisms in the channel domain while ignoring the spatial one. CBAM [28] combines channel and spatial attention by exploiting large kernel size convolutions to aggregate positional information within a certain range. Nevertheless, this design choice leads to increased computational costs, making it less suitable for developing lightweight models. In addition, a single layer with a large convolutional kernel can only capture local position information instead of global position information. Coordinate attention (CA) [29] captures precise positional dependencies by embedding positional information into channel attention. This approach offers benefits for dense prediction tasks in lightweight networks. Incorporating the CA module into the CSPPartialStage results in an improvement in detection accuracy of typical targets in remote sensing images, at an acceptable computational cost.

III. Proposed Method

The CSPPartial-YOLO framework is based on the PPYOLOE-R model, but replaces the computationally resource-intensive RepVGG Block with the PHDC Block in the CSPStage. Furthermore, it embeds the coordinate attention module (CA) in the CSPStage and proposes the lightweight CSPSPartialStage feature extraction module. The model optimizes the depth ratio of different stages to construct the CSPSPartialNet backbone network. In addition, it employs the CSPPartialStage and SPP module to construct a bidirectional feature pyramid for enhancing the fusion of multiscale features. Fig. 2 shows the main structure of the proposed model comprising the backbone, neck, and detection head. The input is an image with three channels of 1024×1024 pixels. The backbone comprises four CSPSPartialStages with a Stem Block preceding them. The output of the last three CSPSPartialStages serves as both the output of the backbone and the input of the fusion module, with feature map sizes of 128×128, 64×64, and 32×32, respectively. Then, the lightweight bidirectional feature pyramid produces uniform feature maps to the detection head. Finally, the model employs the PPYOLOE-R rotation detection head to obtain target position, direction, and category information at multiple scales. The PHDC block is the cornerstone of the model construction. It combines hybrid dilated convolutions with partial convolutions efficiently to extract information at low computational cost, thereby enabling our model to achieve advantages in both speed and accuracy.

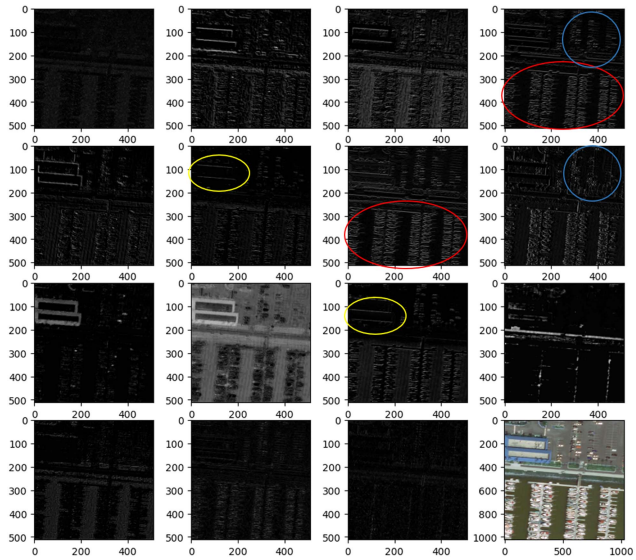


Fig. 1. Comparison of feature maps after the first few layers of convolution in a well-trained neural network. The last image represents the input image, and the circles with the same color indicate the parts with similar features.

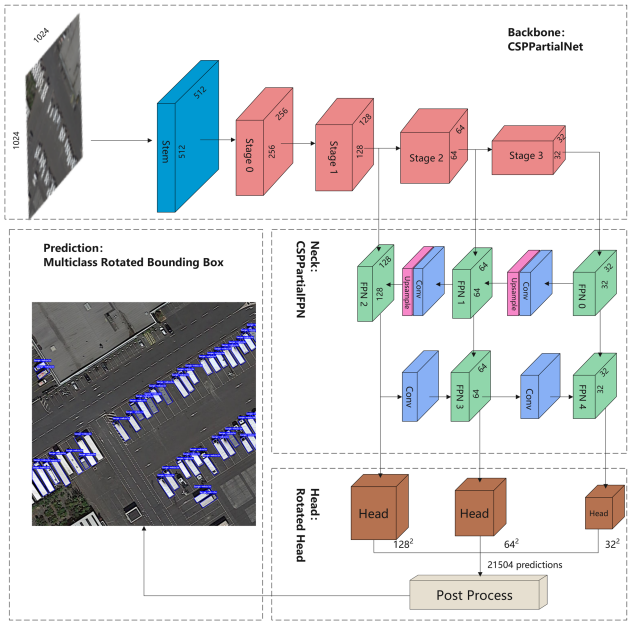


Fig. 2. Flowchart of the proposed method.

A. PHDC Block

Typical convolutional operations usually generate multichannel feature maps. Many studies [25], [30] have shown redundancies among these feature maps. Fig. 1 shows the redundancy of the feature maps.

Partial convolution is a lightweight convolutional operator that efficiently uses redundancies in feature maps, thereby reducing computational costs. Fig. 7(a) illustrates the workflow of partial convolution, which selectively applies convolution

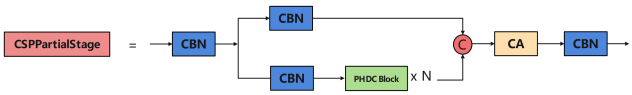


Fig. 3. Workflow of CSPPartialStage.

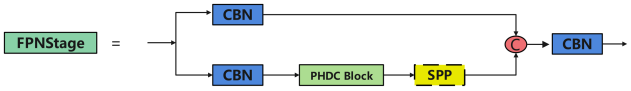


Fig. 4. Details of neck part.

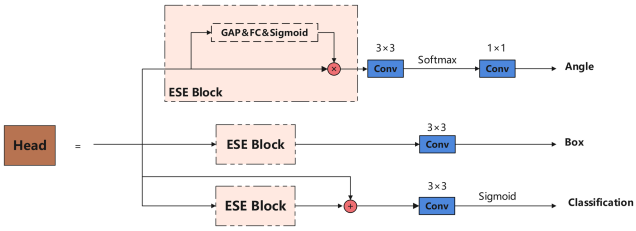


Fig. 5. Details of rotated detection head.

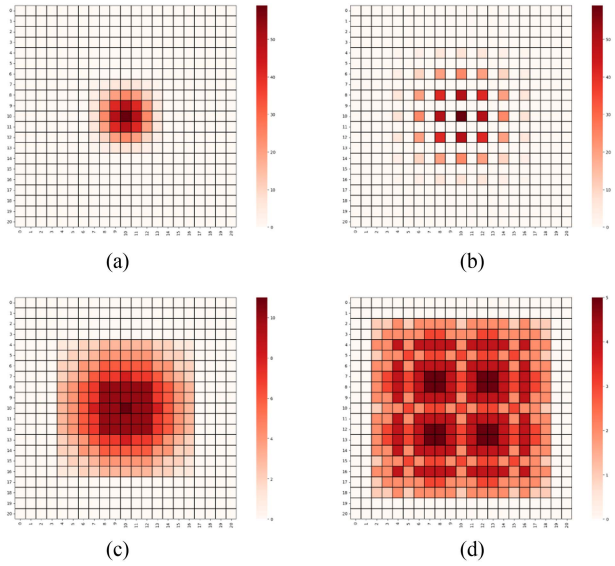


Fig. 6. Heatmap color, ranging from light to dark, indicates the number of times a single pixel involved in computation, among which (a) represents a traditional 3×3 convolution. As seen from the figure, dilated convolutions with hybrid dilation rates have a more comprehensive range of semantic understanding than the standard 3×3 convolution, and can capture more distant and highly related features. Other combinations of dilation rates are shown in subfigures (b), (c), and (d). (a) [1, 1, 1]. (b) [2, 2, 2]. (c) [1, 2, 3]. (d) [1, 2, 5].

operations to a portion of the input channels for spatial feature extraction while keeping the remaining channels unchanged. For sequential memory access, we consider the first or last consecutive C_p channels as representatives for the entire feature map to perform computations. The number of computations for a single forward propagation can be calculated as follows:

$$FLOPs = h \times w \times C_p^2 \times k^2. \tag{1}$$

Here h and w represent the height and width of the output feature map, C_p represents the number of channels involved in the convolution operation for partial convolution, and k represents the size of the convolution kernel. It can be seen that the computational cost of partial convolution is $\left(\frac{C_p}{C}\right)^2$ of that of standard convolution.

Partial convolution has a limited capacity to capture long-distance dependencies, which can impede small target detection in remote sensing images. Expanding the receptive field by utilizing a large convolution kernel such as 5×5 or 7×7 introduces more contextual correlation features to the model. Nevertheless, convolutional layers using large kernels result in an exponential increase in computational burden, challenging our aim of designing a lightweight model. To address this limitation, we use a hybrid dilated convolution (HDC) to replace the regular convolution operation in partial convolution, inspired by [31]. An affordable increase in computational complexity allows us to achieve a wider receptive field. Fig. 6 illustrates the size of the receptive field of three consecutive convolution layers using different dilation rates.

To ensure that the hole convolution group adequately covers the space range while avoiding the sampling loss caused by continuous hole convolutions on the input feature map, it is crucial to carefully select the combination of dilation rates used in HDC. Fig. 6(b) illustrates the adverse effects of improper hole rate combinations, which can cause HDCs to miss adjacent pixel points and result in incomplete feature sampling. In contrast, the [1, 2, 5] dilation rate combination in HDC, as used in this study, improves this situation by encompassing a larger receptive field range and providing comprehensive information on all related pixel points in adjacent areas, as compared with the [1, 2, 3] dilation rate combination shown in Fig. 6(c). Moreover, when combined with partially convolutional module, as displayed in Fig. 2, the HDC with [1, 2, 5] dilation rate combination further enhances the performance of the proposed network.

Partial convolution leads to an inevitable loss of channel information due to its inability to involve all channel features in convolutional operations. Nonetheless, this channel information loss can be mitigated by utilizing pointwise convolution after partial convolution. To achieve this, we apply pointwise convolution to the output of partial convolution, then follow up with a BatchNorm layer and a rectified linear unit (ReLU) activation function, before finally restoring channel dimensionality using pointwise convolution, as shown in Fig. 8(b). To help avert gradient vanish and explosion issues caused by excessively deep convolutional layers, we adopt residual connections as part of our PHDC Block module, which is consistent with the ResNet method [32]. The comparison in Fig. 8 reveals the main building blocks utilized in constructing the stage of the PPYOLOE-R model and the PHDC block that forms our model stage. Despite the implementation of a reparameterization hierarchy in the PPYOLOE-R model, its building blocks exhibit high computational complexity. In contrast, the PHDC block in our model features a simple and efficient structure.

Compared to the inverted residual module used in MobileNetV2 [33], the PHDC Block employs only BatchNorm layer and ReLU activation without performing depthwise convolution after channel expansion with pointwise convolution. This approach effectively avoids the frequent memory access caused

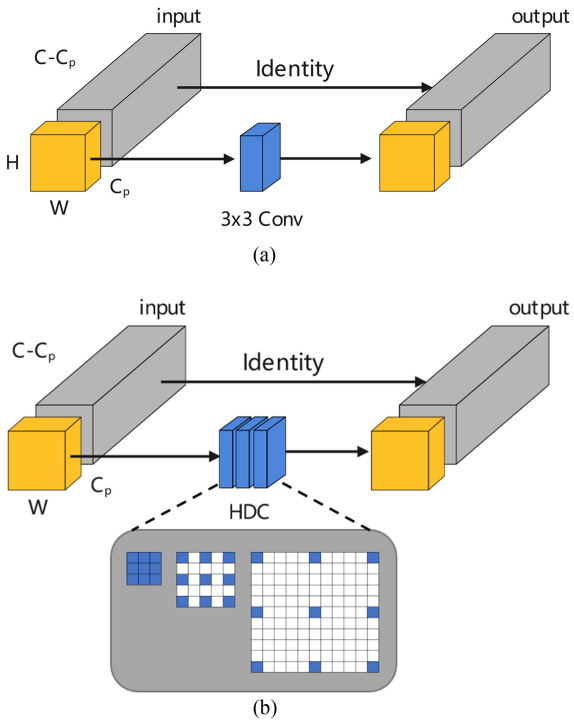


Fig. 7. Improve the PartialConv with HDC [1, 2, 5]. (a) Partial convolution with a single 3 × 3 convolutional layer. (b) Partial convolution with HDC ([1, 2, 5] dilation rates combination).

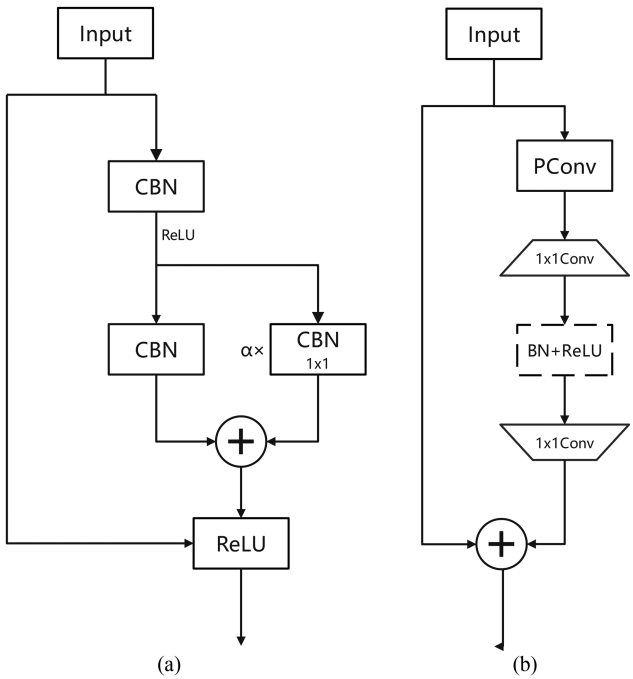


Fig. 8. Comparison of the basic block in PPYOLOE-R and the PHDC block in our model. (a) Basic block in PPYOLOE-R. (b) PHDC block.

by multiple groups of depthwise convolution, thereby increasing the operation efficiency of the module. Moreover, after training is completed, the BatchNorm layer can be easily merged into adjacent convolutional layers, further simplifying the network.

B. CSPPartialStage

1) CSP structure with PHDC Block: Deep convolutional neural networks often involve dense convolution operations as the channel numbers expand, which can exponentially increase the computational cost of the model. This, coupled with a single feature propagation path, can cause repeated usage of gradient information, resulting in redundancy and inefficient network training. To address this issue, CSPNet [34] separates the gradient flow to propagate through different network paths, ensuring that the gradient information obtained has greater correlation differences. Both YOLOv5 [12] and YOLOX [13] utilize a CSPNet-like structure to reduce computational burden without compromising accuracy. In our research, we implemented this method to design the main module.

Fig. 3 illustrates the structure of CSPPartialStage, where CBN refers to the concatenation of convolutional layer, BatchNorm layer, and nonlinear activation layer. Our CSPPartialStage incorporates the PHDC block mentioned earlier as a crucial module into consecutive CSPNet-based feature extraction layers. After the input feature map is processed by the first CBN, the number of its channels is halved, and then it is routed into two parallel branching structures. One of the branches executes only simple CBN operations, while the other goes through a single CBN before being sent to the feature extraction module made up of N PHDC Blocks arranged in series to perform deeper feature extraction. The outputs of both branches are concatenated in the channel dimension and given coordinate attention through the CA attention module. Finally, CBN is used for channel matching to obtain the correct number of channels. It is worth noting that all the CBNs in CSPPartialStage use 1 × 1 convolution, merely changing the number of feature map channels or performing simple feature mapping, without introducing a noticeable increase in computational burden.

2) Coordinate Attention: The general attention mechanism, such as Squeeze-and-Excitation, accounts for the correlation between channels and recalibrates the channel information for effective aggregation, leading to a better model representation. While this approach proves useful for detecting natural images, small object detection in remote sensing images with complex spatial distributions requires more prominent focus on the target's localization features. As such, we utilize the Channel Attention (CA) module [29] to augment the model's ability to extract location-based features. A schematic of the CA module's workflow is illustrated in Fig. 9.

The input feature map is initially encoded for each channel independently by performing global average pooling separately in both the horizontal and vertical directions. Specifically, for the cth channel, the output in the vertical and horizontal directions of dimensions h and w is denoted by the following equation:

$$z_c^h(h) = \frac{1}{w} \sum_{0<i\le w} x_c(h, i) \tag{2}$$

$$z_c^w(w) = \frac{1}{h} \sum_{0<j\le h} x_c(j, w) \tag{3}$$

Following the transformations in the two spatial directions as stated earlier, a pair of direction-sensitive feature maps is

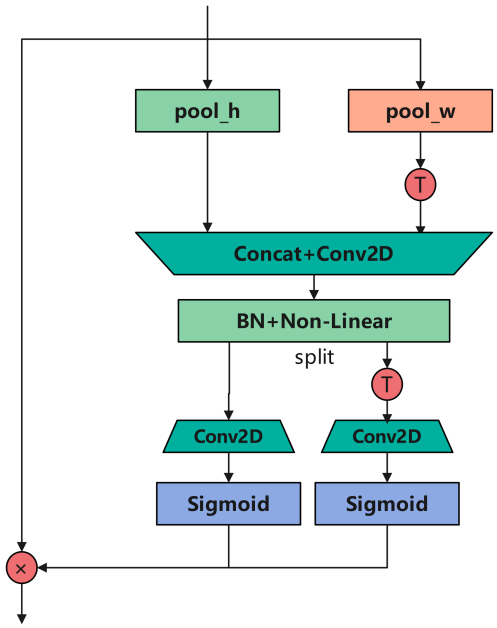


Fig. 9. Workflow of coordinate attention module.

generated. The resulting aggregated feature map is then concatenated and fed into a 1×1 convolution for the purpose of channel reduction.

One direction is transposed and appended to the encoding vector of the other to create a lengthy vector with a spatial dimension of $H + W$. The resulting vector is then subject to a shared 1×1 convolution transform, detailed as follows:

$$f = \delta(F_1\left[z^h, z^w\right]) \tag{4}$$

In the above equation, [., .] denotes the concatenation operation, δ represents the batch normalization layer and the nonlinear activation function. F_1 represents a 1×1 convolution used for channel reduction. f is an intermediate feature map with dimensions $R^{C/r\times(H+W)}$, where r indicates the channel reduction ratio. Subsequently, f is spatially split into two feature maps, $f_h \in R^{C/r\times H}$ and $f_w \in R^{C/r\times W}$. These feature maps are used to recover the number of channels through a 1×1 convolution and then normalized using the Sigmoid function to create attention maps in both spatial directions as follows:

$$g^h = \sigma(F_h(f^h)) \tag{5}$$

$$g^w = \sigma(F_w(f^w)) \tag{6}$$

Where σ represents the Sigmoid function. Finally, multiply the input feature map with the obtained attention and get the final output of the CA module as follows:

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{7}$$

The design process of the channel attention (CA) module avoids standard three-by-three convolutions and reduces computational complexity through channel dimension reduction, making the CA module highly suitable for integration into lightweight networks. In the experiments, the accuracy of the model was improved with minimal changes to inference latency. Details of these experiments are discussed in the following section.

C. CSPPartialNet and CSPPartialFPN

In a single-stage object detection model, the Backbone Network is composed of multiple stages, each of which produces an output feature map with a different resolution. The distribution ratio of the computation module in each stage is often determined through heuristics. In CSPDarknet [10], the distribution ratio of the computation module is 1:3:3:1, while the distribution ratio of the computation module in the backbone network based on CSPRepResNet in PPYOLOE-R [15] is 1:2:2:1. Inspired by the performance of recent Swin-T [35] models in the field of vision, the authors of ConvNeXt [20] suggest a distribution ratio of 1:1:3:1. Following this suggestion, the number of PHDC blocks in the CSPPartialStage is set to [1, 1, 3,1].

For the neck of model, we also use modules similar to the CSPPartialStage, with the difference being the absence of the CA module. The highest level feature map is processed using SPP [36] to achieve feature map-level fusion between local and global features. The workflow of the FPNStage in the neck part can be seen in Fig. 4.

D. Rotation Detection Head

Remote sensing images frequently employ overhead viewing angles (e.g., satellite or airborne imagery), resulting in diverse angle distributions of targets, including vehicles, ships, airplanes, and other modes of transportation. Consequently, we implement the detection head from PPYOLOE-R, utilizing three independent branches for predicting the targets' position, direction, and category.The loss function incorporates Varifocal Loss, ProbIoU Loss, and Distributed Focal Loss to calculate the losses for target classification, bounding box localization, and angle estimation, respectively.The overall loss is calculated by weighting and summing the aforementioned losses with weights of 1.0, 2.5, and 0.05, respectively.These settings align with the PPYOLOE-R model. Fig. 5 shows the details of the rotation detection head.

IV. Experiments and Results

A. Experiments Settings

Our study involves two stages: Training and validation. We utilized PaddlePaddle 2.4 deep learning framework to train on the Intel(R) Xeon(R) Gold 5218 CPU, NVIDIA Tesla V100, and Debian10 stable platforms during the training phase. The SGD optimizer was selected, with the momentum set at 0.9 and batch size at 6 for the training process, while a cosine learning rate decay strategy was employed. The learning rate was initially set at 0.006, and the total number of epochs trained was 300. While ensuring training convergence, we selected the weight results from the best performance on the test set among 300 epochs as the final weights. Based on our experimental observations, all models achieve convergence within 300 training epochs. For the first 10 epochs of training, linear warm-up was applied. During the entire training process, randomized image rotation was used by using four angles 0°, 90°, 180°, and 270°, together with a 50% probability of random rotations at 30°and 60°.

During the validation phase, we assessed the trained model's performance on AMD 5800H, NVIDIA RTX3070 laptop, and

TABLE I
NUMBER OF INSTANCES FOR EACH CATEGORY IN THE TRAINING SET AND TEST SET

Category	Training Set		Test Set	
	DOTA	SODA-A	DOTA	SODA-A
Plane	14989	22817	4658	20586
Small-vehicle	48889	366461	10689	340321
Large-vehicle	34959	14818	8971	8082
Ship	58738	38893	18867	41623
Tota	157575	442989	43179	410612

Windows 10 platforms based on two primary evaluation criteria: mAP on the test set and model forward inference latency.

B. Dataset

To evaluate the effectiveness of our proposed model, we performed experiments on the DOTA [37] dataset and SODA-A [38] dataset showcasing images primarily sourced from Google Earth. The majority of the imagery in the datasets exhibits a spatial resolution under 0.5 m and consists of fifteen(DOTA) and nine(SODA-A)categories of objects annotated using rotated rectangular boxes. For the data preprocessing phase, we selected four typical targets of remote sensing imagery, namely, planes, large vehicles, small vehicles, and ships. The remaining categories were removed. Afterward, the images were cropped into 1024×1024 pixels, with an overlap of 200 pixels being maintained to ensure continuity and consistency between different cropped images. The resulting datasets comprise 6049(DOTA) and 8811(SODA-A) images in the training set and 1718(DOTA) and 5268(SODA-A) images in the testing set. Table I presents a statistical summary of the number of annotated images for each classification in both datasets.

According to the standards of the MS COCO dataset, targets with pixel numbers that are smaller than 1024 pixels as "small," targets with pixel numbers falling between 1024 and 9216 are noted as "medium," while targets with pixel numbers larger than 9216 are designated as "large." By counting the number of pixels annotated within the bounding boxes, it is observed that "small" objects account for 64.7%(DOTA) and 94.5%(SODA-A) of the total number of objects in our datasets. This feature is a vital characteristic that sets remote sensing images apart from natural images.

C. Experimental Evaluation Metrics

To evaluate the model accuracy, we adopted the evaluation methodology of PASCAL VOC [39]. Initially, we computed the precision and recall by using the below representations, where TP corresponds to true positive, FP references false positive, and FN refers to false negative:

$$\text{precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{9}$$

During the computation process, the N predicted boxes are sorted in descending order according to their confidence scores. Subsequently, by progressively adding them, N sets of precision-recall pairs are obtained. Using this approach, the precision-recall curve can be plotted on a 2-D coordinate system. Average precision (AP) is defined as the area enclosed between the precision-recall curve and the x-axis, and is defined by the following formula:

$$\text{AP} = \int_0^1 P(R)dR. \tag{10}$$

For multiclass objectives, mAP is frequently used as an evaluation metric. mAP is the average of each class's AP and is defined as $mAP = \frac{1}{N}\sum_{n=1}^{N} \text{AP}_n$. mAP serves as our primary metric for evaluating the precision of the model.

In addition to prioritizing the accuracy of the model, equal consideration is given to its cost. To determine the number of learnable parameters in the model, we employ M Params, while G FLOPs are used to calculate the total number of multiply and accumulate operations performed during the model's inference process. However, these two metrics frequently only depict the theoretical inference speed of the model. As we are more concerned with the model's overall performance on actual hardware, we also regard inference latency (ms) as a crucial assessment metric. Inference latency is computed by averaging the propagation time of the model in 1000 forward passes, which diminishes the impact of stochastic errors on the experiment.

D. Ablation Study

We used the PPYOLOE-R-S model as the baseline model to conduct ablation experiments on the DOTA dataset, examining the effectiveness of the PHDC and CA modules. The outcomes of the experiment are shown in Table II. It is worth noting that in addition to prioritizing model accuracy, we emphasize its efficiency.

In Experiment 2, we investigated the performance of a model built using the Partial Hybrid Dilated Convolution (PHDC) module, which omitted the Channel Attention (CA) Block. Our aim was to evaluate the efficiency of the PHDC module in terms of inference speed, computational complexity, and model accuracy compared to the baseline model. Our results show that the PHDC module performed well in terms of inference speed. Specifically, it reduced the inference latency by approximately 32.2%, which is a notable improvement over the baseline model. Furthermore, the PHDC module achieved a 0.3% mean average precision (mAP) increase, indicating that it has potential to improve the model efficiency without compromising its accuracy. In addition to its promising results in both inference speed and accuracy, the PHDC module also demonstrated a reduction in the number of parameters and computational complexity. Specifically, it decreased the number of parameters by 17.8% and the computational complexity by 27.0%. These findings suggest that the PHDC module is not only efficient, but also requires fewer resources to achieve the same level of performance as the baseline model.

In Experiment 3, we introduced the Channel Attention (CA) Block to coordinate the network without hybrid dilated convolution in the PHDC Block. The purpose of this experiment was

TABLE II
EXPERIMENTAL RESULTS OF ABLATION OF THE ALGORITHM MODULE

Number	Baseline	HDC 125	CA Block	mAP	Params(M)	FLOPs(G)	Latency(ms)
1	✓			87.85%	11.8	21.8	31
2	✓	✓		88.15%	9.7	**15.9**	**21**
3	✓		✓	88.74%	9.6	16.2	23
4(ours)	✓	✓	✓	**89.75%**	**9.6**	16.2	23

The bold values are the best in the current column.

TABLE III
EXPERIMENTAL RESULTS OF DIFFERENT NUMBERS OF BLOCKS IN EACH STAGE

numbers	mAP	Params(M)	FLOPs(G)	Latency(ms)
1,2,2,1	89.23%	9.5	16.11	12
3,1,1,1	88.88%	9.5	16.13	13
1,1,3,1	89.75%	9.6	16.11	11

to investigate the impact of the CA Block on the model's performance and efficiency. In Experiment 4, we utilized both hybrid dilated convolution and the CA Block to evaluate their combined effect on model accuracy and complexity. Our results show that Experiment 4 outperformed Experiment 2 and Experiment 3 in terms of accuracy. Specifically, Experiment 4 achieved a 1.06% increase in mean Average Precision (mAP) while maintaining comparable parameter (Params) and computational (FLOPs) complexity. This improvement in accuracy is attributed to the larger reception field generated by hybrid dilated convolution, which better captures small remote sensing targets. Although the CA block slightly contributed to the increase in inference latency, the improvement in accuracy justifies its use in the model.

Fig. 10 displays partial results from the DOTA dataset of our CSPPartial-YOLO and PPYOLOE-R models. As observed, our model exhibits fewer missed and false detections compared to the baseline model in scenarios such as detecting ships near or far away the shore, detecting dense vehicles, and distinguishing between confusing object classes. This improvement can be attributed to the use of hybrid dilated convolution and the improved visual attention module, which allow for a larger receptive field and richer semantic information. Specifically, hybrid dilated convolution expands the field of view of each convolutional layer, allowing the network to capture more contextual information and improve object recognition accuracy. The coordinate attention module enhances the network's ability to focus on relevant features by adaptively weighting feature maps. Our proposed models achieved a 1.9% increase in accuracy compared to the baseline model, while also reducing inference latency by 25.8%. These results demonstrate the efficacy of our proposed models in improving object detection performance while maintaining a reasonable inference speed.

Table III presents the experimental results of allocating different ratios of PHDC blocks in CSPPartialStage. The results suggest that employing the suggested [1:1:3:1] ratios in ConvNext [20] yields higher validation accuracy and inference speed with comparable parameter and computational costs.

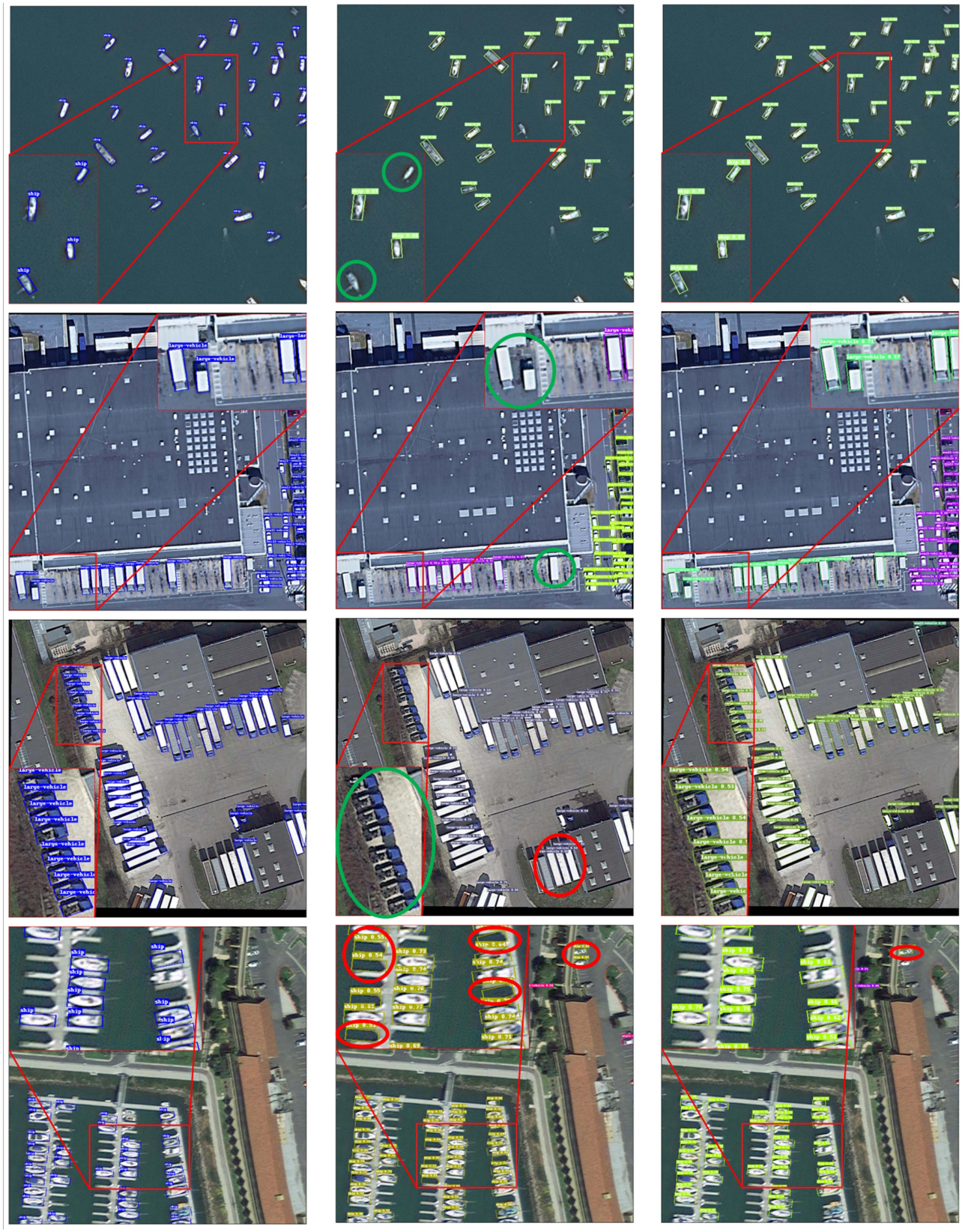
E. Comparison With Other YOLO Models

To ascertain the competitiveness of the model introduced in this article in both speed and accuracy, we conducted a comparison with the modern YOLO series models of the identical model size. The resulting comparison on the DOTA dataset and SODA-A dataset is illustrated in Table IV.

In object detection benchmarks, it is vital to ensure that the models are evaluated on a level playing field. To this end, we employed the same rotation detection head, namely the PPYOLOE-R detection head, across all models for predicting bounding boxes with rotation angles. The results of our experiments indicate that while the YOLOX model achieved the highest mAP score in both DOTA dataset and SODA-A dataset among all models, it also incurred the highest inference delay due to suboptimal inference speed optimization in the CSPDarkNet backbone network used by YOLOX. On the other hand, PPYOLOE-R utilizes reparameterization techniques to enhance the model's inference speed, leading to a better tradeoff between accuracy and speed. In contrast to YOLOX and PPYOLOE-R, YOLOV8 replaces the C3 module of the CSPDarkNet with the C2F module, which maximizes gradient flow information while maintaining a lightweight architecture. This design choice enables YOLOV8 to achieve high accuracy while also maintaining reasonable inference speed. The RTMDet model utilizes a design approach comparable to ours, whereby the receptive field is expanded via an increase in the kernel size of convolution layers, thereby improving the model's capability for feature extraction. However, RTMDet differs in that it directly enlarges the convolution kernel size and employs depthwise separable convolution to achieve a balance between efficiency and effectiveness.

In our proposed CSPPartial-YOLO, we also aimed to achieve a more efficient and lightweight network by utilizing partial convolution in constructing the network. Furthermore, our approach combines typical characteristics of targets in remote sensing images, such as small object sizes and complex backgrounds, with advanced techniques in computer vision. Specifically, we employ hybrid dilated convolutions to establish long-distance dependency relationships, broaden the receptive field, and implement coordinate attention modules to enhance the position information. These design choices result in improved representation ability for small targets in remote sensing images, which are often challenging to detect.

Our model incurred only a 0.19%(DOTA) and 2.21%(SODA-A) loss in mAP score when compared to YOLOX, yet registered declines of 22.0%, 42.6%, and 32.3% in terms of parameter count, computational cost, and inference delay, respectively. Our model also showed a 14.8% speed advantage over the fastest YOLOV8 model, requiring only 67.1% of the parameters and 74.3% of the computational cost, while maintaining some advantages in precision. Overall, taking into consideration both speed and precision, our model maintains competitiveness with current state-of-the-art YOLO models in typical target detection in remote sensing.



(a) (b) (c)

Fig. 10. Prediction results of the baseline model and our CSPPartial-YOLO in some scenes of the DOTA dataset. The green circle indicates false detection, and the red circle indicates missed detection. (a) Ground truth. (b) PPYOLOE-R. (c) Ours.

TABLE IV
EXPERIMENTAL RESULTS OF COMPARISON WITH MODERN YOLO SERIES MODELS

Model	mAP		Params(M)	FLOPs(G)	Latency(ms)
	DOTA	SODA-A			
PPYOLOE-R [15]	87.85%	75.36%	11.8	21.8	31
YOLOX(R) [13]	**89.94%**	**79.99%**	12.3	28.2	34
YOLOV8(R) [40]	88.77%	73.38%	14.3	21.8	27
RTMDet(R) [19]	89.01%	73.40%	13.8	29.1	33
CSPPartial-YOLO(Ours)	89.75%	77.78%	**9.6**	**16.2**	**23**

The bold values are the best in the current column.

TABLE V
EXPERIMENTAL RESULTS OF COMPARATION WITH ADVANCED LIGHTWEIGHT BACKBONE NETWORKS

Model	mAP		Params(M)	FLOPs(G)	Latency(ms)
	DOTA	SODA-A			
MobileNetV3 [41]	87.46%	71.82%	2.7	**4.2**	19
ShuffleNetV2 [23]	88.36%	76.96%	3.2	9.5	26
GhostNet [25]	88.10%	68.56%	10.0	10.9	30
CSPPartialNet(Ours)	**89.75%**	**77.78%**	**2.0**	7.8	**11**

The bold values are the best in the current column.

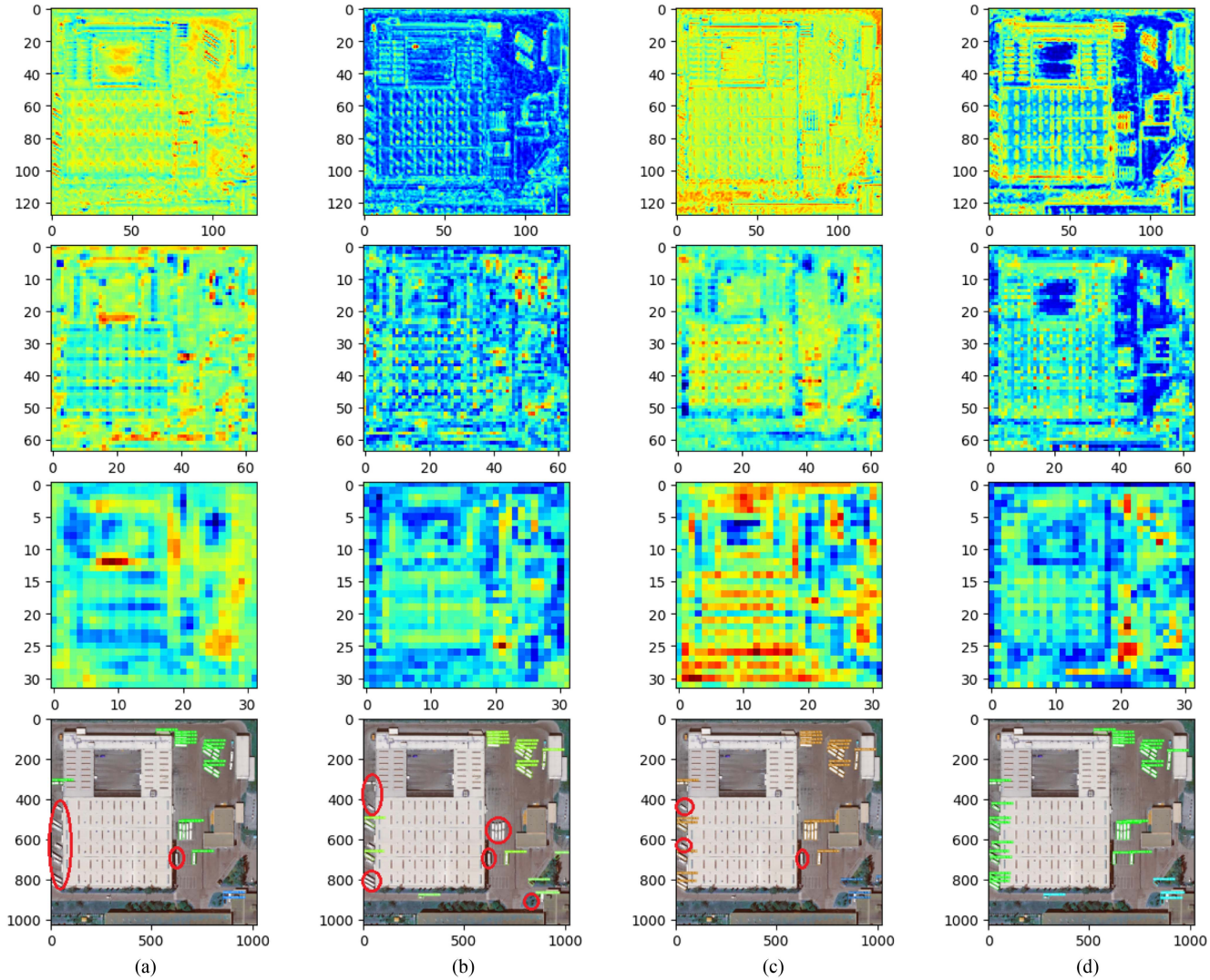


Fig. 11. First three rows show the output feature maps of the last three stages of the backbone network, with sizes of 128 × 128, 64 × 64, and 32 × 32, respectively. The last row displays the prediction results of the model, with the missed detection highlighted by a red circle. (a) MobileNetV3. (b) ShuffleNetV2. (c) GhostNet. (d) Ours.

F. Comparison With Lightweight Backbones

To validate the performance of the proposed backbone network, namely CSPPartialNet, we conducted comparative experiments against several lightweight backbone networks. Table V showcases the experimental results obtained for the DOTA and SODA-A dataset.

In this set of experiments, we solely evaluated the performance of the backbone network by using the same Neck and Head, and solely replacing the backbone network. As such, the metrics "Params," "FLOPs," and "Latency" in the table were exclusively calculated for the backbone network section.

MobileNetV3 is the latest addition to the MobileNet series of networks. It mainly employs depthwise separable convolutions to decrease the number of trainable parameters and computational complexity of the network. In addition, it integrates SE channel attention, and utilizes neural architecture search (NAS) techniques to obtain the optimal parameters. Nevertheless, it is noteworthy that despite having the lowest FLOPs in this set of experiments, MobileNetV3 still has considerable latency. This is because its architecture is more suited for CPU device computation, while GPU computation is more crucial in the experimental environment. Thus, it indicates that FLOPs cannot accurately reflect the model's inference time and one should focus more on latency. Moreover, MobileNetV3 has been observed to exhibit lower sensitivity to small targets and less attention to intricate details, which may lead to reduced effectiveness in typical targets of remote sensing images. Hence, when employing MobileNetV3 for remote sensing image classification, one should exercise caution.

ShuffleNetV2 is a lightweight backbone network that considers the impact of memory access count (MAC) on inference latency. Nonetheless, similar to MobileNetV3, it is better suited for CPUs on mobile devices than for GPUs. Moreover, ShuffleNetV2 exhibits lower capability to concentrate on long-range dependency information, making it difficult to derive essential information from images containing large objects with significant aspect ratios. Consequently, this may yield inadequate results in intricate and uncertain remote sensing object detection scenarios.

GhostNet is among the very first models that concentrate on redundant feature maps in convolutional neural networks. By replacing conventional convolutions with a cheap operation, it facilitates the acquisition of feature maps. Nonetheless, the application of depthwise convolution in the cheap operation can raise the MAC, which negates the previously reduced FLOPs. Consequently, despite having fewer FLOPs, GhostNet still demonstrates high latency.

Fig. 11 indicates the output feature maps of the final three stages and the prediction results for each lightweight backbone network. Our CSPPartialNet provides a clearer representation of the target position at all three scales, especially in the 32×32 feature map output. Our model accurately captures the information about the vehicle parking position and displays it with higher values in the heat map due to our use of the channel attention module (CA), which enhances the target position feature. Compared in the prediction results, our model has the least missed detections.

Based on experimental results, CSPPartialNet achieved the highest mAP compared to the aforementioned three lightweight backbone networks. It also achieved the best inference time on real hardware, lower by 42.1%, 57.7%, and 63.3% than MobileNetV3, ShuffleNetV2, and GhostNet, respectively. In addition, CSPPartialNet has the lowest parameter amount, which makes the model suitable for devices with low storage.

V. Discussion and Conclusion

The detection of objects in remote sensing images has been challenging due to the limited computing and storage resources of remote sensing platforms. Current object detectors struggle to achieve fast and accurate predictions. In this article, we improved the baseline model to achieve a better balance between speed and accuracy, and we refer to the improved model as CSPPartial-YOLO. The new model is specifically designed for the detection of typical targets in remote sensing images.

To improve the model's inference speed and reduce parameters and calculations, we utilized redundant feature maps in the model inference process and introduced the PHDC module, which is a combination of partial convolution with hybrid dilated convolution, with specific dilation rates. Furthermore, we incorporated the CA module to increase the model's sensitivity to target location information considering the multidirectionality, dense distribution, and small size of typical targets in remote sensing images. Finally, we designed the CSPPartialStage to explore the appropriate computational depth ratio for the backbone network, constructed the backbone, and the Neck network.

In this article, we conducted ablative experiments to demonstrate the advantages of the proposed model compared to the baseline model. Furthermore, we evaluated the effectiveness of the main improvement methods through comparative experiments with state-of-the-art YOLO series models and lightweight backbone networks. The proposed model and methods achieved competitive advantages in terms of both accuracy and speed. Our experiments show that the lightweight detector introduced in this article has potential for real-time detection of typical targets in remote sensing images. Our future research endeavors will involve exploration of advanced lightweight network design methods, like neural network pruning and neural architecture search (NAS), in order to further decrease model redundancy and enhance detection efficiency.

References

[1] P. Patil, "Applications of deep learning in traffic management: A review," *Int. J. Bus. Intell. Big Data Analytics*, vol. 5, no. 1, pp. 16–23, 2022.
[2] S. Wang, Y. Han, J. Chen, Z. Zhang, G. Wang, and N. Du, "A deep-learning-based sea search and rescue algorithm by UAV remote sensing," in *Proc. IEEE CSAA Guid., Navigation Control Conf.*, 2018, pp. 1–5.
[3] Y. Xu, M. Zhu, P. Xin, S. Li, M. Qi, and S. Ma, "Rapid airplane detection in remote sensing images based on multilayer feature fusion in fully convolutional neural networks," *Sensors*, vol. 18, no. 7, 2018, Art. no. 2335.
[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, vol. 28.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
[8] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[9] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.
[10] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[11] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[12] G. Jocher, "YOLOv5 by Ultralytics," May 2020. [Online]. Available: https://github.com/ultralytics/yolov5
[13] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.
[14] S. Xu et al., "PP-YOLOE: An evolved version of YOLO," 2022, *arXiv:2203.16250*.
[15] X. Wang, G. Wang, Q. Dang, Y. Liu, X. Hu, and D. Yu, "PP-YOLOE-R: An efficient anchor-free rotated object detector," 2022, *arXiv:2211.02386*.
[16] Y. Guo, S. Chen, R. Zhan, W. Wang, and J. Zhang, "LMSD-YOLO: A lightweight YOLO algorithm for multi-scale sar ship detection," *Remote Sens.*, vol. 14, no. 19, 2022, Art. no. 4801.
[17] M. Cui et al., "LC-YOLO: A lightweight model with efficient utilization of limited detail features for small object detection," *Appl. Sci.*, vol. 13, no. 5, 2023, Art. no. 3174.
[18] H. Zhang et al., "An improved lightweight yolo-fastest V2 for engineering vehicle recognition fusing location enhancement and adaptive label assignment," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 2450–2461, 2023.
[19] C. Lyu et al., "RTMDet: An empirical study of designing real-time object detectors," 2022, *arXiv:2212.07784*.
[20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11976–11986.
[21] J. Chen et al., "Run, don't walk: Chasing higher FLOPS for faster neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 12021–12031.
[22] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[23] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.
[24] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.
[25] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1580–1589.
[26] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
[27] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11534–11542.
[28] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
[29] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13713–13722.
[30] Q. Zhang et al., "Split to be slim: An overlooked redundancy in vanilla convolution," 2020, *arXiv:2006.12085*.
[31] P. Wang et al., "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1451–1460.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
[34] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 390–391.
[35] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
[36] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[37] G.-S. Xia et al., "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3974–3983.
[38] G. Cheng et al., "Towards large-scale small object detection: Survey and benchmarks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13467–13488, Nov. 2023.
[39] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, 2010.
[40] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
[41] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

Siyu Xie received the B.S. degree in electronic information science and technology from the College of Science of Beijing Forestry University, China, in 2021. He is currently working toward the M.S. degree in signal and information processing with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China.

His research interests include deep learning, lightweight model and remote sensing.

Mei Zhou was born in Sichuan, China in 1980. She received the Ph.D. degree in communication and information systems from the Graduate School of the Chinese Academy of Sciences, Beijing, China, in 2007.

She is currently an Associate Professor with Aerospace Information Research Institute, Chinese Academy of Sciences. Her main research direction is multidimensional imaging technology for active and passive sensors.

Chunle Wang was born in Jilin, China, in 1986. She received the B.S. degree in electronic information engineering from Beijing Information Science and Technology University, Beijing, China, in 2008. She received the Ph.D. degree in communication and information systems from University of Chinese Academy of Science, Beijing, China, in 2013.

She is currently an Associate Researcher with the Aerospace Information Research Institute, Chinese Academy of Sciences. Her research interests include spaceborne synthetic aperture radar (SAR) system design and SAR image processing.

Shisheng Huang received the B.S. degree in applied mathematics in 2006, and the Ph.D. degree in system analysis and integration both from National University of Defence Technology, Changsha, China, in 2012.

He is currently working in the field of spaceborne synthetic aperture radar designing and image processing with Beijing Institute of Tracking and Telecommunications Technology, Beijing, China.