ITC 1/54

Information Technology and Control
Vol. 54 / No. 1/ 2025
pp. 290-306
DOI 10.5755/j01.itc.54.1.37512

YOLOv8-GRW: A YOLOv8-based Algorithm for Road Defect Detection

Received 2024/06/02

Accepted after revision 2024/11/14

HOW TO CITE: Wang, D., Xu, A., Zhu, Y., Yang, D., Xu, Y. (2025). YOLOv8-GRW: A YOLOv8-based Algorithm for Road Defect Detection. *Information Technology and Control,* 54(1), 290-306. https://doi.org/10.5755/j01.itc.54.1.37512

YOLOv8-GRW: A YOLOv8-based Algorithm for Road Defect Detection

Dong Wang, Ao Xu

School of Computer Science & Information Engineering, Shanghai Institute of Technology, Shanghai, 201418, China; e-mails: xuao_dyx@qq.com, dongwang@sit.edu.cn

Yongjian Zhu

Ningbo Minjie Information Technology Co., Ningbo Zhejiang, 315300, China; e-mail: zhuyongjian_hn@126.com

Du Yang, Yintian Xu

School of Computer Science & Information Engineering, Shanghai Institute of Technology, Shanghai, 201418, China; e-mails: yangdu98@qq.com, 1972919465@qq.com

Corresponding authors: dongwang@sit.edu.cn

With the rapid growth of modern road construction mileage, timely maintenance of roads is crucial for driving safety. However, traditional road defect detection methods suffer from high costs and low detection efficiency. To address these issues, this paper proposes a road defect detection algorithm based on YOLOv8-GRW. Firstly, in the backbone network, a new convolutional structure GSPConv, improved based on GhostConv and space-to-depth (SPD), is proposed to preserve the fine details of road defect features and enhance the model's feature extraction ability. Secondly, in the feature fusion, the RepGFPN fusion method, improved based on GhostConv, is adopted, effectively reducing the model complexity brought by RepGFPN and enhancing the model's feature fusion capability. Finally, the WNIoU loss function is introduced, adding the Normalized Wasserstein Distance (NWD) metric to the WIoU loss function to solve the bounding box regression balance problem between high-quality and low-quality samples, enhancing the performance of small object detection. Experimental results show that compared to the original YOLOv8n algorithm, the improved model increases detection accuracy by 3.6%, F1 score by 2.2%, mAP@0.5 by 2.6%, and achieves a detection speed of 200 FPS, demonstrating the effectiveness of the proposed improvements in road defect detection.

KEYWORDS: Road Defect Detection, YOLOv8, GhostConv, SPD, RepGFPN, NWD, WIoU.

1. Introduction

Roads are essential infrastructure in social life, playing a vital role in promoting economic development, social connectivity, and living convenience. However, with the increasing traffic flow and the influence of environmental factors, roads frequently suffer from defects like cracks and potholes [10]. Road defects not only severely affect driving comfort but can also cause further vehicle damage and even lead to traffic accidents [35]. Thus, efficient and accurate methods for detecting pavement defects are critical for road maintenance.

Early road defect detection mainly depended on manual visual inspection, which is not only inefficient and expensive but also has safety concerns [37]. Furthermore, the detection results are susceptible to subjective factors, leading to unreliable accuracy. With the development of image processing technologies and machine learning, researchers have progressively explored using grayscale threshold, texture analysis, edge detection, and other image processing technologies combined with machine learning algorithms to carry out road defect detection tasks [18, 45, 22]. For example, Huang et al. [11] applied digital image processing technology to capture and preprocess images of road surfaces, extracting crack features using edge detection and morphological operations, and achieved automatic crack detection and classification through designed algorithms. Jakštys et al. [12] utilized 2D images captured by a single smart device camera, identifying and approximating road pothole contours based on features such as color, shape, and structure, thus achieving detection and contour mapping of road potholes. Azhar et al. [2] and colleagues trained a Naive Bayes Classifier (NBC) to learn HOG features and combined it with the Normalized Graph Cut Segmentation (NGCS) algorithm, achieving detection and localization of potholes on asphalt roads. Additionally, automated road detection vehicles equipped with high-precision detection devices such as laser scanners and radar have shown good results in road defect detection [7]. However, these traditional methods typically depend on significant human resources and costly equipment, and the detection outcomes are easily influenced by environmental factors, resulting in unstable detection, high costs, and inefficiency.

In recent years, with the rapid development of deep learning technology, many researchers have started applying object detection models to pavement measurement and damage detection. Currently, object detection technology is generally divided into two categories: two-stage detection models represented by Faster R-CNN [29], R-FCN [4], and Mask R-CNN [9], and single-stage detection models represented by RetinaNet [24], EfficientDet [34], SSD [25], and YOLO [26, 27, 28, 3, 6, 19, 38].

Two-stage detection models mainly involve two processes: firstly, a region extractor proposes candidate regions that might contain objects, and then a classification network classifies these regions to determine the object's category and position. Although two-stage models have higher accuracy, their detection speed is lower, which fails to meet real-time requirements. Single-stage models view object detection as a regression task, bypassing the complex region selection process and directly extracting the target's category probability and location coordinates from the image, significantly boosting detection speed and aligning better with real-time detection scenarios. However, compared to two-stage detection models, single-stage detection models perform worse in terms of object detection accuracy and localization precision.

As one of the classic single-stage detection algorithms, the YOLO algorithm offers substantial advantages in detection efficiency. Researchers have applied it to road damage detection, resulting in improvements in both detection efficiency and cost for road defect detection. Zhang et al. [41] improved the YOLOv3 algorithm by pretraining on datasets with similar geometric shapes and combining batch normalization and focal loss to enhance detection accuracy. Their research results indicate that this method outperforms the original YOLOv3 and Faster R-CNN algorithms. Jiang et al. [13] proposed the RDD-YOLOv5 model based on drone detection. This model integrates Transformer structures and explicit visual centers to capture long-distance dependencies and aggregate key features with self-attention, while introducing Gaussian error linear units to enhance the model's nonlinear fitting capability, thereby improving road crack detection capabilities. Sun et al.

[32] achieved an enhancement in YOLOv8's accuracy for road defect detection by introducing the SPD-Conv module, utilizing the ASF-YOLO feature fusion method, and improving the C2f structure, among other approaches. However, the above research often encounters issues such as environmental constraints and model complexity. For example, drones cannot perform road inspections under conditions of tree obstructions or low-altitude limitations. The improved target detection model is highly complex, making it unsuitable for deployment on vehicle-mounted edge devices, and its detection frame rate does not meet real-time detection requirements.

To address the above issues, considering the actual work requirements comprehensively, this paper selects the YOLOv8n model from the YOLO series, which balances accuracy and lightweight design, as the baseline model. Considering that defects like cracks and potholes vary in size and shape, limiting the model's ability to learn defect features, and the complex road surface environment makes the detection of defects like cracks susceptible to environmental factors such as lighting and rain, leading to missed detections and false positives in the target detection model, this paper proposes YOLOv8-GRW. Specifically, to improve the model's feature extraction ability, we designed the GSPConv convolutional module to assist the model in capturing rich defect feature information. For the feature fusion, we adopted the RepGFPN enhanced with improved GhostConv, which strengthened the model's feature interaction and optimized its convolution-based cross-scale feature fusion ability. To distinguish it from the original method, this method is referred to as GRepGFPN in the subsequent content. Finally, we designed the WNIoU loss function to enhance the model's generalization ability and improve its detection accuracy for some small objects. The main contributions of this paper are as follows:

1. We introduced the GSPConv convolutional module mainly to replace traditional convolution in the backbone network. Unlike traditional convolutions, this module effectively prevents the loss of fine features during network propagation, preserves fine-grained information, and strengthens the model's feature extraction ability.
2. We introduced an improved RepGFPN feature fusion method to achieve more comprehensive information exchange between high-level semantic information and low-level spatial information. This method enhances the fusion capability of weak features across different scales, introduces GhostConv to replace the traditional convolution in RepGFPN, and effectively reduces the model complexity brought by RepGFPN while maintaining nearly unchanged detection accuracy.
3. We designed the WNIoU loss function to focus the bounding box regression on ordinary quality anchors, while introducing NWD to enhance the detection capability of smaller defects, significantly improving regression accuracy.

The remainder of this paper is organized as follows: Section 2 discusses related work, Section 3 describes the proposed improvements, Section 4 presents the experimental results and analysis, and Section 5 concludes the paper.

2. Related Work

In recent years, the rapid development of machine learning technology has significantly advanced the construction of smart cities [16], particularly in the transportation field. Stanulov et al. [31] combined multiple machine learning algorithms, such as Long Short-Term Memory (LSTM), Support Vector Regression (SVR), and Random Forest (RF), to comprehensively evaluate the hourly passenger volume of flights, providing strong support for future passenger traffic prediction in the aviation industry. Lakhan et al. [17] built a multi-agent reinforcement learning framework based on biometric ticketing data through information fusion, proposing a biometric ticketing authentication algorithm suitable for various transportation environments, paving a new path for the development of smart transportation.

With the in depth research in deep learning technology, neural network-based target detection techniques primarily use Convolutional Neural Networks (CNNs) to train on large volumes of images, effectively extracting key features from the images, enabling rapid and accurate detection of target objects, and thus have been widely applied in scenarios such as traffic monitoring and road detection. As one of the most widely used single-stage target detection algorithms, the YOLOv8 model, with its efficient detection speed, simple architecture, and convenient deployment, has become the preferred solution in the target detection field for balancing accuracy and speed and has been widely applied in various transportation scenarios. For example, Karim et al. [15] proposed a vehicle classification and counting method based on the single-stage YOLOv8 model for traffic analysis in their research. This achieved efficient detection and classification of target objects in complex traffic environments.

Distinct from other YOLO detection models, the main features of the YOLOv8 algorithm are as follows: In the backbone network, Conv and C2f modules are primarily used for feature extraction from input images. C2f, modeled after the ELAN structure in YOLOv7, achieves multi-branch cross-layer connections through the Bottleneck module, enriching the gradient flow information of the model. Then, the SPPF module combines features extracted with different receptive fields through parallel and serial pooling of various kernel sizes and inputs them into the network neck. In the network neck, YOLOv8 adopts the traditional FPN[23]-PAN[20](Feature Pyramid Network-Path Aggregation Network) feature fusion structure to perform feature fusion, introducing lateral connections and cascade operations to effectively integrate features from different levels, constructing a feature pyramid that effectively encompasses multi-scale information. In the output layer, YOLOv8 replaces the Anchor-based approach used in YOLOv5 with Anchor-Free, reducing the cost of parameter tuning. It uses the currently mainstream decoupled head structure, decoupling classification and regression tasks. The detection head uses the TaskAlignedAssigner [5] method to determine positive and negative sample assignments, selecting positive samples based on weighted classification and regression scores. Regarding loss calculation, YOLOv8 abandons the objectness loss typically used in YOLO models, focusing instead on calculating classification and regression losses. The classification loss is computed using Binary Cross Entropy (BCE) [43], while the regression branch leverages Distribution Focal Loss (DFL) [21] and CIoU [44] loss functions. These three losses are weighted in certain proportions to derive the final loss value, enabling the model to more effectively learn features during training. Compared to earlier generations of the YOLO series, YOLOv8 exhibits superior overall performance. In this paper, we further enhance YOLOv8, capitalizing on its notable advantages in road defect detection.

3. Methodology

In this section, we present the specific implementation details of the proposed enhancement methodology for the YOLOv8-GRW model. Distinct from the original YOLOv8, firstly, in the backbone network, we replaced the standard convolution in YOLOv8 with the GSPConv convolution module; secondly, in the network neck, we utilized the GRepGFPN feature fusion method; and finally, we introduced the WNIoU regression loss function. The structure of the improved YOLOv8-GRW model is illustrated in Figure 1.

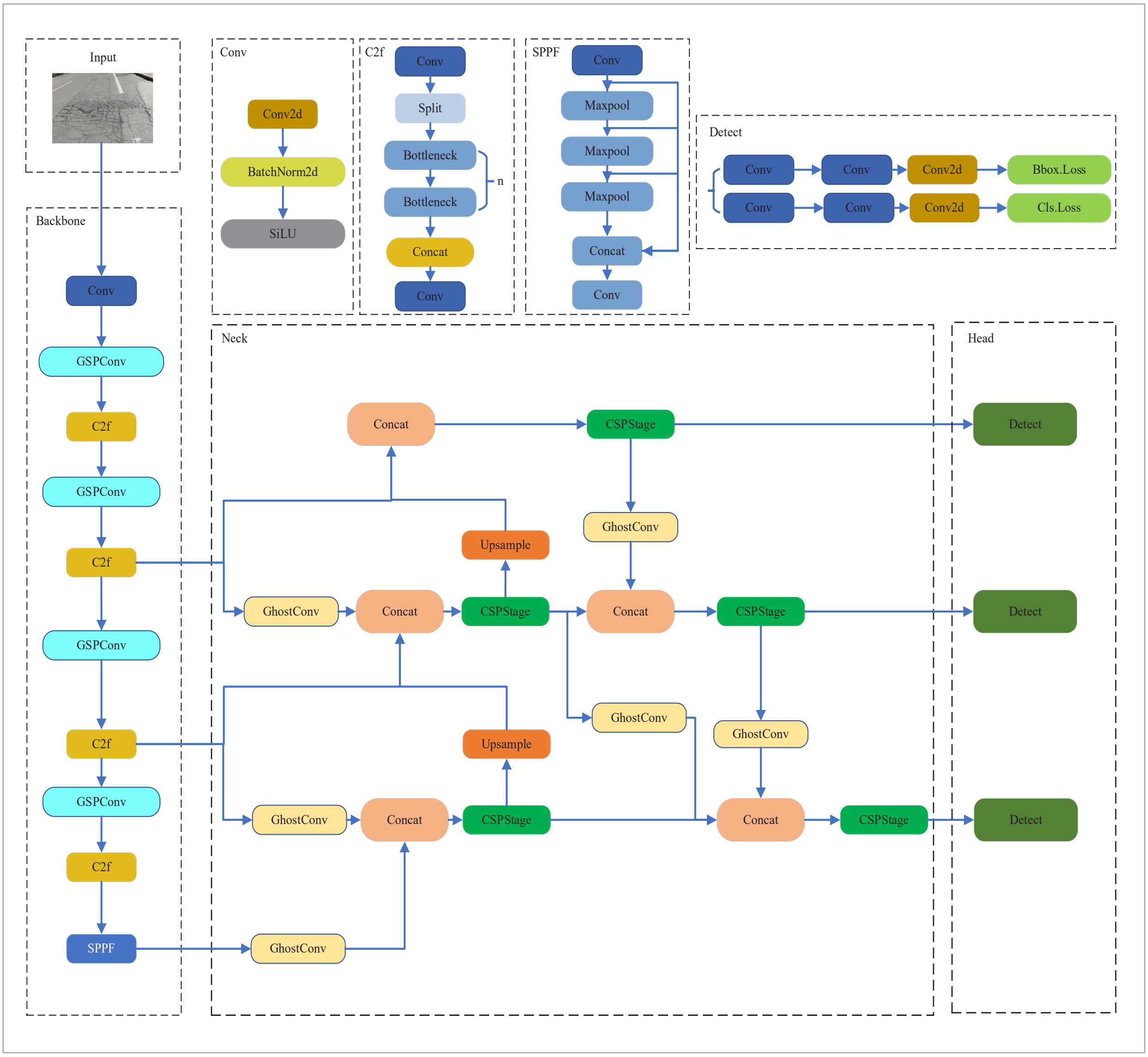
3.1. GSPConv

The YOLOv8 backbone network primarily utilizes multiple standard convolutions and C2f modules for feature extraction from the input feature maps. However, the objects of study in this paper, such as road cracks and holes, vary significantly in shape and size, making their features relatively challenging to extract. Traditional stacked convolutional layers consume substantial network parameters and computational resources but fail to adapt effectively to complex scenarios. To address this issue, we propose the GSPConv structure, specifically designed to enhance feature extraction within the backbone network more efficiently. The detailed architecture of GSPConv integrates the GhostNet convolution module [8], it eliminates the convolutional strides in GhostConv and instead utilizes an SPD operation [33] for downsampling.

The operational procedure depicted in Figure 2 elucidates the process by which the given feature map, with dimensions S×S×C_1, undergoes initial feature extraction through conventional convolution using a minimal set of kernels. This step generates intrinsic feature maps. Subsequently, each channel of the intrinsic feature maps undergoes grouped convolution, serving as a cost-effective linear transformation. The grouped convolution eliminates inter-channel correlations, thereby avoiding the generation of redundant features typically associated with standard convolution and significantly reducing parameter count and computational overhead. These intrinsic

Figure 1

YOLOv8-GRW model structure



feature maps are then combined with their linearly transformed counterparts to create intermediate feature maps with dimensions S×S×C_2. These intermediate feature maps are further processed through an SPD operation, which transforms spatial information into depth information while preserving the fine-grained features of the target. The SPD operation sequences and slices the features of the generated intermediate feature maps S×S×C_2 as illustrated in Equation (1).

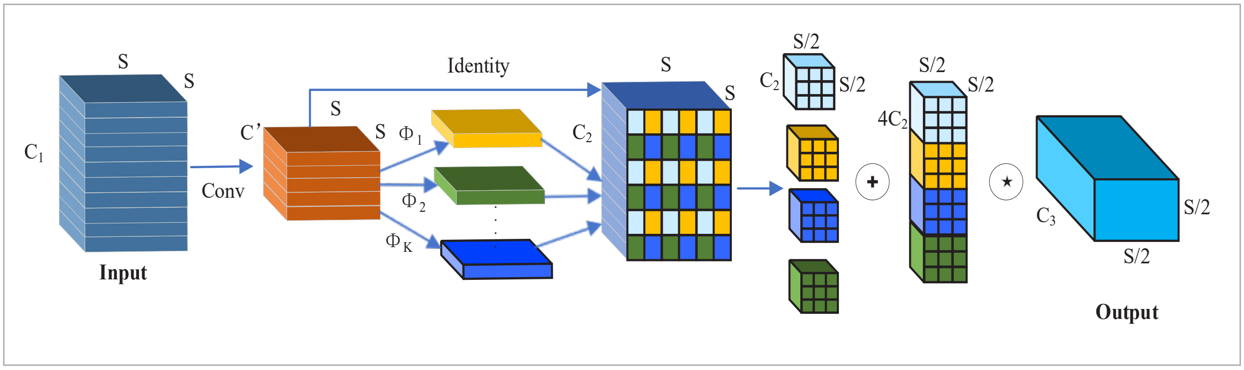
$$f_{0,0} = X[0:S:2,0:S:2], f_{1,0} = X[1:S:2,0:S:2];$$
$$f_{0,1} = X[0:S:2,1:S:2], f_{1,1} = X[1:S:2,1:S:2]; \quad (1)$$

Subsequently, the sliced sub-features are concatenated along the channel dimension, resulting in the final feature map with dimensions S×S×C_3.

The GSPConv module initially utilizes a combination of standard and grouped convolutions to produce intermediate feature maps. This approach not only pre-

Figure 2

GSPConv module structure



serves intrinsic features that are essential for model accuracy, but also results in a reduction in both the number of parameters and computational overhead. However, in scenarios where images contain a significant amount of redundant pixel information, traditional convolutional strides or pooling operations can filter out this redundancy, allowing the model to learn target features more effectively. Yet, these methods face difficulties in challenging scenarios, as stride convolutions or pooling downsampling can lead to the loss of fine details. This issue is particularly critical in the context of road defect detection, where defects vary greatly in size and shape, and features such as cracks are elongated and have low visible resolution. Traditional convolutional strides can cause the model to lose fine-grained information within narrow cracks, resulting in missed detections. Therefore, GSPConv implements SPD operation for downsampling in the processing of intermediate feature maps. While applying SPD to traditional convolutions would significantly increase both computational and parameter burdens, the GSPConv introduces a minimal number of parameters and computations. preventing he loss of fine details during network propagation. This significantly reduces the incidence of missed detections and further enhances the network's capability to extract features.

3.2. GRepGFPN

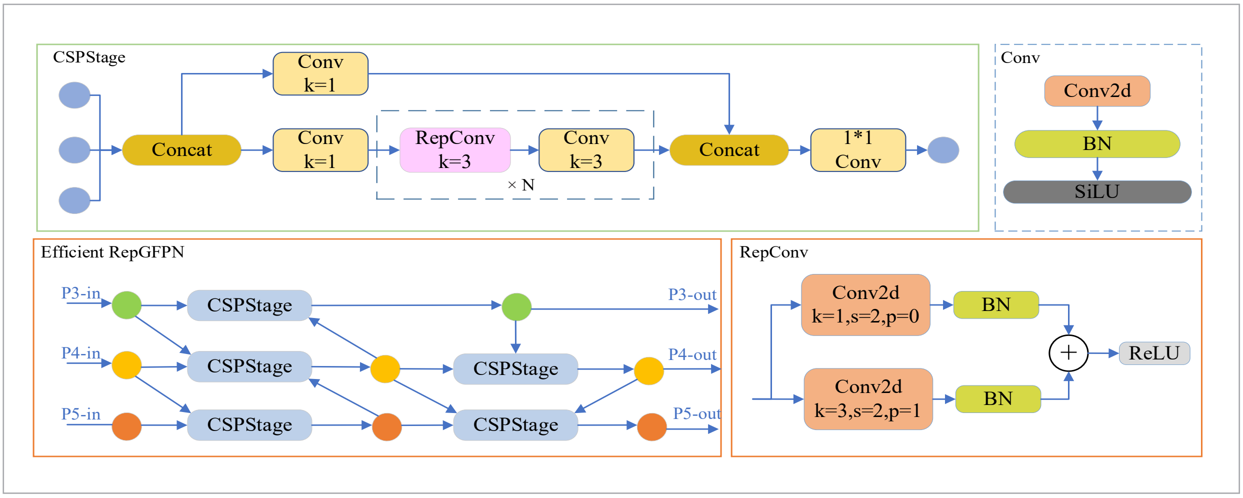
The Feature Pyramid Network (FPN) is specifically designed to aggregate features of different resolutions that are extracted from the backbone network. YOLOv8 incorporates an FPN-PAN structure, which combines Path Aggregation Network (PAN) with FPN. Within this structure, FPN aggregates features outputted from various layers of the backbone network. To address the limitations of unidirectional information flow, an additional bottom-up path aggregation network is added. However, this bottom-up approach may result in insufficient interaction between high-level semantic information and low-level spatial information. High-level semantic information is typically utilized for understanding object categories and overall structure, while low-level spatial information contains details and positional data of objects. The lack of effective interaction between these two types of information can lead to inaccurate spatial localization in detection outcomes, thereby limiting the model's accuracy and robustness.

Therefore, this study utilizes the Efficient-RepGFPN, an efficient layer aggregation network proposed by DAMO-YOLO [40], as the network's neck. This method enhances the feature pyramid used for object detection by more effectively integrating multi-scale features. Building upon the GFPN [14], Efficient-RepGFPN optimizes the topology and fusion methods. It employs different channel counts for features at different scales, eliminates the Queen-Fusion upsampling operation, and incorporates features from different scales using the CSPStage module, which is inspired by re-parameterization ideas and ELAN connections. The specific structure is illustrated in the Figure 3.

Additionally, this paper replaces conventional convolutions in RepGFPN with GhostConv to reduce the computational and parameter load of this feature fusion approach. By utilizing GRepGFPN, the model's accuracy is further enhanced with only a minimal increase in parameter count.

Figure 3

RepGFPN feature fusion module structure



3.3. WNIoU Loss Function

Loss functions are commonly employed to quantify the disparity between the model's predicted output and the actual labels. The regression loss function utilized in YOLOv8 applies CIoU, which is contingent on the minimum bounding box dimensions. However, this approach may not effectively accommodate the significant scale variations often encountered in object detection tasks. In our dataset, targets exhibit substantial differences in scale, and there is considerable variation in the quality of anchor boxes among samples, posing a challenge for gradient updates during training.

To tackle these challenges, we propose the WNIoU loss function as a substitute for the CIoU loss function. Tong et al. [36] introduced the WIoU loss function, with WIoU v1 establishing a boundary loss based on attention mechanisms. This biases anchor box selection towards boxes that closely match the target's height, thereby enhancing detection accuracy. The computational formula is illustrated in Equation (2).

$$L_{WIoUv1} = \exp\left(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g^2+H_g^2)^*}\right)L_{IoU} \qquad (2)$$

In the equation, IoU represents the intersection over union ratio between the predicted bounding box and the actual bounding box. The variables '*x*' and '*y*' denote the coordinates of the center point of the predicted bounding box, while 'x_{gt}' and 'y_{gt}' represent the coordinates of the center point of the actual bounding box. Additionally, '*Wg*' and '*Hg*' refer to the width and height of the smallest bounding box encompassing both the actual and predicted boxes.

WIoU v2 and WIoU v3 improve upon the original WIoU loss function by incorporating a focus mechanism through gradient gain computation. Building on WIoU v1, WIoU v2 introduces a monotonic focusing coefficient to construct the loss function. By using the mean LIoU as a regularization factor, this coefficient is adjusted to accelerate model convergence in the later stages of training. The specific computation formula is presented in Equation (3).

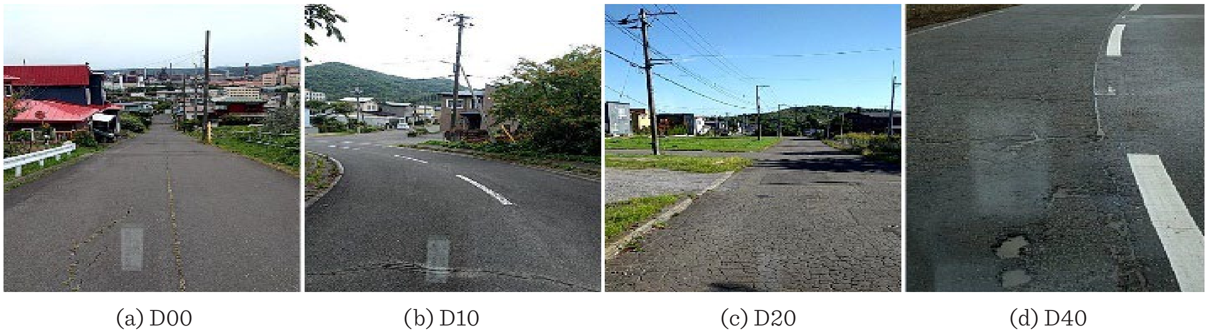
$$L_{WIoUv2} = \left(\frac{L^*_{IoU}}{\overline{L}_{IoU}}\right)^{\gamma} L_{WIoUv1}, \gamma > 0 \qquad (3)$$

In this context, the symbol $\overline{L}_{IOU}$ represents the exponential moving average of momentum *m*, where the superscript * indicates a variable that is detached in the computational graph.

WiseIoU v3 introduces a quality assessment metric for anchor boxes, which utilizes a non-monotonic focusing coefficient β to adjust gradient gains. This adjustment enables the model to allocate more attention to anchor boxes of average quality during training, while mitigating the significant gradient effects

Figure 4

Types of road defect



(a) D00 (b) D10 (c) D20 (d) D40

caused by low-quality samples. By dynamically adjusting gradient gains, the model effectively balances its sensitivity to various quality anchor boxes, thereby enhancing the overall performance of the detection system. The computational formula is presented in Equation (4).

$$L_{WIoUv3} = \frac{\beta}{\delta\alpha^{\beta-\delta}} \exp\left(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g^2+H_g^2)^*}\right) L_{IoU}$$
$$\beta = \frac{L_{IoU}^*}{L_{IoU}} \in [0,+\infty) \tag{4}$$

In this context, β represents the quality of the bounding box, with lower values indicating higher quality. The parameters α and δ are considered as hyperparameters.

This paper primarily utilizes the dynamic non-monotonic focusing mechanism of WIoU v3 to assess the quality of anchor boxes. High-quality and low-quality anchor boxes are allocated smaller gradient gains, enhancing the model's generalization capabilities while mitigating adverse gradients caused by low-quality data. Anchor boxes of average quality are assigned larger gradient gains, focusing more on boundary box regression losses on these average quality anchor boxes. This mechanism prevents harmful gradients caused by low-quality road defect data, addressing the challenge of balancing boundary box regression between high and low-quality data and further improving model performance.

Loss functions based on IoU and its extensions are highly sensitive to positional deviations in small defects, which poses a challenge for the detection of small objects. The NWD [39] introduces a new metric for evaluating the detection of small objects, with the computational formula presented in Equation (5).

$$W_2^2(N_a,N_b) = \left\| \left(\left[cx_a, cy_a, \frac{w_a}{2}, \frac{h_a}{2}\right]^T, \left[cx_b, cy_b, \frac{w_b}{2}, \frac{h_b}{2}\right]^T \right) \right\|_2^2$$
$$L_{NWD} = 1 - \exp\left(-\frac{\sqrt{W_2^2(N_a,N_b)}}{C}\right) \tag{5}$$

In this context, $[cx,cy,w,h]$ represents the coordinates of the bounding box, where C is a constant value.

$W_2^2(N_a,N_b)$ denotes the Wasserstein distance metric.

Consequently, building upon the foundation of WIoU v3, this paper introduces the NWD metric to improve the detection of relatively small defects that are present in limited quantities within the dataset. This leads to the development of the WNIoU loss function, for which the computational formula is presented in Equation (6).

$$L_{WNIoU} = \delta L_{NWD} + (1-\delta) L_{WIoUv3} \tag{6}$$

In the model, δ is a hyperparameter set to 0.02.

4. Experiment

In this section, we first introduce the datasets used for model training and testing. We then provide details of the experimental environment and discuss a series of improvement experiments, ablation studies, and comparative experiments that were conducted.

4.1. Dataset

This study utilizes the open-source road defect dataset RDD2022 [1], which contains road images from various countries. Considering the heterogeneity of the data, the dataset was cleansed and filtered to remove images taken from a top-down perspective. After analysis and processing, the types of defects studied include longitudinal cracks (D00), transverse cracks (D10), alligator cracks (D20), and potholes (D40), as illustrated in Figures 4(a), 4(b), 4(c), and 4(d). The dataset comprises a total of 34,007 images. For the experiments, the dataset was randomly divided into training, validation, and testing sets in a 7:1:2 ratio relative to the total dataset.

4.2. Experimental Environment

The experimental environment for this study was set up on an Ubuntu 20.04 system, with CUDA version 11.6, using the Pytorch 1.13.0 deep learning framework, and Python 3.8 programming language. The computing resources included a 13th generation Intel(R) Core(TM) i7-13700K CPU and an NVIDIA GeForce RTX 4090 (24G) GPU. YOLOv8n served as the baseline for all experiments, which were conducted under consistent hyperparameters. The specific settings for these hyperparameters can be found in Table 1.

Table 1
Hyper Parameter Setting

Hyperparameter	Values
epoch	300
Imagesize	640*640
Learning Rate	0.01
Optimizer	SGD
Momentum	0.937
Weight_decay	0.0005

4.3. Evaluation Indicators

To accurately assess the performance of the model in detecting road defects, this study utilizes the F1 score as a key evaluation metric. The F1 score is a weighted average of precision (P) and recall (R). The calculations for precision and recall are based on the model's performance on the test set, where correctly classified positive samples are True Positives (TP), incorrectly classified positive samples are False Positives (FP), correctly classified negative samples are True Negatives (TN), and incorrectly classified negative samples are False Negatives (FN). The formulas for calculating precision and recall are presented in Equations (7)-(8), respectively, while the formula for the F1 score is shown in Equation (9).

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{9}$$

AP (Average Precision) and mAP (Mean Average Precision) are metrics utilized to evaluate the accuracy and robustness of a model. Higher mAP values indicate superior overall detection performance across all categories. The specific calculations for AP and mAP are presented in Equations (10)-(11), respectively. The term 'Parameter' refers to the size of the model's parameters, reflecting its complexity. GFLOPs (Giga Floating-point Operations Per Second) represent the number of floating-point operations performed per second, commonly used to assess the computational performance of a model during inference or training. FPS (Frames Per Second) is employed to measure inference speed, indicating how many frames per second the model can process.

$$AP = \int_0^1 P(R)dR \tag{10}$$

$$\mathrm{m}AP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{11}$$

In these calculations, N represents the total number of categories.

4.4. Improvement Experiments

To validate the improvement effects of the feature fusion methods, this study conducted comparative experiments involving the original feature fusion approach, RepGFPN, and GRepGFPN within the backbone network enhanced by GSPConv.

Table 2

Comparison of feature fusion methods experiment

Feature Fusion	F1	mAP@0.5	Para/M	GFLOPs
FPN-PAN	0.542	0.532	3.08	10.0
RepGFPN	0.551	0.537	3.41	10.4
GRepGFPN	**0.549**	**0.537**	**3.19**	**9.9**

The experimental results are presented in Table 2. Both RepGFPN and GRepGFPN demonstrated enhancements over the traditional FPN-PAN structure, with improvements observed in both F1 score and mAP@0.5. The introduction of the RepGFPN feature fusion structure allows for effective exchange of high-level semantic information and low-level spatial information, providing more efficient information transfer and enhancing the model's detection capabilities and overall recognition accuracy. Further improvements with the cost-effective GhostConv in GRepGFPN maintained similar performance levels to RepGFPN. This not only strengthened the network's feature fusion capability but also reduced parameter size and computational load introduced by RepGFPN. The experimental results demonstrate that the optimized feature fusion methods achieved a better balance between model size and performance, with increases of 0.7% and 0.5% in F1 score and mAP@0.5, respectively.

To further validate the efficacy of the WNIoU loss function for detection tasks, this study incorporates advanced convolutional and feature pyramid networks, namely GSPConv and GRepGFPN, respectively. A comparative analysis was conducted among several bounding box regression loss functions including CIoU, EIoU [42], GIoU [30], WIoUv1, WIoUv2, WIoUv3, and the proposed WNIoU. As shown in Table 3. To further ascertain the optimal value for the δ hyperparameter in WNIoU, this study conducted comparative experiments by setting δ to 0.01, 0.02, 0.03, and 0.04. As illustrated in Figure 5, the experimental results show that the model achieves optimal performance when the WNIoU hyperparameter δ is set to 0.02.

The above experimental findings demonstrated that the introduction of the WIoU loss function effectively mitigates the adverse gradients generated by

Figure 5

Effect of WNIoU Hyperparameter δ

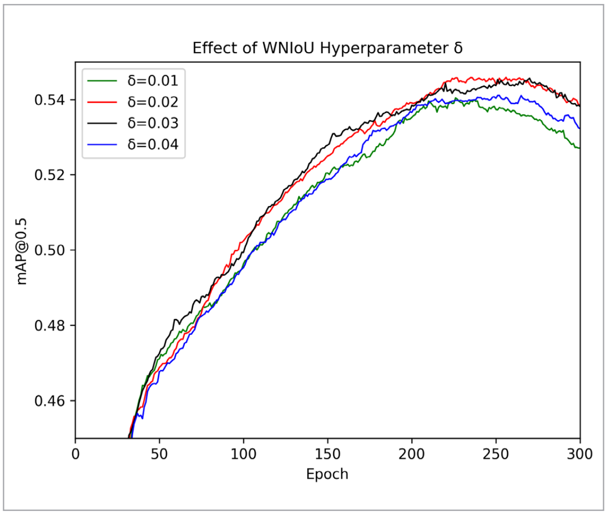


Table 3

Comparison of loss function

Loss function	P	R	F1	mAP@0.5	Para/M	GFLOPs
CIoU	0.586	0.516	0.549	0.537	3.19	9.9
EIoU	0.577	0.522	0.548	0.531	3.19	9.9
GIoU	0.580	0.524	0.551	0.533	3.19	9.9
WIoU v1	0.584	0.520	0.550	0.536	3.19	9.9
WIoU v2	0.594	0.506	0.547	0.533	3.19	9.9
WIoU v3	0.599	0.515	0.554	0.539	3.19	9.9
WNIoU	**0.613**	**0.510**	**0.557**	**0.544**	**3.19**	**9.9**

low-quality samples. This is particularly evident when there is significant overlap between the detection and target boxes, which serves to alleviate the penalties associated with geometric discrepancies. Consequently, this reduction in penalties diminishes the negative impact of low-quality samples on the model's generalization abilities. The WNIoU loss function exhibited superior overall performance. Within the WIoU series, WIoU v3 demonstrated a notable increase in detection precision by 1.3% over the CIoU loss function. Furthermore, the incorporation of a minor proportion of the NWD metric, despite a slight decrease in recall, led to an enhancement in detection precision by an additional 1.4%. The F1 score improved by 0.3%, the mAP@0.5 improved by 0.5%, indicating a more balanced performance between precision and recall, thereby substantiating the effectiveness of the modifications made to the loss function.

4.5. Ablation Experiment

In order to further validate the efficacy of the enhanced algorithm, a series of ablation experiments were conducted in this study. The experimental findings are depicted in Table 4, indicating that the introduction of GSPConv led to an increase of 0.07M parameters and 1.7 GFLOPs computational load, resulting in a 1.4% increase in mAP@0.5 and a 0.7% increase in F1 score, demonstrating its significant impact on performance improvement. Moreover, when only the GRepGFPN and WNIoU modules were introduced, mAP@0.5 decreased by 0.7% compared to using only GSPConv, highlighting GSPConv's ability to retain fine information and enhance object detection performance within the backbone network. Following successive introductions of the GRepGFPN module and WNIoU loss function, there was ultimately an increase by 2.6% and 2.2%, respectively, for mAP@0.5 and F1 score compared to the original model. It shows that GRepGFPN is more conducive to feature extraction and WNIoU loss function is more conducive to feature learning ability in model training.

Through the ablation experiment, the sequential introduction of GSPConv, GRepGFPN, and WNIoU improvement methods resulted in successive enhancements in the model's mAP@0.5 and F1 score based on previous ones, thus demonstrating the effectiveness of different module improvements.

4.6. Comparative Experiment

To more effectively assess the performance of the improved algorithm, this study conducted comparative experiments with Faster RCNN, YOLOX-tiny, YOLOv5s, YOLOv6n, YOLOv7-tiny, YOLOv8n, and YOLOv9s. The results of the experiments are presented in Table 5. Additionally, this study provided a visual comparison of the metrics from the experimental processes of various YOLO series versions, as illustrated in Figure 6.

As shown in Figure 6, YOLOv8-GRW, apart from its significant advantage in overall detection accuracy, can more effectively balance precision and recall, indicating that the improved model enhances the overall model's robustness and practicality, while the well-performing YOLOv5 and YOLOv9 cannot simultaneously balance precision and recall. According to Table 5, the F1 scores of Faster RCNN, YOLOX-tiny, YOLOv5s, and YOLOv9 are higher than the original YOLOv8 model by 1.0%, 1.2%, 0.9%, and 1.0%, respectively. Moreover, YOLOv9s has an mAP@0.5 that is 1.9% higher than the original YOLOv8. Despite the YOLOv8 model hav-

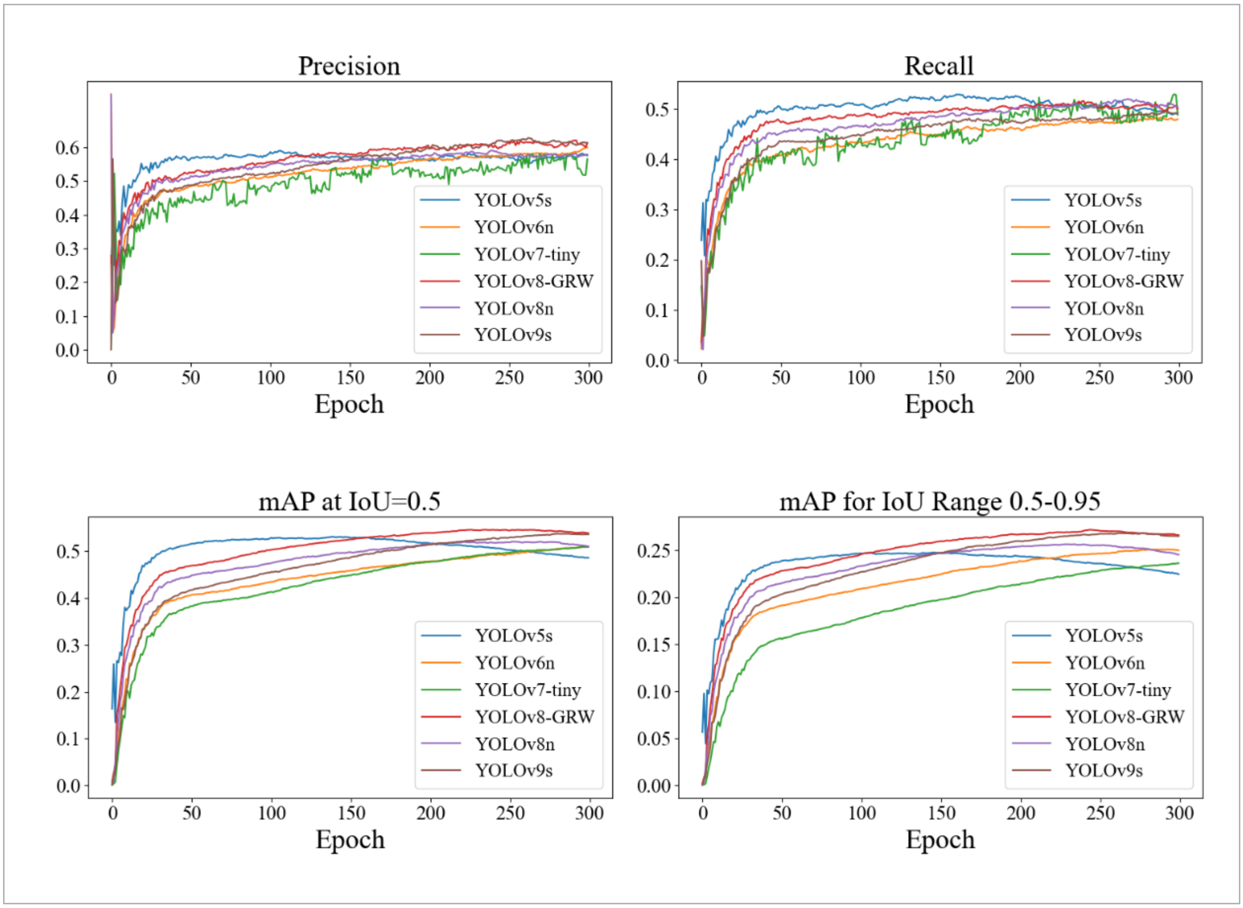
Table 4
Results of ablation experiment

GSPConv	GRepGFPN	WNIoU	F1	mAP@0.5	Para/M	GFLOPs
			0.535	0.518	3.01	8.1
√			0.542	0.532	3.08	9.8
	√	√	0.542	0.525	3.12	8.2
√	√		0.549	0.537	3.19	9.9
√	√	√	0.557	0.544	3.19	9.9

Table 5
Comparison results of different algorithms

Model	P	R	F1	mAP@0.5	FPS	Para/M	GFLOPs
Faster-RCNN	0.584	0.509	0.545	0.503	18	41.48	94.3
YOLOX-tiny	0.611	0.495	0.547	0.489	69	5.06	6.45
YOLOv5s	0.571	0.520	0.544	0.515	192	7.20	16.5
YOLOv6n	0.580	0.486	0.526	0.505	185	4.70	11.4
YOLOv7-tiny	0.567	0.478	0.519	0.490	144	6.02	13.2
YOLOv8n	0.577	0.499	0.535	0.518	213	3.01	8.1
YOLOv9s	0.615	0.490	0.545	0.537	94	4.23	18.3
ours	**0.613**	**0.510**	**0.557**	**0.544**	**200**	**3.19**	**9.9**

Figure 6
Training metrics of various YOLO models



ing notable advantages in terms of lightweight and optimal frame rate, it also shows more serious issues with false detections and missed detections. After applying the improvement methods proposed in this paper, YOLOv8-GRW achieved better performance than Faster RCNN, YOLOX-tiny, YOLOv5s, and YOLOv9s, with F1 scores higher by 1.2%, 1.0%, 1.3%, and 1.2%, and mAP@0.5 values higher by 4.1%, 5.5%, 2.9%, and 0.7%, respectively, achieving the best results.

Although the YOLOv8-GRW model's parameters increased by 0.18M and its computational load by 1.8 GFLOPs compared to the original YOLOv8, its detection speed still reached 200FPS, fulfilling real-time detection needs. Simultaneously, accuracy increased by 3.6%, recall by 1.1%, mAP@0.5 by 2.6%, and F1 score by 2.2%. Among all the compared models, YOLOv8-GRW demonstrated the best performance, making it the most effective model for road defect detection.

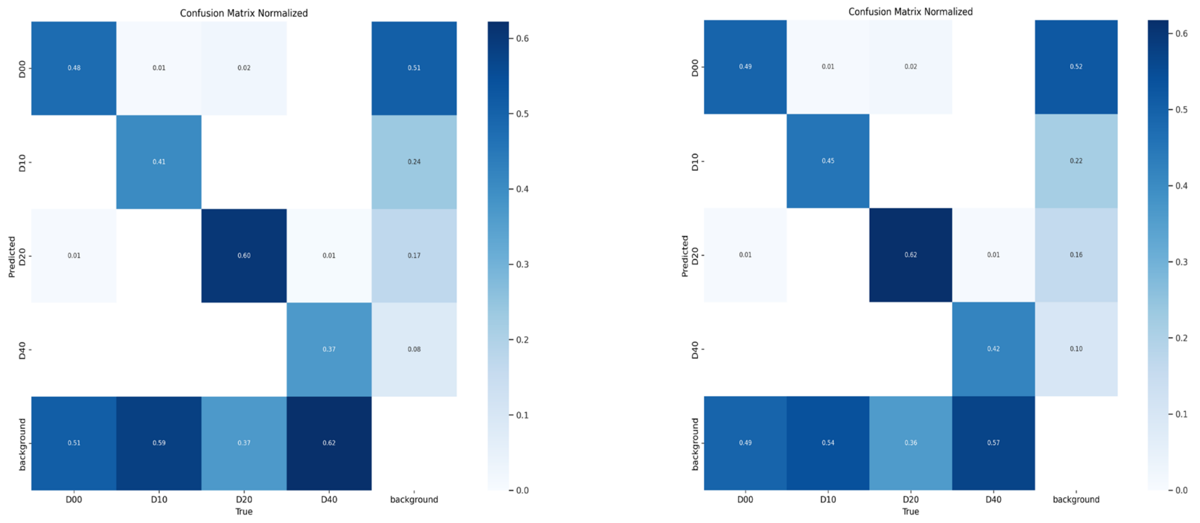
4.7. Visualization Analysis

This study analyzes the confusion matrices of the YOLOv8-GRW model compared to the original model, as shown in Figure 7. YOLOv8-GRW shows improvements across various defect categories, with the most significant enhancement observed in the D40 category, where it improved by 5%. To more vividly demonstrate the effectiveness of the YOLOv8-GRW model, this paper conducted a visual comparison with the original model. The comparative results, depicted in Figure 8, indicate that YOLOv8-GRW has effectively increased the accuracy rate of defect detection across all categories. The issues of missed detections prevalent in the original model have been addressed to some extent, and robustness has also been improved.

For example, Figure 8(i) accurately detects fine longitudinal cracks at the furthest point as well as transverse cracks at the closest point. Figure 8(l) identifies a greater variety of defect categories. Furthermore, the categorization and localization of defects have become more precise. Figure 8(j) illustrates that a large area of crazing is detected, which is not merely confined to small-scale crazing and potholes. The model also precisely identifies and locates smaller and shadowed defect targets, as seen in Figure 8(k), where YOLOv8-GRW accurately recognizes small potholes under vehicle shadows and more distant potholes, showcasing the model's enhanced robustness and validating the effectiveness of the improvements made in this study.

Figure 7

Comparison of Confusion Matrices between the Original YOLOv8 Model and the YOLOv8-GRW Model

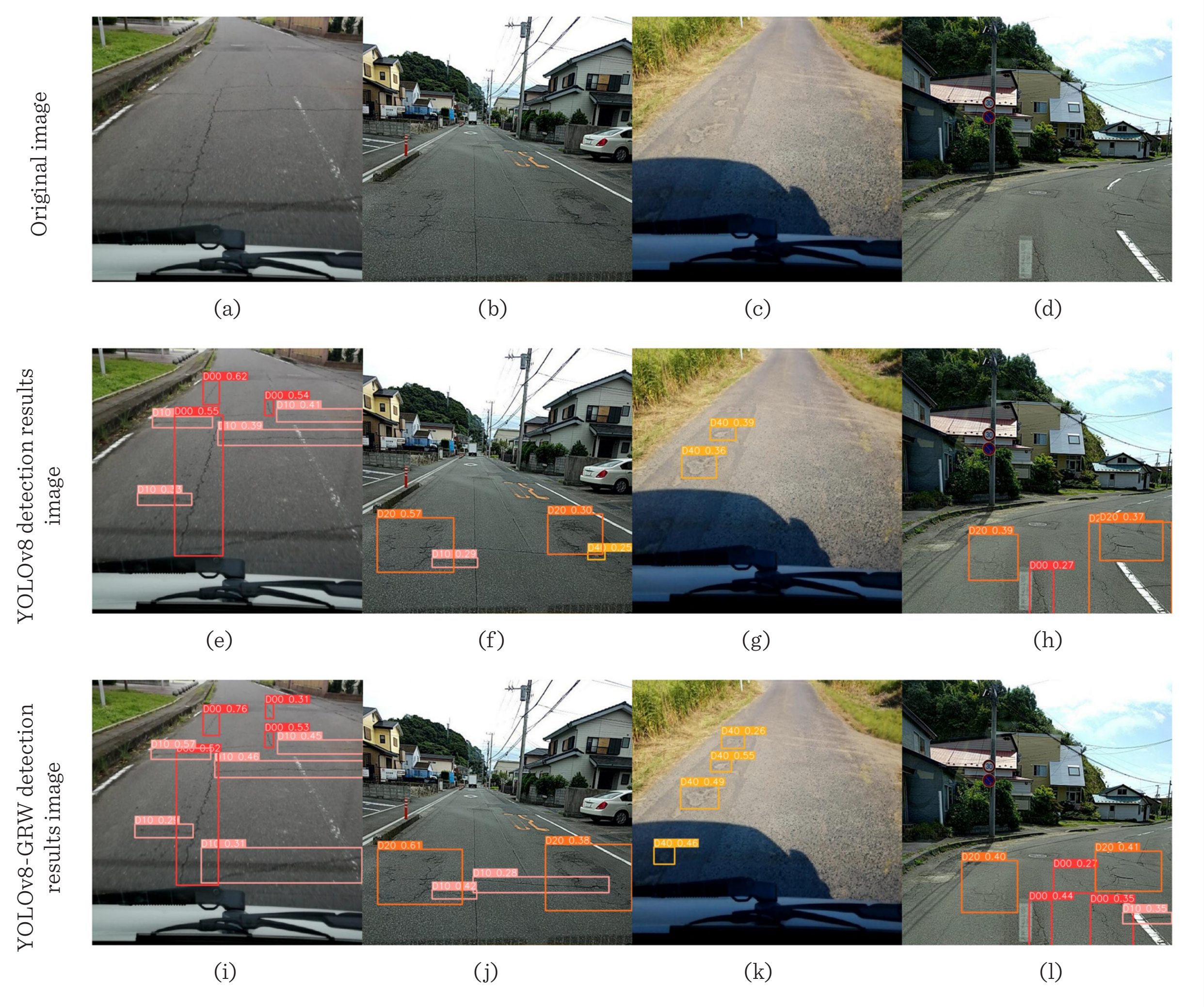


(1) Confusion Matrix of the original YOLOv8 model

(2) Confusion Matrix of the YOLOv8-GRW model

Figure 8

Comparison of performance between the original YOLOv8 model and the YOLOv8-GRW model



5. Conclusion

This paper proposes a road defect detection method based on YOLOv8-GRW. By introducing GSPConv convolution into the backbone network, the fine information of road defects is significantly retained during downward propagation, enhancing the model's perception of different information. In the neck of the network, the RepGFPN feature fusion structure is employed to efficiently integrate multi-scale features, and GhostConv is introduced to make the feature structure more lightweight, enhancing feature fusion while preserving computational efficiency. WNIoU is incorporated into the loss function, leading the model to focus more on samples of ordinary quality, improving its generalization capability, and enhancing its ability to detect smaller targets. Following the improvements made to the network's backbone, neck, and loss function, the F1 score increased by 2.2 percentage points, mAP@0.5 improved by 2.6 percentage points, and detection speed reached 200 FPS. The improved YOLOv8 model not only meets the real-time detection requirements, making it suitable for deployment on edge devices, but also significantly improves the average accuracy of road defect detection, reducing the issues of missed and false detections in target box detection, providing a new solution for related scenarios.

However, the proposed method has certain limitations, such as a single data source and a lack of defect detection data in extreme conditions, which necessitates further improvement in model robustness. Furthermore, the model's accuracy in complex environments still has significant potential for enhancement. In future work, we will introduce more data from different environments and weather conditions to help the model better adapt to various real-world situations. Additionally, we are considering the introduction of attention mechanisms, particularly self-attention mechanisms, to strengthen the model's focus on key areas, thereby further improving its robustness and precision.

Acknowledgement

This work was supported by the Natural Science Foundation of Shanghai (20ZR1455600), the Collaborative Innovation Foundation of Shanghai Institute of Technology (XTCX2022-20) and the Research Initiation Foundation for Newly Recruited Talent of Shanghai Institute of Technology (YJ2022-68).

References

1. 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.
2. Azhar, K., Murtaza, F., Yousaf, M. H., Habib, H. A. Computer Vision-Based Detection and Localization of Potholes in Asphalt Pavement Images. 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, 2016, 1-5. https://doi.org/10.1109/CCECE.2016.7726722
3. Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934, 2020.
4. Dai, J., Li, Y., He, K., Sun, J. R-FCN: Object Detection via Region-Based Fully Convolutional Networks. Advances in Neural Information Processing Systems, 2016, 29.
5. Feng, C., Zhong, Y., Gao, Y., Scott,, M. R., Huang, W. TOOD: Task-Aligned One-Stage Object Detection. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, 2021, 3490-3499. https://doi.org/10.1109/ICCV48922.2021.00349
6. Glenn, J. YOLOv5 Release v6.1, 2022. Source: https://github.com/ultralytics/yolov5/releases/tag/v6.1.
7. Hadjidemetriou, G. M., Vela, P. A. Automated Pavement Patch Detection and Quantification Using Support Vector Machines. American Society of Civil Engineers, 2018. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000724
8. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 1580-1589. https://doi.org/10.1109/CVPR42600.2020.00165
9. He, K., Gkioxari, G., Dollár, P., Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, 2017, 2961-2969. https://doi.org/10.1109/ICCV.2017.322
10. Hou, Y., Li, Q., Zhang, C., Lu, G., Ye, Z., Chen, Y., Wang, L., Cao, D. The State-of-the-Art Review on Applications of Intrusive Sensing, Image Processing Techniques, and Machine Learning Methods in Pavement Monitoring and Analysis. Engineering, 2021, 7(6), 845-856. https://doi.org/10.1016/j.eng.2020.07.030
11. Huang, W., Zhang, N. A Novel Road Crack Detection and Identification Method Using Digital Image Processing Techniques. In Proceedings of the 2012 7th International Conference on Computing and Convergence Technology (ICCCT), Seoul, Republic of Korea, 3-5 December 2012, 397-400.
12. Jakštys, V., Marcinkevičius, V., Treigys, P., Tichonov, J. Detection of the Road Pothole Contour in Raster Images. Information Technology and Control, 2016, 45(3), 300-307. https://doi.org/10.5755/j01.itc.45.3.13446
13. Jiang, Y., Yan, H., Zhang, Y., Wu, K., Liu, R., Lin, C. RDD-YOLOv5: Road Defect Detection Algorithm with Self-Attention Based on Unmanned Aerial Vehicle Inspection. Sensors, 2023, 23(19), 8241. https://doi.org/10.3390/s23198241
14. Jiang, Y., Tan, Z., Wang, J., Sun, X., Lin, M., Li, H. GiraffeDet: A Heavy-Neck Paradigm for Object Detection. arXiv preprint arXiv:2202.04256, 2022.

15. Karim, A. A., Neamah, S. B. Vehicle Classification and Counting for Traffic Analysis Based on Single-Stage YOLOv8 Model. Iraqi Journal of Computers, Communications, Control and Systems Engineering, 2024, 24(2).

16. Lakhan, A., Mohammed, M. A., Abdulkareem, K. H., Deveci, M., Marhoon, H. A., Memon, S., Nedoma, J., Martinek, R. BEDS: Blockchain Energy Efficient IoE Sensors Data Scheduling for Smart Home and Vehicle Applications. Applied Energy, 2024, 369, 123535. https://doi.org/10.1016/j.apenergy.2024.123535

17. Lakhan, A., Rashid, A. N., Mohammed, M. A., Zebari, D., Deveci, M., Wang, L., Abdulkareem, K. H., Nedoma, J., Martinek, R. Multi-Agent Reinforcement Learning Framework Based on Information Fusion Biometric Ticketing Data in Different Public Transport Modes. Information Fusion, 2024, 110, 102471. https://doi.org/10.1016/j.inffus.2024.102471

18. Li, B., Wang, K. C. P., Zhang, A., Fei, Y., Sollazzo, G. Automatic Segmentation and Enhancement of Pavement Cracks Based on 3D Pavement Images. Journal of Advanced Transportation, 2019. https://doi.org/10.1155/2019/1813763

19. Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., Wei, X. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. arXiv preprint arXiv:2209.02976, 2022.

20. Li, H., Xiong, P., An, J., Wang, L. Pyramid Attention Network for Semantic Segmentation. arXiv preprint arXiv:1805.10180, 2018.

21. Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. Advances in Neural Information Processing Systems, 2020, 33, 21002-21012.

22. Li, Z., Wang, W., Shui, P. Parameter Estimation and Two-Stage Segmentation Algorithm for the Chan-Vese Model. 2006 International Conference on Image Processing, IEEE, 2006, 201-204. https://doi.org/10.1109/ICIP.2006.312455

23. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, 2117-2125. https://doi.org/10.1109/CVPR.2017.106

24. Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, 2017, 2980-2988. https://doi.org/10.1109/ICCV.2017.324

25. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A. C. SSD: Single Shot Multibox Detector. In Computer Vision-ECCV 2016, 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I 14, Springer International Publishing, 2016, 21-37. https://doi.org/10.48550/arXiv.1512.02325

26. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

27. Redmon, J., Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, 7263-7271. https://doi.org/10.1109/CVPR.2017.690

28. Redmon, J., Farhadi, A. YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767, 2018.

29. Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems, 2015, 28.

30. Rezatofighi, H., Tsoi, N., Gwak, J. Y., Sadeghian, A., Reid, I., Savarese, S. Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 658-666. https://doi.org/10.1109/CVPR.2019.00075

31. Stanulov, A., Yassine, S. A Comparative Analysis of Machine Learning Algorithms for the Purpose of Predicting Norwegian Air Passenger Traffic. International Journal of Mathematics, Statistics, and Computer Science, 2024, 2, 28-43. https://doi.org/10.59543/ijmscs.v2i.7851

32. Sun, Z., Zhu, L., Qin, S., Yu, Y., Ju, R., Li, Q. Road Surface Defect Detection Algorithm Based on YOLOv8. Electronics, 2024, 13(12), 2413. https://doi.org/10.3390/electronics13122413

33. Sunkara, R., Luo, T. No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Cham: Springer Nature Switzerland, 2022, 443-459. https://doi.org/10.1007/978-3-031-26409-2_27

34. Tan, M., Pang, R., Le, Q. V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/

CVF Conference on Computer Vision and Pattern Recognition, 2020, 10781-10790. https://doi.org/10.1109/CVPR42600.2020.01079

35. Tedeschi, A., Benedetto, F. A Real-Time Automatic Pavement Crack and Pothole Recognition System for Mobile Android-Based Devices. Advanced Engineering Informatics, 2017, 32, 11-25. https://doi.org/10.1016/j.aei.2016.12.004

36. Tong, Z., Chen, Y., Xu, Z., Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. arXiv preprint arXiv:2301.10051, 2023.

37. Torbaghan, M. E., Li, W., Metje, N., Burrow, M., Chapman, D. N., Rogers, C. D. F. Automated Detection of Cracks in Roads Using Ground Penetrating Radar. Journal of Applied Geophysics, 2020, 179, 104118. https://doi.org/10.1016/j.jappgeo.2020.104118

38. Wang, C. Y., Bochkovskiy, A., Liao, H. Y. M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, 7464-7475. https://doi.org/10.1109/CVPR52729.2023.00721

39. Wang, J., Xu, C., Yang, W., Yu, L. A Normalized Gaussian Wasserstein Distance for Tiny Object Detection. arXiv preprint arXiv:2110.13389, 2021.

40. Xu, X., Jiang, Y., Chen, W., Huang, Y., Zhang, Y., Sun, X. DAMO-YOLO: A Report on Real-Time Object Detection Design. arXiv preprint arXiv:2211.15444, 2022.

41. Zhang, C., Chang, C., Jamshidi, M. Concrete Bridge Surface Damage Detection Using a Single-Stage Detector. Computer Aided Civil and Infrastructure Engineering, 2020, 35(4), 389-409. https://doi.org/10.1111/mice.12500

42. Zhang, Y. F., Ren, W., Zhang, Z., Jis, Z., Wang, L., Tan, T. Focal and Efficient IOU Loss for Accurate Bounding Box Regression. Neurocomputing, 2022, 506, 146-157. https://doi.org/10.1016/j.neucom.2022.07.042

43. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(07), 12993-13000. https://doi.org/10.1609/aaai.v34i07.6999

44. Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., Zuo, W. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. IEEE Transactions on Cybernetics, 2021, 52(8), 8574-8586. https://doi.org/10.1109/TCYB.2021.3095305

45. Zou, Q., Cao, Y., Li, Q., Mao, Q., Wang, S. CrackTree: Automatic Crack Detection from Pavement Images. Pattern Recognition Letters, 2012, 33(3), 227-238. https://doi.org/10.1016/j.patrec.2011.11.004