Efficient and Accurate Beach Litter Detection Method Based on QSB-YOLO

Hanling Zhu[1, †], Daoheng Zhu[2, †], Xue Qin[3]*, Fawang Guo[4]*
College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China[1, 3]
College of Electronic and Information Engineering, Guangdong Ocean University, Zhanjiang 524088, China[2]
China Power Construction Group Guiyang Survey and Design Research Institute Co., Ltd., Guiyang 550081, China[4]

***Abstract*—Because of the potential threats it presents to marine ecosystems and human health, beach litter is becoming a major global environmental issue. The traditional manual sampling survey of beach litter is poor in real-time, poor in effect, and limited in the detection area, so it is extremely difficult to quickly clean up and recycle beach litter. Deep learning technology is quickly advancing, opening up a new method for monitoring beach litter. A QSB-YOLO beach litter detection approach based on the improved YOLOv7 is proposed for the problem of missed and false detection in beach litter detection. First, YOLOv7 is combined with the quantization-friendly Quantization-Aware RepVGG (QARepVGG) to reduce the model's parameters while maintaining its performance advantage. Secondly, A Simple, Parameter-Free Attention Module (SimAM) is used in YOLOv7 to enhance the feature extraction capacity of the network for the image region of interest. Finally, improving the original neck by combining the concept of the Bidirectional Feature Pyramid Network (BiFPN) allows the network to better learn features of various sizes. The test results on the self-built dataset demonstrate that: (1) QSB-YOLO has a good detection effect for six types of beach litter; (2) QSB-YOLO has a 5.8% higher mAP compared to YOLOv7, with a 43% faster detection speed, and QSB-YOLO has the highest detection accuracy for styrofoam, plastic products, and paper products; (3) QSB-YOLO has the greatest detection accuracy and detection efficiency when comparing the detection effects in various models. The results of the experiments demonstrate that the suggested model satisfies the need for beach litter identification in real-time.**

***Keywords*—Beach litter detection; QSB-YOLO; YOLOv7; Quantization-Aware RepVGG; a simple, parameter-free attention module; bidirectional feature pyramid network**

I. INTRODUCTION

Environmental pollution issues have gotten progressively worse in recent years, and beach litter pollution is unquestionably one of the planet's biggest environmental issues. Beach litter is characterized by wide distribution, persistence, and cumulative pollution, with two main sources, land-based and sea-based, including metal products, plastic products, wood, and paper products, etc [1]. The majority of beach litter is made up of materials that break down slowly, and because of poor waste management, it spreads out uncontrollably into the environment and interferes with marine traffic, damages ships, pollutes the nearshore area, degrades the environment, and results in many accidental injuries and deaths of marine life, etc [2]. Additionally, pollutants that persist and build up in beach litter can have an impact on people through the biological chain [2]. Today, China places a high value on reducing beach litter pollution, has created pertinent laws and regulations, and is actively doing research to improve the quality and cleanliness of beaches [3].

Many researchers have conducted a series of studies on the monitoring, sorting, and recycling of coastal litter. For example, Merlino et al. [4] evaluated the reliability of non-expert citizen scientist operators (CSO) to manually tag and classify marine litter from aerial photographs taken by drones. According to the study, CSO can support drone-based marine litter surveys with the right training programs and the provision of user-friendly guidance software, but the study is always labor-intensive [4]. Due to the labor-intensive nature of artificial beach litter sorting and cleanup, automated beach litter detection is a good solution to the pollution problem. The traditional method of beach litter monitoring is to monitor the amount of litter on the beach, which does not specifically localize or identify the beach litter [58]. The beach litter detection algorithm based on deep learning can simultaneously complete the classification and location of beach litter and feed it back to relevant staff, which can not only improve the detection accuracy of recyclable or non-degradable beach litter but also save a significant amount of manpower and material resources needed for beach litter cleaning and reduce environmental pollution to the greatest extent [910]. Deep learning-based beach litter detection is therefore very important.

In this research, we examined a few deep learning approaches that can be used to detect beach litter. However, the majority of deep learning-based object detection algorithms are designed for object detection in natural situations and do not completely apply to such unique scenarios as beach litter detection. This is due to the fact that the universal target detection model has a large number of parameters and a sluggish detection speed, making it unable to fulfill the speed requirements of actual applications and ineffective in dealing with issues like the mutual occlusion of targets and complicated beach litter backgrounds. Beach litter detection also be quite challenging due to the significant size differences between the targets of the litter.

In order to address the aforementioned issues, we suggest the QSB-YOLO beach litter detection method, which reduces model parameters to meet real-time requirements in practical applications, improves feature extraction capabilities of the

[†]These authors share first authorship

model to address the problem of missed and false detection in complex background, and enhances feature fusion network capabilities to enhance network's ability to acquire multi-scale features. The approach also offers improved detection speed and accuracy. The following are the primary contributions of this paper:

- Combining YOLOv7 with the quantization-friendly reparameterization architecture Quantization-Aware RepVGG (QARepVGG) [11] reduces the model parameters, and accelerates beach litter detection.
- Introducing A Simple, Parameter-Free Attention Module (SimAM) [12] attention mechanism in the YOLOv7, it enhances the features already acquired, gathers more useful information from images of beach litter, lessens the negative effects of complex backgrounds, focuses the model on beach litter objects, and reduces missed and false beach litter detection.
- Using the modified Bidirectional Feature Pyramid Network (BiFPN) [13] to improve the Path Aggregation networks (PAFPN) [14] structure of the original Neck; increases the model's capacity to extract features of various sizes and raises the model's accuracy in the identification of beach litter.

This paper adopts the following aspects to carry on the research. Section II explains the related research of object detection and beach litter detection. Section III introduces the suggested algorithm and discusses the reasons for and advantages of introducing three innovations. Section IV introduces the self-built dataset and discusses and analyzes the experimental results of the algorithm on the self-built dataset. Section V concludes and looks ahead to future work.

II. Related Works

A. Object Detection

Since 2012, deep learning has significantly advanced target detection technology. Deep learning-based target identification algorithms provide the advantages of high accuracy and resilience when compared to conventional target detection techniques. Depending on whether they must create candidate areas or not, deep learning-based target identification algorithms may be split into single-stage target detection algorithms and two-stage target detection algorithms [15].

R-CNN [16] marked the emergence of two-stage object detection algorithms, which first generate candidate regions and then put the candidate regions into the classifier to classify and correct the positions. Next, other two-stage object detection algorithms were put forth one after the other, including Fast Region-based Convolutional Network (Fast R-CNN) [17], Faster Region-based Convolutional Network (Faster R-CNN) [18], and more. However, due to its slow detection speed, two-stage object detection algorithms perform less well in practical applications.

The single-stage target detection algorithm is simple in structure, scalable, and more widely used, does not require candidate regions to generate branches, and detects the candidate frames and classes of targets directly at multiple locations in the image for a given input image. Its representative algorithms, such as the first You Only Look Once (YOLO) version of the target detection algorithm YOLOv1 proposed by Redmon et al. [19] in 2015, which completely discards the candidate region generation step and integrates classification, localization, and detection functions into one network, greatly improving the detection speed, but there are issues with missed and false detection and poor detection of small and multiple targets because of the simple network structure. The Single-shot multi-box detector (SSD) developed by Liu et al. [20] in 2015, which first introduced the idea of multi-scale detection, can improve the model's ability to detect small objects and be designed with a prediction module and a deconvolution module; however, the algorithm sacrifices a larger detection speed in exchange for a significant improvement in detection accuracy. Later, on the basis of YOLOv1, Redmon et al. [21][22] subsequently suggested YOLOv2 and YOLOv3 in 2017 and 2018, respectively. In order to address the issues of low recall and poor localization accuracy, YOLOv2 borrowed the Anchor mechanism of the Faster R-CNN algorithm, removed the fully connected layer from the YOLO network, and used the convolutional layer to predict the position offset of the detection frame and the category information [21]. In order to save computing effort, the YOLOv3 algorithm creates the DarNet-53 backbone network and makes use of only the 1×1 and 3×3 convolutional layers [22]. The YOLOv4 model, developed by Bochkovskiy et al. [23] in 2020, by fusing CSPDarknet53 and SPP to broaden the sensory field, and increases the detection accuracy of tiny objects. By expanding the effective aggregation network, adding the REP layer to facilitate new deployments, and adding Aux_detect for auxiliary identification, the YOLOv7 approach, which Wang et al. [24] introduced in 2022, obtains advantages in both speed and accuracy.

Although the single-stage technique identifies objects more rapidly than the two-stage method does, the detection results are still insufficient in detection with complex backdrops and enormous variations in object size. This study updates the YOLOv7 network using a method that boosts the network's feature extraction power to address these shortcomings and enhance the capacity of beach litter detection.

B. Beach Litter Detection

In 2011 Nakashima et al. [5] used balloon-assisted aerial photography combined with in situ measurements to estimate the amount of large litter on beaches; however, there is a large difference in litter density from place to place, so there is a large error in monitoring the amount of litter on beaches. Jang et al. [6] suggested employing a method that includes color and morphological image processing to compute the generation rate by applying thresholds to drone photographs in order to extract information about beach litter from the images. In 2012, kako et al. [7] built a low-altitude remote sensing system using a remotely operated digital camera suspended from a balloon, combining projection transformation methods and chromatic aberration in uniform color space (CIELUV) to process the resulting images to identify beach or marine litter. In 2018 Bao et al. [8] used remote sensing to apply a two-step threshold filtering method to images obtained by drones to identify and detect the distribution of beach litter. The aforementioned

study largely uses imaging techniques and projection transformation methods to monitor beach litter, however, it does not explicitly localize or identify beach litter, instead, it just monitors its amount.

Recently, researchers have started to carry out research and application practices for deep learning-based beach litter identification. In 2022, Rfeiffer et al. [9] investigated and compared the detection performance of two deep learning algorithms (YOLOv5 and Faster R-CNN) on images of beach litter taken by drones. The mAP value of YOLOv5 was 54.2% and the mAP of Faster R-CNN was 32.8%, the experimental findings demonstrate that the single-stage detection algorithm based on deep learning performs better for beach litter detection in terms of detection accuracy and detection speed. In a study of beach litter identification using YOLOv5 and camera acquisition picture data, Song et al. [10] achieved 87% detection accuracy, a notable increase over earlier research. In this investigation, over 90% of the dataset consisted of up-close images of beach litter collected at a height of 0.5 m. The beach litter objects in the close-up images were clear and there was mostly single-object beach litter in a single image, so a high detection accuracy was achieved; however, after using 1335 training photographs with complex backgrounds, the mAP of the plastic category significantly decreased from 88% to 26%, and there were some cases when shells were misclassified as plastic. Although there is a certain amount of misdetection, these experimental results show the immense potential and value of the YOLO model in beach litter detection application scenarios and demonstrate that the model's effectiveness in applications for identifying beach litter is significantly influenced by item morphology and characteristics, hyperparameter settings, and training data.

The analysis above shows that, even though YOLOv7's accuracy and efficiency have greatly improved over the previous YOLO model, more focused optimization is still required to increase the performance and suitability of beach litter detection.

First, it must be addressed that the number of model parameters and detection speed do not meet the needs of the actual applications. Wang et al. [25] incorporated GhostNet into the YOLOv5 model to improve its detection effectiveness. As a consequence, the number of parameters was successfully reduced by 47% and the computational complexity by 49.4%, but the detection accuracy fell by 2.2% compared to the original YOLOv5 model. By pruning the filters corresponding to the low-importance channels of the model to make it simpler to deploy the model to devices for real-time intelligent pedestrian monitoring, Xu et al. [26] obtained CAP-YOLO, which is three times faster inference than the original YOLOv3, but with a 7% reduction in mAP. Many current lightweight network models focus on compromising detection accuracy to boost detection speed, but beach litter detection demands high accuracy, hence this research introduces QARepVGG in YOLOv7 to enhance detection speed while enhancing the accuracy of beach litter detection.

Secondly, the beach litter dataset contains a complex background consisting of leaves, branches, and other kinds of interfering objects, with interfering objects obscuring the target, overlapping between the target and the target, and the problem of fuzzy targets exists. The YOLOv7 algorithm still encounters missed and incorrect detection in such intricate backgrounds. The attention mechanism is often used for complex backgrounds. The Coordinate Attention (CA) mechanism that TCA-YOLO [27] introduced in YOLOv5 to weaken the interference of complex backgrounds has greatly improved the detection accuracy of small targets, but the detection effect of larger targets still requires improvement.

Additionally, in the actual application environment for beach litter detection, the size and form of the litter also vary, and there are many tiny and medium-sized targets. The great variety in target sizes makes detection extremely challenging. Li et al. [28] proposed adding jump connection and multi-structure multi-size feature fusion in feature extraction and feature fusion to improve the detection accuracy but slow down the detection speed.

For efficient and accurate beach litter detection, it is not advisable to sacrifice the detection speed or accuracy. Therefore, considering the practical application scenarios of beach litter detection, this paper optimizes the YOLOv7 model according to the difficulties in beach litter detection and proposes QSB-YOLO for beach litter detection, and tests the optimized QSB-YOLO. The experimental results show that QSB-YOLO not only reduces the number of parameters of the model and improves the detection speed of the model, but also improves the detection accuracy of beach litter.

III. Materials and Methods

Considering the three problems in beach litter detection, we propose the beach litter detection method QSB-YOLO, whose overall network structure is shown in Fig. 1. Different from the original YOLOv7, QSB-YOLO replaces the first E-ELAN module of the YOLOv7 backbone network and the E-ELAN module in the neck with the quantization-friendly QARepVGG, then replaces the MP in the neck with the improved S-MP, and finally combines the idea of BiFPN to improve the original neck's PAFPN.

The beach litter image dataset is input into QSB-YOLO, and the image size is adjusted to 640×640 through input. The adjusted image is input into the backbone for feature extraction, and the obtained effective feature layers C3, C4, and C5 are input into the neck to strengthen the feature extraction of the network. The three outputs P3, P4, and P5 obtained from the neck are then sent to YOLOhead to predict the anchor frame, confidence, and category of beach litter after RepCov adjusts the number of channels.

A. Combine the Quantization-friendly QARepVGG

The approach of model compression known as quantization, which successfully reduces a model's number of parameters but also lowers the model's performance, is frequently overlooked in deep neural networks. The reparameterized architecture-based multi-branch design widens the dynamic numerical range, which creates an issue with the difficulty of quantization. But the reparameterization architecture QARepVGG we introduced is simple and efficient.

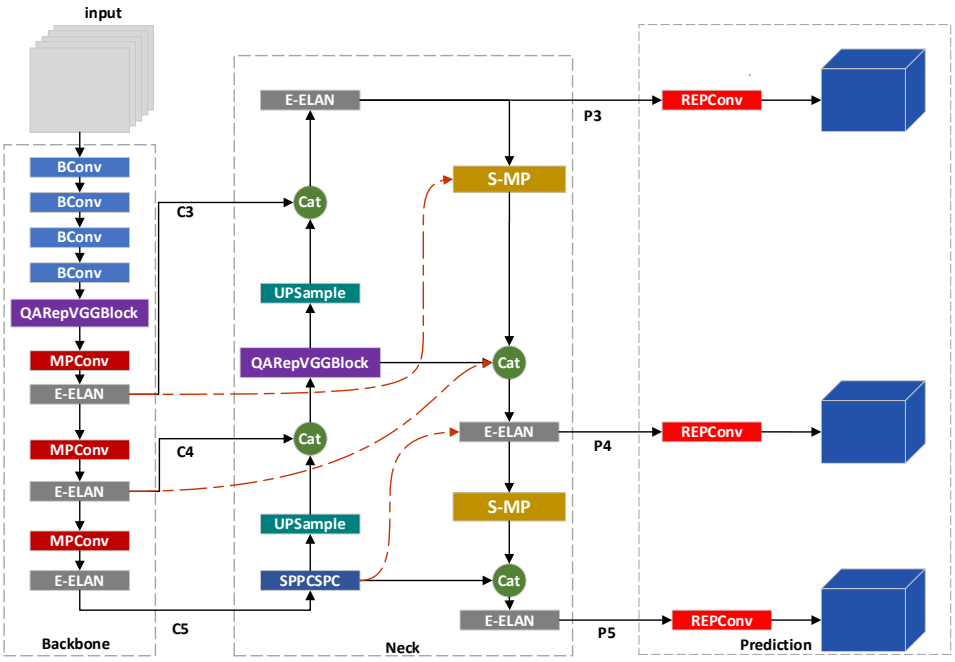


Fig. 1. The network structure of QSB-YOLO.

The custom weight decay design in RepVGG [29] is successful in building a model with a stronger weight distribution, but at the same time the variance of the activation distribution is amplified, the input of the subsequent layer depends on the activation, and the standard deviation rises layer by layer as the depth of the network layer increases, leading to an accuracy decrease. QARepVGG is improved based on RepVGG to solve the problem of quantization failure of RepVGG multi-branch design, which not only reduces the number of parameters of the model but also improves the detection accuracy and detection speed of the model [11]. According to the quantization-friendly features of QARepVGG, in order to improve the efficiency of beach litter detection, we combine QARepVGG in YOLOv7 and prove its effectiveness according to the experiment.

In order to measure quantification error, QARepVGG introduces mean square error (MSE), as in Equation (1).

$$MSE(Q(w,t,n_b),w)=\frac{1}{n}\sum_i (Q(w_i,t,n_b)-w)^2 \quad (1)$$

where Q represents the quantization process, $w \in R^n$ represents the nth channel of the weights, t represents the stage threshold, and n_b represents the number of quantization bits. The output $M_{(2)}$ is:

$$M_{(2)}=BN(3\times3)+BN(1\times1)+BN(Identity) \quad (2)$$

$BN(3\times3)$ is:

$$Y_{(3)}=\gamma_{(3)}\frac{M_{(1)}*W_{(3)}-\mu_{(3)}}{\sqrt{\varepsilon+\sigma_{(3)}}\sigma_{(3)}}+\beta_{(3)} \quad (3)$$

$Y_{(3)}$ is the output of the 3×3 branch, $\gamma_{(3)}$ is the scale factor of the BN layer after 3×3 convolution, $\mu_{(3)}$ is the mean value of the BN layer after 3×3 convolution, $\sigma_{(3)}$ is the standard deviation of the BN layer after 3×3 convolution, $\beta_{(3)}$ is the deviation of the BN layer after 3×3 convolution. $M_{(1)}$ is input, $W_{(2)}$ is 3×3 convolution kernel, $*$ is multiplication, ε is the value that ensures numerical stability (The default value is 10^{-5}). This indicates that BN has the effect of stabilizing the variation of the input.

Introduce random variables X and scalars λ, $D(\lambda X)=\lambda^2 D(X)$, and make $X_{(3)}=M^1 W_{(3)}$, then we have

$$D(Y_{(3)})=\frac{\gamma_{(3)}\gamma_{(3)}}{\varepsilon+\sigma_{(3)}\sigma_{(3)}}D(X_{(3)}) \quad (4)$$

The variation size of $X_{(3)}$ is controlled by $\frac{\gamma_{(3)}\gamma_{(3)}}{\varepsilon+\sigma_{(3)}\sigma_{(3)}}$.

The reparameterization-based architecture needs to quantify the weight distribution and activation distribution. Friendly quantization refers to having a relatively narrow range of values and a narrow distribution of standard deviations; when one of these features is missing; the standard deviation is amplified, which reduces the model's accuracy [11]. The RepVGG custom weight decay is as in Equation (5).

$$L_{2_{custom}}=\frac{\left|W_{wq}\right|_2^2}{\left|\frac{\gamma_{(3)}}{\sqrt{\varepsilon+\sigma_{(3)}}\sigma_{(3)}}\right|_2^2+\left|\frac{\gamma_{(1)}}{\sqrt{\varepsilon+\sigma_{(1)}}\sigma_{(1)}}\right|_2^2} \quad (5)$$

RepVGG reduces the weight loss by enlarging the denominator, but enlarges the standard deviation distribution and amplifies the activation distribution deviation. Based on this, QARepVGG removes the BN in the identity branch and replaces the custom weight decay design in RepVGG with the standard weight decay design (normal L2). As a result, the module successfully completes weight quantization. The output is then rewritten as

$$M_{(2)} = BN(3\times3) + BN(1\times1) + Identity \quad (6)$$

Let the anticipated values of 3×3 branches and 1×1 branches be

$$E\left(Y_{(3)}\right) = \beta_{(3)}, E\left(Y_{(1)}\right) = \beta_{(1)} \quad (7)$$

The variance may grow if $\beta_{(3)} = \beta_{(1)} = \beta$, at this point $\mu = 0$, $\sigma = 0$, $\gamma = 1$, $\beta = 0$, then $Y_{(3)}$ and $Y_{(1)}$ are in balance. To better stabilize the variance, the BN in the 1×1 branch is further removed, at which time the output is

$$M_{(2)} = BN(3\times3) + (1\times1) + Identity \quad (8)$$

BN has the effect of stabilizing the input variance, and to stabilize the training process, QARepVGG adds batch normalization after three branches. At this stage, the output is

$$M_{(2)} = BN(BN(3\times3) + (1\times1) + Identity) \quad (9)$$

As illustrated in Fig. 2, the QARepVGG first converts each of the three branches into a single 3×3 convolution, then combines the three into one, and lastly adds the 3×3 convolution with BN to get the final 3×3 convolution. A multi-branch structure is used during training to improve the model's detection accuracy and the network's capacity for representation. To speed up the model's detection, single-branch inference is employed in the inference process.

Fig. 3(b) depicts the topology of QARepVGG in the YOLOv7+QARepVGG backbone network during training, and Fig. 3(a) depicts the E-ELAN module it replaces; the structure of QARepVGG in the Neck of YOLOv7+QARepVGG is depicted in Fig. 3(d), and Fig. 3(c) is its replacement E-ELAN module. When there is no BN versus with BN, the number of parameters for convolution are calculated as shown in Equations (10) and (11), respectively.

$$params = C_o \times C_{in} \times k_h \times k_w + C_o \quad (10)$$

$$params = C_o \times C_{in} \times k_h \times k_w + C_o \times 2 \quad (11)$$

where, C_o represents the number of output channels, C_{in} represents the number of input channels, k_h denotes the height of the convolution kernel, and k_w represents the width of the convolution kernel.

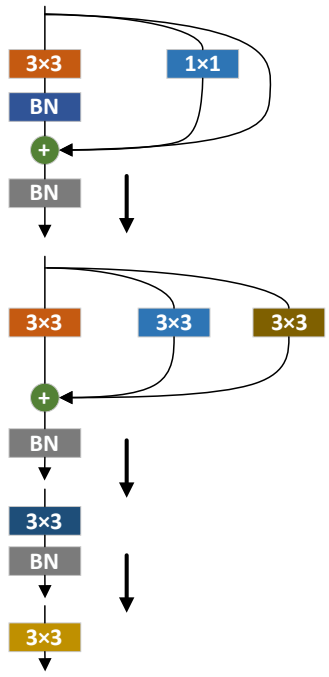
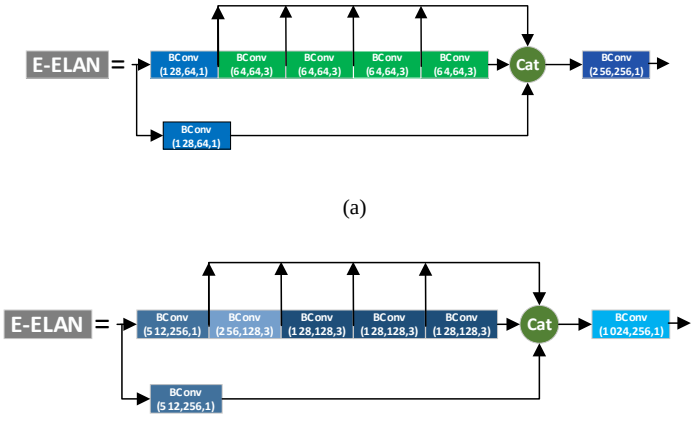
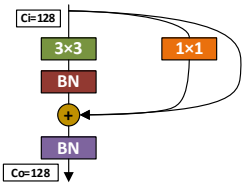


Fig. 2. Schematic diagram of QARepVGG.

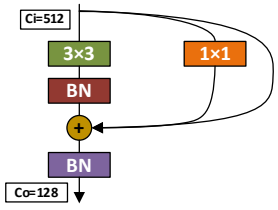


(a)



(b)

(c)



(d)

Fig. 3. Schematic diagram of QARepVGG. Comparison of modules before and after improvement. (a) The E-ELAN that was replaced in the backbone, (b) QARepVGG of the backbone during training, (c) The E-ELAN that was replaced in the neck, (d) QARepVGG of the neck during training.

TABLE I. COMPARISON OF THE NUMBER OF PARAMETERS

YOLOv7		YOLOv7+QARepVGG	
The sum of all parameters to be changed	value	The sum of all parameters to be changed	value
$params_1 + params_3 + params_5 \times 3 + params_7 \times 2$	1857024	$params_2 + params_4 + params_6 \times 3 + params_8 \times 2$	1067776

The number of parameters for the E-ELAM of the backbone network replaced by QARepVGG in the YOLOv7 is denoted by $params_1$, where Cat has no parameters. In YOLOv7+ QARepVGG, the number of parameters for the QARepVGG in the backbone network is denoted as $params_2$, where the 3×3 convolution has BN operation in QARepVGG, the identity branch has no parameters, and a new BN layer is added at the end. Similarly, the number of parameters for the E-ELAM of the Neck replaced by QARepVGG in the YOLOv7 is denoted as $params_3$, and the number of parameters for the QARepVGG in the Neck of YOLOv7+QARepVGG is denoted as $params_4$.

In addition to the replaced modules, the parameters for the remaining parts of the convolution were also affected. There are three convolutions that are $C_{in}=256, C_o=128, k_h=k_w=1$ in YOLOv7 and become $C_{in}=128, C_o=128, k_h=k_w=1$ in YOLOv7 +QARepVGG, and two convolutions in YOLOv7 that are $C_{in}=512, C_o=256, k_h=k_w=1$ become $C_{in}=384, C_o=256, k_h=k_w=1$ in YOLOv7+QARepVGG, the number of parameters for they are denoted as $params_5$, $params_6$, $params_7$, $params_8$, respectively.

The total number of parameters of the affected modules is shown in Table I. It can be inferred from Table I that after the introduction of QARepVGG, the number of parameters decreased by 789248, or about 2.1%.

B. S-MP

Through the analysis of the self-built beach litter dataset in this paper, we found that there exists a complex background composed of leaves, branches, and other kinds of distractors, the distractors obscure the target, the target overlaps with the target, and there is the problem of the target blurring, which easily causes missed and false detection. To improve this type of situation, this paper proposes to introduce an attention mechanism to attenuate the negative effects of complex backgrounds.

Traditional attention mechanisms, which produce one- or two-dimensional weights along the channel dimension or spatial dimension, focused only on this channel or this space, limit the flexibility of learning attention weights to alter throughout space and channel. Fig. 4 displays the SimAM schematic diagram. In contrast to the traditional spatial attention mechanism and channel attention mechanism, SimAM attention is a three-dimensional attention mechanism that assigns a unique weight to each neuron in space or channel without the use of additional parameters, determines the importance of each neuron, and then enhances the features using the three-dimensional weights. First, SimAM defines the energy function for each neuron as shown in Equation (12).

$$e_t\left(w_t, b_t, y, x_i\right) = \left(y_i - \hat{t}\right)^2 + \frac{1}{M-1}\sum_{i=1}^{M-1}\left(y_0 - \hat{x}_i\right)^2 \quad (12)$$

t represents the target neuron, $\hat{t} = w_t t + b_t$ represents a linear transformation of t. x_i indicates other neurons in the same channel as the input feature $X \in \mathbb{R}^{C\times H\times W}$, $\hat{x}_i = w_t x_i + b_t$ is a linear transformation of x_i. i is the index in the spatial dimension, $M = H\times W$ is the number of neurons on the channel, w_t and b_t are respectively the weights and biases of the transformation. When Equation (12) reaches a minimum, the target neuron finds linear differentiability with other neurons in the same channel, at which time $\hat{t} = y_t$, and arbitrary $\hat{x}_i = y_0$. Binary labeling of y_t and y_0, and adding regularization to Equation (12) yields Equation (13).

$$e_t\left(w_t, b_t, y, x_i\right) = \frac{1}{M-1}\sum_{i=1}^{M-1}\left(-1-\left(w_t x_i + b_t\right)\right)^2 + \left(1-\left(w_t t + b_t\right)\right)^2 + \lambda\omega_t^2 \quad (13)$$

$$w_t = -\frac{2\left(t-\mu_t\right)}{\left(t-\mu_t\right)^2 + 2\sigma_t^2 + 2\lambda} \quad (14)$$

$$b_t = -\frac{1}{2}\left(t+\mu_t\right)w_t \quad (15)$$

$\mu_t = \frac{1}{M-1}\sum_{i=1}^{M-1} x_i$ and $\sigma_t^2 = \frac{1}{M-1}\sum_{i=1}^{M-1}\left(x_i-\mu_t\right)^2$ denote the mean and variance of all neurons except for that channel, respectively. The minimum energy is calculated as in Equation (16).

$$e_t^* = \frac{4\left(\hat{\sigma}^2+\lambda\right)}{\left(t-\hat{\mu}\right)^2 + 2\hat{\sigma}^2 + 2\lambda} \quad (16)$$

$$\hat{\mu} = \frac{1}{M}\sum_{i=1}^{M} x_i \quad (17)$$

$$\hat{\sigma}^2 = \frac{1}{M}\sum_{i=1}^{M}\left(x_i-\hat{\mu}\right)^2 \quad (18)$$

$\frac{1}{e_t^*}$ indicates the importance of each neuron. That is the smaller the value of e_t^*, the more the target neuron is distinguished from other neurons and the greater the importance. Subsequently, feature enhancement is performed according to the definition of attention mechanism as in Equation (19).

$$\tilde{X} = sigmoid(\frac{1}{E})\square\ X \quad (19)$$

To reduce missed and false detection in beach litter detection and enable the model to obtain more useful features without increasing the model's parameters, we introduce the SimAM attention mechanism in the MP module in the neck of YOLOv7.

The MP in Neck is down-sampled in YOLOv7 using both Maxpool and BConv, and the features from each are then combined. As shown in Fig. 5, the SimAM attention mechanism is introduced to replace BConv in the second branch of the MP structure. the input of S-MP is down-sampled by Maxpool in the first branch, and then the number of channels is adjusted by BConv. In the second branch, the features are enhanced by the SimAM attention mechanism while adjusting the number of channels and then down-sampling by BConv. Finally, the features obtained from the two branches are feature fused to obtain the results of the enhanced down-sampling and obtain more effective features.

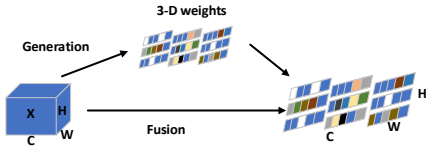


Fig. 4. Schematic diagram of SimAM.

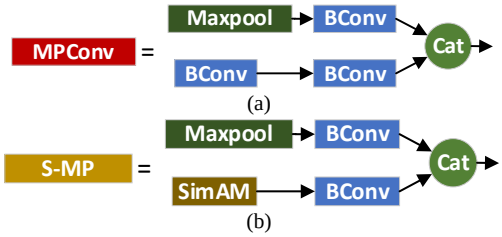


Fig. 5. Structure diagram of MP. (a) MP, (b) S-MP.

C. Improved-BiFPN

BiFPN is a feature fusion technique that combines weighted feature fusion with an efficient bidirectional cross-scale connection. The traditional Feature Pyramid Network (FPN) [30] fuses multi-scale features in a top-down manner, and the down-sampling process loses feature information at the highest level, reducing the multi-scale representation capability. A top-down, bottom-up path aggregation network is added by the PAFPN in the YOLOv7 network design. As shown in Fig. 6(a), in order to fuse more features without incurring excessive costs, the idea behind BiFPN is to add an extra edge to the original input and output nodes at the same layer; additionally, each bidirectional path is treated as a layer of the feature network, and the same layer is repeated multiple times to achieve a higher level of feature fusion.

We combine the idea of BiFPN and modify the PAFPN in the YOLOv7 network into the Improved-BiFPN to realize the bi-directional fusion of different network feature layers and enhance the information transfer between different network feature layers. As shown in Fig. 6(b), the three effective feature layers extracted from the backbone network are C3, C4, and C5, which are passed into the Improved-BiFPN to achieve effective bidirectional cross-scale connectivity and weighted feature fusion, resulting in a total of three different scales of P3, P4, and P5 output features.

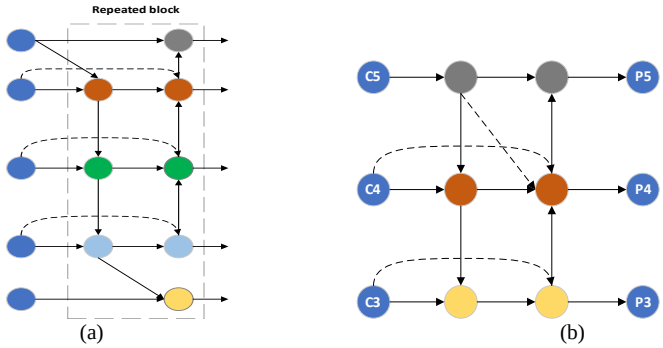


Fig. 6. Structure diagram of BiFPN. (a) The original BiFPN, (b) Improved-BiFPN.

IV. Results and Discussion

A. Dataset Acquisition and Pre-processing

The beach litter image collection used in this paper was collected from several popular tourist beaches along the coast of South China. A total of 1587 beach litter photographs were captured by cameras during the beach survey, with resolutions ranging from 4000×2250 pixels to 480×360 pixels. To avoid the issue of overfitting brought on by insufficient training samples, data enhancement techniques such as rotation, color enhancement, and contrast enhancement were applied to each image separately, as shown in Fig. 7. After data augmentation, there are 6348 pictures overall, of which 90% are separated into training set and 10% into test set.

The self-built beach litter dataset collected a total of six categories of litter, including plastic products, metal products, paper products, wood, styrofoam, and glasswork, and the examples of each category are shown in Fig. 8. The beach litter image dataset used in this study was manually annotated using LableImg annotation software and saved in YOLO format.

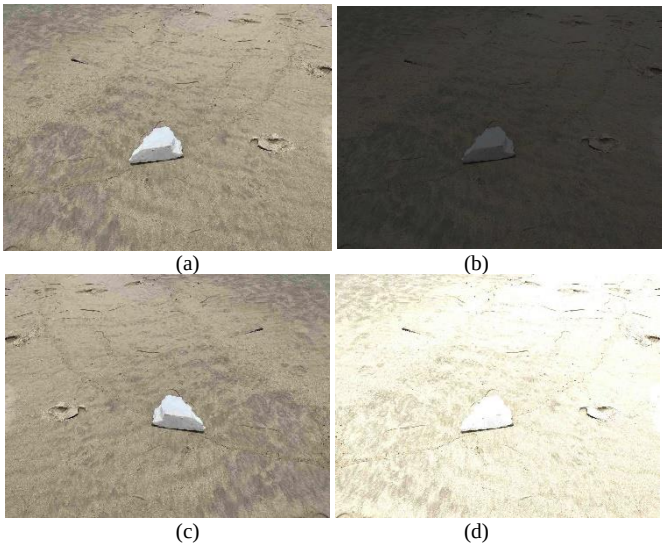


Fig. 7. Data enhancement. (a) Original image, (b) Color enhancement, (c) Rotation, (d) Contrast enhancement.

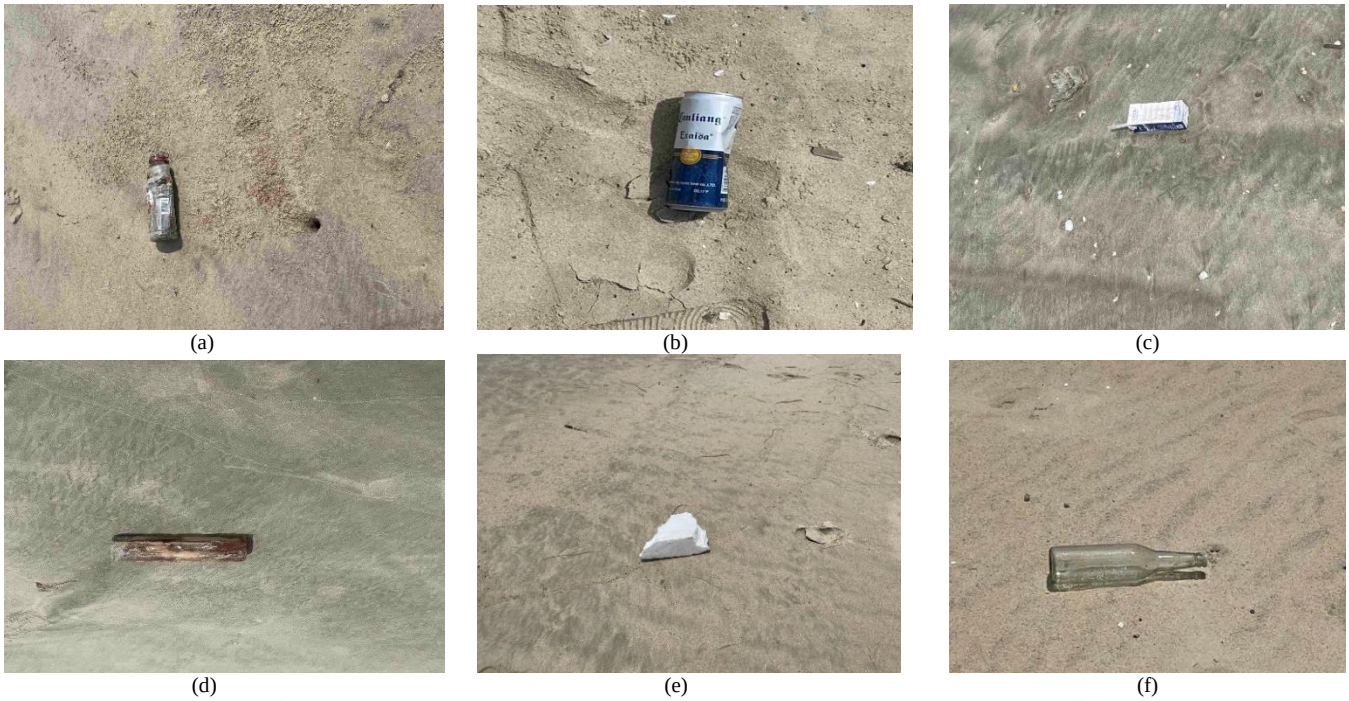


Fig. 8. Categories of beach litter. (a) Plastic products, (b) Metal products, (c) Paper products, (d) Wood, (e) Styrofoam, (f) Glasswork.

B. Experimental Parameter Settings

In this paper, there are 6348 beach litter image dataset, 5714 images are used as the training set, and 634 images as the test set. The experimental configuration is shown in Table II. The operating system used for training is Ubuntu 18.04.1, the GPU is NVIDIA GeForce RTX 3060 with 12G of video memory, the CPU is 11th Gen Intel(R) Core (TM) i5-11600KF, and the memory is 16G. The batch size is 8, the initial learning rate is 0.1, and 300 Epochs are trained.

TABLE II. HARDWARE EQUIPMENT AND DEVELOPMENT ENVIRONMENT

Equipment	Model
GPU	NVIDIA GeForce RTX 3060
CPU	11th Gen Intel(R) Core (TM) i5-11600KF
Ubuntu	18.04
CUDA	10.2
Python	3.7
Pytorch	1.8
Torchvison	0.7.0

C. Evaluation Indicators

This study evaluates the model using Average Precision (AP), Mean Average Precision (mAP), Parameters (Params), Frames Per Second (FPS), and F1. The value of AP, which measures the typical detection accuracy of a single category of beach litter, is the area of the P-R curve. The accuracy rate and recall rate are the P-R curve's vertical and horizontal coordinates, respectively. Equations (20) and (21), respectively, are used to compute the precision rate, P, and recall rate, R. Equation (22) is used to compute the value of the AP. Equation (23) is used to construct mAP, which is a measure of the average detection accuracy across all categories. The Params is used to measure the size of the model. FPS is a unit used to express how quickly a model can identify an object. The fraction of recalled genuine positive categories is measured by F1, which is determined using Equation (24).

Where TP denotes that the actual result is identical to the expected result, FN denotes that the category of beach litter is judged to be another category of beach litter or to be missed detection, and FP denotes that the non-beach litter target is detected as beach litter.

$$P = \frac{TP}{TP + FP} \tag{20}$$

$$R = \frac{TP}{TP + FN} \tag{21}$$

$$AP = \int_0^1 P(R)dR \tag{22}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{23}$$

$$F1 = \frac{2PR}{(P + R)} \tag{24}$$

D. Analysis of Experimental Results

1) Comparison Experiments Combining QARepVGG: To verify the effectiveness of the quantization-friendly QARepVGG method combined in this paper, RepVGG will be introduced in YOLOv7 to conduct comparative experiments with YOLOv7+ QARepVGG on our beach litter dataset, and the experimental results are shown in Table III.

TABLE III. COMPARATIVE EXPERIMENTS COMBINED WITH QAREPVGG

Model	mAP@0.5	Params	Inference times	FPS(f/s)
YOLOv7	79	37223526	11.8ms	40
YOLOv7+RepVGG	77.7	36434534	8.4ms	56
YOLOv7+QARepVGG	**80.7**	**36434278**	**8.0ms**	**58**

The reparameterized architecture of RepVGG's 1×1 branch at training time than QARepVGG's 1×1 branch has one more BN operation, so the overall number of YOLOv7+ RepVGG p arameters is 256 more than YOLOv7+ QARepVGG. The infer ence time of YOLOv7+RepVGG is 3.4ms faster than YOLOv7 , but it is 0.4ms slower than YOLOv7+QARepVGG. The detec tion speed of YOLOv7+RepVGG is 16 f/s faster than YOLOv7 , but 2 f/s slower than YOLOv7+QARepVGG.

RepVGG reduces the inference time and improves the detection speed significantly, but sacrifices the detection accuracy. While reducing the number of parameters in the model and increasing the detection accuracy of beach litter by 1.7%, QARepVGG increases inference speed and detection speed. Therefore, in this paper, we choose to introduce the quantization-friendly QARepVGG for beach litter detection.

2) Comparison experiments with attention mechanisms: In this paper, we combine the Convolutional Block Attention Module (CBAM) [31] and Efficient Channel Attention Module (ECA) [32] with the MP module of Neck in YOLOv7 and compare them with YOLOv7+S-MP for comparison, and the experimental results are shown in Table IV. Where C-MP denotes the use of the CBAM attention mechanism in place of the first convolution of the original Neck's second branch of MP. E-MP denotes that the first convolution of the second branch of MP in the original Neck is swapped out for the ECA attention mechanism.

CBAM is a hybrid attention mechanism, YOLOv7+C-MP detection accuracy is reduced by 3.2% compared to YOLOv7, F1 is reduced by 0.03, and the number of parameters is reduced less. Both ECA and SimAM are parameter-free attention mechanisms, and when the same convolution is replaced, YOLOv7+ E-MP has the same number of parameters as YOLOv7+S-MP. However, when ECA is added, mAP@0.5 is decreased by 10.5%. Beach litter detection on complicated backdrops is not appropriate for either CBAM or ECA, which both decrease the mAP@0.5 of beach litter. While adding the SimAM attention mechanism the mAP@0.5 is increased by 3.9%, F1 by 0.04, and the parameter-free attention mechanism does not negatively affect the number of model parameters.

TABLE IV. COMPARATIVE EXPERIMENTS OF ATTENTION MECHANISMS

Model	mAP@0.5	F1	Params
YOLOv7	79	0.73	37223526
YOLOv7+C-MP	74.8	0.70	37151684
YOLOv7+E-MP	68.5	0.65	**37140838**
YOLOv7+S-MP	**82.9**	**0.77**	**37140838**

3) Comparison experiments of improved enhanced feature extraction network: PAFPN is used as an enhanced feature extraction network in the original YOLOv7's Neck. The Improved-BiFPN is suggested to replace the original PAFPN in this research in order to increase the detection capabilities of the model for various sizes of beach litter in practical applications. The acquired findings are displayed in Table V. When compared to the original model, the Improved-BiFPN increases the network's detection capacity for targets of various sizes, while also increasing mAP@0.5 by 4.1% and the F1 by 0.05. The Improved-BiFPN significantly improves the detection accuracy of the model without a significant increase in the number of parameters and without decreasing the detection speed, therefore we replace the original Neck for beach litter detection.

TABLE V. COMPARATIVE EXPERIMENTS OF NECK

Model	mAP@0.5	F1	Params	FPS(f/s)
YOLOv7	79	0.73	**37223526**	**40**
YOLOv7+Improved-BiFPN	**83.1**	**0.78**	37354598	**40**

4) Comparison Experiment of Beach Litter Classification Results: The AP of each category of various detection models in the self-built beach litter dataset is displayed in Table VI.

Styrofoam, paper products, and glasswork have excellent AP when YOLOv7 is used directly for beach litter detection, whereas the AP of plastic products, metal products, and wood is relatively low. In YOLOv7+QARepVGG, the AP of each category is well balanced, with the exception of metal products, whose AP has decreased by 3.9%. It is clear that the AP of each category has improved after the SimAM attention mechanism was added to the MP module, and glasswork and metal products have the best average accuracy in comparison to the other models in this research. The introduce of improved-BiFPN effectively contributes to the improvement of AP for each category, with the most notable improvement of 12.4% in AP for wood beach litter with high size variation. In QSB-YOLO, the best AP was achieved for styrofoam, plastic products, and paper products relative to all other models, and the AP for glasswork and wood improved by 3.9% and 6.6%, respectively, with only a slight decrease of 2% for metal products. In conclusion, in general, QSB-YOLO is significantly better than the original algorithm.

TABLE VI. AVERAGE PRECISION FOR EACH CLASS OF BEACH LITTER

Model	Styrofoam	Plastic products	Metal products	Paper products	glasswork	Wood
YOLOv7	88.6	69.6	69.4	83	92.7	70.9
YOLOv7+QARepVGG	90.2	77.3	65.3	86.6	92.2	77.5
YOLOv7+S-MP	92.2	71.4	**71.4**	90.4	**98.7**	73.4
YOLOv7+Improved-BiFPN	93.2	77.9	68.1	87.4	88.9	**83.3**
QSB-YOLO	**94.3**	**80.7**	67.4	**92.5**	96.6	77.5

5) Ablation experiments: To verify the effectiveness of each improvement method in this paper, the three improvement modules were added to the original YOLOv7 network structure one by one, and the ablation experimental results are shown in Table VII and Fig. 9.

Table VII shows that, in comparison to the YOLOv7 model, the YOLOv7+QARepVGG model adds the quantization-friendly QARepVGG module, which not only reduces the number of parameters of the model, increases its detection speed and inference speeds, but also contributes to the improvement of the mAP@0.5 of the model. Both YOLOv7+SimAM and YOLOv7 +improved-BiFPN improve the mAP@0.5 and F1 of the model compared to the original YOLOv7. Compared to the original model YOLOv7, QSB-YOLO's mAP@0.5 raises 5.8%, the F1 increases by 0.08, the number of parameters decreases by 281600, the inference speed is accelerated by 3.5ms, and the detection speed of the model is improved by about 43%.

Fig. 9 visualizes the performance difference between each individually added module and YOLOv7 and QSB-YOLO. As can be seen from Fig. 9, the QSB-YOLO model has superior performance and is somewhat advanced in the training process for beach litter detection.

TABLE VII. RESULTS OF THE ABLATION EXPERIMENT ON THE BEACH LITTER DATASET IN THIS PAPER

Model	mAP@0.5	F1	Params	Inference times	FPS(f/s)
YOLOv7	79	0.73	37223526	11.8ms	40
YOLOv7+QARepVGG	80.7	0.75	**36434278**	**8.0ms**	**58**
YOLOv7+S-MP	82.9	0.77	37140838	11.8ms	40
YOLOv7+Improved-BiFPN	83.1	0.78	37354598	11.8ms	40
QSB-YOLO	**84.8**	**0.81**	36941926	8.3ms	57

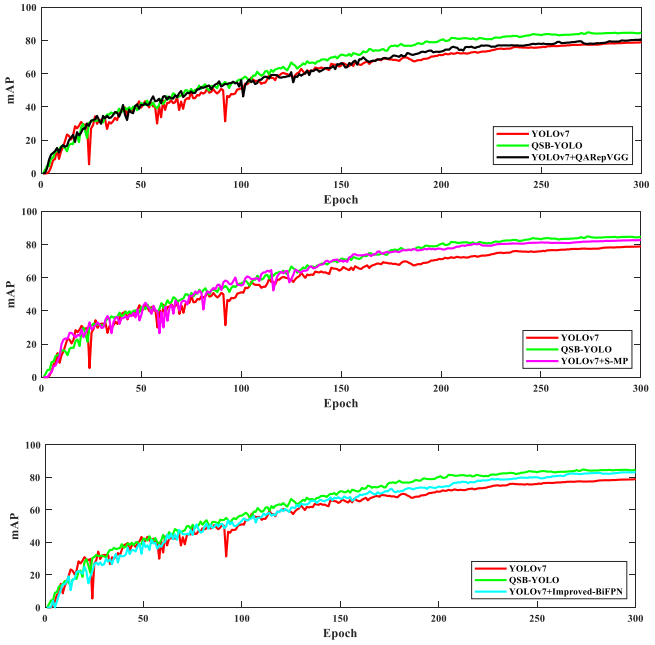


Fig. 9. Comparison chart of ablation experiments.

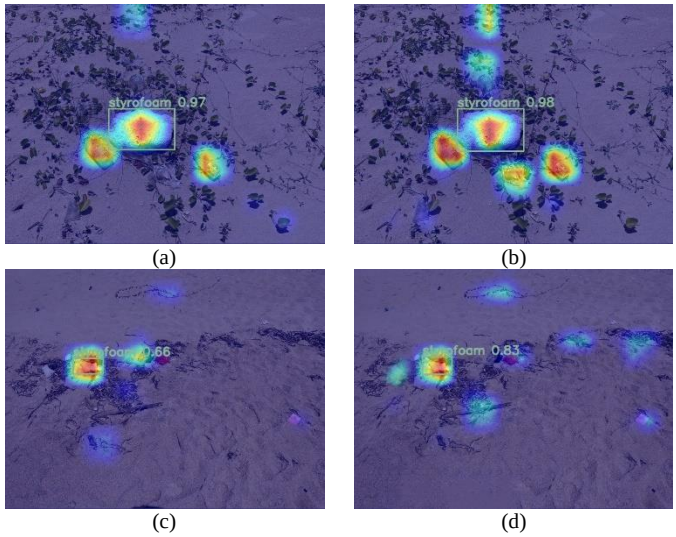


Fig. 10. Heatmap visualization results. (a), (c) the heatmap of YOLOv7; (b), (d) the heatmap of QSB-YOLO.

Fig. 10 shows the heatmap of YOLOv7 and QSB-YOLO, where the darker color represents the more attention of the model. We can see that in a complex background, QSB-YOLO has significantly enhance its ablility to focus on beach litter and has focused on some targets missed in YOLOv7.

6) Comparison with other traditional models: To demonstrate its superiority for beach litter detection, we compare the OSB-YOLO with Faster R-CNN, EfficientDet [13], SSD, YOLOv5 [33], YOLOX [34], and YOLOv7 algorithms. The input image size is 640×640, and the framework used is Pytorch. The experimental results are displayed in Table VIII. Under the same experimental setup and beach litter dataset, QSB-YOLO beats other traditional models in terms of detection speed and mAP@0.5.

TABLE VIII. COMPARISON OF DIFFERENT MODEL DETECTION RESULTS

Model	mAP@0.5	FPS(f/s)
Faster R-CNN	50.9	13
EfficientDet	51.6	23
SSD	62.5	28
YOLOv5	76.8	36
YOLOX	77	37
YOLOv7	79	40
QSB-YOLO	**84.8**	**57**

7) Image detection results: To confirm the real detection performance of QSB-YOLO for beach litter in this paper, we compared the detection effects of YOLOv7 and QSB-YOLO, as illustrated in Fig.11.

It can be found in Fig. 11 that YOLOv7 in Fig. 11(a) does not detect the plastic products beach litter which is fuzzy and smaller in size in the distance, and QSB-YOLO successfully locates and identifies the beach litter. Fig. 11(c) misses a plastic products beach litter of smaller size relative to other

beach litter, which QSB-YOLO also successfully identifies and locates, and the confidence of QSB-YOLO's prediction frame is significantly higher than the confidence of YOLOv7. The beach litter image in Fig. 11(e) has a complex background, and the leaves obscure some of the targets, and YOLOv7 mistakenly detects the leaves as beach litter, and there are also missed detections. In Fig. 11(f), there is no false detection and three fewer missed detections, and only one plastic products beach litter obscured by leaves is not detected.

In summary, compared with the original algorithm, the detection accuracy of QSB-YOLO proposed in this paper is significantly improved, and the leakage and false detection are reduced.

Fig. 11. Comparison of detection effect between YOLOv7 and QSB-YOLO. (a), (c), (e) YOLOv7; (b), (d, (f)QSB-YOLO.

V. CONCLUSION

In our study, we apply data augmentation approaches for the beach litter dataset to increase the model's detection range for beach litter in order to avoid overfitting caused by inadequate samples and propose an efficient QSB-YOLO beach litter detection model to address the issues of missed and false detection in beach litter identification. In addition to reducing the number of model parameters, combining with the quantization-friendly QARepVGG also enhance the model's detection precision and speed. The self-built beach litter dataset used has a complicated backdrop, which causes issues like blurred targets and obscured targets. In order to focus the model on the objective of beach litter and improve detection performance while reducing the possibility of missed and false detection, the MP module in the Neck is paired with the 3D SimAM attention mechanism. The original PAFPN is replaced

with the Improved-BiFPN to increase the network's ability to learn various size characteristics, address the detection difficulties issue brought on by the large size range of beach litter targets, and enhance the model's detection accuracy.

The experimental results show that the QSB-YOLO suggested in this research is far better than the original YOLOv7 and other traditional target detection models for beach litter identification accuracy and detection speed.

In future work, we plan to expand the beach litter dataset to include more diverse beach litter targets and continue research to solve the difficulties beach litter identification in complicated contexts, so as to further improve the practical application value of the model.

ACKNOWLEDGMENT

Thanks are given for the support of Research on Key technologies of intelligent health diagnosis of reservoir Dams driven by knowledge and data (QiankeheSupport [2023] General 251).

REFERENCES

[1] M.L. Campbell, C. Slavin, A. Grage, and A. Kinslow, "Human health impacts from litter on beaches and associated perceptions: A case study of 'clean' Tasmanian beaches," Ocean & Coastal Management, vol. 126, pp. 22-30, 2016.

[2] I. Granado, O.C. Basurko, A. Rubio, *et al.*, "Beach litter forecasting on the south-eastern coast of the Bay of Biscay: A bayesian networks approach," Continental Shelf Research, vol. 180, pp. 14-23, 2019.

[3] R. Pervez, Z.P. Lai, "Spatio-temporal variations of litter on Qingdao tourist beaches in China," Environmental Pollution, vol. 303, 2022.

[4] S. Merlino, M. Paterni, M. Locritani, *et al.*, "Citizen Science for Marine Litter Detection and Classification on Unmanned Aerial Vehicle Images," Water, vol. 13, no. 23, 2021.

[5] E. Nakashima, A. Isobe, S. Magome, S. Kako, N. Deki, "Using aerial photography and in situ measurements to estimate the quantity of macro-litter on beaches," Marine Pollution Bulletin, vol. 62, no. 4, pp. 762-769, 2011.

[6] S. Jang, S. Lee, D. Kim, Y. Hong-Joo, "The Application of Unmanned Aerial Photography for Effective Monitoring of Marine Debris," Journal of the Korean Society of marine environment & safety, vol. 17, no. 4, pp. 307-314, 2011.

[7] S. Kako, A. Isobe, S. Magome, "Low altitude remote-sensing method to monitor marine and beach litter of various colors using a balloon equipped with a digital camera," Marine Pollution Bulletin, vol. 64, no. 6, pp. 1156-1162, 2012.

[8] Z. Bao, J. Sha, X. Li, T. Hanchiso, E. Shifaw, "Monitoring of beach litter by automatic interpretation of unmanned aerial vehicle images using the segmentation threshold method," Marine pollution bulletin, vol. 137, pp. 388-398, 2018.

[9] R. Rfeiffer, G. Valentino, R.A. Farrugia, *et al.*, "Detecting beach litter in drone images using deep learning," In Proceedings of the IEEE International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea), pp. 28-32, 2022.

[10] K. Song, J. Jung, S. Lee, S. Park, "A comparative study of deep learning-based network model and conventional method to assess beach debris standing-stock," Marine Pollution Bulletin, vol. 168, 2021.

[11] X. Chu, L. Li, B. Zhang, "Make RepVGG Greater Again: A Quantization-aware Approach," arXiv 2022, arXiv: 2212.01593.

[12] L. Yang, R. Zhang, L. Li, X. Xie, "SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks," In Proceedings of the International Conference on Machine Learning (ICML), 2021.

[13] M. Tan, R. Pang, Q.V. Le, "EfficientDet: Scalable and Efficient Object Detection," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10778-10787, 2020.

[14] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, "Path Aggregation Network for Instance Segmentation," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8759-8768, 2018.

[15] F. Sultana, A. Sufian, P. Dutta, "A Review of Object Detection Models based on Convolutional Neural Network," arXiv 2019, arXiv: 1905.0614.

[16] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.

[17] R. Girshick, "Fast R-CNN," In Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448, 2015.

[18] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," In Proceedings of the IEEE Transactions on Pattern Analysis & Machine Intelligence, pp. 1137-1149, 2017.

[19] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788, 2016.

[20] W. Liu, D. Anguelov, D. Erhan, *et al.*, "SSD: Single shot multibox detector," In proceedings on European Conference on Computer Vision, pp. 21-37, 2016.

[21] J. Redmon, A. Farhadi, "YOLO9000: Better, Faster, Stronger," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 21–26, 2017.

[22] J. Redmon, A. Farhadi, "Yolov3: An incremental improvement," arXiv 2018, arXiv:1804.02767.

[23] A. Bochkovskiy, C. Wang, H. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv 2020, arXiv:2004.10934.

[24] C. Wang, A. Bochkovskiy, H. Liao, "YOLOv7: Trainable bag- of-freebies sets new state-of-the-art for real-time object detectors," arXiv 2022, arXiv:2207.02696.

[25] H. Wang, Y. Wang, "Improved glove defect detection algorithm based on YOLOv5 framework," In Proceedings of the IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IEEE IAEAC), pp. 1192-1197, 2022.

[26] Z. Xu, J. Li, Y. Meng, X. Zhang, "CAP-YOLO: Channel Attention Based Pruning YOLO for Coal Mine Real-Time Intelligent Monitoring," Sensors, vol. 22, no. 12, 2022.

[27] J. Lin, H. Lin, F. Wang, "A Semi-Supervised Method for Real-Time Forest Fire Detection Algorithm Based on Adaptively Spatial Feature Fusion," Forests, vol. 14, no. 2, 2023.

[28] Y. Li, X. Zhang, Z. Shen, "YOLO-Submarine Cable: An Improved YOLO-V3 Network for Object Detection on Submarine Cable Images," Journal Of Marine Science And Engineering, vol. 10, no. 8, 2022.

[29] X. Ding, X. Zhang, N. Ma, *et al.*, "RepVGG: Making VGG-style ConvNets Great Again," In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13728-13737, 2021.

[30] T. Lin, P. Dollar, R. Girshick, *et al.*, "Feature Pyramid Networks for Object Detection," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936-94, 2017.

[31] S. Woo, J. Park, J. Lee, I.S. Kweon, "CBAM: convolutional block attention module," arXiv 2018, arXiv: 1807.06521.

[32] Q. Wang, B. Wu, P. Zhu, *et al.*, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11531-11539, 2020.

[33] O.M. Lawal, "YOLOv5-LiNet: A lightweight network for fruits instance segmentation," PlOS ONE, vol. 18, no. 3, 2023.

[34] G. Zheng, S. Liu, F. Wang, Z. Li, J. Sun, "YOLOX: Exceeding YOLO Series in 2021," arXiv 2021, arXiv: 2107.08430.