IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 June 2023, accepted 25 June 2023, date of publication 28 June 2023, date of current version 3 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3290480

APPLIED RESEARCH

CA-YOLO: Model Optimization for Remote Sensing Image Object Detection

LINGYUN SHEN[1], **BAIHE LANG**[2], **AND ZHENGXUN SONG**[2,3]

[1]Department of Electronic Engineering, Taiyuan Institute of Technology, Taiyuan 030008, China
[2]School of Electronics and Information Engineering, Changchun University of Science and Technology, Changchun 130022, China
[3]Overseas Expertise Introduction Project for Discipline Innovation No. D17017, Changchun University of Science and Technology, Changchun 130022, China

Corresponding authors: Lingyun Shen (shenshly@163.com) and Baihe Lang (langbh@gmail.com)

This work was supported in part by the Shanxi Province's Introduced Talent Research Funding Projects, China.

ABSTRACT The CA-YOLO (Coordinate Attention-YOLO) model has been optimized for object detection in complex remote sensing images, addressing key issues faced by algorithms that detect multiple objects. These issues include weak multi-scale feature learning capabilities and the challenging trade-off between detection accuracy and model parameter complexity. The CA-YOLO model, built on the framework of YOLOv5, incorporates a lightweight coordinate attention module in the shallow layer to improve detailed feature extraction and suppress redundant information interference. Additionally, a spatial pyramid pooling-fast with a tandem construction module is implemented in the deeper layer. The model employs a stochastic pooling strategy to fuse multi-scale key feature information from low-level to high-level layers, reducing the number of model parameters while improving inference speed. We optimized the anchor box mechanism and modified loss function to improve the ability of the model to detect objects of different sizes and scales. Results show that the CA-YOLO model outperforms the original YOLO in terms of multi-object detection accuracy, with an average mAP@0.5 accuracy improvement of 4.8% and mAP@0.5:0.95 accuracy improvement of 3.8%. Additionally, the CA-YOLO model demonstrates exceptional inference speed, averaging 125 fps, which reinforces its superiority in detection accuracy, generalization ability, and overall efficiency. Notably, these improvements were achieved while maintaining the same number of parameters and complexity as other models, making the CA-YOLO model an exceptional choice for various applications.

INDEX TERMS Object detection, attention mechanism, coordinate attention, SPPF, SIoU loss.

I. INTRODUCTION

Remote sensing images and their interpretation have vast applications across various fields, including intelligent transportation, urban planning, intelligent agriculture, disaster rescue, environmental monitoring, military operations, and public security [1]. The foundation of intelligent interpretation lies in the identification of image objects, which involves achieving fundamental tasks such as object localization and classification. This approach enables swift and precise identification, localization, and tracking of multiple objects, making it a valuable tool for a wide range of applications.

In 2012, AlexNet, a convolutional neural network (CNN) based on a convolutional kernel, dropout fully connected approach, and ReLU activation, emerged victorious in the ImageNet large-scale image recognition competition due to its exceptional feature representation and classification ability. Since then, CNN-based detection and classification methods have received considerable attention from scholars. For object region proposals, Girshick employed R-CNN and SVM for object classification and localization, resulting in significant improvements in object detection performance [2]. A notable advancement in image processing is the study of CNN-based object detection, with researchers focusing on enhancing feature extraction to improve detection and classification accuracy [3].

Object detection approaches using CNN can be classified into two primary methods: the two-stage method and the

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

single-stage method, based on the categories of classification and regression. In the two-stage method, pre-selection bounding boxes identify the target area, followed by classification for regression. The R-CNN series is a representative algorithm for this approach. However, the efficiency of R-CNN is hampered by the need to repeatedly calculate numerous region proposals, which limits its performance compared to more recent object detection techniques. Various methods have been developed to improve object detection efficiency, such as SPPnet [4], [29], R-CNN and its enhanced models [5], [6], and other approaches [7], [8]. Although accurate, the two-stage calculation lowers the inference speed, making real-time tasks suboptimal. While regression-based algorithms are faster, region-proposal-based methods tend to be more accurate. The CNN framework is widely recognized as an important tool for object detection tasks and is commonly used as a network backbone by many object detection technologies.

The single-stage method combines classification and location regression in a single step, which includes approaches such as SSD [9], RetinaNet [10], YOLO [11], [12], [13], etc. While the inference speed of this single-stage method is faster than earlier methods, it has slightly lower accuracy.

Research has explored the application of regression-based algorithms in remote sensing image object detection tasks. Although these approaches are faster than region-proposal-based methods, they typically have inferior accuracy. While CNN architecture is widely recognized as an important tool for object detection, its accuracy and inference speed may be compromised for remote sensing images because of their inherent complexity, such as their large size, variable object sizes, diverse distribution, and high proportion of small objects.

Recent studies have attempted to address these challenges by adapting regression-based object detection methods. Research efforts have also focused on enhancing the effectiveness of CNN-based object detection techniques through feature improvement [14], [14], [16], [17], contextual information fusion [18], [19], [20], hard negative mining [10], [21], the modeling of object deformations [22], [23], [24], and other approaches.

Qu et al. [25] proposed an auxiliary network to enhance object recognition in remote-sensing images using the YOLOv3 model. The CBAM module was integrated with the backbone to improve the network performance and prevent the loss of important information during training.

Li et al. [26] proposed the YOLOSR-IST model, which builds upon the YOLOv5 approach by introducing coordinate attention in the feature fusion process and integrating high-resolution maps.

The YOLO-extract method, built upon the YOLOv5 approach by combining the dilated convolution structure, was developed by Liu et al [27]. The model's ability to extract feature and location information for objects of varying scales was improved, resulting in reduced computation and faster convergence speed.

Huang et al. introduce enhancements that aim to improve the accuracy and stability of the YOLOv2 object detection algorithm [28]. An upgraded dense connection structure is suggested to bolster feature extraction and mitigate gradient vanishing. An enhanced spatial pyramid pooling structure is proposed, which captures more context information from multiple scale features.

The literature suggests a potential solution to the issue of inadequate feature extraction in YOLOv5 through the introduction of a remote sensing image object detection method with a coordinated attention-YOLO (CA-YOLO) architecture. This methodology effectively addresses the need for enhanced multi-scale key feature information and designs the model pooling structure to improve inference speed while keeping the model lightweight and balancing multiple object detection accuracy and real-time inference.

To further improve this methodology, three key steps were taken. Firstly, coordinated attention was introduced to the shallow layer of the network to establish pixel-level contextual information association and strengthen multi-scale feature fusion. Additionally, a Spatial Pyramid Pooling-Fast (SPPF) was constructed with a tandem construction module to decrease the parameter count and enhance inference speed by integrating crucial features. Finally, to increase the efficacy and generalizability of object detection, the prediction box position regression loss function from the original model's CIoU_loss was replaced with EIoU_loss.

II. IMPROVEMENT FOR CA-YOLO MODEL

The proposed CA-YOLO is an improved model of the single-stage algorithm based on the backbone architecture of YOLOv5. The YOLOv5 network module comprises a backbone responsible for feature extraction and a head responsible for fusing these features and predicting the results via the activation function. The backbone architecture of the network includes three key components: the Convolutional layer, the C3 layer, and the SPPF module [29].

A. CONVOLUTIONAL ATTENTION MODULE

While the channel attention mechanism significantly improves model performance, it often neglects important location information required to generate spatially selective attention maps. To address this issue, the lightweight coordinate attention technique is introduced into the model by incorporating location information into channel attention [30]. This technique aims to improve the model's capability to focus on spatial features, thereby enhancing object localization accuracy, capturing spatial dependencies more precisely, and resulting in better object detection. The implementation of lightweight coordinate attention can, therefore, improve accuracy and efficiency performance.

The coordinated attention mechanism splits the attention process into two one-dimensional feature encoding processes that gather features along dual spatial dimensions. This approach creates a map of coordinate-aware attention by identifying the operational coordinates, enabling the capture of long-range dependencies. The resulting feature maps are encoded as attentional maps that are both direction-aware and location-aware, enhancing the representation of objects of interest while avoiding significant computational overhead. This technique also increases the receptive field of the model.

By decomposing channel attention into two parallel one-dimensional features, coordinated attention addresses the loss of location information resulting from global pooling. The process of encoding integrates spatial coordinate information into the feature vector of the channel attention generated. The flexible and minimally demanding coordinate attention module is readily integrated into the YOLO network architecture and surpasses other modules in object identification and semantic segmentation tasks, while also streamlining ImageNet classification.

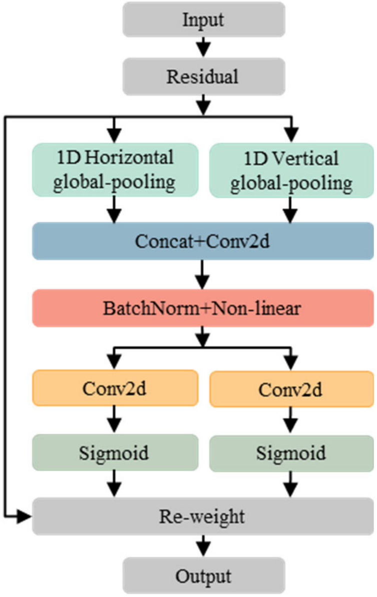

FIGURE 1. Schematic diagram of Coordinate Attention Module (CAM).

1) COORDINATE FEATURE EMBEDDING

The technique of using global pooling to encode spatial information across an intermediate feature tensor is commonly employed in channel attention. This approach summarizes the spatial information of the entire feature map into a single vector, facilitating efficient processing of feature maps in deep learning models. However, it may also lead to the loss of positional information, which is crucial for vision tasks requiring accurate location information. Therefore, it is essential to be mindful of this limitation when employing this technique.

Building interdependence between the channels can increase the sensitivity of the model to information-rich channels that contribute more to the final classification decision. The Squeeze-and-Excitation channel attention block can be decomposed into two steps: squeeze and excitation. These steps are designed for global information embedding and adaptive recalibration of channel relationships, respectively. Given the input X, the squeeze step for the c-th channel can be formulated as Equation (1).

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j) \tag{1}$$

where, $X = [x_1, x_2, \ldots, x_C]$ is intermediate feature tensor. z_c is the output associated with the c-th channel

Capturing the spatial structure in vision tasks requires preserving the positional information of features. However, channel attention techniques make it challenging to maintain this information while compressing global spatial information into channel descriptors for pooling.

To overcome this issue and accurately capture the attention block's location information, the global pooling can be decomposed by separate pooling operations, with the preservation of positional information while still capturing the spatial essence of the entire feature map. Subsequently, convolution kernels of size (H, 1) and (1, W) are obtained along the horizontal and vertical directions of the input feature map, respectively.

These convolution kernels can be understood as 2D arrays, where H and W represent the size of the kernel along the two directions, respectively. During the convolution operation, the kernels slide along the two directions of the input feature map, to extract features along the two axes and generate the output feature map.

Specifically, provided with the input x, two spatial ranges of pooling kernels (H, 1) and (1, W) are utilized to encode the horizontal and vertical coordinates of each channel, respectively. The outputs of the c-channel can be denoted as

$$\begin{cases} z_c^h(h) = \frac{1}{W} \sum\limits_{0 \le i < W} x_c(h, i) \\ z_c^w(w) = \frac{1}{H} \sum\limits_{0 \le j < H} x_c(j, w) \end{cases} \tag{2}$$

The function mapping in question aggregates features along both spatial directions, resulting in a pair of direction-aware feature mappings.

This approach enhances the accurate localization of regions of interest by capturing spatial information. By incorporating this direction-aware feature mapping, the network can better understand the spatial relationships between different regions of the input, leading to improved performance in vision tasks.

2) COORDINATE ATTENTION GENERATION

To obtain the map of attention, **f**, concatenate the outputs of the two-direction pooling and pass them to the shared 1×1 convolutional transform function F_1.

$$\mathbf{f} = \delta(F_1([\mathbf{z}^h, \mathbf{z}^w])), \quad \mathbf{f} \in \mathbb{R}^{C/r \times (H+W)} \tag{3}$$

where **z** denotes the concatenation operation along the spatial dimension. $\delta(\cdot)$ is a nonlinear activation function. r is the reduction rate that controls the block size.

To obtain the feature maps along the horizontal and vertical axes, $\mathbf{f}$ is decomposed into independent tensors, as shown in Equation(4), the other two 1×1 convolution transforms, F_h and F_w, are employed to transform $\mathbf{f}^h$ $\left(\mathbf{f}^h \in \mathbb{R}^{C/r \times H}\right)$ and $\mathbf{f}^w$ $\left(\mathbf{f}^w \in \mathbb{R}^{C/r \times W}\right)$, respectively, into a tensor consistent with the number of channels in the input X.

$$\begin{cases} \mathbf{g}^h = \delta(F_h(\mathbf{f}^h)) \\ \mathbf{g}^w = \delta(F_w(\mathbf{f}^w)) \end{cases} \tag{4}$$

The number of channels is typically decreased using an appropriate reduction rate r ($r = 4, 8, 16, 32$, etc.) to reduce the complexity of the model overhead. Following 1×1 convolution, the attentional weight data are subsequently computed using sigmoid $\delta(\cdot)$.

3) COORDINATE ATTENTION FEATURES

The input feature is multiplied by the weights to generate a coordinate attention map, which highlights the most informative regions in the input. This attention mechanism effectively captures the spatial relationships between different regions of the input, enabling the network to focus on the most relevant features. The outputs g^h and g^w of Equation (4) are then expanded and used as attention weights, respectively.

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{5}$$

B. SPPF WITH A TANDEM CONSTRUCTION

Multi-scale representation methods can improve the effectiveness of remote sensing image detection by capturing object distribution characteristics. To this end, we designed a spatial pyramid pooling-fast (SPPF) module with a tandem construction in the backbone network. This module adaptively adjusts the size of the feature map vector to a fixed value, which helps to counteract the distortion caused by resizing procedures in different regions of the image. By avoiding repeated feature extraction, this approach not only improves network accuracy but also reduces computational costs by avoiding repeated feature extraction.

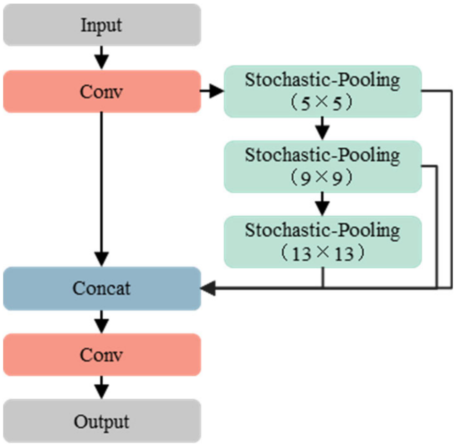

FIGURE 2. Spatial pyramid pooling-fast with a tandem construction.

To achieve fast computation of the spatial pyramid pooling (SPP), a pooling layer with a tandem construction is implemented. This structure is designed to use the output of the previous pooling layer as the basis for each subsequent pooling layer, thereby reducing the number of repetitive operations and improving network efficiency. By reusing the output of each layer, the network can avoid redundant computations and focus on extracting higher-level features from the input.

The SPPF with tandem construction reduces computation time by approximately 50% compared to the parallel structure, as shown in Figure 2. For stochastic pooling in the pooling approach, a random selection of elements between average pooling and maximum pooling is carried out according to probability, improving the generalization ability and preventing overfitting [31].

Although average pooling and maximum pooling are widely used techniques for pooling in computer vision, they tend to lose crucial feature information when the difference between features is not readily apparent. To address this issue and improve the network's robustness, the stochastic pooling technique has been proposed.

This approach balances the trade-off between average and maximum pooling, mimicking average pooling in typical circumstances while adhering to the guidelines for local information computation in maximum pooling. The defining graph for stochastic pooling is illustrated in Figure 3.

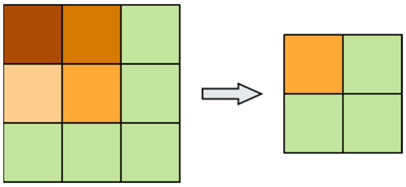

FIGURE 3. The defining graph for the stochastic pooling.

The probability value of the feature is determined by dividing each feature's value by this statistical total. Let f_i be a feature with probability p_i given by:

$$p_i = \frac{f_i}{\sum_{n \in W_s} f_n} \tag{6}$$

After that, stochastic pooling is achieved by applying random sampling following these probability values as in Equation (7).

$$s_m = f_r, r \sim P(p_1, \ldots, p_{|W_m|}) \tag{7}$$

here, W_m stands for the sampling window size, f_r for the sampled feature value, and r for the value chosen at random by p_i.

C. ANCHOR BOX MECHANISM

In the training phase of object detection networks, the anchor box plays a crucial role in linking the ground truth box and the predicted bounding box. It is a pre-defined bounding box with a fixed aspect ratio and scale that is placed at different locations on the image. During training, the network learns to predict the offset between the anchor box and the ground truth box.

To improve detection accuracy for each grid of the input image, the model uses different size scales to match the three anchor boxes, as typically only one to three objects exist at the center points of the spatial points where objects are located. If an object is present in the input image grid, each feature map is a positive sample of the object, and three to nine anchor boxes are matched with the predicted bounding box. This anchor box prediction mechanism addresses the issue of multi-object overlap detection, as opposed to the sliding fixed window object detection method.

The optimization of anchor boxes aims to increase the number of effective positive samples predicted by ground truth boxes located in one or more feature layers. This reduces the model convergence time during training and is advantageous for model performance.

Equation (8) demonstrates how the aspect ratio of predicted bounding boxes is calculated concerning the 9 anchor boxes in the feature layer during model training. This calculation utilizes the shape-matching principle. Let the ratio of the width and height of the predicted bounding box to the anchor box be denoted as:

$$r_{\max} = \max\left(\max\left(\frac{w_p}{w_a}, \frac{w_a}{w_p}\right), \max\left(\frac{h_p}{h_a}, \frac{h_a}{h_p}\right)\right) \tag{8}$$

where, w_p and h_p denote the width and height of the predicted bounding box. w_a and h_a denote the width and height of the anchor box, respectively.

If the aspect ratio is below the threshold t_{anchor} (according to the default hyperparameter list provided by YOLOv5, the optimal value for this hyperparameter is 4.0), as shown in Equation (9), the predicted bounding box is regarded as a positive sample. Otherwise, it is discarded as background. When the ground truth box matches with three anchor boxes of varying sizes, all matching anchor boxes can be utilized to generate the predicted bounding box.

To increase the number of positive samples in object detection, the anchor mechanism is used, which generates a set of pre-defined anchors with various sizes and aspect ratios and places them at different positions on the input image. Additionally, the grid surrounding the predicted bounding box is also used as the prediction grid, further increasing the number of positive samples.

$$r_{\max} < t_{anchor}, \quad t_{anchor} = 4.0 \tag{9}$$

The optimization of anchor box size is achieved through a combination of K-Means and a genetic algorithm. This method relies on the statistical laws governing the size of ground truth boxes for sampled objects within the training set of the ROSD dataset for remote sensing images. The algorithm accomplishes this task by converting the Euclidean distance from N-dimensional space into a two-dimensional planar array distance.

Table 1 shows 9 sets of anchor box size parameters that were obtained and assigned by inputting a 640 × 640 image. While anticipating the object's location, a smaller anchor box size is utilized. This is because, as shown in the table, when the downsampling rate decreases, the relative scale of the feature map increases. Consequently, the receptive field shrinks, and vice versa.

TABLE 1. Anchor box parameters of ROSD dataset based on different clustering algorithms.

Anchor box optimization algorithm	Feature map scale	Anchor box size	Anchor boxes number
K-Means	80x80	(10,13)(16,30)(33,23)	80x80x3
	40x40	(30,61)(62,45)(59,119)	40x40x3
	20x20	(116,90)(156,198)(373,326)	20x20x3
K-Means + Genetic	80x80	(13,14)(20,20)(27,27)	80x80x3
	40x40	(36,38)(47,50)(62,68)	40x40x3
	20x20	(79,89)(155,266)(263,282)	20x20x3

D. LOSS FUNCTION

In the model training phase, the weight parameters must be adjusted until convergence is achieved by performing back-propagation through the loss function. Equation (10) shows that the intersection over Union (IoU) ratio is controlled by the percentage of their overlapping area during training.

$$IoU = \frac{S_{gt} \cap S_p}{S_{gt} \cup S_p} \tag{10}$$

here, S_p and S_{gt} represent the area of the predicted bounding box and the ground truth box, respectively. It is explained that the predicted bounding box information includes the position offset of the two boxes. This offset is calculated based on the IoU. During the training process, the model fits the ground truth box with the predicted bounding box and continues to adjust until the difference reaches its lowest possible value. Then the weight parameters of the model are obtained.

1) PREDICTION BOX POSITION REGRESSION LOSS

The YOLOv5 model uses the CIoU_Loss (Complete IoU) to optimize object detection accuracy. However, the loss function may encounter an issue when the predicted and ground truth bounding boxes have equal aspect ratios and their central points align. It's important to note that the model's regression loss function only considers distance, overlap area, and aspect ratio, while neglecting direction mismatch. This can lead to slower convergence rates and decreased model performance. Therefore, incorporating direction information into the regression loss function is essential to improve accuracy, speed up convergence rates, and ultimately enhance the model's performance.

The designed object detection loss function, SIoU_Loss (Scylla IoU), includes an angle cost penalty term [32].

In Figure 4, α is defined as the horizontal angle of the line connecting the center points of two boxes. When α is less than

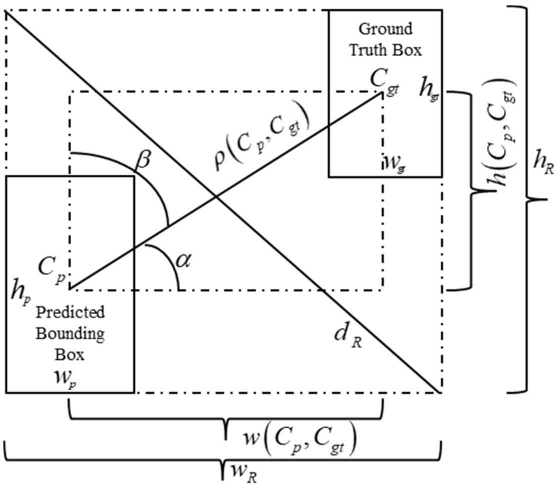

FIGURE 4. **Scheme for calculation of angle cost and distance cost.**

$\pi/4$, the objective is to minimize α. On the other hand, when α is greater than $\pi/4$, the aim is to minimize β, which is equal to $\pi/2$-α.

Angle cost definition:

$$\Lambda = 1 - 2 * sin^2\left(\alpha - \frac{\pi}{4}\right) = sin\,(2\alpha) \tag{11}$$

where $sin(\alpha) = \frac{h(C_p, C_{gt})}{\rho(C_p, C_{gt})}$, the coordinates of the center points of the predicted bounding box and the ground truth box are (x_{C_p}, y_{C_p}) and $(x_{C_{gt}}, y_{C_{gt}})$, respectively.

Where, $\rho\,(C_p, C_{gt}) = \sqrt{(x_{C_p} - x_{C_{gt}})^2 + (y_{C_p} - y_{C_{gt}})^2}$ and $h\,(C_p, C_{gt}) = max\,(y_{C_p}, y_{C_{gt}}) - min\,(y_{C_p}, y_{C_{gt}})$.

The above angle cost is added to the distance cost,

$$\Delta = \sum_{t=x,y} \left(1 - e^{-\gamma \rho_t}\right) \tag{12}$$

where, $\rho_x = \left(\frac{x_{C_p} - x_{C_{gt}}}{w_R}\right)^2$, $\rho_y = \left(\frac{y_{C_p} - y_{C_{gt}}}{h_R}\right)^2$, $\gamma = 2 - \Lambda$.

Let w_R and h_R represent the width and height of the smallest rectangle covering the two boxes, respectively.

Shape cost definition:

$$\Omega = \sum_{t=w,h} \left(1 - e^{-\omega_t}\right)^{\theta} \tag{13}$$

where, $\omega_w = \frac{|w_p - w_{gt}|}{max(w_p, w_{gt})}$, $\omega_h = \frac{|h_p - h_{gt}|}{max(h_p, h_{gt})}$.

The weight of the shape cost is determined by the unique value of θ for the dataset. A value of θ=1 results in quick optimization but limited mobility. The genetic algorithm is utilized to calculate θ for each dataset, and it has been experimentally determined to be approximately 4.

The loss function SIoU_Loss takes the form:

$$box\_loss = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{14}$$

The optimization of the SIoU_Loss function plays a critical role in enhancing the multi-scale feature learning capability, and in turn, improving the training and detection performance of the model. The superiority of the loss function is demonstrated through the experimental training and visual comparison analysis in the next section.

2) BINARY CROSS-ENTROPY LOSS FUNCTION

The object confidence loss Equation (15) and the classification loss Equation (16) are computed by utilizing the PyTorch binary cross-entropy loss function (named BCEWithLogitsLoss function).

$$\begin{aligned} obj\_loss = -w_{n,c}\left[p_c y_{n,c} \cdot \log\sigma\left(x_{n,c}\right) + \left(1 - y_{n,c}\right)\right. \\ \left.\cdot \log\left(1 - \sigma\left(x_{n,c}\right)\right)\right] \end{aligned} \tag{15}$$

$$\begin{aligned} cls\_loss = -\frac{1}{n}\sum_{i}^{n}\left[y_i \cdot \log\sigma\,(x_i) + (1 - y_i)\right. \\ \left.\cdot \log\left(1 - \sigma\,(x_i)\right)\right] \end{aligned} \tag{16}$$

where, $x_{n,c}$ presents the model output corresponding to the nth sample of type C object, which needs to be processed by the sigmoid activation function. $y_{n,c}$ represents categories of model prediction corresponding to the nth sample c-class object. $w_{n,c}$ is the global weight. p_c represents the weight assigned to positive examples of the c-class object.

The category of the i-th sample is denoted as y_i, the predicted probability is denoted as p_i, and $\sigma\,(a) = \frac{1}{1+\exp(-a)}$ is the sigmoid function.

The variable obj_loss describes the outcome of the object confidence loss function, with a smaller value indicating a more precise object detection outcome. Similarly, cls_loss denotes the result of the object category loss function, where a lower value indicates a more accurate object classification.

To obtain the best-predicted bounding box and prevent missed detections caused by object occlusion, the Non-Maximum Suppression (NMS) algorithm is applied during filtering. During the testing phase, the optimal model is trained to perform a forward inference prediction on the test set images.

E. CA-YOLO

Overall, the previous modifications and optimizations have been incorporated into the design of the CA-YOLO network structure, which has significantly improved the performance of the CA-YOLO while increasing the accuracy of the detection system.

The integration of the CA module into the shallow layer allows for more effective feature extraction, while the SPPF module in the deep layer enables the fusion of feature maps with multiple resolutions, leading to a more robust and accurate detection system.

The optimization of the anchor box mechanism and the adoption of the SIoU_loss function also contribute to the increased performance of the model.

Ultimately, these changes have resulted in a more effective and efficient object detection system capable of handling

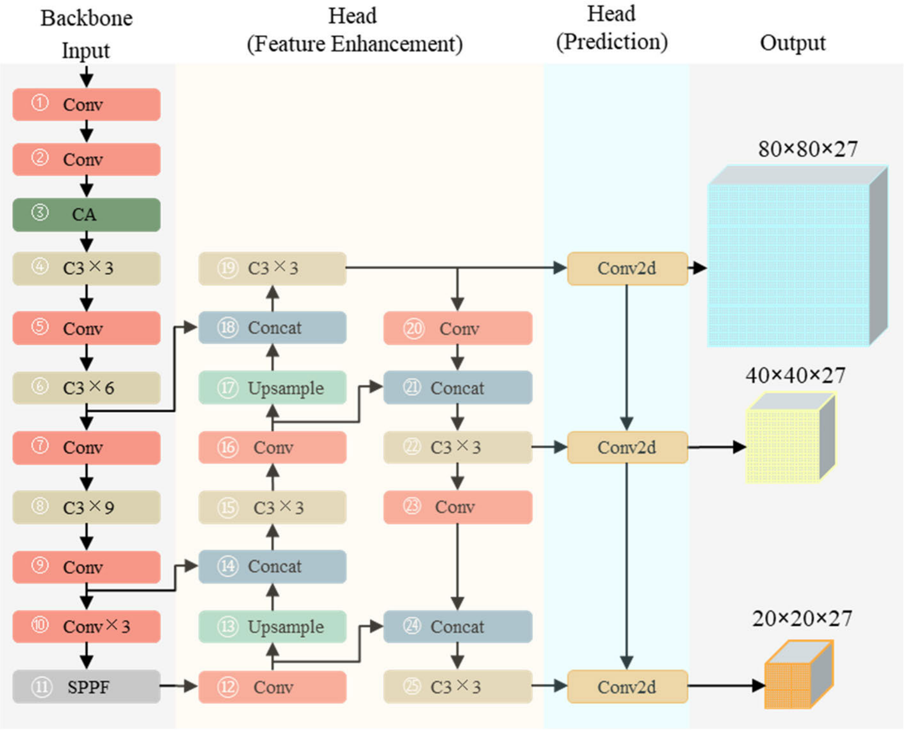

FIGURE 5. The scheme of the CA-YOLO model's network structure.

a wide range of real-world scenarios. The design model's output feature map tensors of 80 × 80×27, 40 × 40×27, and 20 × 20×27, along with their respective downsampling rates of 8, 16, and 32, are illustrated in Figure 5.

III. EXPERIMENTS AND RESULTS ANALYSIS

A. DATA SETS

For the experimental detection benchmark, model training and testing were conducted on open datasets with increasing scale, namely RSOD [33], NWPU VHR-10 [34], and DOTA.

The RSOD includes 40 background-labeled images and 936 objects-labeled images, with the image objects consisting of four different types of labeled objects with significant size differences and uneven distribution: aircraft, oiltank, overpass, and playground. Furthermore, the RSOD dataset is randomly delineated into training, validation, and test sets in the proportion of 6:2:2.

Table 2 displays the statistics of the RSOD dataset object categories and distribution.

TABLE 2. Object classification of THE RSOD dataset.

Data Sets	Dataset labeling	Number of images	Number of objects
RSOD	Aircraft	446	5374
	Oiltank	165	1698
	Overpass	176	178
	Playground	149	150

The NWPU VHR-10 dataset contains 650 labeled object images and 150 labeled background images, with a total of 10 object categories.

In contrast, the DOTA dataset was created by compiling 2806 remote sensing images, each with detailed labeling for 15 categories. The dataset was constructed using a variety of imagery sources, including Google Earth, GF-2, and JL-1 satellites from the China Centre for Resources Satellite Data and Application, as well as aerial images from CycloMedia B.V. The dataset includes both RGB and grayscale images, with the former sourced from Google Earth and CycloMedia, and the latter derived from the panchromatic band of GF-2 and JL-1 satellite imagery.

B. EVALUATION METRICS

1) PRECISION

Precision is the proportion of true positive samples in the positive classification prediction results.

$$P = \frac{TP}{TP + FP} \tag{17}$$

where False Positives (FP) refer to samples that are incorrectly labeled as positive, while True Positives (TP) represent positive samples that are correctly classified as such.

2) RECALL

The recall is the proportion of positive samples correctly identified by the classification model. In other words, it measures the percentage of positive samples that were predicted as positive.

$$R = \frac{TP}{TP + FN} \tag{18}$$

where False Negatives (FN) represent the samples that are incorrectly labeled as negative when they are positive

3) F1 SCORE

The F1 score is defined as the accuracy of a binary classification model's precision and recall. The closer the score is to 1, the more accurate the model is.

$$F1 = \frac{2 \times P \times R}{P + R} \tag{19}$$

4) MEAN AVERAGE PRECISION

The mean Average Precision (mAP) is represented by the arithmetic mean of the average precision for each object category.

$$mAP = \frac{\sum_0^N AP_n}{N} \tag{20}$$

where N stands for the total number of classes. $\mathrm{AP_n}$ stands for the average precision of class n, which is numerically equal to the area covered by the Precision-Recall function curve and the coordinate axes.

For multi-objective classification, the classification accuracy mAP for each category of objects is expressed as the mean value of $\mathrm{AP_n}$.

mAP@0.5 represents the average mean accuracy value for the IoU parameter at a threshold of 0.5.

The mean value of mAP is represented by the notation mAP@0.5:0.95, when the IoU parameter threshold is taken as [0.5:0.5:0.95].

C. MODEL TRAINING AND TESTING

The experiments and tests were conducted on a server equipped with a GeForce RTX 3060 GPU, an Intel Xeon E5-2682v4@2.50GHz CPU, and running 64-bit Windows Server 2016. The PyTorch framework and Python 3.9 were used for the experiments. The model was trained using GPU acceleration, while the tests were conducted using both GPU acceleration and CPU, as illustrated in Figure 6.

This section presents a comprehensive performance analysis of the CA-YOLOs model, which is based on the YOLOv5s framework. Figure 7 shows the training set loss function of the CA-YOLOs model plotted against the training iteration rounds (epochs). The figure displays the computed regression loss function using Equations (14), (15), and (16) respectively.

It is noteworthy that the mean values of the loss function decrease significantly as the number of epochs increases. Once training epochs approach 200, the mean values of the loss function become saturated.

Figure 8 demonstrates the pattern of the F1 score, mAP@0.5, and mAP@0.5:0.95 values as iteration rounds increase. Notably, these values exhibit a consistent upward trend until they reach stable values.

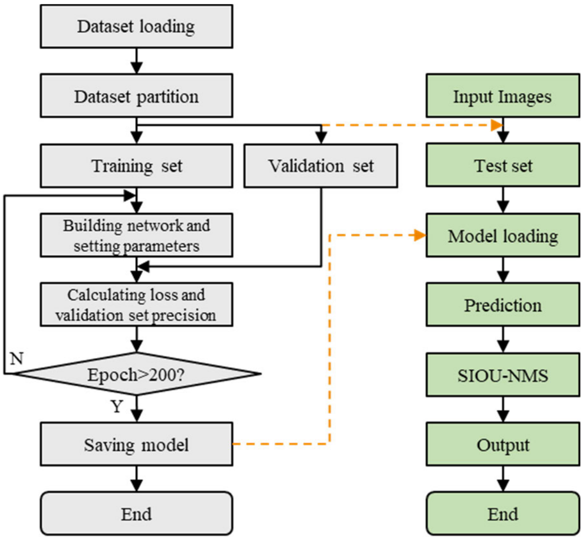

FIGURE 6. Model training and testing for CA-YOLO.

Figure 9 showcases the CA-YOLOs model's ability to accurately label and classify objects in remote sensing images ranging from 1000 × 900 to 1200 × 1000 in size. The model performs well in recognizing remote-sensing objects against a real background, and its inference speed meets the demands of a real-time system.

Additionally, the authors trained and compared four categories of labeled objects from the RSOD dataset to evaluate the detection performance. The CA-YOLOs model was then assessed on the validation set, producing a normalized confusion matrix illustrated in Figure 10. The diagonal values in the matrix indicate the percentage of each category that was predicted accurately. The rows and columns display the predicted and actual categories, respectively, while the diagonal values represent the proportion of each category that was correctly predicted.

The recall rates of aircraft, oiltank, overpass, and playground are 94%, 95%, 97%, and 98%, respectively. Therefore, this model has the highest recall rate for the playground.

To further validate the effectiveness of the CA-YOLO method in object detection and recognition, the authors conducted training, testing, and analysis of remote sensing images using publicly available datasets, DOTA and NWPU VHR-10. Figures 11 and 12 present partial experimental results, which demonstrate that the CA-YOLOs model can accurately detect multiple objects in the test set while meeting real-time inference speed requirements.

D. ABLATION EXPERIMENTS

To investigate the impact of the CA module, SPPF with a tandem construction module and anchor box optimization, on the performance of CA-YOLO models, the authors conducted model training and testing on the publicly available RSOD dataset. The tested results are shown in Table 3.

The inclusion of the CA module optimization led to an enhancement of 1.0% in mAP@0.5, and 1.3% in

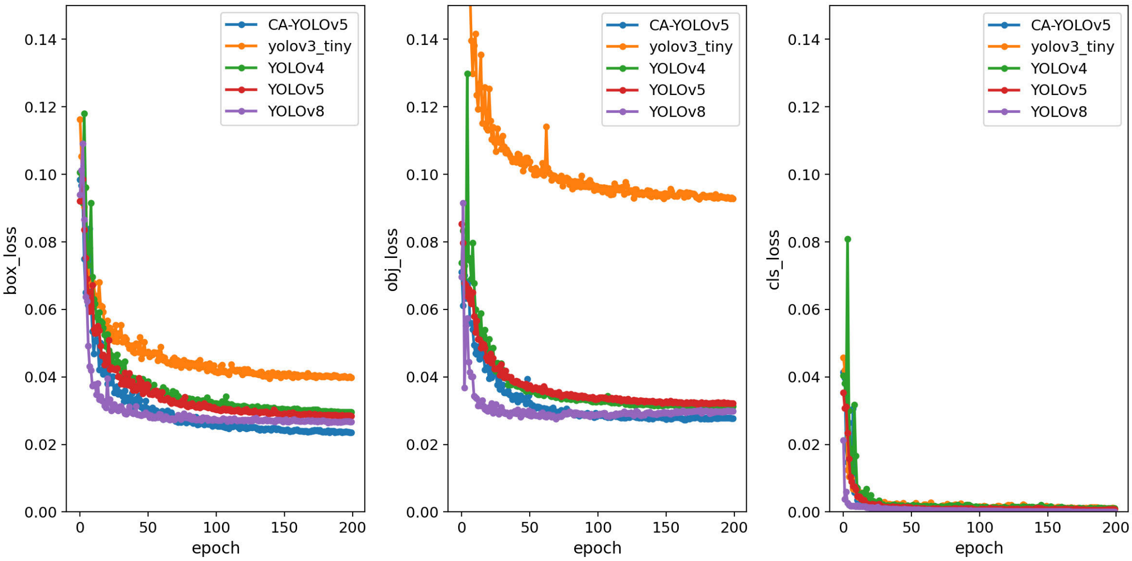

FIGURE 7. Predicted bounding box regression loss, object confidence loss, and object classification loss function variation graph with iterations of the training model.

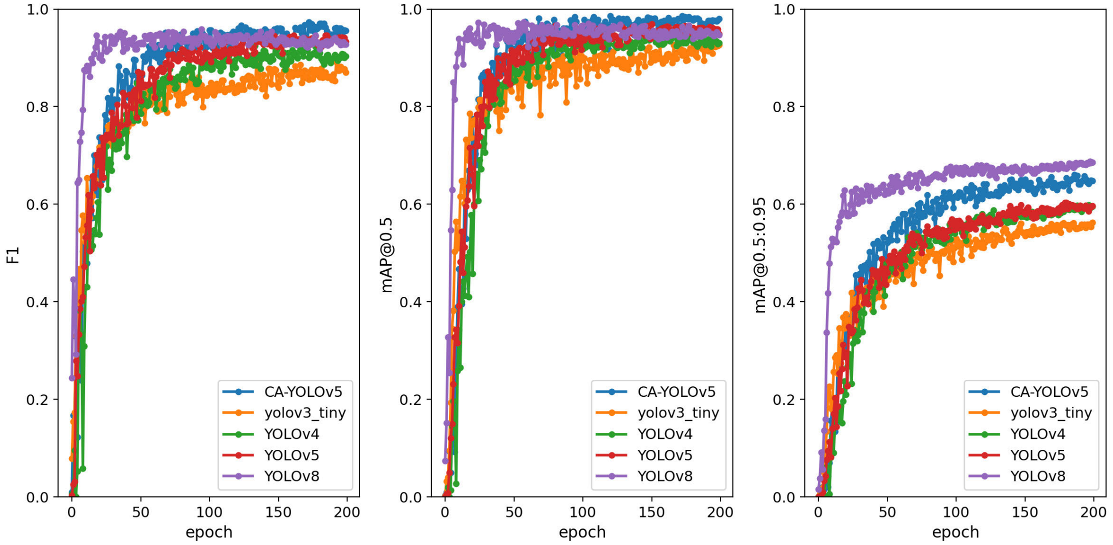

FIGURE 8. F1 score, mAP@0.5 and mAP@0.5:0.95 variation of the training model.

TABLE 3. Ablation experiments of CA-YOLO models on the RSOD.

Model	Precision/%				Precision of All/%	Recall of All/%	mAP@0.5/ mAP@0.5:0.95%
	Aircraft	Oiltank	Overpass	Playground			
YOLOv5s	89.5	90.1	90.8	92.0	90.6	90.5	89.4/79.1
YOLOv5s+CA	91.0	91.2	92.1	92.9	91.8	93.5	90.3/80.2
YOLOv5s+CA+SPPF	93.4	92.8	93.9	95.5	93.9	94.7	92.4/81.8
YOLOv5s+CA+SPPF+SIoU (CA-YOLOs)	93.7	93.3	94.9	95.3	94.3	95.5	94.2/83.2

mAP@0.5:0.95 for the YOLOv5s. Additionally, precision for all categories improved by 1.3%, and recall improved by 2.2%.

The YOLOv5s+CA model exhibited superior performance with the integration of an SPPF with the tandem construction module, resulting in a noteworthy enhancement

TABLE 4. Experimental results of different models on the RSOD.

Model	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Inference time (ms)		Parameters (Mb)	GFLOPS (Giga Floating-point Operations Per Second)
			None GPU	With GPU		
YOLOv3_tiny	84.5	72.1	380	25	243.0	141.0
YOLOv4	86.5	73.2	345	21	234.5	155.9
YOLOv5n	88.0	78.5	170	3.1	3.4	4.7
YOLOv5s	89.4	79.1	182	3.5	6.7	16.8
YOLOv5m	90.1	80.2	190	7.1	35.8	50.1
YOLOv5l	91.2	82.8	369	12.7	46.9	111.6
YOLOv5x	92.5	83.0	380	15.6	88.5	206.5
YOLOv8	89.2	80.5	175	3.0	11.3	28.9
CA-YOLOn	90.0	80.0	168	3.1	3.2	4.6
CA-YOLOs	94.2	83.2	183	3.4	6.3	16.3
CA-YOLOm	95.3	84.1	192	7.2	35.7	52.2
CA-YOLOl	96.4	85.3	380	12.7	46.7	112.2
CA-YOLOx	96.8	86.5	390	15.7	78.3	210.8

TABLE 5. Experimental results of different models on the RSOD, DOTA, and NWPU VHR-10 datasets.

Model	mAP@0.5/%			mAP@0.5:0.95/%			Inference time with GPU (ms)		
	RSOD	NWPU VHR-10	DOTA	RSOD	NWPU VHR-10	DOTA	RSOD	NWPU VHR-10	DOTA
YOLOv3_tiny	84.5	80.2	65.1	72.1	68.1	50.1	25	26	64.5
YOLOv4	86.5	81.5	67.2	73.2	69.5	54.7	21	24	78.4
YOLOv5s	89.4	87.6	68.0	79.1	75.8	57.1	3.5	5.7	65.3
YOLOv8	89.2	87.2	68.4	80.5	76.0	57.8	3.0	4.8	40.5
CA-YOLOs	94.2	90.1	69.3	83.2	80.1	60.2	3.4	5.0	50.1

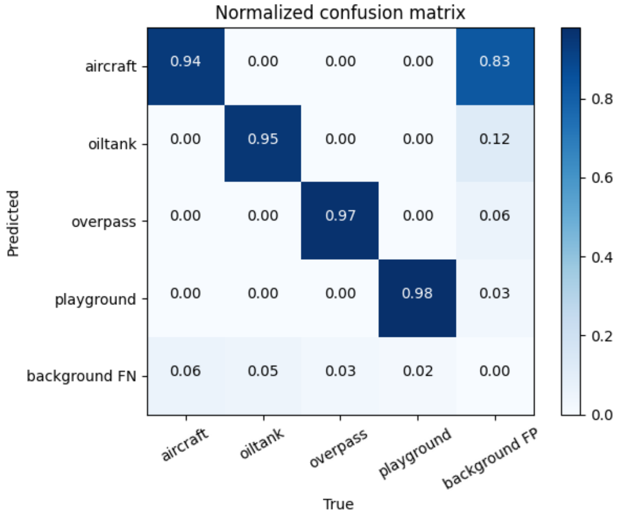

FIGURE 9. Normalized confusion matrix diagram.

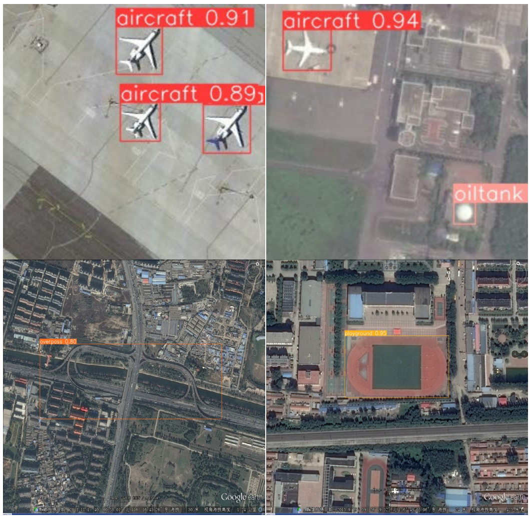

FIGURE 10. Partial results of the object detection and recognition on the RSOD dataset using the CA-YOLOs model.

in mAP@0.5 (2.3%), mAP@0.5:0.95 (2.0%), precision (2.3% on average for each category), and recall (1.3%).

Overall, the CA-YOLO (YOLOv5s+CA+SPPF+SIoU) model outperformed the YOLOv5s+CA+SPPF model, demonstrating improvements of 1.9% in mAP@0.5, 1.7% in mAP@0.5:0.95, 0.4% in precision, and 0.8% in the recall. These findings underscore the effectiveness of the CA-YOLO model.

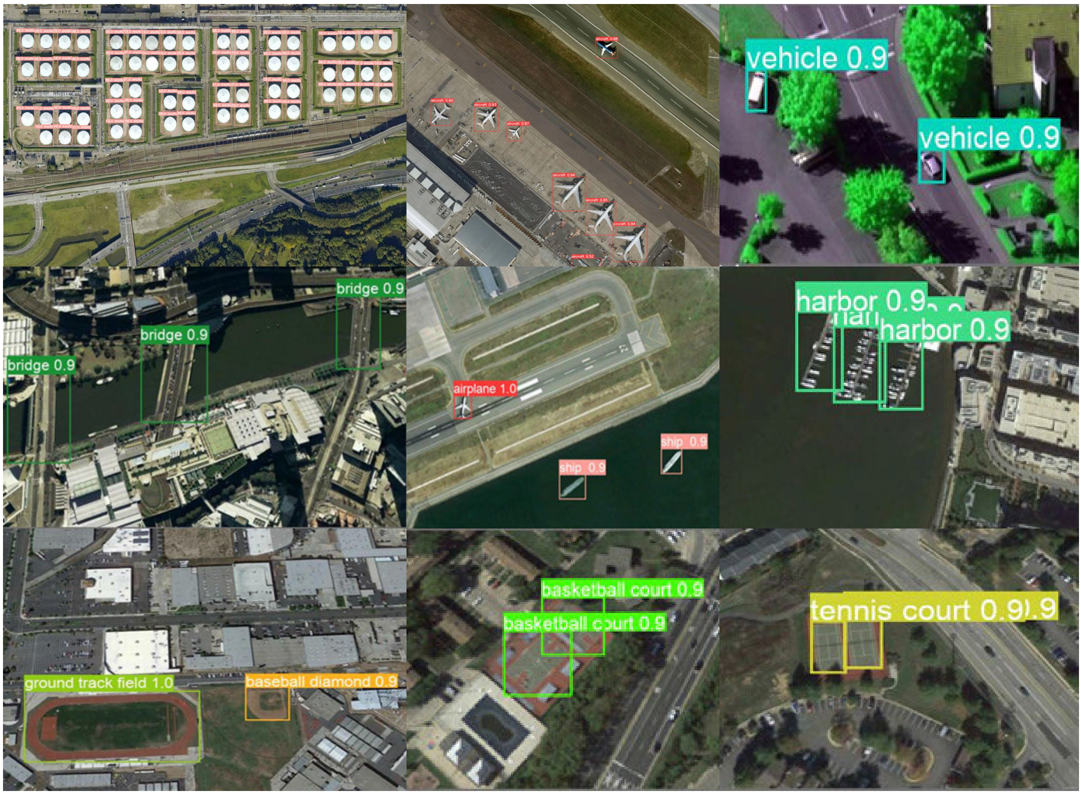

FIGURE 11. Partial results of the object detection and recognition on the NWPU VHR-10 dataset using the CA-YOLOs model.

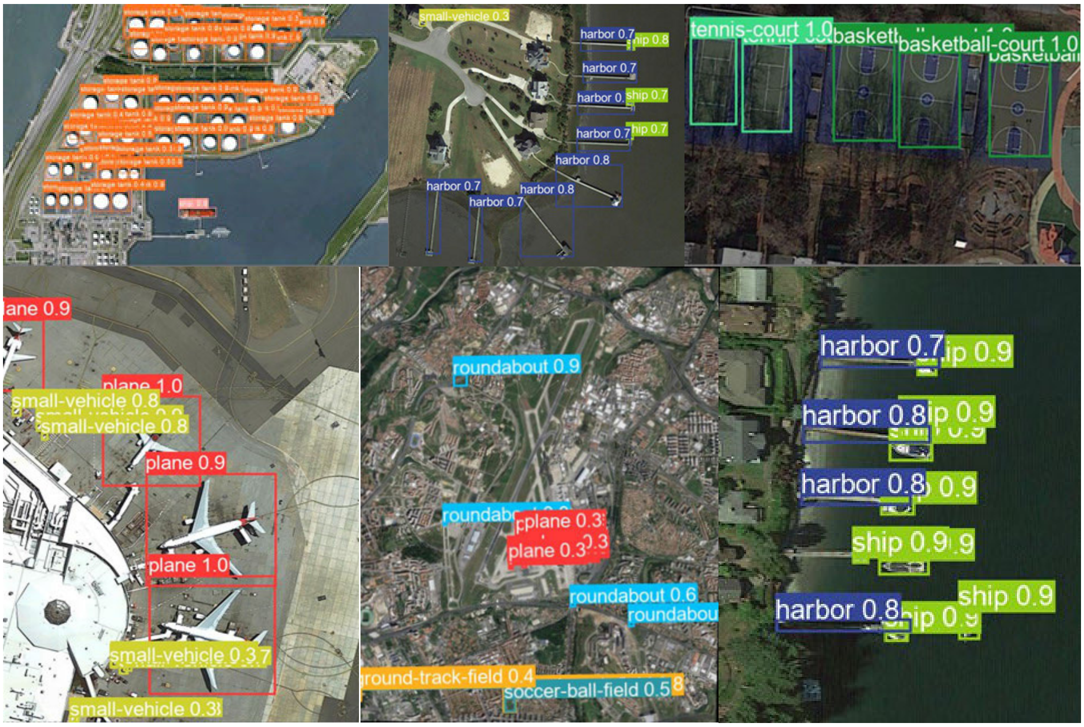

FIGURE 12. Partial results of the object detection and recognition on the DOTA dataset using the CA-YOLOs model.

E. EXPERIMENTAL RESULTS COMPARISON WITH DIFFERENT MODELS

To quantitatively analyze the performance of the CA-YOLO model, the authors applied the method of optimization to the five scale models of the YOLOv5 series, namely YOLOv5n\s\m\l\x, separately. The authors conducted model training and testing experiments on the public dataset of RSOD with the YOLOv3, YOLOv4, YOLOv5, and YOLOv8 models.

The results indicate that the CA-YOLO improves detection accuracy by 4.8% for mAP@0.5 and 3.8% for mAP@0.5:0.95 when compared to YOLOv5 series models of the same scale, when applied to the RSOD dataset.

Despite incorporating a CA module, the CA-YOLO reduces the number of model parameters by 6.1% and the computational complexity of GFLOPS by 3.6% compared to the same-scale YOLOv5 series model, due to the improved tandem construction of the SPPF module.

When GPU acceleration is absent, the CA-YOLO's average inference time increases by 1.55% compared to the YOLOv5 series models of the same scale. The average inference time grows by 1.18% with GPU acceleration, with a slight increase in relative value. Nonetheless, the average inference time of CA-YOLO is about 0.008 seconds, or 125 fps, which meets real-time requirements. Table 4 illustrates the experimental results in detail.

Furthermore, the experimental results demonstrate that the CA-YOLO model surpasses YOLOv3 and YOLOv4 in terms of both detection accuracy and inference speed. Although the CA-YOLO algorithm's inference time is slightly inferior to YOLOv8, its mean average precision is significantly higher than YOLOv8, as shown in Table 5. Evaluation on NWPU VHR-10 and DOTA datasets confirms the exceptional detection and classification performance of CA-YOLO, indicating its robust generalization capability.

IV. CONCLUSION

An improved CA-YOLO model is proposed in this paper to effectively address issues stemming from low accuracy and weak generalization in multi-size, multi-object detection. These issues arise due to the loss of semantic information about small objects after the convolution of remote-sensing images.

The YOLOv5 series of various scale models is enhanced by incorporating a coordinate attention mechanism, which facilitates refined feature extraction and reduces interference from redundant information. Fine-grained features are no longer lost during optimization, thereby improving the model's multi-scale feature learning capabilities. To foster multi-scale feature learning and fusion, a tandem construction module for SPPF is included in the design, which boosts inference speed and detection accuracy. Anchor boxes that match the target size in the dataset are optimized using a combination of the K-Means clustering method and the genetic algorithm. The prediction box position regression loss function, namely the SIoU_loss loss function, is used to optimize the weight and improve the effectiveness of target detection.

The CA-YOLO model is proven to be a highly efficient technique for improving both detection and classification accuracy in a manner that utilizes comparable numbers of model parameters and computational complexity as the YOLOv5 model. Exceptional generalization ability is exhibited by CA-YOLO when compared to alternative YOLO-based algorithms, while maintaining an optimal balance between detection accuracy and inference speed.

REFERENCES

[1] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS J. Photogramm. Remote Sens.*, vol. 159, pp. 296–307, Jan. 2020, doi: 10.1016/j.isprsjprs.2019.11.023.

[2] 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, doi: 10.1109/CVPR.2014.81.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[6] R. Gavrilescu, C. Zet, C. Foşalău, M. Skoczylas, and D. Cotovanu, "Faster R-CNN: An approach to real-time object detection," in *Proc. Int. Conf. Expo. Electr. Power Eng. (EPE)*, Oct. 2018, pp. 165–168, doi: 10.1109/ICEPE.2018.8559776.

[7] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162, doi: 10.1109/CVPR.2018.00644.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020, doi: 10.1109/TPAMI.2018.2844175.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[10] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[11] 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)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[12] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[13] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[14] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 845–853, doi: 10.1109/CVPR.2016.98.

[15] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Computer Vision—ECCV 2016* (Lecture Notes in Computer Science). Springer, 2016, pp. 354–370, doi: 10.1007/978-3-319-46493-0_22.

[16] S. Liu, D. Huang, and Y. Wang, "Receptive field block net for accurate and fast object detection," in *Computer Vision—ECCV 2018* (Lecture Notes in Computer Science). Springer, 2018, pp. 404–419, doi: 10.1007/978-3-030-01252-6_24.

[17] G. Cheng, J. Han, P. Zhou, and D. Xu, "Learning rotation-invariant and Fisher discriminative convolutional neural networks for object detection," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 265–278, Jan. 2019, doi: 10.1109/TIP.2018.2867198.

[18] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2874–2883, doi: 10.1109/CVPR.2016.314.

[19] A. Shrivastava and A. Gupta, "Contextual priming and feedback for faster R-CNN," in *Computer Vision—ECCV 2016* (Lecture Notes in Computer Science). Springer, 2016, pp. 330–348, doi: 10.1007/978-3-319-46448-0_20.

[20] B. Cheng, Z. Li, B. Xu, C. Dang, and J. Deng, "Target detection in remote sensing image based on object-and-scene context constrained CNN," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2021.3087597.

[21] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769, doi: 10.1109/CVPR.2016.89.

[22] W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, K. Wang, J. Yan, C. Loy, and X. Tang, "DeepID-Net: Object detection with deformable part based convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1320–1334, Jul. 2017, doi: 10.1109/TPAMI.2016.2587642.

[23] Z. Xu, X. Xu, L. Wang, R. Yang, and F. Pu, "Deformable ConvNet with aspect ratio constrained NMS for object detection in remote sensing imagery," *Remote Sens.*, vol. 9, no. 12, p. 1312, Dec. 2017, doi: 10.3390/rs9121312.

[24] T. Mordan, N. Thome, G. Henaff, and M. Cord, "End-to-end learning of latent deformable part-based representations for object detection," *Int. J. Comput. Vis.*, vol. 127, nos. 11–12, pp. 1659–1679, Jul. 2018, doi: 10.1007/s11263-018-1109-z.

[25] Z. Qu, F. Zhu, and C. Qi, "Remote sensing image target detection: Improvement of the YOLOv3 model with auxiliary networks," *Remote Sens.*, vol. 13, no. 19, p. 3908, Sep. 2021, doi: 10.3390/rs13193908.

[26] R. Li and Y. Shen, "YOLOSR-IST: A deep learning method for small target detection in infrared remote sensing images based on super-resolution and YOLO," *Signal Process.*, vol. 208, Jul. 2023, Art. no. 108962, doi: 10.1016/j.sigpro.2023.108962.

[27] Z. Liu, Y. Gao, Q. Du, M. Chen, and W. Lv, "YOLO-extract: Improved YOLOv5 for aircraft object detection in remote sensing images," *IEEE Access*, vol. 11, pp. 1742–1751, 2023, doi: 10.1109/ACCESS.2023.3233964.

[28] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang, "DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection," *Inf. Sci.*, vol. 522, pp. 241–258, Jun. 2020, doi: 10.1016/j.ins.2020.02.067.

[29] 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.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8759–8768.

[30] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 13708–13717, doi: 10.1109/CVPR46437.2021.01350.

[31] K. Liu, "STBi-YOLO: A real-time object detection method for lung nodule recognition," *IEEE Access*, vol. 10, pp. 75385–75394, 2022, doi: 10.1109/ACCESS.2022.3192034.

[32] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.

[33] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, "Accurate object localization in remote sensing images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2486–2498, May 2017, doi: 10.1109/TGRS.2016.2645610.

[34] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS J. Photogramm. Remote Sens.*, vol. 98, pp. 119–132, Dec. 2014, doi: 10.1016/j.isprsjprs.2014.10.002.

LINGYUN SHEN received the B.S. and M.S. degrees from the Changchun University of Science and Technology and the Ph.D. degree from the University of Chinese Academy of Sciences, China, in 2015. She is currently an Associate Professor with the Taiyuan Institute of Technology. Her current research interests include machine vision and intelligent processing of information.

BAIHE LANG received the B.S. and M.S. degrees from the Changchun Institute of Optics and Fine Mechanics, China, in 2000. He is currently an Associate Professor with the Changchun University of Science and Technology. His current research interests include deep learning and intelligent processing of information.

ZHENGXUN SONG received the B.S. degree from the Changchun Institute of Optics and Fine Mechanics and the M.S. degree from the Jilin University of Technology, China. He is currently a Professor with the International Research Center for Nano Handling and Manufacturing of China, Changchun University of Science and Technology, and the Overseas Expertise Introduction Project for Discipline Innovation. His research interests include micro-nano manipulation technology, optical communication technology, and intelligent processing of information.