Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 1272896, 15 pages
https://doi.org/10.1155/2022/1272896

Research Article

Yolo-Based Improvements in Remote Sensing Image Applications

Yiming Zhang and Xiang Li

School of Computer Science, China University of Geosciences, Wuhan 430079, Hubei, China

Correspondence should be addressed to Xiang Li; lixiang@cug.edu.cn

Received 26 August 2022; Revised 22 November 2022; Accepted 7 December 2022; Published 15 December 2022

Academic Editor: Paolo Spagnolo

The identification of some specific targets in remote sensing images is still quite challenging despite the adequate accuracy of deep learning-based target detection models. This work proposes a variant of YOLOv3 based on the residual structure as the backbone and the attention mechanism module, which improves the ability of YOLOv3 to extract features. SGE is a lightweight module that can fully extract features from images without bringing an increase in computation. Furthermore, the dilated encoder module used in YOLOF was introduced as a neck to enrich the perceptual field of the C5 feature layer by concatenating four layers of dilated convolution with different expansion coefficients. The C5 feature layer and the residual structure were further processed to contain sufficient scale information for further detection. In terms of the mean average precision (mAP), experimental results demonstrate that the proposed model outperforms the other models: YOLOv3, faster-RCNN-r50+GACL Net, and YOLOv4.

1. Introduction

Remote sensing target detection [1] has got greater scientific attention because of its practical applications in the past decades. Target detection is aimed at identifying and localizing specific targets, such as aircraft, car parks, vehicles, or other objects, on remote sensing images (RSI) [2]. Traditional machine learning models [3] can identify objects in a controlled environment with simplified structures. However, their performance declines in the case of complex scenes and multiscale targets [4], particularly small targets [5]. For example, Haar-AdaBoost [6], deformable part model (DPM) [7], gradient histogram feature-support vector machines (HOG-SVM) [8], and other methods perform quite well for general datasets, but their performance is unsatisfactory for complex RSI with large differences in various targets [9].

Inspired by the recent advances in deep learning and computer hardware, many studies have actively explored deep networks [10] for detecting targets in remotely sensed images. The RCNN [11] was first proposed for detecting targets, with major advances in subsequent variants of the RCNN. Redmon proposed YOLOv3 [12], where the feature extraction network was replaced with DarkNet53 compared to YOLOv2 [13], and the FPN idea was adopted. Also, YOLT [14] was proposed for multiscale target recognition in RSI. Recently, Lu proposed GACL-Net [15] in an attempt to improve the target localization performance. Zhu proposed a brain tumour segmentation method based on the fusion of depth semantics and edge information in multimodal MRI to achieve a more adequate use of multimodal information for accurate segmentation [16]. The deep learning-based target detection models have shown outstanding performance in comparison to the traditional machine learning-based algorithms, particularly for multiscale and small targets.

Owing to their ability to extract spatial contextual information [17], CNNs have predominantly been employed as the models in RSI target detection. The RSI target detection models have mainly been categorized into region proposal methods [18] and anchor regression-based methods [19]. The region proposal methods use the region proposal subnetwork to determine the initial locations of potential objects. After that, the classification subnetwork predicts the class and bounding box [20] of the objects. Typical region proposal-based models include variants of RCNN, fast RCNN [21], and faster RCNN [22]. Although these region proposal-based models can detect targets

accurately, the computational complexity of these models is rather high for utilization in practical applications, especially in high-resolution and large-scale RSI. In contrast, models based on anchor regression can provide equivalent detection accuracy without losing detection speed [23]. The anchor regression-based model, unlike the region proposal-based model, is constructed on a one-stage detection [24] network that treats the entire prediction process as a regression process. Examples of typical models are SSD [25], YOLO [26], and RetinaNet [27].

The variants of YOLO have particularly achieved a great improvement since its development. YOLOv3, YOLOv4 [28], and YOLOv5 have achieved both real-time processing and accurate detection capabilities. However, the characteristics of natural images and RSI are significantly different due to different photographic mechanisms and various capturing angles. Consequently, the models developed for natural images cannot perform satisfactorily when applied to RSI, primarily due to the presence of multiple scales and low resolution.

In this work, using YOLOv3 as the baseline, the residual module in the SGE attention module [29] and the dilated encoder module in YOLOF [30] have been employed. The adopted SGE module can significantly enhance the feature learning of semantic regions [31] by improving the spatial distribution of different semantic subfeatures [32] in the group, thereby producing a large statistical variance. The C5 feature map [33] is simultaneously expanded by concatenating four layers of dilated convolution with expansion factors of 2, 4, 6, and 8. The residual module then fuses the multiscale features into a single layer, thereby enabling multiple receivers to cover as many scales of objects as possible. Since no studies have been conducted to try this, we introduced the SGE module and the dilated encoder module into YOLOv3 to verify if our idea is correct.

The remainder of this paper has been structured as follows: Section 2 discusses the structure of YOLOv3. Section 3 presents the proposed YOLOv3-DE model, whereas Section 4 analyzes the performance of the YOLOv3-DE model through a set of experiments. The entire work has been concluded in Section 5.

1.1. YOLOv3. Since the introduction of YOLO in 2015, a number of its variants have been developed. Furthermore, the official version has been upgraded to Scaled-YOLOv4 [34]. The variants of the YOLO model are based on the regression of each grid bounding box, generating a set of three predictions, namely, (1) the probability of object presence, (2) the coordinate position of the bounding box, and (3) the probability of individual classes. The loss function used in YOLOv3 has been defined as follows:

$$\begin{aligned}\text{Loss} = {} & \lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\left(x_i-\widehat{x}_i^j\right)^2+\left(y_i-\widehat{y}_i^j\right)^2\right]+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\left(\sqrt{\omega_i^j}-\sqrt{\widehat{\omega}_i^j}\right)^2+\left(\sqrt{h_i^j}-\sqrt{\widehat{h}_i^j}\right)^2\right]\\ & -\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\widehat{C}_i^j\log\left(C_i^j\right)+\left(1-\widehat{C}_i^j\right)\log\left(1-C_i^j\right)\right]-\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{noobj}\widehat{C}_i^j\log\left(C_i^j\right)+\left(1-\widehat{C}_i^j\right)\log\left(1-C_i^j\right)\\ & -\sum_{i=0}^{S^2} I_i^{obj}\sum_{c\in classes}\left[\widehat{P}_i^j\log\left(P_i^j\right)+\left(1-\widehat{P}_i^j\right)\log\left(1-P_i^j\right)\right],\end{aligned} \tag{1}$$

where λ_{coord} is a coordination coefficient that coordinates the contribution of rectangular frames of varying sizes to the error function. When the size of the rectangular box is small, the coefficient is greater, which increases the rectangular box's contribution to the loss function. In contrast, when the rectangular box is large, the coefficient is reduced, which decreases the rectangular box's contribution to the loss function. $(i, j)*$ is not included in the equation, although it will be utilized afterwards. It represents the jth anchor box of the ith grid. I_{ij}^{obj} denotes whether (i, j)$*$ anchor is responsible for this object. It indicates that if the IOU [35] of the ground-truth box in the B anchor box of the i grid is greater than the established threshold, it is responsible for predicting the object because the shape and size are most compatible with the present object at this time $I_{ij}^{obj}=1$. Otherwise, $I_{ij}^{obj}=0$. S^2 and B represent the number of grids and anchors, respectively. (x_i, y_i) represents the center coordinates of the rectangular box predicted by the network, while $(\widehat{x}_i^j, \widehat{y}_i^j)$ represents the center coordinates of the rectangular box that has been marked. λ_{noobj} is a weight that balances the weight of its confidence error in the loss function when the anchor fails to predict the target. C_i^j represents the probability score of the target object in the prediction box. $\widehat{C}_i^j$ denotes the true value, which is determined by whether $(i, j)*$ anchor is responsible for predicting a specific anchor. Multiple binary classification challenges have been created for multiple class recognition. For each category, the true value of $\widehat{P}_i^j$ is 1 if the target belongs to that category, whereas it is 0 in all other circumstances. Therefore, the predicted value P_i^j represents the probability that the target exists in that particular category.

As the backbone of YOLOv3, Darknet53 downsamples the input image five times before fusing the features of the final three downsampled layers into the output feature layer. The structure of YOLOv3 is depicted in Figure 1. The 416×416 image is input into a series of downsampling layers, yielding precisely multiscale features of 13×13, 26×26, and 52×52 images. Several upsampling and

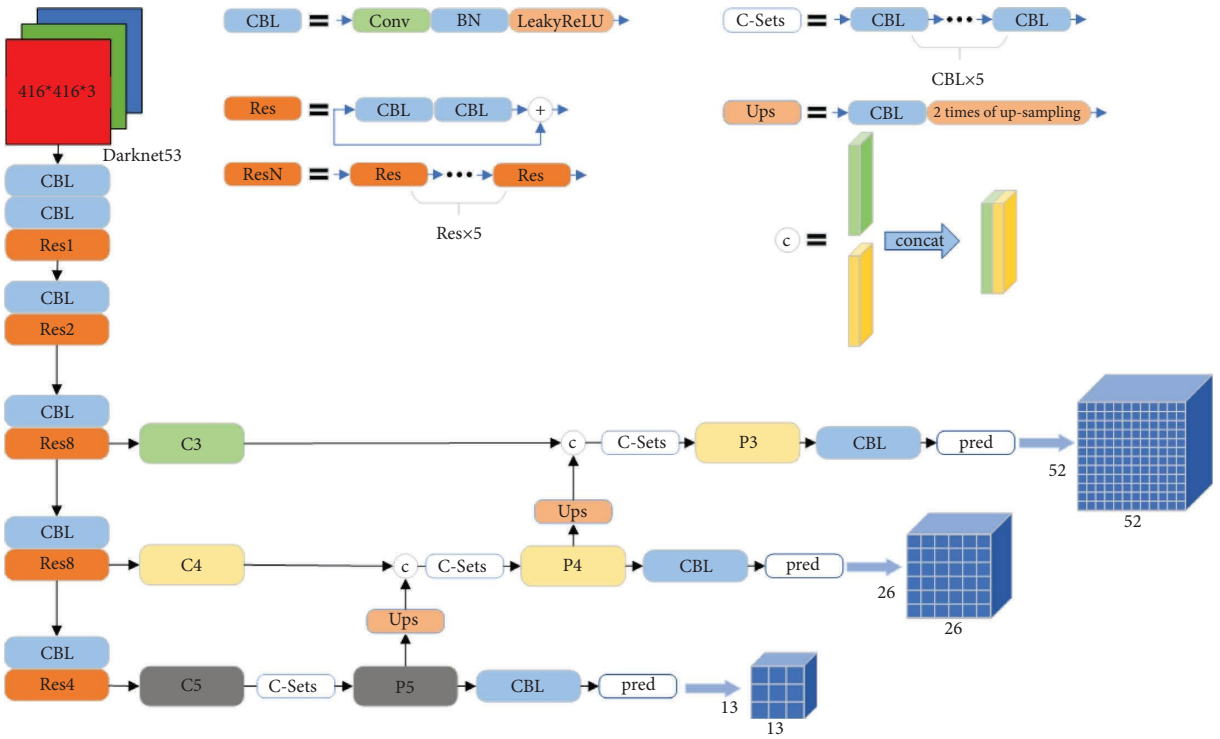


Figure 1: The structure of YOLOv3. BN represents the batch normalization.

convolutional layers are then used to combine these multiscale features again. In this structure, semantic and coordinate features from the deeper and shallower layers are combined to recognize targets effectively.

2. Method

The conventional YOLOv3 demonstrated limited target detection performance for RSI because objects in RSI are complex, have low resolution, and are multiscale. To address this limitation, YOLOv3-DE has been proposed to extract features without deepening the network structure. The structure of YOLOv3-DE is depicted schematically in Figure 2.

In comparison with YOLOv3, the proposed YOLOv3-DE model is based on the following optimizations: (1) The SGE attention mechanism is introduced into the residual structure, thus enabling the model to improve the efficiency of extracting features in more complex scenarios. (2) A neck is added at the C5 feature layer, which is fed into 1×1 convolution to decrease the number of channels from 2048 to 512. Subsequently, the semantic information is refined by a 3×3 convolution in the projector part. Also, four sets of residual blocks are used in series, wherein the convolution layers in each block use a null convolution, and the expansion coefficients of each set are 2, 4, 6, and 8, respectively. A residual connection is then utilized to combine the input features and the processed features in the residual blocks.

2.1. Attention Mechanism-Based Feature Extraction. YOLOv3-DE adopts Darknet53 as the backbone, comprising convolutional networks and residual structural units. The residual structure units enable Darknet53 to avoid the gradient vanishing problem even with a deep structure consisting of 53 layers. However, additional layers in the residual module stack can lower training speed, and a shortcut in a single residual module resulted in a perceptual field that only captured detailed information but not global features. Consequently, a deeper residual structure is ineffective in extracting features for complex scenes in RSI. As a result, we employed the SGE attention mechanism, which groups the feature maps to generate an attention mask by using similarity between global and local features and which leads to the spatial distribution of the improved semantic features of the spatial distribution. As a lightweight attention module, the SGE can improve overall detection accuracy with only a minor increase in computational complexity. As represented in Figure 3, the following steps describe the SGE flow structure:

(a) The input features are grouped and each group is subjected to a different attention operation, allowing each group to capture distinct semantics during learning.

(b) Each set of features is multiplied by the corresponding elements of its globally averaged pooled feature matrix and then normalized, so significantly enhancing the semantic learning of critical areas by utilizing the noise-free characteristics of the entire space. Normalization is intended to prevent intra-group disparities in the distribution of attention masks produced from different samples within the same group.

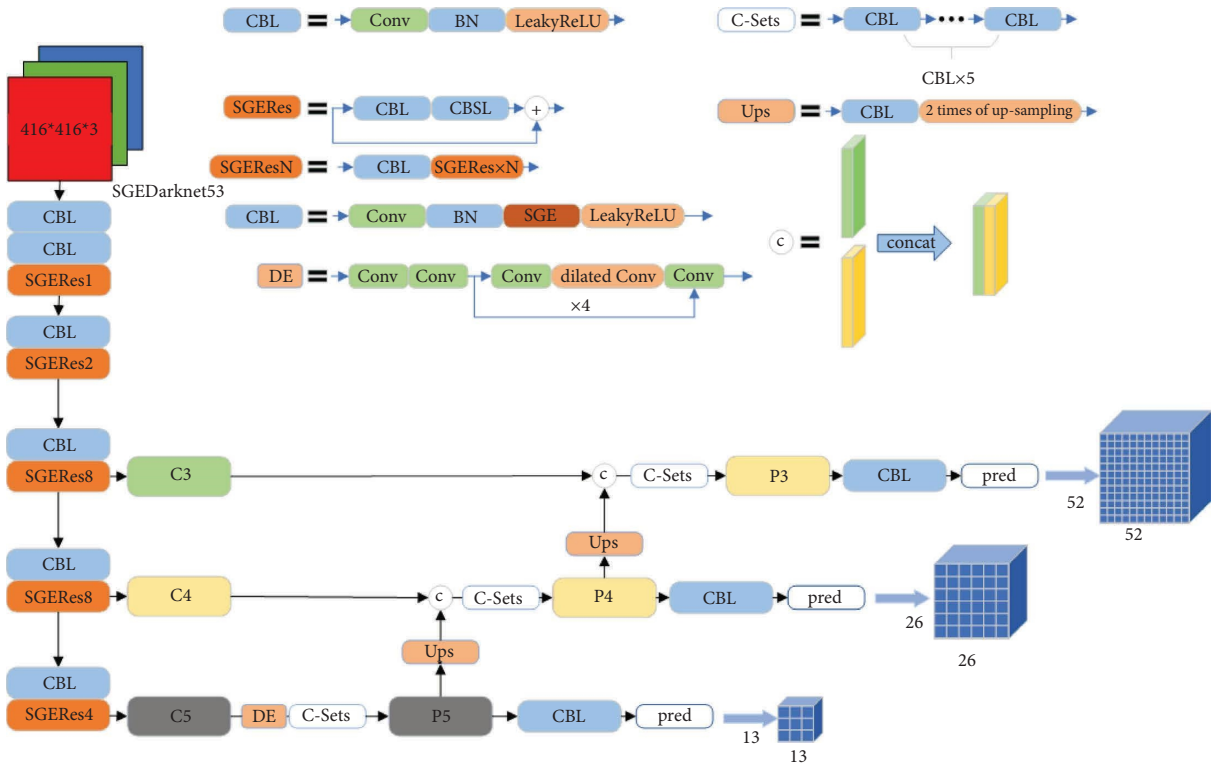


Figure 2: The structure of YOLOv3-DE. BN represents the batch normalization.

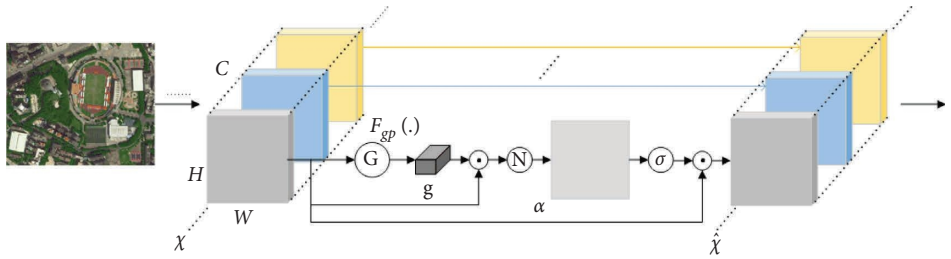


Figure 3: The structure of the SGE model.

(c) The final output features are obtained in accordance with the sigmoid activation function by multiplying them with the original feature map, hence permitting the mapping from the original feature to the added feature.

The SGE module is introduced to the residual structure unit, i.e., SGEResN, for extracting higher-order semantic features from complex scenes in RSI. Figure 4 illustrates that the SGEResN unit comprises convolution, batch normalization, activation, and SGE layers.

Table 1 displays the structure of the SGEDarknet53 backbone formed by the addition of the SGE.

2.2. Dilated Encoder. The last three features were combined by YOLOv3 using the feature pyramid network (FPN), which used a downsample-upsample structure to merge the specific features, as depicted in Figure 5.

In the FPN, the sizes of the last three feature maps are 52×52, 26×26, and 13×13, respectively. The receptive field of individual feature maps is decreased as their size increases. Accordingly, a deeper (smaller) feature map is suitable for larger targets. However, a smaller (13×13) feature map has a reduced resolution and thus loses essential details, which can affect the detection accuracy for larger targets. Therefore, we introduced the dilated encoder module utilized in YOLOF, in which the number of feature channels is reduced to 512 using 1×1 and 3×3 convolutions, and the features are extracted and stitched with four consecutive residual modules. In each residual module, the number of channels is first reduced to half of the original number using a 1×1 convolution. The number of sensory fields is subsequently increased using a 3×3 null convolution, and ultimately, the number of channels is increased by a factor of 2 using a 1×1 convolution. Then, before being fed into the residual structure, they are stitched with the feature map to obtain a $13 \times 13 \times 512$ feature map. Figure 6

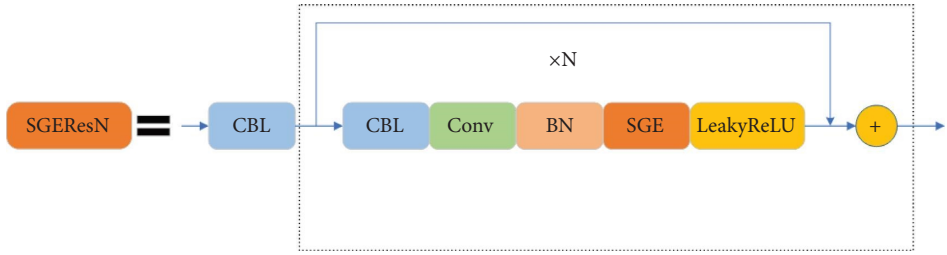


Figure 4: The SGResN unit.

Table 1: The detailed structure of SGEDarknet53 where K, S, and P represent the kernel size, stride, and padding, respectively.

Layers	Filters	Size	OutPut size
Convolutional	32	$K=3,\ S=1,\ P=1$	$416\times416\times32$
Convolutional	64	$K=3,\ S=2,\ P=1$	$208\times208\times64$
Convolutional	32	$K=1,\ S=1,\ P=0$	$208\times208\times32$
SGEResidual	64	$K=3,\ S=1,\ P=1$	$208\times208\times64$
Convolutional	128	$K=3,\ S=2,\ P=1$	$104\times104\times128$
2×Convolutional	64	$K=1,\ S=1,\ P=0$	$104\times104\times64$
2×SGEResidual	128	$K=3,\ S=1,\ P=1$	$104\times104\times128$
Convolutional	256	$K=3,\ S=2,\ P=1$	$52\times52\times256$
8×Convolutional	128	$K=1,\ S=1,\ P=0$	$52\times52\times128$
8×SGEResidual	256	$K=3,\ S=1,\ P=1$	$52\times52\times256$
Convolutional	512	$K=3,\ S=2,\ P=1$	$26\times26\times512$
8×Convolutional	256	$K=1,\ S=1,\ P=0$	$26\times26\times256$
8×SGEReSidual	512	$K=3,\ S=1,\ P=1$	$26\times26\times512$
Convolutional	1024	$K=3,\ S=2,\ P=1$	$13\times13\times1024$
4×Convolutional	512	$K=1,\ S=1,\ P=0$	$13\times13\times512$
4×SGEReSidual	1024	$K=3,\ S=1,\ P=1$	$13\times13\times1024$

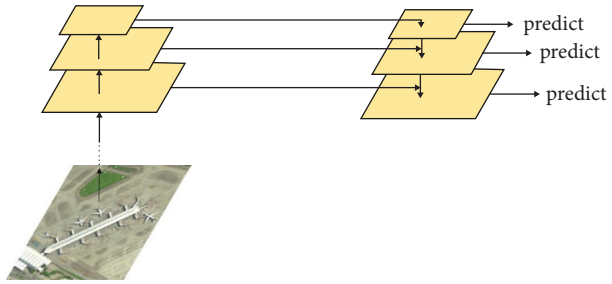


Figure 5: The diagram of the FPN.

depicts the FPN structure with the addition of the dilated encoder module.

2.3. K-Means for Anchor Boxes. The anchor is typically designed manually for the target detection algorithm based on the anchor. In SSD and faster RCNN, for instance, nine anchors with different sizes and aspect ratios are designed. However, manually designed anchors have a disadvantage and in that, their suitability for data sets cannot be guaranteed. The model's detection effect will be impacted if the sizes of anchors and targets differ significantly. Joseph Redmon, the author of YOLOv2, recommended employing K-means clustering [36] rather than manual design. K-means is a basic and widely used unsupervised learning algorithm that divides a data set into K clusters to increase data similarity within the same cluster while decreasing data similarity between clusters. Following is the standard K-means process:

(1) Initialize K cluster centers

(2) Using a similarity metric (often the Euclidean distance), allocate each sample to the closest cluster center

(3) Calculate the average of all samples in each cluster and update the cluster center

(4) Repeat steps 2 or 3 until the average cluster center is no longer altered or until the maximum number of iterations has been reached

By clustering the bounding box of the training set, a more suitable set of anchors for the data set can be

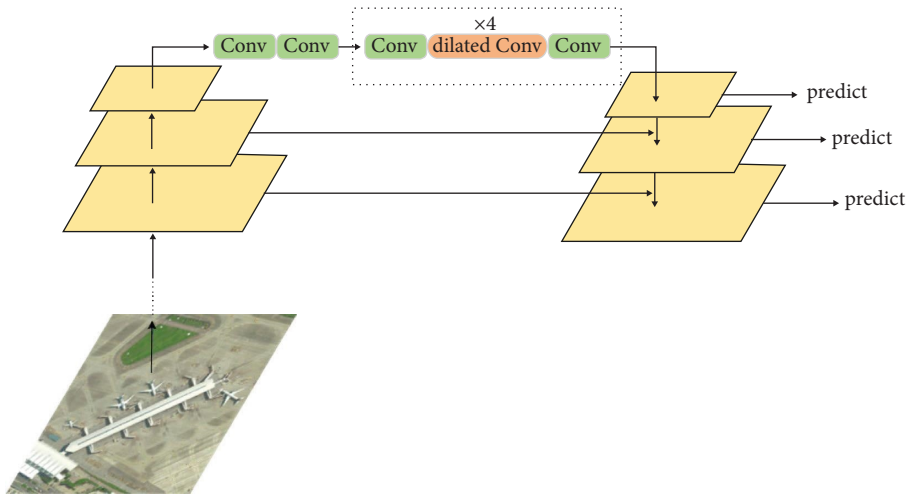


Figure 6: The dilated encoder FPN.

automatically generated, hence improving the network's detection performance. The original structure of YOLOv3 replaces manual design with K-means clustering. This is equally applicable to YOLOv4 and YOLOv5. This demonstrates that K-means is sufficient for our requirements. As a result, we decided to cluster the HRRSD dataset [37] using K-means. The anchor of the original YOLOv3 was determined for the daily-life image data, the COCO dataset [38], which does not apply to remote sensing of images. Thus, for RSI, the anchor was found as follows using K-means clustering on the HRRSD dataset.

(1) The anchors of k clusters were randomly initialized;

(2) The distance between the bounding box (bbox) and the anchor was computed based on the Intersection over Union (IoU) by (2) and (3), where anchor=(ω_a, h_a), box=(ω_b, h_b), and ω, h are the normalized width and height of the anchor bbox, respectively. IOU (box, anchor) represents the IOU value between bbox and anchor, and d (bbox, anchor) denotes the distance between bbox and anchor. IOU ($b$box, anchor) = intersection ($b$box, anchor)/union ($b$box, anchor) − intersection ($b$box, anchor)

$$\begin{aligned}\text{IOU}(b\text{box}, \text{anchor}) &= \frac{\text{intersection}(b\text{box}, \text{anchor})}{\text{union}(b\text{box}, \text{anchor}) - \text{intersection}(b\text{box}, \text{anchor})} \\ &= \frac{\min(\omega_a, \omega_b) \cdot \min(h_a, h_b)}{\omega_a h_a + \omega_b h_b - \min(\omega_a, \omega_b) \cdot \min(h_a, h_b)},\end{aligned} \tag{2}$$

$$d(b\text{box}, \text{anchor}) = 1 - \text{IOU}(\text{box}, \text{anchor}). \tag{3}$$

(3) Each bbox was reassigned to the closest cluster. The mean width and height of all bboxes in each cluster are subsequently updated as the anchor.

(4) Repeat steps 2 and 3 until no elements in the cluster change.

Nine anchors were determined via K-means clustering on the images of 416×416: (22.45, 22.33), (41.25, 48.62), (79.76, 65.71), (49.62, 139.14), (108.74, 109.52), (213.34, 74.59), (152.37, 150.04), (104.29, 235.73), and (222.97, 212.34). As depicted in Figure 7, the different sizes of the anchors permit matching with targets of diverse sizes within a single image.

2.4. Datasets and Evaluation Index. This section verifies the effectiveness of YOLOv3-DE in comparison to other target detection models on the RSI datasets, precisely the HRRSD dataset, and NWPU VHR-10 (NV10) datasets [39].

3. Datasets

The HRRSD dataset is the RSI target detection dataset, constructed in Pascal VOC [40] format, including the following targets: a parking lot, a crossroad, a basketball court, a tennis court, a ship, a ground track field, a baseball diamond, a storage tank, a bridge, an airplane, a harbor, a vehicle, and a T junction. Figure 8 shows the examples of the

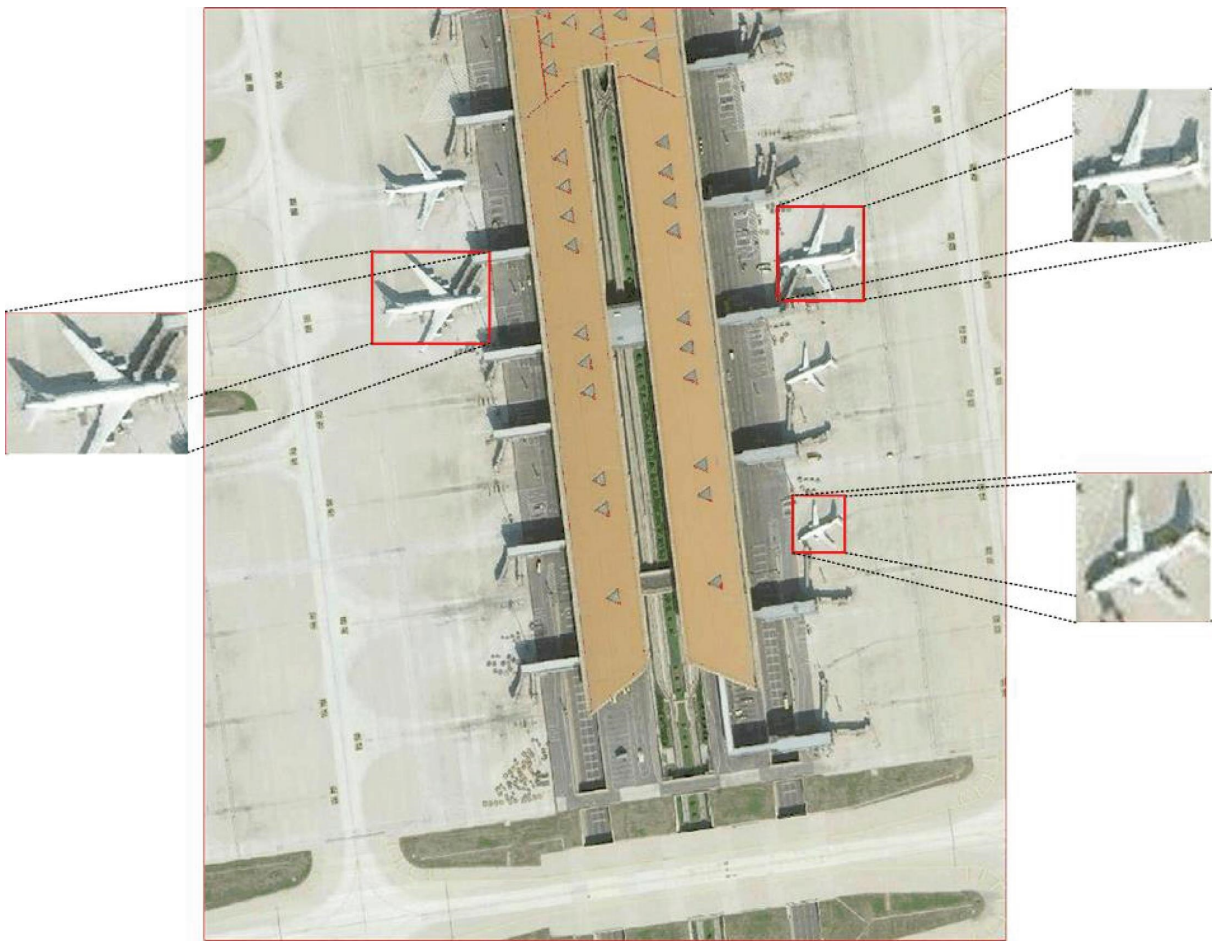
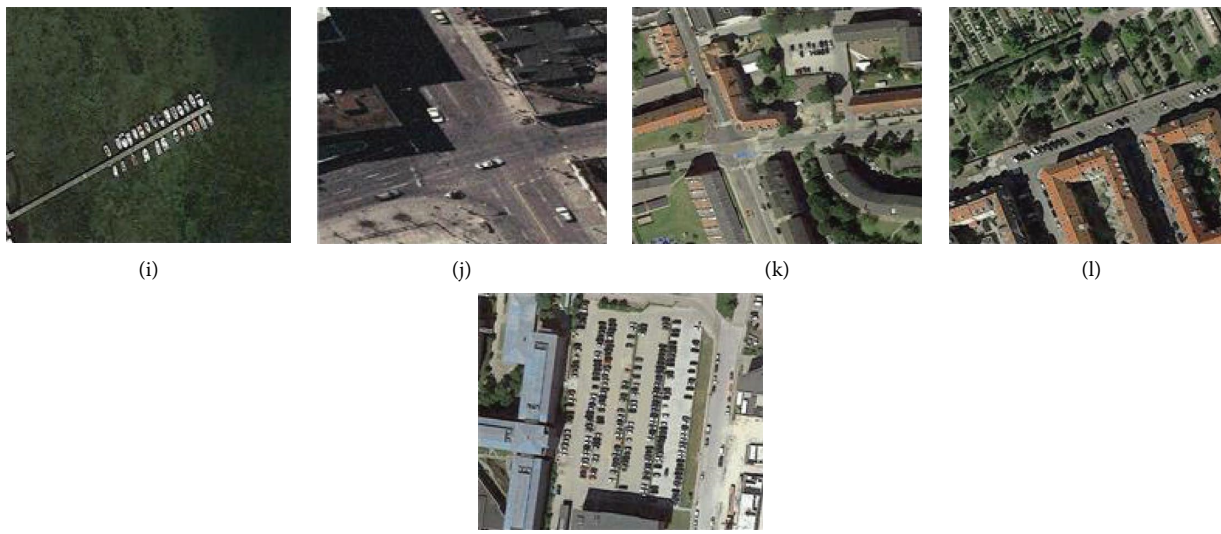


Figure 7: *K*-means algorithm generates anchors to match with targets of different sizes.

(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 8: Continued.



(i) (j) (k) (l) (m)

Figure 8: Images in the HRRSD datasets. (a) A ship; (b) a bridge; (c) a ground track field; (d) a storage tank; (e) a basketball court; (f) a tennis court; (g) an airplane; (h) a baseball diamond; (i) a harbor; (j) a vehicle; (k) a crossroad; (l) a *T* junction; (m) a parking lot.

dataset. The HRRSD dataset comprises training, validation, and test sets in the ratio of 1 : 1 : 2, following the original data division. Table 2 summarizes detailed division data for each target.

The annotation in VOC format was also incorporated into the 10-class remote sensing dataset, NV10. The target in the dataset comprised a harbor, a bridge, a baseball diamond, a ship, a basketball court, a vehicle, a tennis court, a storage tank, and a ground track field. Figure 9 represents the example images of the dataset. In this dataset, 650 images with targets were divided into the training, validation, and test sets in the ratio of 3 : 1 : 1, whereas 150 images without targets were assigned to the test set. Table 3 summarizes the detailed division information for each target.

Table 2: The division of each target in the HRRSD dataset.

Class	Train	Val	Test
Ship	950	948	1988
Bridge	1123	1121	2326
Ground track field	859	856	2017
Storage tank	1099	1092	2215
Basketball court	923	920	2033
Tennis court	1043	1040	2212
Airplane	1226	1222	2451
Baseball diamond	1007	1004	2022
Harbor	967	964	1953
Vehicle	1188	1186	2382
Crossroad	903	901	2219
T junction	1066	1065	2289
Parking lot	1241	1237	2480

3.1. Evaluation Index. The mean Average Precision (mAP) value was used as the accuracy evaluation index in this work. (4) and (5) can be used to calculate the detection's precision (P) and recall (R):

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{4}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{5}$$

where TP, TN, FP, and FN represent true-positive, true-negative, false-positive, and false-negative, respectively. As shown in (4) and (5), P and R are mutually constrained; the higher the value of one, the lower the value of the other. Thus, it is essential to employ a metric that balances P and R. In contrast, the Average Precision (AP) combines the impacts of P and R to indicate how effectively the model recognizes a particular category. The mAP averages the AP of all classes for the overall detection performance, which is defined as follows:

$$\begin{aligned} \mathrm{AP} &= \int_0^1 P_i(R_i)dR_i, \\ \mathrm{mAP} &= \frac{1}{C}\sum_{i=1}^{C} \mathrm{AP}_i, \end{aligned} \tag{6}$$

where C denotes the number of classes.

3.2. Model Learning Details. The HRRSD dataset was used to train the proposed model and compared models which included YOLOv2, YOLOv3, YOLOv4, YOLOv5, fast RCNN, fast-RCNN-r50+GACL Net, faster RCNN, and faster-RCNN-r50+GACL Net. The trained models were then tested on the HRRSD and NV10 dataset test sets. The stochastic gradient descent (SGD) was employed as the optimizer with a momentum equivalent to 0.9 and a weight decay value equal to 0.0005. The batch size is 8, indicating that the network accumulates 8 samples and then performs a

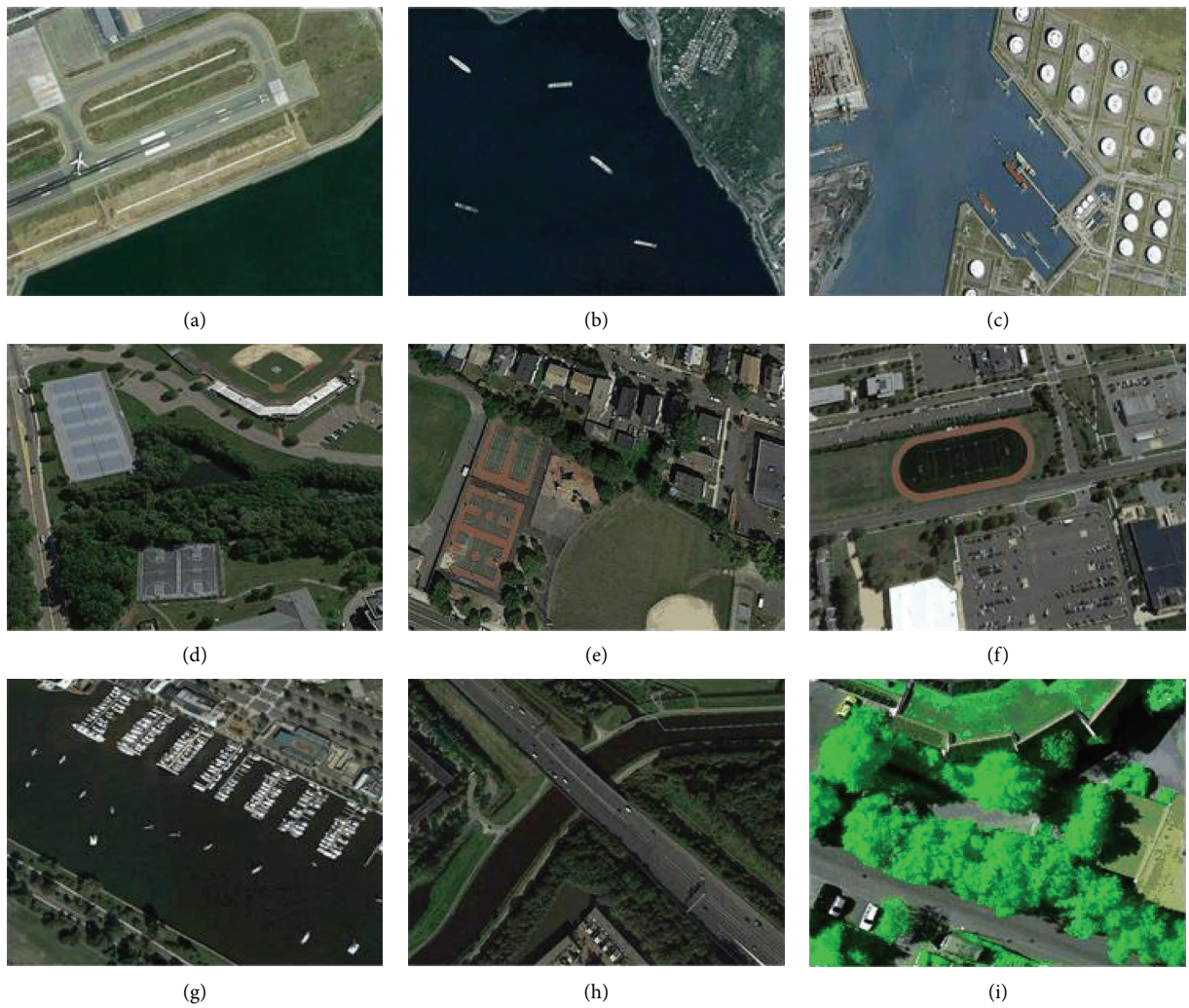


Figure 9: Images in the NWPU VHR-10 dataset. (a) An airplane; (b) a ship; (c) a storage tank; (d) a tennis court; a baseball diamond; (e) a basketball court; (f) a ground track field; (g) a harbor; (h) a bridge; (i) a vehicle.

Table 3: The division of the NV10 dataset for each target.

Target	Train	Val	Test
Ship	181	60	61
Bridge	74	26	24
Ground track field	98	33	32
Storage tank	400	125	130
Basketball court	95	32	32
Tennis court	314	102	108
Airplane	454	151	152
Baseball diamond	234	78	78
Harbor	134	43	47
Vehicle	286	95	96

forward propagation. Max epoch is 200. The initial learning rate is set to $1e-3$, and the learning rate decays every 10 epochs from 100 epochs. Ignore thresh is 0.5 as in the YOLOv3 paper. The object loss weight is 0.5, the categorical loss weight is 0.25, and the regression loss weight is 0.25. The other parameters are the same as the official model of YOLOv3. All experiments were conducted on an NVIDIA RTX3090 with 24 GB RAM.

4. Results

4.1. YOLOv3-DE. In this section, the performance of YOLOv3-DE was assessed using the HRRSD and NV10 dataset test sets. Figures 10 and 11 show the APs for each category, where YOLOv3-DE achieves AP values greater than 0.8 for the ten categories in the HRRSD dataset and all categories in the NV10 dataset. Figure 12 shows the qualitative results of YOLOv3-DE on the HRRSD dataset.

Figure 13 analyzes the TP and FP for each category, demonstrating that YOLOv3-DE has a high number of FP for categories such as vehicles, parking lots, and so on. Further investigation revealed that YOLOv3-DE recognized several small objects that were not marked in the HRRSD dataset, causing the computed AP to drop. Many small vehicles, for example, were detected but not annotated in the ground-truth annotation map, as shown in Figure 14.

4.2. Comparison Results. The proposed YOLOv3-DE model was further validated by comparing it to the following models: fast RCNN with ResNet50 as the backbone, GACL

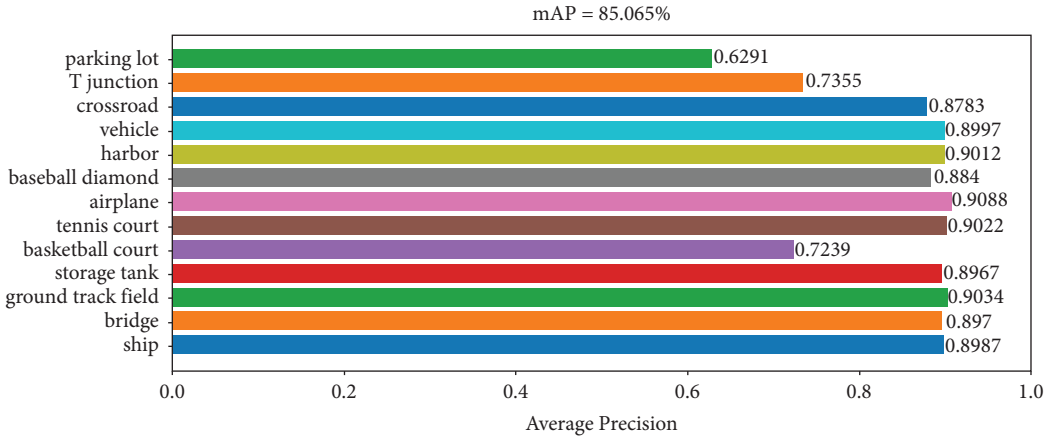


Figure 10: APs of each category on the HRRSD dataset.

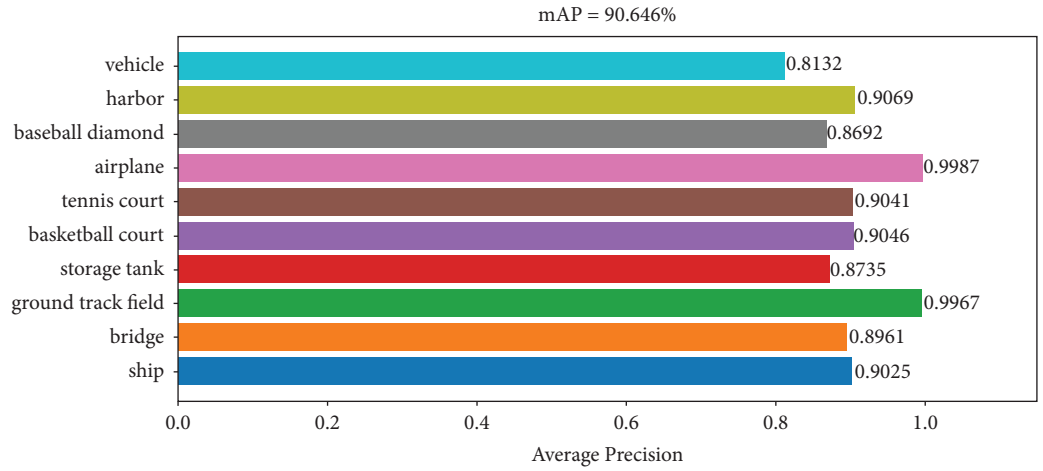


Figure 11: APs of each category on the NV10 dataset.

Net based on fast RCNN, faster RCNN with ResNet50 as the backbone, GACL Net based on faster RCNN, and Darknet53 as YOLOv3 for the backbone. Tables 4 and 5 compare the mAP of these compared models on the HRRSD and NV10 datasets, respectively, demonstrating that the YOLOv3-DE model practically outperforms the other compared models, including the state-of-the-art two-stage detector faster RCNN-based GACL Net on both datasets.

The experimental results demonstrated that our method enhanced the overall detection performance of 13 categories on the HRRSD dataset compared to YOLOv3, with mAP increasing from 80.58% to 85.065%. Some categories even outperform GACL-faster RCNN in terms of detection performance. Among them, the ship, bridge, basketball court, and baseball diamond have gained remarkable promotion. The number of targets in different scales was large in HRRSD, and the difference between the target and the surrounding background was small, making it difficult to extract features. It was discovered that the SGE and dilated encoder modules improve backbone feature extraction and scale fusion, demonstrating the effectiveness of our method. Furthermore, the *T* junction category has not been improved. Our method cannot achieve a high detection effect because the target of a *T* junction is too large or even exceeds that of a crossroad.

The mAP of our technique is the highest on the NV10 dataset. It outperforms the other two-stage networks by 85.596% to 90.646% of YOLOv3, and our detection performance is excellent in most categories. However, the category of baseball diamond has not been improved, and we will analyze it in subsequent ablation experiments.

The GACL faster RCNN has superior performance compared to YOLOv3, thanks to its two-stage model coupled with RPN to achieve high accuracy detection performance. Compared to other one-stage detection models, the two-stage network is more accurate and can solve more multiscale, small-target problems. YOLOv3 has the advantage of fast detection and high generality. YOLOv3 uses the anchor mechanism to generate a dense anchor box, which allows the network to perform target classification and bounding box coordinate regression directly on this basis. However, the accuracy is still lacking compared to the two-stage model. However, the GACL faster RCNN, as a two-stage model, has a clear division of labour between the two stages, which brings improved accuracy, but the speed of the two stages is slower than that of the one-stage model and

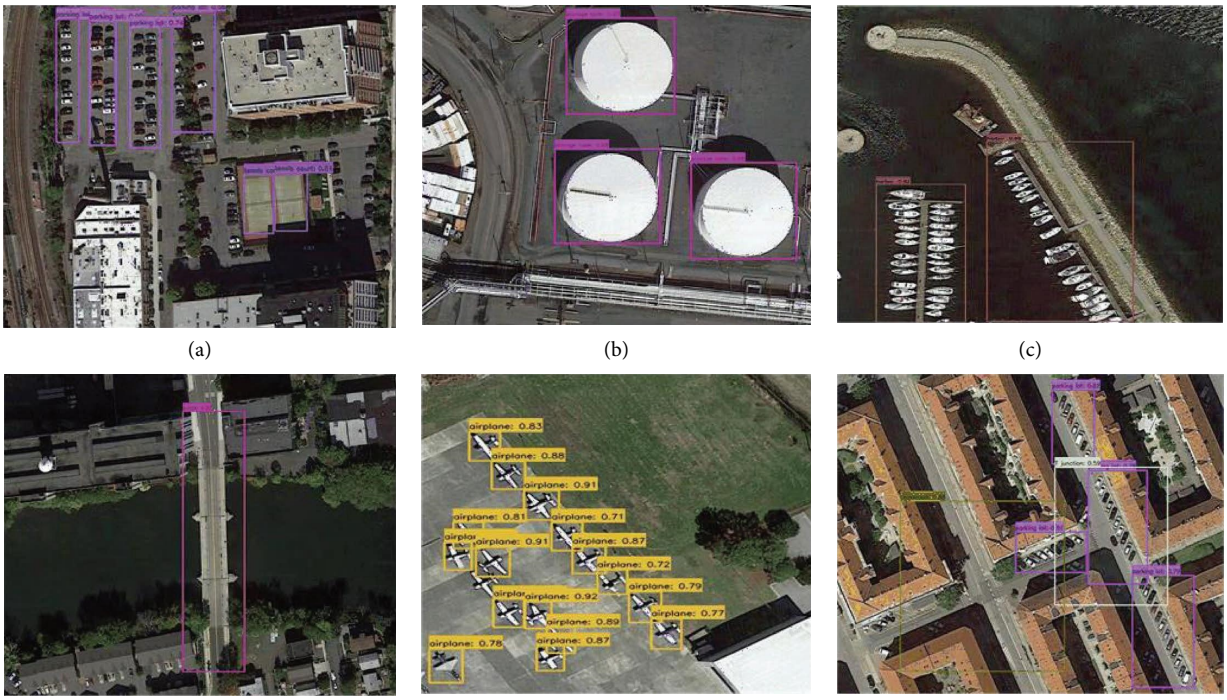


Figure 12: The detection results of YOLOv3-DE on the HRRSD datasets. (a) A parking lot; a basketball court; (b) a storage tank; (c) a harbor; (d) a bridge; (e) an airplane; (f) a *T* junction; a parking lot; crossroad.

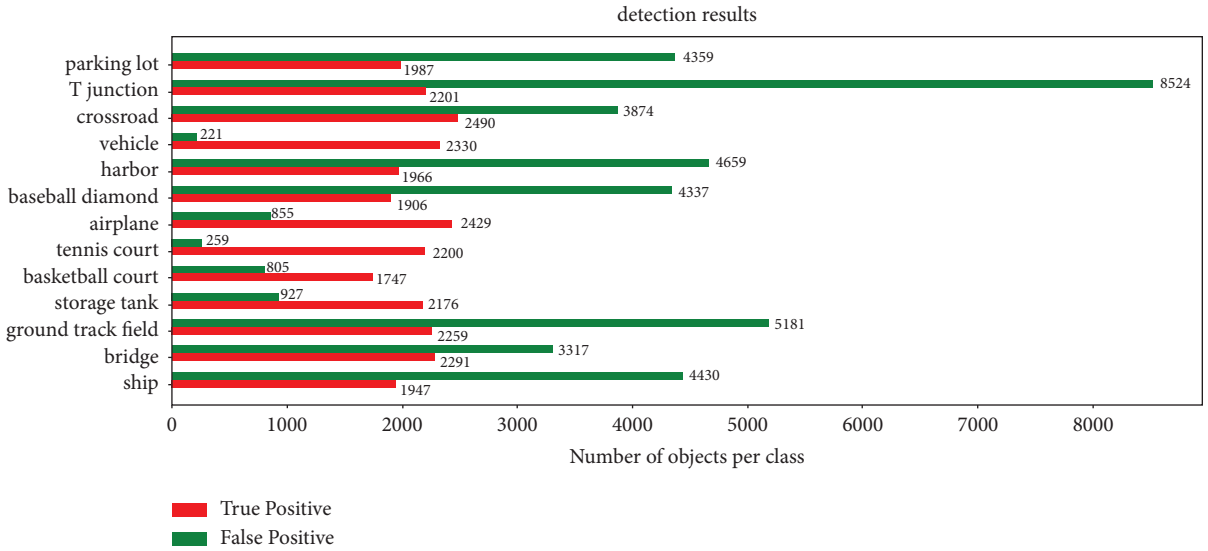


Figure 13: TP and FP for each category.

is not widely used in practice. Therefore, our YOLOv3-DE is based on the premise that the one-stage model can guarantee speed, and the overall improvement is obtained by improving the performance for multiscale, small target detection. But there is still a gap in detection performance compared to the latest YOLOv7. Figures 15 and 16 show the comparison between YOLOv3 and YOLOv3-DE, and it can be seen that the overall recognition of our method has gained some improvement.

The results in Table 6 show that the overall performance of YOLOv3-DE is better than that of YOLOv3. A smaller improvement in FPS and computational FLOPs is obtained. Params gained a 9% improvement in the number of parameters, thanks to the larger sensory field provided by the dilated encoder. Null convolution allows the field to be expanded without adding additional parameters while capturing multiscale contextual information.

Figure 14: Detection results of vehicles without label.

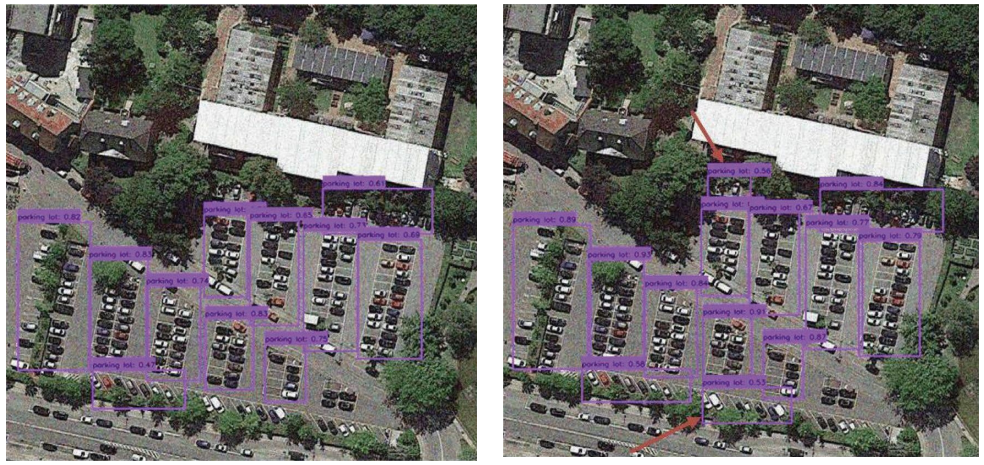
Table 4: Comparison experiment on the HRRSD dataset.

	Fast RCNN	GACL fast RCNN	Faster RCNN	GACL faster RCNN	YOLOv3	YOLOv3-DE	YOLOv4	YOLOv5	YOLOv7
Ship	75.0	74.3	88.5	88.5	78.8	89.87	85.09	95	95.8
Bridge	75.1	76.7	85.5	85.6	79.78	89.7	85.54	83.4	86.9
Ground track field	90.0	89.6	90.6	90.7	90.32	90.34	89.04	90.6	93.4
Storage tank	79.8	80.4	88.7	89.2	89.12	89.67	89.82	96	98.6
Basketball court	36.7	42.1	47.9	49.7	60.92	72.39	67.1	75.7	85.3
Tennis court	75.0	77.0	80.7	80.8	89.41	90.22	88.77	92.6	96.5
Airplane	83.3	85.1	90.8	90.8	90.84	90.88	90.43	94.7	97.7
Baseball diamond	83.6	82.6	86.9	87.2	73.54	88.4	81.38	83.5	88.4
Harbor	76.0	78.4	89.4	89.7	87.68	90.12	90.24	96.8	97.8
Vehicle	46.1	50.7	84.0	86.9	88.59	89.97	83.61	91.5	94.3
Crossroad	67.1	68.7	88.6	88.2	87.41	87.83	88.96	84.4	87.9
T junction	39.2	38.8	75.1	75.0	73.53	73.55	66.98	84	89.5
Parking lot	37.5	39.5	63.3	65.3	57.62	62.91	59.96	61	72.1
Mean AP	66.5	68.0	81.5	82.1	80.58	85.065	82.071	86.8615	91.092

Table 5: Comparison experiment on the NWPU VHR-10 dataset.

	Fast RCNN	GACL fast RCNN	Faster RCNN	GACL faster RCNN	YOLOv3	YOLOv3-DE	YOLOv4	YOLOv5	YOLOv7
Ship	62.8	63.5	89.9	89.8	83.3	90.25	90.5	90.1	93.5
Bridge	79.2	72.7	80.9	86.1	78.14	89.61	80.1	86.2	92.3
Ground track field	99.3	99.4	100.0	99.8	99.85	99.67	99.7	99.1	99.8
Storage tank	44.5	53.9	67.3	68.5	79.27	87.35	80.67	85.2	86.4
Basketball court	80.7	82.2	87.5	88.4	82.77	90.46	90.1	85.3	88.6
Tennis court	76.9	80.9	78.6	79.7	89.16	90.41	90.8	92.1	95.4
Airplane	90.0	90.3	90.7	90.9	90.86	99.87	92.8	97.3	99
Baseball diamond	90.0	96.9	89.2	88.8	90.71	86.92	98.3	94.3	98.9
Harbor	90.4	90.9	89.8	90.0	82.55	90.69	89.1	88.5	92.1
Vehicle	49.7	57.9	88.0	88.5	79.35	81.32	88.8	91.1	96.7
Mean AP	76.4	78.9	86.2	86.6	85.596	90.646	90.087	90.92	94.27

4.3. Ablation Experiments. An ablation study was conducted on the NV10 dataset in an attempt to validate the contribution of individual components of the proposed method to the overall performance. Table 7 summarizes the results of the experimental ablation, showing that both the SGE module and dilated encoder module can improve the detection performance. The results furnish sufficient validation for the feature extraction ability of the SGE module and the rich receptive field of the dilated encoder. However, the improvement in the detection performance for the background



(a) (b)

Figure 15: Comparison of YOLOv3 and YOLOv3-DE recognition results.

(a) (b)

Figure 16: Comparison of YOLOv3 and YOLOv3-DE recognition results.

Table 6: Comparison of YOLOv3 and YOLOv3-DE parameters.

	FPS	GFLOPs	Params (M)
YOLOv3-320	109	19.57	61.97
YOLOv3-416	88	33.08	61.97
YOLOv3-512	74	50.11	61.97
YOLOv3-608	50	70.66	61.97
YOLOv3-640	48	78.30	61.97
YOLOv3-DE-320	111	19.10	57.25
YOLOv3-DE-416	89	32.28	57.25
YOLOv3-DE-512	77	48.90	57.25
YOLOv3-DE-608	51	68.96	57.25
YOLOv3-DE-640	49	76.41	57.25

track field is limited, as it is already 100% without these modules. In addition, the model yields suboptimal baseball diamond detecting results. We hypothesize that the added dilated encoder module affects the performance of the single-stage YOLOv3 model due to the presence of many truncated samples of the baseball diamond in NV10.

Table 7: Ablation experiments on the NWPU VHR-10 dataset.

	Exp 1	Exp2	Exp3
SGE		√	√
Dilated encoder			√
Ship	83.3	86.2	90.25
Bridge	78.14	82.73	89.61
Ground track field	99.85	99.82	99.67
Storage tank	79.27	83.53	87.35
Basketball court	82.77	85.65	90.46
Tennis court	89.16	89.77	90.41
Airplane	90.86	98.61	99.87
Baseball diamond	90.71	90.59	86.92
Harbor	82.55	84.5	90.69
Vehicle	79.35	79.87	81.32
Mean AP	85.596	88.127	90.646

5. Conclusion

This paper presents the YOLOv3-DE model for remote sensing target detection, especially for small and multiscale targets. The attention module within the residual structure backbone network allowed YOLOv3-DE to effectively extract features from complex scenes. In addition, the dilated encoder module at the C5 feature layer superimposes multiscale receptive fields, hence enhancing the accuracy of detection for multiscale targets. In terms of performance, the experimental investigation provided more evidence that the YOLOv3-DE model outperforms other models, notably the faster RCNN-based GACL Net. The YOLOv3-DE model achieved a mAP of 85.065% on the HRRSD dataset, which is 3% higher than the faster RCNN-based GACL Net. In addition, the YOLOv3-DE model achieved 90.646% mAP on the NV10 dataset. Furthermore, an ablation study validated the contributions of the SGE and dilated encoder modules. Despite its adequate detection accuracy, the YOLOv3-De model is not ideal for low-resolution blurred images. Our future work will thus be based on investigations into improving the detection of targets in blurred and unclear images.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] C. I. Chang and D. C. Heinz, "Constrained subpixel target detection for remotely sensed imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, pp. 1144–1159, 2000.

[2] T. Toutin, "Geometric processing of remote sensing images: models, algorithms and methods," *International Journal of Remote Sensing*, vol. 25, pp. 1893–1924, 2004.

[3] N. G. Paterakis, M. Elena, G. Madeleine, S. Bart, and V. A. Walter, "Deep learning versus traditional machine learning methods for aggregated energy demand prediction," in *Proceedings of the 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Turin, Italy, January 2017.

[4] M. Ju, H. Luo, G. Wang, F. Hui, and D. Chang, "The application of improved YOLO V3 in multi-scale target detection," *Applied Sciences*, vol. 9, p. 3775, 2019.

[5] A. Cockburn and A. Firth, "Improving the acquisition of small targets," *People and Computers XVII—Designing for Society*, pp. 181–196, Springer, London, UK, 2004.

[6] J. Whitehill and C. W. Omlin, "Haar features for FACS AU recognition," in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, Southampton, UK, April 2006.

[7] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, June 2008.

[8] Z. Chen, K. Chen, and J. Chen, "Vehicle and pedestrian detection using support vector machine and histogram of oriented gradients features," in *Proceedings of the 2013 International Conference on Computer Sciences and Applications*, Wuhan, China, June 2013.

[9] A. Phaniendra, D. B. Jestadi, and L. Periyasamy, "Free radicals: properties, sources, targets, and their implication in various diseases," *Indian Journal of Clinical Biochemistry*, vol. 30, pp. 11–26, 2015.

[10] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, PR, USA, June 2014.

[12] J. Redmon and F. Ali, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[13] J. Redmon and F. Ali, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, PR, USA, June 2017.

[14] E. Van and G. Adam, "You only look twice: rapid multi-scale object detection in satellite imagery," 2018, https://arxiv.org/abs/1805.09512.

[15] X. Lu, Y. Zhang, Y. Yuan, and Y. Feng, "Gated and axis-concentrated localization network for remote sensing object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, pp. 179–192, 2020.

[16] Z. Zhu, H. Xianyu, Q. Guanqiu, L. Yuanyuan, C. Baisen, and L. Yu, "Brain tumor segmentation based on the fusion of deep semantics and edge information in multimodal MRI," *Information Fusion*, vol. 91, 2022.

[17] M. Li, S. Zang, B. Zhang, S. Li, and C. Wu, "A review of remote sensing image classification techniques: the role of spatio-contextual information," *European Journal of Remote Sensing*, vol. 47, pp. 389–411, 2014.

[18] Bo Li, Y. Junjie, W. Wei, Z. Zheng, and H. Xiaolin, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, PR, USA, June 2018.

[19] Y. Tian, "Anchored neighborhood regression based single image super-resolution from self-examples," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Phoenix, AZ, USA, September 2016.

[20] V. Lempitsky, K. Pushmeet, R. Carsten, and S. Toby, "Image segmentation with a bounding box prior," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*, Kyoto, Japan, September 2009.
[21] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December 2015.
[22] S. Ren, H. Kaiming, G. Ross, and S. Jian, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.
[23] W. Fang, L. Wang, and P. Ren, "Tinier-YOLO: a real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2020.
[24] Z. Tian, "Fcos: fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, Seoul, Korea, October 2019.
[25] W. Liu, "Ssd: single shot multibox detector," *European Conference on Computer Vision*, Springer, Cham, 2016.
[26] J. Redmon, "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, June 2016.
[27] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "Automatic ship detection based on RetinaNet using multi-resolution Gaofen-3 imagery," *Remote Sensing*, vol. 11, p. 531, 2019.
[28] A. Bochkovskiy, C. Y. Wang, H. Yuan, and M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934, Article ID 10934.
[29] Z. Cui, Z. Cao, and J. Yang, "Ship detection in large-scale SAR images via spatial shuffle-group enhance attention," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, pp. 379–391, 2021.
[30] Q. Chen, W. Yingming, Y. Tong, Z. Xiangyu, C. Jian, and S. Jian, "You only look one-level feature," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, June 2021.
[31] R. Zhao and X. Wang, "Counting vehicles from semantic regions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, pp. 1016–1022, 2013.
[32] Q.-L. Zhang and Yu-B. Yang, "Sa-net: shuffle attention for deep convolutional neural networks," in *Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Toronto, Ontario, Canada, June 2021.
[33] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 13, pp. 1–18, 2017.
[34] C.-Y. Wang, A. Bochkovskiy, H. Yuan, and M. Liao, "Scaled-yolov4: scaling cross stage partial network," in *Proceedings of the IEEE/cvf Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, June 2021.
[35] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: faster and better learning for bounding box regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12993–13000, 2020.
[36] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: a k-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100–108, 1979.
[37] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu, "Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, pp. 5535–5548, 2019.
[38] T.-Yi Lin, "Microsoft coco: common objects in context," in *Proceedings of the European Conference on Computer Vision*, Zurich, Switzerland, September 2014.
[39] C. Wang, X. Bai, S. Wang, J. Zhou, and P. Ren, "Multiscale visual attention networks for object detection in VHR remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, pp. 310–314, 2019.
[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.