



Research article

Research on an SSD remote sensing image object detection algorithm based on HSIAM

Chao Chen^{1,2,3} and Bin Wu^{1,*}

¹ School of Information and Control Engineering, Southwest University of Science and Technology, Qinglong Avenue 59, Mianyang, P. R. China

² Key Laboratory of Numerical Simulation of Sichuan Provincial Universities, Hongqiao Street 1, Neijiang, P. R. China

³ School of Mathematics and Big Data, Neijiang Normal University, Hongqiao Street 1, Neijiang, P. R. China

* **Correspondence:** Email: wubin@swust.edu.cn; Tel: +8613909018585; Fax: +86-816-6089115.

Abstract: Aiming at the problems, such as missed detection of small targets, positioning deviation of rotating targets, and complex background interference in remote sensing images, an improved SSD algorithm based on the High-Level Semantic Information Activation Module (HSIAM) and the improved BSWIoU based on Bhattacharyya distance was proposed. The HSIAM module enhances information fusion capabilities within the deep network. The CA mechanism employs adaptive average pooling to enhance focus on central regions of feature maps, distinguishing small targets within complex backgrounds. The RBD_IoU loss function integrates an orientation-matching constraint and a dynamic weighting mechanism to mitigate rotational bounding box regression bias. Experimental results for three benchmark datasets (DIOR, DOTA, and NWPUCHR) showed that, compared with the baseline SSD algorithm, the mAP50 of the improved model increased by approximately 2%. Furthermore, it achieved a balanced trade-off between accuracy and speed, with 12.5% fewer parameters than YOLOv8s. This provides a high-precision and lightweight solution for target detection in remote sensing images.

Keywords: SSD, Remote sensing object detection, HSIAM, modified RBD_IoU

Mathematics Subject Classification: 68T05

Abbreviations:

Table 1. Abbreviations.

Common abbreviations	English full names	Common abbreviations	English full names
CBAM	Convolutional Block Attention Module	PANet	Path Aggregation Network
BiFPN	Bidirectional Feature Pyramid Network	YOLO	You Only Look Once
SSD	Single Shot Multibox Detector	RPN	Region Proposal Network
FLOP	floating point operation	DIOR	Drones In Optics Recognition
Dota	Dataset for Object Detection in Aerial images	NWPUVHR-10	10-level geospatial remote sensing dataset (NWPU, 2014)
ROI Align	Region of Interest Align	RCNN	Region-based Convolutional Neural Networks
Faster R-CNN	Faster Region-based Convolutional Neural Networks	FPN	Feature Pyramid Network
Mask RCNN	Mask Region-based Convolutional Neural Networks	ROI AlignReLU	Region of Interest Align Rectified Linear Unit
BECLogits Loss function	Binary Cross Entropy with Logits Loss	GIOU	Generalized Intersection over Union
mAP50	Mean Average Precision at IoU=0.5	mAP50:95	Mean Average Precision at IoU=0.5:0.95
IoU	Intersection over Union	SPP	Spatial Pyramid Pooling
CA	Coordinate Attention	HSIAM	High Semantic Information Activation Module

1. Introduction

The image target in remote sensing algorithms is the core technology of earth observation and has important application value in fields such as national land survey and disaster monitoring [1, 2]. With the popularity of high-resolution satellites and unmanned aerial vehicle (UAV) platforms, remote sensing images present three key characteristics, posing severe challenges to detection algorithms: multiple occlusions, small targets, and complex backgrounds [3–5].

1.1. Analysis of technical bottlenecks

Current mainstream detection models face three major limitations:

- (1) Missed detection of small objects: Conventional downsampling operations cause features of objects smaller than 32×32 pixels to be lost during processing. Consequently, the recall rate of small objects in drone images remains low [6].
- (2) Localization deviation for rotated objects: Standard horizontal bounding box regression losses are insensitive to the orientation of rotated objects [4].
- (3) The diversity of land cover types leads to a high false alarm rate [7], particularly in urban-rural transition areas.

1.2. Research status

While mainstream algorithms of object detection have achieved significant advancements in accuracy and time performance, they exhibit limitations in remote sensing scenarios [8]. The original PANet structure suffered from insufficient multi-scale feature interaction, achieving only 73.0% mAP50 for the DOTA dataset. Yuan et al. [9] enhanced accuracy by 4.3% (reaching 79.5% mAP50) in low-light crowded scenarios using CycleGAN data augmentation and a CrowdDet-V dense detection mechanism. This model provided reliable technical support for rush-hour traffic management and autonomous driving, but struggled with long-range low-light vehicle detection. Z. J. Khow et al. [10] integrated CA attention, WIoU loss, and geometric distance estimation within the

YOLO framework, offering a new paradigm for real-time visual perception systems. However, the model required focal parameter recalibration for unseen scenes. N. Singh et al. [11] introduced specialized data augmentation techniques to simulate rainy conditions, adjusted the network architecture to improve resilience to rain-induced noise to enhance model performance for high-precision real-time monitoring. Xu et al. [12] proposed a “crop localization + dynamic scaling” joint strategy to balance speed and accuracy for 4K sports video detection, presenting a novel approach for multi-view big-data scenarios. However, this strategy was unsuitable for dense scenes and failed to meet real-time requirements (1.8 fps falls significantly below the 25 fps threshold). The SE module [13] neglects spatial correlations, leading to feature confusion in complex backgrounds. Feng et al. [14] designed a C2f-Faster-EMA module that suppressed background interference through multi-scale attention, reducing parameter count by 36.04%. The CIoU loss function shows poor robustness to changes in aspect ratio, resulting in fluctuations in the positioning accuracy of rotated objects [4]. Chen [15] adopted Focal-EIoU to replace CIoU, thereby improving bounding box regression precision. Various variant YOLO algorithms [16, 17] were based on powerful backbone networks with multi-dimensional attention that extract features through multi-channel fusion, or improvements in activation functions, detection head modules, or loss functions. They have achieved good results in object detection but lack specialized research on remote sensing images. The SSD algorithm feature fusion module constructs a bidirectional network channel for feature fusion through depthwise separable convolution and upsampling operations [18, 19]. The detailed epigenetic information and high-dimensional semantic information from different scale feature layers of the backbone network can be more fully integrated [20–22]. Various improved SSD object detection algorithms (such as those in [23, 24]) have introduced more complex backbone networks to extract high-quality features, but their accuracy and speed are still lower than those of YOLOv8.

Some scholars [25–27] leveraged learning domain-invariant and class-specific information to improve the generalization ability of object detection models. Representation-shared layers [28–30] generated domain-specific and shared interdomain features, making this architecture flexible and powerful in capturing interactive information between shallow and deep layers.

1.3. Innovative contributions

To address the aforementioned challenges, we proposed a HSIAM module and an improved loss function for object detection. The primary contributions are as follows:

- (1) The HSIAM module was proposed to enhance the feature extraction performance of the backbone network.
- (2) Orientation and aware loss function was designed the Shape-aware RBD_IoU loss function, which integrates an orientation-matching constraint and a dynamic weighting mechanism to mitigate rotational bounding box regression bias.
- (3) Experimental verification was conducted on three datasets. The improved model had fewer parameters and a higher mAP50.

2. Theoretical analysis

To address the challenge of backbone networks struggling to generate candidate boxes containing only small targets and their difficulty in distinguishing background information for accurate discriminative feature extraction, we proposed a HSIAM module with superior feature extraction capability and optimized the modified Bhattacharyya Distance-based IoU (BSWIoU) loss function. The innovation lied in the synergistic operation of the following four core modules:

2.1. Coordinate attention mechanism (CA)

Building upon the ground breaking research by Hou et al., the Coordinate Attention (CA) mechanism was developed and deeply embedded within the neural network architecture. Unlike traditional attention mechanisms that rely on channel dimension compression, this innovation employs a bidirectional coordinate axis feature decoupling strategy. This approach efficiently captures spatial information while preserving the integrity of channel information. Its core breakthrough is the decomposition of image spatial coordinates into vertical and horizontal orthogonal basis vectors, establishing a dual-engine driving mechanism that integrates channel dimension interactions with spatial directional awareness. By dynamically calibrating the spatial weight distribution, it reduces target bounding box localization errors compared to conventional methods. At the technical implementation level, the system performs one-dimensional global feature aggregation separately along the vertical and horizontal axes. Vertical aggregation surpasses the feature intensity gradient field limitations of traditional attention mechanisms in the spatial dimension. Horizontal aggregation Achieves vectorized representation of target geometric features (such as aspect ratio, rotation angle, etc.) by constructing a coordinate-aware feature weight distribution matrix, as illustrated in Figure 1. Through the axial feature decoupling within the orthogonal coordinate system, CA simultaneously optimizes channel dimension information interaction and spatial dimension directional perception capabilities. This dual-path feature enhancement mechanism significantly improves the geometric localization accuracy of target regions.

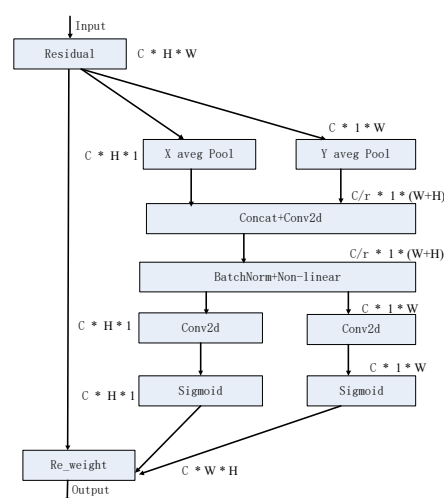


Figure 1. Schematic diagram of coordinate attention.

avgPool module: Average pooling operation; the CA attention mechanism first performs two global average pooling operations on the input feature map, one along the width direction and the other along the height direction. These two operations respectively yield two feature maps: X Avg pool is the average pooling along the width direction results in a feature map; Y Avg pool is the average pooling along the height direction results in a feature map. These two feature maps capture the global features along the width and height directions, respectively. Concat module is concatenation operation. Conv2d module: 2D convolution operation; batchnorm module: Batch normalization followed by a non-linear activation operation. non_linear module: Batch normalization followed by a non-linear activation operation. Sigmoid module: Sigmoid non-linear activation operation; residual module: Residual connection layer; re_weight module: Re-weighting operation.

Given the input feature map x , we separately encode each channel using pooling kernels with spatial extents of $(H, 1)$ and $(1, W)$, respectively. Consequently, for the c -th channel, the output at height h is expressed as shown in Eq (2.1).

$$z_c^h(h) = \frac{1}{W} \sum_{0 \leq i < W} x_c(h, i). \quad (2.1)$$

Similarly, the output for the c -th channel at width w is expressed as shown in Eq (2.2):

$$z_c^w(h) = \frac{1}{H} \sum_{0 \leq j < H} x_c(j, w). \quad (2.2)$$

The synthetic feature is generated as shown in Eq (2.3), where, F_1 represents the 1×1 convolution operation.

$$f = \delta(F_1([z_h, z_w])), \quad (2.3)$$

where $[\cdot, \cdot]$ is the concatenation operation, and δ represents a nonlinear activation function. We split the transformed feature map f along the spatial dimension into two separate tensors (f^h and f^w). Subsequently, F^h and F^w (a pair of 1×1 convolutional transformations) were applied to f^h and f^w , respectively, and converted into tensors with the same number of channels as the input x . This process yielded feature representations matching the original channel dimensionality, as formalized in Eqs (2.4).

$$\begin{cases} g^h = \delta(F_h(f^h)) \\ g^w = \delta(F_w(f^w)) \end{cases}, \quad (2.4)$$

where σ denotes the activation function. The axial feature encoding module was first constructed. Through spatial axis decoupling technology, the traditional 2D global pooling was decomposed into horizontal/vertical directional feature compression operations, generating coordinate-sensitive feature vectors ($\mathbf{g}_h, \mathbf{g}_v$). Subsequently, joint attention modeling was performed. Using a multi-modal feature fusion strategy, the axial feature vectors underwent tensor expansion to establish a spatial-channel cross-correlation weight distribution model. To address feature entanglement effects in shallow networks, a boundary enhancement gating architecture was innovatively introduced. Through a non-linear response enhancement mechanism, selective feature amplification was applied to target contour regions. The output features \mathbf{y} of the coordinate attention block were represented as shown in Eq (2.5).

$$y_c(i, j) = x_c(i, j)g_c^h(i)g_c^v(j). \quad (2.5)$$

The output feature $y_c(i, j)$ is the product of each element in the feature matrix x_c and its $\delta_c^h(i)$ and $\delta_c^w(j)$ corresponding weighted sum.

2.2. Improvements to activation functions

Information transfer between adjacent network layers is facilitated through differentiable nonlinear functions, collectively termed activation functions. Their essential roles are manifested in three key aspects. By stacking nonlinear transformations, networks gain the capacity to approximate any borel measurable function (as established by Cybenko's theorem). Fulfilling the lipschitz continuity condition ensures the effectiveness of the backpropagation algorithm. Selective feature enhancement is achieved through dynamic modulation between saturated and unsaturated regions. ANNs learn patterns through layered information propagation. The fundamental computational units of an ANN satisfy the mathematical characteristics. The feedforward process is formally described as follows: For the j -th neuron in layer l , its input-output relationship is expressed by Eq (2.6):

$$y_j^{(l)} = \delta \left(\sum_{i=1}^{l-1} w_{ji}^{(l)} x_i^{(l-1)} + b_j^{(l)} \right), \quad (2.6)$$

where $w_{ji}^{(l)}$ denotes the connection weight from the i th neuron in layer $l - 1$ to the j th neuron in layer l ; $b_j^{(l)}$ represents the bias term of the j th neuron in layer l ; δ indicates the activation function. Figure 2 visually presents the typical activation units along with their derivative properties.

The ReLU derivative is zero in the negative region, which may lead to the “nerve death” problem, possibly altering the sensitivity to small objects. The derivative is constant (0 or 1) and results in minimal computational overhead during backpropagation, which makes it suitable for real-time detection tasks (such as the YOLO family). The zero gradient in the negative region prevents the gradient flow in the deep network and the convergence of the model. Leaky ReLU introduces a small nonzero gradient in the negative region, maintains weak signal, and improves the detection of small objects. The derivative remains 1 in the positive region, which preserves the validity of the computation. Its backpropagation complexity is similar to ReLU. Though, it is sensitive to hyperparameters; too large a negative slope will cause the gradient to explode. For large absolute input values, the Sigmoid derivative tends to zero, which eliminates the gradient in the deep network. This makes it unsuitable for deep algorithms for object detection. Its output range (0.1) is suitable for probabilistic predictions (e.g., confidence scores for YOLO), but using it in hidden layers requires caution. The output of the Tanh activation function is the center zero, and its derivative is symmetric, which reduces the gradient shift problem with the Sigmoid. However, for large absolute inputs (derivatives close to zero), gradient dissipation exists, limiting its use in the hidden layers of detection models. The Swish derivative combines the original values of the function and enables adaptive gradient adjustment. This improves the ability to represent features in complex scenes (e.g., hidden object detection). It is continuously derivable with no discontinuities, which leads to a more stable optimization. The Mish derivative is continuous without discontinuities, which reduces gradient oscillations and improves the accuracy of the limit box regression (adopted in YOLOv4/v5/v11) [16, 17]. However, its more complex derivatives increase the backpropagation time by about 15% compared to ReLU, which requires a tradeoff between accuracy and speed. Common derivations of activation functions are shown in Figure 3.

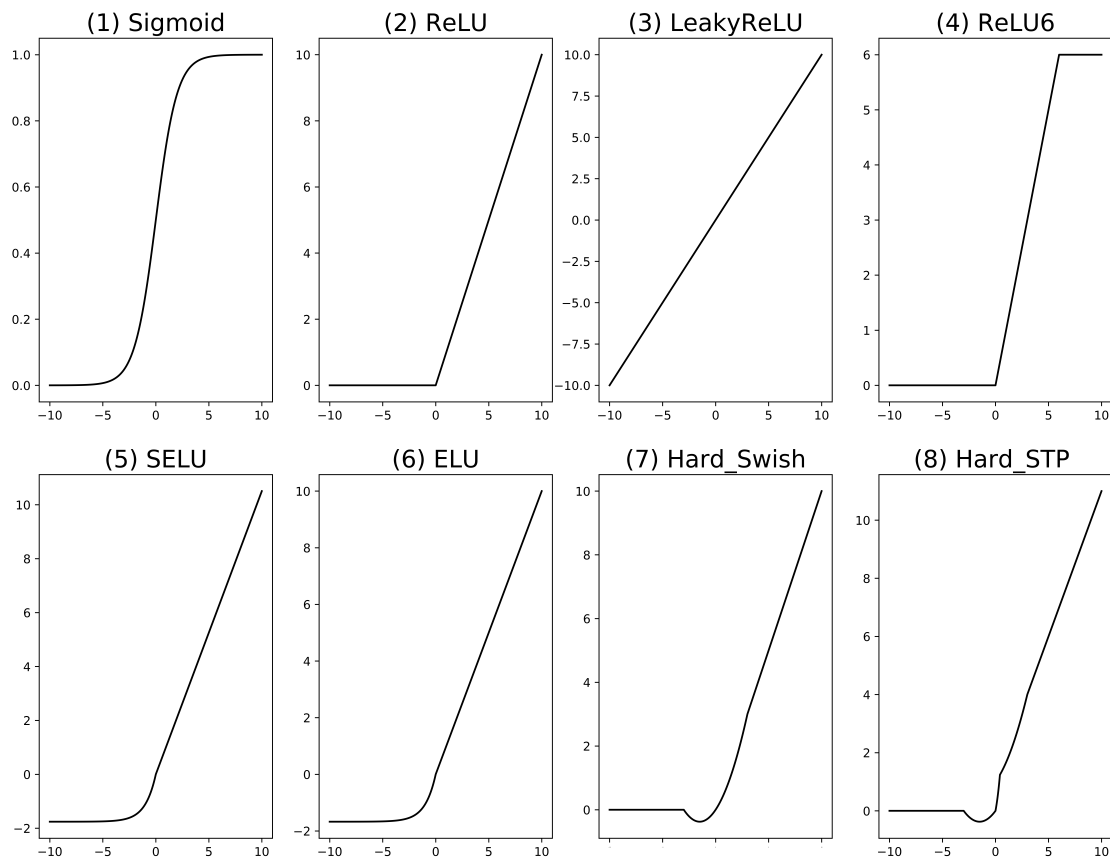


Figure 2. Activation functions.

Regarding the selection strategy for nonlinear activation units within object detection frameworks, the YOLO architecture employs a gradient optimization scheme based on the Rectified Linear Unit (ReLU) and its parametrized variants. The standard rectified unit achieves feature sparsity activation by establishing a threshold gating mechanism. In contrast, the improved variants introduce a negative slope parameter to mitigate the vanishing gradient problem. Their respective mathematical expressions are defined in Eq (2.7).

$$ReLU(x) = \begin{cases} x, & x \geq 0 \\ 0, & x \leq 0 \end{cases} \quad (2.7)$$

Upon input, the derivative values are computed. If excessively large learning rates during the initial training phase drive weight parameters into negative regions, the forward propagation satisfies $x \leq 0$. This results in gradient values remaining zero throughout the backpropagation process. This condition triggers permanent neuron deactivation (Dead Neuron Phenomenon). Experiments for the CIFAR-10 dataset revealed that approximately 12.3% of neurons lost activation capability early in training. To mitigate the vanishing gradient problem, researchers have proposed improvements, such as the LeakyReLU activation function, as given in Eq (2.8).

$$LeakyReLU(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x \leq 0 \end{cases} \quad (2.8)$$

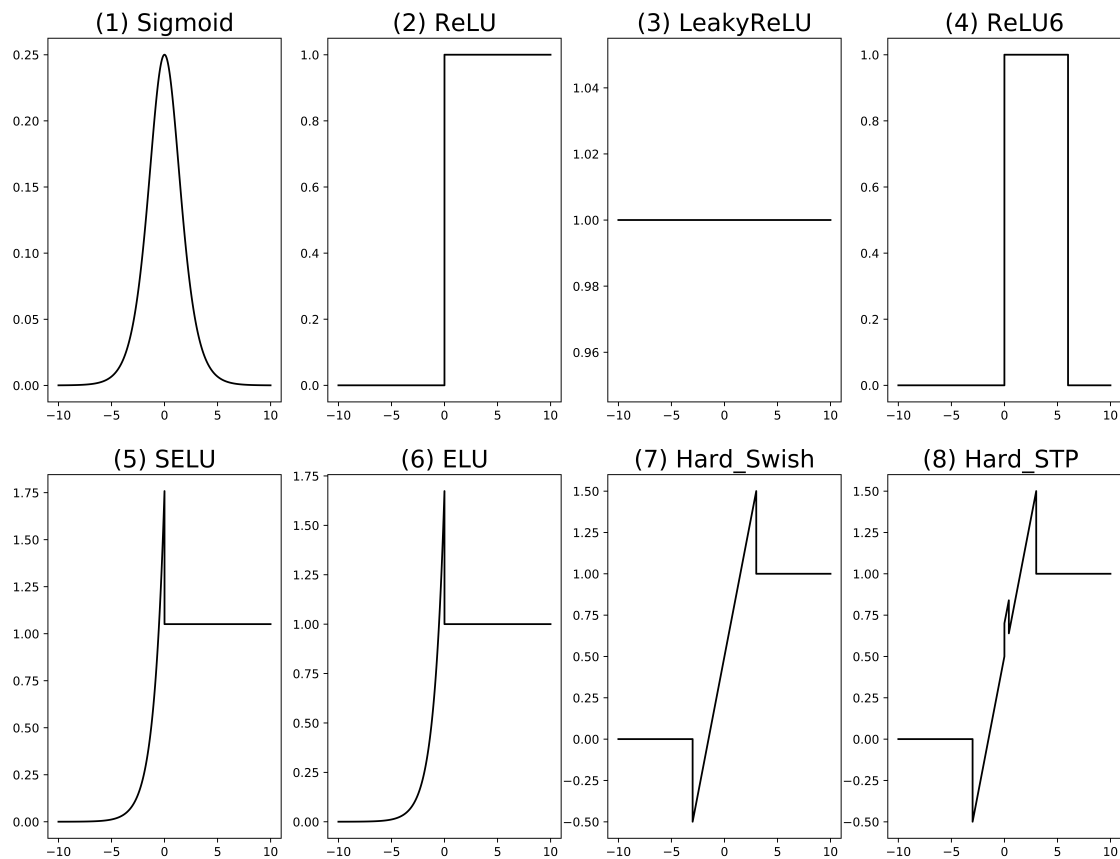


Figure 3. Schematic diagram of activation function derivatives.

Table 2. Ablation experiment results of the SSD algorithm with different conventional activation functions on the DIOR dataset.

Algorithms	mAP%	Model Memory(M)
SSD(Sigmoid)	0.772208	12.60
SSD(SiLU)	0.780808	12.60
SSD(ReLU)	0.780855	12.60
SSD(ReLU6)	0.784256	12.60
SSD+Hard_STP(a=0.1,b=0.5,c=0.5)	0.771808	12.60
SSD+Hard_STP(a=10.1,b=0.5,c=0.5)	0.780205	12.60
SSD+Hard_STP(a=1.0,b=0.5,c=1.6)	0.780757	12.60
SSD+Hard_STP(a=1.0,b=1.5,c=1.0)	0.782807	12.60
SSD+Hard_STP(a=1.05,b=1.5,c=1.6)	0.788802	12.60

Finally, inspired by a series of activation functions in references [16, 17] (such as ReLU6, Swish, Sigmoid, and polynomial activation functions), the novel Hard_STP(x) activation function was adopted for downsampling operations. The Hard_STP(x) function preserved feature information within non-zero distributions and maintained feature extraction capabilities even with minimal gradient values,

thereby enhancing semantic perception during feature fusion. The function is defined in Eq (2.9). Ablation experiment results of the SSD algorithm with different conventional activation functions on the DIOR dataset are shown in Table 2.

Experimental results indicated optimal performance when $a = 1.05$, $b = 1.5$, $c = 1.6$, respectively. This function demonstrated the following advantages during downsampling: The positive interval slope enhances responses to strong features; the negative interval coefficient prevents gradient truncation; The bias term improves geometric awareness for high-level features.

$$\text{Hard_STP}(x) = \frac{x \text{ReLU6}(x + 3)}{6} + (ax^3 + bx^2 + cx). \quad (2.9)$$

Hard_STP(x) enabled shallow information to penetrate deeper layers of the neural network, thereby enhancing the algorithm's accuracy and generalization capability, without incurring an orders-of-magnitude increase in computational load. It was defined mathematically in Eq (2.9). The Hard_STP(x) activation function is a smooth and non-monotonic activation function that incorporates the advantages of most common activation functions, as illustrated in the eighth subfigures of Figures 2 and 3. This property facilitates network optimization and improves generalization performance. Being lightweight and flexible (with the highest term being cubic), this activation function can replace common activation functions within the backbone networks of mainstream models. This substitution enhanced the representational capacity for discerning features of small targets in images.

2.3. HSIAM module

In order to solve the problem of feature confusion caused by low contrast between small objects and background, a HSIAM module [18] was proposed. This module integrates the CA mechanism and Hard_STP(x) activation function to perform the following: This module enhances boundary localization by combining the CA mechanism with the Hard_STP(x) activation function, collectively strengthening target edges that respond through spatial perception. Feature selection optimization suppresses redundant feature propagation under background noise interference. Multi-scale information fusion preserves the geometric integrity of crosschannel feature interactions. The proposed method cannot only extract image features with high accuracy, but also effectively suppresses channels with less information and highlights key features. This significantly improves the accuracy and realtime performance of small object detection in images. The mechanism refines target bounding box localization through the coordinate (cooperative) attention mechanism. This is achieved by performing a Hadamard product (element-wise multiplication) between the feature vectors generated by coordinate attention and the original input features, enabling precise localization of small targets. The detailed architecture of the higher-level semantic activation module is illustrated in Figure 4.

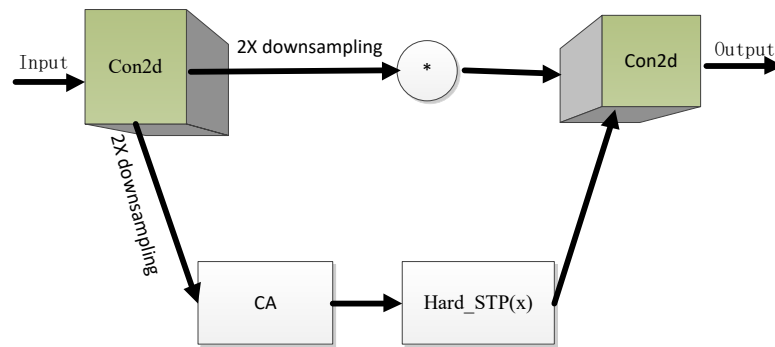


Figure 4. Higher-level semantic activation module.

The $\text{Hard_STP}(x)$ activation function preserves the boundary information of the non-zero distribution region, which facilitates efficient information interaction and fusion.

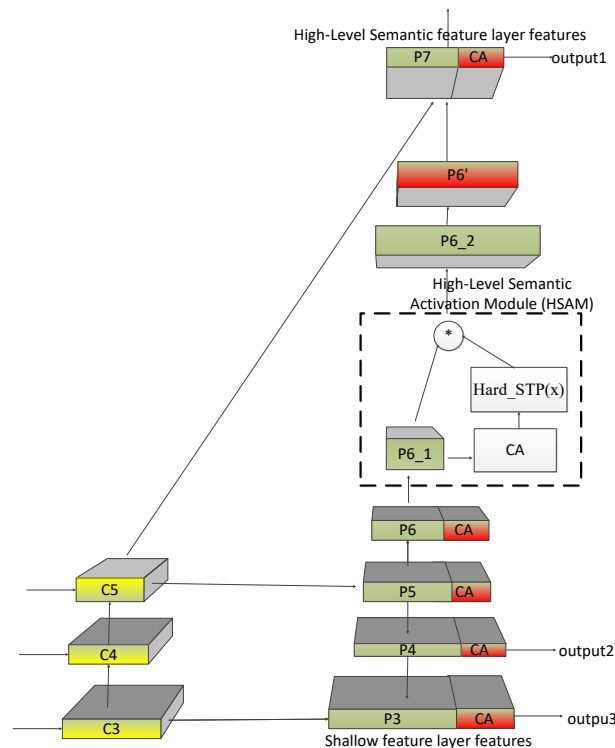


Figure 5. CA-Scale hierarchical feature pyramid structure (CA denotes coordinate attention mechanism).

2.4. CAscale hierarchical feature pyramid structure

We built upon the unidirectional vertical-horizontal feature fusion architecture in [19, 24], which effectively facilitated cross-layer information interaction and integrated the Coordinate Attention (CA) mechanism with the High-Level Semantic Activation Module (HSAM). This integration enabled the redesign of the shallow network architecture, culminating in the proposed CAscale hierarchical feature

pyramid structure for feature enhancement. The cascaded processing pipeline is detailed in Figure 5 and comprises three stages as follows: (a) Features were extracted from the shallow convolutional layers Conv2, Conv3, and Conv4. (b) Feature maps P3–P5 underwent spatial attention focusing via the CA module. Feature map P6 employed the HSAM module to augment long-range dependency modeling. Feature map P7 performed cross-scale feature distillation. (c) Feature fusion: Original features from the shallow convolutional layers were fed into the CAscale hierarchical feature pyramid structure. The CA module was then applied to each level (P3–P7) to accentuate the positional and boundary information of small targets. Crucially, features processed by the P6 convolution were further enhanced by the HSAM to improve bounding box localization capability before being fused with the C5 convolutional features (thus forming high levels of semantic layers after CA augmentation). Feature maps (C3, C4, and C5) first underwent 1×1 convolution, followed by top-down feature fusion to generate the fused feature maps (P3, P4, and P5). Feature map P5, which was rich in semantic information, underwent downsampling to further refine critical features. Subsequent downsampling of P5 with stride 2 produces higher-level feature maps (P6 and P7). The HSIAM was strategically placed between P6 and P7 to enhance information fusion capabilities within the deep network. While the CA mechanism employed adaptive average pooling to enhance focus on central regions of feature maps, distinguishing small targets within complex backgrounds, it necessitated allocating greater attention to relevant areas. The final fused features were obtained as specified in Eq (2.10).

The feature layers output from the convolution of C3, C4, and C5 were enhanced by modules P3, P4, and P5 to extract discriminative features for small objects of interest, respectively.

$$P_7 = C_5 \oplus P'_6. \quad (2.10)$$

Where, C_5 is the feature map of the 11-th layer and P'_6 is the feature map fused via the HSAM Module, respectively; \oplus represents channel-wise concatenation. This module is illustrated in Figure 5.

2.5. Loss function modification

Objects with small subpixel scales have high sensitivity to geometry parameters (size and location) of the border. Such sensitivity results in a double dilemma in model training: Lack of high quality training samples and insufficient alignment of features between candidates and targets. In other words, these constraints hinder optimization of the model and reduce detection accuracy. When a spatial separation takes place between a true and predicted boundary box, it is noted that a conventional cross-summation ratio (IoU) loss cannot provide effective optimization due to a gradient loss. As can be seen from Eq (2.11), the gradient property inherent in the existing target detection loss function not only delays the convergence of the model, but also makes the detection system unable to meet the practical

accuracy requirements.

$$\begin{aligned}
 \text{loss(object)} = & \lambda_{\text{coord}} \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{\text{obj}} \left[(x_i - x'_i)^2 + (y_i - y'_i)^2 \right] \\
 & + \lambda_{\text{coord}} \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{\text{obj}} \left[(2 - w_i - h_i) \left((w_i - w'_i)^2 + (h_i - h'_i)^2 \right) \right] \\
 & - \lambda_{\text{noobj}} \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{\text{obj}} \left[c_i \log(c_i) + (1 - c_i) \log(1 - c_i) \right] \\
 & - \lambda_{\text{noobj}} \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{\text{noobj}} \left[c_i \log(c_i) + (1 - c_i) \log(1 - c_i) \right] \\
 & - \sum_{i=0}^{K \times K} I_{ij}^{\text{obj}} \left[p_i \log(p_i) + (1 - p_i) \log(1 - p_i) \right].
 \end{aligned} \tag{2.11}$$

Parts I and II represent the coordinate loss and quantify the disagreement between the predicted and ground truth coordinates. Parts III and IV constitute the trust loss and measure the difference between the predicted trust score and the actual trust value. Part V represents the classification loss and evaluates the difference between the predicted and actual class probabilities. Where: I_{ij}^{obj} marks the presence of a target object in the prediction; I_{ij}^{noobj} indicates the absence of a target object (background); λ^{coord} is a weighting coefficient; k signifies the total number of target objects across all predicted bounding boxes. x_i , y_i , x'_i , and y'_i denote the center coordinates in horizontal and vertical, respectively; w_i , h_i , w'_i , and h'_i represent the width and height; c_i indicates the object class; p_i is the probability of belonging to class c ; Common loss functions such as DIOU and CIOU address only the aspect ratio and centroid distance of bounding boxes in their penalty terms. The modified centroid position incorporating the Bhattacharyya distance is expressed in Eq (2.12).

$$BD_IoU = \log \left(\frac{1}{2} \frac{(w_1^2 + w_2^2)(h_1^2 + h_2^2)}{w_1 w_2 h_1 h_2} \right) + \left(\frac{(cx_1 - cx_2)^2}{w_1 + w_2} \right) + \left(\frac{(cy_1 - cy_2)^2}{h_1 + h_2} \right). \tag{2.12}$$

To account for bounding box scale information when measuring the distance between bounding box centers [16, 17], we mitigated the influence of bounding box scale on relational assessment by incorporating the modified BD_IoU. For precise characterization of relationships among small-scale target bounding boxes, we refined the loss function based on SIOU and WIoU metrics, combined with the modified BD_IoU. The improved loss function is expressed in Eq (2.13).

$$RBD_IoU = 1 - IoU + \frac{\Delta + \Omega}{2} + \frac{\beta}{2\delta\alpha^{\beta-\delta}} + \frac{1}{2} \log \left(\frac{(w_1^2 + w_2^2)(h_1^2 + h_2^2)}{w_1 w_2 h_1 h_2} \right) + \frac{(cx_1 - cx_2)^2}{w_1 + w_2} + \frac{(cy_1 - cy_2)^2}{h_1 + h_2}, \tag{2.13}$$

where x_1 , y_1 , x_2 , and y_2 denote the horizontal and vertical center coordinates of the predicted box, respectively, and w_1 , h_1 , w_2 , and h_2 represent the width and height of the ground-truth box, respectively.

Some parameters are illustrated in Figure 6.

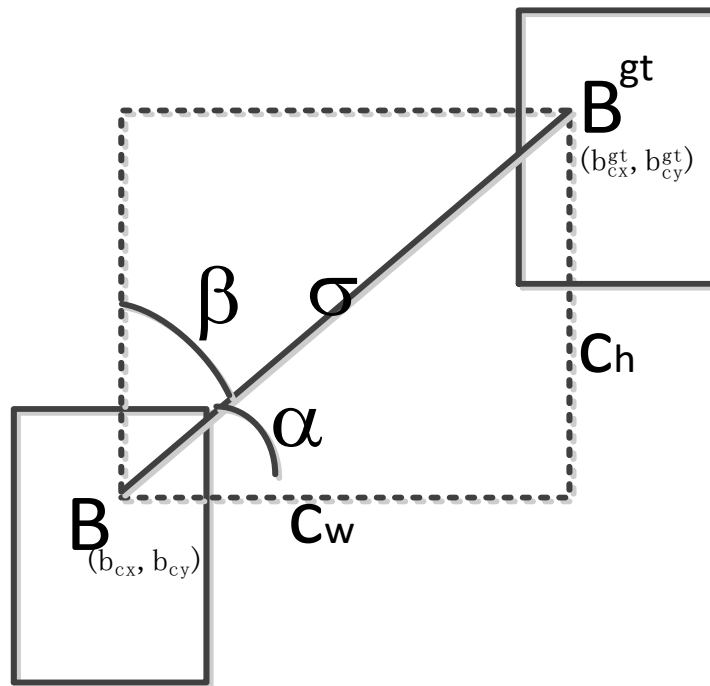


Figure 6. Schematic diagram of loss functions, including overlap area, center point distance, aspect ratio, shape loss, and angular loss.

The corresponding parameters are given by Eqs (2.14).

$$\begin{cases} \Delta = \sum_{t=x,y} (1 - e^{-r\rho_t}) = 2 - e^{-r\rho_x} - e^{-r\rho_y} \\ \rho_x = \left(\frac{b_{cx}^{gt} - b_{cx}}{c_w} \right)^2 \\ \rho_y = \left(\frac{b_{cy}^{gt} - b_{cy}}{c_h} \right)^2 \\ r = 1 - 2 * \sin^2 \left(\arcsin(\alpha) - \frac{\pi}{4} \right) \end{cases}, \quad (2.14)$$

where c_w and c_h represent the width and height of the minimum bounding rectangles for the ground-truth box and predicted box, respectively. The relevant parameters of Ω are defined in Eq (2.15):

$$\begin{cases} \Omega = 1 - 2 * \sin^2 \left(\arcsin \left(\frac{c_h}{\sigma} \right) - \frac{\pi}{4} \right) \\ \frac{c_h}{\sigma} = \sin(\alpha) \\ \sigma = \sqrt{(b_{cx}^{gt} - b_{cx})^2 + (b_{cy}^{gt} - b_{cy})^2} \\ c_w = \max(b_{cx}^{gt}, b_{cx}) - \min(b_{cx}^{gt}, b_{cx}) \\ c_h = \max(b_{cy}^{gt}, b_{cy}) - \min(b_{cy}^{gt}, b_{cy}) \end{cases}, \quad (2.15)$$

where $\sin(\sigma)$ denotes the sine of the angle between the two bounding boxes. The variable σ represents the Euclidean distance between their centroids. c_h represents centroid deviations: $(b_{cx}^{gt}, b_{cy}^{gt})$ and (b_{cx}, b_{cy}) indicate the center coordinates of the ground-truth box and predicted box, respectively. The improved Bhattacharya loss function was introduced to alleviate the influence of the bounding box scale on the

bounding box relationship. This mechanism improved the accuracy of small object classification and location, and finally realizes the efficient detection of small objects in images. Ablation experiment results of the SSD algorithm with different conventional activation functions for the DIOR dataset are shown in Table 3.

Table 3. Ablation experiment results of the SSD algorithm with different loss functions for the DIOR dataset.

Algorithms	mAP%	Model memory(M)
SSD(IoU)	0.75207	12.60
SSD(GIoU)	0.760806	12.60
SSD(DIoU)	0.765855	12.60
SSD(CIoU)	0.764256	12.60
SSD(SIoU)	0.771066	12.60
SSD+(WIoU)	0.778205	12.60
SSD+(BD_IOU)	0.781225	12.60
SSD+(RBD_IOU)	0.782779	12.60
R3Det	0.781238	12.60
S2ANet	0.780247	12.60

3. Algorithm design and implementation

We addressed the challenges prevalent in high-resolution images, including small target scales ($< 32 \times 32$ pixels).

3.1. Insufficient video memory and inference latency

With the increase in the number of parameters of large models, insufficient video memory and inference latency have become major bottlenecks in deployment. For example, a common experiment is running a ten-billion-parameter model on a 16 GB GPU, where memory usage (the surge in parameters causes video memory demand to exceed hardware capacity) and computational efficiency become critical issues (large-scale matrix operations slow down tasks with high real-time requirements). To address these problems, it is necessary to optimize the model from multiple perspectives, including reducing the total model size, model complexity, pruning, and selecting fine-tuning strategies. Because VGG16 is too simple and ResNet101 is too complex, the VGG16 backbone in SSD was replaced with ResNet50. Our experiments were performed on a Windows 10 system with a dual-core CPU, 256 GB RAM, and one NVIDIA TESLA 100 GPU (32GB VRAM).

3.2. Channel-space cooperative attention injection

The VGG16 backbone in SSD was replaced with a ResNet50 backbone integrated with a Coordinate Attention (CA) module. The computational process of CA can be formally expressed by Eq (3.1):

$$\begin{cases} [t]z_c^h(h) = \frac{1}{W} \sum_{0 < j < w} x_c(h, j), & z_c^w(h) = \frac{1}{H} \sum_{0 < j < H} x_c(j, w) \\ f^h = \delta(F_1(z_h)), & f^w = \delta(F_1(z_w)) \\ g^h = \delta(F_h(f^h)), & g^w = \delta(F_w(f^w)) \\ y_c(i, j) = x_c(i, j)g_c^h(i)g_c^w(j) \end{cases} \quad (3.1)$$

This mechanism enhanced the activation strength in small target regions by capturing position-sensitive feature responses. The improvement was primarily due to the integration of the multi-scale hierarchical feature pyramid (MHFP) module within stages 4 to 6 of the ResNet50 backbone (corresponding to layers 44–46 in the SSD framework). The fusion process of this module is formally expressed in Eq (3.2).

$$F^{fuse} = P_3 \oplus P_4 \oplus P_7, \quad (3.2)$$

where, the operator \oplus denotes channel concatenation. Figure 6 illustrates the lightweight multi-scale fusion framework designed in this study. Within the ResNet50 backbone network, cross-layer feature refinement was implemented across stages 4 to 6 (corresponding to layers 44–46 in the SSD framework). A gated fusion unit (GFU) dynamically modulated the contribution weights across hierarchical levels, thereby enhancing mutual information between features. The modified SSD network architecture is shown in Figure 7.

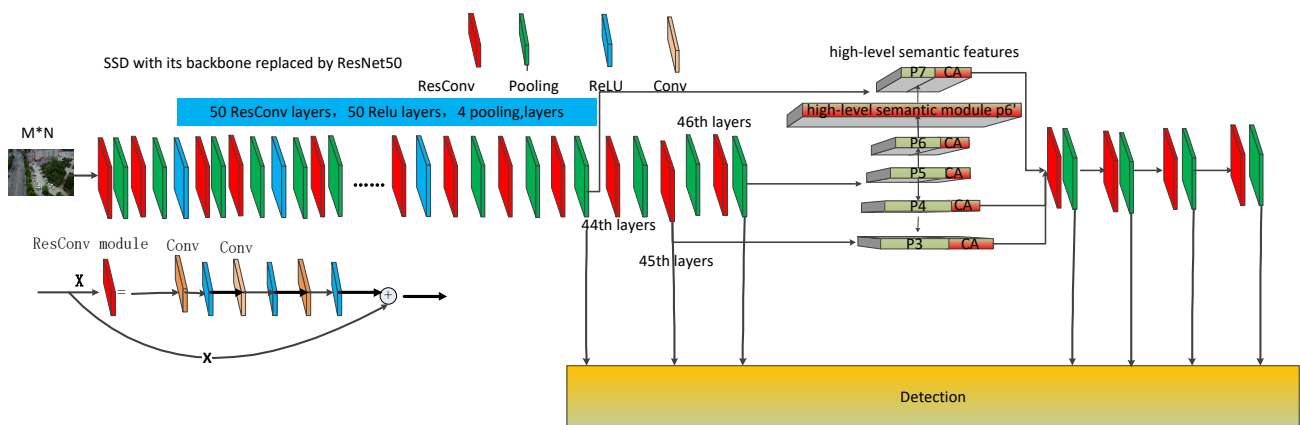


Figure 7. Modified SSD network architecture.

This design achieved non-destructive fusion of cross-scale features through the synergistic interplay between the CAscale hierarchical feature pyramid architecture and the Coordinate Attention mechanism. Consequently, the semantic consistency metric for small object detection was notably improved.

4. Experimental results and analysis

In this section, we detail the comparative experiments performed to evaluate the proposed method. These results optimize the balance between feature representation ability and computing resource allocation in complex background clutter and small target detection scenes, thus providing an effective technical approach for the design of target detection algorithms in aerial images. SSD+1, SSD+2, and improved SSD refer to the algorithms based on the baseline SSD algorithm with the addition of the HSIAM module, the RBD_IoU loss function, and both the HSIAM module and RBD_IoU loss function, respectively.

4.1. Hardware and software configurations

Unless otherwise specified, all experiments utilized SGD with Momentum (momentum coefficient = 0.9) as the optimization algorithm. For models employing a fixed learning rate, the rate was set to 0.01; when learning rate decay was applied, the decay rate was 0.1. Experiments were performed on a Windows 10 system with a dual-core CPU, 256 GB RAM, and one NVIDIA TESLA 100 GPU (32 GB VRAM). The deep learning framework PyTorch (CUDA 10.1) and PyCharm 2023 IDE were used throughout. Mean Average Precision (mAP) was the primary evaluation metric.

4.2. Experimental datasets

These datasets provide a valuable addition to large-scale satellite image repositories. The key characteristics of these datasets are shown in Table 4.

Table 4. Comparative analysis of features in image datasets.

Dataset	Category	Images number	Instances number	Width (px)	Data source	Resolution	Year	Key characteristics
DOTA	15	2806	18822	800	Google Earth, IL-1, GF-2	0.3~1m	2018	multi-sensor, multi-resolution imagery with diverse object categories and significant target deformation cluttered backgrounds, multi-class objects, sensor noise, motion blur, and partial occlusion multi-category, Clean Background
DIOR	20	23463	192472	≥800	Google Earth	0.5~30m	2019	
NWPUHR10	10	800	3775	1000~10001000	Google Earth	0.3~2m	2014	

4.3. Metrics for time and space complexity in training and validation

Based on the above experimental design, in this section, we show the experimental results of SSD in the object detection task, including the comparative analysis of detection accuracy and computational efficiency. Comprehensive evaluation of object detection models typically involves two critical dimensions: (a) Accuracy metrics: Precision, Recall, Average Precision (AP), and includes mean Average Precision (mAP). (b) Computational efficiency metrics: Time and space complexity during training, deployment, and inference phases. AP (Average Precision) includes mAP50 and mAP50:95. Accuracy (ACC) is the proportion of correctly detected instances relative to the total classified instances in the testing set, computed as shown in Eq (4.1).

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (4.1)$$

where, TP shows the number of correctly classified records. FP shows the number of misclassified test data. FN stands for false negative. The AP is calculated as shown in Eq (4.2)

$$AP = \int_0^1 \text{Precision}(R) dR, \quad (4.2)$$

mAP is computed as shown in Eq (4.3):

$$mAP = \frac{1}{N_{cls}} \sum_{c=1}^{N_{cls}} AP_c, \quad (4.3)$$

mAP50 and MAP50:95 represent the mean average precision calculated at IoU thresholds of 0.5 and 0.5 to 0.95, respectively. Recall is a key indicator to evaluate the detection performance and is defined as shown in Eq (4.4). The *Recall* involves the comprehensive detection effect of typical false positives/negatives. For better demonstration, we will give the *Recall* comparison diagrams to illustrate the visualization of typical false positives/negatives.

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\%. \quad (4.4)$$

In the object detection task, F1 score is used as a comprehensive index to evaluate the performance of the model, as shown in Eq (4.5). It balances the ability of the model to minimize false positives (FP) and false negatives (FN).

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (4.5)$$

The validation loss primarily comprises three components: Bounding box regression loss (val/box_loss); objectness loss (val/obj_loss); and classification loss (val/cls_loss). These losses are collectively given by Eq (4.6).

$$\text{loss} = \frac{1}{N} \sum [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]. \quad (4.6)$$

The frame rate (fps) as a real-time evaluation metric were introduced. Temporal and spatial complexity measures in training and validating. Time complexity quantifies the computational workload of a model and can be measured by Floating-point Operations. The time complexity of convolutional networks is formalized in Eq (4.7).

$$\text{Time} \approx O\left(\sum_{d=1}^D M_d^2 \cdot K^2 \cdot C_{d-1} \cdot C_d\right). \quad (4.7)$$

Spatial complexity formally encompasses two components: Total parameter count and output feature maps per layer. Aggregate weight parameters footprints all parametric layers of the model. Memory footprint of output feature maps were computed at each layer during real-time inference, as specified in Eq (4.8).

$$\text{Space} \approx O\left(\sum_{d=1}^D K^2 \cdot C_{d-1} \cdot C_d + \sum_{l=d}^D M^2 \cdot C_d\right), \quad (4.8)$$

where, K is the side length of the convolutional kernel (assuming a square kernel); C_{in} is the number of input feature map channels; C_{out} is the number of output feature map channels; and H_{out} and W_{out} are the height and width of the output feature map, respectively. The factor “2” accounts for one multiplication and one addition (multiply-accumulate operation) per convolution step. $K^2 \times C_{in}$ represents the number of multiply-accumulate operations per spatial position in a single output channel, and $C_{out} \times H_{out} \times W_{out}$ is the total number of spatial positions and channels in the output feature map. The total parameter

count depends solely on the convolutional kernel dimensions, channel numbers, and layer depth, and remains independent of input data size. Floating Point Operations (FLOPs) reflect time complexity. Model parameter count (PARAMs) primarily determines GPU memory requirements. FLOPs directly correlate with GPU computation speed.

4.4. Experimental results and analysis

4.4.1. Experimental results and analysis for the COCO dataset

Since the official has not provided statistical information on the specific number of large, medium, and small targets, as well as the complex background features of the remote sensing image datasets (DIOR, DOTA, and NWPUCHR), the full text adopts the evaluation criterion based on COCO (small target: $\text{area} < 32^2$, medium target: $32^2 < \text{area} < 96^2$, large target: $\text{area} > 96^2$), and the background is relatively complex. Then, we applied our innovation points to the detection task of remote sensing image datasets (DIOR, DOTA, NWPUCHR). The small, medium, and large target mAP indices for the COCO dataset are as shown in Table 5.

Table 5. The mAP50 metrics for small, medium, and large targets for the COCO dataset.

Algorithm	mAP_small	mAP_medium	mAP_large
Faster R-CNN	0.214	0.517	0.675
Faster R-CNN+1	0.226(+1.20%)	0.536(+1.90%)	0.682(+0.70%)
Faster R-CNN+2	0.224(+1.0%)	0.535(+1.80%)	0.685(+1%)
Improved Faster R-CNN	0.235(+2.10%)	0.528(+1.10%)	0.681(+1.60%)
SSD	0.189	0.563	0.712
SSD+1	0.219(+3%)	0.561(+0.20%)	0.714(+0.2%)
SSD+2	0.214(+2.5%)	0.563(+0.0%)	0.715(+0.3%)
Improved SSD	0.231(+4.20%)	0.559(+0.20%)	0.718(+0.60%)
YOLOv8	0.253	0.564	0.726
YOLOv8+1	0.282(+2.9%)	0.568(+0.4%)	0.735(+0.9%)
YOLOv8+2	0.290(+3.7%)	0.570(+0.6%)	0.731(+0.5%)
Improved YOLOv8	0.291(+3.8%)	0.573(+0.9%)	0.739(+1.3%)

From Table 5, it can be seen that the improved scheme of the target detection model has significantly enhanced the performance of small target detection: The improved version of Faster R-CNN increased mAP_{small} by 2.1%; the improved version of SSD achieved a 4.2% increase in mAP_{small} by leveraging the multi-scale feature retention capability of the feature pyramid; the improved version of YOLOv8 increased mAP_{small} by 3.8%. In the detection of medium-sized targets, there is a different performance, with the SSD improved version showing a slight decrease in mAP_{medium} of 0.2%, possibly due to the scale coverage gap of the feature pyramid, while the improved version of YOLOv8 maintained a positive gain of 0.9%. The performance of large target detection remained stable (mAP_{large} fluctuated between 1.3%), but the improved version of SSD showed a 0.6% decrease in mAP_{large} due to insufficient resolution of deep features for large target edge positioning. It is worth noting that Faster R-CNN had a limited improvement for medium-sized targets (+1.1%), revealing a structural defect of the RPN network's insufficient sensitivity to

medium-sized anchor boxes. The experimental data indicated that the training of the target detection models for the COCO dataset showed a significant regularity. The mAP50 indicators of all models (including Faster R-CNN, SSD, YOLOv8, and their improved versions) showed a stable upward trend as the training rounds increased. The Recall of the improved algorithm incorporating the innovation points of this study, was tested for the COCO dataset, are as shown in Figure 8.

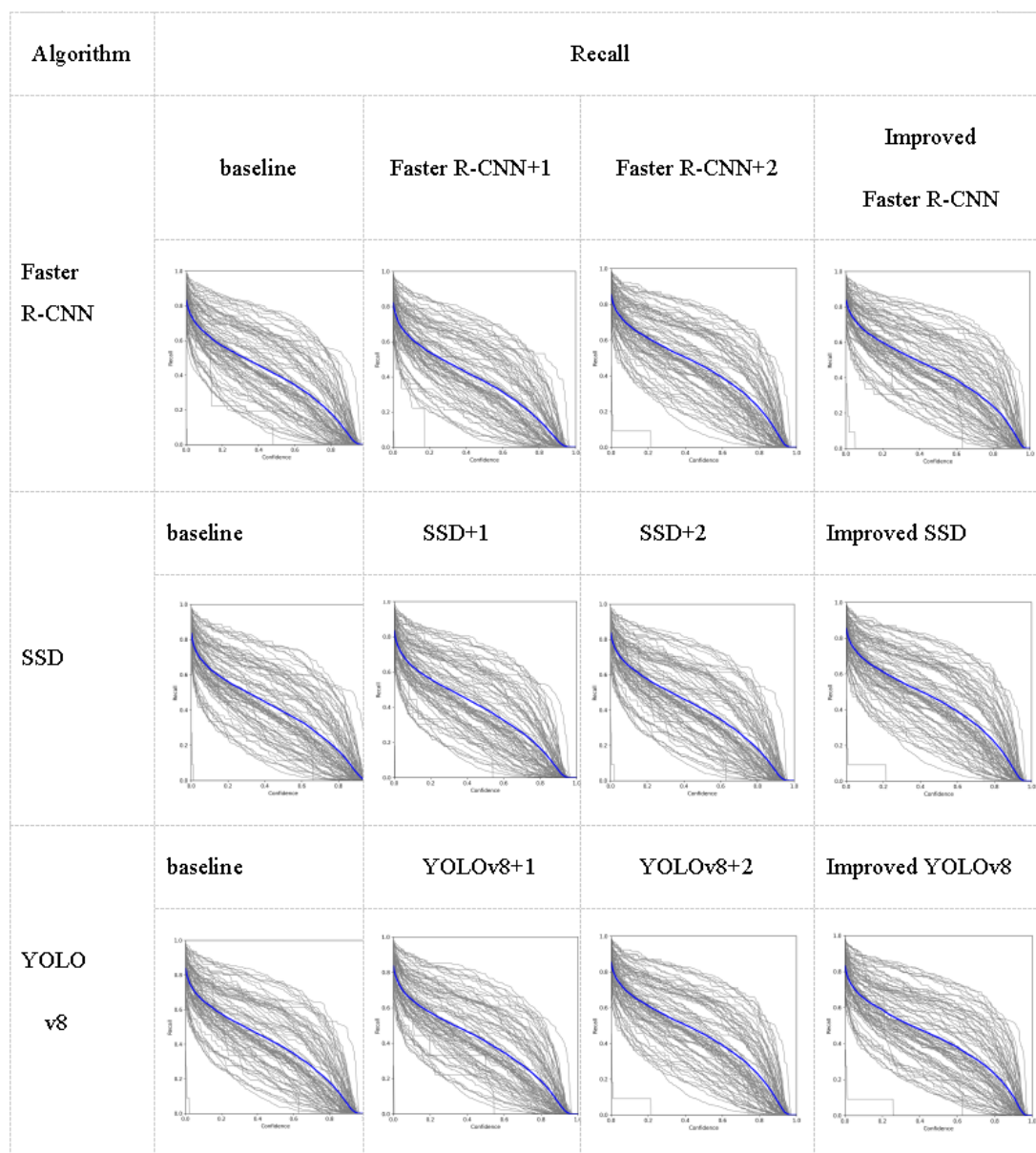


Figure 8. Comparative Recall on the COCO dataset before and after enhancement.

The important results of the detection for the COCO dataset are shown in Figure 9. It can be seen that some small targets were detected, such as sofas and birds on the beach.













Algorithm	Detection comparison			
Faster R-CNN	baseline	Faster R-CNN+1	Faster R-CNN+2	Improved Faster R-CNN
				
SSD	baseline	SSD+1	SSD+2	Improved SSD
				
YOLO v8	baseline	YOLOv8+1	YOLOv8+2	Improved YOLOv8
				

Figure 9. Comparative detection results of the detection for the COCO dataset before and after enhancement.

The improved algorithm can detect more small targets for the complex background COCO dataset, which was verified the effectiveness of the algorithm. Therefore, it was applied to the target detection task in remote sensing images.

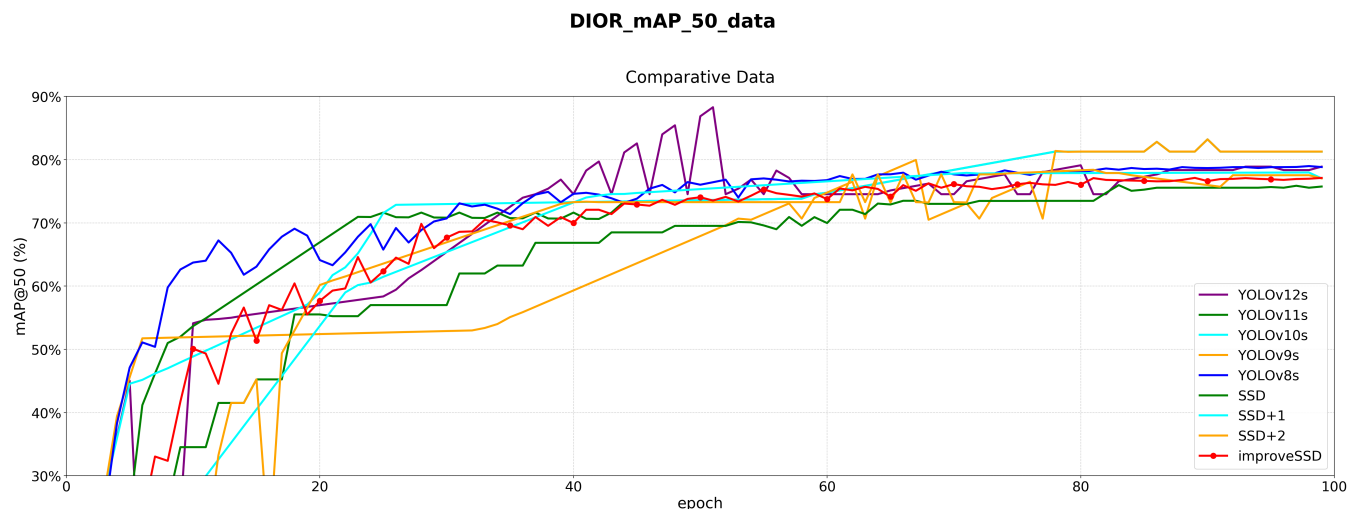
4.4.2. Experimental Results and Analysis for the DIOR Dataset

The DIOR dataset contained 23463 images (800×800 pixels) with 192,472 annotation instances using axis-aligned bounding boxes. Table 6 lists the comparative results of all evaluated models.

Table 6. Comparative experimental results for the DIOR dataset.

Algorithms	mAP50	mAP50:95	val/box_loss	val/obj_loss	val/cls_loss	FLOPs (G)	Time (h)	Model memory (M)	Recall@ mAP-50	F1 Score	@Confidence	fps
YOLOv8s	0.788321	0.531132	0.030614	0.019682	0.005518	22.40	23.54	13.80	0.790	0.80	0.360	137
YOLOv9s	0.789347	0.541137	0.030674	0.019635	0.005619	21.70	22.57	12.54	0.789	0.80	0.362	127
YOLOv10s	0.789201	0.541184	0.030610	0.027642	0.005628	21.60	22.52	12.80	0.779	0.80	0.364	126
YOLOv11s	0.789329	0.541129	0.030610	0.020962	0.005478	20.92	22.53	12.40	0.769	0.80	0.367	124
YOLOv12s	0.791214	0.541225	0.036154	0.020683	0.005587	21.40	20.58	9.10	0.670	0.48	0.260	145
SSD	0.788500	0.572007	0.019620	0.001270	0.008705	19.60	15.67	12.60	0.771	0.79	0.348	99
SSD+1	0.788802	0.572458	0.019614	0.001233	0.008703	19.60	15.67	12.60	0.772	0.78	0.347	99
SSD+2	0.788850	0.572387	0.019625	0.001242	0.008704	19.60	15.67	12.60	0.781	0.79	0.348	99
improved SSD	0.791180	0.581132	0.032495	0.019185	0.004816	19.60	15.85	12.60	0.789	0.80	0.487	99

The improved algorithm obtained optimal performance in the metric mAP50 (0.791180), followed by YOLOv8s (0.788321). It also obtained results higher than mAP50:95 (0.581132) compared to YOLOv8s (0.531132) or YOLOv12s (0.541225). Compared with YOLOv12s, the improved SSD algorithm did not have significant advantages in the metric mAP50, but mAP50:95 was achieved an improvement(there is an increase of 4 or 5 percentage points). Lower loss values in all measurement categories indicated better loss management in model training and validation for SSD improvement. The enhanced SSD showed excellent performance across multiple metrics (mAP50, mAP50:95, and various functions), while maintaining computational resource consumption and balanced inference time. In addition, it offered competitive results in model evaluation metrics (Recall, F1 Score, and Confidence) and showed higher overall performance against comparative algorithms such as Faster RCNN, YOLOv8s, SSD, and their variants. The mAP50 is visualized in Figure 10.

**Figure 10.** Comparative mAP50 performance improvements of three algorithms for the DIOR dataset before and after enhancement.

This indicator reflected the recovery rate below the mAP50 threshold. The upgraded SSD reached 0.791180 in this measurement, showing strong performance, while base SSD reached 0.788500, that is, less efficient. This value balances accuracy and recovery rate as a whole. The F1 score of the upgraded SSD is 0.487, which was better than algorithms such as YOLOv8s(0.36) or YOLOv12s(0.26) , which meant that higher computational requirements were required. In contrast, SSD variants (SSD+1, SSD+2) and upgraded SSDs were significantly smaller in model memory. The running time (time

per hour) reflected the duration of the algorithm execution, and YOLOv8s had a running time of over 23.54 hours, while SSD and improved variants of SSD showed a significant reduction in computational overhead, and running times between 12.60 and 15.67 hours. YOLOv8s achieved higher performance for mAP50 and MAP50:95 metrics, showing higher detection accuracy. In contrast, the enhanced SSD performed well in the confidence limit on comprehensive metrics such as Recall@AP50 and F1 scores, maintained a competitive recovery rate, and ensured an overall performance of accuracy and balance. Both the SSD architecture and the enhanced SSD outperformed YOLOv8s and YOLOv12s in terms of complex computations (FLOPs) and efficient runtime. This advantage made it the best solution for resource-constrained applications that require urgent deployment. The recall comparison diagrams for DIOR dataset are shown in Figure 11.

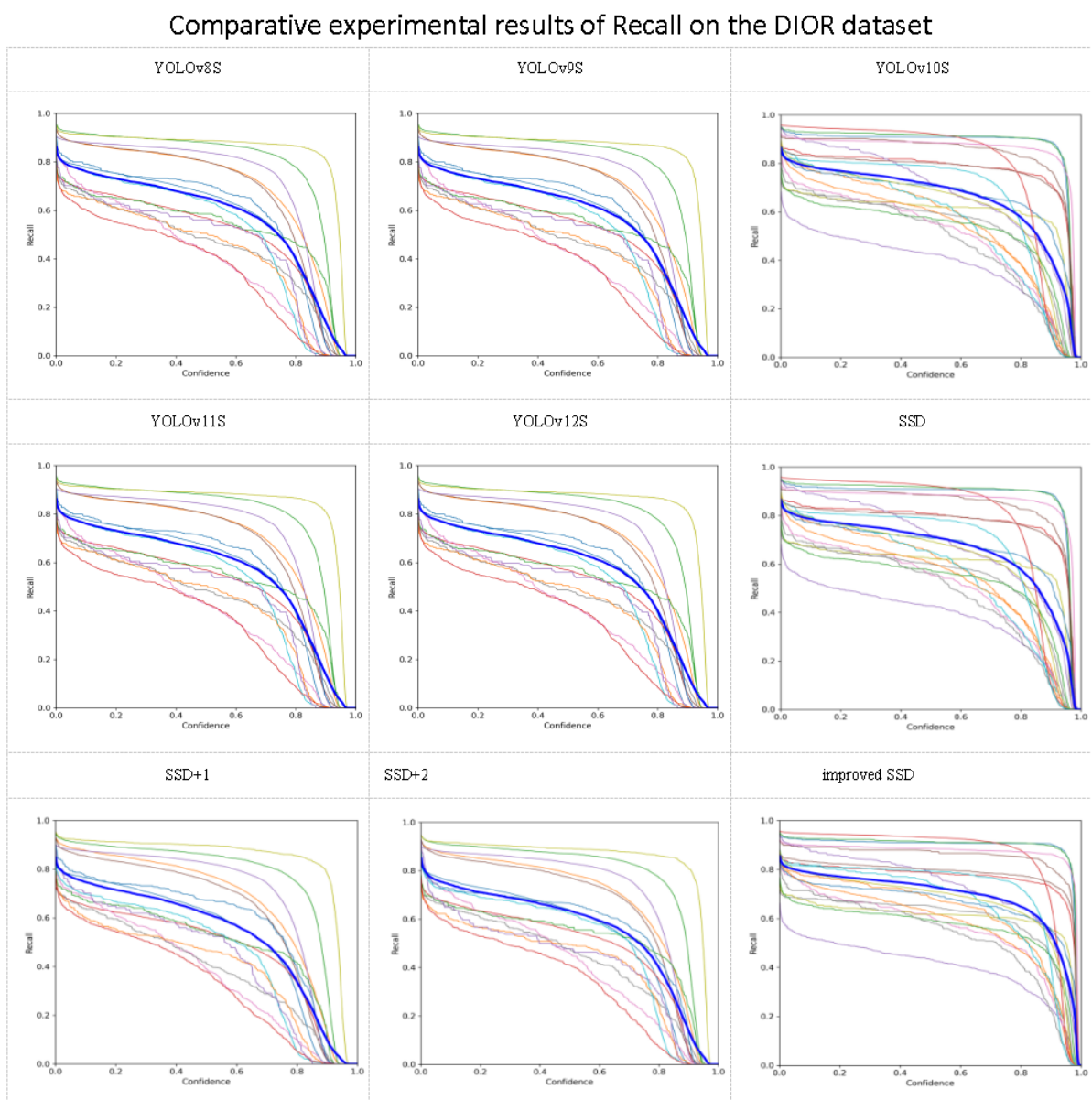


Figure 11. Recall performance comparison for DIOR dataset.

22721

Detection performance comparison on DIOR dataset

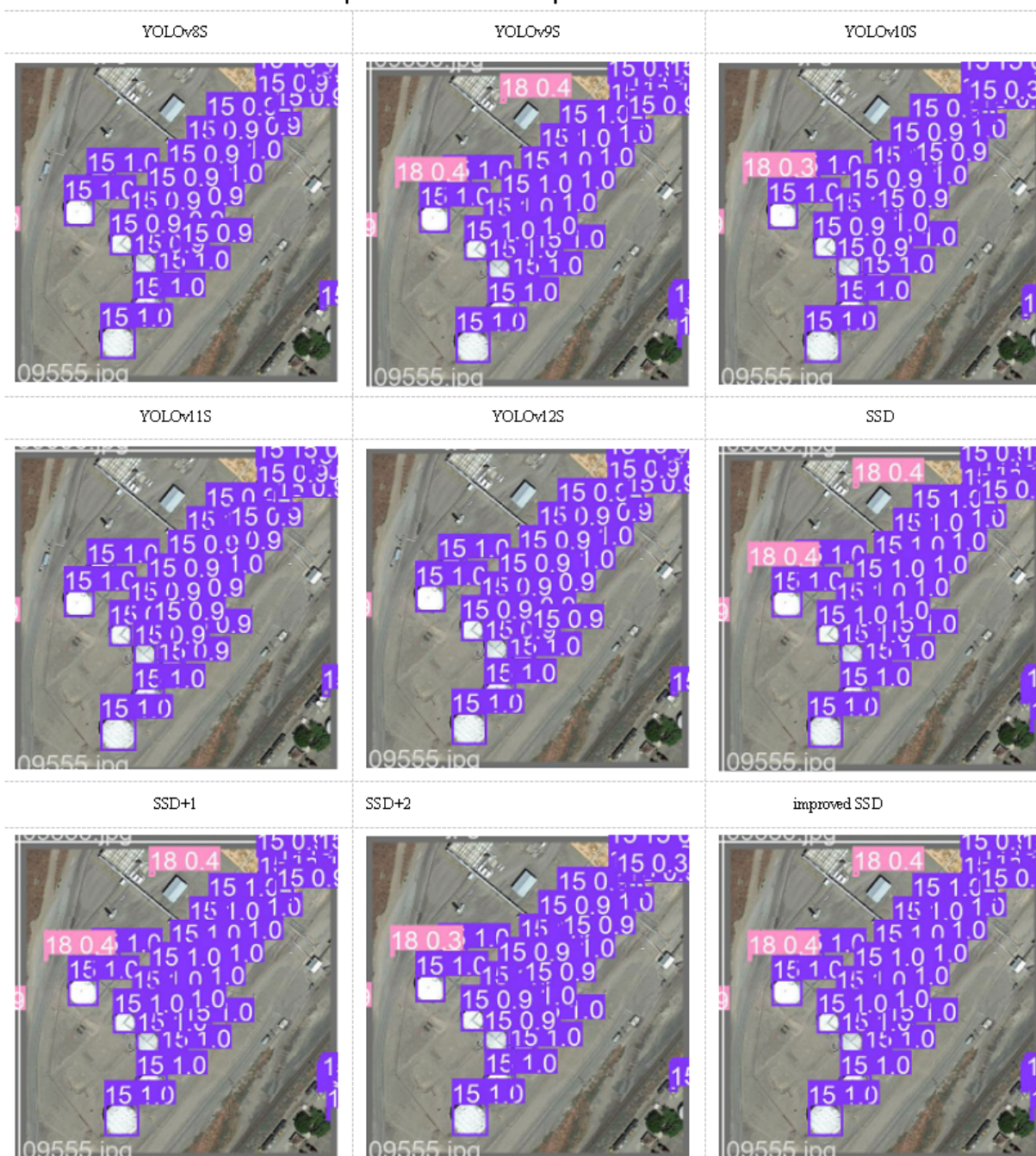


Figure 12. Detection performance comparison for the DIOR dataset.

4.4.3. Experimental results and analysis for the DOTA dataset

The 15 object categories for the DOTA dataset included baseball fields, track and field, vehicles, tennis courts, basketball courts, warehouses, soccer fields, roadblocks, swimming pools, helicopters, Bridges, ports, ships, and airplanes. As a benchmark dataset for object detection in aerial images, the difficulty of DOTA in confusing background information and object features was mainly due to its reasons. Complex stage characteristics in the data: These factors often cause background information to be incorporated into the target feature representation, which in turn violates detection accuracy. Table 7 shows the comparative experimental information for the DOTA dataset.

Table 7. Comparative experimental results for the DOTA dataset.

Algorithms	mAP50	mAP50:95	val/box_loss	val/obj_loss	val/cls_loss	FLOPs (G)	Time (h)	Model memory (M)	Recall@ mAP-50	F1 Score	@Confidence	fps
YOLOv8s	0.795793	0.546474	0.020446	0.001562	0.011800	14.82	22.45	23.53	0.690	0.718	0.468	104
YOLOv9s	0.782179	0.557857	0.027954	0.001828	0.020852	18.50	20.67	10.17	0.648	0.708	0.297	119
YOLOv10s	0.801795	0.568856	0.026457	0.001869	0.021654	20.30	20.58	10.18	0.654	0.718	0.285	115
YOLOv11s	0.817954	0.568517	0.020458	0.001622	0.021750	20.60	20.77	10.22	0.656	0.672	0.286	109
YOLOv12s	0.821794	0.567851	0.023487	0.001762	0.021850	21.40	20.57	9.17	0.646	0.678	0.287	129
SSD	0.769170	0.529155	0.028227	0.021059	0.016720	12.61	21.45	10.58	0.689	0.720	0.386	97
SSD+1	0.769171	0.529156	0.028222	0.021057	0.016730	12.61	21.45	10.58	0.689	0.702	0.487	97
SSD+2	0.769174	0.529154	0.028220	0.021053	0.016700	12.61	21.45	10.58	0.689	0.712	0.586	97
Improved SSD	0.771181	0.531130	0.028291	0.021137	0.016170	12.61	21.45	10.58	0.690	0.713	0.668	97

YOLOv8s leads in the mAP50:90 metric with a value of 0.546474, followed by the improved SSD and SSD algorithm series. Faster R-CNN achieved the lowest performance at 0.508348. The corresponding convergence behavior is detailed in Figure 13. Compared with YOLOv12s, the improved SSD algorithm did not have significant advantages, but mAP50:95 achieved an improvement.

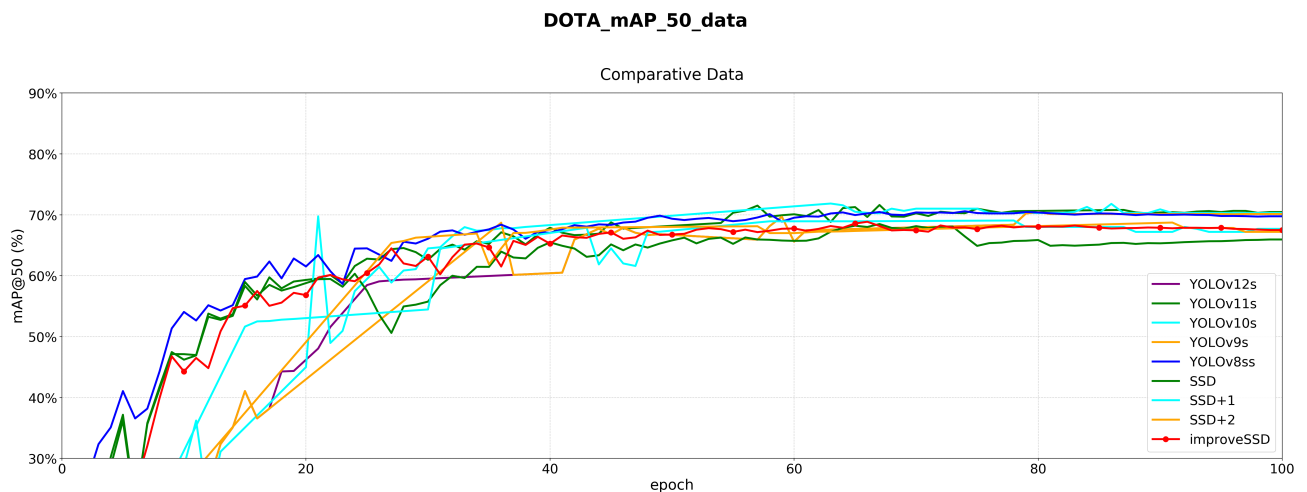


Figure 13. Comparative mAP50 performance of three algorithms before and after improvements for the DOTA dataset.

All algorithms showed comparable performance on the three loss metrics. YOLOv8s achieved a relatively low validation bounding box loss val/box_loss of 0.020446. The differences between the different algorithms in validation object loss (val/obj_loss) and validation classification loss

(val/cls_loss) were small, indicating that the algorithms had similar convergence behavior and loss control ability during optimization. The SSD family (including SSD, SSD+1, SSD+2, and improved SSD) had significantly lower computational requirements, with failures consistently around 0.001673G. In contrast, YOLOv12s(0.021850) and YOLOv8s (0.00180) had higher demand for computing resources, highlighting the efficiency advantage of the SSD-based architecture. The training time of all algorithms was between 20.57–22.45 hours. The training time for the ssd family was slightly faster, but it was comparable for all algorithms. YOLOv8s provided superior object detection performance for the DOTA dataset, with significant advantages in detection accuracy and composite evaluation scores. The ssd variant shows computational efficiency balanced against competitive training time and memory footprint. Among them, the improved SSD achieved particularly good accuracy. The *Recall* comparison diagrams for the DOTA dataset are shown in Figure 14.

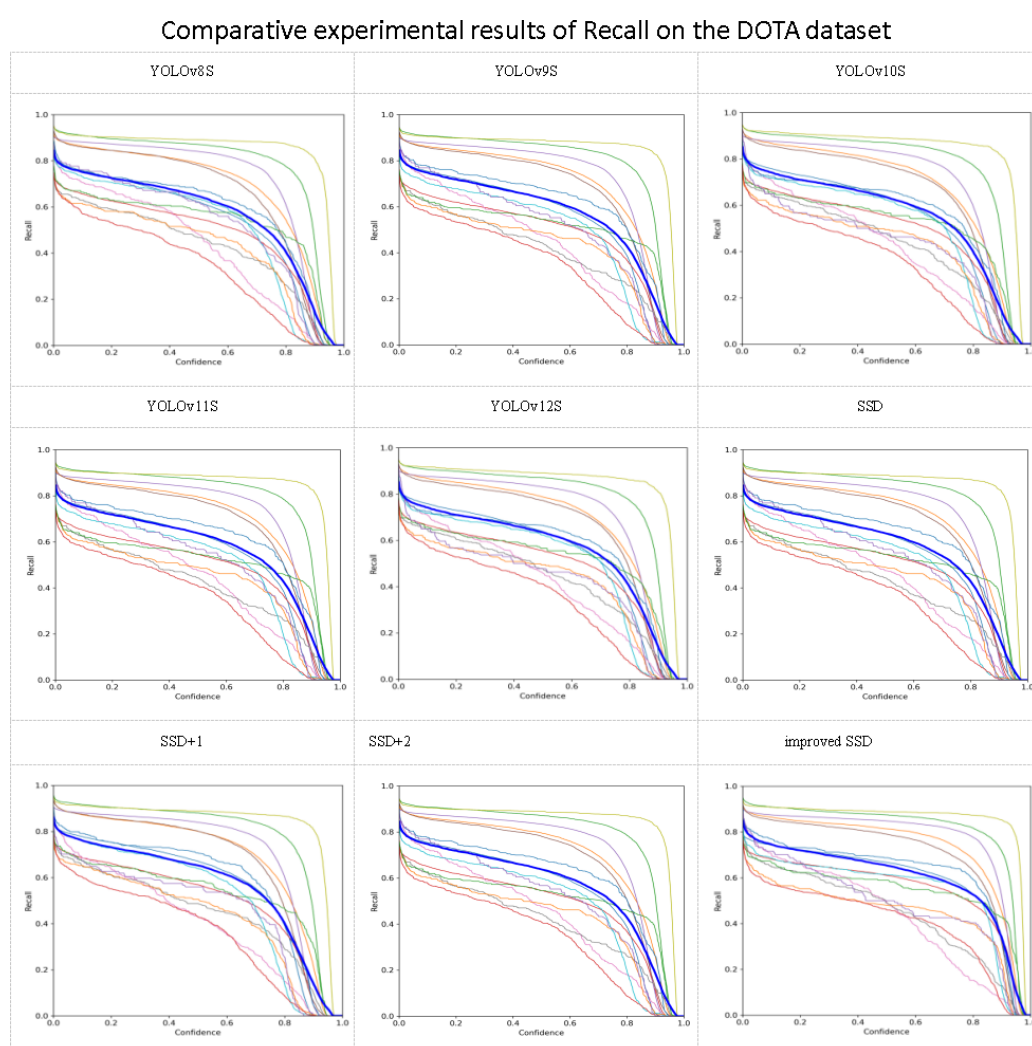
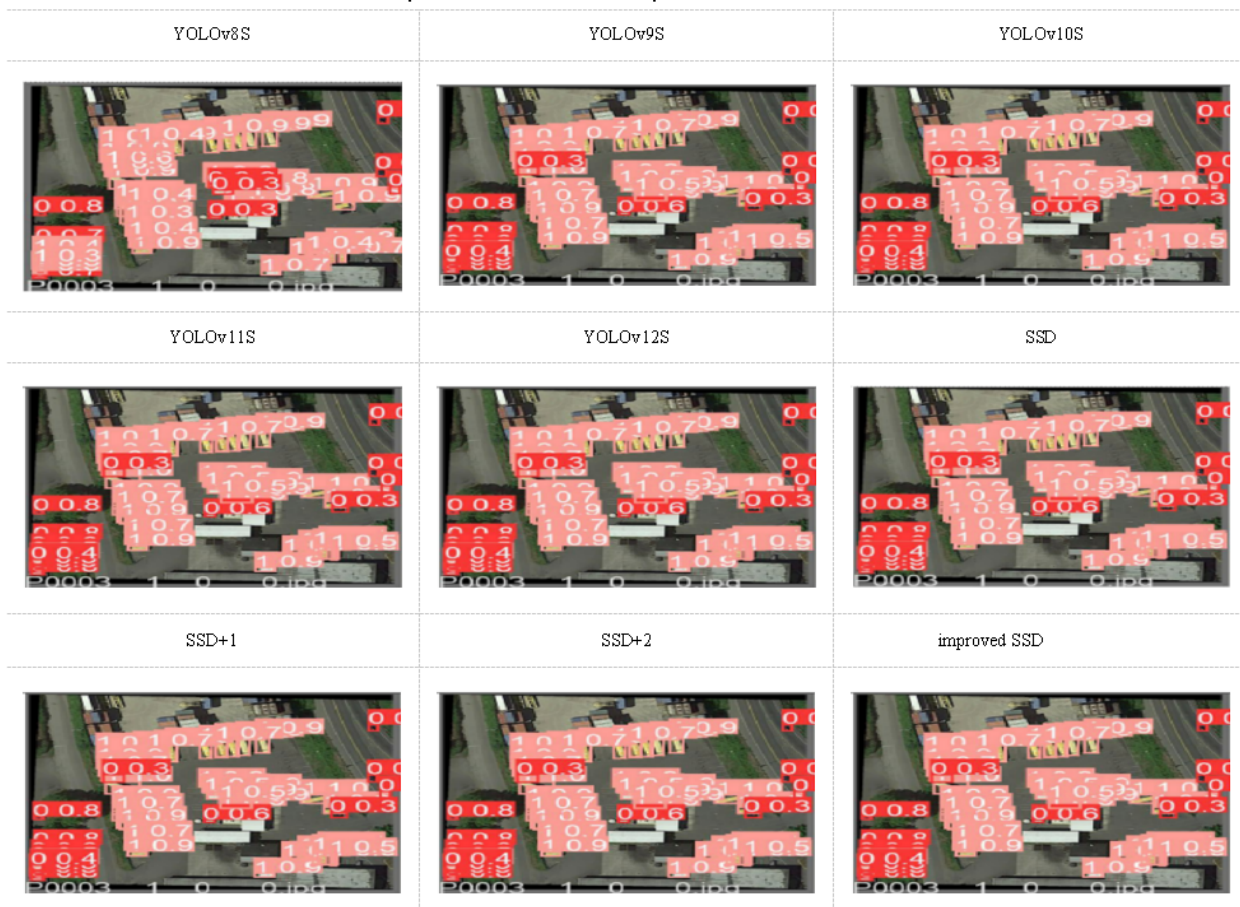


Figure 14. Recall performance comparison on DOTA dataset.

Qualitative results on the DOTA dataset are shown in Figure 15.

Detection performance comparison on DOTA dataset



6

Figure 15. Detection performance comparison for the DOTA dataset.

NWPUCHR_mAP_50_data

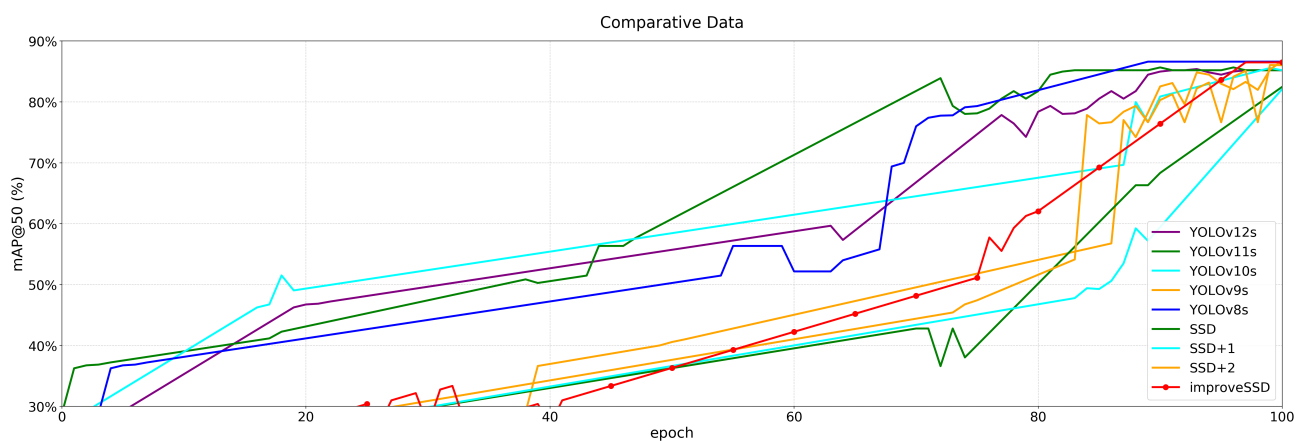


Figure 16. mAP50 performance comparison of three algorithms before and after modification for the NWPUCHR dataset.

Comparative experimental results of Recall on the NWPUCHR dataset

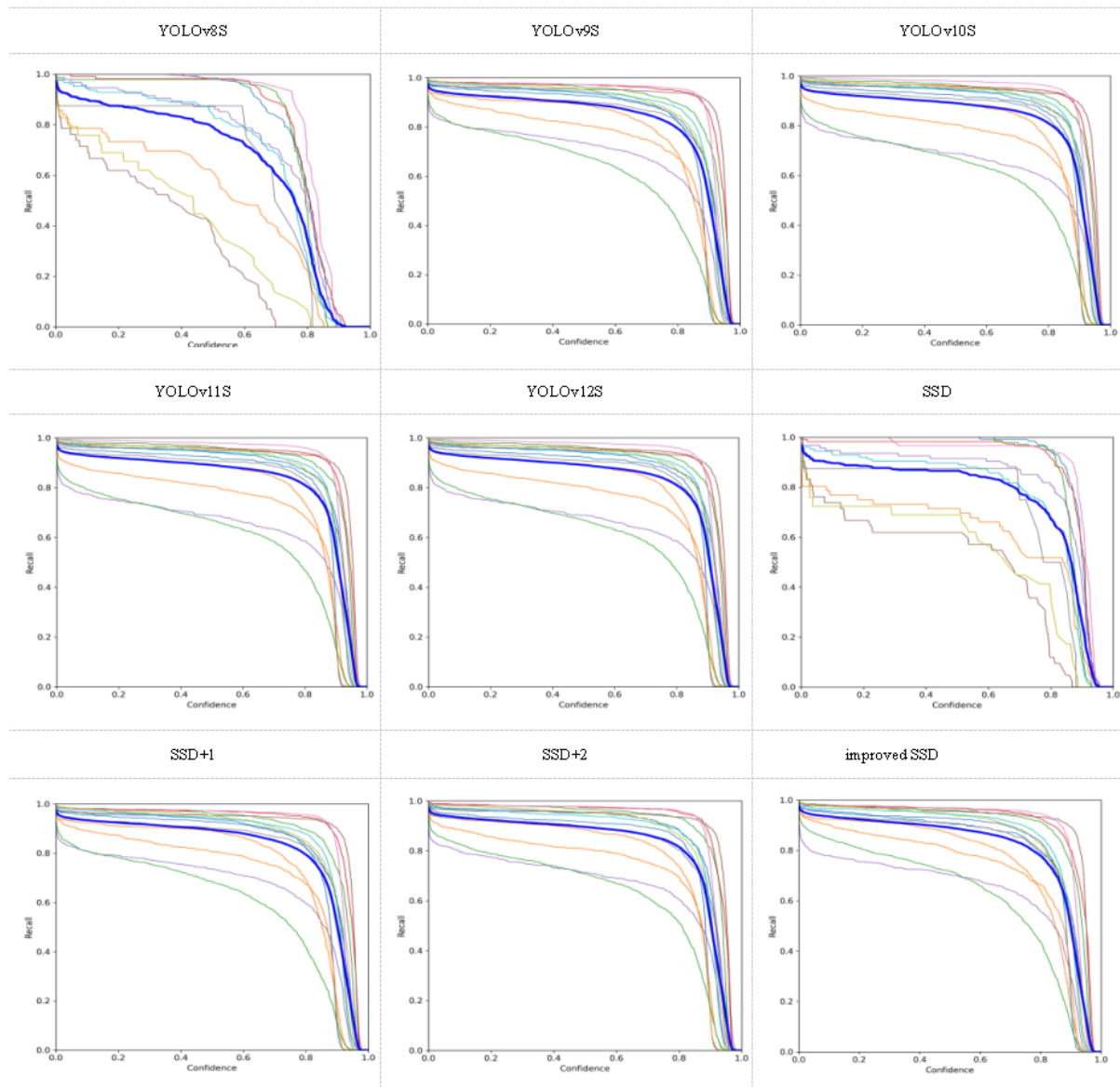


Figure 17. Recall performance comparison on NWPUCHR dataset.

4.4.4. Experimental results and analysis for the NWPUCHR dataset

In order to objectively verify the effectiveness of the proposed algorithm, the summary of key performance indicators for the NWPUCHR dataset is shown in Table 8.

The mAP50 of YOLOv8s (0.848060) and improved SSD (0.830940) demonstrated superior performance, while SSD(0.811908) scored comparatively lower. Qualitative results for the NWPUCHR dataset are shown in Figure 16. The mAP50:95 of the improved SSD achieved the lowest value (0.290161), whereas YOLOv12s (0.267857) and YOLOv8s (0.243712) attained higher results. YOLOv12s (22.40 hours) and SSD (15.51 hours) showed shorter training durations, whereas YOLOv8s consumed the longest time (23.53 hours). YOLOv8s utilized the most memory (13.80

MB), whereas SSD (12.60 MB) and its variants exhibited lower usage. The F1 Score of YOLOv8s (0.831) and improved SSD (0.810) delivered strong results, while YOLOv12s (0.828) underperformed. Recall@mAP50 of YOLOv8 achieved the highest recall (0.860), significantly surpassing YOLOv12s (0.795). Improved SSD delivered superior comprehensive performance for object detection tasks. Compared with YOLOv12s, the improved SSD algorithm did not have significant advantages in the metric mAP50, but mAP50:95 achieved an improvement (there is an increase of 3 or 5 percentage points).

The recall comparison diagrams for the NWPUCHR dataset are shown in Figure 17.

Experimental results for the NWPUCHR dataset are illustrated in Figure 18.

The improved SSD exhibited balanced performance across all metrics, maintaining low computational requirements and time consumption, while providing competitive results, thus demonstrating clear practical advantages.



Figure 18. mAP50 performance comparison of three algorithms before and after modifications for the NWPUCHR dataset.

Table 8. Experimental comparison results for the NWPUCHR dataset.

Algorithms	mAP50	mAP50:95	val/box_loss	val/obj_loss	val/cls_loss	FLOPs (G)	Time (h)	Model memory (M)	Recall@ mAP-50	F1 Score	@Confidence	fps
YOLOv8s	0.848060	0.243712	0.042503	0.024823	0.006742	22.40	23.53	13.80	0.860	0.831	0.480	112
YOLOv9s	0.850805	0.253700	0.048527	0.025082	0.006904	20.40	21.50	11.70	0.760	0.830	0.488	120
YOLOv10s	0.850062	0.250237	0.049509	0.025825	0.006844	20.42	21.50	11.78	0.786	0.835	0.488	122
YOLOv11s	0.850090	0.251371	0.049503	0.021923	0.006142	20.40	21.54	12.01	0.783	0.834	0.491	122
YOLOv12s	0.852194	0.267857	0.049425	0.021772	0.004850	19.50	20.64	09.47	0.795	0.828	0.497	135
SSD	0.811908	0.470510	0.046457	0.006083	0.005941	15.51	16.35	12.60	0.812	0.750	0.720	97
SSD+1	0.820810	0.370512	0.040468	0.006777	0.005910	15.51	16.35	12.60	0.843	0.805	0.716	97
SSD+2	0.824000	0.400514	0.041457	0.006079	0.006590	15.51	16.35	12.60	0.814	0.803	0.714	97
improved SSD	0.830940	0.290161	0.051645	0.028569	0.018621	15.52	16.47	12.60	0.855	0.810	0.721	98

5. Conclusions

Addressing the core challenge of low feature distinguishability between targets and backgrounds in remote sensing images, we developed a multi-scale hierarchical feature pyramid based on coordinate attention mechanisms, enhancing discriminative representation of small targets through cross-layer feature fusion. We introduced a bounding box regression loss function utilizing modified Bhattacharyya distance to optimize localization accuracy. Experimental validation demonstrated that the proposed approach significantly improved detection performance while maintaining computational efficiency. These advancements exhibited substantial practical utility in critical engineering applications such as harbor vessel identification and urban traffic surveillance.

Author contributions

Chao Chen: Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing—original draft preparation, Writing—review and editing, Visualization, Supervision, Project administration, Funding acquisition; Bin Wu: Conceptualization. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare we have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by Research on Stochastic Sampling Synchronization Control for Reaction Diffusion Neural Networks with Probability Distributions (No.2024ZYD0025), Natural Science Foundation of Sichuan Province (2025ZNSFSC0079), Research on optimization of oilfield well pattern and production system based on data and knowledge (No.2023QYY04).

Conflict of interest

There is no conflict of interest/competing interests. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this study. There are no financial or non-financial (e.g., professional, personal, or institutional) conflicts of interest related to the research, authorship, or publication of this article. All

authors confirm that this statement accurately reflects their current circumstances and relationships, and they have no obligations or commitments that might compromise the objectivity, integrity, or transparency of the work.

References

1. K. Ding, Z. Ding, Z. Zhang, M. Yuan, G. Ma, G. Lv, Scd-yolo: A novel object detection method for efficient road crack detection, *Multimedia Syst.*, **30** (2024), 351. <http://doi.org/10.1007/s00530-024-01538-y>
2. P. Huangfu, L. Dang, A multi-scale pyramid feature fusion-based object detection method for remote sensing images, *Int. J. Remote Sens.*, **44** (2024), 7790–7807. <http://doi.org/10.1080/01431161.2023.2288947>
3. L. Xu, Y. Zhao, Y. Zhai, L. Huang, C. Ruan, Small Object Detection in UAV Images Based on YOLOv8n, *Int. J. Comput. Intell. Syst.*, **17** (2024), 223. <http://doi.org/10.1007/s44196-024-00632-3>
4. J. Han, D. Zhang, G. Cheng, L. Guo, J. Ren, Object Detection in Optical Remote Sensing Images Based on Weakly Supervised Learning and High-Level Feature Learning, *IEEE Trans. Geosci. Remote Sens.*, **53** (2015), 3325–3337. <http://doi.org/10.1109/TGRS.2014.2374218>
5. V. Zermatten, J. Castillo-Navarro, D. Marcos, D. Tuia, Learning transferable land cover semantics for open vocabulary interactions with remote sensing images, *ISPRS J. Photogramm. Remote Sens.*, **220** (2025), 621–636. <http://doi.org/10.1016/j.isprsjprs.2025.01.006>
6. Z. Liu, X. Wu, L. Zhang, P. Yu, LightYOLO-S: A lightweight algorithm for detecting small targets, *J. Real-Time Image Process.*, **21** (2024), 111. <http://doi.org/10.1007/s11554-024-01485-x>
7. H. Wang, H. Qian, S. Feng, Ssd-kdgan: A lightweight SSD target detection method based on knowledge distillation and generative adversarial networks, *J. Supercomput.*, **80** (2024), 23544–23564. <http://doi.org/10.1007/s11227-024-06361-w>
8. S. Li, F. Yan, Y. Liu, Y. Shen, L. Liu, K. Wang, A multi-scale rotated ship targets detection network for remote sensing images in complex scenarios, *Sci. Rep.*, **15** (2025), 170–183. <http://doi.org/10.1038/s41598-025-86601-y>
9. X. Yuan, Q. Chen, J. Li, S. Gong, C. Lin, X. Hu, YOLOv5-LC: Enhancing vehicle detection for evening rushing hour, *J. Trans. Eng., Part A: Syst.*, **151** (2025), 4025057. <https://doi.org/10.1061/JTEPBS.TEENG-8652>
10. Z. J. Khaw, Y.-F. Tan, H. A. Karim, H. A. A. Rashid, Improved YOLOv8 Model for a Comprehensive Approach to Object Detection and Distance Estimation, *IEEE Access*, **12** (2024), 63754–63782. <http://doi.org/10.1109/ACCESS.2024.3396224>
11. N. Singh, C. P. Maurya, B. Mahaur, S. K. Singh, Improved YOLOv11 with weights pruning for road object detection in rainy environment, *SIViP*, **19** (2025), 473. <http://doi.org/10.1007/s11760-025-04070-2>
12. J. Xu, L. Kanokphan, K. Tasaka, Fast and accurate object detection using image cropping/resizing in multi-view 4K sports videos, *Proceedings of the 1st International Workshop on Multimedia Content Analysis in Sports (MMSports'18)*, 2018. <https://doi.org/10.1145/3265845.3265852>

13. C. S. Parvathy, J. P. Jayan, Automatic Lung Cancer Detection Using Computed Tomography Based on Chan Vese Segmentation and SENET, *Opt. Mem. Neural Networks*, **33** (2024), 339–354. <http://doi.org/10.3103/S1060992X2470022X>
14. Q. Feng, M. Fu, Z. Yao, Y. Liu, T. Liang, Research on small-scale foreign object intrusion detection algorithm for railway tracks based on improved YOLOv8, *Modern Electron. Tech.*, **48** (2025), 174–179. <http://doi.org/10.16652/j.issn.1004-373x.2025.11.027>
15. S. Ding, W. Jing, H. Chen, C. Chen, Yolo Based Defects Detection Algorithm for EL in PV Modules with Focal and Efficient IoU Loss, *Appl. Sci.*, **14** (2024), 7493. <http://doi.org/10.3390/app14177493>
16. J. R. Yang, Y. N. Qin, T. X. Li, H. Zhuang, Underground helmet detection algorithm based on improved YOLOv8s, *Saf. Coal Mines*, **56** (2025), 221–228. <http://doi.org/10.13347/j.cnki.mkaq.20241167>
17. K. Rabia, E. Alperen, Real-time multi-object detection and tracking in UAV systems: Improved YOLOv11-EFAC and optimized tracking algorithms, *J. Real-Time Image Proc.*, **22** (2025), 178. <http://doi.org/10.1007/s11554-025-01758-z>
18. P. Sharma, I. Malhotra, P. Handa, N. Goel, Real-time detection of household objects using single-shot detection with mobileNet, *Artificial Intelligence and Speech Technology 2024*, 2025, 104–117. http://doi.org/10.1007/978-3-031-91340-2_9
19. X. Zhong, CAL-SSD: Lightweight SSD object detection based on coordinated attention, *Signal, Image Video Process.*, **19** (2025), 31. <http://doi.org/10.1007/s11760-024-03716-x>
20. J. Lei, W. Yang, R. Yang, A Deep Learning Method for Automated Site Recognition of Nasopharyngeal Endoscopic Images, *J. Med. Bio. Eng.*, **45** (2025), 240–251. <http://doi.org/10.1007/s40846-025-00936-5>
21. Y. Hou, Y. Rao, H. Song, H. Song, Z. Nie, T. Wang, et al., A Rapid Detection Method for Wheat Seedling Leaf Number in Complex Field Scenarios Based on Improved YOLOv8, *Smart Agric.*, **6** (2024), 128–137. <http://doi.org/10.12133/j.smartag.SA202403019>
22. J. Li, Q. Hou, J. Xing, J. Ju, SSD object detection model based on multi-frequency feature theory, *IEEE Access*, **8** (2020), 82294–82305. <http://doi.org/10.1109/ACCESS.2020.2990477>
23. W. Zheng, W. Tang, S. Chen, L. Jiang, C. Fu, CIA-SSD: Confident iou-aware single-stage object detector from point cloud, *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, **35** (2021), 3555–3562. <https://doi.org/10.1609/aaai.v35i4.16470>
24. L. Gong, X. Huang, Y. Chao, J. Chen, B. Lei, An enhanced SSD with feature cross-reinforcement for small-object detection, *Appl. Intell.*, **53** (2023), 19449–19465. <http://doi.org/10.1007/s10489-023-04544-1>
25. L. Lin, H. Zhao, S. Gao, J. Wang, Z. Zhang, Spatial-Spectral Linear Extrapolation for Cross-Scene Hyperspectral Image Classification, *Remote Sens.*, **17** (2025), 1816. <http://doi.org/10.3390/rs17111816>
26. F. Guo, Z. Li, G. Ren, L. Wang, J. Zhang, J. Wang, Instance-Wise Domain Generalization for Cross-Scene Wetland Classification With Hyperspectral and LiDAR Data, *IEEE Trans. Geosci. Remote Sens.*, **63** (2025). <http://doi.org/10.1109/TGRS.2024.3519900>

27. A. Sarkar, U. Nandi, B. Paul, S. Kr. Ghosal, M. M. Singh, J. K. Mandal, et al., Searching Optimizers for Deep Learning Based Hyperspectral Image Classification, *Computational Technologies and Electronics. ICCTE 2023*, 2025. https://doi.org/10.1007/978-3-031-81935-3_7
28. W. W. Y. Ng, Q. Zhang, C. Zhong, J. Zhang, Improving domain generalization by hybrid domain attention and localized maximum sensitivity, *Neural Networks*, **171** (2024), 320–331. <http://doi.org/10.1016/j.neunet.2023.12.014>
29. H. Zhou, A. Liu, C. Zhang, P. Zhu, Q. Zhang, M. Kankanhalli, Multi-Modal Meta-Transfer Fusion Network for Few-Shot 3D Model Classification, *Int. J. Comput. Vis.*, **132** (2024), 673–688. <http://doi.org/10.1007/s11263-023-01905-8>
30. Z. Zhang, D. Gao, D. Liu, G. Shi, Spectral-Spatial Domain Attention Network for Hyperspectral Image Few-Shot Classification, *Remote Sens.*, **16** (2024), 22–35. <http://doi.org/10.3390/rs16030592>



AIMS Press

©2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)