IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 November 2024, accepted 11 December 2024, date of publication 16 December 2024,
date of current version 24 December 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3517632*

RESEARCH ARTICLE

# Crack Detection on Road Surfaces Based on Improved YOLOv8

**HAIYANG WU, LINGYUN KONG, AND DENGHUI LIU**
School of Electronic Information, Xijing University, Xi'an, Shaanxi 710123, China
Corresponding author: Lingyun Kong (1400100383@qq.com)

**ABSTRACT** Road defect detection is vital for road maintenance but remains challenging due to the complexity of backgrounds, low resolution, and crack similarity. This paper introduces YOLOv8-VOS(VOS means 'vanillaNet+ODConv+SEAttention'), an enhanced road crack detection algorithm that incorporates an improved Vanilla Net backbone with Squeeze-and-Excitation (SE) attention and ODConv modules. The loss function is replaced with WIoU to better balance bounding box regression. Experiments on the RDD2022 dataset demonstrate a 2% improvement in average accuracy over the original YOLOv8, achieving 53.7%. The proposed model effectively identifies road cracks in complex traffic backgrounds, contributing to safer and more efficient road maintenance.

**INDEX TERMS** Vanilla Net, YOLOv8, RDD2022, road crack detection, ODConv, road maintenance, complex background.

## I. INTRODUCTION

Highways as a crucial component of the national transportation system, play a vital role in economic and social development. As the total mileage of roads in China continues to increase, road management has gradually shifted from planning and construction to large-scale maintenance. By the end of 2021, the total length of roads in China reached 5.28 million kilometers, with 99.40% of them undergoing maintenance. This highlights the growing importance of road maintenance alongside expanding construction, while technological advancements are driving the shift toward intelligent road maintenance.

Pavement defect detection, particularly the identification and repair of cracks, is a critical aspect of road maintenance. Timely detection and repair can significantly extend the service life of roads, reduce maintenance costs, and enhance traffic safety. Early methods of crack detection relied on manual on-site inspections. However, these methods are time-consuming, labor-intensive, and often yield inconsistent and subjective results, making it challenging to ensure data accuracy and consistency, while also posing safety risks to personnel. To improve efficiency and accuracy, the development of automated pavement crack detection methods has become a significant research focus.

With the rapid advancement of computer hardware and image processing technologies, computer vision-based automatic detection techniques have been widely applied in pavement crack detection. Researchers utilize vehicle-mounted cameras to capture road surface images, which are then analyzed using computer vision algorithms to achieve fast crack detection. Compared to traditional manual methods [1], automated detection not only improves efficiency but also enhances accuracy and reduces safety risks, making it an essential tool in modern road maintenance [2].

This paper proposes a novel road detection model, YOLOv8-VOS, which incorporates the Omni-Dimensional Dynamic Convolution(ODConv) model combined with vanillaNet, replacing the original backbone network. vanillaNet, known for its simple yet effective structure, significantly reduces the model's parameter count and computational load by eliminating unnecessary shortcut operations, while maintaining both inference speed and accuracy. By integrating vanillaNet with ODConv, the backbone network meets the requirements of the neck network without modifying its structure, preserving the backbone's simplicity and efficiency. This improves the detection speed and accuracy for road surfaces. Additionally, an SEAttention module is

The associate editor coordinating the review of this manuscript and approving it for publication was Zhengmao Li.

introduced at the junction of the neck network and the detection head, enhancing the model's ability to focus on relevant channels, further boosting detection accuracy, especially for complex road surfaces. In the detection head, the loss function is replaced with wise-Intersection over Union(wise-IoU), a more precise alternative that allows the model to focus on common detection boxes while reducing the impact of special cases. This adjustment enhances the model's robustness and generalization capabilities.

## II. RELATED WORK

### A. CRACK DETECTION METHODS BASED ON TRADITIONAL DIGITAL IMAGE PROCESSING

Traditional digital image processing methods preprocess collected crack images using techniques such as smoothing and filtering to remove background noise and enhance the target area, then detect cracks based on extracted features like pixel values and texture information. The main methods include threshold segmentation, edge detection, and region growing.

#### 1) CRACK DETECTION METHODS BASED ON THRESHOLD SEGMENTATION

Threshold segmentation methods separate crack regions from background regions by setting appropriate pixel intensity thresholds. Oliveira and Correia [3] proposed a threshold algorithm combining dynamic thresholding and image entropy techniques to identify cracks. Wei [4] proposed a moving average adaptive threshold segmentation algorithm combined with Hough transform to automatically detect fine cracks. Quan et al. [5] improved Otsu's threshold segmentation method to enhance crack detection accuracy. However, threshold segmentation methods are prone to errors under varying lighting conditions and uneven crack textures, leading to poor robustness.

#### 2) CRACK DETECTION METHODS BASED ON EDGES

Edge detection methods detect crack edges in images using edge detection operators such as the Sobel operator [6], Prewitt operator [7], and Canny operator [8]. Zhao et al. [9] improved the threshold acquisition step of the Canny operator to effectively detect weak edge cracks. Ayenu-Prah and Attoh-Okine [10] combined two-dimensional empirical mode decomposition (BEMD) with Sobel edge detection to remove noise before detecting cracks. However, edge detection methods are sensitive to noise, reducing segmentation accuracy.

#### 3) CRACK DETECTION METHODS BASED ON REGIONS

Region growing methods describe internal pixel information of cracks by aggregating pixel regions with similar features based on specific criteria. Zou et al. [11] introduced a minimum spanning tree for recursive tree edge pruning, while Li et al. [12] used seed growing strategies for crack detection. These methods perform well in detecting blurred and discontinuous crack lines but lack a global perspective.

In summary, traditional digital image processing methods for crack detection have poor robustness in complex environments, limited generalizability, and require complex post-processing and parameter adjustment.

### B. CRACK DETECTION METHODS BASED ON TRADITIONAL MACHINE LEARNING

Machine learning has been widely applied in crack detection, predicting cracks by learning rules from data. Some traditional machine learning methods, such as artificial neural networks, AdaBoost algorithms, and random forest methods, have been used for crack detection. Cord and Chambon [13] proposed a supervised learning algorithm based on AdaBoost that learns the most suitable filters to describe crack texture information. Shi et al. [14] designed the Crack Forest high-performance crack detection model, addressing texture unevenness and complex topological structures in crack detection. However, traditional machine learning methods rely on manual feature extraction, requiring different methods for different detection targets, resulting in low accuracy and portability.

### C. CRACK DETECTION METHODS BASED ON DEEP LEARNING

Deep learning has shown outstanding performance in computer vision tasks, capable of automatically learning and constructing target features through hierarchical structures. Deep convolutional neural networks (CNNs) have shown excellent performance in crack detection. Unlike traditional machine learning, which requires manual feature extraction, deep learning can automatically learn features, making it more intelligent and efficient.

Zhang et al. [15] first applied supervised deep CNNs to road crack detection, achieving crack image classification but with low detection speed and accuracy. Tang et al. [16] designed a crack detection model based on Towards Real-Time Object Detection with Region Proposal Networks (Faster R-CNN), improving detection accuracy using transfer learning and multi-task enhancement methods, though at the cost of detection speed. Maedal et al. [17] proposed detection models based on Single Shot MultiBox Detector(SSD)Inception V2 and SSD MobileNet for vehicle-mounted road image defect detection, achieving acceptable accuracy and recall rates. Mandal et al. used the You Only Look Once (YOLO)v2 network to accurately identify and locate cracks at different positions, while Du et al. [18] used the YOLOv3 network to improve detection accuracy and speed.

Object recognition methods can quickly and effectively detect crack types and location information, while pixel-level crack detection methods based on deep learning also perform well. Zou et al. [19] established the DeepCrack network based on SegNet, improving detection accuracy by fusing feature maps from the encoder and decoder networks. Jenkins et al. [20] improved the fully convolution network

(U-Net), achieving precise segmentation of road cracks. Despite the excellent performance of deep CNNs in crack detection, their convolution operations' limitations only allow focus on small-scale features, lacking long-term dependency modeling. To address this, Yan and Zhang [21] enhanced model detection accuracy by adding deformable convolutions, and Jia Guohui incorporated dilated convolutions with different dilation rates to capture crack context information and used attention mechanisms to enhance global detail information in feature maps.

In conclusion, deep learning-based crack detection methods have been proven superior to traditional image processing and machine learning methods, but still have limitations. Future research will further optimize models to improve detection accuracy and robustness.

## III. ALGORITHM DESIGN

### A. INTRODUCTION TO THE YOLOv8 ALGORITHM

YOLO [22] is an end-to-end single-stage object detection model that has evolved over the years, with YOLOv8 [23] being the latest and most stable iteration with the smallest parameter count. YOLOv8 builds upon the success of its predecessors by introducing new features and improvements to enhance performance and flexibility. Its specific innovations include an entirely new backbone network, a new anchor-free detection head [24], and an improved loss function. Additionally, YOLOv8 can run on various hardware platforms, from CPUs to GPUs.

The structure of YOLOv8 is illustrated in Figure 1. The backbone and neck parts of YOLOv8 are inspired by the design philosophy of YOLOv7-ELAN [25], replacing the Concentrated-Comprehensive Convolution(C3) structure from YOLOv5 with the CSP Bottleneck with 2 Convolutions(C2F) structure, which provides richer gradient flows. Different channel numbers are adjusted for models of different scales, significantly improving performance. However, operations like "split" in the C2F module can hinder deployment on specific hardware.

In the head part, YOLOv8 makes substantial changes compared to YOLOv5, adopting the current mainstream decoupled head structure, separating the classification and detection heads. It also transitions from anchor-based to anchor-free. For loss calculation, YOLOv8 employs the TaskAlignedAssigner positive sample assignment strategy and introduces Distribution Focal Loss. In the training data augmentation phase, YOLOv8 incorporates a strategy from YOLOX, turning off mosaic [26] augmentation for the last 10 epochs, effectively enhancing precision.

### B. LIMITATIONS AND IMPROVEMENT IDEAS FOR YOLOv8

With the increasing demand for object detection and the growing diversity of its application scenarios, reducing model parameters and computational costs to enable deployment on lightweight edge computing devices is of great significance for the engineering applications and development of object detection. YOLOv8, as an excellent algorithm framework, draws on the design of several outstanding algorithms, including YOLOv5 [27], YOLOX, and YOLOv7, making it highly practical for engineering applications. However, YOLOv8 still has some issues and room for improvement.

Firstly, the replacement of the C3 structure with the C2F structure leads to deployment problems on certain hardware. The large number of convolutional and pooling layers make this computationally intensive algorithm difficult to apply on lightweight edge detection devices. Additionally, the numerous shortcut operations in the model consume a significant amount of off-chip memory bandwidth when merging features from different layers. Secondly, the distribution regression loss proposed in YOLOv8 is not always suitable for all types of object detection tasks.

To address these issues, this paper proposes replacing the loss function of YOLOv8 with Wise-IoU, which features dynamic non-monotonic Focal Mechanism (FM) [28]. This loss function can effectively reduce the contribution of easy examples to the loss value, allowing the model to focus more on difficult cases and thus improve classification performance. Additionally, by constructing a gradient gain calculation method to incorporate a focal mechanism, the model can better optimize the object detection task during training. This improvement aims to enhance the applicability of YOLOv8 on lightweight edge computing devices and improve its overall detection performance.

### C. YOLOv8-VOS

We have mainly improved the backbone network, neck network, and loss function of YOLOv8 to construct a lightweight and more efficient single-stage object detection model. The architecture of YOLOv8-VOS is shown in Figure 2. We will describe our improvements in more detail later in this section.

#### 1) IMPROVING THE VANILLA NETWORK WITH ODConv MODULE

Although complex networks perform exceptionally well, their increasing complexity poses numerous challenges for practical deployment. For instance, the shortcut operations in ResNet [29] consume significant off-chip memory bandwidth when merging features from different layers. To address this issue, Huawei Noah's Ark Lab proposed a new network architecture in 2023 called VanillaNet [39]. This network structure features a simple and elegant design while maintaining remarkable performance in visual tasks. By discarding excessive depth and shortcut operations, VanillaNet addresses complexity issues, making it highly suitable for resource-constrained environments. The specific structure of the VanillaNet module is shown in Figure 3.

Most state-of-the-art (SOTA) classification network structures typically consist of three parts: a stem block that converts the input three-channel image into multiple channels and performs down-sampling, a main body for feature

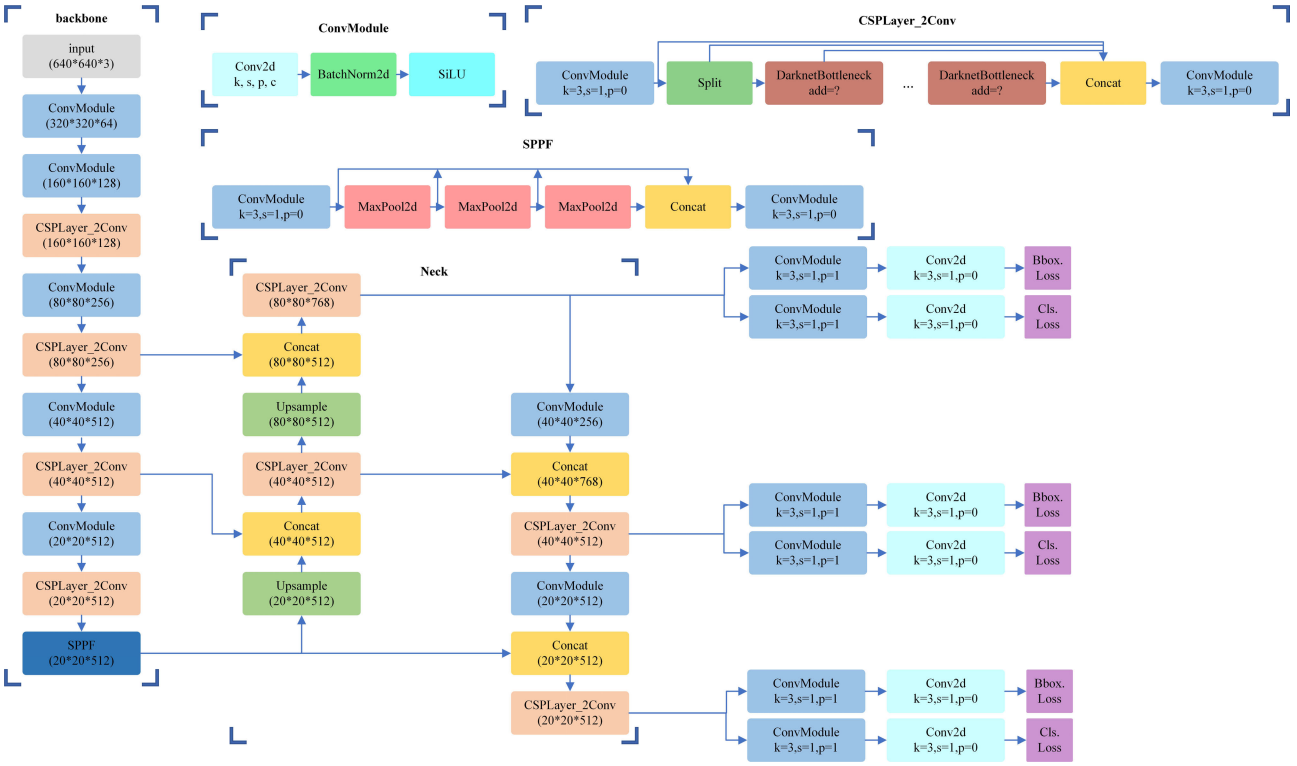


**FIGURE 1.** YOLOv8 network architecture diagram.

extraction, and a fully connected layer for outputting the classification result. The main body usually includes four stages, each containing multiple identical blocks, where the feature map resolution is decreased by reducing the number of channels after each stage. The main difference between networks lies in the design of these blocks. VanillaNet also follows this popular design architecture, but the key difference is that each stage contains only a single layer, thus constructing an extremely simple network.

Despite the simple structure and few layers of VanillaNet, its weak nonlinearity limits its performance. To address this issue, the authors propose the following methods:

1. Deep Training Strategy

The core idea of the deep training strategy is to train two convolutional layers and an activation function in the early stages of training, instead of just one convolutional layer. As training progresses, the activation function gradually becomes an identity mapping. At the end of the training, the two convolutional layers can be merged into one through structural re-parameterization, reducing inference time, as shown in Figure 3. This approach significantly enhances the nonlinearity and overall performance of VanillaNet while maintaining its simplicity.

For an activation function $A(x)$, we combine it with an identity mapping as follows (1):

$$A'(x) = (1 - \lambda)A(x) + \lambda x \tag{1}$$

where $\lambda$ is a hyperparameter that balances the nonlinearity of the modified activation function $A'(x)$. Suppose the current training iteration and the total training iterations are $e$ and $E$, respectively, we can set $\lambda = \frac{e}{E}$. Thus, at the beginning of training i.e., when $e = 0$, $A'(x) = A(x)$, meaning the network has strong nonlinearity. When training is complete, $A'(x) = x$, indicating that no activation function exists within the two convolutional layers, which can then be merged into one convolutional layer through structural re-parameterization.

2. Stacking Known Activation Functions

The poor performance of simple and shallow networks is mainly due to insufficient nonlinearity. Two methods to enhance nonlinearity are: stacking nonlinear activation layers and enhancing the nonlinearity of each activation layer. VanillaNet chooses the latter by improving the nonlinearity of each activation layer through weighted stacking. The specific formula is as follows (2):

$$A_s(x) = \sum_{i=1}^{n} a_i A(x + b_i) \tag{2}$$

where $n$ represents the number of stacked activation functions, and $a_i$ and $b_i$ are the weights and biases of each activation function, respectively. This stacking method significantly enhances the nonlinearity of the activation function.

In subsequent experiments, it was found that using the VanillaNet network structure alone, due to the shallow depth of the backbone network, the combination of shallow and deep features when connecting to the neck network not only fails to improve overall detection accuracy but also

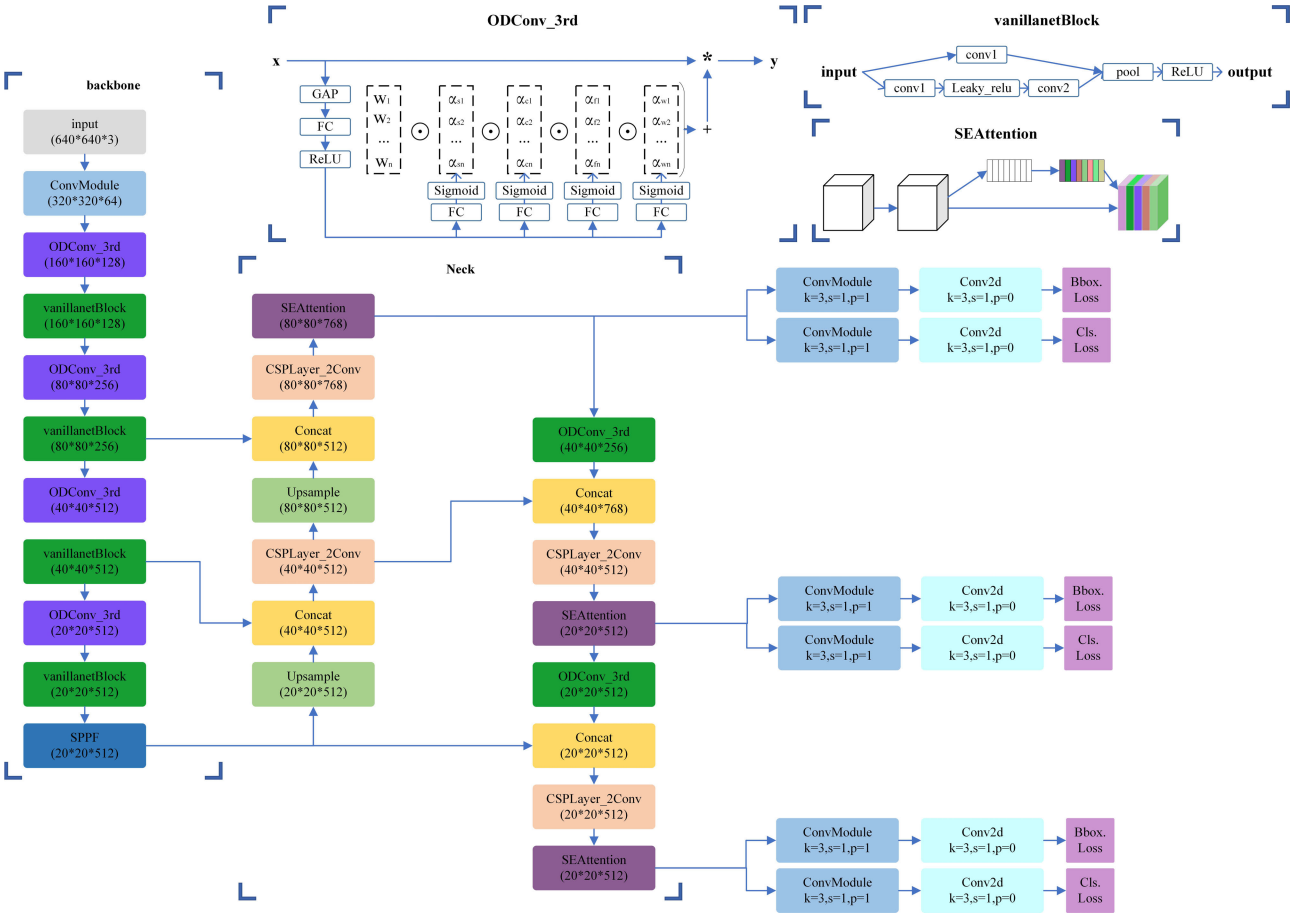


**FIGURE 2.** YOLOv8-VOS network architecture diagram.

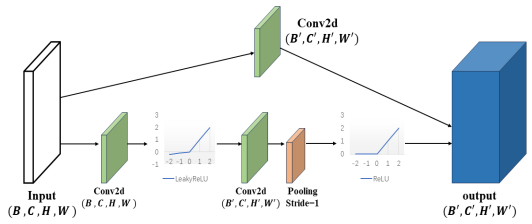


**FIGURE 3.** Internal structure diagram of the VanillaNet module.

affects the original accuracy of deep features. Therefore, this paper proposes using the ODConv module [31] to deepen the network depth of VanillaNet. On one hand, this makes the backbone network structure more compatible with the subsequent neck network structure; on the other hand, it enables the model to learn deeper features, thereby improving overall performance.

Traditional convolution operations use a static convolution kernel, which is independent of the input sample and is general across the dataset. However, the performance of convolutional neural networks [32] often relies on expanding the network's width, depth, and resolution. In previous attention mechanism models, the attention mechanism was mainly applied to feature maps, such as weighting different channels of the feature map (e.g., SE Net [33]) or weighting different spatial locations of the feature map (spatial attention [34]).

The core idea of dynamic convolution is to linearly weight multiple convolution kernels, with these weights related to the input data, making the dynamic convolution input-dependent. Specifically, this dynamic convolution operation can be expressed by the following formula (3):

$$y = (\alpha_{w1} W_1 + \cdots + \alpha_{wn} W_n) * x \tag{3}$$

where $W_n$ represents different convolution kernels, and $\alpha_{wn}$ are the weights corresponding to each convolution kernel. Different convolution kernels are used for different inputs, and these different convolution kernels are attention-weighted.

In this study, we replace traditional convolutional kernels with ODConv, a more versatile and elegantly designed form of dynamic convolution. The key distinction between ODConv and regular dynamic convolution lies in the calculation of $\alpha_{wi}$ using the strategy $\pi_{wi}(x)$ and the implementation of dynamic convolution layers. These differences result in improvements in model accuracy, size, and inference

efficiency. By adopting ODConv, we significantly enhance model performance, achieving high accuracy while optimizing computational efficiency and resource utilization.

In ODConv research, the authors highlight the critical role of attention mechanisms within dynamic convolutions. Designing more effective attention mechanisms can strike a better balance between model accuracy and size. ODConv extends the dynamic properties of dynamic convolution across various dimensions, including spatial dimensions, input channels, and output channels, hence earning its name as omni-dimensional dynamic convolution.

Using a parallel strategy, ODConv employs multi-dimensional attention mechanisms across the spatial dimensions of the convolutional kernel, learning complementary attention patterns to enhance model expressiveness. Figure 4 illustrates the network structure of ODConv. Through this design, ODConv not only comprehensively expands on dynamic characteristics but also achieves optimized performance in terms of accuracy and efficiency.

Based on the above discussion, ODConv introduces a multi-dimensional attention mechanism through a parallel strategy across the four dimensions of the convolutional kernel space. Continuing from the definition of dynamic convolution, ODConv can be described by the following formula (4):

$$y = \left( \begin{matrix} \alpha_{w1} \odot \alpha_{f1} \odot \alpha_{c1} \odot \alpha_{s1} \odot W_1 + \\ \cdots + \alpha_{wn} \odot \alpha_{fn} \odot \alpha_{cn} \odot \alpha_{sn} \odot W_n \end{matrix} \right) * x \tag{4}$$

ODConv can be embedded into various convolutional neural network architectures. Extensive experiments on datasets like ImageNet and MS-COCO demonstrate that ODConv significantly improves model accuracy across popular CNN backbone architectures, from lightweight to large-scale models. This versatility makes ODConv a powerful tool for enhancing the performance of CNN models of different scales.

### 2) ENHANCING NECK NETWORK WITH SEAttention MODULE

Channel features in convolutional neural networks represent the outputs generated by each convolutional kernel, where each channel corresponds to a specific kernel's response to the input. These features capture abstract information from different aspects of the input data, such as textures, colors, edges, etc., responsible for extracting and representing information at different levels throughout the network.

The channel attention mechanism [35] enhances the ability to capture these channel features effectively within convolutional neural networks. It improves the model's focus on different channel features, enabling more efficient learning and utilization of input data information.

The SEAttention mechanism module consists of three parts: Squeeze, Excitation, and Scale, as depicted in Figure 5. Here's the specific implementation process:

1. Squeeze: Globally average pools the input feature map to reduce each channel's feature values to a global vector, aiming to capture global information for each channel.

2. Excitation: Consists of two fully connected layers, a ReLU function, and a Softmax activation. It reduces dimensionality and then expands it, generating a weight vector via a Sigmoid function to ensure a sum of 1. The weight vector is formulated as (5):

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma\left(g(\mathbf{z}, \mathbf{W})\right) = \sigma\left(\mathbf{W}_2 \delta(\mathbf{W}_1 \mathbf{z})\right) \tag{5}$$

where $\delta$ is the ReLU function, $\sigma$ is the Sigmoid function, $W_1$ and $W_2$ are weights of the fully connected layers, and $\mathbf{z}$ is the input feature vector.

3. Scale: Multiplies the channel attention weights obtained in the previous step with the original input feature map $U$, adjusting each channel's feature values to emphasize important channels and suppress less important ones:

$$x = F_{scale}(U, s) = U \cdot s \tag{6}$$

In this study, to enhance the model's capability in capturing channel features, we integrate the SEAttention module into the connection part between the neck and head networks. Experimental results show that incorporating the SEAttention module improves the model's mAP by 1.5%.

### 3) USE WISE-IoU TO REPLACE THE ORIGINAL LOSS FUNCTION

In this study, the original loss function used by YOLOv8 was Generalized Intersection over Union(GIoU) [36], proposed in 2019. While IoU loss function measures the overlap of bounding boxes in a ratio manner, it doesn't fully optimize bounding box regression effectively in object detection tasks. Additionally, L1 norm is sensitive to object sizes, and IoU cannot directly optimize cases where there is no overlap between boxes. Therefore, GIoU introduces the concept of Generalized IoU, which addresses these issues by introducing a minimum enclosing shape C. This allows the predicted box to move towards the ground truth box even in non-overlapping scenarios, thus solving the issue of loss when there is no overlap. GIoU not only focuses on overlapping regions but also considers non-overlapping areas, providing a more comprehensive reflection of the intersection of two boxes within the closed shape region.

To address the issues associated with GIoU, this paper replaces the loss function in the network with WIoU. As a boundary box regression loss, WIoU employs a dynamic non-monotonic mechanism and establishes a reasonable gradient allocation strategy, which can mitigate the occurrence of large or harmful gradients from extreme samples. This strategy enables the model to focus more on samples of ordinary quality, thereby enhancing the model's generalization ability and overall performance.

The WIoUv3 used in this study is derived from WIoUv1, combined with a non-monotonic focal coefficient. WIoUv1 incorporates two layers of attention mechanisms: distance attention based on distance metrics, where the ordinary

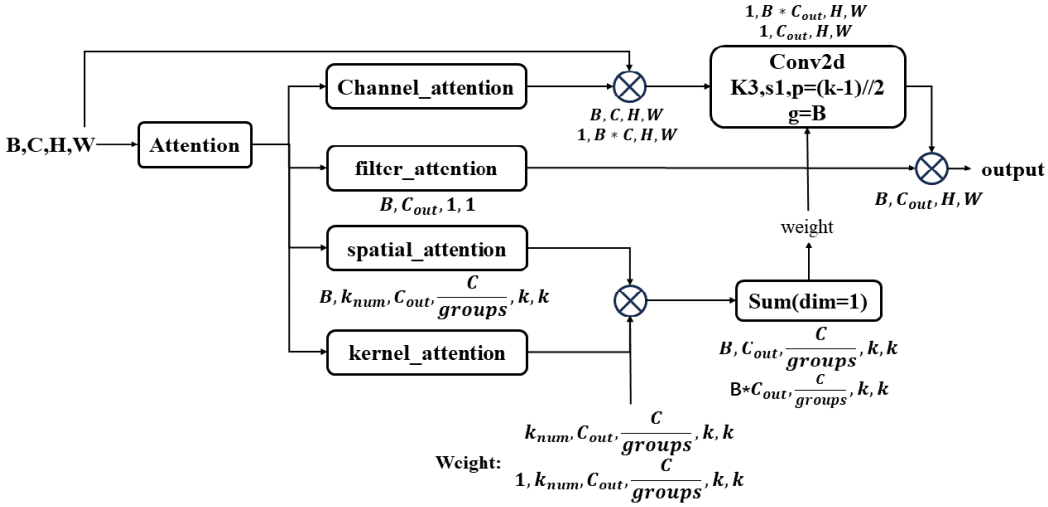


**FIGURE 4.** Schematic diagram of ODConv internal structure.

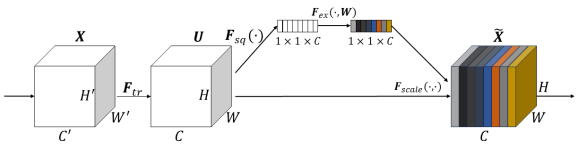

**FIGURE 5.** SEAttention feature processing flow.

loss function is denoted as $L_{IoU}$, and the metric attention mechanism is represented as $R_{WIoU}$. The loss function for WIoUv1 can be expressed as follows (7) (8) (9):

$$\mathcal{L}_{IoU} = 1 - IoU \tag{7}$$

$$\mathcal{R}_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*}\right) \tag{8}$$

$$\mathrm{L_{WIoUv1}} = \mathrm{R_{WIoU}} \times \mathrm{L_{IoU}} \mathrm{L_{WIoUv1}} = \mathrm{R_{WIoU}} \times \mathrm{L_{IoU}} \tag{9}$$

We define the outlier degree $\beta$ as a measure of anchor box quality, expressed as (10):

$$\beta = \frac{\mathrm{L^*_{IoU}}}{\mathrm{L_{IoU}}} \in [0, +\infty) \tag{10}$$

Using $\beta$, we construct a non-monotonic focal coefficient to ensure the model allocates more attention to anchor boxes of ordinary quality. This non-monotonic coefficient, denoted as $r$, can be represented as (11):

$$r = \frac{\beta}{\delta \alpha^{\beta - \delta}} \tag{11}$$

Combining the equations (10),(11), the loss function for WIoUv3 can be derived as follows:

$$L_{WIoUv3} = \frac{\beta}{\delta \alpha^{(\beta - \delta)}} \times L_{WIoUv1} \tag{12}$$

Here, when $\beta = \delta$, $\delta$ ensures $r = 1$. When the anchor box's outlierliness meets $\beta = C$, it receives the highest gradient boost. Since $L_{IoU}$ is dynamic, the standard for dividing anchor box quality is also dynamic, enabling WIoUv3 to dynamically allocate gradient boosts that best suit the current situation.

## IV. DATASET DESCRIPTION

The dataset used in this paper is from RDD2022, which includes a total of 47,420 road images from six countries (Japan, India, Czech Republic, Norway, USA, and China) [37]. The number of road damage labels from each country is shown in Figure 6. These images contain annotations for over 55,000 road damage instances. The dataset captures four types of road damage: longitudinal cracks, transverse cracks, alligator cracks, and potholes. This dataset is an extended version of the RDD2020 dataset [38] and was used for the Crowd Sensing-based Road Damage Detection Challenge (CRDDC2022). The CRDDC2022 challenge invited researchers worldwide to propose solutions for automatic road damage detection across multiple countries. Researchers in computer vision and machine learning can also use this dataset to evaluate the performance of different algorithms in similar image applications.

### A. ORIGIN AND DEVELOPMENT OF THE RDD DATASET

The RDD2018 dataset was proposed in 2018, containing 9,053 road images and 15,435 road damage instances. The RDD2018 dataset recorded eight types of road damage information and was used for the Road Damage Detection Challenge organized by the IEEE Big Data Cup in 2018. A total of 59 teams from 14 countries participated in this challenge. Participants also improved the annotation files of the RDD2018 dataset.

In response to these suggestions and the imbalance of road damage categories found in the RDD2018 dataset, the authors proposed an extended version of the original

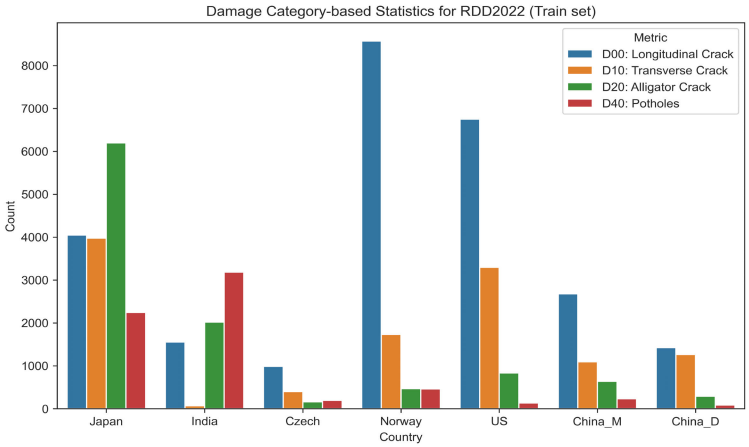


**FIGURE 6.** The number of four different types of injuries in the RDD2022 dataset for each country.

dataset, RDD2019, by correcting the annotations of the 2018 dataset and using Generative Adversarial Networks (GANs) to extend its content. This dataset contains 13,133 images and 30,989 road damage instances. The extended dataset helps to improve the accuracy of road damage detection and classification models.

In 2020, Arya et al. attempted to detect road damage outside of Japan using models trained on RDD2019 and conducted experiments using datasets from India and the Czech Republic. The results showed that the model's performance significantly declined when applied to roads outside of Japan. Therefore, the authors proposed a new dataset, RDD2020, which combined newly collected data from India and the Czech Republic with data from RDD2019. With the increase in the number of images and damage instances in RDD2020, the types of road damage considered were also updated.

Since the data in RDD2018 and RDD2019 came from a single country, the damage categories defined in the ''Japanese Road Maintenance and Repair Guide'' were used. However, in RDD2020, due to the involvement of multiple countries, various road damage standards needed to be considered. Given the significant differences in standards for evaluating road marker degradation (such as intersections or blurred white lines) between countries, these categories were excluded from the RDD2020 dataset. Ultimately, the RDD2020 dataset included four types of damage: longitudinal cracks, transverse cracks, alligator cracks, and potholes.

### B. INTRODUCTION TO RDD2022

Analysis by Arya et al. indicated that adding data from other countries could improve the model's ability to detect road damage in any country. This analysis and the success of GRDDC2020 inspired the introduction of RDD2022, which aims to address road damage detection across multiple countries, including India, Japan, the Czech Republic, Norway, China, and the USA.

The RDD2022 dataset adopts the well-known PASCAL Visual Object Classes (VOC) dataset format, suitable for developing new deep convolutional neural network architectures or modifying existing architectures to enhance network performance. Additionally, this dataset can be used to train, validate, and test algorithms for automatic identification of road damage in six countries.

## V. EXPERIMENTS AND RESULTS

The experimental setup is as follows: the operating system is Ubuntu 22.04, the deep learning framework is Pytorch, and the hardware configuration includes an NVIDIA GeForce RTX 3060 (with 8GB of memory), an Intel i7-12700H processor, and 16GB of RAM. Initial input image size is set to 640 $\times$ 640 pixels. The model training is conducted for 300 epochs with a batch size of 16. The initial learning rate is set to 0.01, using a cosine annealing learning rate schedule. Momentum for learning rate adjustment is set to the default value of 0.937, and weight decay coefficient is 0.0005. Default values are used for warm-up time and warm-up momentum (0.8). The optimizer chosen is Stochastic Gradient Descent (SGD). Data augmentation strategy aligns with the original YOLOV8n model.

### A. EVALUATION METRICS

Algorithm evaluation is primarily divided into two aspects: computational cost and accuracy. Computational cost is measured primarily by the number of parameters and GFLOPs. Typically, fewer parameters and GFLOPs indicate lower demands on hardware computing resources and performance.

Accuracy evaluation involves several key metrics:

1. Precision:

$$Precision = \frac{tp}{tp + fp} \tag{13}$$

where *tp* is the number of true positives correctly identified, and *fp* is the number of false positives (negative samples incorrectly identified as positive).

2. Recall:

$$Recall = \frac{tp}{tp + fn} \tag{14}$$

where *fn* is the number of false negatives (positive samples incorrectly identified as negative).

3. F1 Score:

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{15}$$

The F1 score is the harmonic mean of precision and recall, used to balance these two metrics.

4. Average Precision (AP):

$$\mathrm{AP} = \int_0^1 P(R)\mathrm{dR} \tag{16}$$

AP measures the performance of a classifier by calculating the area under the precision-recall curve.

5. mean Average Precision (mAP):

$$\mathrm{mAP} = \frac{\sum_{i=1}^{K} \mathrm{AP}_i}{K} \tag{17}$$

where $K$ is the total number of detected object categories, and $AP_i$ is the average precision for the i-th category.

These metrics collectively provide a comprehensive assessment of model performance. Combined with an analysis of computational costs, they help determine the suitability of the model for different application scenarios. Optimizing for higher accuracy metrics and lower computational costs ensures excellent performance even in resource-constrained environments.

### B. ABLATION EXPERIMENTS

To evaluate the effectiveness of optimizing various components of our model, we designed a series of ablation experiments to validate the effects of our model optimizations.

Firstly, we conducted ablation experiments targeting the backbone network. We used the improved VanillaNet as the backbone network and named it YOLOv8-Vanilla. Throughout all subsequent experiments, we used YOLOv8n as the baseline model. The experimental results indicate that after replacing the backbone network, there was a significant reduction in model parameters and GFLOPs, while also showing an improvement in mAP@0.5 metric. This suggests that our model is not only more lightweight but also shows improved performance. The experimental results are shown in Table 1.

From Table 1, it can be observed that compared to YOLOv8n, the improved YOLOv8 Vanilla model shows significant improvements in multiple aspects. Specifically, the model's parameter count reduced by 15.6%, GFLOPs reduced by 28.1%. Moreover, the mAP@0.5 metric increased by 0.5%, and the F1 score improved by 5%. This indicates that compared to the original model, the improved model is not only more compact in size but also shows significant improvements in inference speed and accuracy.

**TABLE 1.** Experimental results of backbone network ablation.

| models | params | GFLOPs | mAP@0.5 | F1-score |
|---|---|---|---|---|
| yolov8n | 3011628 | 8.2G | 0.516 | 0.52 |
| yolov8n-vanilla | **2541332** | **5.9G** | **0.521** | **0.57** |

**TABLE 2.** Results of neck network ablation experiment.

| models | params | GFLOPs | mAP@0.5 | F1-score |
|---|---|---|---|---|
| baseline | 2541332 | 5.9G | 0.521 | 0.57 |
| baseline ODConv | 2554824 | 5.6G | 0.518 | 0.52 |
| baseline ODConv SEAttention | 2555720 | 5.6G | **0.531** | **0.59** |
| baseline SEAttention | 2542228 | 5.9G | 0.519 | 0.53 |

Next, we conducted ablation experiments on the neck network to further validate the effectiveness of each optimization module. The ablation experiments primarily targeted the ODConv and SEAttention modules. Specifically, we designed two ablation models: one using only the ODConv module, named Baseline ODConv, and another using both the ODConv and SEAttention modules, named Baseline ODConv SEAttention. Through these ablation experiments, we aim to explore the contribution and impact of each module on the overall model performance. The specific data and results of the ablation experiments are presented in Table 2.

From Table 2, it can be observed that by simultaneously applying the ODConv and SEAttention modules, despite an increase in parameter count by 14390, and a slight decrease in GFLOPs by 0.3G, the model's performance has been significantly enhanced. Specifically, mAP@0.5 improved by 1%, and the F1 score also increased by 2%. However, when ODConv or SEAttention modules were applied individually, the improvement was lower compared to the baseline model. This indicates that the proposed enhancement approach in this study significantly enhances model performance when both modules are used together, whereas using either module alone does not yield significant improvements.

Further analysis of the experimental results reveals a synergistic interaction between the ODConv and SEAttention modules. The ODConv module enhances the model's expressive power across different feature dimensions by extending the application of dynamic convolutions. Meanwhile, the SEAttention module boosts the model's ability to capture crucial channel features, thereby increasing its focus on important characteristics. Their combination not only optimizes the model's architecture but also significantly enhances detection accuracy and robustness without a notable increase in computational complexity.

In summary, this paper validates through ablation experiments the effectiveness of the ODConv and SEAttention modules in enhancing the performance of the YOLOv8 model. Particularly, their concurrent application

**TABLE 3.** Experimental results of loss function ablation.

| models | params | GFLOPs | mAP@0.5 | F1-score |
|---|---|---|---|---|
| baseline | 2555720 | 5.6G | 0.531 | 0.59 |
| baseline WIoU | 2555720 | 5.6G | **0.537** | **0.62** |

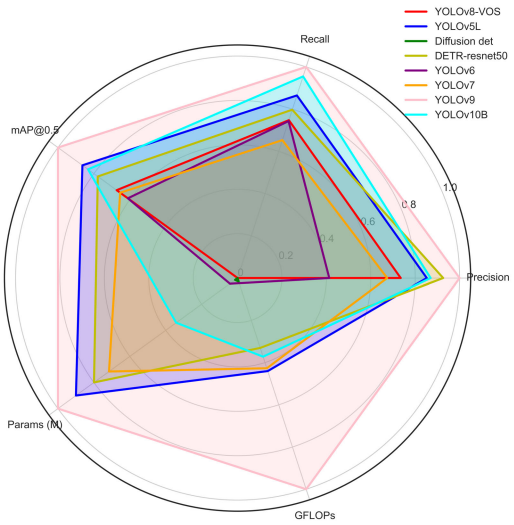


**FIGURE 7.** Radar chart with multiple algorithm parameters.

demonstrates substantial performance gains. These improvements are crucial for achieving more efficient and accurate object detection. To validate the effectiveness of the WIoU loss function proposed in this paper, we designed a ablation experiment. Keeping the backbone and neck network structures unchanged, the model using the original loss function is named baseline, while the model using the WIoU loss function is named baseline WIoU. The results of the ablation experiment are shown in Table 3.

The experimental results indicate that without introducing additional parameters, there was a 0.4% improvement in mAP. Compared to increasing feature quantity and expanding receptive fields, modifying the loss function to better suit the task is undoubtedly a simpler and more effective approach. Specifically, WIoU demonstrated improved robustness and generalization when handling data containing low-quality instances. The introduction of WIoU, utilizing a distance attention mechanism through distance metrics, allows the model to focus more on challenging examples during optimization, thereby enhancing overall detection accuracy and classification performance.

Through comparing the experimental results of baseline and baseline WIoU, it is evident that the WIoU loss function exhibits significant advantages in enhancing object detection tasks. These experiments validate the effectiveness of the proposed loss function replacement strategy in this paper, providing theoretical support and experimental evidence for further optimizing and improving the performance of object detection models.

Overall, we conducted incremental ablation experiments on all modules and algorithms, with the experimental results shown in Table 4.

Based on the data analysis from Table 4, YOLOv8-VOS in this study represents a significantly improved version of YOLOv8n and serves as the baseline model for the ablation experiments. The experiments encompass eight different cases where various modules are combined to generate distinct models. YOLOv8-VOS, optimized by applying all modules and algorithms, represents the final model. Compared to the baseline model, YOLOv8-VOS shows a notable improvement in overall performance.

From Table 4, it can be observed that using only the vanilla module resulted in an mAP@0.5 lower than the average level. Due to the parameter disparity, vanilla net-9 was replaced with the lighter vanilla net-5 as the backbone network in our study. However, from a network structure perspective, the first and second layers connecting the backbone and neck networks extracted very shallow features or none at all, leading to poor performance in Cases 1, 3, 4, and 7. After improving the backbone network with ODConv, the enhanced network (Case 2) demonstrated a 0.5% increase in mAP, confirming the necessity of improving the vanilla module.

Furthermore, replacing the original model's loss function with the more effective WIoU continued to improve the mAP@0.5 without increasing the parameter count of the original model. Ultimately, the model stabilized at an mAP@0.5 of 53.7%, which is 2.1% higher than the original model.

In recent years, the field of object detection has seen many representative and innovative algorithms emerge. To accurately compare the optimization effects of each part and evaluate the performance of our YOLOv8-VOS in road damage detection tasks, we selected several representative algorithms for experimentation. Detailed experimental data is listed in Table 5.

From the above table 5, it is evident that YOLOv8-VOS demonstrates significant advantages in terms of parameter count and GFLOPs. In Diffusion DET [39], although the model's parameter count and computational load (GFLOPs) show slight differences compared to models like YOLOv6, YOLOv7, YOLOv10B [40], and YOLOv5L, YOLOv8-VOS surpasses these models in accuracy through fewer parameters, faster inference speed, and higher accuracy.

We have plotted the parameters of all these models on a radar chart for a more intuitive comparison, as shown in Figure 7. Despite YOLOv8-VOS having 13 times fewer parameters and 17 times fewer GFLOPs than YOLOv7, it still improves accuracy by 0.5%. Compared to models like YOLOv5L, DETR-resnet50, and YOLOv10B, which have more than ten times the parameters and computational load of YOLOv8-VOS, YOLOv8-VOS manages to keep the mAP difference within 5%. In contrast, YOLOv9, with

**TABLE 4.** Gradual ablation experiment.

| | backbone | | neck | | loss | mAP@0.5 | params |
|---|---|---|---|---|---|---|---|
| model | vanilla | ODConv | SEAttention | ODConv | WIoU | | |
| baseline | | | | | | 0.516 | **3011628** |
| case1 | ✓ | | | | | 0.466 | 1732476 |
| case2 | ✓ | ✓ | | | | 0.521 | 2541332 |
| case3 | ✓ | | ✓ | | | 0.454 | 1733372 |
| case4 | ✓ | | | ✓ | | 0.473 | 1745968 |
| case5 | ✓ | ✓ | ✓ | | | 0.519 | 2542228 |
| case6 | ✓ | ✓ | | ✓ | | 0.518 | 2554824 |
| case7 | ✓ | | ✓ | ✓ | | 0.472 | 1746864 |
| case8 | ✓ | ✓ | ✓ | ✓ | | 0.531 | 2555720 |
| YOLOv8-VOS | ✓ | ✓ | ✓ | ✓ | ✓ | **0.537** | **2555720** |

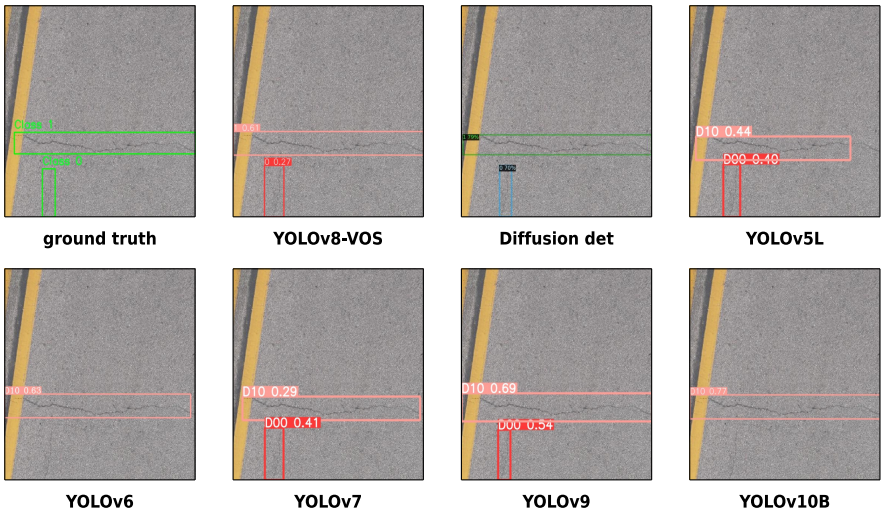


**FIGURE 8.** Comparison of multiple algorithms for lateral crack damage detection.image from china drone.

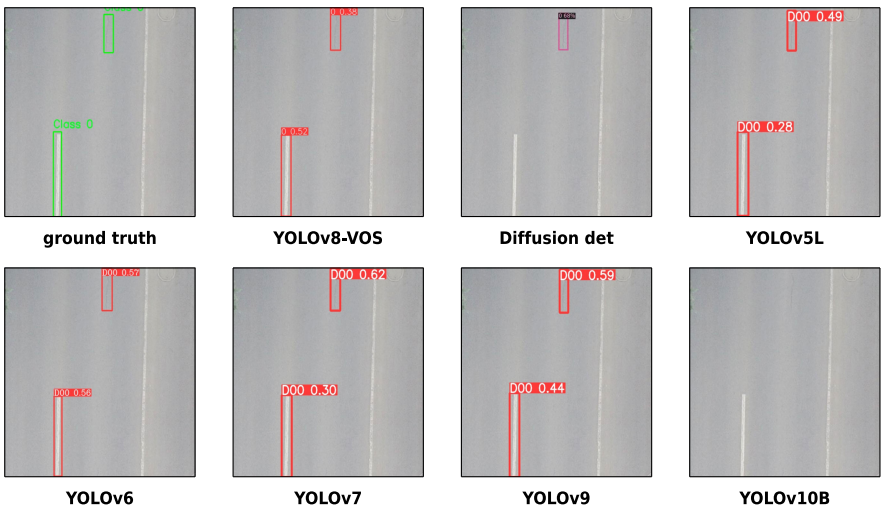


**FIGURE 9.** Comparison of multiple algorithms for foggy weather detection.image from china drone.

20 times more parameters and 40 times more GFLOPs than YOLOv8-VOS, outperforms YOLOv8-VOS in mAP by 8.4%, a difference that is understandable given the disparity in computational resources.

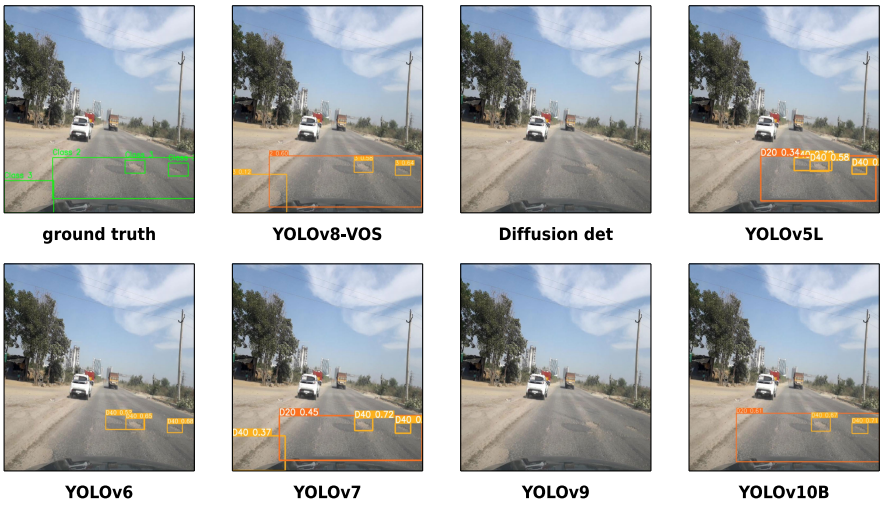


**FIGURE 10.** Comparison of multiple algorithms for multi-task small object detection.image from india.

**TABLE 5.** Comparison of multiple algorithms.

| models | params | GFLOPs | Precision | Recall | mAP@0.5 |
|---|---|---|---|---|---|
| YOLOv8-VOS | **2555720** | **5.6G** | 0.595 | 0.533 | 0.537 |
| YOLOv5L | 46154449 | 108.3G | 0.622 | 0.559 | 0.586 |
| Diffusion det | 3234964 | 9.5G | 0.426 | 0.367 | 0.364 |
| DETR-resnet50 | 41280266 | 82.7G | 0.639 | 0.544 | 0.564 |
| YOLOv6 | 4632134 | 11.3G | 0.521 | 0.532 | 0.521 |
| YOLOv7 | 37232405 | 105.2G | 0.581 | 0.512 | 0.532 |
| YOLOv9 | 51006520 | 238.9G | **0.656** | **0.589** | **0.621** |
| YOLOv10B | 19108600 | 92.4G | 0.626 | 0.579 | 0.578 |

## VI. COMPARATIVE VISUALIZATION EXPERIMENT RESULTS

To better demonstrate the performance advantage of our YOLOv8 VOS under complex road conditions, we compared YOLOv8 VOS with several other algorithms in various scenarios. Using image labels as ground truth, we annotated the true positions and classes of damages in the images. We employed YOLOv8 VOS and other algorithms for detection, displaying the positions, damage classes, and probabilities of detected boxes in the images.

Overall, stability of the model is crucial in complex and changing road conditions. Algorithms that perform well in such scenarios have better prospects for practical engineering applications. In this experiment, we selected four different scenarios and analyzed the comparative experiments in the following sections.

### A. LATERAL CRACK DAMAGE DETECTION

Detecting lateral crack damage correctly is more challenging than ordinary crack damage, as shown in Figure 8. YOLOv6 and YOLOv10 exhibited instances of missed detections. While other algorithms had similar detection regions, YOLOv8 VOS demonstrated superior accuracy. This indicates that our optimizations significantly enhance the algorithm's ability to capture contextual information at longer distances.

### B. FOGGY WEATHER DETECTION

Detecting road damage becomes exceptionally challenging under dim lighting and low visibility conditions, especially in foggy environments. In such situations, the impact of traffic lane markings on crack-type damage detection is further amplified, posing significant challenges for feature extraction and learning capabilities of prediction algorithms. Therefore, this experiment selected an image with fog and visible lane markings as a sample, and the experimental results are shown in Figure 9.

Among the evaluated models, only YOLOv7 demonstrated optimal performance, accurately detecting road damage. In contrast, DiffusionDET and YOLOv10 exhibited instances of missed detections in foggy conditions. Notably, YOLOv7 showed superior prediction accuracy under these conditions compared to other models. In comparison, the algorithm proposed in this paper showed slightly lower detection accuracy under foggy conditions compared to YOLOv7 and YOLOv9, despite its model parameters being only one-tenth of these two models. This indicates the high reliability and stability of YOLOV8-VOS in foggy conditions, demonstrating its potential for excellent performance in complex visual environments.

### C. MULTI-TASK SMALL OBJECT DETECTION

If various types of cracks and minor damages appear simultaneously on road surfaces, it could severely impact driving safety. Therefore, we conducted experiments using images containing multiple small target pit damages, as shown in Figure 10. Compared to the other six models, the

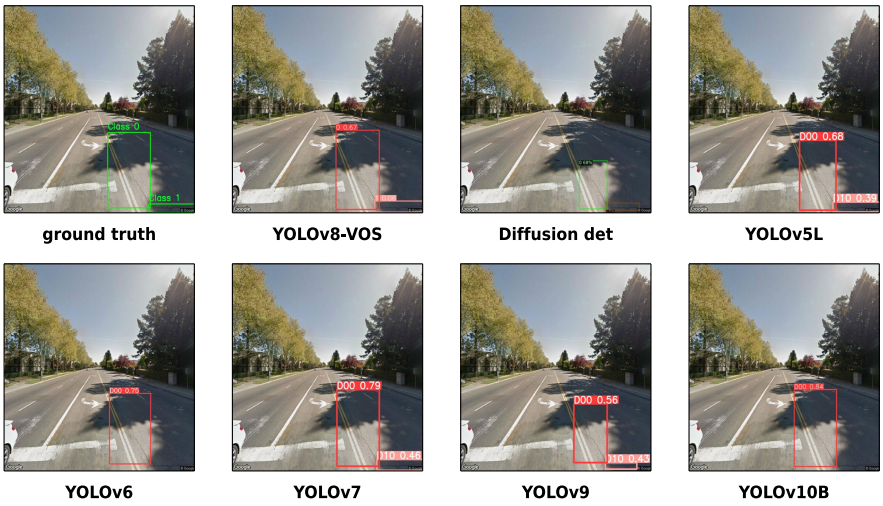


**FIGURE 11.** Comparison of multiple algorithms for shadow detection.image from united states.

YOLOv8-VOS proposed in this paper exhibited no false positives or misses during the detection process and showed high confidence, significantly outperforming the other models. This experiment demonstrates that YOLOv8-VOS achieves high precision and excellent robustness in multi-task small object detection.

### D. SHADOW DETECTION

In practical detection scenarios, the influence of shadows on detection accuracy must be considered. For instance, shadows cast by plants and buildings along the sides of roads may appear over cracks or potholes on the road surface. Therefore, we conducted detection experiments using images containing two types of cracks with shadows, as shown in Figure 11. Among all detection algorithms, the YOLOv8-VOS proposed in this paper, along with YOLOv5, YOLOv7, and YOLOv9, did not miss any detections, unlike the other three models which have significantly more parameters than YOLOv8-VOS and much slower inference speeds. The study results demonstrate that the proposed model exhibits high robustness against road shadows in bright light environments.

## VII. SUMMARY

Based on an in-depth analysis of existing road damage detection methods, we propose a lightweight and efficient road damage detection algorithm, YOLOv8-VOS, aimed at addressing the high computational costs and insufficient accuracy of current algorithms. We trained this algorithm on the RDD2022 dataset to detect various complex road damages. In terms of network structure optimization, we first introduced the ODConv module to enhance VanillaNet and designed a more optimized backbone network. Subsequently, we replaced the standard convolution modules in the neck network with ODConv modules and added SEAttention modules at the junction between the neck and head networks. By optimizing the model's parameter count and GFLOPS, we further improved the model's performance and inference speed. Additionally, replacing the original GIOU loss function with the superior WIOU effectively enhanced the model's convergence speed and performance.

The experimental phase was divided into two parts: incremental ablation experiments and multi-model comparison experiments. Ablation experiment results showed that compared to YOLOv8, our YOLOv8-VOS increased mAP by 2.1% while reducing model parameters and GFLOPS. Compared to high-computation models, we demonstrated significant computational efficiency advantages while maintaining accuracy. Compared to recent excellent lightweight models, YOLOv8-VOS exhibited notable accuracy advantages with fewer parameters. Furthermore, visual detection results demonstrated YOLOv8-VOS's superior robustness and reliability in complex road environments.

This study provides important theoretical references for lightweight road damage detection algorithms and opens up new possibilities for deploying road damage detection methods on edge computing devices.

Future research can expand in several directions. Firstly, optimizing and expanding datasets to enhance model generality and adaptability. Secondly, with advancements in hardware technology, exploring further optimizations in network structure. Lastly, exploring lightweight object detection and post-processing methods to further expand the application scope of object detection technology.

## APPENDIX A
## DATA AVAILABILITY

The dataset for this experiment can be obtained from https://github.com/sekilab/RoadDamageDetector

## APPENDIX B
## CONFLICTS OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of this article.

## REFERENCES

[1] K. C. P. Wang, "Designs and implementations of automated systems for pavement surface distress survey," *J. Infrastructure Syst.*, vol. 6, no. 1, pp. 24–32, Mar. 2000.

[2] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14531–14544, 2020.

[3] H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *Proc. 17th Eur. Signal Process. Conf.*, Aug. 2009, pp. 622–626.

[4] C. Wei, "Automatic detection method for tiny cracks and micro gray difference cracks based on adaptive threshold," *China Foreign Highway*, vol. 39, no. 1, pp. 58–63, 2019.

[5] Y. Quan, J. Sun, Y. Zhang, and H. Zhang, "The method of the road surface crack detection by the improved Otsu threshold," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Tianjin, China, Aug. 2019, pp. 1615–1620.

[6] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE J. Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, Apr. 1988.

[7] W. Dong and Z. Shisheng, "Color image recognition method based on the Prewitt operator," in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, Wuhan, China, 2008, pp. 170–173.

[8] L. Er-Sen, Z. Shu-Long, Z. Bao-Shan, Z. Yong, X. Chao-Gui, and S. Li-Hua, "An adaptive edge-detection method based on the Canny operator," in *Proc. Int. Conf. Environ. Sci. Inf. Appl. Technol.*, vol. 1, Kunming, China, Jul. 2009, pp. 465–469.

[9] H. Zhao, G. Qin, and X. Wang, "Improvement of Canny algorithm based on pavement edge detection," in *Proc. 3rd Int. Congr. Image Signal Process.*, vol. 2, Yantai, China, Oct. 2010, pp. 964–967.

[10] A. Ayenu-Prah and N. Attoh-Okine, "Evaluating pavement cracks with bidimensional empirical mode decomposition," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, pp. 1–9, Dec. 2008.

[11] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 227–238, Feb. 2012.

[12] Q. Li, Q. Zou, D. Zhang, and Q. Mao, "FoSA: F* seed-growing approach for crack-line detection from pavement images," *Image Vis. Comput.*, vol. 29, no. 12, pp. 861–872, Nov. 2011.

[13] A. Cord and S. Chambon, "Automatic road defect detection by textural pattern recognition based on AdaBoost," *Computer-Aided Civil Infrastructure Eng.*, vol. 27, no. 4, pp. 244–259, Apr. 2012.

[14] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016.

[15] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3708–3712.

[16] J. Tang, Y. Mao, J. Wang, and L. Wang, "Multi-task enhanced dam crack image detection based on faster R-CNN," in *Proc. IEEE 4th Int. Conf. Image, Vis. Comput. (ICIVC)*, Xiamen, China, Jul. 2019, pp. 336–340.

[17] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 12, pp. 1127–1141, Dec. 2018.

[18] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, "Pavement distress detection and classification based on YOLO network," *Int. J. Pavement Eng.*, vol. 22, no. 13, pp. 1659–1672, Nov. 2021.

[19] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "DeepCrack: Learning hierarchical convolutional features for crack detection," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1498–1512, Mar. 2019.

[20] M. D. Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy, and G. Morison, "A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Roma, Italy, Sep. 2018, pp. 2120–2124.

[21] K. Yan and Z. Zhang, "Automated asphalt highway pavement crack detection based on deformable single shot multi-box detector under a complex environment," *IEEE Access*, vol. 9, pp. 150925–150938, 2021.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[23] G. Jocher and A. Chaurasia. (Jan. 2023). *Ultralytics YOLO Version 8.0.0*. [Online]. Available: https://github.com/ultralytics/ultralytics

[24] K. Duan, L. Xie, H. Qi, S. Bai, Q. Huang, and Q. Tian, "Location-sensitive visual recognition with cross-IOU loss," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2021, pp. 4567–4576.

[25] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[26] W. Hao and S. Zhili, "Improved mosaic: Algorithms for more complex images," *J. Phys., Conf. Ser.*, vol. 1684, no. 1, Nov. 2020, Art. no. 012094.

[27] G. Jocher, "Ultralytics/YOLOv5: v3.1—Bug fixes and performance improvements," *Zenodo*, Oct. 2020, doi: 10.5281/zenodo.4154370.

[28] Z. Tong, Y. Chen, Z. Xu, and R. Yu, "Wise-IoU: Bounding box regression loss with dynamic focusing mechanism," 2023, *arXiv:2301.10051*.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] H. Chen, Y. Wang, J. Guo, and D. Tao, "VanillaNet: The power of minimalism in deep learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Jan. 2023.

[31] C. Li, A. Zhou, and A. Yao, "Omni-dimensional dynamic convolution," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Jan. 2022. [Online]. Available: https://openreview.net/forum?id=DmpCfq6Mg39

[32] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.

[33] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020, doi: 10.1109/TPAMI.2019.2913372.

[34] X. Zhang, C. Liu, D. Yang, T. Song, Y. Ye, K. Li, and Y. Song, "RFAConv: Innovating spatial attention and standard convolutional operation," 2023, *arXiv:2304.03198*.

[35] J. Hu and L. Shen, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.

[36] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[37] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, "Crowdsensing-based road damage detection challenge (CRDDC'2022)," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2022, pp. 6378–6386.

[38] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2020: An annotated image dataset for automatic road damage detection using deep learning," *Data Brief*, vol. 36, Jun. 2021, Art. no. 107133.

[39] S. Chen, P. Sun, Y. Song, and P. Luo, "DiffusionDet: Diffusion model for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Paris, France, Oct. 2023, pp. 19773–19786.

[40] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-time end-to-end object detection," 2024, *arXiv:2405.14458*.

**HAIYANG WU** received the degree from Chongqing College of Mobile Communication, in 2022. He is currently pursuing the master's degree in electronic information engineering with the School of Electronic Information, Xijing University. They have authored a Chinese article and a software work. Their research interests include diffusion models, generative models, and computer vision.

**LINGYUN KONG** received the Ph.D. degree in engineering.

He is currently working as a Professor, the Head of the Control Engineering Discipline, and a Supervisor of master's students. He has published more than 50 papers in domestic and international journals, including more than ten SCI-indexed papers. He has led the completion of 12 provincial and municipal-level scientific research projects and eight enterprise-sponsored projects. He has also been recognized as an outstanding technology talent in private education at the municipal level. His primary research interests include robotics, artificial intelligence, and nonlinear control theory and applications. He has long been dedicated to research in robotics control technology, intelligent perception, natural language processing, and nonlinear chaotic dynamics. He was awarded the provincial Academic and Technical Leader title in 2015, recognized as an outstanding educational manager at the provincial level in 2016, and honored with the Shaanxi Province Teacher Ethics Model title in 2020.

**DENGHUI LIU** received the degree from Henan University of Engineering, in 2022. He is currently pursuing the master's degree in control engineering with the School of Electronic Information, Xijing University. They have written a Chinese article. Their main research interest includes fault diagnosis. They won the first prize at the provincial level in the 2024 National Graduate Electronic Design Competition.