Journal of *Marine Science and Engineering*

MDPI

Article

Advancing Ton-Bag Detection in Seaport Logistics with an Enhanced YOLOv8 Algorithm

Xiulin Qiu [1], Haozhi Zhang [2], Chang Yuan [2], Qinghua Liu [1,*] and Hongzhi Yao [2]

[1] Colleage of Automation, Jiangsu University of Science and Technology, Zhenjiang 212003, China; qiuxiulin@njust.edu.cn
[2] Colleage of Computer, Jiangsu University of Science and Technology, Zhenjiang 212003, China; 221210701230@stu.just.edu.cn (H.Z.); yuanc@stu.just.edu.cn (C.Y.); 221210701227@stu.just.edu.cn (H.Y.)
* Correspondence: liuqh@just.edu.cn; Tel.: +86-139-1455-7059

Abstract: Intelligent logistics and freight transportation is an important part of realizing the intelligence of port terminals. Due to the problems of inaccurate ton bag identification, high costs, large model sizes, and long computation times in traditional freight transportation—issues that hinder meeting real-time requirements on resource-constrained operational equipment—this paper proposes an improved lightweight ton bag detection algorithm, YOLOv8-TB (YOLOv8-Ton Bag), which is optimized based on YOLOv8. Firstly, the improved LZKAC module is introduced to combine with SPPF to form a new SPPFLKZ module, which improves the feature expression performance. Then, with reference to spatial and channel reconstruction convolution and deformable convolution, the C2f-SCTT block is designed for the backbone network, which reduces the spatial and channel redundancy between features in the network. Finally, the C2f-ORECZ block based on a linear scaling layer is designed for the neck, which reduces the training overhead and strengthens the feature learning of the feature extraction network for the targets in the complex background of the harbor and adds the 160 × 160 scale detection head to strengthen small target detection abilities. On the logistics ton bag operation dataset provided by shipping port enterprises, the improved algorithm improves by 3.7% and 5% compared with the original algorithm in mAP50 and mAP50-95, respectively, the model size is reduced by 4.42 MB and the amount of model computation is only 8 G, which is capable of accurately detecting logistics ton bags in real time. The superiority of the method is verified by comparing it with other classical target detection algorithms.

Keywords: port logistics inspection; ton bags; target detection; YOLOv8; attention mechanisms; port congestion

Citation: Qiu, X.; Zhang, H.; Yuan, C.; Liu, Q.; Yao, H. Advancing Ton-Bag Detection in Seaport Logistics with an Enhanced YOLOv8 Algorithm. *J. Mar. Sci. Eng.* **2024**, *12*, 1916. https://doi.org/10.3390/jmse12111916

Academic Editors: Lingxiao Wu and Shuaian Wang

Received: 4 October 2024
Revised: 23 October 2024
Accepted: 25 October 2024
Published: 27 October 2024

1. Introduction

Since the rise of intelligent shipping terminals, research on logistics and transportation automation technology has become more and more in-depth [1]. In port logistics transportation, ton bags are mainly used to transport goods. The original port logistics ton bag freight required manual identification of ton bags, their locations, quantities, and the release of trucks. The traditional manual identification method had problems such as low accuracy, slow speed, and high cost, requiring a large amount of human resources and time, making it difficult to meet the demand for fast and accurate ton bag processing, which significantly lowers the efficiency of port operations. In recent years, the automated detection of freight logistics requirements has become higher and higher, in addition to improving the detection accuracy, due to logistics ton bag detection algorithms are usually deployed to the port monitoring, crane controllers and other equipment, model parameters, the reduction in calculation volume and real-time detection presents a higher demand.

The Ultralytics team proposed the YOLOv8 (You Only Look Once version 8) in 2023 with higher accuracy, smaller parameter counts, and model sizes. From the perspectives of

conjoined ton bag occlusion, small target detection, and lightweight modeling in dense ton bag detection, targeted improvements are made to target detection in port logistics. Based on YOLOv8n, this model is optimized, focusing on the improvement and optimization of the three perspectives of the backbone, neck, and detection layer in the model to improve the accuracy, and for the first time, a YOLOv8n-based ton bag detection algorithm for port logistics YOLOv8-TB is proposed, and the main contributions are as follows:

1. Introduce the new attention module SPPFLKZ (Modified SPPF module with Large Kernel Attention with Convolution) and combine it with the backbone and neck to greatly enhance the feature extraction of small targets.
2. Add the C2f-SCTT block (Modified C2f block with DSRU and DCRU) composed of DSRU (Deformable Spatial Reconstruction Unit) and DCRU (Deformable Channel Reconstruction Unit) reduces the spatial and channel redundancy between features in the network and increases accuracy while achieving lightweight requirements.
3. Design the C2f-ORECZ (Modified C2f block with ORECZ) block based on ORECZ (Online Convolutional Reparameterization Extended Block), which reduces a certain amount of training overhead and strengthens the feature extraction network for feature learning of targets in complex backgrounds of ports, and better adapts to complex and changeable situations with small targets in ton bags.
4. Add a 160 × 160 scale detection layer to strengthen small target detection capabilities and improve the accuracy of small target positioning and identification.

The rest of the paper is organized as follows. In the related work section, traditional portlet detection and deep learning portlet detection-related contents are introduced. YOLOv8-related content is introduced in the YOLOv8 model section. In the YOLOv8-TB model section, the innovative model of port logistics tonnage bag based on improved YOLOv8: YOLOv8-TB proposed in this paper is introduced in detail, and the structure of the method is analyzed in focus. In the results and discussion section, the performance indexes of the model are illustrated through the experimental and analytical results of different algorithms and different improved parts. Finally, the overall innovation of YOLOv8-TB is summarized to illustrate the feasibility and effectiveness of the model improvement.

2. Related Work

2.1. Traditional Small Target Detection in Harbors

Regarding port target detection, domestic and foreign scholars have performed a lot of related work. The traditional research methods for port terminal target recognition mainly include traditional edge detection methods [2], object-oriented extraction methods [3], and feature-based port detection methods for remote sensing images [4]. The research on traditional edge detection methods can be traced back to the phase grouping method [5], which determines its location and attributes based on the partial grayscale change characteristics of the target edge. Liu et al. [6] determined the features based on grayscale features and structural features. Under the object-oriented analysis framework, Bhagavathy et al. [7] proposed a new model which effectively characterizes the port and the targets inside it by learning shared texture features. In terms of feature-based detection methods, Bovolo et al. [8] used radar imagery and combined a hierarchical change detection method. The traditional port detection relies on features built based on prior knowledge. When faced with a background containing complex elements such as waters, buildings, and ships, it is difficult to accurately detect small and medium-sized targets in ports such as ton bags. Characterization has become more difficult.

2.2. Deep-Learning-Based Detection of Small Targets in Harbors

Therefore, deep-learning-based target detection algorithms can break new ground in the field of port logistics and transportation, which are divided into two categories according to the number of detection stages. One class is two-stage detection and the other class is single-stage detection. Classic two-stage detection algorithms include R-CNN [9], Fast-RCNN [10], Faster-RCNN [11], Mask-RCNN [12], etc. Commonly used single-stage

detection algorithms include the YOLO series [13–15] and SSD [16], among others. As far as the requirements of real time and accuracy are concerned, YOLO has received extensive attention from the industry in small target detection. In 2021, Yan et al. [17], based on the Complete Ensemble Empirical Mode Decomposition (CEEMD) algorithm, proposed an adaptive training time-step strategy to enhance feature information interaction and improve the detection capability of small floating targets in sea clutter. In 2023, Zhang et al. [18] improved YOLOv7 and integrated two attention mechanisms to enhance the feature extraction capabilities of the backbone and neck. In 2024, Li et al. [19] designed a new receptive field amplification module, based on YOLOv7 to reduce the model's parameters and expand its receptive field, thereby improving the detection of small targets.

3. YOLOv8 Model

Based on previous YOLO versions, YOLOv8 [20] is divided into YOLOv8n, s, m, l, and x with a total of five different size structures, which are designed to meet the user's needs under different application devices. The difference between the five models is the model parameters, computational volume, and the size of the model, among which YOLOv8n is optimally designed for embedded low-cost devices. Figure 1 shows an architecture represented by YOLOv8n. Compared to the C3 structure of YOLOv5 [21], YOLOv8n replaces it with a C2f (Cross Stage Partial Bottleneck with 2 Conv layers and Feature Fusion) structure with richer gradient flow to enhance the feature fusion capability of convolutional neural networks and improves the inference speed for further lightweight. For different scales of models, the C2f structure is adjusted with different numbers of channels to better adapt to different scales of inputs.

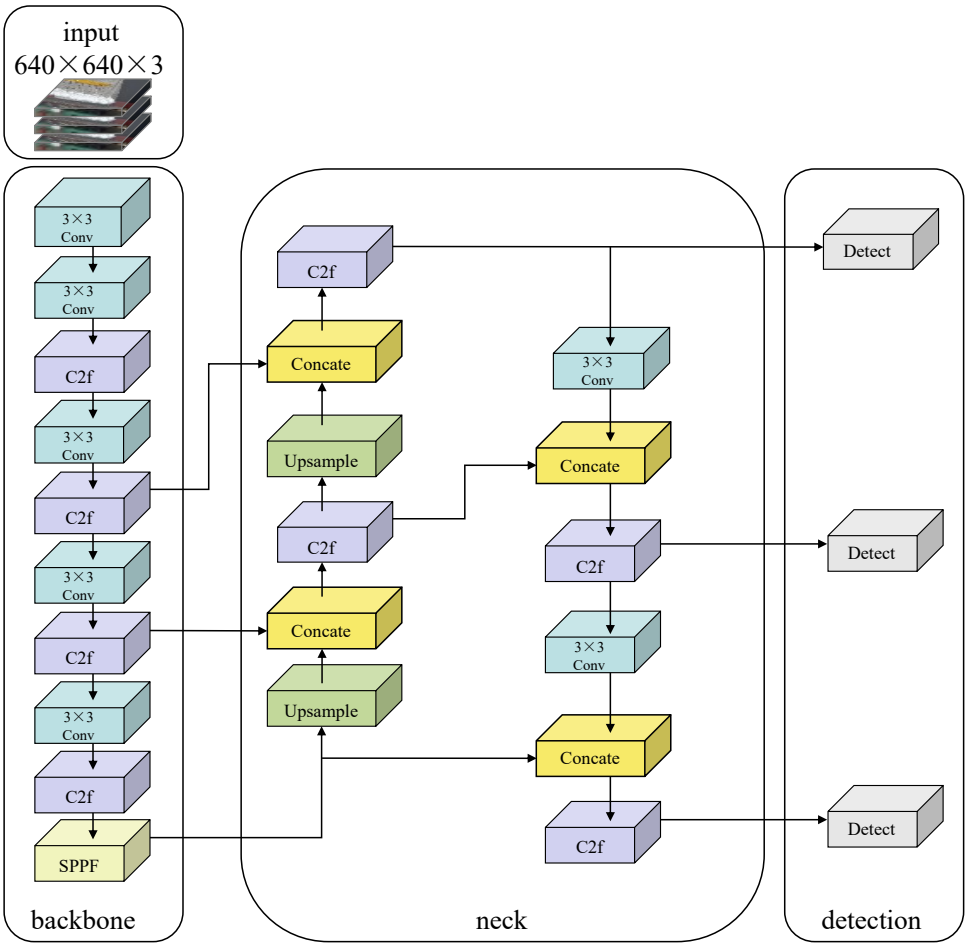


Figure 1. YOLOv8 structure.

4. YOLOv8-Based Ton Bag Innovation Model for Port Logistics: YOLOv8-TB

4.1. YOLOv8-TB Structure

The task of logistics ton bag detection is usually limited by operational equipment (port area monitoring system), which requires a lightweight, low-latency, and high-accuracy model. The structure of the YOLOv8-TB proposed in this paper is detailed in Figure 2. This network consists of four parts, which are input, backbone, neck, and output.

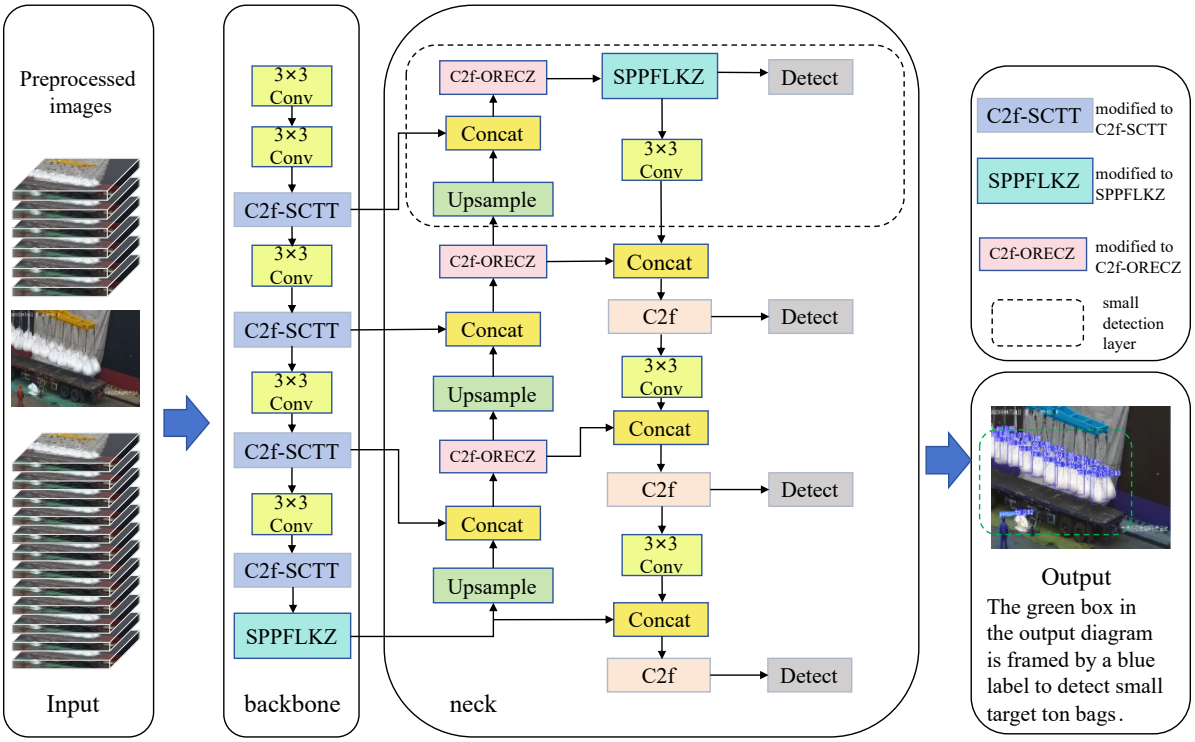


Figure 2. YOLOv8-TB structure.

YOLOv8-TB optimizes the original YOLOv8 structure, improves the small target detection effect by improvement, and achieves the light weight of the model under the premise of improving the detection performance. This paper mainly carries out the following improvements.

First, in the backbone part, the improved LZKAC (Large Kernel Attention with Convolution) self-attention mechanism module is combined with SPPF (Spatial Pyramid Pooling Fast) to form a new SPPFLKZ module to replace the original SPPF, which improves the feature expression performance and strengthens the feature learning of the feature extraction network for the target ton bag in the complex port background.

Second, also in the backbone part, C2f is replaced by C2f-SCTT, which enables the model to better extract capabilities, reduces the computational cost and model storage, and ensures that the performance and the model magnitude are balanced with each other.

Third, in the neck part, the upsampled C2f is replaced with the ORECZ-based C2f-ORECZ module, which further reduces the model parametric quantities while maintaining the feature extraction performance.

Fourth, in the neck part, as shown in the dotted box in Figure 2, a 160 × 160 scale detection neck is added, in which the C2f-ORECZ block is used and the SPPFLKZ attention module is added in the downsampling process to improve the accuracy of the detection of small targets.

The technical details related to the SPPFLKZ attention mechanism, the C2f-SCTT block, the C2f-ORECZ block, and the small target detection layer are described below.

4.2. SPPFLKZ Attention Mechanism

4.2.1. LZKAC

Large Kernel Attention (LKA) [22] is a novel attention mechanism for visual tasks proposed by Tsinghua and Nankai in 2022, on which the City University of Hong Kong & TCL AI Lab proposed the attention mechanism of Large Separable Kernel Attention(LSKA) [23] in 2023. LKA and LSKA generally contain the advantages of self-attention [24], such as adaptivity and distance dependence.

Based on the original LKA and LSKA, an innovative decomposition of the large kernel convolution operation is carried out to address the respective deficiencies of self-attention and large kernel convolution, as a way to capture long-term spatial relations more effectively. Based on this improvement idea, the LZKAC module is proposed. This module aims to take advantage of the global sensing ability of self-attention and the extensive sensing field of the big kernel convolution; by optimizing the computational steps and reducing parameters, it effectively enhances the model's ability to capture long-distance dependencies, while minimizing the resource consumption as much as possible.

The LZKAC module captures long-term relationships by performing decomposition operations on the large kernel convolution using a few computations and parameters. The large kernel convolution is divided into 3 parts: spatial local convolution (deep convolution), i.e., DW-Conv, spatial remote convolution (deep dilation convolution), i.e., DW-D-Conv, and channel 1×1 convolution, i.e., Conv. As shown in Figure 3, demonstrating the network structure of LZKAC, in order to capture the local spatial information first undergoes a kernel size of $1 \times (2d-1)$ deep convolutions for extracting features in one direction (usually horizontal or vertical) while reducing the number of parameters. Then, in order to extract features in another direction, complementary to the previous step, a $(2d-1) \times 1$ depth convolution is performed, followed by a $[k/d][k/d]$ depth expansion convolution to further extract and integrate the features extracted in the previous two steps, and finally, a 1×1 convolution is performed and multiplied element-by-element with the original input feature maps to obtain the output results.

Given an input feature map $\mathrm{F} \in R^{C \times H \times W}$, where C is the number of input channels and H and W denote the height and width of the feature map, respectively, a new and improved large kernel convolutional decomposition configuration, namely the LZKAC module, is obtained by decomposing the conventional two-dimensional depth convolution kernel into two cascaded one-dimensional separable convolution kernels. The outputs of the LZKAC can be obtained by the following Equations (1)–(4), where d is the expansion rate. The input feature mapping F^C is convolved with two cascaded one-dimensional separable depth convolutions W with kernel sizes of $1 \times (2d-1)$ and $(2d-1) \times 1$, and each channel C in F is rolled up with the corresponding channel in W. Finally, $\bar{Z}^C$ is computed by the following equation and this output captures the local spatial information:

$$\bar{Z}^C = \sum_{H,W} W^C_{(2d-1)\times 1} * \left(\sum_{H,W} W^C_{1\times(2d-1)} * F^C \right) \tag{1}$$

The output $\bar{Z}^C$ of Equation (1) is convolved with the large kernel convolutional decomposition of the kernel size $[k/d][k/d]$ for the depth dilation convolution W. Convolutional operation is performed and compensates for the lattice effect of the depth dilation convolution, and Z^C is computed by the following equation:

$$Z^C = \sum_{H,W} W^C_{\lfloor \frac{k}{d} \rfloor \times \lfloor \frac{k}{d} \rfloor} * \bar{Z}^C \tag{2}$$

The output Z^C of Equation (2) is convolved with a kernel size of 1×1 convolution W. The attention map A^C is obtained by the following equation:

$$A^C = W_{1\times 1} * Z^C \tag{3}$$

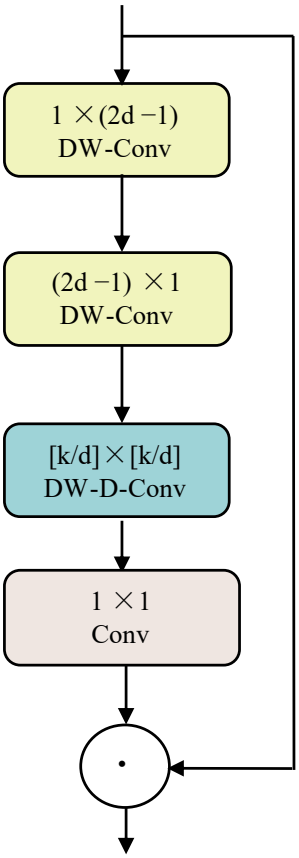


Figure 3. LZKAC structure.

Finally, the obtained attention map A^C from Equation (3) is subjected to element-wise multiplication with the input feature map F^C, resulting in the output $\bar{F}^C$ calculated by Equation (4):

$$\bar{F}^C = A^C \odot F^C \tag{4}$$

where $*$ and $\odot$ denote the convolution operation and Hadamard product, respectively. This decomposition of large kernel convolution helps to alleviate the problem of quadratic increase in computation, reducing the computational cost of feature extraction by only incurring the cost associated with depth-wise convolution and larger kernel sizes.

It is assumed that the input size and output size of the feature mapping to LKA and LZKAC are the same. Equations (5)–(8) below provide the computational equations to derive the floating point operands and parameters for LZKAC and LKA, where k is the size of the core and d is the expansion rate. From the comparison of Equations (5) with (7) and (6) with (8), the LZKAC proposed in this paper preserves $\frac{2d-1}{2}$ more effective parameters than the original LKA, and thus, the LZKAC is more accurate in target feature extraction:

$$Param = (2d-1) \times C \times 2 + \left\lfloor \frac{k}{d} \right\rfloor^2 \times C + C \times C \tag{5}$$

$$\text{FLOPs} = \left((2d-1) \times C \times 2 + \left\lfloor \frac{k}{d} \right\rfloor^2 \times C + C \times C \right) \times H \times W \tag{6}$$

$$Param = (2d-1)^2 \times C + \left\lfloor \frac{k}{d} \right\rfloor^2 \times C + C \times C \tag{7}$$

$$\text{FLOPs} = \left((2d-1)^2 \times C + \left\lfloor \frac{k}{d} \right\rfloor^2 \times C + C \times C \right) \times H \times W \tag{8}$$

Overall, LZKAC combines the advantages of large kernel convolution and self-attention, effectively improving the model's adaptability in both spatial and channel dimensions.

4.2.2. SPPFLKZ

Figure 4 below shows the SPPFLKZ attention mechanism module reconstructed using LZKAC, consisting of LZKAC, MaxPool2d (Spatial Pyramid Pooling Fast), and Concat layers.

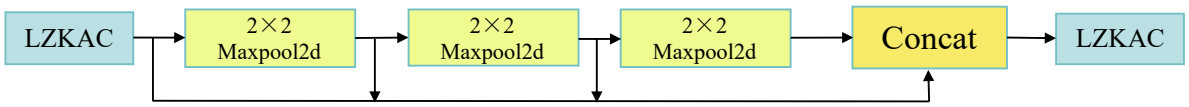


Figure 4. SPPFLKZ structure.

The SPPFLKZ attention mechanism is an automated feature selection method that replaces the convolutional layer in the traditional SPPF by integrating the LZKAC module. This mechanism integrates local contextual information, a large perceptual range, and dynamically changing features. This mechanism can dynamically select important features that contribute to the task based on the features of the incoming data, while automatically filtering out irrelevant noise. In this way, the SPPFLKZ attention mechanism significantly improves the feature representation. Therefore, this paper concludes that there exists feasibility to optimize the SPPFLKZ attention mechanism in combination with YOLOv8n for small target detection.

4.3. C2f-SCTT

SCConv (Split Convolution) [25] was proposed by research from a team consisting of researchers from the East China Normal University and Tongji University in 2023. The goal of this module is to reduce the computational cost due to redundant feature extraction in vision tasks. In response to the problems of accuracy degradation and insufficient reduction of redundant information in SCConv, the idea of Deformable Conv [26] is incorporated into the original SCConv for the construction of DSRU and DCRU in the C2f-SCTT module.

Figure 5 illustrates the concept of Deformable Conv. Deformable Conv has a main and subbranch structure, where an offset is added to each convolutional sampling point, and the subbranch is responsible for learning the offset through a 3 × 3 convolutional layer, interpolating operations based on the offset generated by the subbranch to the main branch, and then performs normal convolution. As a result, deformable convolution can better extract the complete features of the target object.

In summary, the offset introduced in deformable convolution is to find valid information in the right place, and a coefficient is introduced as a positional weight. This makes the accurate extraction of valid information better. As shown in Figure 6, the C2f-SCTT module replaces the original two bottlenecks with DSRU and DCRU sequentially. C2f-SCTT is designed to effectively limit feature redundancy, not only by reducing model parameters and FLOPs but also by enriching representation features.

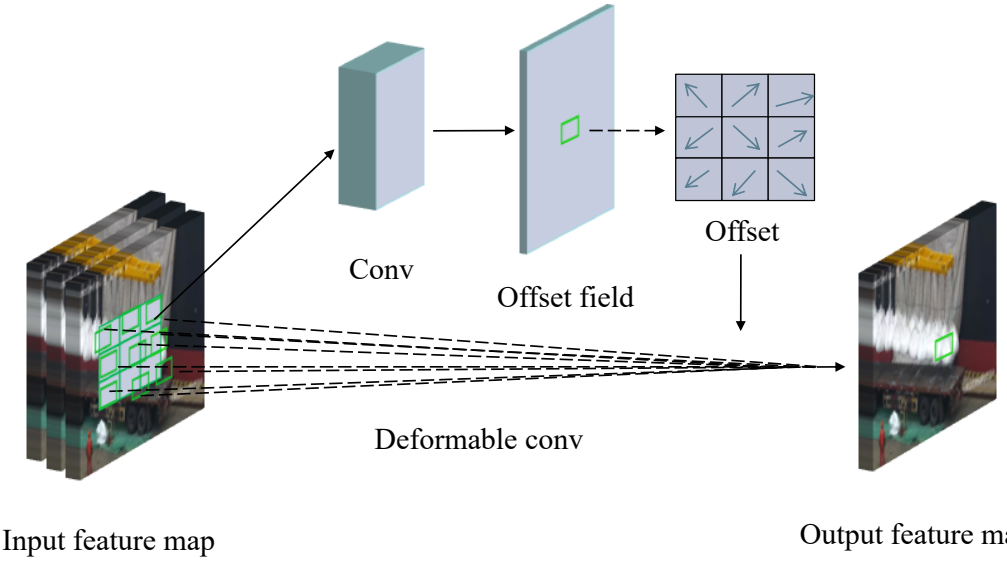


Figure 5. Deformable conv.

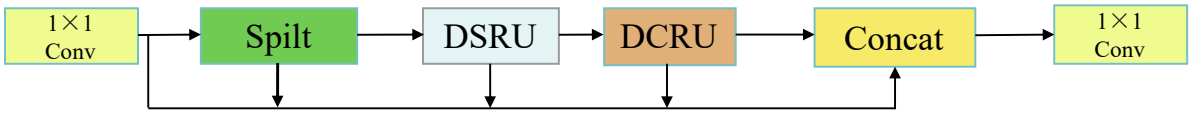


Figure 6. C2f-SCTT structure.

4.3.1. DSRU

In Figure 7, the DSRU (Deformable Spatial Reconstruction Unit) adopts a main and secondary branch structure to optimize feature extraction by integrating deformable convolution and dynamic gating mechanisms. The whole process includes two main parts: feature separation and feature reconstruction.

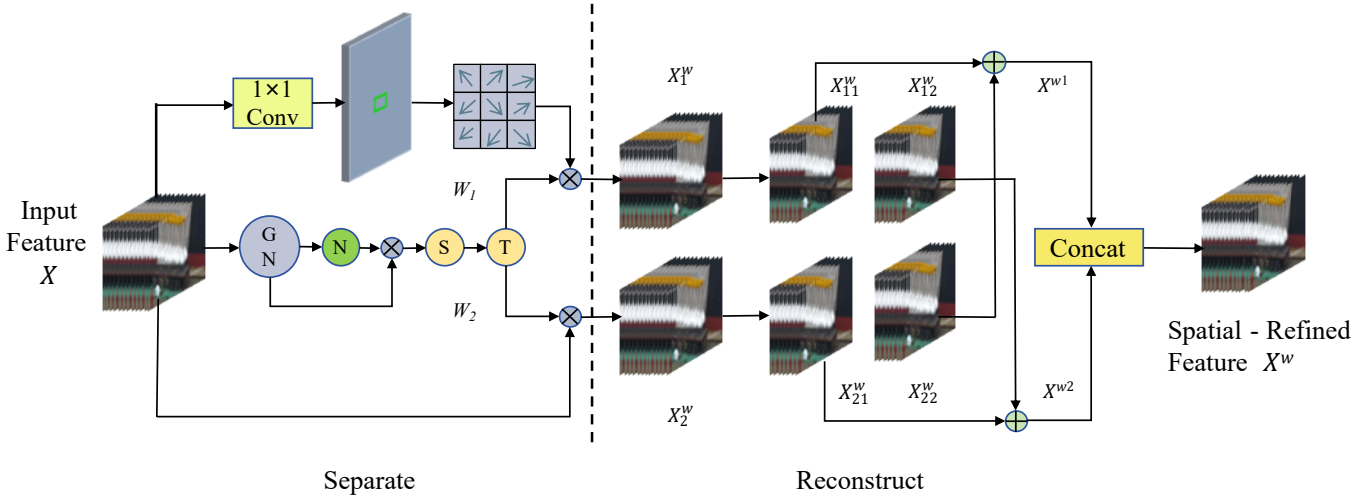


Figure 7. DSRU structure.

A deformable offset offset is introduced in the separation part. Input feature map $X \in R^{C \times H \times W}$, where N is the batch axis, C is the channel axis, and H and W are the spatial height and width axes. Learning offset by a convolutional layer on the subbranch is used to adjust the sampling points of the convolutional layer on the main branch. The trainable parameter $\gamma \in R^C$ in the GN (Group Normalization) layer is utilized on the main branch as a measure of pixel variance per batch and channel space. After N layers, normalized

weights $W_\gamma \in R^C$ related to each batch of data and channel are obtained, and the result is multiplied with the feature map processed through the GN layer. Then, the weights of the feature map reweighted by W_γ in the range (0, 1) are generated by a Sigmoid function and controlled by a threshold T-layer to generate two sets of weights: information weights W_1 (above the threshold) and non-information weights W_2 (below the threshold), thereby separating and forming the dual branches.

In the reconstruction part, two weighted feature maps X_1^W with enriched information and X_2^W with less informative features, are obtained by multiplying the feature map obtained after deformable interpolation by W_1 and the original input feature map X by W_2. X_1^W contains spatial content that is informative and expressive, while X_2^W contains redundant and less useful information.Subsequently, the cross-reconstruction operation is used to mix them in order to enhance the exchange of information between different features. Finally, the cross-reconstructed features X_1^W and X_2^W are spliced together to obtain the spatially refined feature resultant map X^W, and the whole reconstruction process yields through the following Equations (9)–(13):

$$X_1^W = W_1 \otimes X \tag{9}$$

$$X_2^W = W_2 \otimes X \tag{10}$$

$$X_{11}^W \oplus X_{22}^W = X^{W1} \tag{11}$$

$$X_{21}^W \oplus X_{12}^W = X^{W2} \tag{12}$$

$$X^{W1} \cup X^{W2} = X^W \tag{13}$$

After passing the input feature map through DSRU, not only is the distinction between feature-rich and less informative features effectively made, but a series of reconstruction steps are also employed to enhance the expressive power of these features and suppress unnecessary redundancy in spatial dimensions. However, the spatially refined feature maps still maintain redundancy in the channel dimension.

4.3.2. DCRU

Figure 8 illustrates the two-branch structure of the DCRU (Deformable Channel Reconstruction Unit), where the features are processed through the three stages shown in Figure 8.

In the split stage, The result obtained after applying the DSRU operation, denoted as $X \in R^{C \times H \times W}$, is used as the input to the DCRU to further extract rich representative features. Firstly, the channels of X^W are divided into two parts, a parameter α $(0 \leq \alpha \leq 1)$ is set to control the allocation ratio of the channels to ensure the efficiency while balancing the computational cost, and the number of channels of the feature mapping is compressed using the 1×1 convolution operation to improve the processing speed. After segmentation and compression, the spatially refined feature X^W is divided into upper part X_{up} and lower part X_{low}.

In the transformation part, the upper branch X_{up} acts as a "rich feature extractor" to extract feature information and reduce computational cost by parallel GWConv (group convolution) and PWConv (point-by-point convolution), where group convolution reduces the computational burden and restricts the flow of information between different channel groups, point-by-point convolution compensates for possible information loss, and finally, the output feature results are summed to form a combined representative feature map Y_1. The lower branch X_{low} is input to the lower transformation stage, and the feature mappings with shallow hidden details are extracted as a complement to the upper branch

through deformable interpolation and splicing of the original X_{low}. Finally, the output of Y_2 is formed.

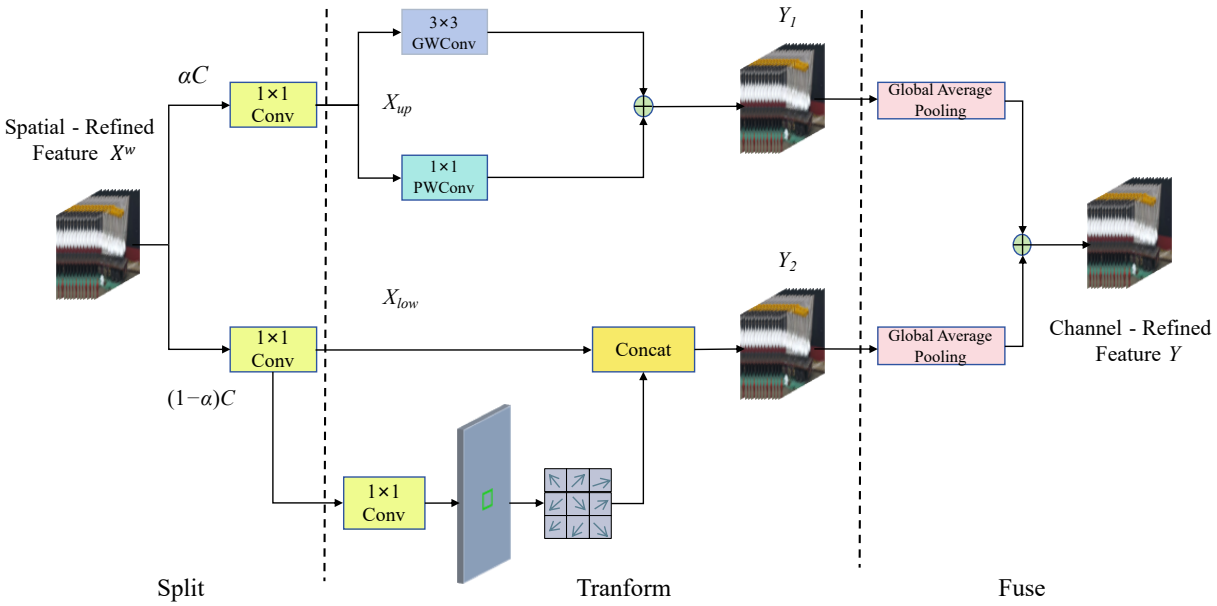


Figure 8. DCRU structure.

After the conversion is completed in the fusion stage, Y_1 and Y_2 undergo a global average pooling to collect the global spatial features, respectively. Finally, the upper and lower features Y_1 and Y_2 are merged to obtain the channel-refined feature Y.

In short, The DCRU uses a three-phase strategy to reduce channel redundancy, which effectively reduces the redundancy along the channel dimension. In addition, DCRU utilizes lightweight convolution to extract rich representative features, while feature redundancy is performed through low-consumption operations and feature reuse.

4.4. C2f-ORECZ

By referring to the advantages of online convolutional reparameterization (OREPA) [27], ORECZ, a new convolutional module, is constructed by combining it with convolutional neural network (CNN) for the problem of accuracy degradation in OREPA. The structure of the ORECZ module is shown in Figure 9, which consists of a block linearization stage and a block compression stage.

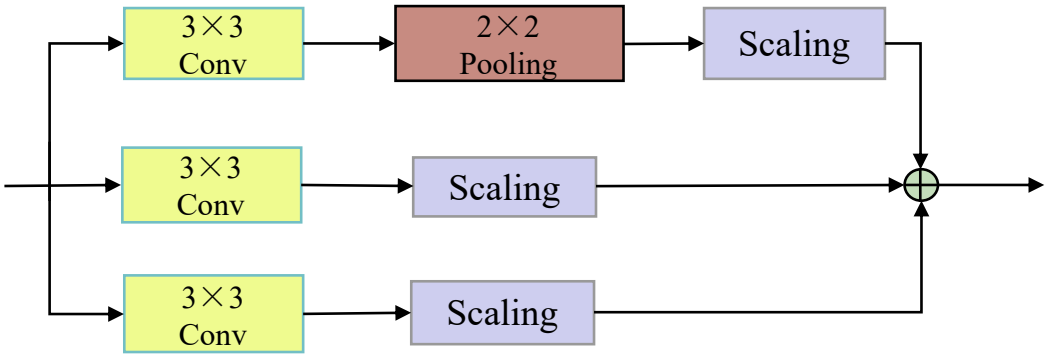


Figure 9. ORECZ structure.

In the block linearization stage, a Scaling layer (linear scaling) is used to replace the traditional normalization layer. The Scaling layer contains a learnable vector for scaling the feature maps in the channel dimension to motivate the network to prevent inter-layer

dependencies during the training period and to allow branches to move independently in different directions.

In the block compression stage, following linearization, all linear layers of the network are concentrated in a re-parameterization block (re-param block), allowing for holistic optimization during the training process. The structure consisting of multiple linear layers in the three branches is compressed and merged into an ORECZ block by converting the operations on the feature mapping in the middle of the linear block to more efficient kernel operations. This reduces the additional training cost of the reparameterization from $O_{(H\times W)}$ to $O_{((K\_H,K\_W))}$, where (H, W), (K_H, K_W) are the spatial shapes of the feature maps and the convolution kernel.

Overall, the ORECZ module not only maintains the detection performance but also reduces the parameters through the linearization and compression process and ensures the diversity of different optimization paths with the network's representational capabilities. A three-branch structure compression is used to merge into a single linear block for feature extraction, which reduces the complex training time block to a single convolutional layer and maintains high accuracy.

Figure 10 illustrates the structure of the C2f-ORECZ module, where the ORECZ module replaces the Bottleneck layer in C2f. To maintain optimization diversity and stabilize the training process, a BN (Batch Normalization) layer and a Relu (Rectified Linear Unit) layer are sequentially added before the concert operation. The excessive use of BN and Relu layers has been shown to introduce significant computational overhead during the training stage, leading to excessively large model weights in past reparameterization models. Therefore, considering efficiency, the C2f-ORECZ module in this study employs only one BN layer and one Relu layer, reducing certain training overheads.

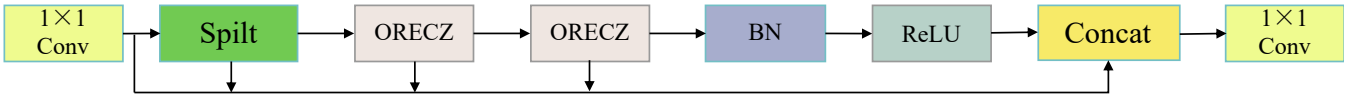


Figure 10. C2f-ORECZ structure.

4.5. Small Detection Layer

The research scenario is a port logistics scenario, which contains many small targets, such as ton bags and operational workers. After feature fusion of the original YOLOv8, the detection layer outputs three feature maps with different sizes for detecting targets of different sizes, which are 20 × 20, 40 × 40, and 80 × 80, respectively. Since the maximum feature map is only 80 × 80, the feature map scale is small and not suitable for detecting small targets. To improve the accuracy of tonnage bag inspection, while avoiding too much loss of detailed features due to downsampling, this method is improved for the original YOLOv8 network structure. The dashed box part shown in Figure 2 is the added 160 × 160 small target detection layer, YOLOv8-TB initially extracts features from the sixth layer of the backbone and uses Concat splicing to fuse the shallow features extracted from the neck with the contextual information extracted by the SPPFLKZ attention mechanism, and finally, output the fourth detector header, which is used as the small target detection header by enhancing the extraction of the feature details of small targets to enhance the detection capability of YOLOv8-TB for small targets. Under the condition of reducing the leakage and false detection rate of small targets, deeper feature transfer and feature fusion are carried out, which makes small target localization and recognition more accurate.

5. Results and Discussion

5.1. Data Preparation

Due to factors such as individual differences in ton bags (e.g., cargo volume and model), mutual occlusion, different lighting conditions, and viewing angles in real port operation environments, the dataset needs to be representative and highly diverse.

This paper is aimed at the environment of port operation, where existing datasets for ton bag detection are relatively scarce, with insufficient image annotations, and often contain blurred images. Therefore, it is difficult for public datasets to fulfill the research content of this paper and serve as an ideal benchmark for challenging port logistics scenarios.

Therefore, this paper cooperates with relevant port enterprises, obtains the logistics operation video of relevant ports for the port logistics scenario, obtains a certain amount of data through frame extraction, and forms a dataset for the training and validation of the final model by means of data enhancement. The dataset contains four classes of objects, including ton bags, trucks loaded with ton bags, operation workers, and empty trucks. The four categories are labeled dai, truck, person, and emptytruck.

The distribution of the four categories in the original dataset of 6000 images obtained from frame extraction is shown in Table 1. Since the main research focus of this study is the detection of ton bags, we selected samples based on typicality and complexity criteria. This selection includes samples with representative characteristics under complex conditions, such as heavy occlusion in the background, unclear object contours, or varying lighting conditions. After filtering 2000 images, the distribution ratio of the four categories (dai, truck, person, emptytruck) was 4:2:2:2.

Table 1. The number of raw data in each category.

Category	Number
Dai	95,453
Truck	31,263
Person	14,589
Emptytruck	16,728

To address the data imbalance issue, data augmentation techniques were applied to balance the distribution of the four categories, while also increasing the diversity and complexity of the samples. Ultimately, the dataset was expanded to 3000 images, and the distribution of the four categories in the 3000 images is shown in Table 2, ensuring that the data balance among the categories was maintained.

Table 2. The number of each category after data augmentation.

Category	Number
Dai	26,453
Truck	25,271
Person	25,589
Emptytruck	25,728

The dataset for this article has 3000 images, which are divided into train, val, and test. To ensure balanced data distribution after the split, each category was separately divided into the training, validation, and test sets, ensuring that the proportions of each category remain consistent across all sets.The data are divided according to 7:2:1, including 2100 for the training set, 600 for the validation set, and 300 for the test set, and the size of the input images is uniformly 640×640.

The dataset is also processed for clarity and contour, and the metrics are suitable for better evaluation of ton-bag detection algorithms in heavily occluded or poorly illuminated scenarios, as well as being very challenging.

5.2. Experimental Environment and Parameter Configuration

The operating system of this experimental platform is Windows 10, the CPU is Inter (R) Core (TM) i9-12900H @2.9GHz, the RAM is 16 GB, and the graphics card is Nvidia GeForce RTX 3060 with a total of 6 GB of video memory. The deep learning framework uses

Pytorch-GPU 2.0.1, the CUDA version used for 11.3, and cuDNN with version v8.9.1.23 for GPU acceleration, and Table 3 demonstrates the corresponding parameters.

Table 3. Parameter configuration.

Parameters	Settings	Parameters	Settings
Optimizer	SGD	lrf	0.01
Epochs	400	weight_decay	0.0005
Batchsize	8	momentum	0.937
Workers	2	warmup_epochs	3
Imgs	640	warmup_momentum	0.8
lr0	0.01	close_mosaic	10

5.3. Evaluation Indicators

In this experiment, precision P, recall R, mean Average Precision mAP, model computational power (GFLOPs), and model size are used as performance reference indicators. Among them, P, R, and mAP are used as indicators to evaluate the model detection effect, and the larger their values, the higher the model detection accuracy. GFLOPs and model size are used to evaluate how lightweight the model is; smaller values represent a higher degree of model being lightweight and lower hardware performance requirements.

5.4. Experimental Results

In order to validate the effectiveness of the SPPFLKZ attention mechanism, C2f-SCTT, C2f-ORECZ, and the small target detection head were used in this experiment. In this paper, ablation experiments and comparison experiments are set up to investigate the performance impact of the proposed improved method on the YOLOv8n model.

5.4.1. SPPFLKZ Validity Analysis

In this paper, we conduct comparative experiments for the SPPFLKZ attention mechanism with other mainstream attention mechanisms under the same conditions (uniform configurations as well as the same dataset), including six attention mechanisms, namely ACmix [28], SE [29], EMA [30], ECA [31], LKA, and LSKA.

According to Table 4, SPPFLKZ significantly performs better than ACmix, SE, EMA, ECA, LKA, and LSKA in accuracy metrics such as Precision, Recall, etc. Despite the fact that SPPFLKZ is slightly larger than some of the models in terms of parameter calculations and model sizes, it still leads in terms of overall accuracy. This suggests that SPPFLKZ performs well across multiple evaluation metrics, demonstrating its combined strength in effectiveness.

Table 4. Comparative trials of attention mechanisms.

Model	Precision	Recall	mAP50	mAP50-95	FLOPs/G	Model Size/MB
ACmix	94.2	91.1	95.1	77.2	9.3	13.21
SE	93.8	90.3	92.7	76.8	9.1	12.01
EMA	93.7	90.7	92.5	76.6	9.0	10.11
ECA	93.4	90.5	94.7	76.3	9.1	11.15
LKA	94.3	91.2	94.9	77.4	9.4	13.16
LSKA	94.2	90.6	94.6	76.9	9.3	12.06
SPPFLKZ	94.5	91.9	95.3	78.1	9.2	12.03

5.4.2. Model Comparison Experiment

In this paper, nine classical target detection methods including SSD, Faster R-CNN, YOLOv3-tiny, YOLOv4-tiny, YOLOX-s, YOLOv5s, YOLOv7-tiny, YOLOv8n, and YOLOv9-c [32–34], while small target detection algorithms are used in the target detection comparison experiments to compare their performance with the present YOLOv8-TB algorithm under the same conditions.

According to Table 5, YOLOv8n has higher Precision, Recall, and mAP than SSD, Faster R-CNN, YOLOX-s, YOLOv5s, YOLOv7-tiny, and YOLOv9, while the amount of parameter computation and the size of the model is smaller than that of the other six networks. YOLOv8n has a larger parameter computation and model size is larger than YOLOv3-tiny and YOLOv4-tiny, but Precision, Recall, and mAP are much larger than these two networks. Compared to the original YOLOv8n, YOLOv8-TB not only achieved a 3.7% and 5% improvement in mAP50 and mAP50-95, respectively, but also increased precision by 4.3%. Additionally, FLOPS was reduced by 0.9 G, and the model size decreased by 4.42 MB. Compared with the related literature algorithms in recent years, such as Refs. [33,34], this paper's algorithm shows its superiority in all indicators.

Table 5. Model comparison experiment.

Model	Precision	Recall	mAP50	mAP50-95	FLOPs/G	Model Size/MB
SSD	79.6	23.8	42.1	56.3	34.8	46.1
Faster R-CNN	80.8	71.5	76.2	62.7	206.6	108
YOLOv3-tiny	61.7	56.8	62.5	51.2	5.6	5.8
YOLOv4-tiny	63.5	58.2	65.7	49.8	7.0	6.4
YOLOX-s	77.1	62.4	76.5	59.4	26.8	39.3
YOLOv5s	74.9	59.7	73.6	56.1	16.5	27.1
YOLOv7-tiny	81.6	78.2	82.5	66.9	13.9	18.6
YOLOv8n	91.4	91.7	92.7	76.1	8.9	11.3
YOLOv9-c	90.1	88.9	91.8	75.1	238.9	98.6
33	93.3	89.6	92.3	76.2	8.5	7.11
34	93.5	89.1	92.4	76.6	8.6	7.32
YOLOv8-TB	95.7	92.2	96.4	81.1	8.0	6.88

5.4.3. Ablation Experiment

In order to verify the effectiveness and superiority of each improvement module of this paper's algorithm YOLOv8-TB, ablation experiments are carried out by different combinations of multiple improvement modules using Precision, Recall, mAP@50, mAP@50-95, FLOPs/G, and model size/M as evaluation indexes. Table 4 shows the corresponding experimental data.

Table 6 shows the comparative results of the ablation experiments, indicating that the performance of several YOLOv8-TB improvement modules has been enhanced. Furthermore, the effectiveness of YOLOv8-TB can be concluded by the combination of different improvement modules. The model with the SPPFLKZ module added improves by 3.1%, 2.6%, and 2% over YOLOv8n in terms of precision, mAP50, and mAP50-95, respectively, indicating that the SPPFLKZ module based on the LZKAC self-attention mechanism is more effective in improving the feature expression performance when performing feature extraction. In comparison with YOLOv8n, the models with the individual addition of the C2f-SCTT and C2f-ORECZ modules showed improvements in precision, mAP50, and mAP50-95, while significantly reducing flops and model size. Additionally, the model that combines both C2f-SCTT and C2f-ORECZ achieved a 1.7% increase in precision, a 2.2% increase in mAP50, and a 0.8% increase in mAP50-95, along with a reduction of 1.5 G in flops and 6.23 MB in model size compared to YOLOv8n. These performance comparisons indicate that the use of C2f-SCTT and C2f-ORECZ in the network significantly reduces spatial and channel redundancy, while also improving small object detection capabilities. Compared to YOLOv8n, the model with the addition of the small target detection layer improved by 2.7% and 1.6% on mAP50 and mAP50-95, respectively, indicating that the addition of the small-target detection layer improves the accuracy of the small-target detection and proves the effectiveness of the small-target detection layer.

Table 6. Ablation experiments.

Small Detection Layer	SPPFLKZ	C2f-SCTT	C2f-ORECZ	Precision	Recall	mAP50	mAP50-95	FLOPs/G	Model Size/MB
×	×	×	×	91.4	91.7	92.7	76.1	8.9	11.3
✓	×	×	×	94.8	90.3	95.4	77.7	9.2	12.18
×	✓	×	×	94.5	91.9	95.3	78.1	9.2	12.03
×	×	✓	×	92.5	91.1	94.5	75.2	8.2	7.67
×	×	×	✓	93.9	90.2	94.7	74.8	8.1	7.51
✓	✓	×	×	94.6	91.6	96.4	79.8	9.5	13.49
✓	×	✓	×	93.2	91.7	95.2	77.1	8.5	8.78
✓	×	×	✓	95.1	91.3	95.6	78.4	8.4	8.45
×	✓	✓	×	94.3	90.9	94.6	79.2	8.4	8.50
×	✓	×	✓	94.6	91.8	95.7	78.6	8.5	8.67
×	×	✓	✓	93.1	90.6	94.9	76.9	7.4	5.07
✓	✓	✓	×	95.1	91.6	96.2	79.6	8.8	9.44
✓	✓	×	✓	93.7	90.1	96.3	79.2	8.7	9.15
×	✓	✓	✓	94.8	91.2	96.2	79.1	7.7	5.43
✓	×	✓	✓	95.3	91.8	96.1	79.4	7.8	5.62
✓	✓	✓	✓	95.7	92.2	96.4	81.1	8.0	6.88

Note: '×' indicates that the component is not added to the model, while '✓' indicates it is added to the model. Bold values highlight the highest scores.

5.4.4. Cross-Validation Experiment

To evaluate the robustness of the model, this paper introduces K-fold cross-validation, with the performance metrics shown in Table 7. Through 5-fold cross-validation, the model demonstrated stable performance across different validation sets. The Precision values for each fold were 95.6%, 95.7%, 95.9%, 95.8%, and 95.5%, respectively. The mAP50 values for each fold were 96.3%, 96.5%, 96.4%, 96.6%, and 96.2%, while the mAP50-95 values were 80.8%, 81.1%, 80.9%, 81.2%, and 80.8%. The final average Precision was 95.7%, the mAP50 was 96.4%, and the mAP50-95 was 80.9%. These results indicate that YOLOv8-TB consistently maintains high detection accuracy and generalization ability across different dataset partitions.

Table 7. The results of the cross-validation.

Folds	Precision	Recall	mAP50	mAP50-95
1	95.6	92.2	96.3	80.9
2	95.7	92.4	96.5	81.1
3	95.9	92.3	96.4	80.9
4	95.8	92.5	96.6	81.3
5	95.5	92.1	96.2	80.8
Average	95.7	92.3	96.4	81

Overall, this paper proposes the YOLOv8-TB model and comprehensively validates it through 5-fold cross-validation. The experimental results show that the model performs well across multiple dataset partitions, further confirming its robustness and generalization ability. Cross-validation not only reduces the bias caused by random splits but also provides a more comprehensive evaluation of the model's practical application value.

5.5. Algorithm Effect Verification

In this experiment, three representative scenarios in the port logistics ton-bag dataset are selected to be divided into three groups, A, B, and C. From left to right, we show the comparison of the detection effects of YOLOv8-TB [33], YOLOv9, YOLOv8n, and YOLOv7-tiny in daytime operation, nighttime operation, and ton-bag contiguous blocking situations. Figure 11 illustrates the corresponding detection effect.

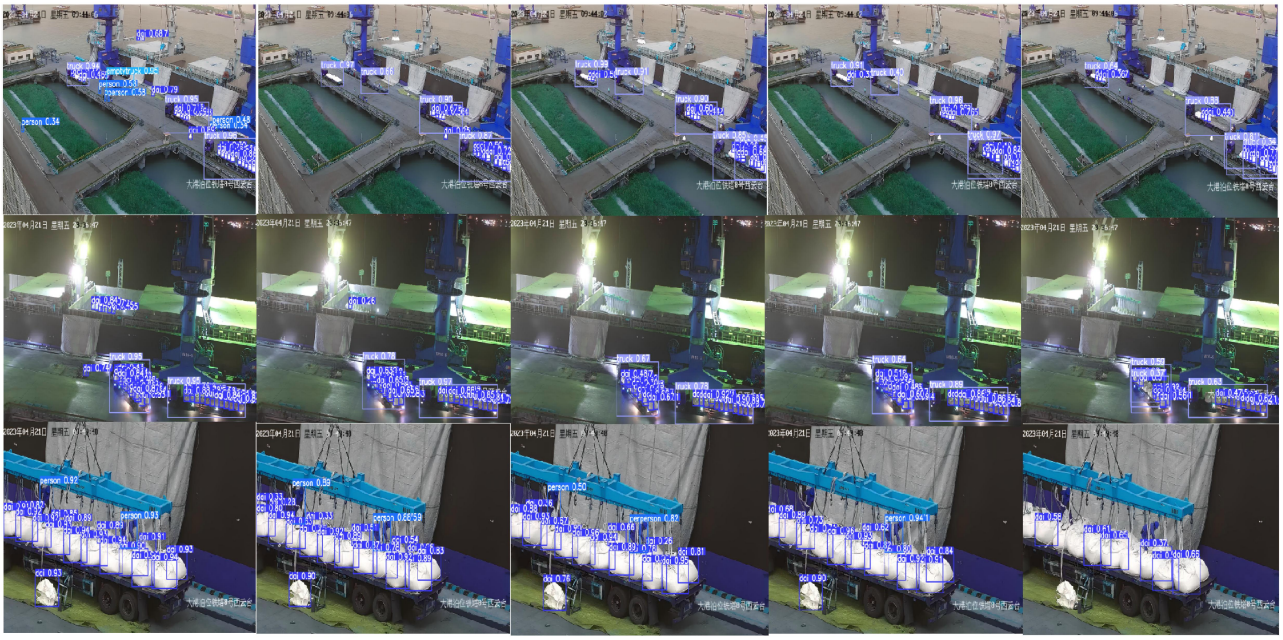


Figure 11. Comparison of the effect of the logistics of ton bag detection.

As can be seen from the detection effect comparison graph, YOLOv8-TB shows obvious advantages in dealing with complex scenarios such as daytime operation, nighttime operation, and ton bag contiguous occlusion situations.

In the daytime operation experiments of Group A, for the distant small targets ton bags, workers, etc. [33], YOLOv9, YOLOv8n, and YOLOv7-tiny have a certain amount of missed detections, and some of the detected targets are detected with low confidence of the detection frame, while YOLOv8-TB accurately identifies the targets that are missed by the other comparative models.

In Group B dark operation experiments, under the dark light environment, YOLOv9, YOLOv8n, and YOLOv7-tiny missed most of the ton bag targets, and some workers mistakenly detected ton bags, with a low confidence level of the target detection frame, as in Ref. [33]. In Group B dark operation experiments, under the dark light environment, YOLOv9, YOLOv8n, and YOLOv7-tiny showed most of the missed detection of the ton bag targets, and also some workers mistakenly detected them as ton bags, with a low confidence level of the target detection frame of the literature [33]. In contrast, YOLOv8-TB was able to accurately detect nearly all of the ton bag targets without false detection results, which greatly improves the ability to detect small targets under poor lighting conditions.

In the experiments of Group C ton bags with contiguous occlusion [33], YOLOv9, YOLOv8n, and YOLOv7-tiny can identify the ton bags with obvious contours in the front row, but almost completely miss the detection of ton bags with inconspicuous contours or mixed backgrounds, especially those rear ton bags that are heavily occluded, and the confidence level of the partially detected detection frames is not high. On the contrary, YOLOv8-TB detects ton bags with inconspicuous contours and conjoined body targets four times higher than YOLOv9, YOLOv8n, and YOLOv7-tiny [33], and detects ton bags with severely occluded rear rows six times higher than YOLOv9, YOLOv8n, and YOLOv7-tiny [33], which greatly improves the detection of severely occluded ton bags with inconspicuous contours and contiguous body targets, especially those severely occluded and inconspicuous contours of conjoined ton bags, while still maintaining a high confidence level.

From the detection comparison experiments, it can be seen that the improved algorithm YOLOv8-TB can detect targets that cannot be detected by other models, which proves that the algorithm in this paper can improve the problems of inaccurate positioning of small targets of ton bags and insufficient expression of target features during port operations.

5.6. Data Analysis

Figure 12 shows the P-R curve of YOLOv8-TB on the validation set. Precision is a measure of the relevance of the results, while recall is a measure of how many truly relevant results were returned. The average precision-recall on the validation set is 0.932, 0.995, 0.942, and 0.995 for ton bags, trucks, persons, and empty trucks, respectively, and 0.966 overall for all categories.

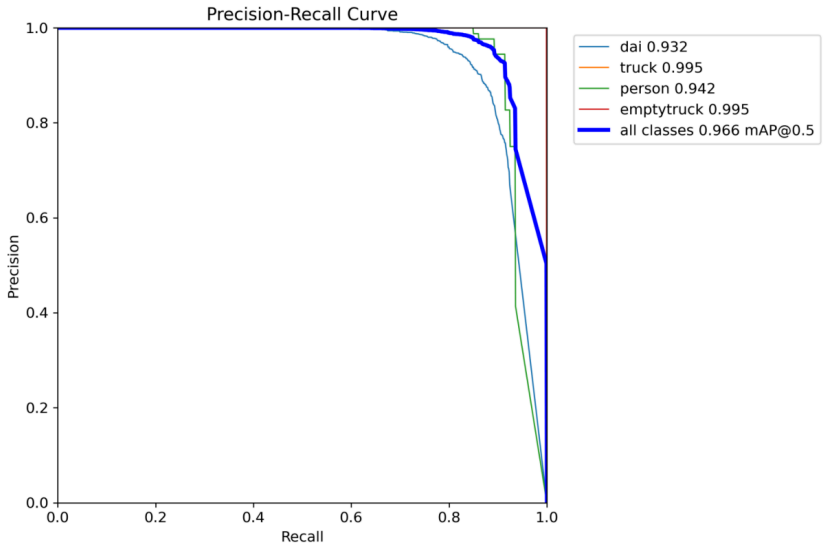


Figure 12. P-R curve.

Figure 13 illustrates the confusion matrix of YOLOv8-TB on the validation set. From the figure, it can be seen that the TP rates for these four categories are 0.87, 1, 0.89, and 1 for tonne bags, trucks, persons, and empty trucks, respectively.

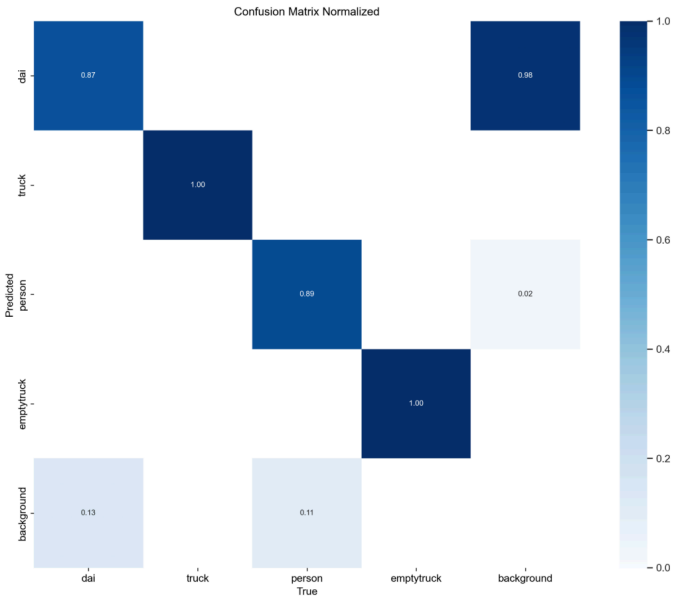


Figure 13. Confusion matrix.

After the data for validation, the performance of the model in this paper is affirmed, and the detection ability of the small target of the tonnage bag is effective.

In order to compare the detection effect of different improvement modules on different types of objects, Figure 14 shows the mAP for the four types of targets in the ShipPort Logistics ton bag operation dataset. According to the experimental results, it can be seen

that compared with YOLOv8n, the detection accuracy of the model with the addition of the four different improvement schemes alone has risen to a certain extent, among which the SPPFLKZ Attention Module and the Small Target Detection Layer have significantly increased the accuracy of the recognition of the ton bag's small targets. Using the C2f-SCTT and C2f-ORECZ modules alone, small target recognition accuracies for ton bags show a small improvement along with a significant reduction in model size. When these modules are used in conjunction with the small target detection layer, the recognition of small targets such as ton bags and workers shows a significant and stable improvement.

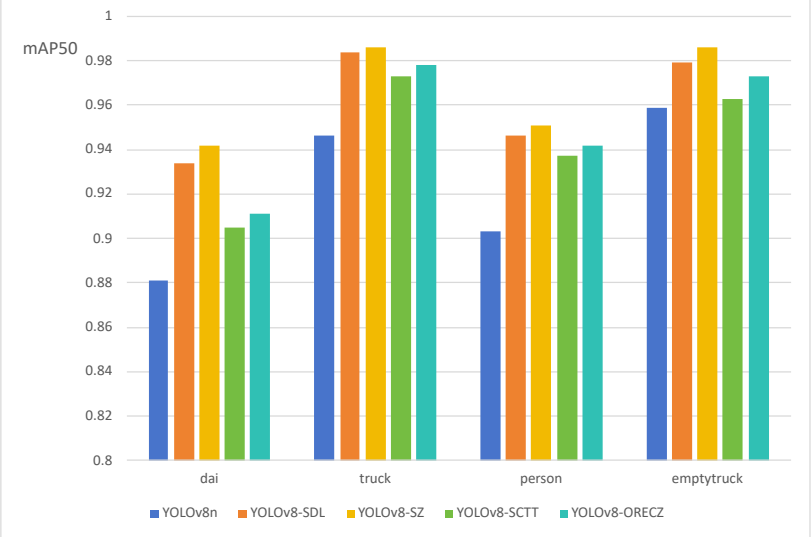


Figure 14. Visual comparison of detection accuracy.

Figures 15 and 16 demonstrate the overall situation of Recall and mAP50 after 400 rounds for YOLOv8n and YOLOv8-TB. It can be seen that the overall convergence of YOLOv8-TB is more stable than that of YOLOv8n, and at the same time, from the point of view of the exact value, YOLOv8-TB rises more rapidly.

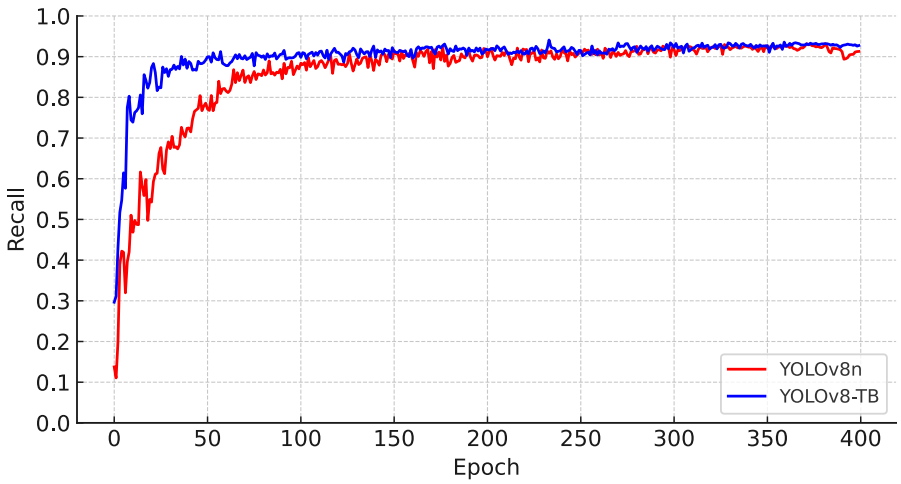


Figure 15. Recall trend.

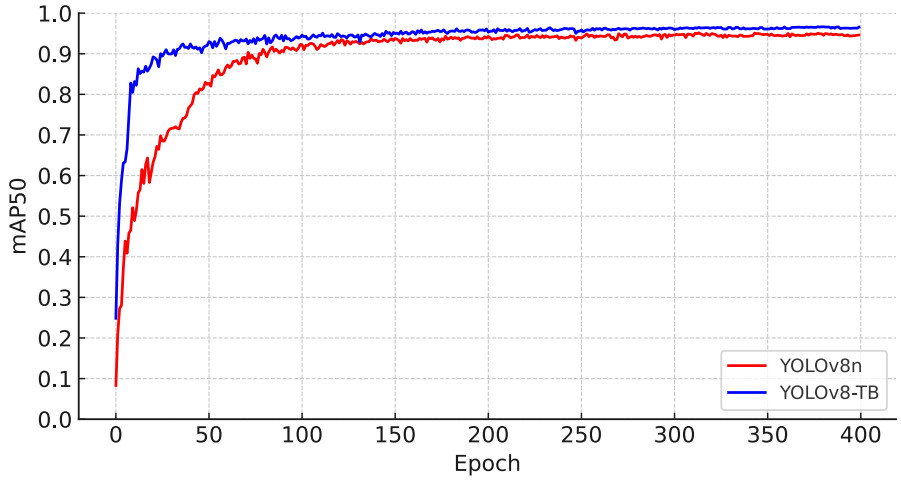


Figure 16. mAP50 trend.

Figures 17 and 18 show the loss function curves and accuracy curves of YOLOv8-TB on the training set and validation set, respectively. From the loss function curves, it can be seen that the loss curves of the training set and validation set almost decline synchronously and gradually stabilize, indicating that the model is well-fitted and has good generalization ability. From the accuracy curves, it is evident that the precision and confidence curves of the model for different categories ("dai", "truck", "person", "emptytruck") are very close, and there is no significant difference between the overall performance of the validation set and the training set. This demonstrates the consistency of the model during the training and validation phases, with no obvious signs of overfitting. The model shows high precision across different categories, and the confidence curves rise with increasing confidence and tend to stabilize, indicating that the model has strong generalization ability.

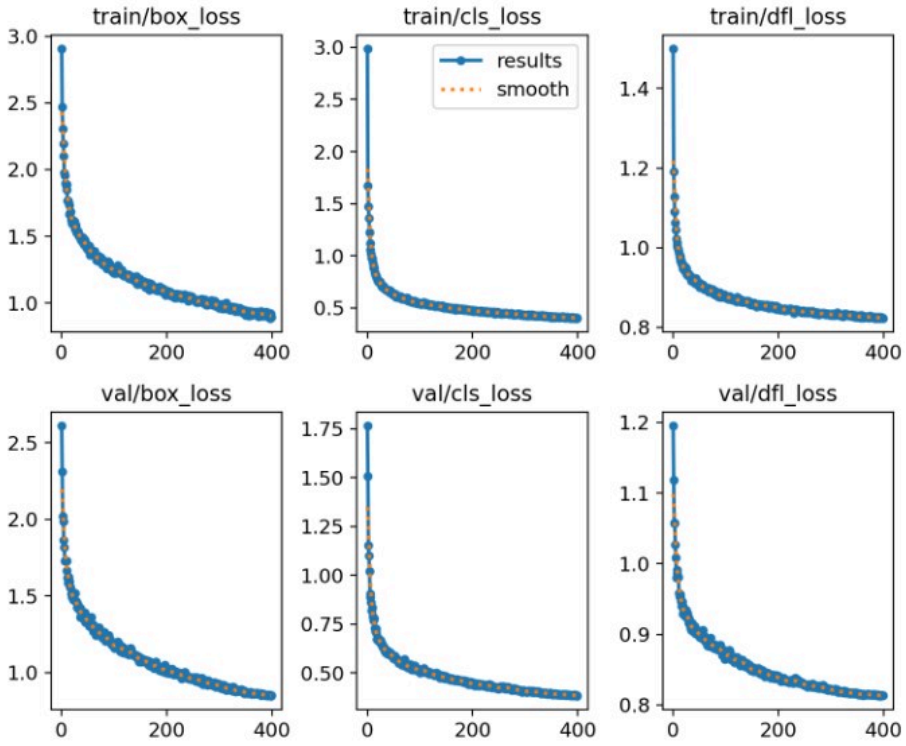
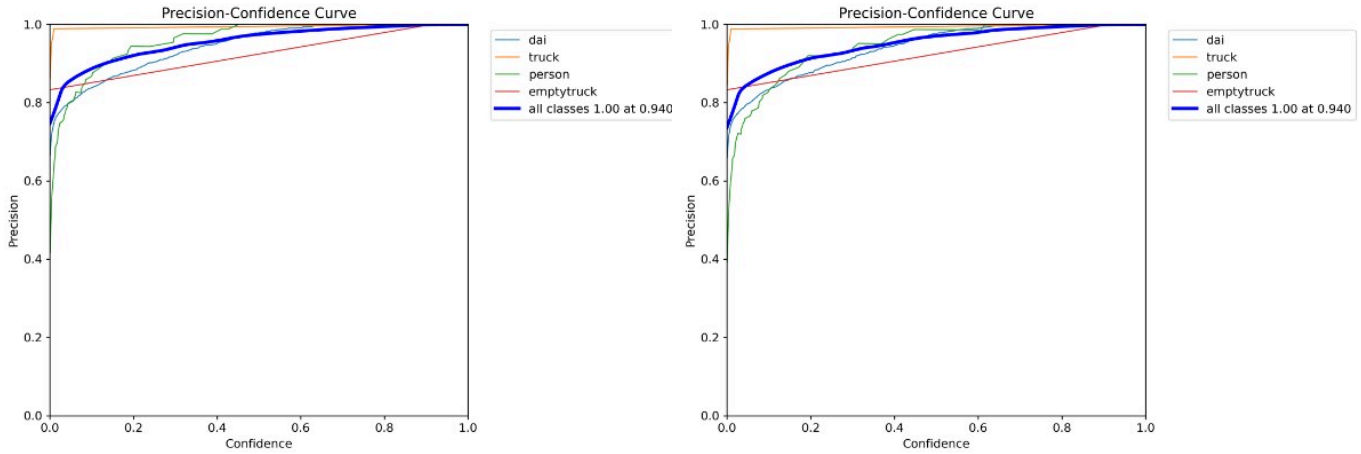


Figure 17. The loss function curves for the training set and validation set.



train val

Figure 18. The accuracy curves for the training set and validation set.

5.7. Performance Evaluation of the Training, Validation, and Test Sets

The performance metrics of YOLOv8-TB on the training, validation, and test sets are shown in Table 8. The precision and recall values on the three datasets are very close, indicating that the model can effectively distinguish between positive and negative samples, demonstrating strong generalization ability. The mAP50 on the training, validation, and test sets is almost identical, showing that the model maintains high object detection performance across different datasets when the IoU is set to 0.5. The mAP50-95, which is a stricter metric covering performance at various IoU thresholds, is also very close across the training, validation, and test sets. This further confirms that the model performs consistently across different datasets and possesses good generalization ability.

Table 8. The overall performance metrics of the partition set.

Set	Precision	Recall	mAP50	mAP50-95
Train	96.4	91.8	96.5	80.4
Val	95.7	92.2	96.4	81.1
Test	96.1	92.4	96.4	80.6

Figure 19 shows the detection performance metrics of YOLOv8-TB on the training, validation, and test sets for the four annotated categories. The Precision, Recall, mAP50, and mAP50-95 values across the training, validation, and test sets are very similar for each category, indicating that the model can accurately detect the targets without significant loss in performance. This demonstrates that the YOLOv8-TB model performs well in terms of generalization in this task, with no evident signs of overfitting.

Overall, the YOLOv8-TB model demonstrates consistent and high-level performance across the training, validation, and test sets, indicating no signs of overfitting and maintaining strong detection capabilities on unseen data. The metrics, including Precision, Recall, mAP50, and mAP50-95, suggest that the model possesses strong generalization ability, handling tasks across different datasets effectively. Therefore, YOLOv8-TB exhibits high reliability and practicality for real-world applications.

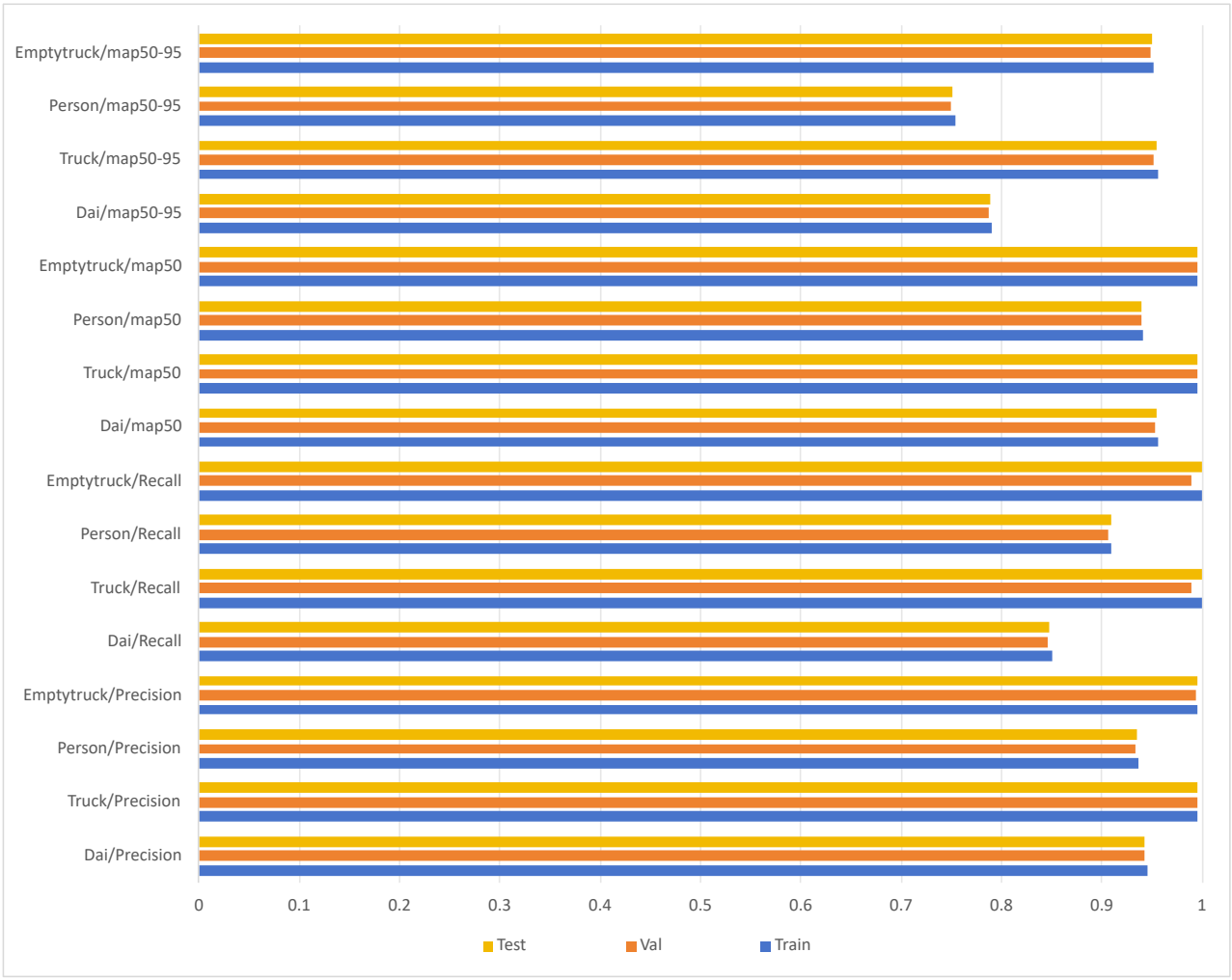


Figure 19. Class-wise detection performance metrics on training, validation, and test sets.

6. Conclusions

This paper proposes a ton bag target detection algorithm, YOLOv8-TB, based on improvements to YOLOv8. From a scientific perspective, YOLOv8-TB introduces an enhanced attention module, SPPFLKZ, which significantly improves feature extraction performance with only a minor increase in model parameters. Furthermore, the newly proposed C2f-SCTT and C2f-ORECZ modules not only achieve the goal of model lightweight but also enhance detection capabilities. The addition of a 160 × 160 small target detection layer improves the model's sensitivity to conjoined and occluded small targets in dense ton bag detection, solving the issues of misdetection and omission. This research paves the way for future innovations in real-time detection systems. From a practical perspective, YOLOv8-TB addresses the challenges of small target feature representation and the low accuracy of traditional port detection. In terms of societal impact, the accurate real-time detection provided by YOLOv8-TB helps reduce delays, prevent cargo accidents, and optimize labor utilization, ultimately lowering costs for shipping companies and consumers. Additionally, the adoption of AI-based detection systems in ports accelerates the advancement of smart port technologies, contributing to the sustainable development and innovation of the global logistics industry.

Through experimentation, YOLOv8-TB had the advantages of lower model size, less computation, and higher detection accuracy. On the dataset, mAP@50 and mAP@50-95 were improved by 3.7% and 5%, respectively, with a reduction of 4.42 MB in model size. YOLOv8-TB both improved the accuracy of the model and could be deployed and run smoothly on resource-constrained embedded inspection devices through its lightweight

design, which made YOLOv8-TB capable of accomplishing the task of being deployed in the field in port operation scenarios, and it was an effective and high-performance network model to deal with the problem of detecting ton bags in port logistics.

Author Contributions: X.Q., H.Z., C.Y., Q.L. and H.Y. performed the research; H.Y. designed the research study; X.Q. and H.Z. analyzed the data; and X.Q. and H.Z. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 52275251, the Six Talent Peaks project in Jiangsu under Grant XYDXX-117, the Key Research and Development Program of Zhenjiang under Grant GY2023049, and NDF under Grant JCKY2023***007.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used and analyzed during the current study are available from the corresponding author upon reasonable request. We have made the dataset public on the following website: https://github.com/zhzfighting/YOLOv8-TB-data (accessed on 14 September 2024).

Conflicts of Interest: The authors declare no competing interests.

Abbreviations

The following abbreviations are used in this manuscript:

TB	Ton Bag
YOLOv8	You Only Look Once version 8
SPPFLKZ	Modified SPPF module with Large Kernel Attention with Convolution
C2f-SCTT	Modified C2f block with DSRU and DCRU
DSRU	Deformable Spatial Reconstruction Unit
DCRU	Deformable Channel Reconstruction Unit
C2f-ORECZ	Modified C2f block with ORECZ
ORECZ	Online Convolutional Reparameterization Extended Block
C2f	Cross Stage Partial Bottleneck with 2 Conv layers and Feature Fusion
LZKAC	Large Kernel Attention with Convolution
SPPF	Spatial Pyramid Pooling Fast
LKA	Large Kernel Attention
LSKA	Large Separable Kernel Attention
MaxPool2d	Max Pooling 2D
SCConv	Split Convolution
GN	Group Normalization
DW	Deep Width
GWConv	Group Convolution
PWConv	Point-by-point Convolution
CNN	Convolutional neural network
BN	Batch Normalization
ReLU	Rectified Linear Unit
mAP	mean Average Precision

References

1. Argyriou, I.; Tsoutsos, T. Assessing Critical Entities: Risk Management for IoT Devices in Ports. *J. Mar. Sci. Eng.* **2024**, *12*, 1593. [CrossRef]
2. Li, J.; Xiao, R.; Zhao, Y. Docked ship detection based on edge line analysis and aggregation channel features. *Acta Opt. Sin.* **2019**, *39*, 0815004.
3. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
4. Liu, C.; Xiao, Y.; Yang, J.; Yin, J. Harbor detection in polarimetric sar images based on the characteristics of parallel curves. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1400–1404. [CrossRef]
5. Burns, J.B.; Hanson, A.R.; Riseman, E.M. Extracting straight lines. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 425–455. [CrossRef]

6. Jintao, Y.; Haitao, G.; Chuanguang, L.; Jun, L. Coast dock extraction method based on waterline and perceptual organization. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 6201–6204.
7. Bhagavathy, S.; Newsam, S.; Manjunath, B. Modeling object classes in aerial images using texture motifs. In Proceedings of the 2002 International Conference on Pattern Recognition, Quebec City, QC, Canada, 11–15 August 2002; Volume 2, pp. 981–984.
8. Bovolo, F.; Marin, C.; Bruzzone, L. A hierarchical approach to change detection in very high resolution SAR images for surveillance applications. *IEEE Trans. Geosci. Remote Sens.* **2012**, *51*, 2042–2054. [CrossRef]
9. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
10. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef] [PubMed]
12. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
13. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
15. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
16. 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; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
17. Yan, Y.; Xing, H. Small floating target detection method based on chaotic long short-term memory network. *J. Mar. Sci. Eng.* **2021**, *9*, 651. [CrossRef]
18. Zhang, F.; Zhang, W.; Cheng, C.; Hou, X.; Cao, C. Detection of small objects in side-scan sonar images using an enhanced YOLOv7-based approach. *J. Mar. Sci. Eng.* **2023**, *11*, 2155. [CrossRef]
19. Li, Z.; Ren, H.; Yang, X.; Wang, D.; Sun, J. LWS-YOLOv7: A Lightweight Water-Surface Object-Detection Model. *J. Mar. Sci. Eng.* **2024**, *12*, 861. [CrossRef]
20. Talaat, F.M.; ZainEldin, H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput. Appl.* **2023**, *35*, 20939–20954. [CrossRef]
21. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Wong, C.; Zeng, Y.; Montes, D.; et al. YOLOv5 by Ultralytics. 2023. Available online: https://github.com/ultralytics/yolov5 (accessed on 25 May 2024).
22. Guo, M.H.; Lu, C.Z.; Liu, Z.N.; Cheng, M.M.; Hu, S.M. Visual attention network. *Comput. Vis. Media* **2023**, *9*, 733–752. [CrossRef]
23. Lau, K.W.; Po, L.M.; Rehman, Y.A.U. Large separable kernel attention: Rethinking the large kernel attention design in cnn. *Expert Syst. Appl.* **2024**, *236*, 121352. [CrossRef]
24. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-attention with relative position representations. *arXiv* **2018**, arXiv:1803.02155.
25. Li, J.; Wen, Y.; He, L. Scconv: Spatial and channel reconstruction convolution for feature redundancy. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 6153–6162.
26. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
27. Hu, M.; Feng, J.; Hua, J.; Lai, B.; Huang, J.; Gong, X.; Hua, X.S. Online convolutional re-parameterization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 568–577.
28. Pan, X.; Ge, C.; Lu, R.; Song, S.; Chen, G.; Huang, Z.; Huang, G. On the integration of self-attention and convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 815–825.
29. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
30. Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; Liu, H. Expectation-maximization attention networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9167–9176.
31. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11534–11542.
32. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.

33. Lu, Y.; Lin, Y.; Wu, H.; Xian, X.; Shi, Y.; Lin, L. SIRST-5K: Exploring Massive Negatives Synthesis with Self-supervised Learning for Robust Infrared Small Target Detection. *IEEE Trans. Geosci. Remote Sens.* **2024**. [CrossRef]
34. Xu, X.; Sun, Z.; Wang, Z.; Liu, H.; Zhou, J.; Lu, J. DSPDet3D: Dynamic Spatial Pruning for 3D Small Object Detection. *arXiv* **2023**, arXiv:2305.03716.