Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 8879622, 17 pages
https://doi.org/10.1155/2023/8879622

Research Article

LE-YOLOv5: A Lightweight and Efficient Road Damage Detection Algorithm Based on Improved YOLOv5

Zhuo Diao,[1] **Xianfu Huang**,[2,3] **Han Liu**,[4] **and Zhanwei Liu**[1]

[1]*School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China*
[2]*State Key Laboratory of Nonlinear Mechanics (LNM), Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China*
[3]*School of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, China*
[4]*Beijing Institute of Structure and Environment Engineering, Beijing 100076, China*

Correspondence should be addressed to Zhanwei Liu; liuzw@bit.edu.cn

Received 26 March 2023; Revised 27 August 2023; Accepted 12 September 2023; Published 28 September 2023

Academic Editor: Said El Kafhali

Road damage detection is very important for road safety and timely repair. The previous detection methods mainly rely on humans or large machines, which are costly and inefficient. Existing algorithms are computationally expensive and difficult to arrange in edge detection devices. To solve this problem, we propose a lightweight and efficient road damage detection algorithm LE-YOLOv5 based on YOLOv5. We propose a global shuffle attention module to improve the shortcomings of the SE attention module in MobileNetV3, which in turn builds a better backbone feature extraction network. It greatly reduces the parameters and GFLOPS of the model while increasing the computational speed. To construct a simple and efficient neck network, a lightweight hybrid convolution is introduced into the neck network to replace the standard convolution. Meanwhile, we introduce the lightweight coordinate attention module into the cross-stage partial network module that was designed using the one-time aggregation method. Specifically, we propose a parameter-free attentional feature fusion (PAFF) module, which significantly enhances the model's ability to capture contextual information at a long distance by guiding and enhancing correlation learning between the channel direction and spatial direction without introducing additional parameters. The K-means clustering algorithm is used to make the anchor boxes more suitable for the dataset. Finally, we use a label smoothing algorithm to improve the generalization ability of the model. The experimental results show that the LE-YOLOv5 proposed in this document can stably and effectively detect road damage. Compared to YOLOv5s, LE-YOLOv5 reduces the parameters by 52.6% and reduces the GFLOPS by 57.0%. However, notably, the mean average precision (mAP) of our model improves by 5.3%. This means that LE-YOLOv5 is much more lightweight while still providing excellent performance. We set up visualization experiments for multialgorithm comparative detection in a variety of complex road environments. The experimental results show that LE-YOLOv5 exhibits excellent robustness and reliability in complex road environments.

1. Introduction

Road construction has long been a very important part of infrastructure development. Whether it is a regular road, highway, or airport pavement, damage such as cracks and depressions can occur after a long period of service. This damage can be accelerated when affected by rain and snow, further damaging the road. There is no doubt that for motorists, pavement damage creates uncertainty and many unsafe factors. At the same time, pavement damage largely increases the cost of operating and maintaining infrastructure. Detecting pavement damage in a timely manner can effectively maintain traffic safety and provide the basis for subsequent pavement rehabilitation and care. Therefore, rapid pavement damage detection has a very wide range of engineering applications.

In recent years, hardware has been gradually upgraded and iterated, deep learning has developed rapidly, deep learning-based target detection has also been developed, and convolutional neural networks are widely used in tasks such

as image classification, object detection, and semantic segmentation [1]. Current deep learning-based target detection is mainly divided into two categories. The first category is based on candidate region target detection, including region-based convolutional neural networks (R-CNN) [2], spatial pyramid pooling in deep convolutional networks (SPP-NET) [3], region-based fully convolutional networks (R-FCN) [4], and mask region-based convolutional neural networks (mask R-CNN) [5]. These algorithms have obvious advantages and disadvantages. Each of them can meet high accuracy requirements, but the detection speed is slow and the real-time performance is poor. Consequently, it is difficult to apply it to scenes that require high detection speed. The second category is regression-based target detection, including the you only look once (YOLO) [6, 7] series, single shot MultiBox detector (SSD) [8], and RetinaNet [9]. These algorithms are slightly less accurate than the first category but have a very fast detection speed.

The application of deep learning-based target detection in road damage detection has also evolved, with Arya et al. [10] proposing RDD-2020 in 2020, a large-scale road damage dataset with over 26,620 images from multiple countries, and the IEEE 2020 Global Big Data Challenge applying RDD-2020. Various road damage detection models were proposed by scholars from various countries in the competition. Maeda et al. [11] proposed using progressive growth of generative adversarial networks (PG-GANs) [12] and Poisson blending methods to generate real road damage images as new training data to improve the accuracy of pothole detection. Hedge et al. [13] proposed an improved model based on YOLOv5x using test-time data augmentation (TTA) [14], ensemble prediction, and ensemble models to achieve the title of the IEEE 2020 Global Big Data Challenge. Although the abovementioned studies have made some contributions to the road damage detection task, they all share a common problem. The competition does not require the detection speed and only uses accuracy as the judging criterion, which leads to a large number of model parameters in the abovementioned studies, resulting in a high computational cost. Thus, these algorithms are not applicable to some lightweight edge computing devices. Therefore, many scholars have also conducted research in the direction of lightweight detection, and Shim et al. [15] proposed a detection model with feature extraction through hierarchical neural networks and training prediction using multiloss function weighted soft voting in 2021, which achieved good results on their dataset. Wan et al. [16] proposed a lightweight LRDD-YOLOv5 in 2022 by replacing the backbone network as well as the loss function to achieve further lightweighting on top of YOLOv5s. The abovementioned research study provides a reference for lightweight road damage detection models, but there is still room for further optimization of the lightweight degree and detection accuracy of the models.

In this paper, we propose a lightweight and efficient road damage detection algorithm LE-YOLOv5 based on YOLOv5. In addition, it reduces the number of model parameters and improves the average accuracy. Overall, our study has four main contributions:

(1) In this paper, we propose the global shuffle attention module (GSAM), which focuses on global feature information while using channel shuffling instead of convolution to enhance interchannel feature information exchange. It is used to fill the defects of the SE-block in MobileNetV3 to construct a new backbone feature extraction network for the LE-YOLOv5 algorithm.

(2) We analyze the flaws of the C3 module in YOLOv5 for the detection task in this article. The lightweight coordinate attention module is introduced into the cross-stage partial network to design the VCACSP module.

(3) We propose a parameter-free attentional feature fusion (PAFF) module, which improves the utilization of information from multilateral features and further improves the accuracy without introducing additional parameters.

(4) We analyze the RDD-2022 dataset and introduce the K-means clustering algorithm to recalculate the initial anchor boxes and use the label smoothing algorithm to enhance the generalizability of the model.

The remainder of this study is organized as follows: in Section 2, we analyze the limitations of the YOLOv5 algorithm and briefly describe the product improvement perspective of this paper. In Section 3, the lightweight and efficient road damage detection algorithm LE-YOLOv5 is introduced and expanded into four parts for a detailed description. Section 4 shows the results of the ablation experiments and comparative analysis. In Section 5, we show visualization results and a comparative analysis of detection in a variety of complex road environments. Finally, Section 6 presents a summary of our study and an outlook for the future.

2. Deficiencies and Improvement Ideas of YOLOv5

Currently, the demand for target detection is increasing, and the application scenarios are becoming more diverse. Therefore, reducing the number of model parameters and computational cost so that the model can be applied to more lightweight edge computing devices is of great practical value for the engineering application and development of target detection.

YOLO is an end-to-end single-stage target detection model that has evolved over the years. YOLOv5, based on the PyTorch framework, stands out for its excellent inference speed and structural scalability, as shown in Figure 1. However, it still has some problems. Its detection accuracy for targets of different scales is not sufficiently stable, especially in the detection of small targets that are prone to error detection and omission detection situations. The large number of convolutional and pooling layers results in a computationally intensive algorithm that is difficult to apply to lightweight edge detection devices. At the same time, the lack of ability to capture and fuse features leaves room for further improvement in its detection accuracy. In this paper, we focus on the backbone feature extraction network and the feature fusion

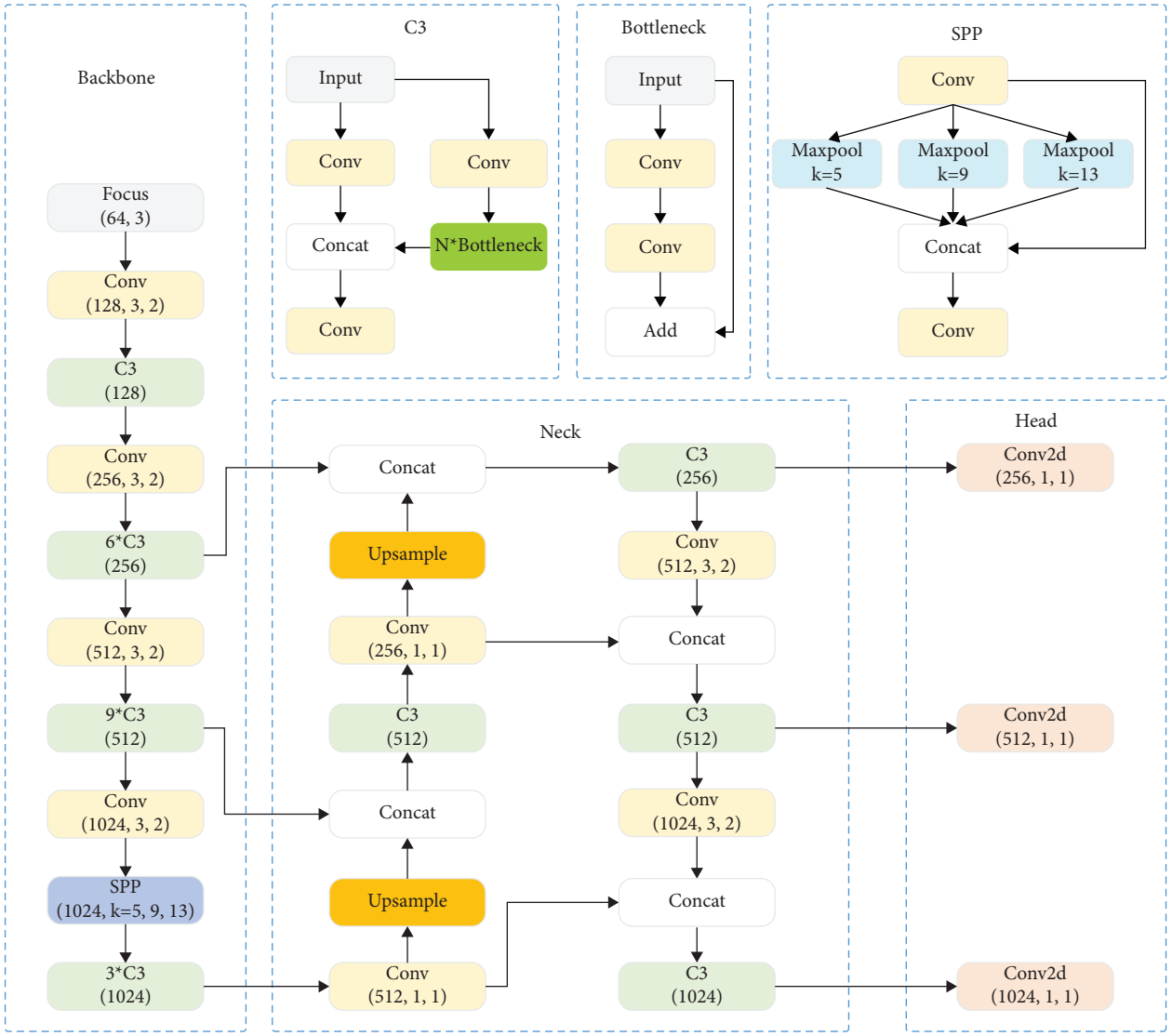


Figure 1: YOLOv5 network architecture.

module to make the algorithm perform better in coping with multiscale targets and long-range contextual information and improve its detection accuracy. To have a wider range of applications in lightweight detection, we have lightweighted the algorithm from multiple perspectives while ensuring the accuracy of the algorithm.

3. LE-YOLOv5

We mainly improve the YOLOv5 backbone network and the neck network to build a lightweight and better performance detection model. The architecture of LE-YOLOv5 is shown in Figure 2. We will describe our improvements in more detail later in this section.

3.1. Improving the MobileNetV3 Network with Global Shuffle Attention Module. The characteristics of less computation and faster computation make lightweight networks have a wider range of applications. MobileNet [17] can be considered one of the leaders. In 2019, Howard et al. proposed the MobileNetV3 [18] network, which inherits the deep separable convolution of the V1 version and the inverse residual structure with the linear bottleneck of the V2 [19] version. MobileNetV3 parameters are obtained by network architecture searching (NAS) [20], with excellent performance and speed.

However, the SE attention mechanism in MobileNetV3 is still deficient. It focuses only on channel feature information and ignores spatial feature information. However, the spatial attention mechanism is equally important and is seen as an adaptive screening process for key spatial areas [21]. The combination of channel attention and spatial attention can yield more comprehensive and reliable attention information for more rational guidance on computational resource allocation [22]. Furthermore, the SE attention module only considers attention in the channel dimension and cannot capture attention in the spatial dimension [23], which is detrimental to the learning ability of the network.

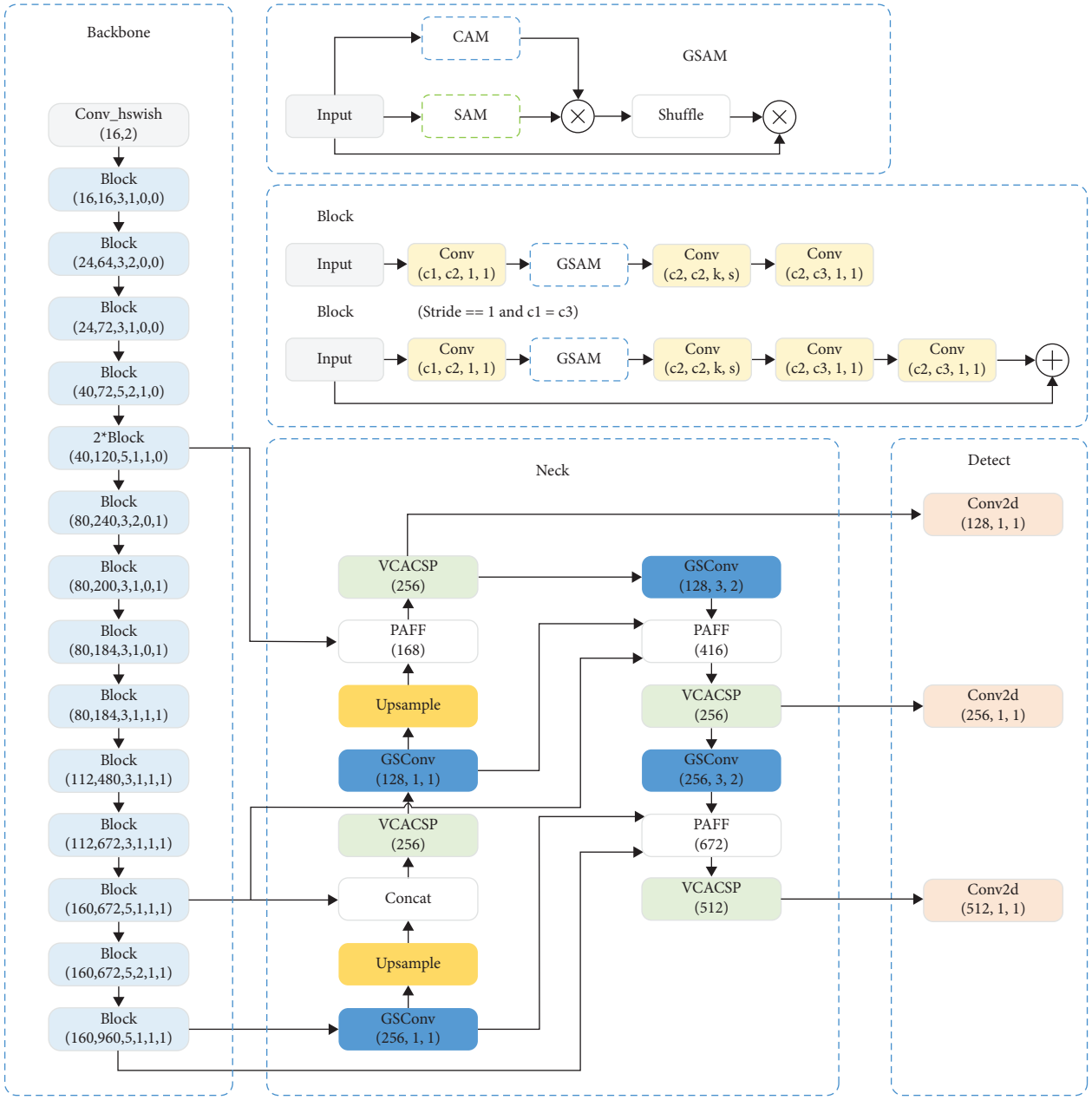


Figure 2: LE-YOLOv5 network architecture.

Therefore, it is crucial to rationalize how to combine the channel attention mechanism with the spatial attention mechanism. We do not want to have too much computation in the attention module, and we also need to avoid the attention mechanism that focuses too much on local features. For this purpose, we propose a global shuffle attention module (GSAM). The module focuses on both channel and spatial attention. Unlike traditional hybrid attention, it uses parallel channel and spatial attention mechanisms to accelerate computational efficiency and improve performance. Moreover, we use channel shuffling to avoid the computational effort brought by convolutional layers while enhancing the information exchange between channels. Its schematic structure is shown in Figure 3.

For the given input feature map $F \in R^{H\times W\times C}$, we compute spatial and channel attention independently as parallel branches. Elementwise multiplication is performed after resizing both $\mathrm{Ms}(F) \in R^{H\times W}$ and $\mathrm{Mc}(F) \in R^{C}$ obtained from $R^{H\times W\times C}$. Then, we obtain a 3D attention map by channel shuffling. The GSAM attention map $M(F) \in R^{H\times W\times C}$ can be computed as follows:

$$M(F) = S(\mathrm{Ms}(F) \otimes \mathrm{Mc}(F)), \tag{1}$$

where $\otimes$ represents elementwise multiplication and $S()$ represents channel shuffling. Finally, the GSAM attention map is multiplied element by element with the input tensor $F \in R^{H\times W\times C}$ to obtain the output tensor $F' \in R^{H\times W\times C}$. Its calculation can be expressed as follows:

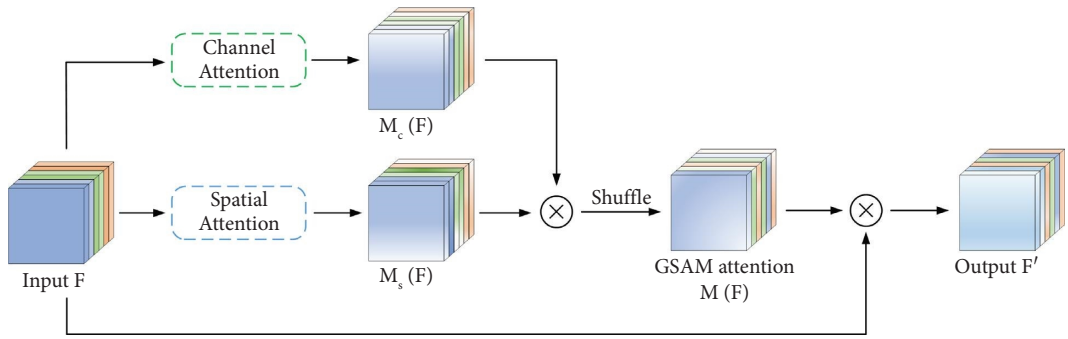


Figure 3: The structure of the global shuffle attention module.

$$F' = \sigma(M(F) \otimes F), \tag{2}$$

where σ is a sigmoid function.

As shown in Figure 4, in the channel attention module, we use both average pooling and maximum pooling. It preserves the overall information of the image and reduces the impact of spatial details while still focusing on the most important information. After feeding the two descriptors into the shared multilayer perceptron (MLP) network, the channel attention $\mathrm{Mc}(F)$ is obtained by elementwise summation and activation functions. It is computed as follows:

$$\mathrm{Mc}(F) = \sigma(\mathrm{MLP}(\mathrm{AvgPool}(F)) + \mathrm{MLP}(\mathrm{MaxPool}(F))), \tag{3}$$

where σ is a sigmoid function.

As shown in Figure 5, spatial attention similarly conveys the idea of both maximum and average pooling. Considering that spatial attention involves different directions, we use concatenation to stitch them together. Then, the spatial attention $\mathrm{Ms}(F)$ is obtained by convolutional and activation functions. It is computed as follows:

$$\mathrm{Ms}(F) = \sigma(f([\mathrm{AvgPool}(F), \mathrm{MLP}(\mathrm{MaxPool}(F))])), \tag{4}$$

where σ is a sigmoid function, $[,]$ denotes a concatenation operation, and f denotes a convolutional operation.

We improve MobileNetV3 with the proposed GSAM to get a better backbone feature extraction network. It concurrently focuses on global and local features, which improve the feature capturing and learning ability of the backbone network.

3.2. Hybrid Convolution and Cross-Stage Partial Network of Coordinate Attention. Li et al. [25] designed the GSConv module based on depthwise separable convolution combined with the channel shuffle operation in ShuffleNet [26]. The structure of the GSConv module is shown in Figure 6. Branch learning of features applies the idea of dimensionality enhancement followed by dimensionality reduction. Finally, information exchange between channels is carried out by concatenation and shuffling.

Compared to the backbone network, in the neck network, the feature maps finally output by the backbone network are smaller in size and have more channels, the feature transformation and circulation are softer, and the semantic information of the features is better retained. Therefore, LE-YOLOv5 arranges the GSConv module in the neck network to reduce the number of calculations. The feature map input to the GSConv module is first passed through a standard convolution with a step size and convolutional kernel size of 1 to reshape the number of channels into half the number of output channels. The resulting feature map is fed into a depthwise separable convolution (DWConv) with convolutional kernel size 5 and step size 1. The output feature map is then concatenated with the standard convolutional output. Finally, the feature map from the previous step is subjected to channel shuffling operation to get the final output feature maps.

Module C3 in the former neck network is an important learning module. However, it is computationally intensive and insensitive to small targets. The research study in [27] has shown that attention mechanisms can help networks improve their ability to learn about small targets. Meanwhile, the study in [25] has shown that the network responds better to attention after deploying the GSConv module. Therefore, we designed the VCACSP module by introducing the coordinate attention module into VoV-GSCSP$_1$, which reduces the computational effort while enhancing the sensitivity to small targets through attention guidance. The VCACSP means visual coordinate attention cross-stage partial network. Its structure is schematically shown in Figure 7.

The structure of the lightweight coordinate attention module is shown in Figure 8. It first divides channel attention into two one-dimensional feature encodings along the width and height of the feature map, aggregating features along their respective spatial directions, with the following equation:

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j). \tag{5}$$

Specifically, for the upper-level input features, the features are encoded along the height and width with pooling kernels of size $(H, 1)$ and $(1, W)$ with the following respective formulas:

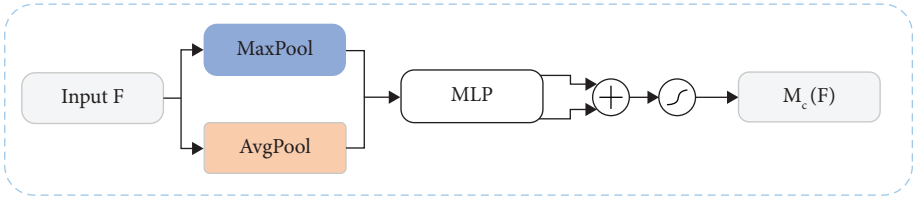


Figure 4: The structure of the channel attention module.

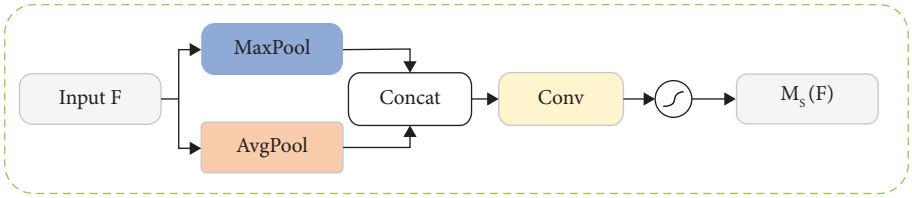


Figure 5: The structure of the spatial attention module [24].

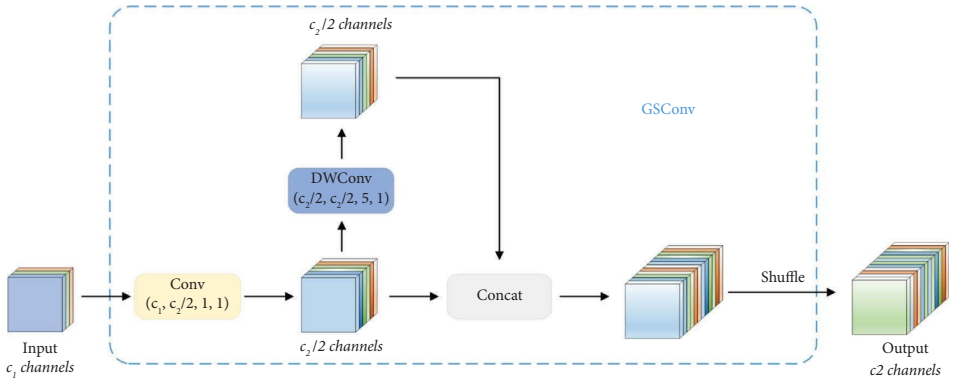


Figure 6: The structure of the GSConv module.

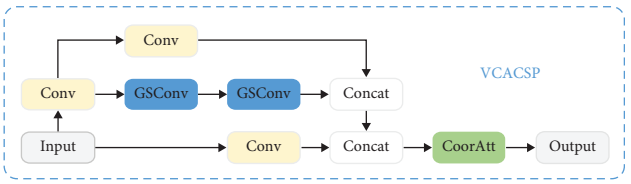


Figure 7: The structure of the VCACSP module. CoorAtt in the figure means coordinate attention.

$$Z_c^h(h) = \frac{1}{W}\sum_{0\le i\le W} x_c(h,i),$$
$$Z_c^w(w) = \frac{1}{H}\sum_{0\le j\le W} x_c(j,w). \tag{6}$$

The encoded feature map in both width and height directions is stitched together and fed into the convolutional module with a shared convolutional kernel 1×1 for dimensionality reduction to the initial C/r, where r is used to control the reduction rate, and then the batch normalized feature map F_1 is fed into the activation function of the sigmoid [28] to obtain the feature map f, which is given in the following equation:

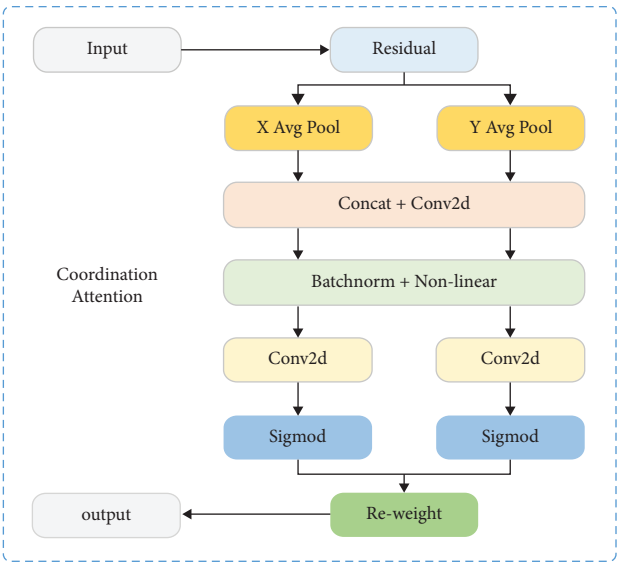


Figure 8: The structure of the coordinate attention.

$$f = \delta\left(F_1\left(\left[Z_c^h, Z_c^w\right]\right)\right), \tag{7}$$

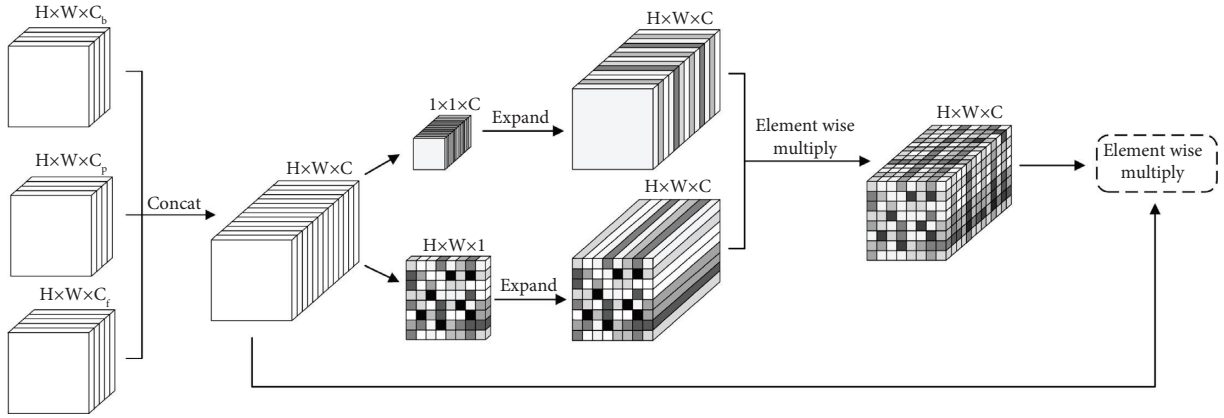


Figure 9: The structure of the PAFF module. The grayscale differences in the figure represent different attention weights.

where the function [,] is the concatenation operation along the spatial dimension, δ is the nonlinear activation, and f is the intermediate feature mapping that encodes the spatial information in the horizontal and vertical directions. Subsequently, by decomposing and reshaping, we obtain the following g^h and g^w, which are expressed by equations (8) and (9).

$$g^h = \sigma\left(F_h\left(f^h\right)\right), \tag{8}$$

$$g^w = \sigma\left(F_w\left(f^w\right)\right), \tag{9}$$

where f^h and f^w denote tensors decomposed along the space and F_h and F_w denote the reshaped tensors. Finally, the final feature map with attention weights in the width and height directions is obtained by multiplying and weighting the original feature map with the following equation:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j). \tag{10}$$

Through rationalization and construction, LE-YOLOv5's neck network is lightweight while eliminating the shortcomings of insensitivity to small targets.

3.3. Parameter-Free Attention Feature Fusion Module. Feature fusion has always been a very important part of target recognition networks, and common feature fusion structures include feature pyramid network (FPN) [29] and path aggregation network (PANet), for instance, segmentation [30]. In the backbone network, the input image is continuously downsampled as the network deepens, and in this process, since the pixels occupied by small targets in the input image are much smaller than those of large targets, the feature information of large targets is more easily retained and that of small targets is easily lost in the downsampling process of the backbone network. To solve this problem, a combined neck structure of the FPN and PANet is designed in YOLOv5. However, the feature fusion part still has more rooms for improvement in terms of the combined performance of inference speed and accuracy.

The neck structure of the YOLOv5 combined FPN and PANet is adapted to its underlying backbone network. After we replace the backbone network with improved MobileNetV3, the available features with more channel sizes are contained in the same detection layer. At the same time, from the perspective of lightweight, we do not want to introduce additional parameters in the feature fusion part, so we propose a parameter-free attention feature fusion (PAFF) module, and the schematic diagram of the PAFF module is shown as Figure 9.

The PAFF module is mainly divided into two parts: multilateral feature splicing in the front end and a parameter-free attention module in the back end. First, after replacing the backbone network, more available features of the channel size are included in the same level features, so in the feature splicing part, we use add-edge processing to add the feature splicing part originally from the FPN layer and PANet part to the features extracted from the corresponding backbone network at the same level to obtain a feature block with a richer channel scale. It contains the feature information from the three sides of the backbone network, FPN layer, and PANet part, which greatly improves the feature information utilization between the backbone network and the neck network. The three-side feature splicing is as follows, where F is the spliced midsegment feature map, F_f is the feature map from the FPN layer, F_p is the feature map from the PANet part, F_b is the feature map from the backbone network at the same detection layer, and [,] means concatenating the feature map.

$$F = \left[F_f, F_p, F_b\right]. \tag{11}$$

At the same time, after obtaining richer feature blocks on the channel scale, it is especially important to filter out the feature information we want the network to notice among the rich feature information. To solve such problems, various types of attention mechanisms have also been proposed by numerous scholars. However, from the perspective of lightweighting, we would like to solve this problem without adding additional parameters, so we introduce the parameter-free attention feature fusion module.

A feature map $F \in R^{H\times W\times C}$ is first decomposed into channel attention $A_c \in R^{1\times 1\times C}$ and spatial attention $A_s \in R^{H\times W\times 1}$, where spatial attention is obtained by averaging the pooling of spatial features along the channel

direction, and the mean value of each spatial element $x_{H\times W} \in R^C$ can be calculated as follows:

$$A_s(x_{H\times W}) = \frac{1}{C}\sum_{i=1}^{C} x_{H\times W}(i). \tag{12}$$

By averaging along the channels, the dimensionality is reduced, and a spatial feature weight map is generated, where each element represents the average across channels. As a result, high-activation spatial regions are emphasized, while low-activation regions are suppressed, thus highlighting locations with detected features.

While channel attention is obtained by averaging the pooling of channel features along the two spatial directions of width and height, the average value of each channel element $y_c \in R^{H\times W}$ can be calculated as follows:

$$A_c(y_c) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} y_c(i,j). \tag{13}$$

Then, the expansion of the dimensionality reduction direction is carried out, and the corresponding elements of the two expanded feature block matrices are multiplied to obtain the final output feature block $H \times W \times C$ with additional weights for each feature unit $1 \times 1 \times 1$. The overall process can be summarized as follows, where $\otimes$ means elementwise multiplication [31].

$$F_o = \sigma(A_s \otimes A_c) \otimes F. \tag{14}$$

By improving the efficiency of feature utilization, the PAFF module significantly improves the model's ability to learn features. At the same time, the PAFF module significantly enhances the model's ability to capture contextual information [32] at a distance by guiding and enhancing correlation learning between the channel direction and spatial direction.

3.4. K-Means Clustering and Label Smoothing. In the target detection training process, our model needs to learn the location and size of the target along with the target category. The target instances of the same category have similar aspect ratios. Therefore, we can prepare several anchor boxes with higher probability ratios as benchmarks, which greatly reduce the difficulty of model learning and improve the stability of training. The anchor of the YOLOv5 model is obtained from the COCO [33] dataset, the category and aspect ratio of the target in different datasets vary greatly, and the design of the initial anchor box must be carried out according to the dataset corresponding to the target detection task. Therefore, we use the K-means clustering algorithm [34–36] to cluster the training set by analyzing the shape and aspect ratio of the four classes of damage in the dataset to obtain the a priori anchor box in this experimental dataset.

The initial anchor boxes of YOLOv5 are [10, 13, 16, 23, 30, 33], [30, 61, 62, 45, 59, 119], and [116, 90, 156, 198, 373, 326]. For the RDD-2022 dataset used in this experiment, we use the K-means clustering algorithm to cluster the dataset, and the initial anchor boxes obtained are [13, 13, 53, 25, 31, 68], [71, 62, 166, 29, 67, 143], and [152, 85, 160, 186, 374, 202]. The size of the anchor boxes fits well with the dataset containing mostly narrow cracks. Training the model on this basis effectively reduces inference frame loss and improves detection accuracy.

For multiclassification problems, it is often necessary to transform the vector into a unique heat vector, i.e., the probability of considering the target should be 1 and the probability of the nontarget should be 0. The traditional expression of the unique heat vector y_i should be expressed as follows:

$$y_i = \begin{cases} 1, & i = \text{target}, \\ 0, & i \neq \text{target}. \end{cases} \tag{15}$$

Two problems arise when fitting the true probability function of the unique heat vector: the first is that the generalization ability of the model cannot be guaranteed and the second is that the full probability and zero probability result in the widest possible gap between the category to which they belong and the other categories, which are difficult to fit by the bounded gradient.

Label smoothing [37, 38] is an effective regularization method in the field of deep learning that aims to prevent models from predicting labels too confidently during training, improve the network overfitting problem during training, and improve the generalization ability of models [39]. We can consider the existence of incorrect labels in the dataset and assign them probabilities so that we can obtain a new label vector Y_i by expressing it as follows:

$$\mathrm{Y_i} = \begin{cases} 1-\varepsilon, & i = \text{target}, \\ \dfrac{\varepsilon}{K}, & i \neq \text{target}, \end{cases} \tag{16}$$

where K is the total number of categories and ε is a smaller adjustment parameter. In this experiment, we have performed many analyses on the dataset since transverse cracks and longitudinal cracks account for most of the four categories of damage. At the same time, transverse and longitudinal cracks are two separate classifications, and we do not want the model to rely on the labels too much during the training process. Based on the research study in [40] and group experiments, we set the adjustment parameter ε to 0.1, which can yield good results in solving the road damage from this experiment. The group experimental data are shown in Table 1.

4. Experiments and Discussion

The experimental environment is the Windows operating system, the model algorithm is implemented by the PyTorch deep learning framework, the graphics card is NVIDIA GeForce RTX 3060, the running memory of the graphics card is 12 GB, the CPU is Intel i5-12400F, and the memory is 32 GB. The initial input image size is set to 640×640, the model training period (epoch) is set to 100 epochs, and the batch size is set to 32. The initial learning rate is set to 0.02, the loop learning rate is set to 0.12, and the learning rate

Table 1: Experiments on label smoothing parameters.

ε	0.00	0.02	0.04	0.06	0.08	0.10	0.12	0.14	0.16	0.18
ΔmAP (%)	0.00	−0.13	0.22	0.31	0.24	**0.42**	0.40	0.41	0.17	−0.27

The bold values in the table mean the largest increase in mAP in the grouped experiments.

(a) (b) (c) (d)

Figure 10: Examples of types of damage: (a) transverse crack, (b) longitudinal crack, (c) grid crack, and (d) pothole.

momentum is set to the default value of 0.937. The weight decay parameter, warmup epochs, and warmup momentum are set to default values of 5×10^{-4}, 3, and 0.8, respectively. The SGD is used as the optimization function, cosine annealing is used for training, and the data enhancement is set in the same way as the original YOLOv5s model.

4.1. Dataset for the Experiment. The dataset used for the experiments in this paper is the open-source road damage dataset RDD-2022 [41], which includes 47,420 road images from six countries: Japan, India, the Czech Republic, Norway, the United States, and China. There are more than 55,000 road damage labels in the images, and the images in the dataset contain a variety of views, such as unmanned aerial vehicle (UAV) views and vehicle handheld cameras, which is beneficial for enhancing the generalization capability of the model for better application in real-world conditions. Various types of road damage are included in the dataset, and we focus on four types of road damage, namely, transverse crack D00, longitudinal crack D10, grid crack D20, and pothole D40, which are the main types of road damage, and their specific shapes are shown in Figure 10. Other forms of road damage in the dataset, as well as background images, are not included in the detection targets of this experiment, so it is necessary to process the images and labels in the dataset to identify the images and labels corresponding to the detection targets that are not included in this experiment. After analysis and processing, a total of 23,767 images contain detection targets of interest for this experiment. After processing the dataset, we divide the dataset according to the ratio 1 : 9, in which the training set has 21,392 and the validation set has 2,375, and we add 1% of the background images in the training set as in the COCO dataset to make the composition of the dataset more reasonable. The number of labels for four types of road impairments in the training and validation sets is shown in Figure 11.

4.2. Evaluation Parameters. Evaluation of the algorithm is considered in two main parts: computational cost and accuracy. Here, the computational cost is mainly characterized by the number of parameters (Params) and giga floating-point operations per second (GFLOPS). Usually, smaller Params and GFLOPS indicate that the model requires less computational cost and less performance from the hardware. The accuracy is mainly characterized by the following parameters: precision, recall, average precision (AP), mean average precision (mAP), and $F1$ score. Each evaluation parameter is specifically calculated as follows:

$$\begin{aligned} \text{Precision} &= \frac{\text{tp}}{\text{tp} + \text{fp}}, \\ \text{Recall} &= \frac{\text{tp}}{\text{tp} + \text{fn}}, \\ \text{F1} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \\ \text{AP} &= \int_0^1 P(R)\text{dR}, \\ \text{mAP} &= \frac{\sum_{i=1}^{K} \text{AP}_i}{K}, \end{aligned} \tag{17}$$

where tp represents the number of positive samples correctly detected, fp represents the number of samples that are detected as positive but are actually negative, fn represents the number of positive samples incorrectly detected as negative samples, and K is the total number of categories of detected targets.

Frames per second (FPS) is also a highly regarded evaluation metric in practical engineering applications. Especially for lightweight detection tasks, it is crucial to be able to perform the detection in real time. FPS is calculated from the detection response time and rounded to the nearest whole number.

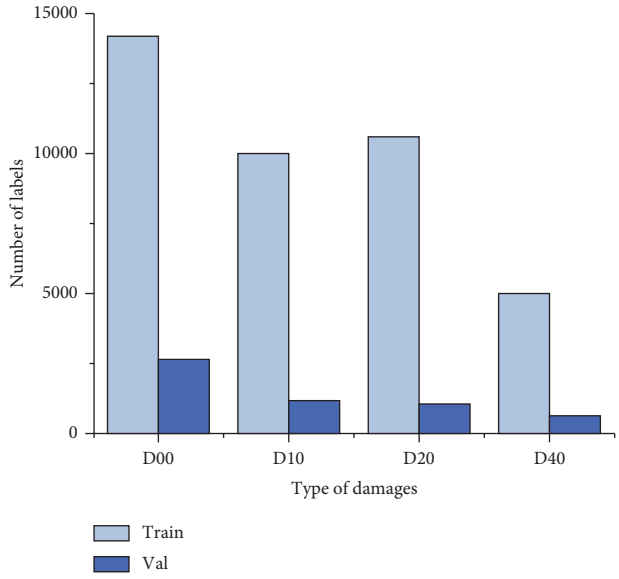


Figure 11: The number of labels for four types of road impairments in the training and validation sets.

$$\mathrm{FPS} = \frac{1}{T},$$
$$T = \mathrm{Tpre} + \mathrm{Tinf} + \mathrm{Tnms}, \tag{18}$$

where Tpre denotes the image preprocessing time, Tinf denotes the inference time, and Tnms denotes the postprocessing time.

4.3. Ablation Experiments. To evaluate the effectiveness of our optimization of the various parts of the model, we design stepwise ablation experiments for our improvements.

First, we design ablation experiments for the backbone network. The ablation model that uses improved MobileNetV3 as the backbone network is named YOLOv5s-impV3, and we use YOLOv5s-impV3 as the baseline in all subsequent experiments. The experimental data are shown in Table 2. From the results, it is clear that the parameters and GFLOPS of the model are significantly lower after replacing the backbone network, while mAP is also increased. Notably, this means that we have made the model lighter and the performance has been improved.

Second, we design ablation experiments optimized for the neck network. The ablation part includes the GSConv module and the VCACSP module, where the corresponding ablation models are named Baseline-GSConv and Baseline-GSConv-VCACSP, respectively, and the ablation experiment data are shown in Table 3. Our results demonstrated that applying the GSConv module and VCACSP module simultaneously further reduces the model parameters and results in a more substantial increase to the mAP. As discussed, the results of the ablation experiment confirm that our constructed neck network is more efficient and lightweight.

Similarly, to evaluate the parameter-free attention feature fusion (PAFF) module that we proposed, we also design an ablation experiment. Considering that the main purpose of the PAFF module is to improve the efficiency of utilizing the features of the same detection layer in the new backbone network, we still perform the ablation experiment on the basis of the baseline, and the ablation model is named Baseline-PAFF. The data of the ablation experiment are shown in Table 4. Without introducing additional parametric quantities, the mAP gains 1.3%. Improving the utilization of existing features is definitely a lighter and more efficient way than increasing the number of features and expanding the field of perception.

Overall, the stepwise ablation experiment is performed for all modules and algorithm optimization, and the experimental data are shown in Table 5. LE-YOLOv5 is improved on the basis of YOLOv5, so YOLOv5s is used as the baseline model in this part of the ablation experiments. Cases 1–8 are the models obtained by different combinations of module optimization. Our LE-YOLOv5 is the final model applying all the modules and algorithmic optimizations we have designed. Compared to the baseline, LE-YOLOv5 has significantly improved the overall performance.

In recent years, there have also been many representative and innovative algorithms in the field of target detection. To better compare the optimization effects of each part and evaluate the performance of our LE-YOLOv5 in the road damage detection task, we selected some representative algorithms. The specific experimental data are shown in Table 6.

To provide a more comprehensive assessment of the performance of LE-YOLOv5, the scope of our experiments included the baseline model and variants of YOLOv5. We also conducted experiments on faster R-CNN, SSD, DETR [42], and YOLOv7-tiny [43]. Among them, the faster R-CNN is the classical and more widely used two-stage detection algorithm. SSD and DETR are also classical single-stage target detection algorithms that have been used in recent years. YOLOv7 has been proposed in 2022 and is also recognized as a more advanced target detection algorithm in recent years. YOLOv7-tiny is the lightweight version of YOLOv7 proposed by its authors. YOLOv5l broadens the depth and width of the network based on YOLOv5s. This improves the performance of the algorithm but builds on a huge amount of computation. Comparisons between LE-YOLOv5 and some classical algorithms, and more advanced algorithms in recent years, are included in the comparison experiments. Meanwhile, the aim of this paper is to realize lightweight and efficient detection. In addition to the large computation algorithm, supplementing our comparison experiments with a lightweight detection algorithm such as LE-YOLOv5 can make our algorithm more convincing. Thus comparison experiments also include a comparison of the performance of the large-computing model and the lightweight model.

From the experimental results in Table 6, it is easy to see that LE-YOLOv5 has a huge advantage in terms of parameters and GFLOPS, and it is ranked first for precision. In

Table 2: Backbone network ablation experiment.

Models	Params (M)	GFLOPS	Precision (%)	Recall (%)	$mAP_{0.5}$ (%)	*F*1-score (%)
YOLOv5s	7.20	16.5	57.5	50.5	51.6	53.8
YOLOv5s-impV3	4.37	9.9	57.2	52.1	52.0	54.4

Table 3: Neck network ablation experiment.

Models	Params (M)	GFLOPS	Precision (%)	Recall (%)	$mAP_{0.5}$ (%)	*F*1-score (%)
Baseline	4.37	9.9	57.2	52.1	52.0	54.4
Baseline-GSConv	4.03	9.3	58.6	51.7	52.3	54.9
Baseline-GSConv-VCACSP	3.38	6.9	59.6	52.5	53.6	55.8

Table 4: The parameter-free feature fusion module ablation experiment.

Models	Params (M)	GFLOPS	Precision (%)	Recall (%)	$mAP_{0.5}$ (%)	*F*1-score (%)
Baseline	4.37	9.9	57.2	52.1	52.0	54.4
Baseline-PAFF	4.37	9.9	58.5	52.9	53.3	55.5

Table 5: Stepwise ablation experiment.

Models	Backbone	Modules			Algorithms		Params (M)	$mAP_{0.5}$ (%)
	ImpV3	GSConv	VCACSP	PAFF	K-means	Label smoothing		
Baseline							7.20	51.6
Case 1	✓						4.37	52.0
Case 2	✓	✓					4.03	52.3
Case 3	✓		✓				3.72	52.1
Case 4	✓			✓			4.37	53.3
Case 5	✓	✓	✓				3.38	53.6
Case 6	✓	✓		✓			4.03	54.6
Case 7	✓		✓	✓			3.72	54.1
Case 8	✓	✓	✓	✓			3.41	56.0
LE-YOLOv5	✓	✓	✓	✓	✓	✓	3.41	56.9

✓ means the modules or algorithms are used.

terms of recall, *F*1-score, and mAP, YOLOv5l has some advantages. However, as can be understood from its large parameters and GFLOPS, its advantages are based on a huge amount of computation. In comparison, LE-YOLOv5 has an outstanding lightweight advantage while not lagging behind in these three indicators. In terms of mAP, DETR, YOLOv7-tiny, and faster R-CNN lag behind significantly. Considering the transformer structure used in DETR, it requires a huge amount of training. Therefore, its mAP is predictably low at 100 epochs. However, the low training cost is also one of the advantages of LE-YOLOv5. YOLOv7-tiny is the closest lightweight model to LE-YOLOv5 parametric quantities, and its accuracy has a significant gap compared to LE-YOLOv5. However, it is still considered to be an excellent lightweight detection algorithm in recent years by virtue of its fastest detection speed. SSD is relatively close to LE-YOLOv5 in terms of mAP, but it has a large amount of computation and low FPS. Generally, by default in engineering applications, an FPS of 30 and above can be considered to meet the real-time detection requirements. In this regard, YOLOv7-tiny and YOLOv5s have obvious advantages. But with an FPS of 71, the LE-YOLOv5 performs real-time inspection tasks perfectly just as well.

In terms of the combined evaluation metrics, LE-YOLOv5 has a huge advantage in both parameters and GFLOPS, while the rest of the evaluation metrics are both ranked in the top two of all algorithms. It has no significant shortcomings in terms of overall performance, with an excellent balance of light weight and efficiency.

For a more visual and comprehensive comparison of multiple algorithms, we plot a schematic comparison of the comprehensive performance of the seven algorithms, as shown in Figure 12. Note that the purpose of the schematic diagram is to visualize the combined performance of the individual algorithms. Therefore, the values in the charts are processed with consistent conversions. The real data can be viewed in Table 6.

5. Visual Comparison of Detection Results in Complex Road Conditions

To better visualize the performance advantages of our LE-YOLOv5 in road damage detection, we compare LE-YOLOv5 with six other algorithms in multiple cases. The true location and class of the damage are labeled in the image according to the image's label as the true value. The images

Table 6: Multiple algorithms' comparison.

Models	Params (M)	GFLOPS	Precision (%)	Recall (%)	$F1$-score (%)	FPS	$mAP_{0.5}$ (%)
YOLOv5l	46.5	109.1	61.6	**55.3**	**58.3**	42	**57.4**
Faster R-CNN-ResNet50	41.48	94.3	57.0	52.2	54.5	9	50.3
DETR-ResNet50	36.74	100.9	48.7	38.3	42.8	36	39.7
SSD-VGG16	26.79	60.9	61.7	49.8	55.1	15	53.2
YOLOv5s	7.20	16.5	57.5	50.5	53.8	102	51.6
YOLOv7-tiny	6.02	13.2	51.4	52.3	51.8	**111**	49.1
LE-YOLOv5	**3.41**	**7.0**	**63.9**	53.2	58.1	71	56.9

The best results for each column in the table are bolded. The FPSs of all the abovementioned models are calculated on the basis of an input image resolution of 640 pixels × 640 pixels.

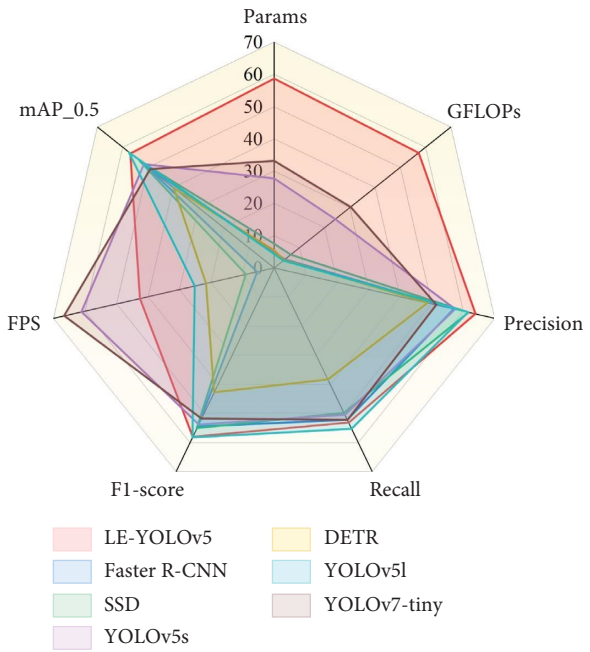


Figure 12: Comprehensive performance comparison diagram of multiple algorithms. Higher values indicate better performance.

are detected with LE-YOLOv5 and each of the other five algorithms, showing the location of the detection frame, the damage category, and probability. Note that for a more intuitive comparison, we have harmonized the display of the detection results of YOLOv7-tiny faster R-CNN, SSD, and DETR with the YOLOv5 algorithm.

Generally, the specific conditions of roads are complex and varied, so the robustness of the model is particularly important. Algorithms that maintain good performance in complex road environments have better prospects for practical engineering applications. In this experiment, we select five different cases and we analyze the comparative experiments for each of the five settings in the following sections.

Correctly detecting long-distance crack damage is more difficult than detecting ordinary crack damage. As shown in Figure 13, the precision of LE-YOLOv5 is more prominent when the detection regions of all algorithms are basically the same. YOLOv7-tiny and YOLOv5s, which are also lightweight models, have a significant gap in the accuracy level with LE-YOLOv5. This implies that our optimization results in a significant improvement in the ability of the algorithm to capture contextual information over long distances.

As shown in Figure 14, we use an image containing grid cracks and potholes of different sizes for the detection experiments. Compared with the ground truth, we find that only LE-YOLOv5 and YOLOv7-tiny correctly detect all damages. The other detection algorithms show omissions and incorrect detections. However, a careful comparison of the detection results of LE-YOLOv5 and YOLOv7-tiny shows that LE-YOLOv5 has a great advantage in the level of accuracy. This indicates that LE-YOLOv5 can fully accomplish detection in complex environments containing both types of damage.

The influence of weather factors on the detection should also be considered in road damage detection. For example, shadows of roadside plants and buildings can appear on the road surface in a strongly illuminated environment. Therefore, we use images containing transverse cracks across the shadows for detection experiments. As shown in Figure 15, among the seven detection algorithms, only LE-YOLOv5 detects the correct area of transverse cracks. The other detection algorithms are severely affected by shadows and only detect transverse cracks in bright areas. Our results demonstrate that LE-YOLOv5 is highly resistant to the shadows of pavement in strong lighting environments.

As shown in Figure 16, the foggy environment with reduced light and reduced visibility poses a challenge for road damage detection. The effect of partial traffic white lines on crack-type damage is further amplified in the foggy environment. This is a major challenge to the feature acquisition capability and learning ability of the prediction algorithm. In this detection experiment, only LE-YOLOv5 among the seven detection algorithms successfully detects longitudinal cracks covered by traffic white lines. Our results show that LE-YOLOv5 maintains high reliability and accuracy in a low-light environment.

Some of the smaller potholes in the pavement are often difficult to observe, which seriously impacts traffic safety. Therefore, we conduct detection experiments using images containing damage to a single small target pothole. As shown in Figure 17, the performance of the detection region and the confidence level of LE-YOLOv5 are excellent compared to the ground truth. Compared to the other six detection algorithms, the confidence of LE-YOLOv5 is substantially ahead. It is easy to observe from the results of

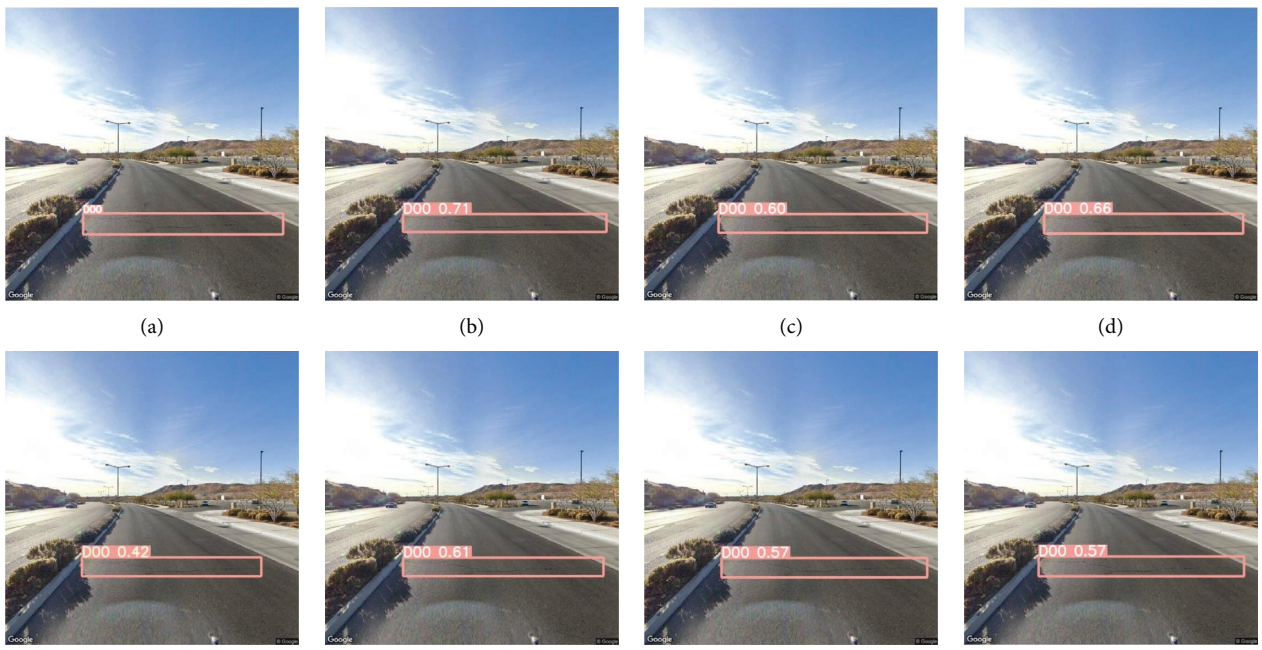


Figure 13: Visual comparison of long-distance transverse crack detection results: (a) ground truth, (b) LE-YOLOv5, (c) YOLOv5s, (d) YOLOv5l, (e) YOLOv7-tiny, (f) DETR, (g) SSD, and (h) faster R-CNN.

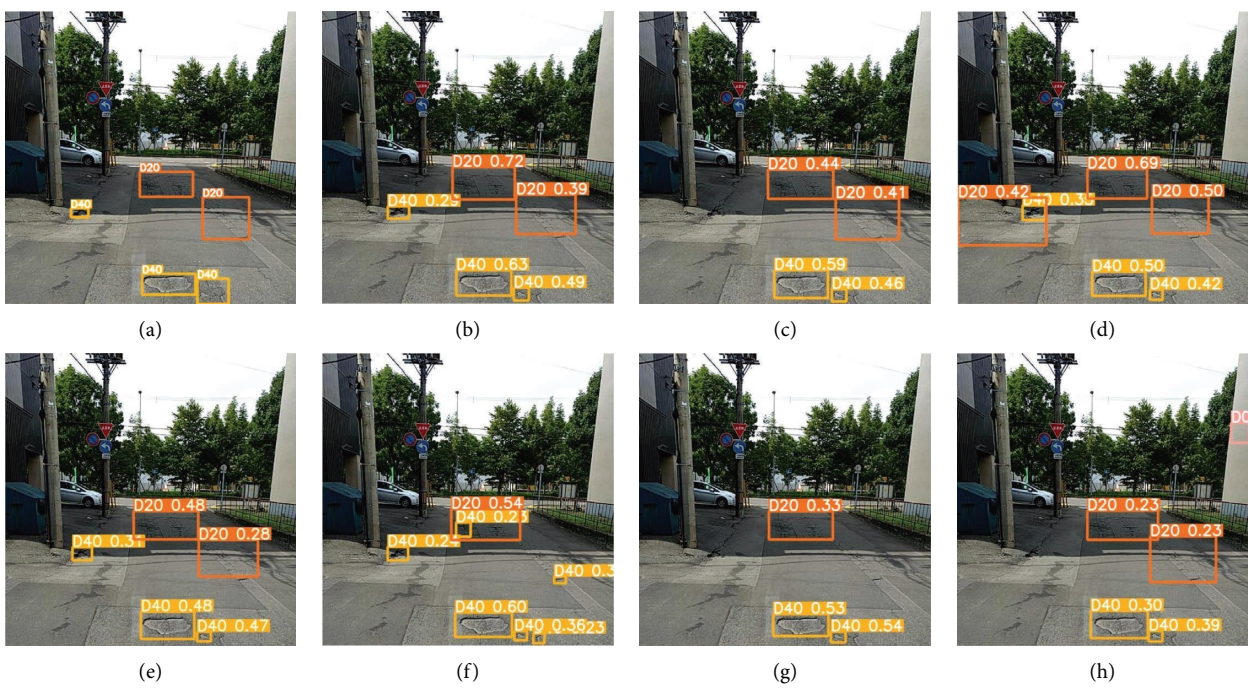
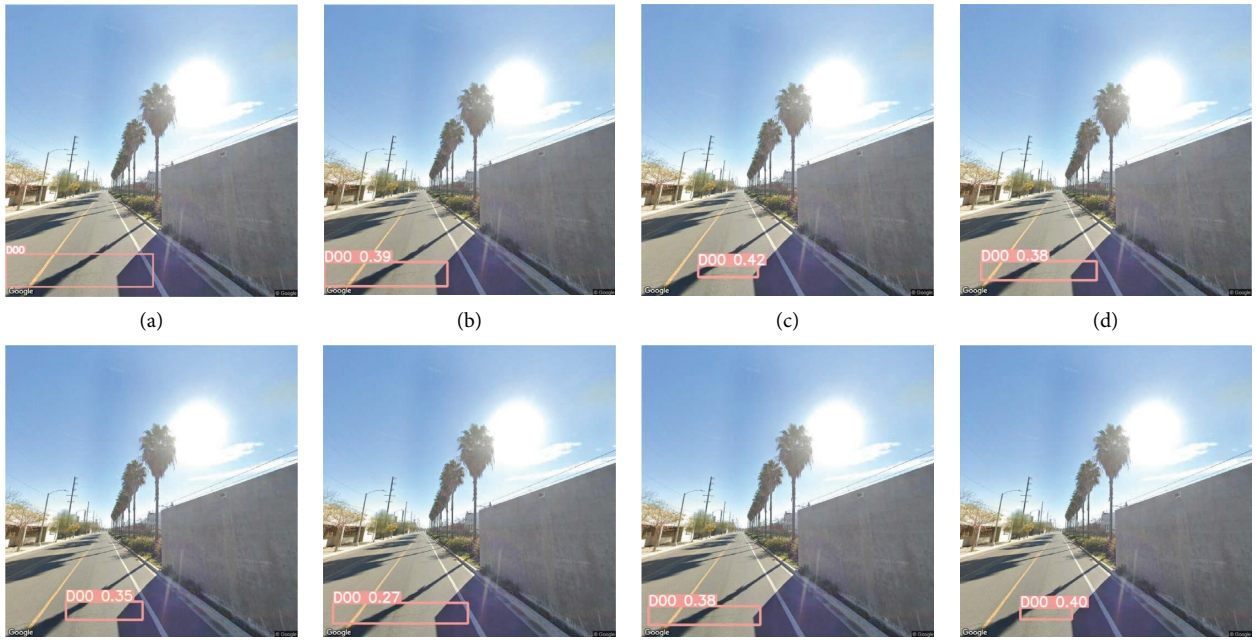


Figure 14: Visual comparison of detection results for complex pavements with multiple grid cracks and potholes coexisting: (a) ground truth, (b) LE-YOLOv5, (c) YOLOv5s, (d) YOLOv5l, (e) YOLOv7-tiny, (f) DETR, (g) SSD, and (h) faster R-CNN.

this experiment that LE-YOLOv5 also has very high reliability for the detection of a single small target.

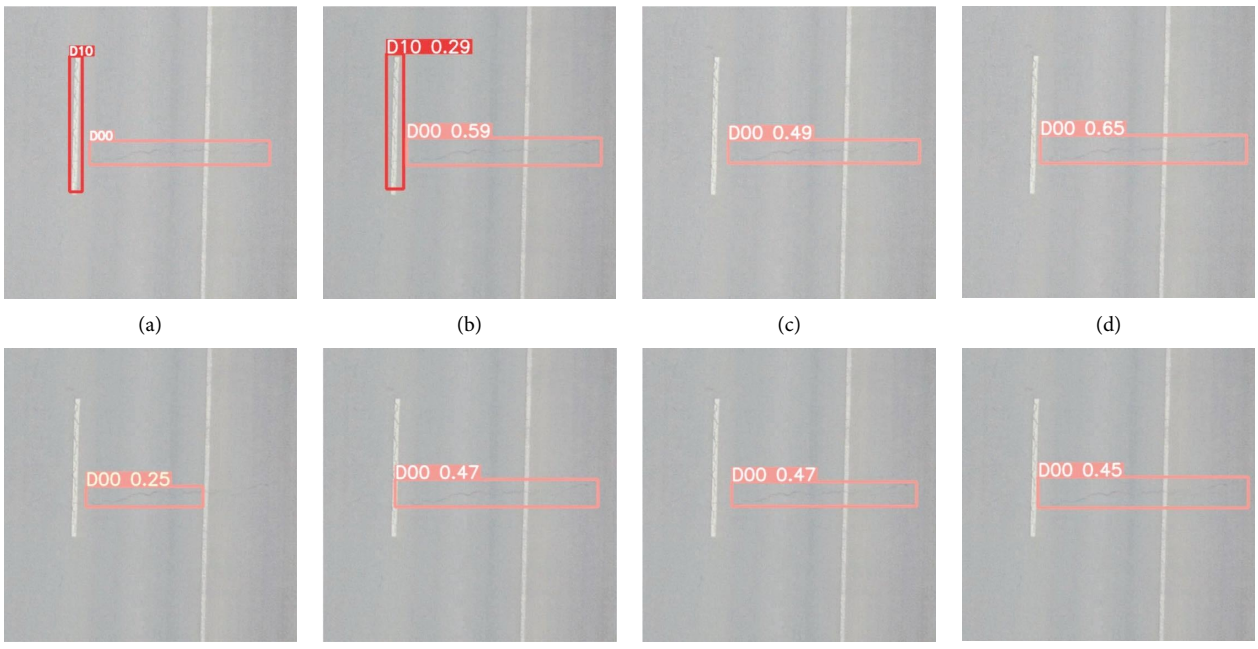
As in the discussion above, we conducted comparative experiments with six algorithms in a variety of complex environments. In all experiments, LE-YOLOv5 demonstrates a very high level of accuracy and excellent robustness at the lowest computational cost. Comparative experiments with other algorithms highlight the effectiveness of each improvement in this paper. The proposed GSAM module plays a good guiding role in the process of e-learning. The feature extraction ability and positioning accuracy of the overall algorithm are significantly improved. In addition, the



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 15: Visual comparison of crack damage across the shadow: (a) ground truth, (b) LE-YOLOv5, (c) YOLOv5s, (d) YOLOv5l, (e) YOLOv7-tiny, (f) DETR, (g) SSD, and (h) faster R-CNN.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 16: Visual comparison of the detection results of cracks covered by white lines in foggy weather: (a) ground truth, (b) LE-YOLOv5, (c) YOLOv5s, (d) YOLOv5l, (e) YOLOv7-tiny, (f) DETR, (g) SSD, and (h) faster R-CNN.

proposed PAFF module plays an important role in the feature fusion process. It improves the context feature association capability of the algorithm and improves the utilization efficiency of features at different levels in the network. These improvements make LE-YOLOv5 lightweight, robust, and efficient.

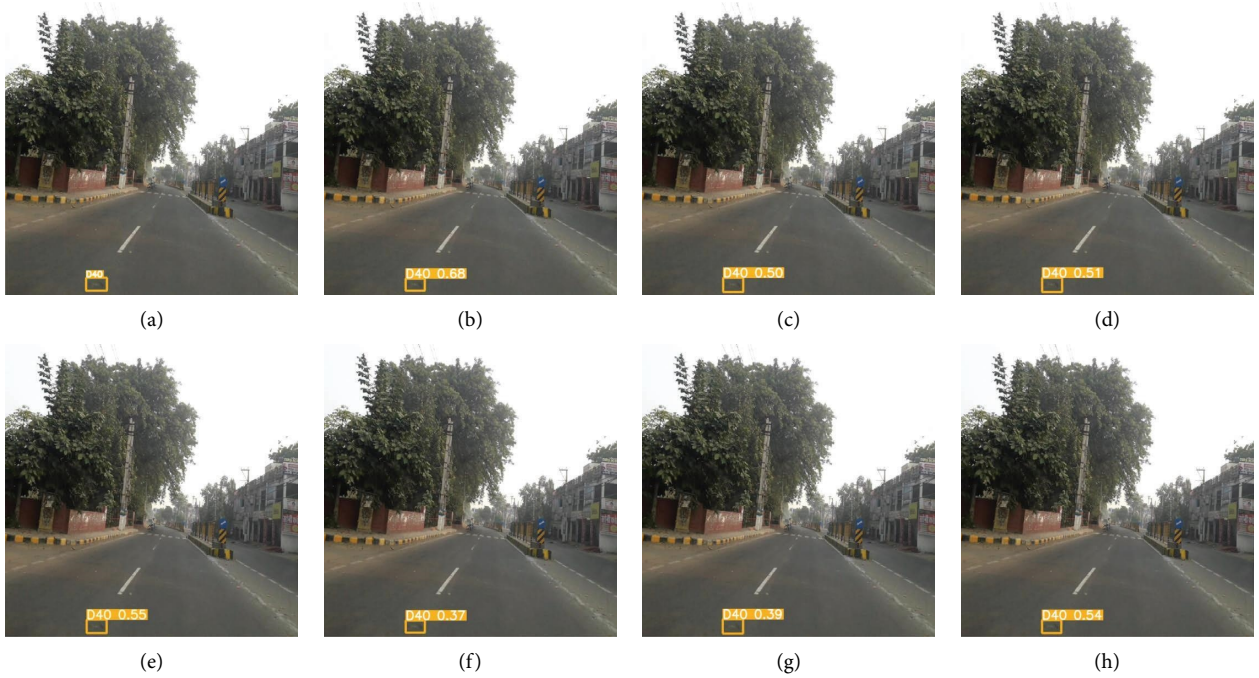


Figure 17: Visual comparison of a detection result of a single small target pothole: (a) ground truth, (b) LE-YOLOv5, (c) YOLOv5s, (d) YOLOv5l, (e) YOLOv7-tiny, (f) DETR, (g) SSD, and (h) faster R-CNN.

6. Conclusion

Based on the analysis of existing road damage detection methods, we propose a lightweight and efficient road damage detection algorithm LE-YOLOv5 to solve the problem of the high computational cost and insufficient accuracy of existing algorithms. We use the RDD-2022 dataset to train the model to detect various types of complex road damage. In terms of network structure optimization, above all, we propose a global shuffle attention module (GSAM) to improve MobileNetV3, thus designing a better backbone feature extraction network. Then, we use the GSConv module to replace the standard convolutional module in the neck network, thus significantly reducing the model parameters and GFLOPS. We introduce lightweight coordinate attention to design the VCACSP module and improve the network's ability to learn the spatial information of the feature map. To significantly enhance the model's ability to capture contextual information at a long distance without introducing additional parameters, we propose a parameter-free attentional feature fusion (PAFF) module. It further improves the learning ability of the network on the channel information and spatial information of the feature map without introducing additional parameters, which further improves the model performance. Finally, we use the K-means clustering algorithm to optimize the initial anchor boxes and the label smoothing algorithm to improve the generalization ability of the model. The experimental phase is divided into two main parts: the stepwise ablation experiment and the multimodel comparison experiment. Ablation experiments show that, compared to YOLOv5s, our LE-YOLOv5 reduces the model size by 52.7% and improves the mAP by 5.3%, which means that the model performance has been further improved while the model is lighter. Compared to large computational volume models, we have a clear volume advantage without losing accuracy. And compared to the excellent lightweight models YOLOv7-tiny and YOLOv5s in recent years, we have a significant accuracy advantage based on a much lower parameter count. In addition, it is easy to see in the detection visualization comparison results that our LE-YOLOv5 has better robustness and reliability and can cope with various complex road environments excellently. Our study provides an important reference value for lightweight road damage detection algorithms, offering the possibility of deploying road damage detection algorithms on more lightweight edge computing devices.

There are several possible directions to extend this work in the future. Regarding the improvement of the generalizability of the model, the dataset still has the possibility of expansion and optimization. We can improve the method of data collection as much as possible while further increasing the amount of data on pothole-type damage in the dataset. Then, the development of hardware and technology has made it possible to move from optimization of modules to optimization of the network structure. Finally, it is worth considering how to further expand the application of target detection, both from the perspective of lightweight and postprocessing of target detection.

Data Availability

The data used to support the findings of this study are available from the corresponding author.

Conflicts of Interest

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

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China (grant no. 11972084) and the National Natural Science Foundation of China (grant no. 12372178).

References

[1] F. Sultana, A. Sufian, and P. Dutta, "Evolution of image segmentation using deep convolutional neural network: a survey," *Knowledge-Based Systems*, vol. 201, Article ID 106062, 2020.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, Honolulu, HI, USA, July 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[4] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," *Neural Information Processing Systems*, vol. 29, 2016.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, Corfu, Greece, September 2017.

[6] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. J. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934.

[8] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the Computer Vision–ECCV 2016: 14th European Conference*, vol. 14, pp. 21–37, Amsterdam, The Netherlands, October 2016.

[9] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, Venice, Italy, October 2017.

[10] D. Arya, H. Maeda, S. Ghosh, and D. Toshniwal, "RDD2020: an annotated image dataset for automatic road damage detection using deep learning," *Data in Brief*, vol. 36, Article ID 107133, 2021.

[11] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto, and H. Omata, "Generative adversarial network for road damage detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 1, pp. 47–60, 2020.

[12] G. Ramachandra, "GAN Augmentation: augmenting training data using generative adversarial networks," 2017, https://arxiv.org/abs/1810.10863.

[13] V. Hegde, D. Trivedi, A. Alfarrarjeh, A. Deepak, S. Kim, and C. Shahabi, "Yet another deep learning approach for road damage detection using ensemble learning," in *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, pp. 5553–5558, Atlanta, GA, USA, December 2020.

[14] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, "Test-time augmentation with uncertainty estimation for deep learning-based medical image segmentation," 2018, https://arxiv.org/abs/1807.07356.

[15] S. Shim, J. Kim, S. W. Lee, and G. C. Cho, "Road surface damage detection based on hierarchical architecture using lightweight auto-encoder network," *Automation in Construction*, vol. 130, Article ID 103833, 2021.

[16] F. Wan, C. Sun, H. He, G. Lei, L. Xu, and T. Xiao, "YOLO-LRDD: a lightweight method for road damage detection based on improved YOLOv5s," *EURASIP Journal on Applied Signal Processing*, vol. 2022, no. 1, p. 98, 2022.

[17] A. G. Howard, M. Zhu, B. Chen et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[18] A. Howard, M. Sandler, G. Chu et al., "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, Seoul, Korea (South), November 2019.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, Salt Lake City, UT, USA, June 2018.

[20] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, Salt Lake City, UT, USA, June 2018.

[21] V. Mnih, N. Heess, and A. Graves, "Recurrent models of visual attention," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[22] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, Munich, Germany, September 2018.

[23] H. Zhang, K. Zu, J. Lu, Y. Zou, and D. Meng, "EPSANet: an efficient pyramid squeeze attention block on convolutional neural network," 2021, https://arxiv.org/abs/2105.14447.

[24] W. Xu, W. Wang, J. Ren, C. Cai, and Y. Xue, "A novel object detection method of pointer meter based on improved YOLOv4-tiny," *Applied Sciences*, vol. 13, no. 6, p. 3822, 2023.

[25] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, "Slim-neck by GSConv module: a better design paradigm of detector architectures for autonomous vehicles," 2022, https://arxiv.org/abs/2206.02424.

[26] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, Salt Lake City, UT, USA, June 2018.

[27] K. Xu, J. Ba, R. Kiros et al., "Show, attend and tell: neural image caption generation with visual attention," 2015, https://arxiv.org/abs/1502.03044.

[28] J. Han and P. P. Nair, "The influence of the sigmoid function parameters on the speed of backpropagation learning. From natural to artificial neural computation: international workshop on artificial neural networks malaga-torremolinos, Spain, june 7–9," *Cancer*, vol. 76, no. 2, pp. 195–200, 1995.

[29] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[30] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8759–8768, Salt Lake City, UT, USA, June 2018.

[31] G. K. Dziugaite and D. M. Roy, "Neural network matrix factorization," 2015, https://arxiv.org/abs/1511.06443.

[32] G. Kellas, S. T. Paul, M. Martin, and G. B. Simpson, "Contextual feature activation and meaning access," *Advances in Psychology*, Elsevier, Amsterdam, The Netherlands, 1991.

[33] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft CoCo: common objects in context," in *Proceedings of the Computer Vision–ECCV 2014: 13th European Conference*, pp. 740–755, Zurich, Switzerland, September 2014.

[34] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: a k-means clustering algorithm," *Applied statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[35] D. S. Modha, "Feature weighting in k-means clustering," *Machine Learning*, vol. 52, no. 3, pp. 217–237, 2003.

[36] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: a comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.

[37] Y. Xu, Y. Xu, Q. Qian, H. Li, and R. Jin, "Towards understanding label smoothing," 2020, https://arxiv.org/abs/2006.16653.

[38] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" *Neural Information Processing Systems*, vol. 32, 2019.

[39] B. Chen, L. Ziyin, Z. Wang, and P. P. Liang, "An investigation of how label smoothing affects generalization," 2020, https://arxiv.org/abs/2010.12648.

[40] G. Pereyra, G. Tucker, J. Chorowski, U. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," 2017, https://arxiv.org/abs/1701.06548.

[41] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2022: a multi-national image dataset for automatic Road Damage Detection," 2022, https://arxiv.org/abs/2209.08538.

[42] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillow, and S. Zagoruyko, "End-to-end object detection with transformers," *Computer Vision-ECCV 2020*, Springer International Publishing, Cham, Switzerland, 2020.

[43] 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," 2022, https://arxiv.org/abs/2207.02696.