2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Road Damage Detection Method Based on Improved YOLO-World

Yuli Zhou[1], Lei Li[1], Yushan Ma[1], Wen Ya[1], Bin Gu[1], and Guanchun Song[2]

[1] School of Information Engineering, Yangzhou University, Yangzhou 225127, JS, China
[2] College of Civil Science and Engineering, Yangzhou University, Yangzhou 225127, JS, China
221302129@stu.yzu.edu.cn

Abstract. Artificial Intelligence (AI)-driven road damage detection is a crucial component of intelligent transportation and smart cities. Given the outstanding performance of the You Only Look Once (YOLO) model in computer vision tasks, intelligent road damage detection technologies based on the YOLO model are currently among the mainstream methods. However, existing methods have limitations such as low accuracy and poor real-time performance when dealing with small objects and complex backgrounds. To alleviate these issues, this paper proposes an intelligent road damage detection method based on an improved YOLO-World model. Firstly, the Spatial Pyramid Pooling Cross Stage Partial Channel (SPPCSPC) convolutional structure and the FasterNet architecture are introduced into the detection backbone of the YOLO-World model. The aim is to simultaneously enhance the model's ability to extract multi-scale features and its detection speed. Secondly, the Convolutional Block Attention Module (CBAM) attention mechanism module is introduced into the detection head to improve the model's ability to extract key features. Finally, experimental results on the constructed complex road damage dataset show that the improved YOLO-World model outperforms existing state-of-the-art methods in terms of accuracy and detection speed. In particular, the mAP50 index of the improved model is 18.2 percentage points higher than that of YOLOv10.

Keywords: Road Damage Detection; Complex Scenarios; YOLO-World; Real-time Detection

1 Introduction

Driven by the global upsurge in smart city construction, the intelligent transportation system, as a key part of urban digital governance, is reshaping modern urban infrastructure management. Road health directly impacts urban operation and public safety. However, traditional manual inspection, with low efficiency and high error rate, fails to meet smart city management needs. Thus, AI-based intelligent road damage detection technology emerges, enabling high-precision and all-weather detection, ensuring smart city operation [1].

The road damage detection technology based on computer vision evolves through several stages. From the traditional machine learning framework based on Haar features+Support Vector Machine (SVM) in the early stage [2], to the two-stage detection architectures represented by Faster Region-based Convolutional Neural Networks (R-CNN) [3], R-CNN series [4] and Cascade R-CNN [5], and then to the one-stage detection paradigm led by the YOLO series, object detection algorithms achieve multi-dimensional technological breakthroughs. Two-stage methods construct a candidate region generation mechanism through the Region Proposal Network (RPN) [6]. They exhibit significant advantages under complex background interference. Sun Chaoyun et al. [7] effectively improve the crack detection accuracy by optimizing Faster R-CNN, while Dongye Chang-lei et al. [8] enhance Mask R-CNN to achieve a breakthrough in pavement damage detection efficiency. However, the high computational cost of such algorithms restricts their real-time applications. Single-stage object detection algorithms significantly simplify the detection process by completing target class prediction and bounding box regression simultaneously in one stage through a unified network [9], which is more in line with the urgent needs of smart cities for real-time monitoring and response capabilities of road infrastructure. Representative algorithms include Single Shot MultiBox Detector (SSD) [10], YOLO series [11] and RetinaNet [12]. In recent years, one-stage algorithms have made much important progress in the field of road damage detection. Yanbo J. Wang et al. [13] realize the recognition of road damages based on the Faster-RCNN and SSD models and win the IEEE Big Data Road Damage Detection Challenge. Among them, the YOLO series algorithms continuously refresh the performance boundaries through architectural innovation, as Gege Guo et al. [14] propose the MN-YOLOv5 algorithm. By integrating the lightweight MobileNetV3 backbone, coordinate attention, and K-Means clustering for prior box optimization, and combining label smoothing regularization and structural re-parameterization, the accuracy and efficiency of road damage detection are significantly improved, and at the same time, the lightweight deployment of the model is realized. Vung Pham et al. [15] propose a YOLOv7-based road damage detection method, integrating a coordinated attention module and optimization techniques. Trained on Google Street View data, the model secures a silver award (data contribution) and a bronze award (model prediction) in the Crowdsensing-based Road Damage Detection Challenge (CRDDC'2022) challenge, demonstrating the efficacy of crowdsourced data and lightweight models. Jiayi Zeng et al. [16] propose the YOLOv8-PD lightweight algorithm. Through multi-scale feature fusion optimization and lightweight design, the accuracy of road damage recognition is significantly improved while maintaining efficient detection, verifying the robustness and practicability of the improved model in complex pavement scenarios. Although researchers have made some progress in road damage detection tasks based on general object detection algorithms such as the YOLOv8 model, there are still three key challenges in complex road scenarios: Firstly, the features of small-scale damages are easily interfered by the complex background, resulting in insufficient multi-scale feature extraction and seriously affecting the detection accuracy [17]. Secondly, the existing models lack the ability to adaptively focus on the key areas of damages during the feature extraction process, and it is difficult to effectively

distinguish background noise from real damage features [18]. Thirdly, there is an inherent contradiction between the computational complexity of high-precision models and the real-time requirements, restricting their deployment feasibility in edge computing scenarios [19]. As an advanced YOLO model, YOLO-World shows good performance in object detection [20] but has limitations in road damage scenarios: insufficient multi-scale feature fusion, weak attention distribution, and a deployment bottleneck due to computational density and edge device resource limitations. These call for improvements to achieve better detection accuracy and practicability. To address these bottlenecks, this study improves the YOLO-World model with contributions in four aspects:

(1) In view of the deficiency of the YOLO-World model in multi-scale feature fusion, we borrow the SPPCSPC [21] structure, which integrates the Spatial Pyramid Pooling (SPP) and the feature fusion module. The multi-scale pooling of SPP can extract features at different scales in parallel, obtaining rich contextual information. Other components of SPPCSPC further cooperate with it to enhance feature fusion. This makes the model more accurate in detecting small objects, more adaptable to object scale changes, and significantly improves the multi-scale object detection performance.

(2) To solve the limitations of the YOLO-World model in feature extraction under complex backgrounds, we introduce the CBAM attention mechanism [22]. This mechanism, with the synergistic effect of the channel and spatial dimensions, strengthens the model's ability to focus on key features, and significantly improves the feature extraction accuracy of the model under complex backgrounds.

(3) Considering the large consumption of computing resources of the YOLO-World model and its difficulty in efficient deployment on edge devices, we draw on the lightweight design concept of the FasterNet module [23]. Integrating FasterNet to the YOLO-World model effectively reduces the computational complexity while ensuring the efficiency of feature extraction, effectively meeting the dual requirements of real-time performance and high accuracy in practical applications.

(4) To optimize the model's generalizability and practical utility, this study constructs a comprehensive road damage dataset incorporating multi-regional samples with varied damage types and severity levels. By significantly expanding the data diversity, the performance and versatility of the model are improved.

2 Our Method

2.1 Overall Architecture

To tackle the limitations of the YOLO-World model in road damage detection tasks, we introduce several key modules to enhance its performance. Fig. 1 illustrates the architecture of the improved YOLO-World model.

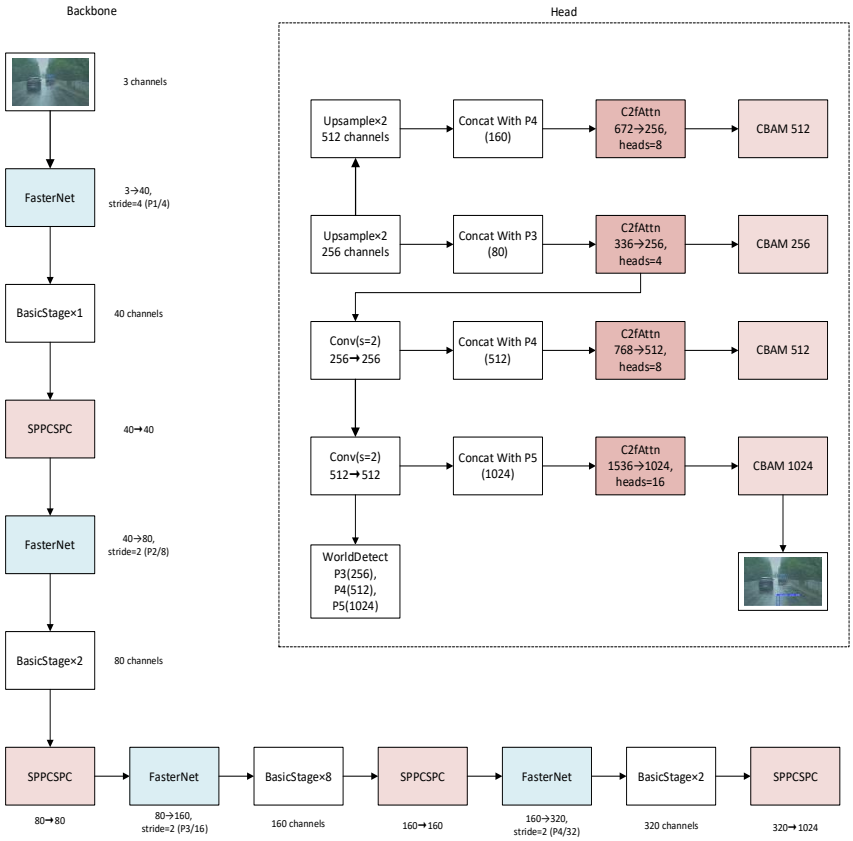


Fig. 1. Architecture Diagram of the Improved YOLO-World Model

Compared with existing YOLO-based road damage detection methods, our approach significantly differs both in architecture and working principles. (1) Differences in Architecture: a) Introduction of FasterNet and SPPCSPC Modules: These modules are incorporated into the backbone network. The FasterNet module utilizes PConv (Partial Convolution) and PWConv (Pointwise Convolution), effectively reducing computational complexity and memory load, thus enhancing the model's detection speed. The SPPCSPC module integrates SPP and CSPC, achieving effective extraction and fusion of multi-scale features. b) Introduction of CBAM Module: The CBAM is integrated into the detection head. This module comprises channel attention and spatial attention sub-modules. The channel attention sub-module emphasizes globally important features, while the spatial attention sub-module focuses on local areas, suppressing background noise interference, thereby improving the model's ability to extract critical features. (2) Differences in Working Principles: a) Multi-Scale Feature Extraction: Our

model revolutionizes multi-scale feature extraction through the SPPCSPC module. Unlike traditional YOLO models that rely solely on basic operations, our SPPCSPC module employs diverse max-pooling kernels (5x5, 9x9, 13x13) to extract multi-granular features and efficiently fuses them. b) Complex Background Adaptation: In complex real-world scenarios, CBAM plays a crucial role in distinguishing true damage features from background noise. Its channel and spatial attention mechanisms enhance the model's capability to handle complex backgrounds by emphasizing global and local important features respectively. c) Computational Efficiency: The FasterNet module addresses computational efficiency issues. By utilizing PConv and PWConv, it reduces computational and memory loads without sacrificing feature representation quality, enabling real-time road damage detection. Overall, the SPPCSPC, FasterNet, and CBAM modules work synergistically. SPPCSPC provides rich multi-scale feature information, CBAM enhances the extraction of critical features, and FasterNet ensures efficient feature extraction and cost-effective operation. Together, these enhancements make our method superior in road damage detection. The specific details of each module are as follows.

2.2 Improvement of the Backbone Network

FasterNet module. FasterNet is a Convolutional Neural Network (CNN) architecture specifically designed for efficient feature extraction. Its core design concept lies in optimizing the network structure to significantly reduce computational complexity and latency while enhancing the feature representation ability. The core innovation of this architecture is reflected in the collaborative design mechanism of PConv and PWConv, as shown in Fig. 2.

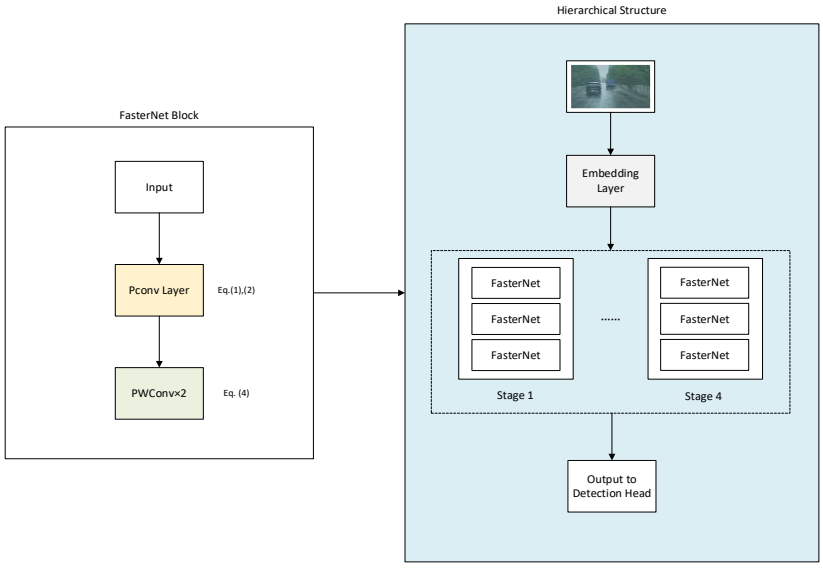


Fig. 2. Structure of the FasterNet Module

On the left side of the figure, the internal structure of the FasterNet block is displayed. Starting from the input, it goes through the Pconv layer and the PWConv×2 layer in sequence. On the right side, the hierarchical structure of FasterNet is presented. After the input image passes through the embedding layer, it is processed by multiple FasterNet modules in several stages and finally output to the detection head.

Based on the redundancy characteristics among the channels of the feature map (for example, adjacent channels have similar features), PConv adopts a channel-selective convolution strategy. It only performs convolution operations on some input channels (such as $\mathrm{c_p} = \frac{1}{4}c$), and its floating-point operations (FLOPs) can be expressed as:

$$FLOPs_{PConv} = h \times \omega \times k^2 \times c_p^2 \tag{1}$$

Where h and ω are the height and width of the feature map, k is the size of the convolution kernel, and c_p is the number of channels involved in convolution. Equation (1) shows that when c_p is only $\frac{1}{4}$ of the total number of channels c, the computational amount of PConv is approximately 93.75% less than that of the standard convolution ($FLOPs = h \times \omega \times k^2 \times c^2$), which greatly reduces the computational cost. The corresponding memory access amount can be expressed as:

$$Memory\ Access_{PConv} = h \times \omega \times 2c_p + k^2 \times c_p^2 \tag{2}$$

Equation (2) demonstrates that by reducing the number of channels involved in calculations, PConv not only decreases the computational load but also significantly optimizes memory usage (a reduction of approximately 60%). This is of great significance for deployment on edge devices with limited computing power, such as road inspection drones. When PConv is combined with PWConv, it is equivalent to a composite convolution with a T-shaped structure whose floating-point operations are expressed as:

$$FLOPs_{T-shaped\ Conv} = h \times \omega \times \left(k^2 \times c_p \times c + c \times \left(c - c_p\right)\right) \tag{3}$$

Compared with Equation (3), traditional T-shaped convolution needs to calculate all channels ($k^2 \times c^2$), while PConv only needs to calculate some channels ($k^2 \times c_p \times c$), thereby minimizing computational redundancy. In contrast, the joint optimization design of PConv and PWConv can further reduce the computational complexity, and the expression of its floating-point operations is:

$$FLOPs_{PConv+PWConv} = h \times \omega \times \left(k^2 \times c_p^2 + c^2\right) \tag{4}$$

The core advantage of Equation (4) is that, by decoupling channel selection and feature fusion, PConv + PWConv, while ensuring the feature representation ability, greatly reduces the floating-point operations, improves the computational efficiency, reduces the consumption of hardware resources, and enables the model to operate efficiently with

low latency in complex tasks. Building on these optimization strategies, FasterNet significantly improves the floating-point operations per second (FLOPS) and effectively reduces the model latency. The latency calculation model can be expressed as follows:

$$Latency = \frac{FLOPs}{FLOPS} \tag{5}$$

The practical significance of Equation (5) is that, when the hardware computing power (FLOPS) is fixed, by optimizing the FLOPs (as shown in Equations 1-4), the model latency is inversely proportional to the computational efficiency.

SPPCSPC module. To enhance the model's detection capability for multi-scale objects, especially small objects in complex scenes, we introduce the SPPCSPC convolution structure into the YOLO-World model. The introduced SPPCSPC structure is shown in Fig. 3, which displays the module structure of SPPCSPC. The input data (C_in) first goes through a 1×1 convolution to convert the number of channels from C_in to C_mid, and then enters the SPP Block. Three branches are split beside the SPP Block, each performing max-pooling operations with different kernel sizes. After that, the data from these branches is concatenated with the data from the main branch in the channel dimension, and the number of channels becomes 4*C_mid. Then, the number of channels is adjusted through a 1×1 convolution. Finally, the data is added to the data from the initial Identity Branch and output. The SPPCSPC integrates the concepts of SPP and Cross Stage Partial (CSP) channels to construct an efficient feature fusion and enhancement mechanism.

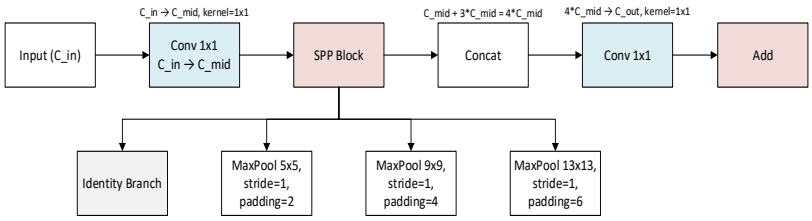


Fig. 3. Structure of the SPPCSPC Module

The SPPCSPC module has a clear design logic and distinct hierarchy, mainly consisting of three stages: feature extraction, multi-scale pooling, and feature fusion and optimization. In the feature extraction stage, the input feature map goes through a 1×1convolution kernel to achieve feature compression and dimensionality regularization. Let the input feature map be $I \in R^{H\times W\times C}$, where H is the height, W is the width, and C is the number of channels. The mathematical expression of the convolution operation is:

$$F_1 = SiLU(I * W_1 + b_1) \tag{6}$$

Where $W_1 \in R^{1\times1\times C\times C'}$ denotes the convolution kernel weight, $b_1 \in R^{C'}$ is the bias term, * represents the convolution operation, SiLU represents the Sigmoid-weighted linear unit activation function, $F_1 \in R^{H\times W\times C'}$is the feature map after convolution and activation, and C' is the number of channels after compression. The core function of this stage is to extract high-order abstract features while suppressing noise interference. After completing feature extraction, it enters the multi-scale pooling stage. The extracted feature F_1 enters the multi-scale max-pooling module. This module adopts a multi-branch parallel pooling structure and uses pooling kernels of different sizes (such as 5×5, 9×9, 13×13) to extract multi-granularity features. Let $k_1 = 5$, $k_2 = 9$, $k_3 = 13$ be the sizes of the three pooling kernels, respectively. For the k_1 pooling kernel, the mathematical expression of its max-pooling operation is as follows:

$$P_1(i,j,c) = max_{m=0}^{4} max_{n=0}^{4} F_1(i+m, j+n, c), \quad i \in [0, H_1 - 1],\ j \in [0, W_1 - 1],\ c \in [0, C' - 1] \tag{7}$$

Among them, $H_1 = \left\lfloor \frac{H-5+1}{1} \right\rfloor$, $W_1 = \left\lfloor \frac{W-5+1}{1} \right\rfloor$.

Similarly, for the pooling kernels k_2 and k_3, we respectively have:

$$P_2(i,j,c) = max_{m=0}^{8} max_{n=0}^{8} F_1(i+m, j+n, c), \quad i \in [0, H_2 - 1], j \in [0, W_2 - 1], c \in [0, C' - 1]; \tag{8}$$

This formula represents the max-pooling operation with a pooling kernel of size $k_2 = 9 \times 9$. Here, we search for the maximum value within a 9×9 neighborhood for each position (i, j) in the feature map F_1across the channel dimension c. The range of i and j is determined by the size of the feature map after applying the pooling operation with a 9×9 kernel, where H_2 and W_2 are calculated based on the formula $\left\lfloor \frac{H-9+1}{1} \right\rfloor$ and $\left\lfloor \frac{W-9+1}{1} \right\rfloor$ respectively, which are used to ensure that the pooling operation is performed within the valid region of the feature map.

$$P_3(i,j,c) = max_{m=0}^{12} max_{n=0}^{12} F_1(i+m, j+n, c), \quad i \in [0, H_3 - 1], j \in [0, W_3 - 1], c \in [0, C' - 1]. \tag{9}$$

This formula is for the max-pooling operation using a pooling kernel of size $k_3 = 13 \times 13$. It finds the maximum value within a13×13neighborhood for each position (i, j) in the feature map $F_1$for each channel c. The ranges of i and j are defined according to the size of the feature map after the 13×13 pooling operation, andH_3 andW_3 are calculated as $\left\lfloor \frac{H-13+1}{1} \right\rfloor$and $\left\lfloor \frac{W-13+1}{1} \right\rfloor$respectively, so as to keep the pooling operation within the boundaries of the feature map.

Among them, $H_2 = \left\lfloor \frac{H-9+1}{1} \right\rfloor$, $W_2 = \left\lfloor \frac{W-9+1}{1} \right\rfloor$, $H_3 = \left\lfloor \frac{H-13+1}{1} \right\rfloor$, $W_3 = \left\lfloor \frac{W-13+1}{1} \right\rfloor$.

The outputs P_1, P_2, and P_3 of each branch are concatenated along the channel dimension to form a multi-scale feature matrix F_{pool}:

$$F_{pool} = [P_1; P_2; P_3] \tag{10}$$

This multi-scale pooling design enables the model to capture both global and local features simultaneously, effectively improving the recognition ability for objects in complex scenes.

Finally, it comes to the feature fusion and optimization stage. In this stage, the multi-scale features F_{pool} are fused with the original input feature map I in the channel dimension. Let the fused feature map be F_{merge}, and its mathematical expression is:

$$F_{merge} = [I; F_{pool}] \tag{11}$$

Subsequently, the fused features F_{merge} are further optimized through several convolutional layers. Assuming there are N convolutional layers, the weight of the convolution kernel of the n-th convolutional layer is W_n, and the bias is b_n, then the output F_n of the n-th convolutional layer is:

$$F_n = SiLU(F_{n-1} * W_n + b_n), \quad n = 1,2,\cdots,N \tag{12}$$

where $F_n = F_{merge}$. The final output detection feature map is $F_{out} = F_N$. These convolutional layers are highly scalable and can flexibly adjust the number of channels and dimensions of the output features according to the task requirements.

The SPPCSPC module caters to the scale-difference characteristics of objects such as cracks and potholes in road distress detection tasks. It enhances the adaptive representation of multi-resolution features through a dynamic feature weighting mechanism, providing an efficient solution for distress identification in complex road scenarios.

2.3 Improvement of Detection Head

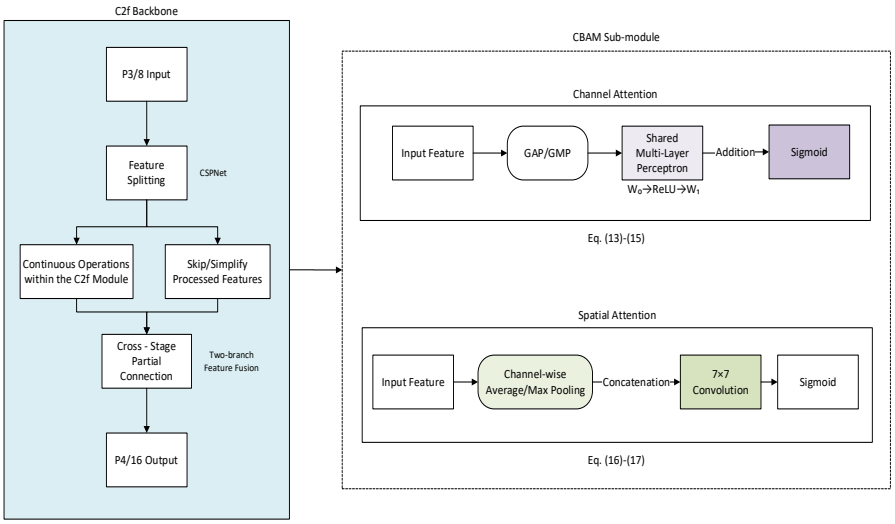


Fig. 4. Combination of CBAM and C2f Module

To enhance the object detection accuracy of the YOLO-World model in complex scenarios, the CBAM attention mechanism is introduced., and the combined structure of CBAM and the C2f module is shown in Fig. 4. The module employs CBAM to guide multi-scale feature map fusion, thus enhancing its focus on key features. The C2f backbone takes the P3/8 feature map as input. By leveraging CSPNet, features are split: one part operates is processed within the C2f module, while the other adopts identity mapping or streamlined operations. These two parts are cross-stage connected and fused to output the P4/16 feature map. The CBAM sub-module consists of channel and spatial attention. For the channel attention, it applies global average and max-pooling to input features, then uses a parameter-shared Multi-Layer Perceptron (MLP) and the Sigmoid function to generate a channel attention map for channel weighting. As for the spatial attention,it pools the input feature map in the channel dimension, concatenates the results, and applies a 7×7 convolution to produce a spatial attention map, which helps in capturing local features. CBAM acts on different-scale feature maps. By integrating these two types of attention, it strengthens the model's focus on key information and cross-layer feature representation, thereby improving object recognition in complex scenarios.

First, the channel attention module models the input feature map's global info, learns and assigns channel weights to identify task-crucial features. Given input feature map F, global average and max-pooling operations yield one-dimensional feature vectors f_{avg} and f_{max} respectively, with calculation formulas as follows:

$$f_{avg} = AvgPool(F),\ f_{max} = MaxPool(F) \tag{13}$$

Then, the compressed feature vectors are input into a shared MLP. The MLP contains a hidden layer, and the size of the activation units is $\frac{C}{r}$ (where r is the channel reduction ratio), and a Rectified Linear Unit (ReLU) activation function is used for non-linear transformation. The formula is as follows:

$$f_{avg}^{c} = W_1\left(ReLU\left(W_0 f_{avg}\right)\right),\ f_{max}^{c} = W_1\left(ReLU\left(W_0 f_{max}\right)\right) \tag{14}$$

Finally, the resultant vectors undergo element-wise summation, and a channel attention map is generated through the Sigmoid function:

$$M_c(F) = \sigma\left(f_{avg}^{c} + f_{max}^{c}\right) \tag{15}$$

where W_0 and W_1 are the weight matrices of the MLP respectively, and σ represents the Sigmoid function.

Secondly, the spatial attention module captures prominent features in the local spatial dimension from the input feature map and identifies the regions of interest in the spatial dimension for the objects. Specifically, average-pooling and max-pooling operations are performed on the input feature map in the channel dimension to generate two two-dimensional feature maps:

$$f_{avg}^{s} = Mean(F, dim = 1),\ f_{max}^{s} = Max(F, dim = 1) \tag{16}$$

Subsequently, the above two feature maps are concatenated in the channel dimension, and a spatial attention map is generated through a 7×7 convolution operation:

$$M_s(F) = \sigma\left(f^{7\times7}\left(\left[f^s_{avg}; f^s_{max}\right]\right)\right) \tag{17}$$

where $f^{7\times7}$represents a 7×7 convolution operation, and $[\cdot;\cdot]$represents the concatenation operation in the channel dimension.

By applying the channel attention map and the spatial attention map to the input feature map in sequence, CBAM can effectively enhance the model's focus on key features, thus improving the accuracy of object detection.

3 Experiments

3.1 Experimental Datasets and Environment Setup

Constructed Dataset. In the field of road distress detection, existing datasets, such as Road Damage Detector 2022 (RDD2022) [24] covering road scenes from multiple countries and Street View Image Dataset for Automated Road Damage Detection (SVRDD) [25] based on high-resolution street-view images, have certain value, yet they suffer from incomplete data coverage. To address this, we extend the existing dataset by collecting and annotating new data. Our collected dataset, consisting of 2000 images collected by vehicle-mounted devices and meticulously annotated, focuses on supplementing samples of long-tail distresses like ruts and raveling, as well as complex scenario data including low-light conditions and occlusions, significantly enhancing data diversity, as shown in Table 1. The two subsets in Table 1 that constitute the dataset are collected by our two authors separately. During the data preprocessing stage, various enhancement strategies are employed to expand the environmental adaptability of the data. Moreover, all integrated datasets are divided into a training set (70%), a validation set (20%), and a test set (10%) at the same ratio to ensure the fairness of experiments.

Table 1. Statistical Information of the Self-annotated Dataset.

	Subset1	Subset2	Total
Alligator Crack	78	6	84
Transverse Crack	765	1000	1765
Longitudinal crack	753	904	1657
Net crack	222	31	253
Pothole	25	86	111
Raveling	102	507	623
Ruts	368	521	889
train_images	700	700	1400
val_images	200	200	400
test_images	100	100	200
total_images	1000	1000	2000
Total instances	2313	3055	5368

Experimental Environment Setup. The hardware is including an NVIDIA GeForce RTX 4060 Laptop GPU, an Intel Core i7-14700HX, and 16.0GB of memory in a lightweight setup. The model runs on Windows 11 with Python 3.10.15, PyTorch 2.1.2+cu118. Poetry locks albumentations 1.4.4, numpy 1.26.3, opencv-python 4.10.0.84.

3.2 Experimental Setup

Parameter Settings. Employing the genetic algorithm, we globally optimize the model's key hyperparameters, searching in the space of learning rate (1e-5 to 1e-1) and data augmentation rotation angle (0.0 to 45.0). Input image resolution is set at 640×640 pixels. We select the AdamW optimizer with an adaptive learning rate of 0.05, weight decay of 0.85, an initial learning rate of 2e-4, and a minimum learning rate ratio of 0.1. The 200-epoch training enhances stability by adjusting data augmentation and extending warmup to 10 epochs. The model is saved every 10 epochs with a patience value of 20. We use the commonly-used classification loss with a classification loss weight (cls) of 0.5 and distribution focal loss with a distribution focal loss weight (dfl) of 1.5 for model training. GPU, mixed-precision training, 4 data-loading processes, and freezing the first 12 network layers improve speed and performance. All parameters, fine-tuned multiple times, ensure experiment fairness and consistency.

Evaluation Indicators. To ensure comparability with related research in the field, this study adopts the same evaluation metrics as those in Ref. [14], including mean Average Precision (mAP), F1-score, Giga-Floating Point Operations Per Second (GFLOPs), number of parameters (in millions), model size (in MB), Frames Per Second (FPS), Precision, and Recall. These metrics comprehensively measure the model's performance from multiple dimensions such as detection accuracy, computational efficiency, and model scale.

3.3 Experimental Results

Table 2. Comparison of the Experimental Results.

Modules	mAP50	mAP50-95	F1/%	GFLOPs	Params/10^6	Model size/MB	FPS
YOLOv9	0.55	0.31	58	236.7	50.71	98	1.0
YOLOv10	0.50	0.28	54	**8.2**	**2.73**	**5.51**	10.4
YOLOv11	0.62	0.36	64	194.4	56.98	109.12	1.3
Improved YOLO-World (ours)	**0.68**	**0.47**	**66**	21.3	5.19	10.1	**12.4**

Table 2 shows the comparative experiment results. The bolded numbers indicate the best results. The improved YOLO-World model surpasses YOLOv9, YOLOv10, and YOLOv11 in key metrics. Its mAP50 hits 68.1%, 13, 18.2, and 6.2 percentage points

higher than the three respectively; mAP50-95 reaches 47.2%, 16.7, 19.4, and 11.1 points higher. Feature representation optimization boosts performance, especially in multi-scale and small-object detection. In terms of accuracy-recall balance, the improved YOLO-World excels. Its F1 value at 66% is 8, 12, and 2 points higher than the others. It raises recall while keeping high accuracy, enhancing distress detection in complex roads. For efficiency, the improved YOLO-World requires only 21.3 GFLOPs, reducing computational cost by 91% and 89% compared to YOLOv9 and YOLOv11; parameters drop to 5.19M, 89.8% and 90.9% less. The FasterNet-inspired design drives this. Model size is cut to 10.1 MB, 89.7% and 90.7% smaller, and FPS rises to 12.4, meeting real-time needs.

3.4 Ablation Study

Table 3. Results of the Ablation Experiment.

Modules	Precision	Recall	mAP50	mAP50-95
YOLO-World	0.55	0.46	0.45	0.22
+CBAM	0.60	0.50	0.54	0.26
+SPPCSPC	0.63	0.61	0.65	0.37
+FasterNet	0.64	0.52	0.575	0.28
+CBAM、SPPCSPC	0.70	0.55	0.62	0.43
+CBAM、FasterNet	0.71	0.59	**0.71**	0.42
+SPPCSPC、FasterNet	0.66	**0.63**	0.65	**0.48**
+CBAM、SPPCSPC、FasterNet	**0.73**	0.60	0.68	0.47

To systematically analyze the impact mechanisms of the three modules, namely CBAM, FasterNet, and SPPCSPC, on the model's evaluation metrics, we conduct ablation experiments. Table 3 shows the results of the ablation experiments. The experimental results indicate that each module and various combinations exhibit unique performance optimization characteristics in different evaluation dimensions. For example, in terms of the Precision metric, when CBAM is combined with either FasterNet or SPPCSPC and applied to the YOLO-World model, the Precision value reaches around 0.70. When the three modules of CBAM, FasterNet, and SPPCSPC are used in synergy, the Precision value reaches 0.73, showing a further improvement compared with pairwise combinations and demonstrating a more powerful optimization effect under the synergy of multiple modules. In terms of the mAP50 metric, after adding the CBAM module to YOLO-World, the mAP50 increases from 0.45 to 0.54. When the FasterNet and SPPCSPC modules are used alone, they increase the mAP50 to 0.575 and 0.65, respectively. When CBAM and FasterNet are used in combination, the mAP50 value exceeds 0.7, significantly improving the model's detection performance. When the three modules of CBAM, FasterNet, and SPPCSPC act in concert, the mAP50 value reaches 0.68, indicating the different effects of different module combinations in improving the model performance. In short, these modules work well together as CBAM pinpoints key features, FasterNet boosts efficiency, and SPPCSPC enriches multi-scale features.

Their combined action optimizes different aspects of the model, leading to improved performance.

3.5 Analysis of Detection Capability in Complex Scenes

Fig. 5. Detection Results Comparison of Each Model under Rainy Road Conditions

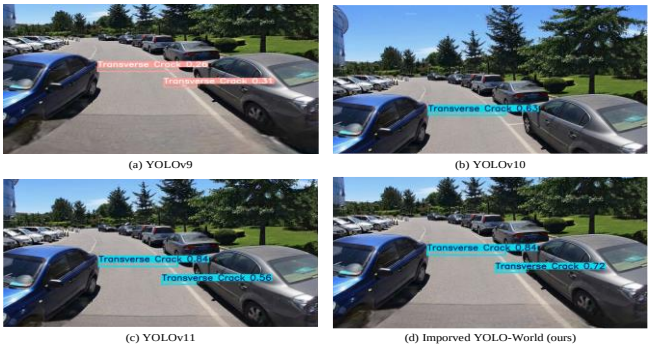


Fig. 6. Detection Results Comparison of Each Model for Small Objects

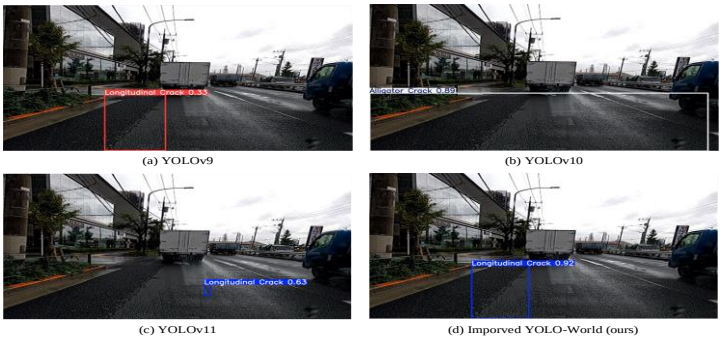
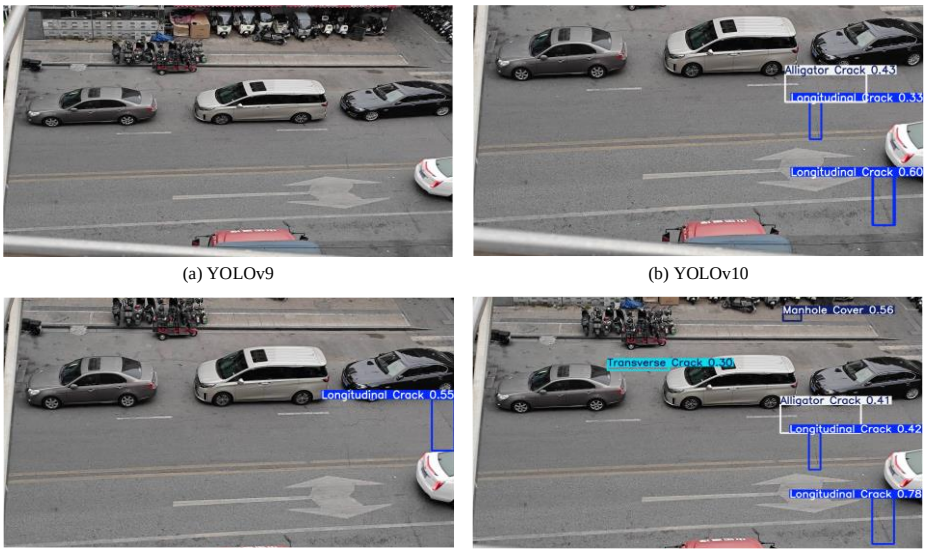


Fig. 7. Detection Results Comparison of Each Model under Poor Lighting Conditions



(a) YOLOv9 (b) YOLOv10

(c) YOLOv11 (d) Imporved YOLO-World (ours)

Fig. 8. Detection Results Comparison of Each Model under Multi-vehicle Interference Conditions

To comprehensively evaluate the performance of the improved YOLO-World model in complex scenarios, we carry out a multi-scene detection ability test. The test results are shown in Figs. 5-8. The improved YOLO-World outperforms baseline models across all test scenarios: In Fig. 5, YOLOv9 has a problem of missed detection, while YOLOv10, YOLOv11, and the improved YOLO-World can all detect all the objects, among which the improved YOLO-World has the highest confidence level. In Fig. 6, YOLOv10 has a missed detection phenomenon, and YOLOv9, YOLOv11, and the improved YOLO-World can all detect the objects, with the improved YOLO-World still maintaining the leading confidence level. In Fig. 7, YOLOv10 and YOLOv11 have false detections, while the improved YOLO-World and YOLOv9 can accurately detect the objects, and the confidence level of the improved YOLO-World is significantly higher than that of YOLOv9. In Fig. 8, all four models have missed detection problems, but the improved YOLO-World has the fewest missed detections, the largest number of detected objects, and a relatively high confidence level. In conclusion, the improved YOLO-World model can stably detect road damages in various complex environments and shows a high detection accuracy.

4 Conclusion

This study addresses road damage detection challenges by optimizing the YOLO-World framework and validating it via comprehensive experiments. Introducing the

SPPCSPC, FasterNet, and CBAM modules into the backbone and detection head enhances multi-scale feature extraction, detection speed, and key feature extraction. Experimental results on our constructed road damage dataset demonstrate superior performance over state-of-the-art methods. Future research will investigate open-set detection models and their Unmanned Aerial Vehicle (UAV) deployment to advance intelligent road maintenance systems.

Acknowledgments. This study is funded by Jiangsu Province University Innovation Training Program (grant number 202411117184Y) and the Natural Science Foundation of Jiangsu Province (grant number BK20230564).

References

1. Zhou, Y., Guo, X., Hou, F., Wu, J.: Review of intelligent road defects detection technology. Journal Sustainability 14(10), 6306 (2022).
2. Rani, P., Sharma, R.: Intelligent transportation system for internet of vehicles based vehicular networks for smart cities. Computers and Electrical Engineering 105, 108543 (2023).
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6), 1137–1149 (2017).
4. Yu, C., Hu, Z., Li, R., Xia, X., Zhao, Y., Fan, X., Bai, Y.: Segmentation and density sstatistics of mariculture cages from remote sensing images using mask R-CNN. Information Processing in Agriculture 9 (3), 417–430 (2022).
5. Cai, Z., Vasconcelos, N.: Cascade R-CNN: Delving into High Quality Object Detection. In: IEEE Conference on Computer Vision and Pattern Recognition 2018, pp. 6154–6162. IEEE Computer Society, Los Alamitos, CA (2018).
6. Varghese, R., Sambath, M.: A Comprehensive Review on Two-Stage Object Detection Algorithms. In: International Conference on Quantum Technologies, Communications, Computing, Hardware and Embedded Systems Security, pp. 1 – 7. IEEE (2023).
7. Sun, C.Y., Pei, L.L., Li, W., Hao, X.L., Chen, Y.: Improved Faster R-CNN-based pavement sealing crack detection method. Journal of South China University of Technology (Natural Science Edition) 48(2), 84-93 (2020).
8. Dongye, C.-L., Liu, H.: A Pavement Disease Detection Method based on the Improved Mask R-CNN. In: 5th International Conference on Information Science, Computer Technology and Transportation, pp. 619–623. IEEE, Piscataway, NJ (2020).
9. Li, W., Feng, X. S., Zha, K., Li, S., Zhu, H. S.: Summary of Target Detection Algorithms. Journal of Physics: Conference Series 1757 (1), (2021).
10. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision-ECCV 2016, vol. 9905, pp. 21-37. Springer, Cham (2016).
11. Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J.: Object Detection in 20 Years: A Survey. Proceedings of the IEEE 111(3), 257–276 (2023).
12. Ale, L., Zhang, N., Li, L.: Road Damage Detection Using RetinaNet. In: IEEE International Conference on Big Data (BigData 2018), pp. 5197–5200. IEEE, Piscataway, NJ (2018).

13. Wang, Y.J., Ding, M., Kan, S., Zhang, S., Lu, C.: Deep Proposal and Detection Networks for Road Damage Detection and Classification. In: IEEE International Conference on Big Data (BigData 2018), pp. 5224–5227. IEEE, Piscataway, NJ (2018).
14. Guo, G., Zhang, Z.: Road Damage Detection Algorithm for Improved YOLOv5. Sci Rep 12, 15523 (2022).
15. Pham, V., Nguyen, D., Donan, C.: Road Damage Detection and Classification with YOLOv7. In: IEEE International Conference on Big Data (BigData 2022), pp. 6416–6423. IEEE, Piscataway, NJ (2022).
16. Zeng, J., Zhong, H.: YOLOv8-PD: An Improved Road Damage Detection Algorithm Based on YOLOv8n Model. Sci Rep 14, 12052 (2024).
17. Xue, Z., Chen, W., Li, J.: Enhancement and Fusion of Multi-Scale Feature Maps for Small Object Detection. In: 39th Chinese Control Conference (CCC 2020), pp. 7212–7217. IEEE (2020).
18. Zhang, Y., Zhang, M.L., Lyu, X.L., Guo, C., Jiang, Z.H.: Review of Research on Small Target Detection Based on Deep Learning. Computer Engineering and Applications 58(15), 1–17 (2022).
19. Dong, W. X., Liang, H. T., Liu, G. Z., Hu, Q., Yu, X.: Review of Deep Convolution Applied to Target Detection Algorithms. Journal of Frontiers of Computer Science and Technology 16(5), 1025–1042 (2022).
20. Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., Shan, Y.: YOLO-World: Real-Time Open-Vocabulary Object Detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2024), pp. 16901–16911. IEEE, Piscataway, NJ (2024).
21. Qi, X. M., Chai, R., Gao, Y. M.: Algorithm of Reconstructed SPPCSPC and Optimized Downsampling for Small Object Detection. Computer Engineering and Applications 59(20), 158–166 (2023).
22. Woo, S., Park, J., Lee, J.-Y., Kweon, I. S.: CBAM: Convolutional Block Attention Module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3 - 19 (2018).
23. Chen, J., Kao, S.-H., He, H., Zhuo, W., Wen, S., Lee, C.-H., Chan, S.-H. G.: Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12021 - 12031 (2023).
24. Arya, D., Maeda, H., Ghosh, S.K., Toshniwal, D., Sekimoto, Y.: RDD2022: a multi-national image dataset for automatic road damage detection. arXiv preprint arXiv:2209.08538 (2022).
25. Ren, M., Zhang, X., Zhi, X., Wei, Y., Feng, Z.: An annotated street view image dataset for automated road damage detection. Scientific Data 11, 407 (2024).