

## RESEARCH ARTICLE

# Remote Sensing Image Detection Based on YOLOv4 Improvements

XUNKAI YANG<sup>1</sup>, JINGYI ZHAO<sup>1</sup>, HAIYANG ZHANG<sup>2</sup>, CHENXU DAI<sup>1</sup>, LI ZHAO<sup>3</sup>, ZHANLIN JI<sup>1,4</sup>, (Member, IEEE), AND IVAN GANCHEV<sup>4,5,6</sup>, (Senior Member, IEEE)

<sup>1</sup>College of Artificial Intelligence, North China University of Science and Technology, Tangshan 063210, China

<sup>2</sup>Department of Computing, Xi'an Jiaotong-Liverpool University, Suzhou 215000, China

<sup>3</sup>Research Institute of Information Technology, Tsinghua University, Beijing 100080, China

<sup>4</sup>Telecommunications Research Centre (TRC), University of Limerick, Limerick, V94 T9PX Ireland

<sup>5</sup>Department of Computer Systems, University of Plovdiv "Paisii Hilendarski," 4000 Plovdiv, Bulgaria

<sup>6</sup>Institute of Mathematics and Informatics—Bulgarian Academy of Sciences, 1040 Sofia, Bulgaria

Corresponding authors: Zhanlin Ji (zhanlin.ji@gmail.com) and Ivan Ganchev (ivan.ganchev@ul.ie)


This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0135700; in part by the MES under Grant No. D01-168/28.07.2022 for NCDSC part of the Bulgarian National Roadmap on RIs; and in part by the Telecommunications Research Centre (TRC), University of Limerick, Ireland.

**ABSTRACT** Remote sensing image target object detection and recognition are widely used both in military and civil fields. There are many models proposed for this purpose, but their effectiveness on target object detection in remote sensing images is not ideal due to the influence of climate conditions, obstacles and confusing objects presented in images, image clarity, and associated problems with small-target and multi-target detection and recognition. Therefore, how to accurately detect target objects in images is an urgent problem to be solved. To this end, a novel model, called YOLOv4\_CE, is proposed in this paper, based on the classical YOLOv4 model with added improvements, resulting from replacing the backbone feature-extraction CSPDarknet53 network with a ConvNeXt-S network, replacing the Complete Intersection over Union (CIoU) loss with the Efficient Intersection over Union (EIoU) loss, and adding a coordinate attention mechanism to YOLOv4, as to improve its remote sensing image detection capabilities. The results, obtained through experiments conducted on two open data sets, demonstrate that the proposed YOLOv4\_CE model outperforms, in this regard, both the original YOLOv4 model and four other state-of-the-art models, namely Faster R-CNN, Gliding Vertex, Oriented R-CNN, and EfficientDet, in terms of the mean average precision (*mAP*) and *F1 score*, by achieving respective values of 95.03% and 0.933 on the NWPU VHR-10 data set, and 95.89% and 0.937 on the RSOD data set.

**INDEX TERMS** Remote sensing, target object detection, ConvNeXt, EIoU loss, coordinate attention.

## I. INTRODUCTION

Remote sensing image target detection aims to judge the content of remote sensing images according to the individual features of the images, identify their attributes, and then locate and classify the target objects. This task has found wide application in multiple fields for civil use. For instance, it can be used for investigation and monitoring of natural resources, such as land-, mineral-, forest-, wetland, and water resources. In the field of environmental control and protection, remote sensing image detection can be

The associate editor coordinating the review of this manuscript and approving it for publication was Wenming Cao .

used for monitoring and evaluating atmospheric-, water-, ecological-, soil environments, etc. In the field of disaster emergency, its main applications include monitoring of disaster elements and evaluating the risk of their appearance, with subsequent recovery and reconstruction. In the field of agriculture in rural areas, its applications mainly include stabilization of food production, prevention and control of major disease outbreaks and epidemics, development and expansion of rural industries, monitoring of rural habitat environment, and performing agricultural statistics. Remote sensing image target detection is used also in many other fields, such as autonomous driving cars, unmanned aerial vehicles (UAVs), intelligent robotics, etc. In the military

field, remote sensing image target detection based on high-definition satellite images is used for military survey, defense, action prevention, etc.

Prior to 2012, the traditional feature-based object detection models were mainly based on manual feature extraction performed by experts. Since 2012, the rise of the convolutional neural networks (CNNs) has been a major step forward in this area, especially with the emergence of the Visual Geometry Group Network (VGGNet) [1], GoogleNet [2], ResNet [3], and Region-based CNN (R-CNN) [4]. Consequently researchers began to optimize and improve R-CNN and, as a result of these efforts, the Scale Pyramid Pooling Network (SPPNet) [5], Fast R-CNN [6], and Faster R-CNN [7] emerged one after another. All these models are representatives of the *two-stage* target object detection models which first generate a series of sparse candidate frames, followed by candidate frames verification, classification, and regression to improve the scores and locations [8]. At present, the horizontal bounding box representation is widely used in the area of target object detection. However, with this method, a confusion of horizontal objects may occur when trying to detect dense small objects. A model of sliding vertices of the horizontal bounding box to detect multi-oriented objects, called Gliding Vertex, is proposed in [9]. A Rotation-equivariant Detector (ReDet) is proposed in [10] to encode rotation equivariance and rotation invariance. On the basis of rotation equivariance features, a Rotation-invariant Region of Interest (RiRoI) Align is also presented there to extract rotation-invariant features from equivariant features according to the orientation of the Region of Interest (RoI). Based on Faster R-CNN, a context-aware detection network (CAD-Net) is proposed in [11] to integrate global context information into target detection. In addition, a spatial-and-scale-aware attention module is designed with the focus on more informative regions and features. The Oriented R-CNN model, proposed in [12], utilizes an oriented Region Proposal Network (RPN) to directly generate high-quality oriented proposals at almost no cost. Even though high accuracy and localization can be achieved with the *two-stage* models, their more complex training and low operational speed limit their application for *real-time* target object detection. But the pursuit of accuracy needs to be supported by speed as well. So, *one-stage* target object detection models, such as You Only Look Once (YOLO) [13] and Single Shot Multibox Detector (SSD) [14], have appeared with the aim of losing an acceptable range of accuracy in order to maximize the speed of detection to the extent of approaching a real-time detection. Both YOLO and SSD, however, cannot perfectly handle the graphic area, resulting in high detection error and missing rates. In addition, SSD does not consider the relationship between different scales, so it has limitations in detecting small objects, whereas for YOLO it is easier to learn general features, and its operational speed is higher [15]. Among different YOLO versions, YOLOv4 [16] is the most outstanding one with respect to both the performance and operational speed achieved.

The objective of this paper is to come up with a novel model, called YOLOv4\_CE, based on YOLOv4 improvements, as to achieve better remote sensing image detection performance. The main contributions of the paper are the following:

- 1) Replacing the feature extraction backbone (CSP Darknet53) of YOLOv4 with ConvNeXt-S [17] in order to make the model extract features more effectively and by this to lessen the computation of redundant information at the feature layer and reduce the model size;
- 2) Integrating the coordinate attention (CA) mechanism [18] into YOLOv4, so as to increase the receptive field and allow the model to pay more attention to important parts of the processed images;
- 3) Replacing the Complete Intersection over Union (CIoU) loss [19] with the Efficient Intersection over Union (EIoU) loss [20] in the loss function of YOLOv4 as to achieve faster convergence and improve the regression precision;
- 4) Verifying (by comparison to five state-of-the-art models based on experiments conducted on two open data sets – NWPU VHR-10 and RSOD) that these new elements, introduced into YOLOv4, do indeed improve its remote sensing image detection performance.

## II. BACKGROUND

### A. ATTENTION MECHANISMS

Attention mechanisms were first proposed and used for natural language processing (NLP) and text alignment in machine translation. In the field of computer vision, attention mechanisms are used to improve the performance of the utilized neural networks. The existing attention mechanisms include Squeeze-and-Excitation (SE) [21], Convolutional Block Attention Module (CBAM) [22], Coordinate Attention (CA) [18], etc. SE is used to solve the loss problem caused by the diverse importance of different channels of the feature map during the convolution pooling but it ignores the importance of positional information. Considering the shortcomings of SE, CBAM integrates two attention mechanisms, namely channel attention and spatial attention. By reducing the number of channels and using a large-scale convolution for the utilization of location information, CBAM can not only reduce the number of parameters and save computing power, but also can be integrated seamlessly into any CNN architecture. However, convolutions can only capture local relations and fail in modeling long-range dependencies which are essential for computer vision tasks, [18]. CA effectively integrates spatial coordinate information into the generated attention graph by embedding positional information into the channel attention in order to reduce the loss caused by the 2D global pooling and decomposes the channel attention into two parallel 1D feature encodings, resulting in a significant gain for intensive prediction tasks.

### B. MULTI-SCALE FEATURE INTEGRATION

In the field of target object detection, integrating the features of different scales is a vital task to improve the performance of target objects distinguishing from the image background. The resolution of high-level features is low, and the perception of details is poor, but the semantic information is rich. On the contrary, the resolution of low-level features is high, and the details and location information are rich, but the semantic information is poor. The integration of features at different levels allows to improve the target object detection performance. The existing feature integration techniques can be divided into early integration and late integration ones, depending on whether the prediction takes place before or after the feature integration. Early integration includes classic methods such as *concatenation*, *addition*, etc. *Concatenation* directly connects two features, and the final output feature dimension is the sum of the two feature dimensions. *Addition* adopts a parallel strategy to combine two feature vectors into a complex vector. Late integration combines the detection results of different levels. For instance, the feature pyramid network (FPN) [23], [24] first performs pyramid fusion followed by detection performed separately on each fused feature level. FPN conveys strong semantic features from top to bottom and combines upper-level feature information through upsampling to obtain the prediction map. In general, FPN can reduce the extra consumption of computation power and memory. The FPN structure, utilized by YOLOv4, is shown in Figure 1, where  $C_i (i = 2, 3, 4, 5)$  represents the  $i^{\text{th}}$  ResNet convolution groups and  $P_i$  represents the  $i^{\text{th}}$  feature map.  $P_5$  is obtained by a  $1 \times 1$  convolution of  $C_5$ . Integration with the upsampled feature maps is used to obtain the new feature map  $P_j (j = 4, 3, 2)$  from the corresponding features of  $C_j$ . As shown in Figure 2, a  $1 \times 1$  convolution operation is performed first on each feature map  $C_j$  and the result is then integrated with the upsampled feature map  $P_{j+1}$  to obtain the new feature map  $P_j$ , which has the same size as the lower-layer feature map. The final feature maps are generated by a  $3 \times 3$  convolution.

### III. RELATED WORK

As mentioned in the Introduction, the target object detection models, adopting CNNs, are divided into two main groups:

- 1) *Two-stage* models, which first generate regional recommendations and then perform classification and regression (Figure 3).
- 2) *One-stage* models, which skip the process of generating the selected area through the candidate framework and directly generate the category probability and location coordinate value of the object to be detected, identified, and classified, which increases their operational speed despite the slight flaw in accuracy. In addition, these models are smaller in size and easier to optimize [8].

The main (anchor-based) representatives of these two groups are briefly described in the subsections below.

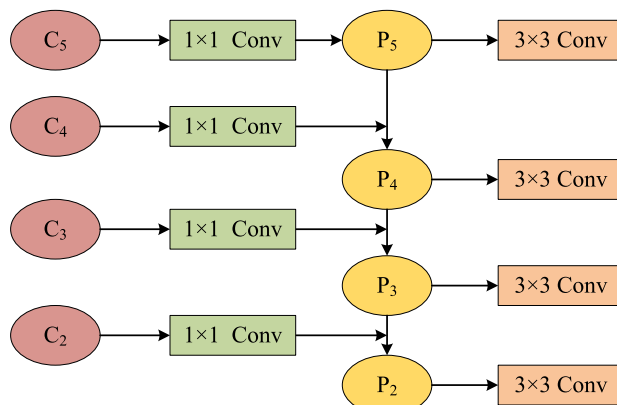


FIGURE 1. The FPN structure, utilized by YOLOv4.

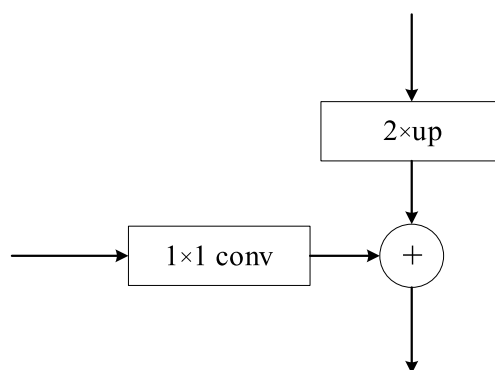


FIGURE 2. The side connection schema of the FPN, utilized by YOLOv4.

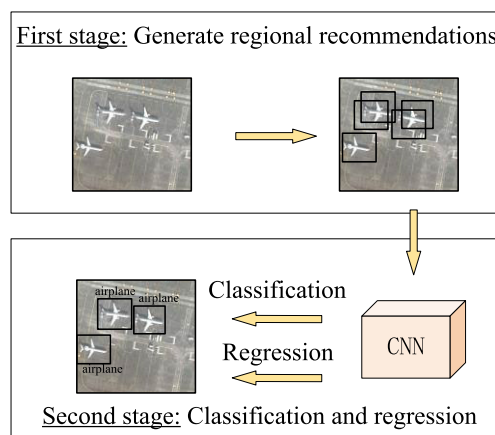


FIGURE 3. Two-stage target object detection.

### A. TWO-STAGE TARGET OBJECT DETECTION MODELS

The two-stage target object detection models are mostly represented by the R-CNN series, which achieve excellent target object detection accuracy by using deep CNNs to classify object locations, a.k.a. “object proposals” [7]. From the emerged incarnations of R-CNN (i.e., Fast R-CNN [6], Faster R-CNN [7], Mask R-CNN [25], and Mesh R-CNN [26]), Faster R-CNN is the current leading model used in several benchmarks [25]. Thus, it was selected as the main representative of the R-CNN group for performance comparison with the proposed YOLOv4\_CE model.

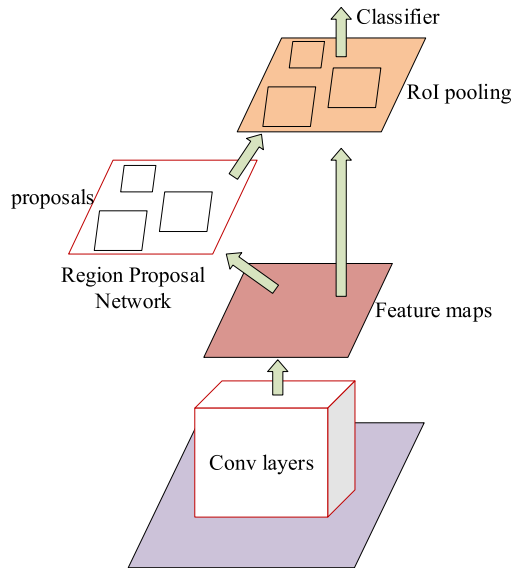


FIGURE 4. The Faster R-CNN model.

Faster R-CNN was proposed by Ren *et al.* [7]. The neural network used is VGG16, and the dimensions of the input image are  $224 \times 224$ . As shown in Figure 4, Faster R-CNN first extracts the feature maps of the image from the convolutional layers, and the maps are shared to a RPN layer to generate region proposals. The RPN layer judges whether anchors are positive or negative by *SoftMax*, and then corrects anchors by bounding box regression to obtain accurate proposals. The RoI pooling layer combines feature maps and proposals, which are sent to the fully connected layer to judge the category of the target object and obtain its exact location.

Overall, Faster R-CNN is not only a cost-efficient model, but also presents an effective way for improving the accuracy of target object detection [7]. It integrates feature extraction, proposal extraction, bounding box regression and classification into a network, which is really an end-to-end framework. The model performs well when trained and tested using single-scale images, which also improves its operational speed, but it still cannot meet the requirements for *real-time* target object detection.

### B. ONE-STAGE TARGET OBJECT DETECTION MODELS

The existing versions of YOLO are the most balanced one-stage target object detectors in terms of accuracy and operational speed achieved [8]. However, a new set of object detection models, called EfficientDet [27], has been recently proposed, utilizing a weighted bi-directional FPN (BiFPN) in trying to achieve better accuracy and efficiency [8]. These models are presented in the following subsections.

#### 1) YOLO

You Only Look Once (YOLO) is a family of models started out in 2016 by Redmon *et al.* [13]. With its different versions, YOLO presents a new approach to target object detection as it only needs to “look” once at an image to detect the objects

and their locations on it. For this, instead of repurposing classifiers to perform detection, it frames object detection as a single regression problem to spatially separated bounding boxes and associated class probabilities, which are predicted by a single CNN directly from the entire image in one step. YOLO trains on full images and directly optimizes its performance for object detection.

Among the different YOLO versions, the Darknet-based version 4 (YOLOv4) is the most accurate YOLO version, especially if a computer-vision engineer is in pursuit of state-of-the-art results and can perform additional customization on the model [28]. That is why YOLOv4 was selected as a basis for the elaborated model, proposed in this paper, and as the main YOLO representative for the performance comparison of models performed.

The YOLOv4 structure is shown in Figure 5. The model uses many optimization strategies based on maintaining the original YOLO target object detection structure. The backbone network, utilized for extracting the features of the target objects, is CSPDarknet53 [29]. In the feature integration stage, a Spatial Pyramid Pooling (SPP) module [5] and a Path Aggregation Network (PAN) [30] are used to further improve the ability of feature integration, and the CIOU loss [19] is used by the loss function to further consider the aspect ratio, overlapping area, and center distance between the prediction frame and target frame. The CBM module is composed of convolution (Conv), Batch normalization (BN) [31], and Mish activation function, whereas the CBL module is composed of Conv, BN, and Leaky\_ReLU [32] activation function. The dimensions of convolution cores in front of the Cross-Stage Partial connections (CSP) module are  $3 \times 3$ , which is equivalent to downsampling [33]. SPP uses fixed-block pooling operation, with the maximum pooling for the blocks with a kernel size of  $1 \times 1$ ,  $5 \times 5$ ,  $9 \times 9$ , and  $13 \times 13$ , which refers to tensor splicing, dimension expansion, and outputting, after a series of concatenations.

#### 2) EFFICIENTDET

EfficientDet [27] uses as a backbone the EfficientNet [34] – a pre-trained network based on ImageNet data set. The 3-7 level feature maps (i.e., P3, P4, P5, P6, and P7) are extracted from the backbone, fed into the BiFPN layer, then integrated (from top to bottom), and finally sent to the prediction network and category prediction network, as shown in Figure 6.

EfficientDet proposes a new compound scale method for target object detection by using a larger backbone and changing all aspects of the backbone, BiFPN, classification network, bounding box prediction network, and resolution through a recombination coefficient  $\varphi$ , as follows:

- 1) Backbone: The recombination coefficient  $\varphi$  corresponds to EfficientNet\_B +  $\varphi$  in EfficientNet.
- 2) BiFPN:  $W_{bifpn}$  is the width of BiFPN and  $D_{bifpn}$  is the depth of BiFPN, as shown below:

$$W_{bifpn} = 64 \times (1.35^\varphi); D_{bifpn} = 2 + \varphi. \quad (1)$$

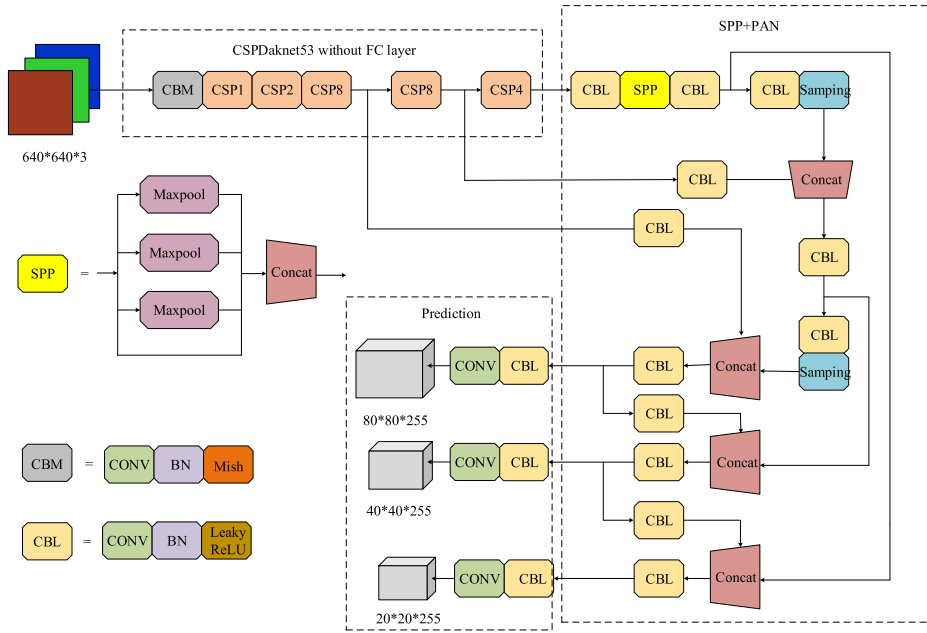


FIGURE 5. The YOLOv4 structure.

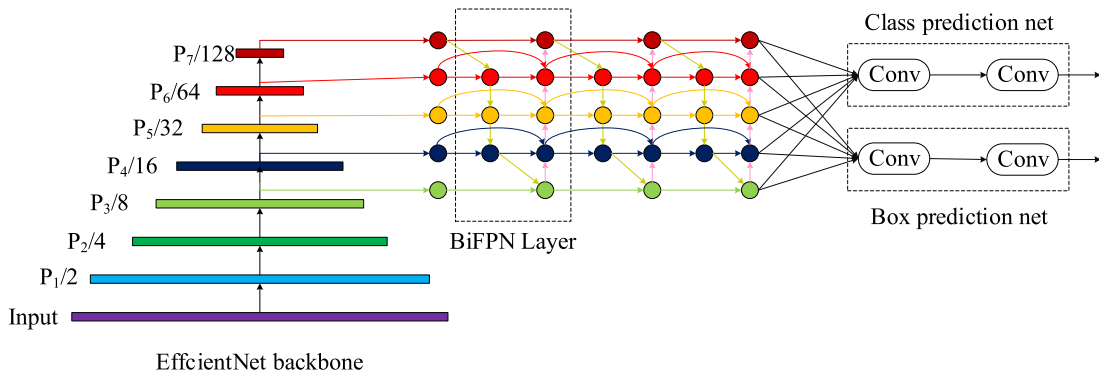


FIGURE 6. The EfficientDet-d0 structure.

- 3) Classification and bounding box prediction networks: The width is the same as that of BiFPN,  $W_{class}$  and  $D_{class}$  are the width and depth of the classification network, respectively, and  $W_{predict}$  and  $D_{predict}$  are the width and depth of the prediction network, respectively, as shown in (2) and (3):

$$W_{class} = W_{predict} = W_{bifpn}; \quad (2)$$

$$D_{class} = D_{predict} = 3 + \lceil \varphi/3 \rceil. \quad (3)$$

- 4) Resolution of input images: As the feature map input to the BiFPN layer is done at levels 3 to 7, the input resolution must be divisible by  $2^7$ . When increasing the resolution, the following linear relation shall be satisfied:

$$Input = 512 + 128 \times \varphi. \quad (4)$$

#### IV. PROPOSED MODEL-YOLOv4\_CE

This section proposes various improvements to the classical YOLOv4 model, namely replacing the CSPDarknet53

backbone with ConvNeXt-S [17], integrating the coordinate attention (CA) mechanism [18], and replacing the CIoU loss [19] with the EIou loss [20] in the loss function. The resultant model, whose structure is shown in Figure 11, is called YOLOv4\_CE.

##### A. ConvNeXt-S

ConvNeXt [17] refers to the structural design idea of Swin Transformer [35] to improve the CNN, based on ResNeXt [36], Figure 7. Macroscopically, ConvNeXt has four stages stacked by several blocks. The number of blocks in each stage is different and the stacking times are adjusted from (3, 4, 6, and 3) to (3, 3, 9, and 3). The stem cell of ResNet50 contains a  $7 \times 7$  convolution layer with a step size of 2 and a maximum pooling layer. The stem cell is replaced with a convolution layer with a convolution core size of 4 and a step size of 4. By using the idea of ResNet, the block convolution is used for the  $3 \times 3$  convolution layer in the bottleneck block to increase the network width to the same number of channels as Swin Transformer (i.e., from 64 to 96). ResNeXt first reduces

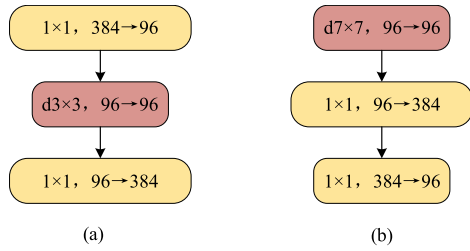


FIGURE 7. (a) The ResNeXt block structure; (b) The ConvNeXt block structure.

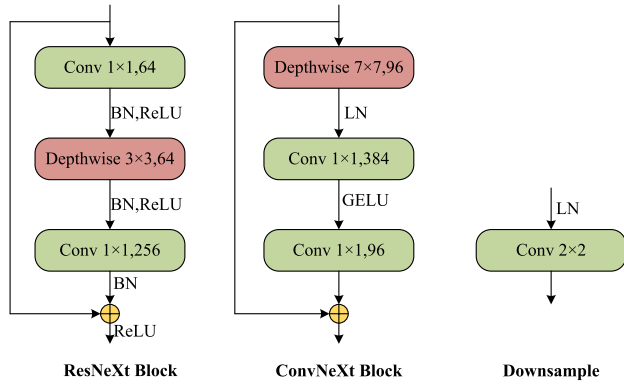


FIGURE 8. The ResNeXt, ConvNeXt, and downsampling structure.

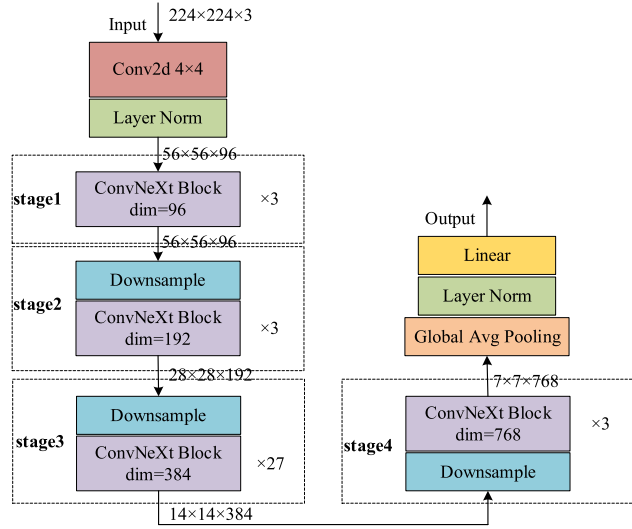


FIGURE 9. The ConvNeXt-S block structure, utilized by the proposed YOLOv4\_CE model.

the dimension by  $1 \times 1$  convolution, then applies depthwise convolution, and finally increases the dimension by  $1 \times 1$  convolution to form a bottleneck (Figure 7a). ConvNeXt lifts the depthwise convolution up and increases the convolution kernel to  $7 \times 7$ . So, first it applies depthwise convolution, then  $1 \times 1$  convolution to increase the dimension, and finally  $1 \times 1$  convolution to reduce the dimension to form an inverted bottleneck (Figure 7b).

Compared to ResNeXt, in the detailed ConvNeXt design (Figure 8), ReLU [37] is replaced with GELU [38], the

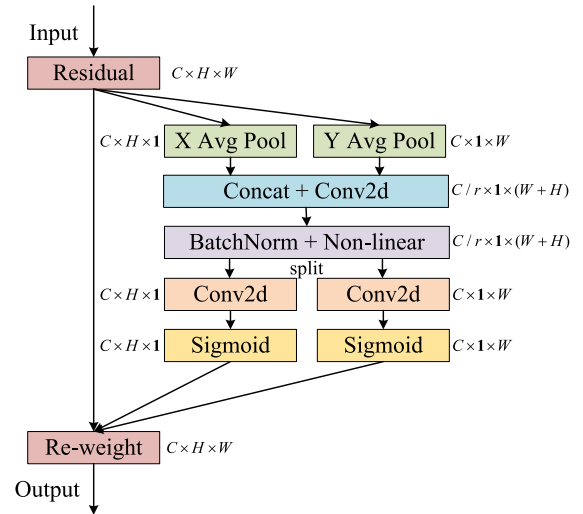


FIGURE 10. The coordinate attention (CA) mechanism.

activation function is reduced, two standardized BatchNorm (BN) [31] layers are removed, BN is replaced with Layer Normalization (LN) [39], and a  $2 \times 2$  convolution layer with a step size of 2 for spatial downsampling is used.

ConvNeXt has different architectures depending on different stacks of blocks used. In the proposed YOLOv4\_CE model, the ConvNeXt-S architecture is utilized with (3, 3, 27, 3) stacking, as shown in Figure 9.

### B. COORDINATE ATTENTION (CA)

The Coordinate Attention (CA) mechanism [15] encodes the channel relationship and long-term dependencies by accurate positional information with a simple overall structure flow, as shown in Figure 10.

Firstly, the input feature graph is divided into two directions of width and height for global average pooling. The output at height  $h$  and width  $w$  of channel  $C$  can be expressed as follows:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \leq i < W} x_c(h, i); \quad (5)$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \leq j < H} x_c(j, w). \quad (6)$$

Then, CA stitches the generated aggregation feature map, performs  $1 \times 1$  convolution, and obtains function  $f$  after applying normalization and activation function, as shown below:

$$f = \delta \left( F_1 \left( \left[ z^h, z^w \right] \right) \right). \quad (7)$$

This is followed by two convolutions and *sigmoid* activation function for  $f^h$  and  $f^w$ , respectively, and transforming these to tensors with the same channel, as follows:

$$g^h = \sigma \left( F_h \left( f^h \right) \right); \quad (8)$$

$$g^w = \sigma \left( F_w \left( f^w \right) \right). \quad (9)$$

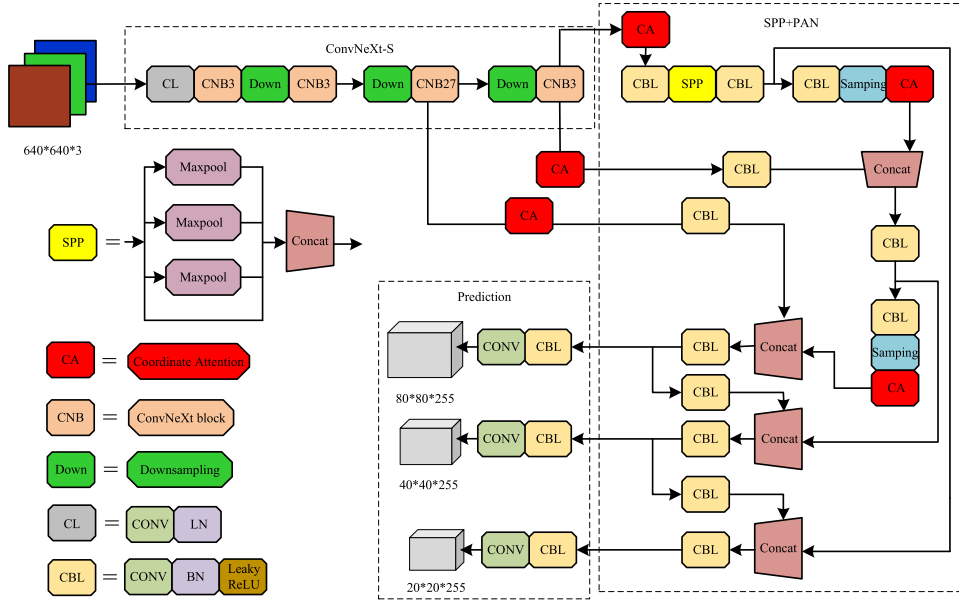


FIGURE 11. The structure of the proposed YOLOv4\_CE model.

Finally, CA extends  $g^h$  and  $g^w$  outputs as to use these as attention weights. The final output is:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j). \quad (10)$$

### C. LOSS FUNCTION

The task of the target object detection is to recognize and locate target objects, for which a loss function is utilized to make the recognition and localization more accurate. In YOLOv4, the Complete Intersection over Union (CIoU) loss is used as a loss function, which is formulated in [19] as:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v, \quad (11)$$

where  $b$  and  $b^{gt}$  denote the central points of the predication box set  $A$  and the ground truth box set  $B$ , respectively,  $\rho^2(b, b^{gt})$  denotes the Euclidean distance between these central points,  $\alpha$  denotes the weighting factor,  $v$  is used to measure the consistency of the relative proportions of the two rectangular boxes, and  $IoU$  is calculated as follows:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (12)$$

Although the CIoU loss simultaneously considers the overlap area, the distance between central points, and the aspect ratio of the bounding box, the difference in the aspect ratio is measured only by  $v$ , ignoring the real difference between the width and height, and their confidence levels.

The Efficient IoU (EIoU) loss allows to achieve faster convergence by calculating the height and width of the target and predicted frames separately, as shown below:

$$L_{EIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{C_w^2} + \frac{\rho^2(h, h^{gt})}{C_h^2}, \quad (13)$$

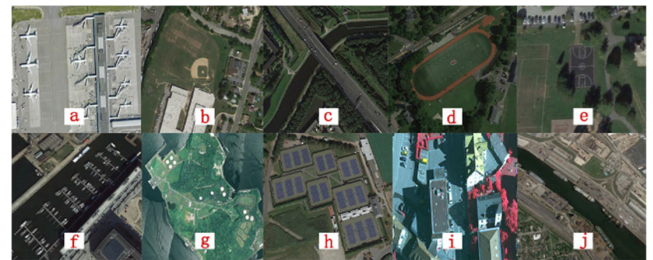


FIGURE 12. Sample images of the NWPU VHR-10 data set, containing objects of a given class: (a) airplane; (b) baseball diamond; (c) bridge; (d) ground track field; (e) basketball court; (f) harbor; (g) storage tank; (h) tennis court; (i) vehicle; (j) ship.

where  $h$  and  $w$  denote the height and width of the target frame,  $h^{gt}$  and  $w^{gt}$  denote the height and width of the predicted frame, and  $C_h$  and  $C_w$  denote the height and width of the minimum bounding rectangle covering the target and predicted frames.

As the EIoU loss splits the loss item of the aspect ratio into the difference between the width and height of the predicted frame and the width and height of the minimum bounding box, respectively, it allows to accelerate the convergence and improve the regression precision. These were the main reasons for adopting the EIoU loss for use as a loss function by the proposed YOLOv4\_CE model.

## V. EXPERIMENTS

### A. DATA SETS

Experiments were conducted on two open data sets, which are used for object detection in remote sensing images, namely the Northwestern Polytechnical University Very High Resolution 10 (NWPU VHR-10)<sup>1</sup> data set and the Remote Sensing Object Detection (RSOD)<sup>2</sup> data set.

<sup>1</sup><http://pan.baidu.com/s/1hqwzXcG>

<sup>2</sup><https://github.com/RSIA-LIESMARS-WHU/RSOD-Dataset->

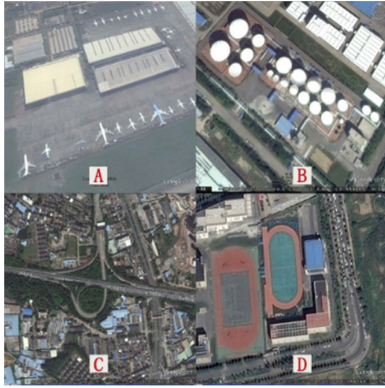


FIGURE 13. Sample images of the RSOD data set, containing objects of a given class: (a) aircraft; (b) oil tank; (c) overpass; (d) playground.

TABLE 1. Splitting of data sets into training, validation, and test subsets.

	NWPU VHR-10	RSOD
Training subset	416 images	625 images
Validation subset	104 images	156 images
Test subset	130 images	195 images

The NWPU VHR-10 data set includes 650 very high resolution (VHR) optical remote sensing images, containing 3775 total object instances covered by 10 object classes, including 757 airplanes, 390 baseball diamonds, 124 bridges, 163 ground track fields, 159 basketball courts, 224 harbors, 655 storage tanks, 524 tennis courts, 477 vehicles, and 302 ships, examples of which are shown in Figure 12. In addition, this data set includes 150 images that do not contain any target objects, which are used for semi-supervised learning-based object detection and weakly supervised learning-based object detection [40], and thus these images were not used in the conducted experiments.

The RSOD data set includes 976 images containing 6950 total object instances covered by four object classes, including 4993 aircrafts, 1586 oil tanks, 180 overpasses, and 191 playgrounds [41], examples of which are shown in Figure 13.

In the experiments, the utilized images of the data sets were randomly split into three different subsets, used respectively for models’ training (64% of the total images utilized), validation (16%), and testing (20%), as shown in Table 1. By using these percentages, the utilized images in each dataset were split five times in different subset conglomeration as to eliminate the test contingency, and the obtained results are shown under the corresponding experiment number in Tables 2-7 & 9-14 below.

**B. EVALUATION METRICS**

In the conducted experiments, the target object detection performance of the proposed YOLOv4\_CE model was compared to that of five state-of-the-art models, namely Faster R-CNN (backbone: ResNet50), Gliding Vertex (backbone: ResNet50), Oriented R-CNN (backbone: ResNet50), EfficientDet, and YOLOv4, based on *precision* and *recall*.

TABLE 2. Average precision results (%) of Faster R-CNN on NWPU VHR-10 data set.

Object class	Experiment 1	Experiment 2	Experiment 3	Experiment 4	Experiment 5
Airplane	95.36	98.84	96.39	99.84	94.80
Baseball diamond	95.27	96.02	96.64	95.29	93.62
Basketball court	85.49	89.29	85.16	92.20	84.99
Bridge	69.95	75.30	70.49	74.06	69.37
Ground track field	99.24	99.49	99.70	99.28	99.20
Harbor	90.48	93.07	89.65	91.26	93.07
Ship	82.85	79.57	84.93	80.87	79.56
Storage tank	65.30	68.71	68.77	67.64	62.77
Tennis court	88.79	87.04	90.74	93.43	83.02
Vehicle	57.10	51.15	56.62	52.62	44.87
<i>mAP</i>	<b>82.98</b>	<b>83.85</b>	<b>83.91</b>	<b>84.65</b>	<b>80.53</b>

*Precision* indicates the proportion of true positive (TP) samples in the prediction results, whereas *recall* indicates the proportion of correct predictions in all positive samples, as follows:

$$precision = \frac{TP}{TP + FP}; \tag{14}$$

$$recall = \frac{TP}{TP + FN}, \tag{15}$$

where TP represents the number of samples that are actually positive and are classified as positive, false positive (FP) represents the number of samples that are incorrectly classified as positive, i.e., the number of samples that are actually negative but are classified as positive, and false negative (FN) represents the number of samples that are actually positive but are classified as negative.

Based on *precision* and *recall*, the *F1 score* and mean average precision (*mAP*) were used as the main evaluation metrics in the experiments. These metrics are defined as follows:

$$F1 = \frac{2 \times precision \times recall}{precision + recall}; \tag{16}$$

$$mAP = \frac{\sum_{i=1}^N AP_i}{N}, \tag{17}$$

where  $AP_i$  denotes the average precision of class  $i$  and  $N$  denotes the total number of classes. The average precision (*AP*) corresponds to the area under the precision-recall curve, i.e.:

$$AP = \int_0^1 p(r)dr \tag{18}$$

where  $p(r)$  denotes the precision function of *recall* ( $r$ ).

In the conducted experiments, the precision-recall curves were first created for each of the compared models, for each class of objects in the corresponding data set used, based on the obtained values of *recall* and *precision*. Then, these curves were used to calculate the *AP* of each model for each class of objects, separately for each experiment, based on (18). Finally, in order to compare the overall target object detection



**TABLE 3.** Average precision results (%) of Gliding Vertex on NWPU VHR-10 data set.

Object class	Experi	Experi	Experi	Experi	Experi
	ment 1	ment 2	ment 3	ment 4	ment 5
Airplane	99.39	100.00	97.96	99.50	100.00
Baseball diamond	94.82	93.73	91.80	94.55	97.78
Basketball court	78.95	81.85	86.38	83.21	64.28
Bridge	66.49	75.30	60.63	64.96	62.24
Ground track field	95.59	99.89	96.86	99.09	100.00
Harbor	86.36	88.39	83.10	95.70	83.96
Ship	95.07	83.22	88.41	91.33	93.39
Storage tank	89.86	68.28	92.48	95.85	97.94
Tennis court	75.77	82.46	72.94	79.13	79.29
Vehicle	81.41	86.44	81.67	88.76	94.03
<b><i>mAP</i></b>	<b>86.37</b>	<b>85.96</b>	<b>85.22</b>	<b>89.21</b>	<b>87.29</b>

**TABLE 4.** Average precision results (%) of Oriented R-CNN on NWPU VHR-10 data set.

Object class	Experi	Experi	Experi	Experi	Experi
	ment 1	ment 2	ment 3	ment 4	ment 5
Airplane	99.87	100.00	99.93	100.00	100.00
Baseball diamond	89.96	90.52	89.13	89.86	98.93
Basketball court	72.73	81.45	80.47	81.82	71.94
Bridge	70.83	70.42	72.00	71.64	67.08
Ground track field	99.72	99.72	99.45	99.72	99.15
Harbor	81.82	81.82	81.82	90.66	81.55
Ship	89.34	89.78	89.72	89.93	90.32
Storage tank	90.47	90.34	99.00	90.55	98.93
Tennis court	81.50	90.12	81.82	81.67	81.82
Vehicle	79.67	87.83	87.54	88.48	87.70
<b><i>mAP</i></b>	<b>85.59</b>	<b>88.20</b>	<b>88.09</b>	<b>88.43</b>	<b>87.74</b>

**TABLE 5.** Average precision results (%) of EfficientDet on NWPU VHR-10 data set.

Object class	Experi	Experi	Experi	Experi	Experi
	ment 1	ment 2	ment 3	ment 4	ment 5
Airplane	99.74	99.75	99.56	99.09	98.90
Baseball diamond	97.30	97.18	97.29	97.15	98.21
Basketball court	95.20	93.69	90.80	92.34	94.49
Bridge	77.54	82.54	82.01	80.38	81.33
Ground track field	99.24	95.22	99.60	97.72	98.30
Harbor	86.08	83.74	94.35	94.32	88.76
Ship	88.58	83.75	86.97	85.93	87.92
Storage tank	70.78	81.78	77.48	79.08	72.79
Tennis court	95.54	95.59	95.59	96.08	93.42
Vehicle	73.56	67.17	70.81	68.01	71.86
<b><i>mAP</i></b>	<b>88.36</b>	<b>88.04</b>	<b>89.45</b>	<b>89.01</b>	<b>88.60</b>

performance of the models across all classes of objects, the *mAP* values were calculated, based on (17), separately for each experiment, and then averaged to obtain the final *mAP* result for the particular model, shown in Tables 8 and 15

**TABLE 6.** Average precision results (%) of YOLOv4 on NWPU VHR-10 data set.

Object class	Experi	Experi	Experi	Experi	Experi
	ment 1	ment 2	ment 3	ment 4	ment 5
Airplane	99.40	99.70	99.30	99.12	99.58
Baseball diamond	95.27	97.50	95.85	98.07	97.73
Basketball court	75.22	78.51	89.65	88.52	81.22
Bridge	71.95	69.12	72.71	61.44	78.37
Ground track field	94.17	98.86	99.41	99.49	98.49
Harbor	93.12	97.37	96.44	95.05	90.99
Ship	92.63	90.61	91.79	91.69	94.12
Storage tank	99.81	99.15	99.85	98.93	99.30
Tennis court	98.30	99.48	98.02	99.84	100.00
Vehicle	87.80	85.19	89.03	86.73	90.09
<b><i>mAP</i></b>	<b>90.77</b>	<b>91.55</b>	<b>93.21</b>	<b>91.89</b>	<b>92.99</b>

**TABLE 7.** Average precision results (%) of YOLOv4\_CE on NWPU VHR-10 data set.

Object class	Experi	Experi	Experi	Experi	Experi
	ment 1	ment 2	ment 3	ment 4	ment 5
Airplane	100.00	99.98	99.98	100.00	99.99
Baseball diamond	96.63	94.20	96.56	95.84	96.92
Basketball court	94.64	95.65	95.01	97.46	96.05
Bridge	87.69	83.26	79.56	86.11	84.14
Ground track field	99.50	99.40	99.80	99.50	99.80
Harbor	97.34	96.29	93.80	90.16	97.25
Ship	91.48	90.68	91.02	90.89	91.39
Storage tank	94.53	99.01	98.79	99.11	99.09
Tennis court	99.19	98.50	98.76	99.41	98.25
Vehicle	91.83	94.33	91.48	92.55	88.68
<b><i>mAP</i></b>	<b>95.28</b>	<b>95.13</b>	<b>94.48</b>	<b>95.10</b>	<b>95.16</b>

**TABLE 8.** *mAP* and *F1 score* results of compared models on NWPU VHR-10 data set.

Model	<i>F1 score</i>	<i>mAP</i> (%)
Faster R-CNN	0.733	83.18
Gliding Vertex	0.873	86.81
Oriented R-CNN	0.931	87.61
EfficientDet	0.844	88.69
YOLOv4	0.892	92.08
<b>YOLOv4_CE</b>	<b>0.933</b>	<b>95.03</b>

below. Then the other metric, *F1 score*, was used, separately for each model in each of the five experiments, and the corresponding values were averaged to obtain the final *F1 score* result for each model, as summarized in Tables 8 and 15.

### C. RESULTS

#### 1) NWPU VHR-10 DATA SET

The calculated *AP* and *mAP* values of each model for each class of objects in each of the five experiments, conducted on this data set, are presented in Tables 2-7. Based on these, the

**TABLE 9.** Average precision results (%) of Faster R-CNN on RSOD data set.

Object class	Experiment	Experiment	Experiment	Experiment	Experiment
	1	2	3	4	5
Aircraft	63.60	58.89	58.94	60.22	59.23
Oil tank	95.91	94.63	94.85	95.03	96.01
Overpass	82.46	88.12	85.83	86.78	93.12
Playground	98.83	100.00	98.83	99.00	99.68
<b><i>mAP</i></b>	<b>85.20</b>	<b>85.41</b>	<b>84.61</b>	<b>85.26</b>	<b>87.01</b>

**TABLE 10.** Average precision results (%) of Gliding Vertex on RSOD data set.

Object class	Experiment	Experiment	Experiment	Experiment	Experiment
	1	2	3	4	5
Aircraft	93.30	83.74	91.51	91.77	87.95
Oil tank	95.52	94.24	96.54	94.46	97.87
Overpass	60.06	66.86	73.42	68.24	92.41
Playground	100.00	95.45	92.59	100.00	96.23
<b><i>mAP</i></b>	<b>87.22</b>	<b>85.07</b>	<b>88.52</b>	<b>88.62</b>	<b>93.62</b>

**TABLE 11.** Average precision results (%) of Oriented R-CNN on RSOD data set.

Object class	Experiment	Experiment	Experiment	Experiment	Experiment
	1	2	3	4	5
Aircraft	90.32	90.12	90.19	90.06	89.95
Oil tank	90.89	90.86	90.89	90.86	90.86
Overpass	80.74	90.04	90.69	90.05	80.85
Playground	100.00	99.64	90.91	100.00	99.64
<b><i>mAP</i></b>	<b>90.49</b>	<b>92.67</b>	<b>90.67</b>	<b>92.74</b>	<b>90.33</b>

**TABLE 12.** Average precision results (%) of EfficientDet on RSOD data set.

Object class	Experiment	Experiment	Experiment	Experiment	Experiment
	1	2	3	4	5
Aircraft	70.32	70.22	69.44	69.72	70.45
Oil tank	97.51	97.99	98.06	97.81	98.26
Overpass	90.17	88.59	90.44	86.11	90.87
Playground	100.00	100.00	100.00	100.00	100.00
<b><i>mAP</i></b>	<b>89.50</b>	<b>89.20</b>	<b>89.49</b>	<b>88.41</b>	<b>89.90</b>

averaged *mAP* value was calculated for each model, as shown in Table 8. The obtained results confirm that the proposed YOLOv4\_CE model outperforms, in terms of *mAP*, all five state-of-the-art models on this data set. More specifically, Faster R-CNN, Gliding Vertex, Oriented R-CNN, EfficientDet, and YOLOv4 are outperformed by 11.85, 8.22, 7.42, 6.34, and 2.95 points, respectively.

Then, the *F1 score* values were calculated in each experiment for each model and then averaged to produce the final results presented in Table 8. These results confirm that the proposed YOLOv4\_CE model outperforms all

**TABLE 13.** Average precision results (%) of YOLOv4 on RSOD data set.

Object class	Experiment	Experiment	Experiment	Experiment	Experiment
	1	2	3	4	5
Aircraft	91.79	92.03	91.60	91.65	92.35
Oil tank	97.26	98.70	98.35	97.87	97.54
Overpass	78.76	75.14	82.12	74.28	82.74
Playground	100.00	100.00	99.68	99.68	100.00
<b><i>mAP</i></b>	<b>91.95</b>	<b>91.47</b>	<b>92.94</b>	<b>90.87</b>	<b>93.16</b>

**TABLE 14.** Average precision results (%) of YOLOv4\_CE on RSOD data set.

Object class	Experiment	Experiment	Experiment	Experiment	Experiment
	1	2	3	4	5
Aircraft	94.52	92.87	94.08	93.47	93.60
Oil tank	98.29	97.90	98.02	97.71	98.03
Overpass	94.40	94.90	95.75	83.92	94.45
Playground	100.00	100.00	95.83	100.00	100.00
<b><i>mAP</i></b>	<b>96.80</b>	<b>96.42</b>	<b>95.92</b>	<b>93.78</b>	<b>96.52</b>

**TABLE 15.** *mAP* and *F1 score* results of compared models on RSOD data set.

Model	<i>F1 score</i>	<i>mAP</i> (%)
Faster R-CNN	0.767	85.50
Gliding Vertex	0.893	88.61
Oriented R-CNN	0.904	91.38
EfficientDet	0.883	89.30
YOLOv4	0.909	92.08
<b>YOLOv4_CE</b>	<b>0.937</b>	<b>95.89</b>

five state-of-the-art models on this evaluation metric too. More specifically, Faster R-CNN, Gliding Vertex, Oriented R-CNN, EfficientDet, and YOLOv4 are outperformed by 0.200, 0.060, 0.002, 0.089, and 0.041 points, respectively.

The most challenging for target object detection proved to be the images with complex background (bridge and basketball court classes) and the images containing intensive small targets (vehicle and harbor classes).

## 2) RSOD DATA SET

The calculated *AP* and *mAP* values of each model for each class of objects in each of the five experiments, conducted on this data set, are presented in Tables 8-14. Based on these, the averaged *mAP* values were calculated, as shown in Table 15. The obtained results confirm that the proposed YOLOv4\_CE model outperforms, in terms of *mAP*, all five state-of-the-art models on this data set too. More specifically, Faster R-CNN, Gliding Vertex, Oriented R-CNN, EfficientDet, and YOLOv4 are outperformed by a similar degree as on the other data set, namely by 10.39, 7.28, 4.51, 6.59, and 3.81 points, respectively.

Then, the *F1 score* values were calculated in each experiment for each model and then averaged to produce the final

results presented in Table 15. These results also confirm that the proposed YOLOv4\_CE model outperforms all five state-of-the-art models, based on this evaluation metric, on this data set too. More specifically, Faster R-CNN, Gliding Vertex, Oriented R-CNN, EfficientDet, and YOLOv4 are outperformed by a similar degree as on the other data set, namely by 0.170, 0.044, 0.033, 0.054, and 0.028 points, respectively.

The most challenging for target object detection again proved to be the images with complex background (overpass class) and the images containing intensive small targets (aircraft class).

## VI. CONCLUSION

This paper has proposed a more accurate target object detection model, called YOLOv4\_CE, based on the classical YOLOv4 model with additional improvements. One of the ideas, utilized by YOLOv4\_CE, was to make the model extract features more effectively and by this to lessen the computation of redundant information at the feature layer and reduce the size of the model itself. This was achieved by replacing the original feature extraction backbone (i.e., CSPDarknet53) of YOLOv4 with ConvNeXt-S [17]. In addition, in order to increase the receptive field and allow the proposed model to pay more attention to important parts of the processed images, the coordinate attention (CA) mechanism [18] was integrated into YOLOv4. Moreover, in the proposed model, the original loss function of YOLOv4, namely the CIoU loss [19], was replaced with the EIou loss [20] in order to achieve faster convergence of the model and improve its regression precision. The incorporation of these improvements into YOLOv4 resulted in overall better target object detection. This was confirmed by a series of experiments conducted for evaluating and comparing the target object detection performance of the proposed model to that of the original YOLOv4 model and four other state-of-the-art models, namely Faster R-CNN, Gliding Vertex, Oriented R-CNN, and EfficientDet, based on two open data sets – NWPU VHR-10 and RSOD. The obtained results clearly demonstrated that the proposed YOLOv4\_CE model outperforms these five models, in terms of the mean average precision (mAP) and *F1 score*, on both data sets.

The proposed YOLOv4\_CE model is very suitable for detecting target objects in remote sensing images. Due to the replacement of the feature extraction module with a more complex module for the network, in the future we plan to introduce some specially designed lightweight modules into the model in order to increase its operational speed.

## REFERENCES

- [1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, pp. 1–14, Sep. 2015.
- [2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
- [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.*, Jun. 2014, pp. 580–587.
- [5] 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, Jul. 2015.
- [6] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Dec. 2015, pp. 1440–1448.
- [7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [8] M. Haris and A. Glowacz, "Road object detection: A comparative study of deep learning-based algorithms," *Electronics*, vol. 10, no. 16, p. 1932, Aug. 2021.
- [9] Y. Xu, M. Fu, Q. Wang, Y. Wang, K. Chen, G.-S. Xia, and X. Bai, "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1452–1459, Apr. 2020.
- [10] J. Han, J. Ding, N. Xue, and G.-S. Xia, "ReDet: A rotation-equivariant detector for aerial object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2786–2795.
- [11] G. Zhang, S. Lu, and W. Zhang, "CAD-Net: A context-aware detection network for objects in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10015–10024, Aug. 2019.
- [12] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3520–3529.
- [13] 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. (CVPR)*, Jun. 2016, pp. 779–788.
- [14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 21–37.
- [15] W. Kong, J. Hong, M. Jia, J. Yao, W. Cong, H. Hu, and H. Zhang, "YOLOv3-DPPIN: A dual-path feature fusion neural network for robust real-time sonar target detection," *IEEE Sensors J.*, vol. 20, no. 7, pp. 3745–3756, Apr. 2020, doi: [10.1109/JSEN.2019.2960796](https://doi.org/10.1109/JSEN.2019.2960796).
- [16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [17] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," 2022, *arXiv:2201.03545*.
- [18] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13713–13722.
- [19] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12993–13000.
- [20] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," 2021, *arXiv:2101.08158*.
- [21] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
- [22] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
- [23] B.-Y. Sun, X.-M. Zhang, J. Li, and X.-M. Mao, "Feature fusion using locally linear embedding for classification," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 163–168, Jan. 2009.
- [24] Z. Baojun, Z. Boya, T. Linbo, W. Wenzheng, and W. Chen, "Multi-scale object detection by top-down and bottom-up feature pyramid network," *J. Syst. Eng. Electron.*, vol. 30, no. 1, pp. 1–12, 2019.
- [25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. ICCV*, Venice, Italy, 2017, pp. 2961–2969.
- [26] G. Gkioxari, J. Johnson, and J. Malik, "Mesh R-CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9785–9795.
- [27] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.
- [28] J. Nelson and J. Solawetz. (2020). *Responding to the Controversy About YOLOv5*. [Online]. Available: <https://blog.roboflow.com/yolov4-versus-yolov5/>

- [29] C.-Y. Wang, H.-Y. Mark 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 (CVPRW)*, Jun. 2020, pp. 390–391.
- [30] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.
- [31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [32] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*.
- [33] Y. Wang, L. Wang, H. Wang, and P. Li, "Information-compensated down-sampling for image super-resolution," *IEEE Signal Process. Lett.*, vol. 25, no. 5, pp. 685–689, May 2018.
- [34] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
- [35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.
- [36] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
- [37] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
- [38] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with Gaussian error linear units," 2016, *arXiv:1606.08415*.
- [39] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
- [40] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.
- [41] A. Körez, N. Barışçı, A. Çetin, and U. Ergün, "Weighted ensemble object detection with optimized coefficients for remote sensing images," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 6, p. 370, Jun. 2020.



**HAIYANG ZHANG** received the B.S. degree from Jilin University, China, in 2013, and the Ph.D. degree from the University of Limerick, Ireland, in 2018. She is currently a Lecturer with Xi'an Jiaotong-Liverpool University, Suzhou, China. Her current research interests include recommender systems, data mining, collaborative filtering, and natural language processing (NLP).



**CHENXU DAI** received the B.S. and M.S. degrees from the North China University of Science and Technology. She is currently working as a Research Associate with the North China University of Science and Technology. Her current research interests include neural networks, intelligent optimization algorithms, and deep learning.



**LI ZHAO** received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1997 and 2002, respectively. He is currently working as an Associate Professor with the Research Institute of Information Technology, Tsinghua University. His current research interests include mobile computing, the Internet of Things (IoT), e-health systems, intelligent transportation systems (ITS), home networking, machine learning, and digital multimedia.



**ZHANLIN JI** (Member, IEEE) received the M.Eng. degree from Dublin City University, Ireland, in 2006, and the Ph.D. degree from the University of Limerick, Ireland, in 2010.

He is currently a Professor with the North China University of Science and Technology, China, and an Associate Researcher with the Telecommunications Research Centre (TRC), University of Limerick. He has authored/coauthored more than 100 research papers in refereed journals and conferences.

His research interests include ubiquitous consumer wireless world (UCWW), the Internet of Things (IoT), cloud computing, big data management, and data mining.



**IVAN GANCHEV** (Senior Member, IEEE) received the Engineering (*summa cum laude*) and Ph.D. degrees from the Saint-Petersburg University of Telecommunications, in 1989 and 1995, respectively. He is an International Telecommunications Union (ITU-T) Invited Expert and an Institution of Engineering and Technology (IET) Invited Lecturer, currently associated with the University of Limerick, Ireland, the University of Plovdiv "Paisii Hilendarski," and IMI-BAS, Bulgaria.

He was involved in more than 40 international and national research projects. He has served on the TPC of more than 350 prestigious international conferences/symposia/workshops, and has authored/coauthored one monographic book, three textbooks, four edited books, and more than 300 research papers in refereed international journals, books, and conference proceedings. He is on the editorial board of and has served as a guest editor for multiple renowned international journals.

...



**XUNKAI YANG** was born in 1998. He received the B.S. degree from Zhoukou Normal University, in 2020. He is currently pursuing the master's degree with the North China University of Science and Technology. His research interests include machine vision and graphic image processing.



**JINGYI ZHAO** was born in 1998. She received the B.S. degree from the North China University of Science and Technology, in 2019, where she is currently pursuing the master's degree. Her research interests include machine vision and graphic image processing.