Lightweight Road Defect Detection Algorithm Based on Improved YOLOv10

Kaihui Zhang, Yanze Wu , Nan Li, Xiaoyu Guo, Ran Cheng, Lijun Zhu

Abstract—**Road defect detection is an important task for road repair. Aiming at the problems of current road defect detection algorithms, such as large occupied space, low precision and complicated calculation, this paper introduces a lightweight algorithm for detecting road defects, improved by YOLOv10n. Firstly, In the backbone, we propose to replace C2f module with DERC2f module. The DERC2f module uses residual connection and Efficient Local Attention mechanism (ELA)to retain more original input information and accurately obtain the location information of defect features,enhance the feature extraction ability of the model. Secondly, the depthwise separable convolution is introduced, and the size of the convolution kernel is re-optimized, which lightens the detection head part and improves the detection efficiency of the model. Finally, SIoU and Slide Loss were introduced to replace the loss functions, which accelerated the convergence of the model, assigned higher weights to difficult samples, and further improved the accuracy. The experimental results show that in the dataset RDD2022, compared with the YOLOv10n model, the improved model has a 2.5% increase in mAP50, a 30% and 37.8% reduction in Params and FLOPs, respectively, which realize the lightweight of the model and the accurate detection of road defects.**

Index Terms—**road defect detection, YOLOv10n, lightweight, attention mechanism**

I. Introduction

IN recent years, road defect issues are increasingly drawing public attention. Road defects, including cracks and potholes, if not detected and repaired in their early stages, can evolve into more severe pavement damage. This not only affects traffic efficiency but also poses certain safety hazards. Traditional road defect detection methods are and costly time-consuming, and Small initial defects are difficult to detect. These methods are prone to being influenced by the subjective experience and judgment of inspectors, which in turn affects the inspection results. High-frequency manual inspections are also difficult to apply to current traffic volumes. With the advancement of technology and to accelerate the construction of smart cities, achieving efficient and accurate detection of road defects has become an urgent task to solve.

Traditional detection methods depend on manually extracting defect features, which makes it challenging to use a single model to identify multiple types of defects in complex and variable road environments, lacking practicality. Since the beginning of the 21st century, many countries have deployed vehicles for detecting road defects that can inspect road conditions without disrupting traffic. Additionally, Roadware Company has developed road damage detection vehicles suitable for nighttime operations. Due to the high application costs of road measurement vehicles, which can be as expensive as $500,000[1], they are currently not suitable for mass deployment.

Due to the inability of traditional detection algorithms to meet current needs, the widespread use of deep learning for road defect detection is driven by advances in deep learning technology, achieving remarkable results. C. Sun[2] et al. improved the Faster R-CNN by integrating it with the feature extraction layer of VGG16. L. Peng[3] et al. optimized the parameters of U-net to achieve its optimal structure on the Crack500 dataset. Although these methods offer higher detection accuracy, their computational demands are relatively high, leading to slower detection speeds. C. Wu[4] et al. optimized the YOLOv5 model, aiming to maintain its high performance while reducing computational requirements, which enhances its suitability for mobile devices operating under resource constraints. J. Zhao[5] et al. used GhostNet to improve YOLOv5 for lightweighting. Y. Jia[6] et al. effectively lightweighted the network by combining YOLOv5 with MobileNetV3 and GhostNet. X. Li[7] introduced BIFPN in YOLOv8's Neck, adjusted the number of channels, and drastically reduced the model's Params. Y. Li[8] et al. proposed an algorithm for detecting road defects using YOLOv8, utilizing the SimAM attention mechanism and GHostConv for enhancement. H. Geng[9] et al. replaced the traditional Bottleneck structure with FasterBlock and added the SE attention mechanism after each improved C2f module to boost model performance. The models mentioned above suffer from redundancy and are not highly efficient in terms of detection speed. There are still some difficulties with the above models, such as the inability to take into account the cost of computation in the improvement of detection accuracy, or the lack of efficiency

Manuscript received December 7, 2024; revised March 9, 2025.

This work was supported by the Educational science research project of Liaoning Province of China (LJ2020023).

Kaihui Zhang is a Postgraduate of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, 110142, China. (e-mail: 1831260001@qq.com).

Yanze Wu is a Postgraduate of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, 110142, China. (e-mail: 2998670393@qq.com).

Nan Li is a Postgraduate of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, 110142, China. (e-mail: 1160980927@qq.com).

Xiaoyu Guo is a Postgraduate of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, 110142, China. (e-mail: 2540961152@qq.com).

Ran Cheng is a Postgraduate of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, 110142, China. (e-mail: 2117534224@qq.com).

Lijun Zhu is a Professor of School of Computer Science and Technology, Shenyang University of Chemical Technology, Shenyang, 110142, China. (Corresponding author to provide phone: +86-159-9818-3125; e-mail: zhulijun@yeah.net).

regarding detection speed.

II. RELATED WORK

As deep learning technology advances, object detection techniques have been utilized in a diverse array of road defect detection applications. Object detection techniques include two-stage algorithms and one-stage algorithms. The two-stage algorithm consists of two steps, during the first step, several candidate areas are generated for the input image, in some of which may contain information about the target object. During the second step, bounding box regression and classification prediction is performed on these candidate regions. It has higher detection accuracy,and it is effective in detecting smaller targets, representative algorithms are R-CNN [10], Fast R-CNN[11] and Faster R-CNN[12]. One-stage algorithms take an image as input and directly process it to output the target, offering faster processing speeds, Among the algorithms included are SSD (Single Shot MultiBox Detector)[13] along with various models from YOLO (You Only Look Once) Series[14-16].

In May 2024, researchers from Tsinghua University introduced YOLOv10[17], an improvement over the YOLOv8 model. YOLOv10 enhances efficiency by eliminating Non-Maximum Suppression (NMS) and introduces spatial-channel decoupled downsampling, which retains information while reducing computational costs.

The YOLOv10n model is constructed with three main parts: the Backbone, the Neck, and the Head. The backbone of YOLOv10n extracts features at different levels from the input information through CBS modules, SCDown modules, C2f modules, and C2fCIB modules. Afterward, the features pass through the SPPF module, which primarily normalizes feature maps of different scales. The SPPF module has been enhanced with an additional partial self-attention (PSA) feature, which expands the receptive field and boosts the network's capability for capturing global information. In the YOLOv10n, a Path Aggregation Network-Feature Pyramid Network (PAN-FPN) is used as the feature pyramid network, responsible for multi-scale feature fusion. This structure helps in aggregating and refining features from different scales, increasing the model's multi-scale object detection capability. In the detection Head, a one-to-one head structure is introduced, similar to the one-to-many branches in YOLOv8. The optimization objectives for both heads are the same. During training, both heads are optimized together, providing rich supervision. However, during inference, only the one-to-one head is used for predictions. This approach allows for end-to-end deployment and reduces the inference cost. The computation of Loss consists of two branches, regression and classification. In regression loss， this approach employs the Dual-Focal Loss (DFL) and the CIOU loss functions. In classification loss， this approach employ the binary cross-entropy (BCE) loss function. The network structure diagram is shown in Fig. 1.

Focusing on the challenges faced by existing road defect detection methods such as large space occupancy, insufficient precision, high computational complexity, high cost and great environmental impact, this paper proposes a lightweight road defect detection algorithm based on YOLOv10n.

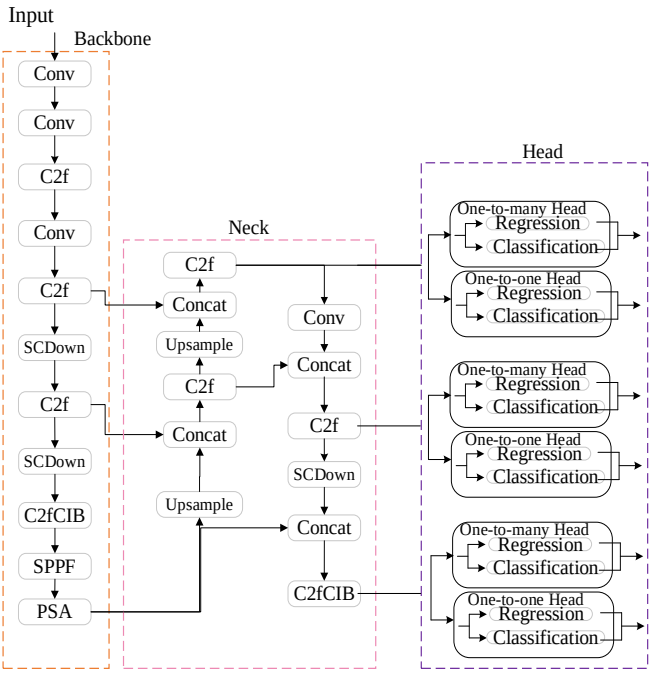


Fig. 1. YOLOv10n network structure

III. ALGORITHM DESIGN

A. *Efficient and Lightweight DERC2f module*

In YOLOv10, the C2f module plays a crucial role in the extraction of the features. As shown in Fig. 2, the bottleneck of the C2f module, which contains two convolutional layers. The input feature map passes through these layers, undergoing convolution, normalization, and activation operations. In the C2f module, after the first convolution, the output is split into two components through a Split process. One component is directly passed through, and the other component undergoes n bottleneck processing units. The outputs from both components are then concatenated, and the concatenated result is passed through a second convolution before being output. The structure of C2f is shown in Fig. 3.

Traditional convolutional operations generate a lot of redundant information during feature extraction, leading to larger models. To solve this problem, we have designed the DBottleneck model. First, a 1×1 convolution is used to increase the number of channels. Then, DWConv (Depthwise Convolution) applies an independent convolution kernel to each input channel, reducing the number of model parameters. Finally, another 1×1 convolution is used to reduce the number of channels and produce the output. The output is then combined with the original input through a residual connection. The structure diagram is shown in Fig. 4.

When using the DBottleneck module, we refer to it as the DC2f module. Due to the fact that passing information between various layers may easily distort the input feature information, we add an ELA (Efficient Local Attention) [18] module to the DC2f module.

The ELA mechanism is designed to accurately capture the positions of regions of interest, thereby enhancing the model's ability to extract multi-scale defects. As shown in Fig. 5, in the spatial dimension, ELA uses strip pooling to obtain feature vectors extracted in both the horizontal and vertical

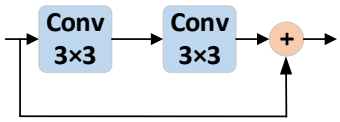


Fig. 2. Structure of the Bottleneck

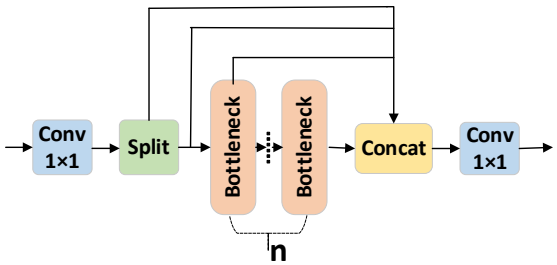


Fig. 3. Structure of the C2f

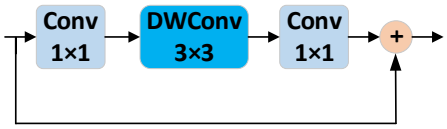


Fig. 4. Structure of the DBottleneck

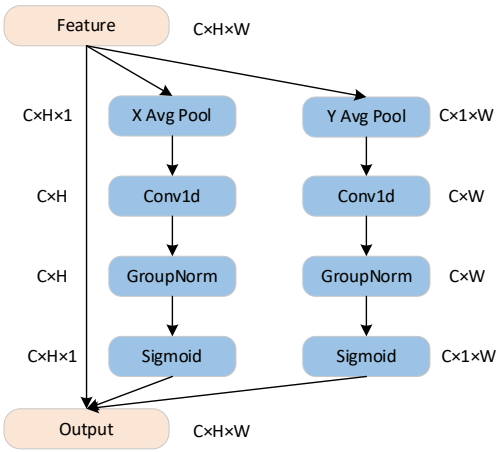


Fig. 5. Structure of the ELA

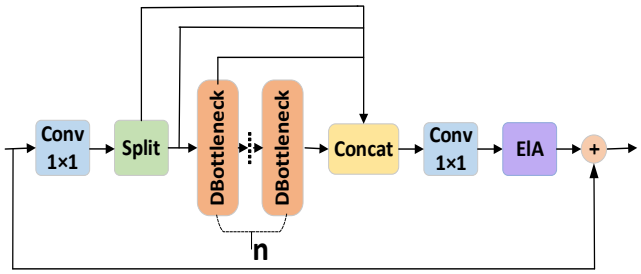


Fig. 6. Structure of the DERC2f

dimensions. It maintains a slim kernel shape to capture long-range dependencies, preventing irrelevant regions from interfering with label predictions. For targets in any direction ,this approach generates rich positional features. ELA processes the feature vectors in each direction separately to obtain attention predictions, which are then combined through a multiplication operation. This enhances the precise location information of the regions of interest.

Finally, by performing a residual connection with the input information, the risk of gradient vanishing is reduced, detail features are preserved, and the network's performance in detecting road defects is improved. The structure of DERC2f is shown in Fig. 6.

B. Design of the Head

In YOLOv10, the head section adopts the decoupled head structure used in YOLOv8, where classification and regression tasks are handled by separate branches. Each branch contains two 3×3 one-dimensional convolutions followed by one 1×1 two-dimensional convolution. This structure results in a significant amount of parameters within the Head, increasing the computational load.

To address this issue, we introduces the DSC (Depthwise Separable Convolution) [19] module, which consists of DWConv(Depthwise Convolution) and PWConv(Pointwise Convolution).

In the DWConv, a 3-channel input image passes through 3 single-channel convolutions, resulting in 3 single-channel feature maps. In the PWConv, the feature maps obtained from the DWConv undergo 4 convolutions with 1×1×3 kernels, producing 4 feature maps, effectively using 1×1 convolutions to increase the dimensionality. DSC schematic diagram is shown in Fig. 7.

Assuming that in the DWConv, the kernel size is $W_k \times W_k \times 1$, and the number of kernels is M, and in the PWConv, the kernel size is $1\times1\times M$, and the number of kernels is N, then the ratio of the number of Params processed by DSC to Conv is as follows:

$$\frac{Params_{DSC}}{Params_{Conv}} = \frac{W_k \times W_k \times M + M \times N}{W_k \times W_k \times M \times N} = \frac{1}{N} + \frac{1}{W_k^2} \quad (1)$$

In DWConv and PWConv operations, each convolution kernel performs $W_F \times W_F$ multiply-accumulate operations. then the ratio of the number of FLOPs processed by DSC to Conv is as follows:

$$\frac{\text{FLOPs}_{DSC}}{\text{FLOPs}_{Conv}} = \frac{W_k \times W_k \times M \times W_F \times W_F + M \times N \times W_F \times W_F}{W_k \times W_k \times W_F \times W_F \times M \times N \times W_F \times W_F}$$
$$= \frac{1}{N} + \frac{1}{W_k^2} \quad (2)$$

Assuming the convolution kernel size W_k is 3, and N is so large that it can be ignored for this calculation, and W_k^2 is 9, it can be concluded that using DSC reduces the Params and FLOPs to approximately 1/9 of the original.

We propose to redesign the head section using DSC module, which we name the DDetect module. In the DDetect module, we replace the first Conv module in both branches with a DSC, and set the kernel of the second Conv to 1×1 to reduce the number of input channels. This design significantly reduces the model's Params count and FLOPs, while having a negligible impact on detection accuracy. Additionally, it enhances the tightness of information flow between modules.The structure of DDetect is shown in Fig. 8.

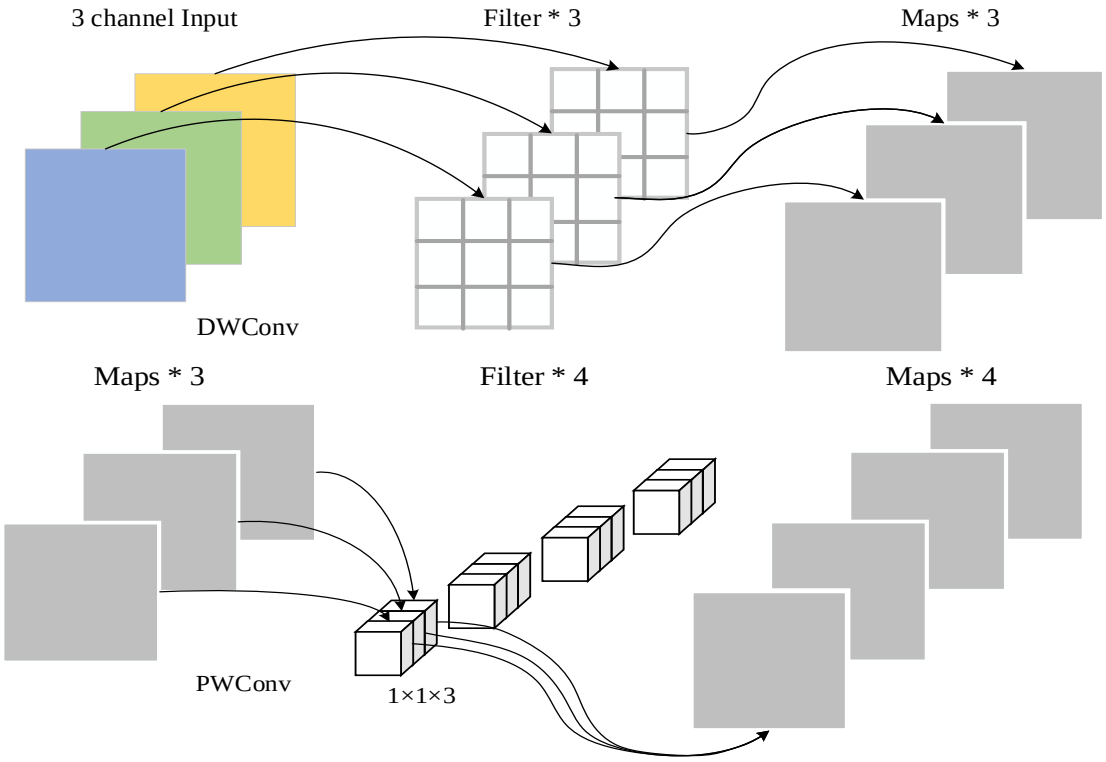


Fig. 7. The schematic diagram of DSC

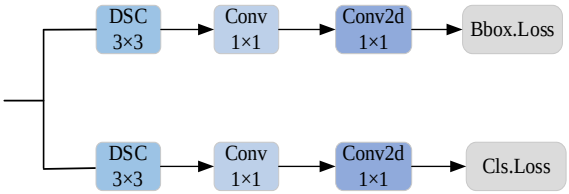


Fig. 8. Structure of the DDetect

C. *Improvement of the Loss Function*

YOLOv10 uses CIoU as the regression loss function, but it ignores the differences in difficulty between different categories and the imbalance in sample counts among classes, which can impact detection performance. To resolve these problems, we adopts the SIoU[20] function, which considers the orientation between the real box and the predicted box by introducing a vector angle, thereby accelerating convergence. The calculation of the SIoU loss function proceeds is described below:

1) Angle cost

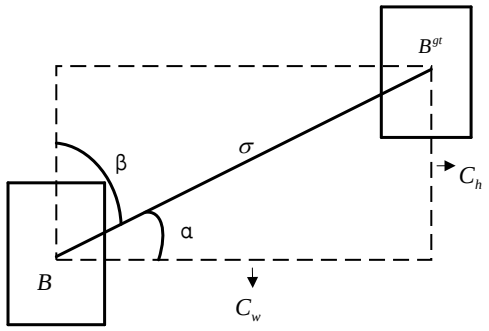


Fig. 9. Angle cost diagram

As shown in Fig. 9, B^{gt} and B represent the centres of the real and predicted boxes. c_h and σ represent the height difference and the distance. Angle cost can solve the problem of changing bounding box prediction. The Angle cost equation is shownin equation (3).

$$\Lambda = 1 - 2\sin^2(\arcsin(c_h / \sigma) - \pi / 4) = 1 - 2\sin^2(\alpha - \pi / 4) \quad (3)$$

2) Distance cost

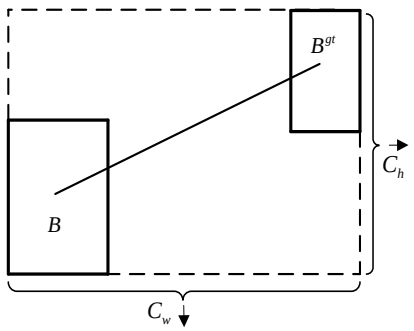


Fig. 10. Distance cost diagram

As shown in Fig. 10, The figure has an outer rectangle , Its height is c_h and its width is c_w. The Distance cost equation is shown in equation (4).

$$\Delta = 2 - e^{-\gamma\rho_x} - e^{-\gamma\rho_y} \quad (4)$$

In the equation:

$$\rho_x = (\frac{b_{c_x}^{gt} - b_{c_x}}{c_w})^2 \ , \ \rho_y = (\frac{b_{c_y}^{gt} - b_{c_y}}{c_h})^2 , \gamma = 2 - \Lambda \quad (5)$$

$(b_{c_x}^{gt}, b_{c_y}^{gt})$ and (b_{c_x}, b_{c_y}) are the coordinates of B^{gt} and B.

3) Shape cost

The Shape cost equation is shown in equation (6).

$$\Omega = (1 - e^{-w_w})^{\theta} + (1 - e^{-w_h})^{\theta} \quad (6)$$

In the equation:

$$w_w = \frac{|w - w^{gt}|}{max(w, w^{gt})} , w_h = \frac{|h - h^{gt}|}{max(h, h^{gt})} \quad (7)$$

(w^{gt}, h^{gt}) and (w, h) are the width and height of the real box and the predicted box, and θ denotes the attention level.

4) IoU cost

The IoU cost equation is shown in equation (8).

$$IoU = \frac{B^{gt} \cap B}{B^{gt} \cup B} \quad (8)$$

The final equation for the SIoU loss function is shown in equation (9).

$$Loss_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \quad (9)$$

In YOLOv10, the classification loss function uses BCEWithLogitsLoss. We propose to improve this by using Slide Loss[21]. Slide Loss assigns higher weights to difficult samples. In Slide Loss, easy and difficult samples are distinguished relying on the IoU between the predicted bounding box and the real bounding box. The average IoU of all bounding boxes is calculated as the threshold μ. Samples with an IoU below μ are classified as negative samples, whereas those with an IoU above μ are classified as positive samples. However, samples near the boundary, which have larger losses, should be optimized to make better use of them for training the network. To emphasize the learning of these boundary samples, a weighting function (Equation 10) is used. This function aims to give more importance to samples that are close to the decision boundary, this improves the capacity of the model for difficult cases. The weighting function is visualized in Fig. 11.

$$f(x) = \begin{cases} 1 & x \le \mu - 0.1 \\ e^{1-\mu} & \mu < x < \mu - 0.1 \\ e^{1-x} & x \ge \mu \end{cases} \quad (10)$$

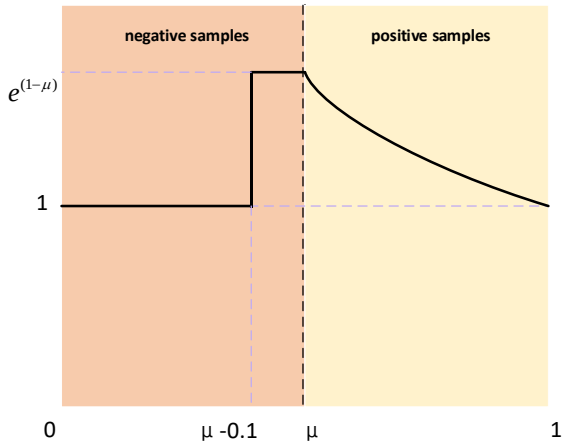


Fig. 11. Visualization of weighting function

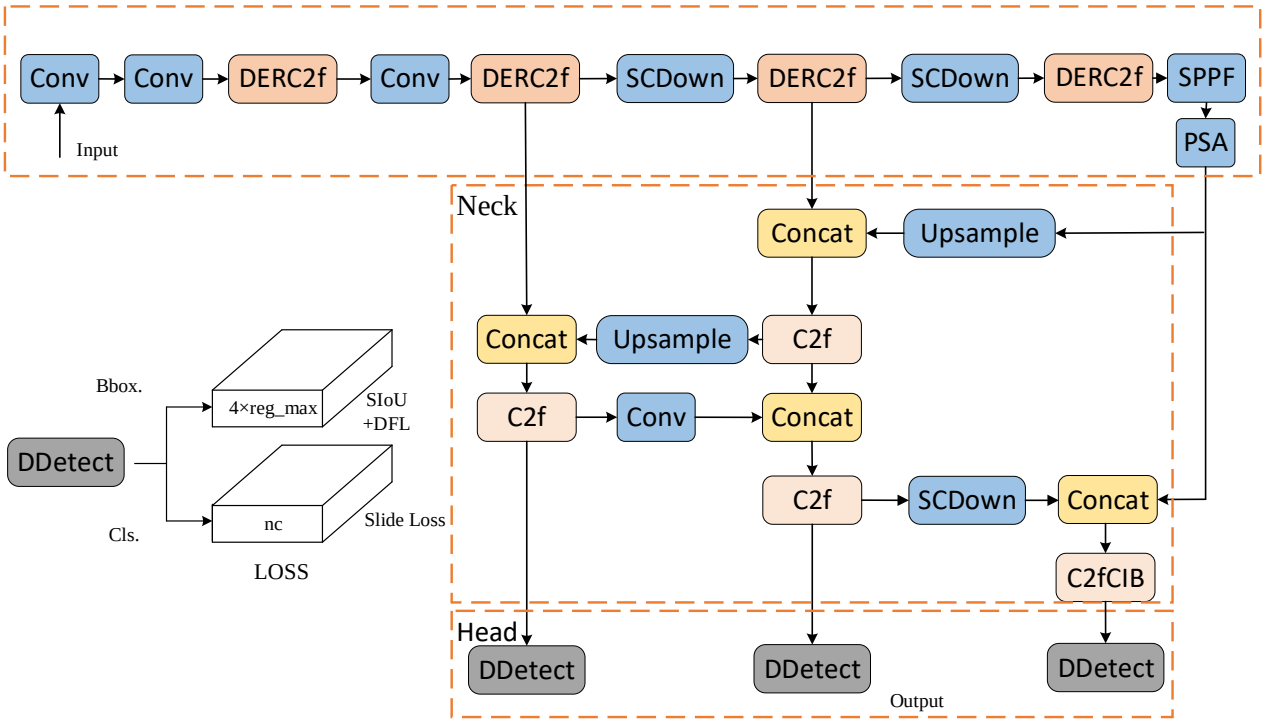


Fig. 12. Network Structure of DDS-YOLO

D. Network Structure of DDS-YOLO

The structure of DDS-YOLO network is shown in Fig. 12. In the backbone, we propose to replace C2f module with DERC2f module, which makes the network lightweight and enhances the feature extraction capability. After that, we redesign the Detect module in the Head, which greatly lightens the Head part. Finally, we improved loss function using Siou and Slide Loss, which accelerated the model's convergence, assigned more learning weight to difficult samples, and thereby improved detection accuracy.

IV. EXPERIMENT

A. Datasets

This study uses the publicly available RDD2022[22] dataset, the dataset consists of road images from a number of countries, including China, Japan, the United States, India, and the Czech Republic, etc. In this paper, 4378 Chinese road images contained in the dataset are selected as experimental objects. The selected dataset covers five type of road defects, including longitudinal cracks (coded as D00), transverse cracks (coded as D10), alligator cracks (coded as D20), potholes (coded as D40), and repairs (coded as Repair), where the numbers of the defects are 4104, 2359, 934, 321, and 1046, respectively. The dataset was randomly split 8:1:1 into training, validating and testing sets.

B. Experimental Environment and Parameter Setting

As shown in Table 1, This is the experimental configuration for this study, which also includes some parameter tuning. The input image size is 640 × 640.

TABLE I
EXPERIMENTAL ENVIRONMENT AND PARAMETER

Project	Settings
Operating System	Ubuntu 18.04
GPU	Nvidia RTX2070 8GB
CUDA	CUDA 11.6
Pytorch	Pytorch 1.13.1
CPU	Intel(R)Core(TM) i7-9700K
Optimization Function	SGD
epochs	260
Initial learning rate	0.01
batch size	16

C. Evaluation Metrics

To provide an objective assessment of the algorithm's detection performance, we use the following evaluation metrics: mAP (mean Average Precision), FLOPs, Params (number of parameters), model size, FPS (frames per second processed).

The equations for calculating mAP is given by equations (11-14):

$$P = TP / (TP + FP) \tag{11}$$

$$R = TP / (TP + FN) \tag{12}$$

$$AP = \int_0^1 P(R)dR \tag{13}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{14}$$

Where P and R represent Precision and Recall respectively, TP is a positive samples predicted correctly, FP is a negative samples predicted positively, FN is a positive samples predicted negatively. In general, a higher mAP means a higher performance of the model. Params, FLOPs, and model size reflect the degree of lightness of the model, and the smaller these values are, the lighter the network in terms of computational load, the lower the necessary hardware performance. FPS reflects how many images the model can detect in one second, a higher FPS value signifies improved real-time performance of the model.

D. Comparison of Feature Extraction Modules

To validate the superiority of the DERC2f module In feature extraction module, we conducted experiments by replacing it with the C2f module, the StarNet[23], the FasterNet[24], the C2f_ghost module, and the C3 module at the same position within the model for comparison. The data obtained using the various modules are shown in Table 2. By substituting original C2f module with DERC2f module, mAP improved by 1.6%, which is the most significant increase among the modules tested. Additionally, the DERC2f module contributes to a more lightweight model.

E. Comparison with the other attention mechanism

As shown in Table 3, to validate the superiority of the ELA, it was added to DERC2f alongside other attention mechanisms including Parameter-free attention mechanism SimAM[25], SE[26], combined spatial and access attention mechanism CBAM[27], and multi-scale attention mechanism EMA[28]. The experimental results indicate that when using ELA, the mAP is the highest, outperforming SimAM, CBAM, SE, and EMA by 1.2%, 1%, 2.4%, and 3.2%, respectively.

F. Comparison with the other lightweight convolutions

In the detection head, we conducted experiments by replacing the original model's components with DSC , GSConv[29],and GhostConv[30] to compare their performance. As shown in Table 4, all these modules can significantly reduce the number of parameters and computational cost. What's different is that when using GSConv, the mAP drops slightly. In contrast, GhostConv improves accuracy by 0.2%, while DSC enhances accuracy by 0.3%. Therefore, we introduce DSC as the more advantageous option.

G. Comparison with the other Loss Function

The original olgorithm places significant emphasis on the defect background when detecting difficult sample data, and the ability to suppress non-targets needs to be improved. So we improved the loss function and tried to replace CIoU with DIoU ,EIoU,WIoU and SIoU. As shown in Table 5,We found that the other three metrics (FLOPs, Params, and model size) remained unchanged. Only EIoU and SIoU resulted in an improvement in mAP, with SIoU providing the most significant increase.

H. Heatmaps of different improvement modules

As shown in Fig. 13. To more intuitively demonstrate the

TABLE II
EXPERIMENT OF COMPARISON WITH THE FEATURE EXTRACTION MODULES

Model	mAP50/%	FLOPs	Params	Model Size /M
C2f	82.0	8.2	2.7	5.8
C3	80.0	7.4	2.4	5.2
StarNet	82.6	7.5	2.5	5.4
FasterNet	81.5	7.4	2.4	5.2
C2f_ghost	79.5	7.1	2.3	5.0
DERC2f	83.6	7.5	2.5	5.4

TABLE III
EXPERIMENT OF COMPARISON WITH THE ATTENTION MECHANISM

Model	mAP50/%	FLOPs	Params	Model Size /M
SimAM	82.4	7.4	2.4	5.3
CBAM	82.6	7.5	2.5	5.2
SE	81.2	7.4	2.5	5.4
EMA	80.3	7.8	2.4	5.2
ELA	83.6	7.5	2.5	5.4

TABLE IV
EXPERIMENT OF COMPARISON WITH THE LIGHTWEIGHT CONVOLUTIONS

Model	mAP50/%	FLOPs	Params	Model Size /M
Conv	82.0	8.2	2.7	5.8
GSConv	80.8	5.6	2.0	4.4
GhostConv	82.2	5.6	2.0	4.4
DSC	82.3	5.7	2.1	4.5

TABLE V
EXPERIMENT OF COMPARISON WITH THE LOSS FUNCTION

Model	mAP50/%	FLOPs	Params	Model Size /M
CIoU	82.0	8.2	2.7	5.8
DIoU	81.0	8.2	2.7	5.8
EIoU	82.1	8.2	2.7	5.8
WIoU	81.4	8.2	2.7	5.8
SIoU	82.2	8.2	2.7	5.8

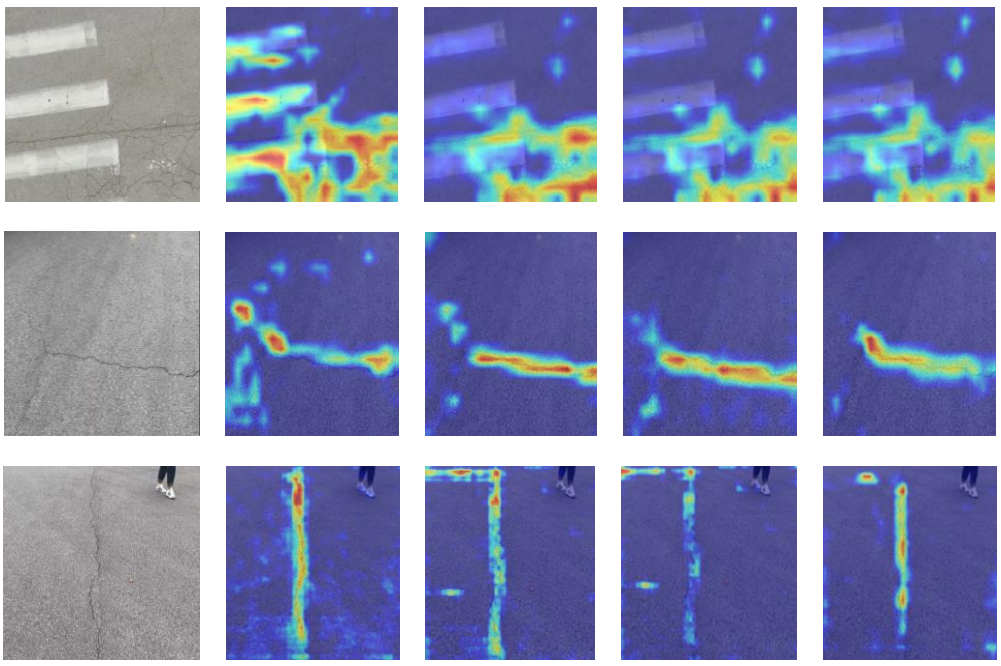


Fig. 13. Heatmaps of different improvement modules

comparison before and after each module's improvement, we used heatmaps to visualize the output layer of the model, where darker regions indicate higher attention from the model. We randomly selected 3 images. The Baseline model extracts features more broadly, whereas the improved modules focus more concentrated attention on the characteristics of road defects.

I. Ablation experiments

We conducted ablation experiments with the same experimental environment and parameter settings. As shown in Table 6, we displayed the data generated by the experiment. DERC2f means replacing the C2f module in backbone with DERC2f module, DDetect means replacing the Detect module in Head with DDetect module, and SIoU means replacing the regression loss function with SIoU. At the end, we improved the classification loss function with SlideLoss.

In Table 6, The design of DERC2f module led to a lightweight model, with FLOPs and Params dropping by 0.7G and 0.2M, respectively, and mAP rising by 1.6% due to the introduction of ELA, which makes the model focus on defective features. The DDetect module significantly lightens Head. with FLOPs and Params dropping by 2.5G and 0.5M, respectively. At the end, the improvement of the loss function has made model's mAP higher. Specifically, DDS-YOLO increases the model's mAP by 2.5%, while reducing FLOPs by 3.1G and Params by 30%.

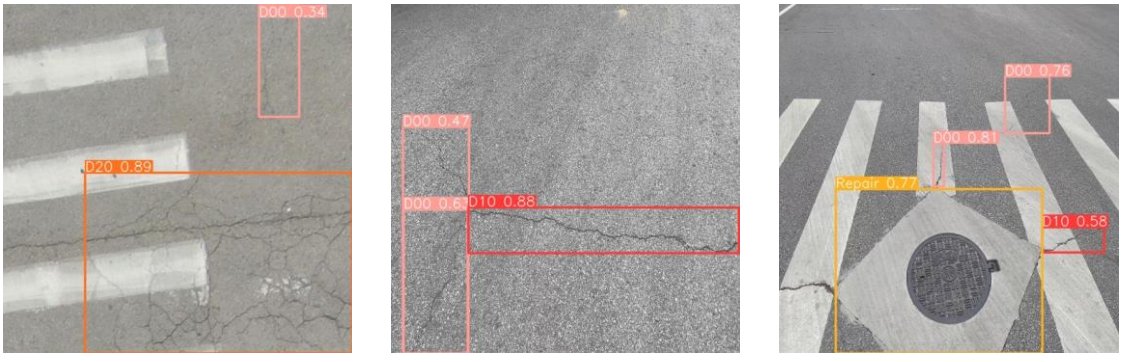
Fig. 14 shows the performance comparison between YOLOv10n and DDS-YOLO. It can be seen very clearly that YOLOv10n has issues with low confidence scores and missed detections. In contrast, the improved algorithm provides more accurate localization and recognition of defects.

J. Comparison with other advanced detection algorithms

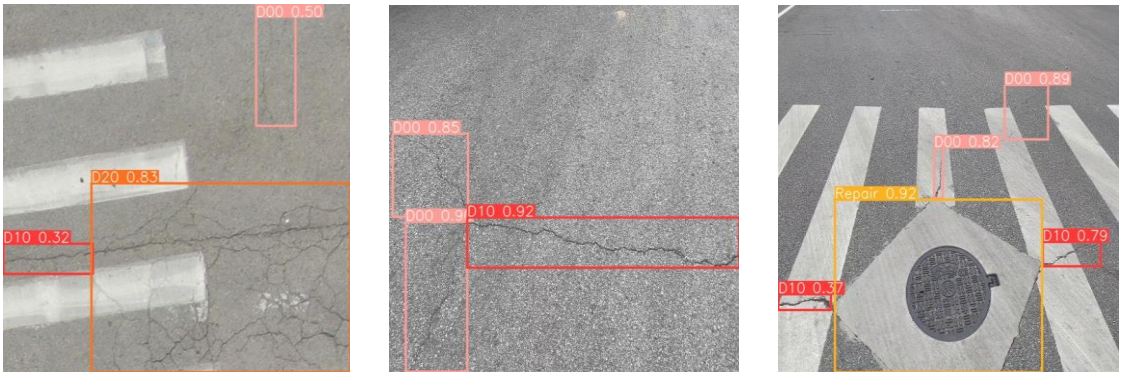
To further analyze the performance of the improved algorithm. We use FasterR-CNN, YOLOv3-tiny, YOLOv5s, YOLOv6n, YOLOv7-tiny, RT-DETR, YOLOv8n and YOLOv10n to do comparison experiments with DDS-YOLO in this paper on the dataset. As shown in Table 7, DDS-YOLO has higher mAP50, which is 13.3, 8.9, 0.8, 1, 0.4, 0.3, 0.3, and 2.5 percentage points higher than the other network models in order. YOLOv3-tiny has higher FPS. DDS-YOLO also has lower FLOPs and params, which are only 5.1G and 1.9M. FPS is also in the high level range. It can be seen that DDS-YOLO possesses higher accuracy and

TABLE VI
RESULT OF ABLATION STUDY

Model	mAP50/%	FLOPs	Params	Model Size /M
YOLOv10n	82.0	8.2	2.7	5.8
DERC2f	83.6	7.5	2.5	5.4
DDetect	82.3	5.7	2.1	4.5
DERC2f+DDetect+SIoU	84.2	5.1	1.9	4.1
DDS-YOLO	84.5	5.1	1.9	4.1



(a)YOLOv10n



(b)DDS-YOLO

Fig. 14. Visualization of YOLOv10n and DDS-YOLO detection results

TABLE VII
RESULT OF COMPARISON WITH OTHER ADVANCED DETECTION ALGORITHMS

Model	mAP50/%	mAP50-90/%	GFLOPs	Params	FPS
YOLOv3-tiny	75.6	47.1	19.1	8.7	220
YOLOv5s	83.7	53.5	15.8	7.0	71
YOLOv6n	83.5	53.3	11.2	4.2	92
YOLOv7-tiny	84.1	52.7	13.2	6.0	114
FasterR-CNN	71.2	42.2	370.2	137.1	12
RT-DETR	84.1	53.8	57.1	19.9	80
YOLOv8n	84.2	54.2	8.1	3.0	118
YOLOv10n	82.0	54.0	8.2	2.7	120
Ours	84.5	54.3	5.1	1.9	110

lower computational cost, which meets the actual road defect detection needs.

V. CONCLUSION

In this research, our algorithm is based on the YOLOV10n, considering the actual needs of road defect detection, we improved the C2f module. This is followed by mitigation of the Head section using the DSC module and convolutional kernel tuning. And finally we used the SIoU function as the regression loss function , and the Slide Loss as the classification loss function. In the process of improvement, many experimental comparisons were done to compare their performence. In the future work, more in-depth training and tuning will be done for the road defect detection task to improve the accuracy and robustness of the model. And there is also, the detection speed of the model is further improved, and the transplantation of embedded devices is verified to make it better meet the actual needs of detection.

REFERENCES

[1] D. Arya, H. Maeda, S. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Deep learning-based road damage detection and classification for multiple countries," Automation in Construction, vol. 132, no. 000, 2021.

[2] S. Sun, L. Pei, W. Li, X. Hao, and Y. Yao, "Pavement sealed crack detection method based on improved faster R-CNN," Journal of South China University of Technology (Natural Science Edition), vol. 48, no. 2, pp. 84-93, 2020.

[3] L. Peng, and H. Zhang, "Road defect detection based on U-net," Computer Science, vol. 48, no. S2, pp. 616-619, 2021.

[4] C. Wu, M. Ye, J. Zhang, and Y. Ma, "YOLO-LWNet: A Lightweight Road Damage Object Detection Network for Mobile Terminal Devices," Sensors, vol. 23, no. 6, pp. 3268-3268, 2023.

[5] J. Zhao, Y. Tao, Z. Zhang, C. Huang, and W. Cui, " Lightweight Road Damage Detection Network Based on YOLOv5," Engineering Letters, vol. 32, no. 8, pp.1708-1720, 2024.

[6] Y. Jia, Z. Zhang, X. Li, and Z. Zhang, "Defect detection of insulator on transmission line based on lightweight YOLOv5 network," Journal of Electric Power, vol. 39, no. 1, pp. 36-44, 2024.

[7] X. Li, and Y. Zhang, " A Lightweight Method for Road Damage Detection Based on Improved YOLOv8n," Engineering Letters, vol. 33, no. 1, pp.114-123, 2025.

[8] Y. Li, C. Yin, Y. Lei, J. Zhang, and Y. Yan, "RDD-YOLO: Road Damage Detection Algorithm Based on Improved You Only Look Once Version 8," Journal Title, vol. 14, no. 8, p. 2076-3417, 2024.

[9] H. T. Geng, Z. Liu, J. Jiang, Z. Fan, and J. Jia, "Embedded road crack detection algorithm based on improved YOLOv8," Journal of Computer Applications, vol. 44, no. 5, pp. 1613-1618, 2024.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hier-archies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Columbus, OH, 2014, pp. 580–587.

[11] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision(ICCV), Santiago, Chile, 2015, pp. 1440–1448.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, Montreal, Canada, 2015, pp.91-99.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in Computer VisionECCV2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp. 21–37, 2016.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Las Vegas, NV, USA, 2016, pp. 779–788.

[15] Ultralytics.YOLOv5[EB/OL].(2020-06-3)[2021-4-15]. https://github.com/ultralytics/yolov5.

[16] V. Rejin, and M. Sambash,"YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 2024, pp. 1-6.

[17] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," ArXiv Preprint ArXiv:2405.14458, 2024.

[18] X. Wei and W. Yi, "ELA: Efficient Local Attention for Deep Convolutional Neural Networks," ArXiv Preprint ArXiv:2403.01123, 2024.

[19] C. Francois, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1800-1807.

[20] G. Zhora, "SIoU Loss: More Powerful Learning for Bounding Box Regression," ArXiv Preprint ArXiv:2205.12740, 2022.

[21] X. Wang, "YOLO-FaceV2: A Scale and Occlusion Aware Face Detector," Pattern Recognition, vol. 155, no. 000, pp. 110714, 2024.

[22] D. Arya, H. Maeda, S. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2022: A multi-national image dataset for automatic road damage detection," Geoscience Data Journal, vol. 11, no. 4, pp. 846-862, 2024.

[23] X. Ma, X. Dai, Y. Bai, Y. Wang, and Y. Fu, "Rewrite the Stars," 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2024, pp. 5694-5703.

[24] J. Chen, S.-H Kao, H. He, W. Zhuo, S. Wen, C.-H Lee, and S.-H. G. Chan, " Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, pp. 12021-12031.

[25] L. Yang, R. Zhang, L. Li, and X. Xie, "SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks," Volume 139: International Conference on Machine Learning , Virtual, 2021, pp. 11863-11874.

[26] J. Hu, L. Shen, S. Albanie, G. Sun, and E Wu, "Squeeze-and-Excitation Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 8, pp. 2011-2023, 2020.

[27] S. Woo, J. Park, J. Lee, and I. Kweon, "CBAM: Convolutional Block Attention Module," Lecture Notes in Computer Science, vol. 11211, pp. 3-19, 2018.

[28] D. Ouyang, S. He, G. Z. Zhang, M. Z. Luo, H. Y. Guo, J. Zhan, and Z. J. Huang, "Efficient Multi-Scale Attention Module with Cross-Spatial Learning," ICASSP 2023 - 2023 IEEE International Conference on

Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 2023, pp. 1-5.
[29] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, “Slim-neck by GSConv: A lightweight-design for real-time detector architectures,” Journal of Real-Time Image Processing, vol. 21, no. 3, pp. 1-13, 2024.
[30] K. Han, Y. Wang, Q. Tian, J. Guo, and C. Xu, "GhostNet: More Features From Cheap Operations," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, USA, 2020, pp.1577–1586.