applied sciences

MDPI

*Article*

# YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition

**Tianyong Wu and Youkou Dong ***

College of Marine Science and Technology, China University of Geosciences, Wuhan 430074, China; wutianyong@cug.edu.cn
* Correspondence: 1202221660@cug.edu.cn

**Abstract:** Object detection remains a pivotal aspect of remote sensing image analysis, and recent strides in Earth observation technology coupled with convolutional neural networks (CNNs) have propelled the field forward. Despite advancements, challenges persist, especially in detecting objects across diverse scales and pinpointing small-sized targets. This paper introduces YOLO-SE, a novel YOLOv8-based network that innovatively addresses these challenges. First, the introduction of a lightweight convolution SEConv in lieu of standard convolutions reduces the network's parameter count, thereby expediting the detection process. To tackle multi-scale object detection, the paper proposes the SEF module, an enhancement based on SEConv. Second, an ingenious Efficient Multi-Scale Attention (EMA) mechanism is integrated into the network, forming the SPPFE module. This addition augments the network's feature extraction capabilities, adeptly handling challenges in multi-scale object detection. Furthermore, a dedicated prediction head for tiny object detection is incorporated, and the original detection head is replaced by a transformer prediction head. To address adverse gradients stemming from low-quality instances in the target detection training dataset, the paper introduces the Wise-IoU bounding box loss function. YOLO-SE showcases remarkable performance, achieving an average precision at IoU threshold 0.5 (AP50) of 86.5% on the optical remote sensing dataset SIMD. This represents a noteworthy 2.1% improvement over YOLOv8 and YOLO-SE outperforms the state-of-the-art model by 0.91%. In further validation, experiments on the NWPU VHR-10 dataset demonstrated YOLO-SE's superiority with an accuracy of 94.9%, surpassing that of YOLOv8 by 2.6%. The proposed advancements position YOLO-SE as a compelling solution in the realm of deep learning-based remote sensing image object detection.

**Keywords:** object detection; remote sensing images; multi-scale; loss functions



**Citation:** Wu, T.; Dong, Y. YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition. *Appl. Sci.* **2023**, *13*, 12977. https://doi.org/10.3390/app132412977

Academic Editor: Andrea Prati

Received: 13 November 2023
Revised: 28 November 2023
Accepted: 30 November 2023
Published: 5 December 2023



## 1. Introduction

Object detection serves as a prerequisite for advanced visual tasks such as scene understanding. Compared to object detection in videos, detecting objects in static images is more challenging. The utilization of object detection in optical remote sensing images has been widespread, encompassing diverse applications such as monitoring, traffic surveillance, agricultural development, disaster planning, and geospatial referencing [1–4]. This has attracted considerable interest from researchers in recent years.

Traditional remote sensing image object-detection algorithms can be categorized into several types: threshold-based methods, feature engineering-based methods, template matching methods, machine learning-based methods, segmentation-based methods, and spectral information-based methods [5].

(1) Threshold-Based Methods: These methods typically use image brightness or color information to set appropriate thresholds for separating targets from the background. When pixel values exceed or fall below specific thresholds, they are considered as targets or non-targets. These methods are simple and user-friendly, but they are less stable under changing lighting conditions and complex backgrounds.

(2) Feature Engineering-Based Methods: These methods rely on manually designed features, such as texture, shape, and edges, to identify objects. Information is often extracted using filters, shape descriptors, and texture features. Subsequently, classifiers like support vector machines or decision trees are used to categorize the extracted features.

(3) Template Matching: Template matching is a method for identifying objects by comparing them with predefined templates or patterns. When the similarity between the target and the template exceeds a certain threshold, it is detected as an object. This method works well when there is a high similarity between the target and the template, but it is not very robust for target rotation, scaling, and other variations.

(4) Machine Learning-Based Methods: Machine learning algorithms, such as neural networks, support vector machines, and random forests, are employed to learn how to detect objects from data. Feature extraction and classifier parameters are often automatically learned from training data. This approach tends to perform well in complex object detection tasks, but it requires a significant amount of labeled data and computational resources.

(5) Segmentation-Based Methods: Object detection methods based on segmentation first divide the image into target and non-target regions and then perform further analysis and classification on each region. Segmentation methods can include region growing, watershed transform, and graph cuts. This approach works well when there are clear boundaries between objects and the background.

(6) Spectral Information-Based Methods: Remote sensing images typically contain information from multiple spectral bands, and this information can be leveraged for object detection. Methods based on spectral information often use spectral features like spectral reflectance and spectral angles to distinguish different types of objects.

The swift advancement of deep learning technologies, particularly the introduction of convolutional neural networks (CNNs), has ushered in new possibilities and applications for object detection in remote sensing images [6,7].

At present, the predominant frameworks for object detection in remote sensing images can be broadly classified into two major categories: single-stage and two-stage methods. Notable two-stage object detection algorithms encompass Spatial Pyramid Pooling Networks (SPP-Net) [8], Region-based CNN (R-CNN) [9], and Faster R-CNN [10]. Two-stage methods often achieve high detection accuracy but tend to be slower in terms of detection speed with larger model sizes and parameter counts, due to their two-stage nature. Single-stage detection algorithms have effectively addressed these issues, with representative algorithms such as YOLO [11], SSD [12], RetinaNet [13], CornerNet [14], and RefineDet [15]. As these single-stage algorithms—for example, YOLO—have matured, they not only outperform two-stage methods in terms of detection speed but also match or surpass them in terms of accuracy.

Despite the current state of the art, there is still room for improvement in the detection of objects at multiple scales and small targets. Several object categories in remote sensing display size variations, even within the same category. For instance, ships in ports can range in size from tens of meters to hundreds of meters. Additionally, the height of capture and distance from the target can affect an object's size in the image. Moreover, many small objects are present in aerial images, which are often filtered out in the pooling layers of convolutional neural networks (CNNs) due to their small size, making them challenging to detect and recognize.

To tackle these challenges, this paper suggests an improved network model built upon the foundation of YOLOv8. Our approach improves conventional convolution techniques, incorporates state-of-the-art attention mechanisms, and enhances the loss functions. The primary contributions of this paper are as follows:

(1) A lightweight convolution SEConv was introduced to replace standard convolutions, reducing the network's parameter count and speeding up the detection process. To address multi-scale object detection, the SEF module was proposed, based on SEConv.

(2) A novel EMA attention mechanism was introduced and integrated into the network, resulting in the SPPFE module, which enhances the network's feature extraction capabilities and effectively addresses multi-scale object detection challenges.

(3) To improve the detection of small objects, an additional prediction head designed for tiny-object detection was added. Furthermore, the original detection head was replaced by a transformer prediction head to capture more global and contextual information.

(4) To mitigate the adverse gradients generated by low-quality examples, the Wise-IoU loss function was introduced.

(5) In the evaluation on the SIMD dataset, the AP50 reached 86.5%, marking a 2.1% improvement over YOLOv8, outperforming the state-of-the-art model YOLO-HR by 0.91%. Furthermore, we conducted validation on the NWPU VHR-10 dataset, where YOLO-SE achieved 94.9% accuracy, outperforming YOLOv8 by 2.6%.

The paper unfolds as follows: Section 2 delves into a comprehensive review of existing work concerning object-detection networks in remote sensing images, with a particular focus on attention mechanisms. Section 3 provides an intricate overview of both the YOLOv8 network and our proposed YOLO-SE network. In Section 4, we present a detailed account of our experiments and conduct a thorough analysis of the results, using both the SIMD dataset and the NWPU VHR-10 dataset. Finally, Section 5 encapsulates our conclusions.

## 2. Related Work

### *2.1. Object-Detection Networks for Remote Sensing Images*

While deep learning has demonstrated remarkable success in object detection for remote sensing images, effectively detecting multi-scale and small objects continues to pose a substantial challenge. Researchers have made notable contributions to address these challenges. Ma et al. [16] improved upon Faster R-CNN by proposing a method for identifying medium- and small-sized animals in large-scale images. They utilized the HRNet feature extraction network to enhance small-object detection. Sun et al. [17] presented the partial-based convolutional neural network (PBNet) to address compound object detection in high-resolution optical remote sensing images. Lai et al. [18] devised a feature extraction module that integrates convolutional neural networks (CNNs) and multi-head attention, leading to an expanded receptive field and the development of the STC-YOLO algorithm. Additionally, they introduced the Normalized Gaussian Wasserstein Distance (NWD) metric to enhance sensitivity to small-object position deviations. Han et al. [19] proposed the Ghost module for building efficient neural network architectures. GhostNet, constructed using this new module, achieved a balance between efficiency and accuracy. Lin W et al. [20] introduced a Scale-Aware Aggregation Module (SMT) that effectively simulates the transition from local to global dependencies with network depth, offering better performance with fewer parameters. Wan et al. [21] presented the YOLO-HR algorithm for high-resolution optical remote sensing object recognition. This algorithm employs multiple detection heads for object detection and reuses output features from the feature pyramid, further enhancing detection performance. Xu et al. [22] proposed a multi-scale remote sensing object detection model based on YOLOv3. They improved the existing feature extraction network by introducing DenseNet. This method exhibited good performance in multi-scale remote sensing object detection. Cao et al. [23] proposed a GhostConv-based backbone lightweight YOLO network (GCL_YOLO). The network first establishes a lightweight backbone network based on ghost convolutions with a minimal number of parameters. Subsequently, a novel small-object prediction head is designed to replace the existing large-object prediction head used for natural scene objects. Finally, the network utilizes the focus-effective intersection over union (Focus-EIOU) loss as the localization loss.

The above-mentioned research endeavors have led to various improvements in object-detection algorithms for remote sensing images, contributing to the advancement of this field. In light of the challenges posed by multi-scale and small-object detection, this paper aims to further enhance the state-of-the-art YOLOv8 algorithm.

### 2.2. Attention Mechanism

The attention mechanism empowers neural networks to concentrate on crucial features while disregarding less pertinent ones [24]. Convolution operations combine channel information and spatial information to extract features, making attention-mechanism designs consider both channel and spatial aspects. Currently, there are three main types of attention mechanisms:

(1) Channel Attention: This type of attention mechanism prioritizes important features or channels within the data. For example, SENet [25] focuses on crucial channels for better feature representation.

(2) Spatial Attention: Spatial attention directs the model's focus to essential spatial positions within the data. Self-attention mechanisms, such as those used in deformable convolutional networks (DCNs) [26], excel at capturing spatial relationships.

(3) Mixed Attention Mechanisms: Some attention mechanisms, like the Convolutional Block Attention Module (CBAM) [27], combine both channel and spatial attention characteristics. CBAM can simultaneously attend to channels and spatial positions, contributing to improved model accuracy and noise suppression by considering both aspects of the data.

Additionally, we acknowledge the significance of other works. Many works also use neural attention to improve feature learning, such as Motion-attentive Transition for Zero-shot Video Object Segmentation [28] and Regional Semantic Contrast and Aggregation for Weakly Supervised Semantic Segmentation [29]. These works further demonstrate the diverse applications of attention mechanisms for feature learning.

These attention mechanisms play a crucial role in enhancing model performance by selectively emphasizing relevant information, which is particularly beneficial in tasks like object detection and image analysis.

### 2.3. Vision Transformer

The transformer [30] was initially devised for machine translation tasks within the realm of natural language processing (NLP). Owing to its potent representation capabilities, researchers have been investigating avenues to leverage transformers for computer vision tasks. Models based on transformers have demonstrated performance on diverse visual benchmarks comparable to or surpassing other network types such as convolutional and recurrent neural networks. The growing attention within the computer vision community toward transformers is attributed to their superior performance and reduced need for domain-specific feature engineering [31].

Chen et al. [32] conducted training on a sequence transformer for pixel-level regression, achieving results akin to CNNs in image-classification tasks. Another notable model, ViT, directly applies pure transformers to image patch sequences, enabling the classification of entire images. Dosovitskiy et al. [33] recently introduced a model that attained state-of-the-art performance across various image recognition benchmarks.

While traditional visual transformers excel in capturing long-range dependencies between patches, they often neglect local feature extraction and project 2D patches onto vectors, using simple linear layers. In response to this, recent research has concentrated on enhancing modeling capacity for local information. TNT [34], for instance, divides patches into multiple sub-patches and introduces a novel transformer-in-transformer architecture. This leverages inner transformer blocks to model relationships between sub-patches and outer transformer blocks for patch-level information exchange. Twins [35] and CAT [36] alternatively perform local and global attention at different layers. Swin transformers [37] execute local attention within a window and introduce a shift-window partition method for cross-window connections. RegionViT [38] generates region tokens

and local tokens from images, with local tokens receiving global information through attention to region tokens. Beyond local attention, some research suggests enhancing local information through local feature aggregation, such as T2T. Improved computations for self-attention layers have also garnered attention. DeepViT [39], for instance, proposes building cross-head communication to regenerate attention maps, fostering increased diversity at different levels.

Zhu et al. [40] integrated a transformer prediction head into the YOLOv5 structure, proposing the TPH-YOLOv5 model. This model introduces an additional prediction head to detect objects at different scales, utilizing a transformer prediction head (TPH) to leverage the predictive potential of self-attention mechanisms. Building upon this, Zhao et al. [41] enhanced the model with the TPH-YOLOv5++ version, significantly reducing computational costs and improving detection speed. In TPH-YOLOv5++, they introduced a cross-layer asymmetric transformer (CA-Trans) to replace the extra prediction head while maintaining its functionality. The use of the Sparse Local Attention (SLA) module effectively captures asymmetric information between additional heads and other heads, enriching the features of other heads.

## 3. Materials and Method

### *3.1. YOLOv8*

YOLOv8 utilizes a similar backbone to YOLOv5, but with some modifications in the CSPLayer, now referred to as the C2f module. The C2f module, which consists of a two-convolution cross-stage partial bottleneck, combines high-level features with contextual information to enhance detection accuracy. YOLOv8 employs an anchor-free model with decoupled heads to independently handle object-detection, classification, and regression tasks. This design allows each branch to focus on its specific task, leading to improved overall model accuracy. In the output layer of YOLOv8, they use the sigmoid function as the activation function for object scores, indicating the probability of an object being present within the bounding box. They use the softmax function to represent class probabilities, signifying the probability of an object belonging to each possible class.

For bounding box loss, YOLOv8 uses the CIoU [42] and DFL [43] loss functions, and for classification loss, it employs binary cross-entropy. These loss functions enhance object-detection performance, especially when dealing with smaller objects. For this paper, we selected YOLOv8 as the baseline model, which consists of three key components: the backbone network, the neck network, and the prediction output head. The backbone network is the core part of the YOLOv8 model and is responsible for extracting features from the input RGB color images. The neck network is positioned between the backbone network and the prediction output head. Its primary role is to aggregate and process the features extracted by the backbone network. In YOLOv8, the neck network plays a crucial role in integrating features of different scales. Typically, the neck network adopts a Feature Pyramid Network (FPN) structure, which effectively fuses features of various scales to construct a more comprehensive representation.

The prediction output head is the topmost part of the YOLOv8 model and is responsible for identifying and locating object categories in the images. The output head usually contains multiple detectors, with each detector responsible for predicting the position and category of objects. In YOLOv8, three sets of detectors are employed, each with a different scale, aiding the model in recognizing objects of various sizes. The network architecture of YOLOv8 is illustrated in Figure 1.

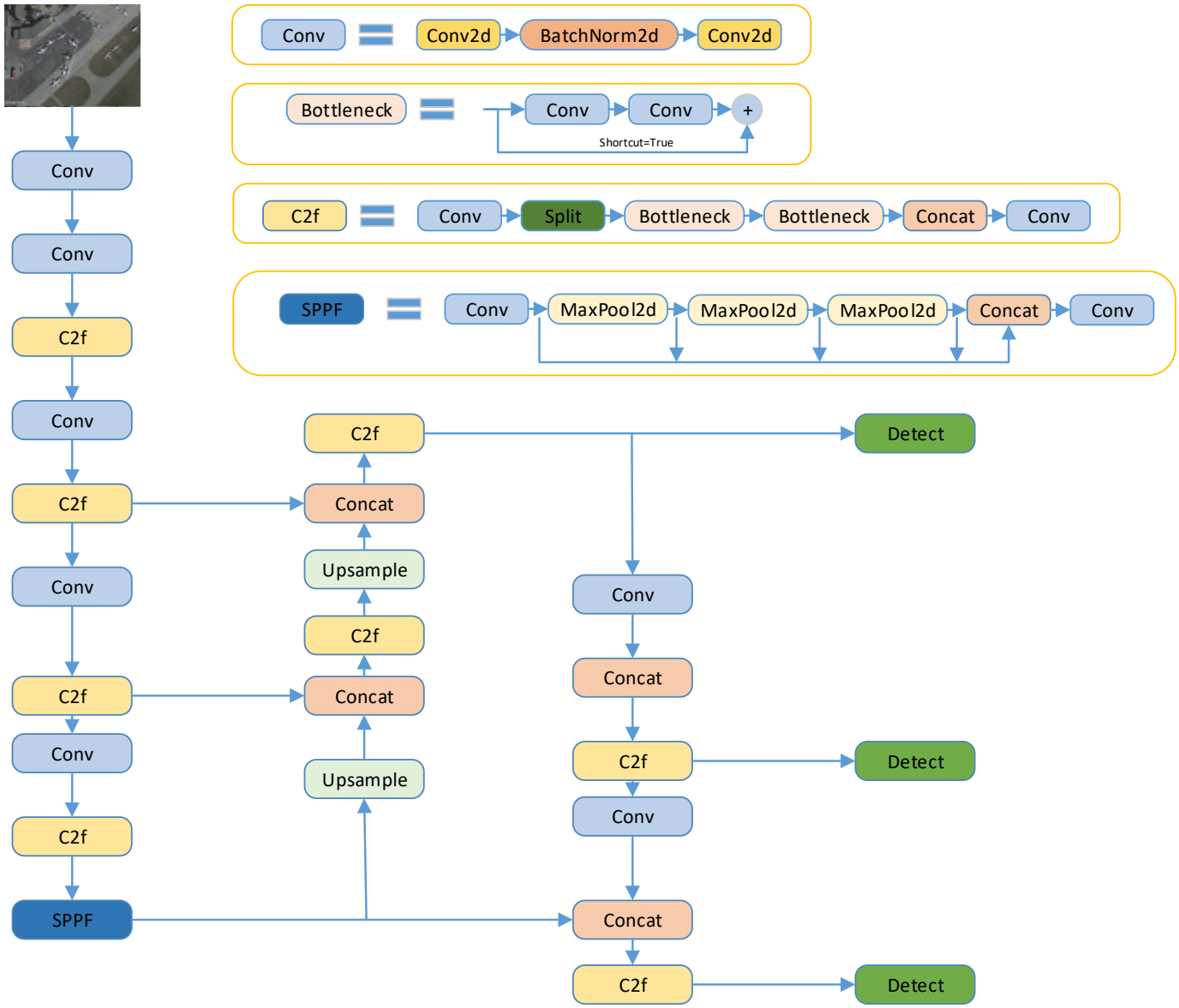



**Figure 1.** The architecture of YOLOv8.

### *3.2. YOLO-SE*

To address the issues related to detecting small objects and objects at multiple scales with the YOLOv8 network, we propose the YOLO-SE algorithm, as discussed in this section. We first provide an overview of the YOLO-SE architecture. Building on this, we introduce the essential components of YOLO-SE, including the Efficient Multiscale Convolution Module (SEF), improvements to convolution through the introduction of the EMA attention mechanism in the SPPFE module, replacing the original YOLOv8 detection head with a transformer prediction head, and adding an additional detection head to handle objects at different scales. Finally, we replace the original CIOU bounding box loss function with Wise-IoU. The network structure of YOLO-SE is depicted in Figure 2.

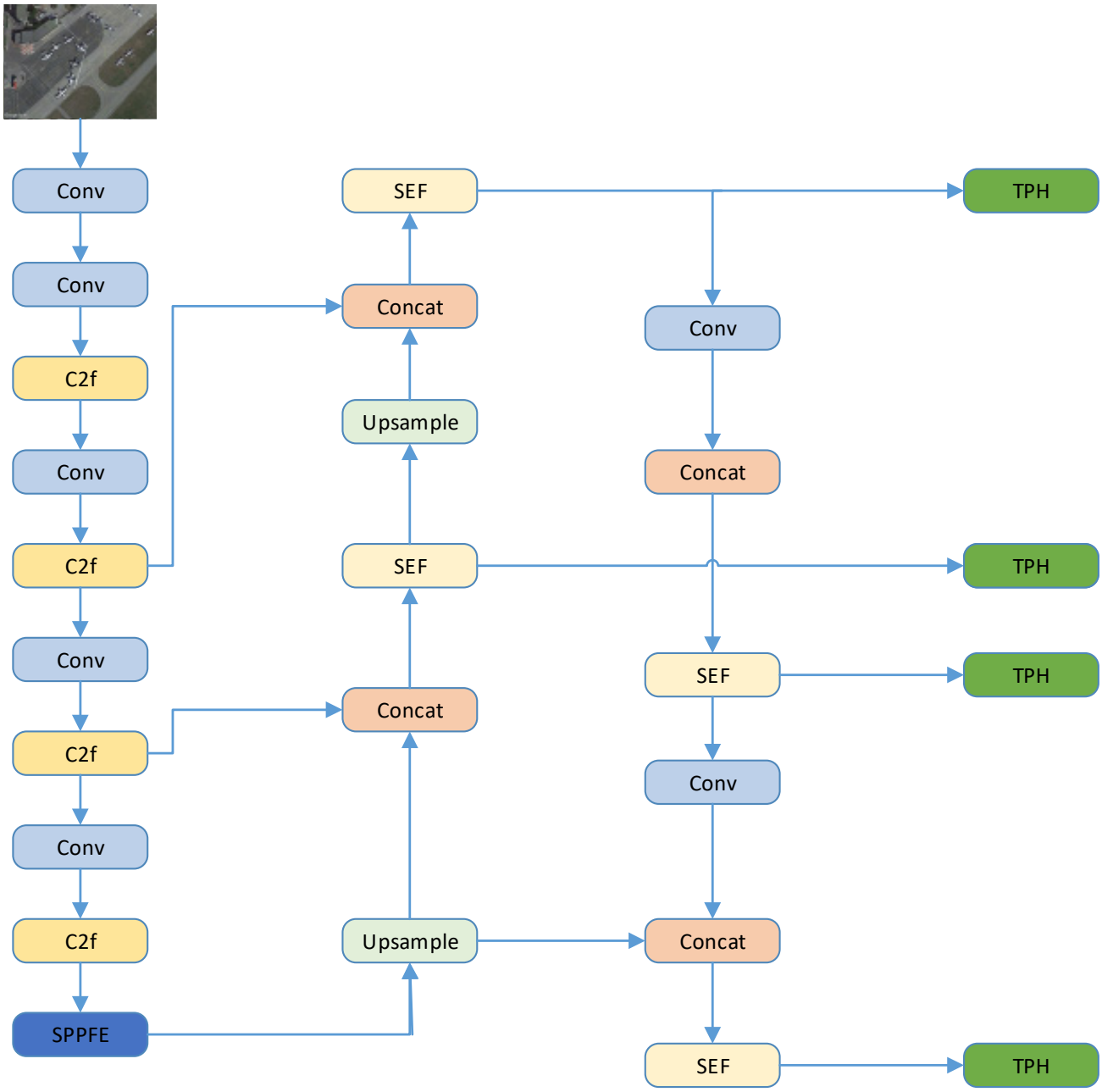



**Figure 2.** The architecture of proposed YOLOv8.

### *3.3. SEF*

We replaced the standard convolutions in C2f with a more lightweight and efficient multi-scale convolution module called SEF. This module introduces multiple convolutions with different kernel sizes, enabling it to capture various spatial features at multiple scales. Additionally, SEF extends the receptive field using larger convolution kernels, enhancing its ability to model long-range dependencies.

As shown in Figure 3, the SEConv2d operation partitions the input channels into four smaller channels. The first and third smaller channels remain unchanged, while the second and fourth channels undergo $3 \times 3$ and $5 \times 5$ convolution operations, respectively. Subsequently, a $1 \times 1$ convolution consolidates the features from these four smaller channels. By employing half of the features for convolution and then integrating them with the original features, the objective is to generate redundant features, decrease parameters and computational workload, and alleviate the influence of high-frequency noise. This

approach aims to reduce the number of parameters and computational expenses while preserving essential feature information. Each distinct convolutional mapping learns to focus on features of varying granularities adaptively. SEF excels in capturing local details, preserving the nuances and semantic information of target objects as the network deepens.

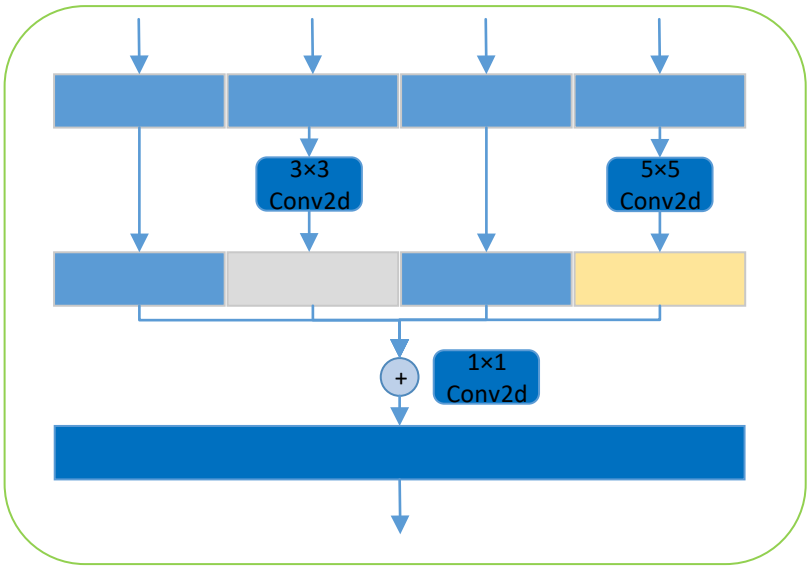



**Figure 3.** The architecture of SEConv2d.

The structure of SEF is shown in Figure 4. In summary, the SEF module reduces the network's parameter count, accelerates detection speed, and effectively captures multi-scale features and local details of the target.

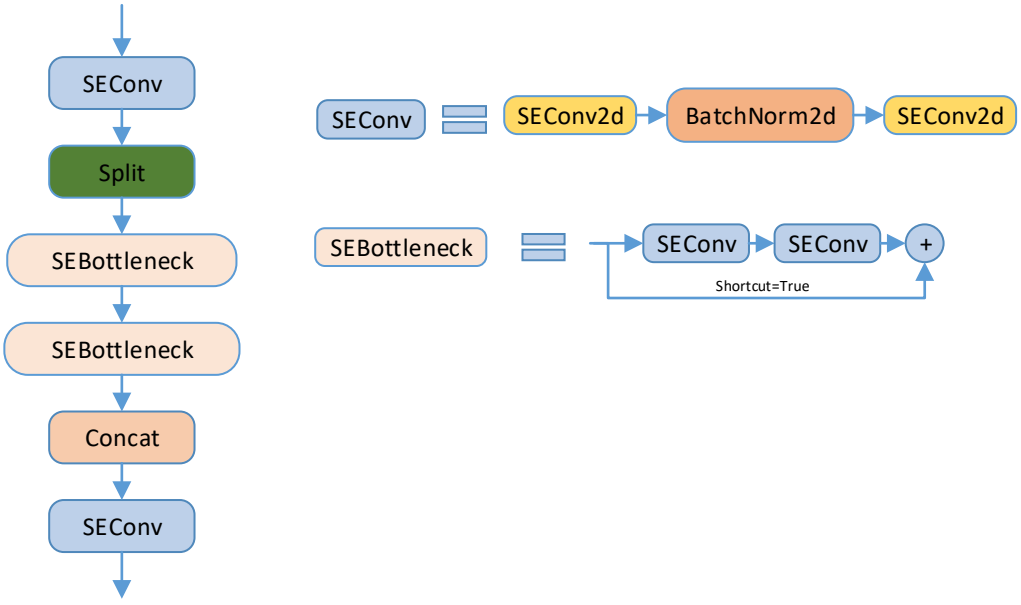



**Figure 4.** The architecture of SEF.

### *3.4. SPPFE*

The SPPF module in YOLOv8 has demonstrated its advantages in enhancing model performance through multi-scale feature fusion, particularly in certain scenarios. However, we must acknowledge that the SPPF module may have limitations in complex backgrounds and situations involving variations in target scales. This is because it still lacks a fine-grained mechanism to focus on task-critical regions.

To address the limitations of the SPPF module and enhance feature extraction capabilities, we introduce the Efficient Multi-Scale Attention (EMA) mechanism [44], which dynamically adjusts the weights in the feature maps based on the importance of each region in an adaptive manner. The EMA attention mechanism is employed to retain information on each channel while reducing computational costs. We achieve this by restructuring a portion of the channels into the batch dimension and grouping the channel dimensions into multiple sub-features, ensuring an even distribution of spatial semantic features within each feature group. This approach helps maintain channel-wise information while minimizing computational expenses. This allows the module to focus on task-critical regions, making it more targeted in complex scenes. The structure of SPPFE is depicted in Figure 5, and we incorporate the EMA attention mechanism into this module. The SPPEF module not only performs multi-scale feature fusion but also finely adjusts features at each scale, effectively capturing information at different scales. This enhancement significantly improves the model's ability to detect small objects.

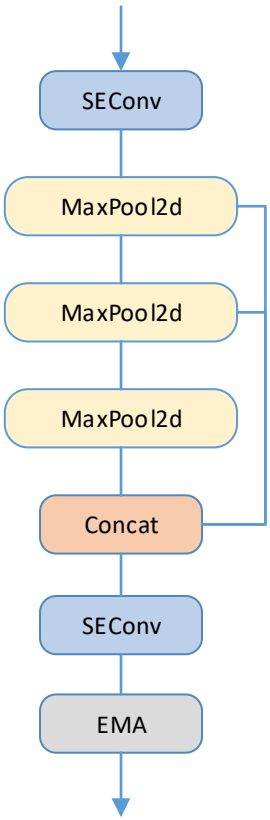



**Figure 5.** The architecture of SPPFE.

### *3.5. TPH*

Due to the significant variation in object sizes within remote sensing images, including numerous extremely small instances, experimental results have shown that YOLOv8's original three detection heads do not adequately address the challenges presented by remote sensing imagery. As a result, we added an additional prediction head specifically designed for detecting tiny objects. When combined with the other three prediction heads, this approach enables us to capture relevant information about small targets more effectively while also detecting objects at different scales, thus improving overall detection performance.

We replaced the original detection head with a transformer prediction head to capture more global and contextual information. The structure of the Vision Transformer is depicted in Figure 6. It consists of two main blocks: a multi-head attention block and a feedforward neural network (MLP). The LayerNorm layer aids in better network convergence and prevents overfitting. Multi-head attention allows the current node to focus not only on

the current pixel but also on the semantic context. While the additional prediction head introduces a considerable amount of computational and memory overhead, it has improved the performance of tiny-object detection.

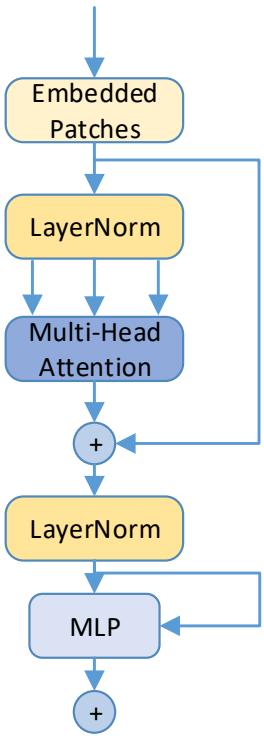



**Figure 6.** The architecture of the Transformer Encoder module.

### *3.6. Wise-IoU Loss*

YOLOv8 uses Complete Intersection over Union (CIoU) [42] as the default loss-calculation method. CIoU builds upon Distance Intersection over Union (DIoU) by introducing the aspect ratio of the predicted bounding box and the ground-truth bounding box, making the loss function more attentive to the shape of the bounding boxes. However, the computation of CIoU loss is relatively complex, leading to higher computational overhead during the training process. Weighted Intersection over Union (WIoU) [45] proposes a dynamic non-monotonic focus mechanism, replacing IoU with dissimilarity to assess the quality of anchor boxes. It adopts a gradient gain allocation strategy, reducing the competitiveness of high-quality anchor boxes and mitigating harmful gradients caused by low-quality anchor boxes. This allows WIoU to focus on low-quality anchor boxes, ultimately improving the overall performance of the detector. WIoU comes in three versions, namely $\mathrm{WIoU_{v1}}$, which constructs an attention-based bounding box loss, and $\mathrm{WIoU_{v2}}$ and $\mathrm{WIoU_{v3}}$, which build upon $v1$ by adding gradient gain to the focus mechanism.

The formula for calculating $WIoU_{v1}$ is as shown in Equation (2):

$$L_{IoU} = 1 - \frac{Bbox \cap Tbox}{Bbox \cup Tbox} \tag{1}$$

$$L_{WIoU_{v1}} = L_{IoU} R_{WIoU} \tag{2}$$

The calculation formula for Region-based Weighted Intersection over Union (R-WIoU) is as follows, as shown in Equation (3):

$$R_{WIoU} = \exp\left(\frac{\left((x_{Bbox} - x_{Tbox})^2 + (y_{Bbox} - y_{Tbox})^2\right)}{(w^2 + h^2)^*}\right) \tag{3}$$

The values of $w$, $h$, $(x_{Bbox}, y_{Bbox})$, and $(x_{Tbox}, y_{Tbox})$ are illustrated in Figure 7.

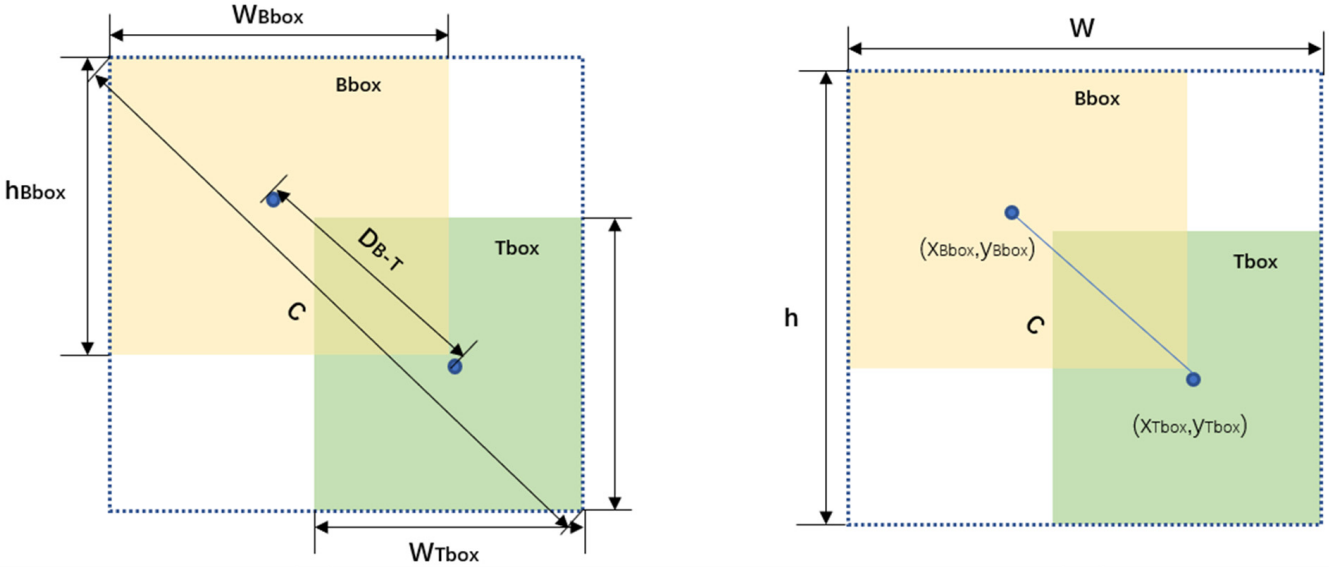



**Figure 7.** The values of $w$, $h$: (**a**) CIoU; (**b**) WIoU.

To prevent $R_{WIoU}$ from producing gradients that hinder convergence, w and h are separated from the computation graph. $R_{WIoU}$ takes values in the range [1, e), significantly amplifying the importance of low-quality anchors. Loss Intersection over Union (LIoU), on the other hand, takes values in the range [0, 1], significantly reducing $R_{WIoU}$ for high-quality anchors and focusing on the distance between the center points when Bbox and Tbox overlap.

The dynamic non-monotonic focus mechanism uses "outlyingness" to assess anchor box quality instead of IoU, and it provides a wise gradient gain allocation strategy. This strategy reduces the competitiveness of high-quality anchor boxes while also mitigating harmful gradients generated by low-quality examples. This allows WIoU to focus on ordinary-quality anchor boxes and improve the overall performance of the detector.

## 4. Experimental and Results Analysis

### *4.1. Experimental Setup and Evaluation Metrics*

#### 4.1.1. Experimental Environment

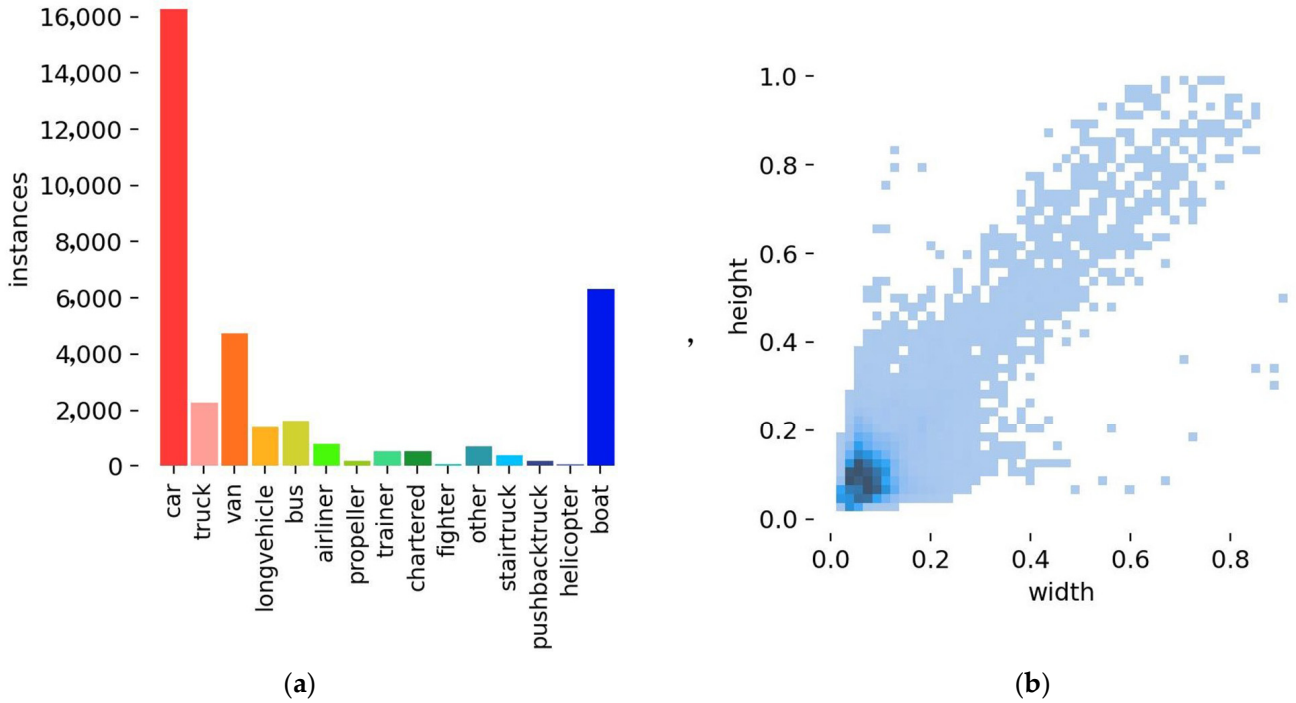
The experimental platform used in this study is shown in Table 1.

**Table 1.** Experimental platform.

| Name | Version |
|---|---|
| CPU | Intel(R) Xeon(R) CPU E5-2696 v4 @ 2.20 GHz |
| GPU | NVIDIA GeForce RTX 3090, 24 GB |
| Memory | 64 GB |
| Operating system | Ubuntu 22.04 |
| Deep learning framework | Pytorch 1.13 |

The SIMD dataset is a multi-class, open-source, high-resolution, and fine-grained remote sensing object-detection dataset. It consists of 5000 images with a total of 45,096 objects distributed across 15 different classes such as cars, airplanes, and helicopters. The distribution of classes and the sizes of objects in the training set are shown in Figure 8. SIMD is characterized by class imbalance and the presence of small objects. As Figure 8a illustrates, the "car" class has more than 16,000 objects, while some classes like "helicopter"

and "fighter" have fewer than 500 objects. We partitioned the images into 80% for training and 20% for testing.




(**a**) (**b**)

**Figure 8.** The distribution of targets in the SIMD dataset: (**a**) the distribution of the number of categories; (**b**) the distribution of target width and height; the color gradient from white to blue (from light to dark) signifies a more concentrated distribution.

In this experiment, YOLOv8s' pretrained weights were used, and the training was conducted for 200 epochs with a batch size of 16 and an input image size of 1024 × 1024 pixels.

#### 4.1.2. Evaluation Metrics

In target-detection tasks, metrics such as recall (R), precision (P), and average precision (AP) are commonly used for evaluation. Recall represents the proportion of correctly detected targets out of the total number of targets, calculated using the formula shown in Equation (4).

Precision represents the proportion of correctly detected targets out of the total predicted targets, as shown in Equation (5).

$$R = \frac{TP}{TP + FN} \times 100\% \tag{4}$$

$$P = \frac{TP}{TP + FP} \times 100\% \tag{5}$$

where $TP$ represents the count of correctly identified targets, $FP$ is the count of erroneously detected targets, and $FN$ is the count of targets that remain undetected. The Average Precision (AP) is the measure of the area under the Precision-Recall ($P$-$R$) curve, where recall is plotted on the x-axis and precision is plotted on the y-axis. The calculation formula for Average Precision (AP) is given by Equation (6).

$$\mathrm{AP} = \int_0^1 \mathrm{P_i}(\mathrm{R_i})\mathrm{dR_i} \tag{6}$$

To obtain the mean Average Precision (mAP) for multiple classes, the AP values for each class are averaged. The formula for calculating mAP is as shown in Equation (7).

$$mAP = \frac{\sum_{i=1}^{C} AP_i}{C} \times 100\% \quad (7)$$

In this paper, mAP@0.5 is used as the primary evaluation metric for precision. For convenience, throughout the rest of this paper, mAP@0.5 will be referred to as AP0.5.

### 4.2. Results Analysis

#### 4.2.1. Experimental Results

The AP50 for various categories in the SIMD dataset reached 86.5%. Table 2 presents the detection results for all categories using the YOLO-SE algorithm on the SIMD dataset. To further validate the effectiveness of the algorithm, we conducted experiments on the NWPU VHR-10 dataset. NWPU VHR-10 is a publicly available geospatial object detection dataset comprising a total of 800 remote sensing (RS) images. We split the dataset into 80% for training and 20% for testing. Table 3 presents the detection results on the NWPU VHR-10 dataset; YOLO-SE achieved an accuracy of 94.9%.

**Table 2.** Detection results on SIMD dataset.

| Categories | P/% | R/% | AP50/% | mAP/% |
|---|---|---|---|---|
| car | 85.7 | 91.3 | 94.1 | 75.0 |
| truck | 80.4 | 79.6 | 86.0 | 71.0 |
| van | 79.7 | 80.8 | 85.8 | 69.6 |
| long vehicle | 80.4 | 82.1 | 84.8 | 68.3 |
| bus | 87.0 | 88.3 | 93.8 | 79.6 |
| airliner | 93.3 | 96.6 | 98.9 | 91.3 |
| propeller | 92.0 | 93.9 | 95.1 | 82.9 |
| trainer | 91.3 | 99.2 | 98.2 | 85.4 |
| chartered | 87.1 | 97.2 | 95.2 | 85.7 |
| fighter | 79.3 | 100.0 | 97.1 | 87.0 |
| other | 62.4 | 42.7 | 44.7 | 34.5 |
| stair truck | 68.9 | 57.8 | 66.5 | 46.1 |
| pushback truck | 88.2 | 50.8 | 68.0 | 49.0 |
| helicopter | 82.0 | 70.4 | 90.8 | 53.3 |
| boat | 96.2 | 98.1 | 98.5 | 81.1 |
| all | 83.6 | 81.9 | 86.5 | 70.7 |

**Table 3.** Detection results on NWPU VHR-10 dataset.

| Class | P/% | R/% | AP50/% | mAP/% |
|---|---|---|---|---|
| airplane | 87.8 | 94 | 97.2 | 88.7 |
| ship | 86.2 | 100 | 99.5 | 93.2 |
| storage tank | 99.2 | 100 | 99.5 | 84.8 |
| baseball diamond | 98.1 | 95.9 | 97.9 | 82.6 |
| tennis court | 94.5 | 93.3 | 97.6 | 81.5 |
| basketball court | 100 | 83.2 | 95.3 | 58.1 |
| ground track field | 97.2 | 76.9 | 92.5 | 60.1 |
| harbor | 88.5 | 86.7 | 93.6 | 69.5 |
| bridge | 90.3 | 100 | 98 | 86.3 |
| vehicle | 64.6 | 80 | 77.5 | 67.3 |
| all | 90.6 | 91 | 94.9 | 77.2 |

Figure 9 displays some visual results of the proposed method on the SIMD dataset. It is evident from Figure 6 that the algorithm introduced in this paper effectively addresses the challenges of multi-scale objects and noise in complex environments. Additionally, it demonstrates good performance in detecting small objects.

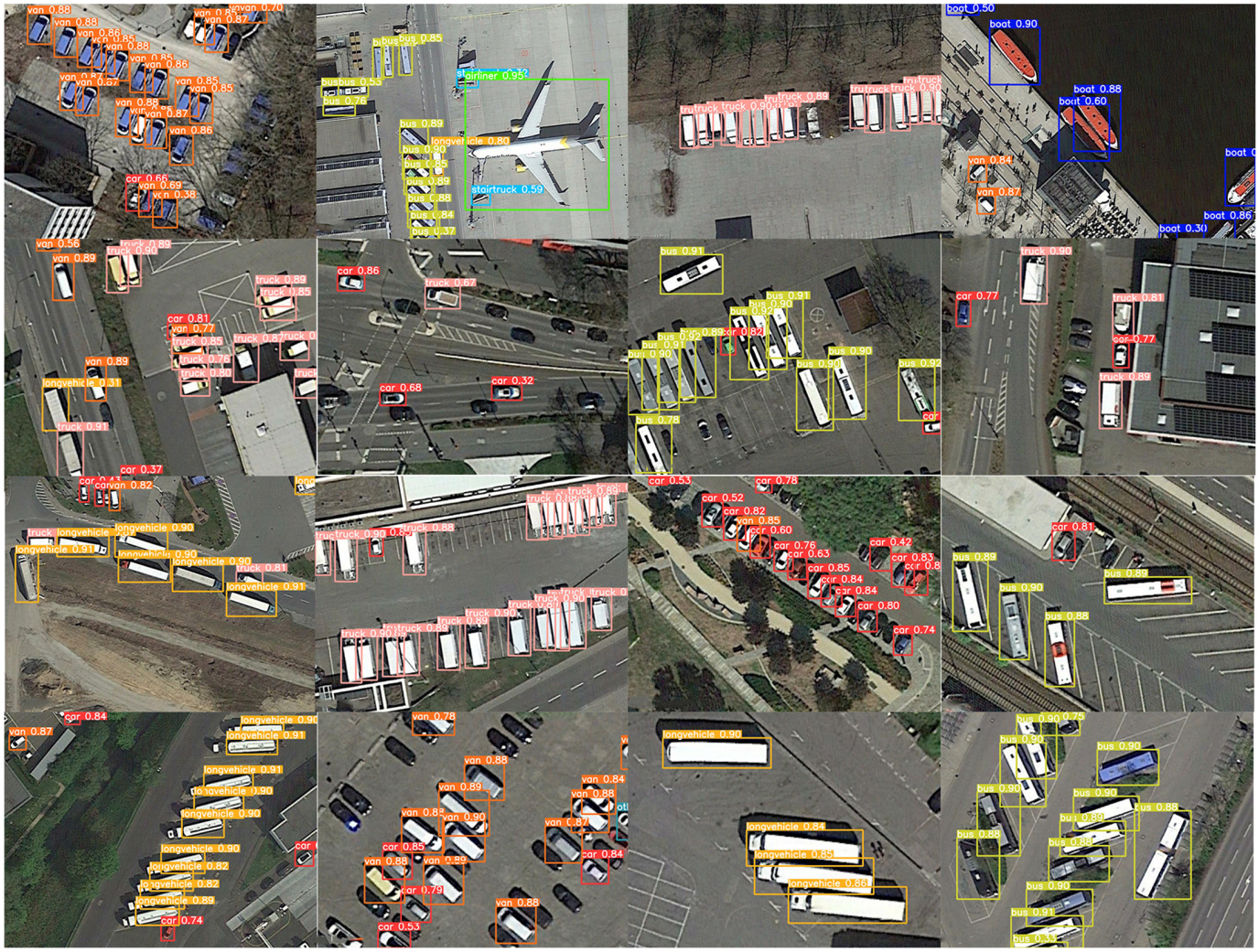


**Figure 9.** Visual results on the SIMD dataset.

Figure 10 displays Precision-Recall (PR) curves for each class in the SIMD dataset, illustrating the average precision for each class. From the graph, we can observe that categories like 'airliner,' 'propeller,' 'trainer,' 'chartered,' 'fighter,' and 'boat' perform remarkably well, with average precisions all exceeding 0.95. In contrast, the 'other' category has an average precision of only 0.447. This could be attributed to the 'other' category being diverse, with no unified features for the network to learn.

Figure 11 shows the progression of various metrics during training and validation, including box loss, object loss, class loss, and metrics after each epoch, such as accuracy and recall.

Figure 12 presents the confusion matrix for our model. It is evident that 'other,' 'stair truck,' and 'pushback truck' are sometimes not detected and are considered as background. Moreover, 'stair truck' and 'pushback truck' have a similar appearance, making it challenging to distinguish them in remote sensing images.

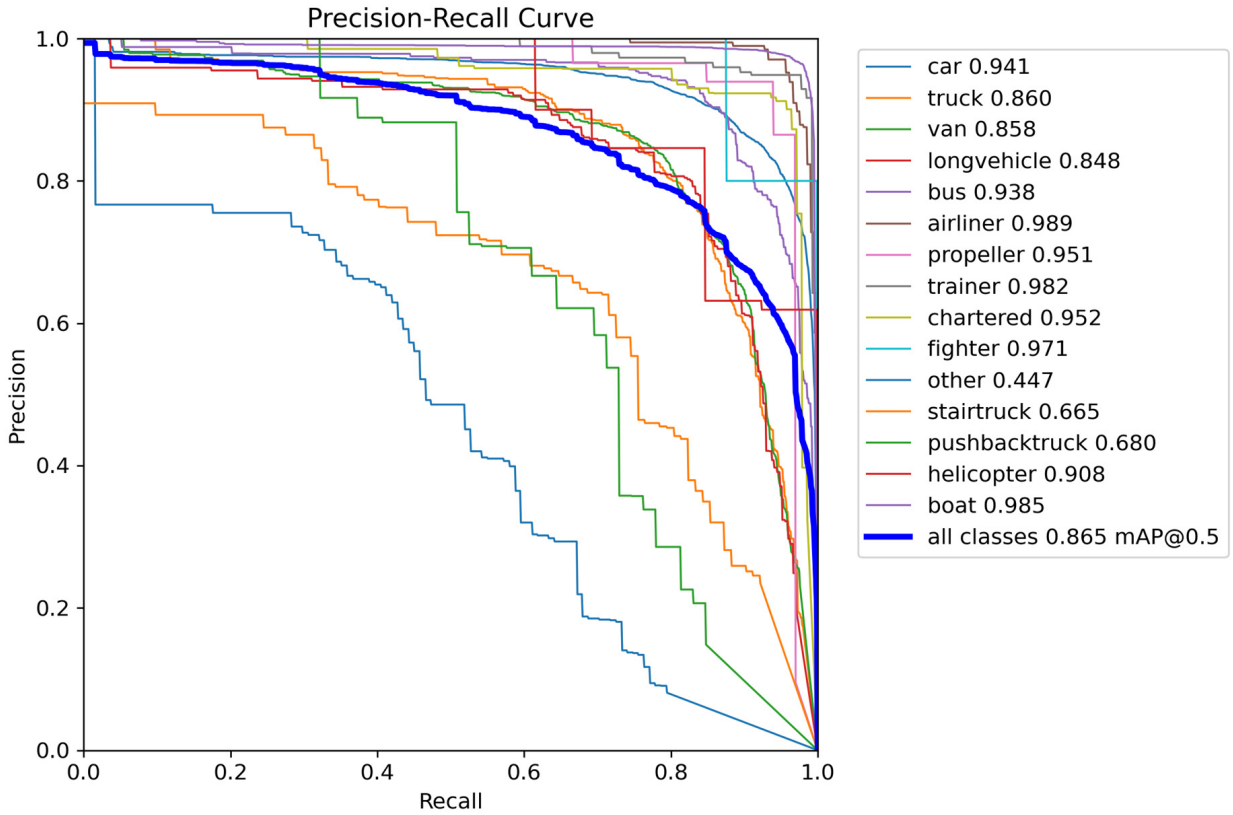



**Figure 10.** The P-R curve during training on SIMD dataset.

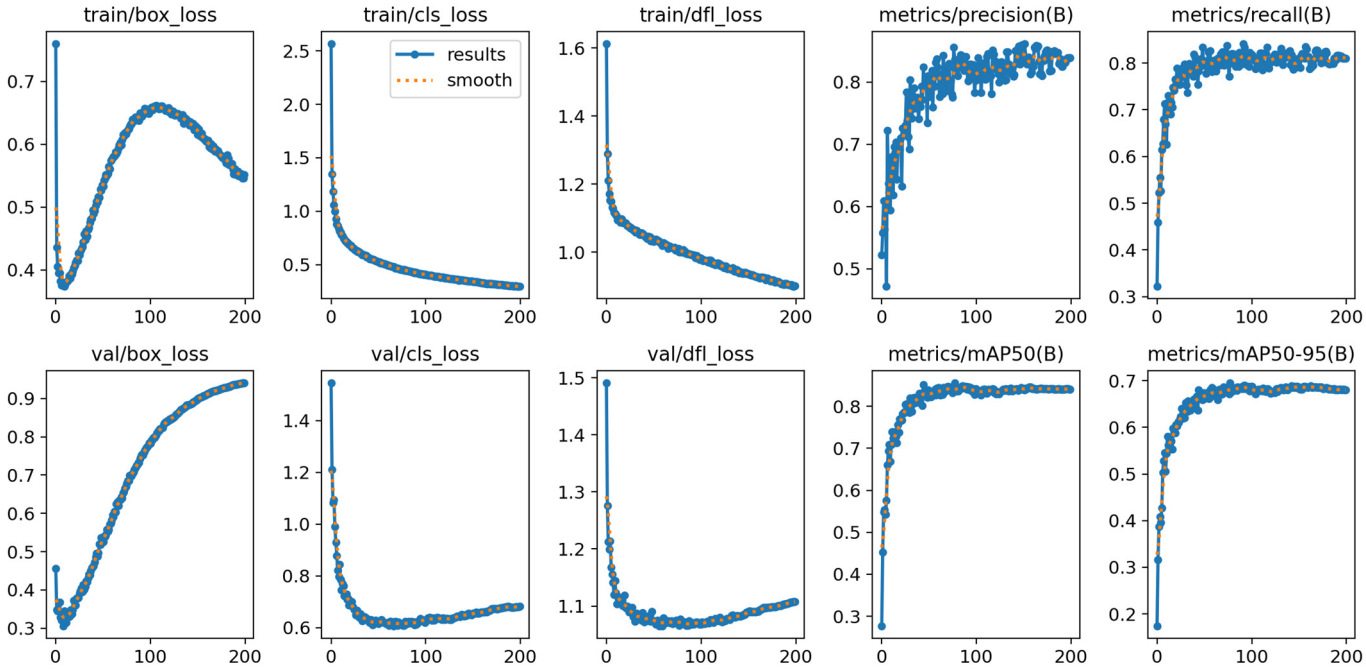



**Figure 11.** Network convergence on the SIMD dataset.

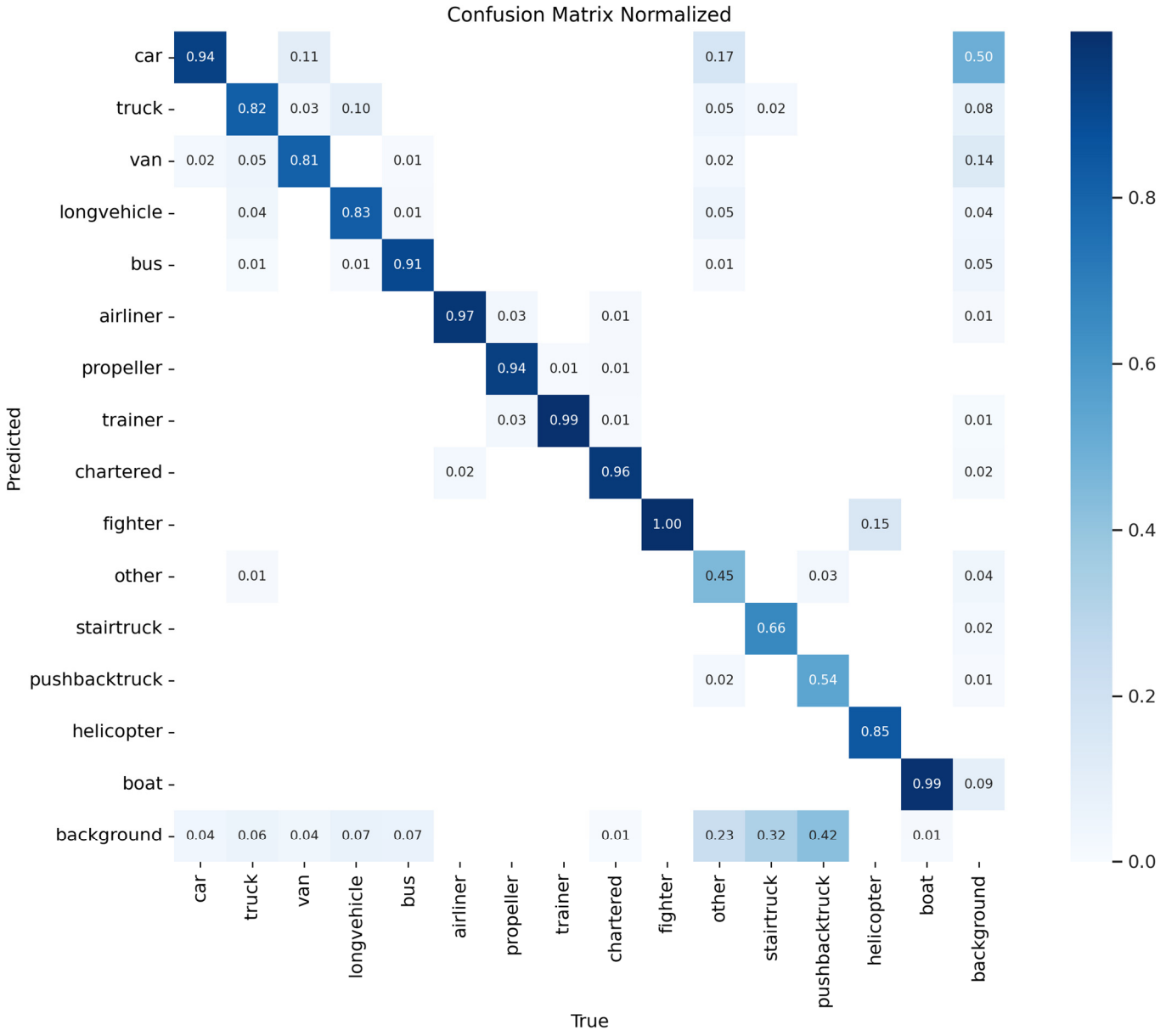



**Figure 12.** Confusion matrix on SIMD dataset.

#### 4.2.2. Comparison Experiments

To validate the performance of YOLO-SE, this paper conducted comparative experiments with a variety of classical algorithm models, including YOLOv3-Tiny [27], Faster RCNN [10], YOLOv5, YOLOv7 [46], YOLOX [47], YOLOv8, YOLO-DA [48], and YOLO-HR [21].

The experimental results are presented in Table 4. The Fast R-CNN model has the highest parameter count, yet its AP50 is relatively low compared to that of other models. Subsequent YOLO models have not only increased AP50 but also reduced the model's parameter count, highlighting the excellence of YOLO models. YOLO-SE consistently outperforms the other algorithm models in terms of detection accuracy; the AP50 reached 86.5%, outperforming the state-of-the-art model YOLO-HR by 0.91%. Additionally, YOLO-SE exhibits a lower parameter count, further demonstrating the superiority of the algorithm.

**Table 4.** Comparison with other algorithms.

| Name | AP50 (%) | Map (%) | Params (M) |
|---|---|---|---|
| Faster RCNN | 77.7 | - | 41.2 |
| YOLOv3-Tiny | 77.2 | 54.5 | 8.68 |
| YOLOv5s | 83.85 | 66.0 | 6.72 |
| YOLOv7s | 83.80 | 66.5 | 8.92 |
| YOLOX-s | 76.6 | 56.8 | 8.94 |
| YOLO-DA | 80.6 | - | - |
| YOLO-HR | 85.59 | 65.0 | 13.2 |
| YOLOv8s | 84.4 | 68.8 | 11.2 |
| YOLO-SE | 86.5 | 70.7 | 13.9 |

The comparison results between YOLOv8 and YOLO-SE on the SIMD dataset are illustrated in Figure 13. From the figure, it is evident that YOLO-SE performs well in focusing on small targets when there are large numbers of objects in an image. Additionally, when an image contains multi-scale targets, YOLO-SE effectively handles targets of various sizes. This comparison demonstrates that YOLO-SE outperforms the YOLOv8 algorithm in terms of detection effectiveness.

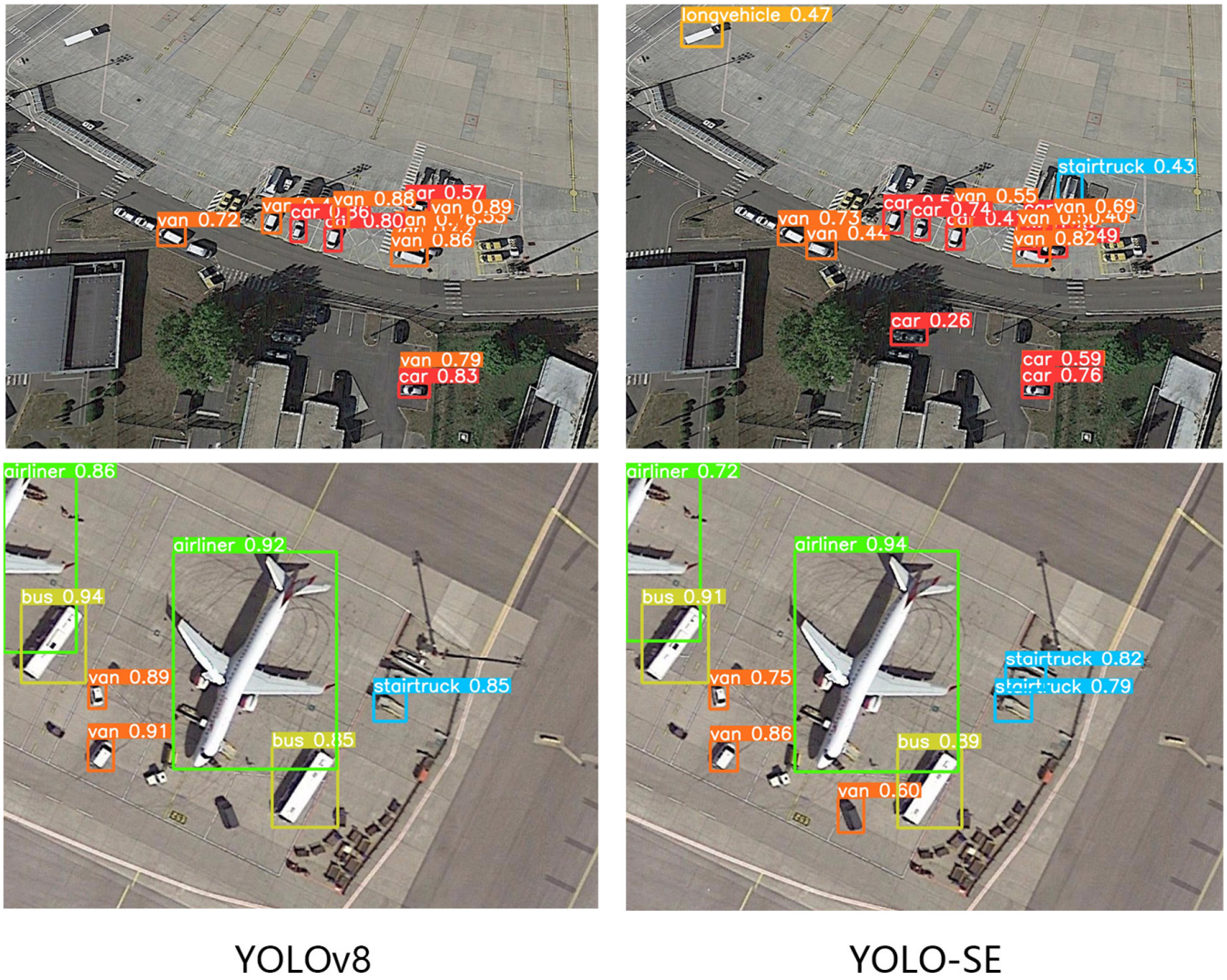



**Figure 13.** The comparison of detection results between YOLOv8 and YOLO-SE on the SIMD dataset.

#### 4.2.3. Ablation Test

To evaluate the impact of various modules in the proposed method, we conducted ablation experiments on the SIMD dataset. The experimental results are shown in Table 5.

**Table 5.** Results of ablation test, "√" represents the selection of the corresponding method.

| | SEF | SPPFE | WIoU | TPH | AP50(%) |
|---|---|---|---|---|---|
| Exp 1 | | | | | 84.4 |
| Exp 2 | √ | | | | 85.3 |
| Exp 3 | √ | √ | | | 85.5 |
| Exp 4 | √ | √ | √ | | 85.9 |
| Exp 5 | √ | √ | √ | √ | 86.5 |

Exp 1 represents the original YOLOv8 model without any modifications. Exp 2 introduces the SEF module on top of YOLOv8. Exp 3 incorporates the SPPFE module, Exp 4 enhances the WIoU loss function, and Exp 5 integrates the TPH module, all built upon the YOLOv8 framework. The results indicate that the SEF module, the SPPFE module, the transformer predict head (TPH), and the introduced WIoU loss function all contribute to improving the algorithm's detection performance.

We conducted a total of four sets of experiments, and from Table 5, it is evident that the SEF module has the most significant impact on improving the algorithm's detection performance, with an increase of 0.9% in AP50. The SPPFE module has improved by 0.2%. The WIoU loss function and TPH module led to improvements of 0.2% and 0.3% in AP50, respectively.

## 5. Conclusions

To address the challenges of multi-scale and small-object detection in remote sensing image detection, this paper introduced the YOLO-SE network based on YOLOv8. First, we successfully introduced the SEF module, a lightweight design that significantly improves network parameters and inference speed. The SEF module effectively handles multi-scale features, providing a solid foundation for the model's performance. Second, by introducing the SPPFE module, we not only improved the efficiency of feature extraction but also successfully introduced the EMA attention mechanism. The multi-scale convolution operations of this module help capture different scale information, thus enhancing the model's accuracy. At the same time, we added an additional prediction head specifically designed for detecting tiny objects. When combined with the other three prediction heads, this allowed us to capture relevant information about small targets more effectively. We also replaced the original detection head with a transformer prediction head (TPH) to capture more global and contextual information. Finally, to improve the training process, we introduced the Wise-IoU bounding box loss function. The use of this loss function helps reduce the negative impact of low-quality instances during training, improving the model's stability and robustness. The experimental results demonstrated that YOLO-SE achieved significant performance improvement on the optical remote sensing dataset SIMD, with a mAP value of 86.5%—a 2.1% increase compared to YOLOv8. This research demonstrated that the introduction of multi-scale convolutions, attention mechanisms, and transformer prediction heads can achieve higher performance in the field of remote sensing image object detection while maintaining a certain level of efficiency. This provides powerful tools and methods for remote sensing image analysis.

In our upcoming efforts, we will aim to further optimize the network model by prioritizing a balance between reduced complexity and improved detection accuracy. We intend to expand the application of the proposed network structure modifications to various object detection algorithms. Moreover, we will investigate alternative strategies for feature reuse and thoroughly address deployment and application challenges associated with the algorithm discussed in this paper.

**Author Contributions:** Conceptualization, T.W.; methodology, T.W.; software, T.W.; validation, T.W.; formal analysis, T.W.; investigation, T.W.; resources, T.W.; data curation, T.W.; writing—original draft preparation, Y.D.; writing—review and editing, Y.D.; visualization, Y.D.; supervision, Y.D.; project administration, T.W.; funding acquisition, T.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data set used in the paper can be downloaded here: GitHub-ihians/simd:SatelliteImageryMulti-ehiclesDataset(SIMD), https://github.com/ihians/simd (accessed on 28 January 2021).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

| | |
|---|---|
| YOLO | You Only Look Once |
| YOLO-SE | You Only Look Once–Slight Edition |
| SEConv | Slight Edition Convolution |
| IoU | Intersection of Union |
| SPP | Spatial Pyramid Pooling |
| SPPF | Spatial Pyramid Pooling with less FLOPs |
| SPPFE | Spatial Pyramid Pooling with less FLOPs Edition |
| C2f | Coarse to Fine |
| SEF | Slight Editor Fine |
| CSP | Cross Stage Partial |
| CIoU | Complete Intersection of Union |
| DFL | Distribution Focal Loss |
| WIoU | Wise Intersection of Union |
| EMA | Efficient Multi-Scale Attention |
| TPH | Transformer Prediction Head |

## References

1. Mao, M.; Zhao, H.; Tang, G.; Ren, J. In-Season Crop Type Detection by Combing Sentinel-1A and Sentinel-2 Imagery Based on the CNN Model. *Agronomy* **2023**, *13*, 1723. [CrossRef]
2. Cardama, F.J.; Heras, D.B.; Argüello, F. Consensus Techniques for Unsupervised Binary Change Detection Using Multi-Scale Segmentation Detectors for Land Cover Vegetation Images. *Remote Sens.* **2023**, *15*, 2889. [CrossRef]
3. Zhang, F.; Du, B.; Zhang, L.; Xu, M. Weakly supervised learning based on coupled convolutional neural networks for aircraft detection. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 5553–5563. [CrossRef]
4. Tang, T.; Zhou, S.; Deng, Z.; Zou, H.; Lei, L. Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. *Sensors* **2017**, *17*, 336. [CrossRef] [PubMed]
5. Zheng, Z.; Lei, L.; Sun, H.; Kuang, G. A review of remote sensing image object detection algorithms based on deep learning. In Proceedings of the 2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC), Beijing, China, 10–12 July 2020; IEEE: New York, NY, USA, 2020; pp. 34–43.
6. Mou, L.; Bruzzone, L.; Zhu, X.M. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 924–935. [CrossRef]
7. Khankeshizadeh, E.; Mohammadzadeh, A.; Moghimi, A.; Mohsenifar, A. FCD-R2U-net: Forest change detection in bi-temporal satellite images using the recurrent residual-based U-net. *Earth Sci. Inform.* **2022**, *15*, 2335–2347. [CrossRef]
8. Purkait, P.; Zhao, C.; Zach, C. SPP-Net: Deep absolute pose regression with synthetic views. *arXiv* **2017**, arXiv:1712.03452.
9. Gkioxari, G.; Hariharan, B.; Girshick, R.; Malik, J. R-cnns for pose estimation and action detection. *arXiv* **2014**, arXiv:1406.5212.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015.
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Paradise, NV, USA, 26 June–1 July 2016; pp. 779–788.
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part I 14. Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
13. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference On Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

14. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
15. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4203–4212.
16. Ma, J.; Hu, Z.; Shao, Q.; Wang, Y.; Zhou, Y.; Liu, J.; Liu, S. Detection of large herbivores in uav images: A new method for small target recognition in large-scale images. *Diversity* **2022**, *14*, 624. [CrossRef]
17. Sun, X.; Wang, P.; Wang, C.; Liu, Y.; Fu, K. PBNet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 50–65. [CrossRef]
18. Lai, H.; Chen, L.; Liu, W.; Yan, Z.; Ye, S. STC-YOLO: Small object detection network for traffic signs in complex environments. *Sensors* **2023**, *23*, 5307. [CrossRef] [PubMed]
19. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
20. Lin, W.; Wu, Z.; Chen, J.; Huang, J.; Jin, L. Scale-Aware Modulation Meet Transformer. *arXiv* **2023**, arXiv:2307.08579.
21. Wan, D.; Lu, R.; Wang, S.; Shen, S.; Xu, T.; Lang, X. YOLO-HR: Improved YOLOv5 for Object Detection in High-Resolution Optical Remote Sensing Images. *Remote Sens.* **2023**, *15*, 614. [CrossRef]
22. Xu, D.; Wu, Y. Improved YOLO-V3 with DenseNet for multi-scale remote sensing target detection. *Sensors* **2020**, *20*, 4276. [CrossRef] [PubMed]
23. Cao, J.; Bao, W.; Shang, H.; Yuan, M.; Cheng, Q. GCL-YOLO: A GhostConv-Based Lightweight YOLO Network for UAV Small Object Detection. *Remote Sens.* **2023**, *15*, 4932. [CrossRef]
24. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
25. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
26. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
27. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
28. Zhou, T.; Wang, S.; Zhou, Y.; Yao, Y.; Li, J.; Shao, L. Motion-attentive transition for zero-shot video object segmentation. In Proceedings of the 2020 AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13066–13073.
29. Zhou, T.; Zhang, M.; Zhao, F.; Li, J. Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4299–4309.
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 10–16 December 2017.
31. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [CrossRef]
32. Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Luan, D.; Sutskever, I. Generative pretraining from pixels. In Proceedings of the International Conference on Machine Learning—PMLR 2020, Virtual, 13–18 July 2020; pp. 1691–1703.
33. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
34. Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in transformer. In Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Virtual, 6–14 December 2021; pp. 15908–15919.
35. Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; Shen, C. Twins: Revisiting the design of spatial attention in vision transformers. In Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021), Virtual, 6–14 December 2021; pp. 9355–9366.
36. Lin, H.; Cheng, X.; Wu, X.; Yang, F.; Shen, D.; Wang, Z.; Song, Q.; Yuan, W. Cat: Cross attention in vision transformer. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; IEEE: New York, NY, USA, 2022; pp. 1–6.
37. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the 2021 IEEE/CVF International Conference On Computer Vision, Montreal, BC, Canada, 1–17 October 2021; pp. 10012–10022.
38. Chen, C.F.; Panda, R.; Fan, Q. Regionvit: Regional-to-local attention for vision transformers. *arXiv* **2021**, arXiv:2106.02689.
39. Zhou, D.; Kang, B.; Jin, X.; Yang, L.; Lian, X.; Jiang, Z.; Hou, Q.; Feng, J. Deepvit: Towards deeper vision transformer. *arXiv* **2021**, arXiv:2103.11886.
40. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the 2021 IEEE/CVF International Conference ON Computer Vision, Montreal, BC, Canada, 1–17 October 2021; pp. 2778–2788.
41. Zhao, Q.; Liu, B.; Lyu, S.; Wang, C.; Zhang, H. TPH-YOLOv5++: Boosting Object Detection on Drone-Captured Scenarios with Cross-Layer Asymmetric Transformer. *Remote Sens.* **2023**, *15*, 1687. [CrossRef]

42. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the 2020 AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
43. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual, 6–12 December 2020; pp. 21002–21012.
44. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; IEEE: New York, NY, USA, 2023; pp. 1–5.
45. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
46. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 20–22 June 2023; pp. 7464–7475.
47. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
48. Lin, J.; Zhao, Y.; Wang, S.; Tang, Y. YOLO-DA: An Efficient YOLO-based Detector for Remote Sensing Object Detection. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 6008705. [CrossRef]