Improved Road Damage Detection Algorithm Based on YOLOv8n

Xudong Li, Yujun Zhang*

Abstract—**This paper introduces an advanced road damage detection algorithm that effectively addresses the shortcomings of existing models, including limited detection performance and large parameter sizes, by utilizing the YOLOv8n model. Key enhancements are integrated into the proposed algorithm to bolster its efficacy. First,the ConvNeXt V2 backbone network is integrated to improve the extraction of contextual features, thereby enhancing the effectiveness of road damage detection. Second, the algorithm employs a C2f_GhostNetV2 block structure to strengthen feature representation while simultaneously reducing computational costs. Additionally, PConv is utilized in the neck region to optimize spatial feature extraction, thereby minimizing redundant computations. The experimental results indicate that the proposed algorithm performs effectively on the Chinese subset of the RDD2022 dataset. Specifically, detection accuracy improved by 1.9%, recall by 1.8%, and mAP@0.5 by 2.1%, while the number of parameters decreased by 24% compared to the initial model. The optimized algorithm increases FPS by 4, meeting the dual requirements of mobile devices for accuracy and real-time object detection.**

Index Terms—**road damage detection, YOLOv8n, ConvNeXtV2 backbone network, C2f_GhostNetV2 block structure, PConv**

I. Introduction

With the rapid advancement of artificial intelligence, real-time performance is becoming crucial for object detection across various applications. One notable application is road damage detection technology, which plays a crucial role in ensuring road maintenance and safety [1]. The issue of road damage not only impacts traffic safety and driving efficiency but also imposes significant economic and environmental burdens on vehicle owners and society at large. This highlights the necessity for timely maintenance by transportation agencies [2]. As of 2022, China has maintained over 5.3503 million kilometers of highways, representing 99.9% of the nation's total highway mileage, thereby emphasizing the growing importance of highway maintenance efforts in the country [3]. Given these factors, road damage detection technology has become an indispensable tool in the realm of road maintenance and safety.

Road damage detection technology aims to distinguish between different types of road damage. This is achieved through various methods, including manual inspection techniques, automated systems techniques, and image processing techniques. Historically, manual inspections were the primary method used to assess road conditions. However, this approach encountered several challenges, including harsh working conditions, inadequate safety measures for personnel, and limitations in detection accuracy due to inspectors' varying levels of expertise and experience. As technology has advanced, there has been a shift towards automated methods, such as vehicle sensor system. Despite their increased accuracy and efficiency, these automated systems are expensive, posing significant challenges for financially constrained regions or institutions [4, 5]. Additionally, the operation and maintenance of these systems require personnel with a high level of technical expertise. In contrast, image processing techniques provide a more cost-effective and efficient alternative, with their accuracy improving over time as technology progresses. Nevertheless, these techniques are not without their own challenges. Image processing algorithms need to be further developed to improve their generalization and robustness in real-world scenarios, address suboptimal performance in detecting small targets, and reduce high computational complexity. Additionally, they must become less susceptible to environmental factors such as weather and lighting conditions. Consequently, ongoing research efforts are dedicated to developing road damage detection algorithms that can overcome these limitations [6–8].

Advancements in deep learning techniques have led to significant progress in the field of object detection. Various convolutional neural network architectures, including R-CNN [9], Fast R-CNN [10], Faster R-CNN [11], SSD [12], and YOLO [13], have been employed to enhance the accuracy and stability of road damage detection. Among these approaches, YOLO has notably distinguished itself by achieving a commendable balance between speed and accuracy, thus facilitating rapid and reliable object identification in images. Its design also features a streamlined set of hyperparameters, which makes it relatively straightforward to adjust. However, despite these advancements, there remains potential for enhancing the accuracy of road damage detection when directly applying these methodologies. Moreover, achieving real-time performance that meets the accuracy requirements for mobile terminal devices continues to present a significant challenge.

YOLOv8 algorithm has demonstrated substantial advancements in both speed and detection accuracy over its predecessors. Nevertheless, these improvements have been accompanied by an increase in the model complexity and operational cost. Despite significant progress, there is still potential for improvement in detection accuracy and model performance. To address this, this paper proposes a road damage detection technology based on the YOLOv8 algorithm. The following

Manuscript received February 14, 2024; revised September 7, 2024. The work was supported by the Intelligent Construction Internet of Things Application Technology Key Laboratory of Liaoning Province, under project number 2021JH13/10200051.

Xudong Li is a graduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China. (e-mail: 546133815@qq.com).

Yujun Zhang is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China. (Corresponding author to provide e-mail: 1997zyj@163.com).

are the main contributions of this study:

This paper introduces the integration of a newly developed convolutional neural network module, ConvNeXt V2 [14], into a redesigned backbone network structure with the objective of enhancing feature extraction capabilities. ConvNeXt V2, an advanced convolutional neural network architecture, has exhibited significant performance improvements across a variety of visual recognition tasks, surpassing traditional convolutional neural networks in numerous recognition benchmarks. This module incorporates innovative techniques such as inverted bottleneck layers and large convolutional kernels, which serve to expand the network's receptive field, thus enhancing the capture of contextual information. Furthermore, the inclusion of group normalization enables the module to effectively manage information at various scales, thereby improving its capacity to extract contextual features. Additionally, depth-wise separable convolution is employed to streamline computational complexity without compromising expressive power. With these enhancements, the ConvNeXt V2 module is designed to enhance feature extraction capabilities, thereby improving performance in road damage detection tasks.

To enhance the performance of object detection, this paper introduces the C2f_GhostNetV2, a lightweight attention module derived from the original C2f module. The core purpose of the C2f_GhostNetV2 is to generate more feature maps while utilizing fewer parameters. This objective is achieved by substituting the original C2f DarknetBottleneck with GhostNetV2 blocks [15]. The design of the C2f_GhostNetV2 module enriches gradient flow within the model by establishing connections across layers, thereby fusing shallow and deep features and consequently enhancing object detection performance. It is recognized that reducing parameter counts to improve real-time performance may lead to compromised detection accuracy. However, the GhostNetV2 block is selected to address this challenge as it maintains lightweight characteristics without sacrificing representational capability.

This paper introduces Partial Convolution (PConv) [16], an efficient technique for capturing spatial information while reducing redundant operations and memory access overhead. This dual function ensures real-time processing and improves model accuracy. By minimizing unnecessary computations, PConv helps in maintaining high floating-point operations per second (FLOPS) while reducing floating-point operations (FLOPs). This optimization greatly enhances the model's computational performance and memory utilization.

The core value of this research lies in enhancing the YOLOv8n algorithm, achieving real-time performance while maintaining high detection accuracy and robustness. This advancement provides essential technical support for maintaining road health and safety. Additionally, the findings of this research offer valuable insights and methodologies for researchers in other detection fields grappling with similar issues. To assess the practical efficiency and application potential of the improved YOLOv8n algorithm for road damage detection, this paper provides a comprehensive summary and analysis of the research results. The structure of this study is outlined as follows: The second part summarizes road damage detection technologies based on traditional and deep learning methods, and provides a detailed introduction to the YOLOv8n algorithm. The third part focuses on the development of new strategies for road damage detection. This is followed by the fourth part, which presents and analyzes the experimental results. Finally, this paper summarizes the research in the fifth part.

II. RELATED WORK

In recent years, object detection has become a key area of computer vision research, demonstrating broad applicability across various industries. Research in object detection includes both traditional image processing methods and deep learning-based approaches. Prominent traditional detection methods include Scale-Invariant Feature Transform (SIFT) [17], Histogram of Oriented Gradients (HOG) [18], and Deformable Part Model (DPM) [19], and Speeded-Up Robust Features (SURF) [20]. Historically, pavement crack detection systems relied on line scan or plane scan cameras to capture images of the road, which were subsequently analyzed manually to assess road conditions. However, these conventional methods were often heavily dependent on the quality of the original images and involved intricate algorithms, limiting their practicality and effectiveness.

The use of deep learning-based methods for road defect detection has become increasingly prevalent, especially with Convolutional Neural Networks (CNNs) being central to learning feature representations and classifying images. These methods' primary advantage is their capacity to automatically learn features suitable for various types of road defects. Performance improvements can be achieved by augmenting training data and adjusting network structures. Maeda et al. [21] evaluated road defect detection on resource-limited devices by using deep learning methods to classify road defects from smartphone images, showcasing a cost-effective, efficient, and viable approach for mobile devices. Seungbo Shim et al. [22] proposed a hybrid algorithm combining semi-supervised learning and generative adversarial networks (GANs), which effectively improved the recognition accuracy. Naddaf et al. [23] proposed a road damage detection method based on EfficientDetD7, introducing a series of scalable and efficient models that achieved notable success in a challenge. However, due to the large number of parameters, the detection speed is slow, making it unsuitable for tasks requiring real-time detection. In contrast, Fang Wan et al. [24] proposed a lightweight road damage detection algorithm based on YOLOv5s, incorporating a new backbone network combined with ShuffleNetV2 and ECA attention to balance accuracy and detection speed, making it deployable on mobile devices. Further advancing this field, Liu Haohan et al. [25] introduced an object detection algorithm based on YOLOv7-tiny, utilizing an enhanced ShuffleNetv1 backbone network with a lightweight feature pyramid to improve detection accuracy. The Mish activation function was also employed to enhance the model's generalization ability, addressing deployment challenges on edge devices. The computational complexity of two-stage target detection algorithms is often unsuitable for real-time tasks, and the Single Shot MultiBox Detector (SSD) algorithm also demands more computational resources compared to the YOLO algorithm. Consequently, for improved performance, this study selects the latest YOLOv8 as the foundational algorithm because of its high detection performance and

real-time processing ability. Therefore, this research aims to refine the YOLOv8n algorithm and investigate road damage detection algorithms based on this improved foundation.

YOLOv8, developed by Ultralytics, the same company responsible for YOLOv5, presents a sophisticated architecture comprised of backbone, neck, and head components. The backbone structure in YOLOv8 closely resembles that of the YOLOv5 series, featuring identical Conv and SPPF layers. However, noteworthy adjustments include reducing the size of the initial convolution kernel and modifying the CSPLayer. These changes draw inspiration from the structural design of YOLOv7, culminating in the introduction of the C2f module. The C2f module enhances gradient flow by introducing cross-level connections and optimizes the integration of shallow and deep features, thereby improving feature fusion capabilities. In the neck, YOLOv8 utilizes a combination of Path Aggregation Network (PANet) and Feature Pyramid Network (FPN) to achieve feature fusion [26, 27]. Unlike its predecessors, YOLOv8 employs an anchor-less mechanism and features a separate head structure for handling classification and regression tasks independently. This innovative architecture minimizes interference between classification and regression tasks, enabling each to focus on its respective responsibilities. Consequently, this enhancement leads to improved detection capabilities for irregular objects and elevates the overall model accuracy. Furthermore, YOLOv8 uses binary cross-entropy (BCE) to address classification task losses, with the formula as follows:

$$Loss_n = -w\left[y_n \log x_n + (1-y_n)\log\left(1-x_n\right)\right] \tag{1}$$

Where w represents the weight, y_n denotes the ground truth, and x_n represents the predicted value by the algorithm. YOLOv8 uses Complete Intersection over Union (CIoU) loss for regression of bounding boxes and introduces the Dual-FOcus Loss (DFL) to address regression challenges. The main purpose of the DFL is to rapidly converge the network to values near the labels, addressing the issue of localization accuracy in detection. The calculation process is outlined as follows:

$$\begin{aligned} DFL(p(i), p(i+1)) = &-\left[(y_{i+1}-y)\cdot\log(p(i))\right. \\ &\left.+\,(y-y_i)\cdot\log(p(i+1))\right] \end{aligned} \tag{2}$$

Where $p(i)$ and $p(i+1)$ are the predicted bounding box distribution probabilities, y_i and y_{i+1} are adjacent localization labels, and y is the actual label value. The CIOU loss effectively incorporates overlap/non-overlap and distance between center points to improve the model's accuracy in learning bounding box positions and shapes. The formula is:

$$L_{CIoU} = 1 - IoU + \left(\frac{\rho^2\left(b, b^{gt}\right)}{c^2}\right) + \alpha v \tag{3}$$

$$\mathrm{IoU} = \frac{\mathrm{A}\cap\mathrm{B}}{\mathrm{A}\cup\mathrm{B}} \tag{4}$$

Where IoU is the ratio of the intersection area of two bounding boxes to their union area, $\rho^2(\cdot)$ is the linear distance between the center of the candidate bounding box and the center of the ground truth bounding box, and c^2 is a parameter used for scaling the distance. α is a hyperparameter used to maintain equilibrium between the different components of the loss function, and its formula is given by:

$$\alpha = \frac{v}{(1-IoU)+v} \tag{5}$$

v measures how well the aspect ratio of the bounding boxes is aligned, and its formula is as follows:

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{6}$$

III. IMPROVED MODEL

This chapter provides a detailed discussion of the proposed improved model. Section 3.1 delineates the overall framework of the improved model, setting the foundation for the subsequent detailed discussions. In Section 3.2, the focus shifts to the newly integrated Convolutional Neural Network module, ConvNeXt V2, providing insights into its novel contributions. Following this, Section 3.3 delivers a comprehensive introduction to the attention lightweight module, namely C2f_GhostNetV2, which is derived from the C2f module proposed in this study. Finally, Section 3.4 discusses the implementation and significance of the Partial Convolution (PConv) within the context of the improved model. Each section systematically addresses critical components, thereby enhancing the clarity and logical cohesion of the proposed model.

A. Improved YOLOv8n Model

The improved algorithm structure based on YOLOv8n, illustrated in Figure 1. To begin, a novel ConvNeXtV2 backbone network is introduced, which incorporates Global Response Normalization (GRN) to balance the amplitude differences between feature maps to improve the network's stability and robustness in capturing multi-scale features. This enhancement empowers the model to extract contextual features and enhance detection capabilities for road damage. Subsequently, the paper advocates for the incorporation of a C2f_GhostNetV2 block, a modified version of the C2f block, in the neck portion. The C2f GhostNetV2 block aims to minimize computational expenses without compromising the original representation capacities. Finally, the PConv module is introduced in the neck section to efficiently capture and process spatial information while minimizing redundant computations. This strategy enables the preservation of high FLOPS levels while reducing overall FLOPs.

B. ConvNeXt V2 Backbone

Backbone networks are essential for feature extraction in convolutional neural networks (CNNs). Specifically, in CNNs, the backbone network incrementally abstracts and extracts image features through successive layers, leading to the formation of more meaningful and effective feature representations for subsequent tasks. YOLOv8n, for instance, utilizes CSPDarkNet as its backbone network, which incorporates the C2f structure for feature extraction. Although this configuration enhances the network's capability to extract features, it also results in a high parameter count. To address this issue and improve the extraction of contextual features,

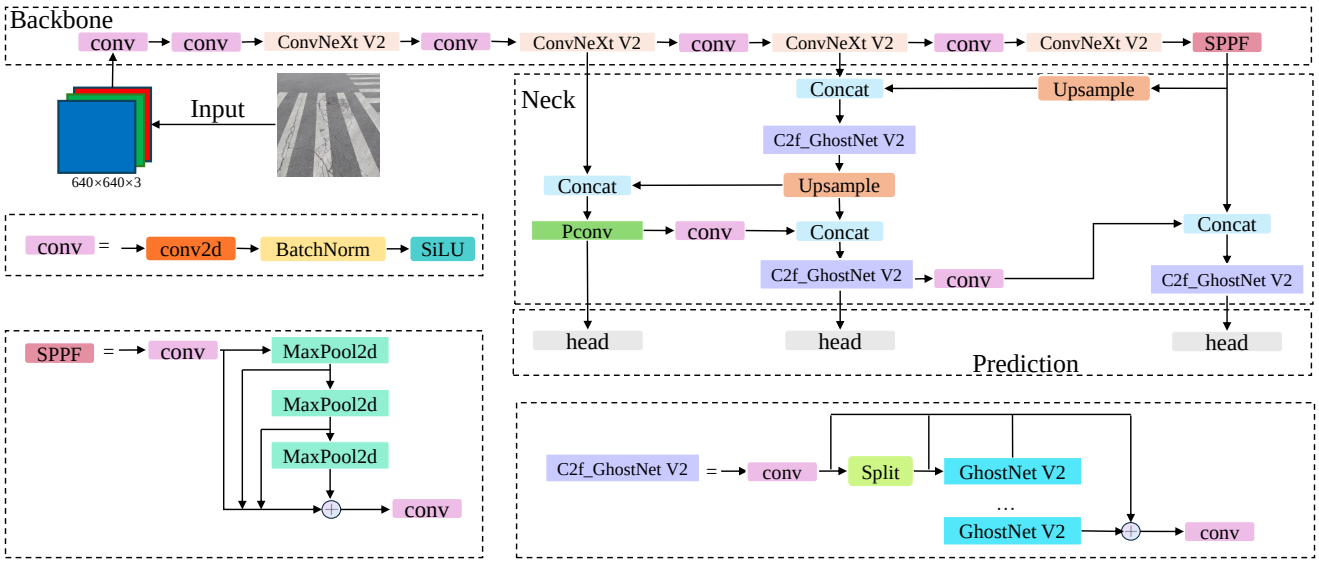


Fig. 1: The improved YOLOv8n structure diagram (redrawing based on [28])

this study introduces the ConvNeXt V2 backbone network structure as an alternative.

ConvNeXt V2 , the second iteration of the ConvNeXt model [29], offering significant improvements in the performance of computer vision tasks. The new backbone network is implemented by replacing the C2f module in the original backbone. ConvNeXt, inspired by Swin Transformers, is a purely convolutional model that originates from ResNet50. One of the key features of ConvNeXt V2 is its reverse bottleneck structure, which is similar to the structure found in Transformers' MLPs, where the fully connected layers possess a dimensionality four times that of the endpoints. Originally proposed in ResNet with a (large-small-large) size configuration, the reverse bottleneck structure in MobileNetV2 was adjusted to a (small-large-small) size format. The purpose of this adaptive approach is to prevent information loss during data transmission between multi-dimensional features and to maintain information integrity. Furthermore, leveraging the power of Transformers to capture long-range correlations, which often employ large windows like 7x7 or even 12x12, this paper increases the kernel size of the depthwise convolution from 3x3 to 7x7 to adopt a similar strategy. In addition to these architectural changes, several intra-layer optimizations are implemented. GELU is utilized in place of ReLU, which results in a reduction in the number of activation functions and normalization layers required. Layer normalization (LN) replaces batch normalization (BN), and a separate subsampling layer is added to enhance performance.

To enhance contrast and selectivity among channels, Global Response Normalization (GRN) aims to achieve three key steps: Global feature aggregation, Feature normalization, and Feature calibration [29]. Initially, the global function $G()$ is used to aggregate the input feature map X_i and convert it into a vector gx. This process can be mathematically represented by the formula:

$$\mathcal{G}(X) := X \in \mathcal{R}^{H\times W\times C} \rightarrow gx \in \mathcal{R}^{C} \tag{7}$$

The input feature map is denoted as $X \in \mathcal{R}^{H\times W\times C}$, where H represents the height, W represents the width, and C represents the number of channels in the feature map. Improved feature aggregation is achieved through L2 normalization. Here is a set of aggregated values $G(X) = gx = \{\|X_1\|, \|X_2\|, \ldots, \|X_C\|\} \in \mathcal{R}^{C}$, where $G(X)_i = \|X_i\|$ represents a scalar of statistical information of the i-th channel being aggregated. After global feature aggregation, the features are subjected to division normalization, which can be expressed using the following formula:

$$\mathcal{N}\left(\|X_i\|\right) := \|X_i\| \in \mathcal{R} \rightarrow \frac{\|X_i\|}{\sum_{j=1,\ldots,C}\|X_j\|} \in \mathcal{R} \tag{8}$$

Here, $\|X_i\|$ represents the L2 norm of the i-th channel. The computed feature normalization score is then applied to adjust the original input responses:

$$X_i = X_i * \mathcal{N}\left(\mathcal{G}(X)_i\right) \in \mathcal{R}^{H\times W} \tag{9}$$

The integration of Global Response Normalization (GRN) refines the ConvNeXt block by making it more attentive to important features while suppressing less significant ones. Serving as an attention mechanism, GRN aids the model in enhancing inter-channel feature competition, thereby facilitating the effective handling of complex data. Through the integration of the ConvNeXt V2 backbone network, this integration potentially improves both generalization performance and computational efficiency.

C. C2f_GhostNetV2 block

The primary function of the C2f module is to enhance the accuracy and stability of object detection by facilitating effective feature fusion. This module aims to bolster the model's expressive power while maintaining a lightweight structure. To achieve these objectives, this paper introduces the C2f_GhostNetV2 module, as shown in Figure 1. This

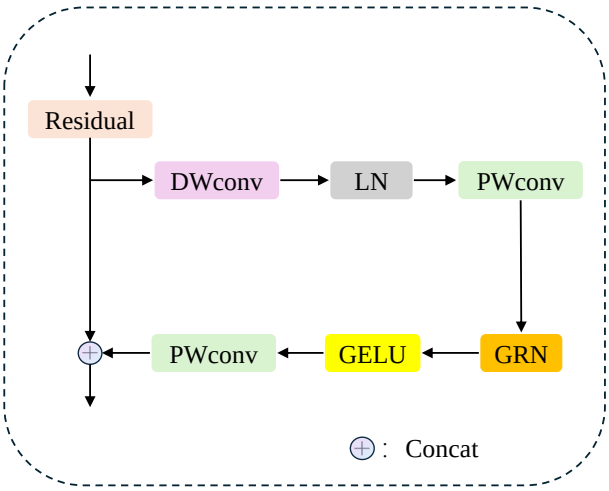


Fig. 2: The design of the ConvNeXt V2 block (redrawing based on [14])

module reduces the model's parameters while ensuring the network's real-time performance.

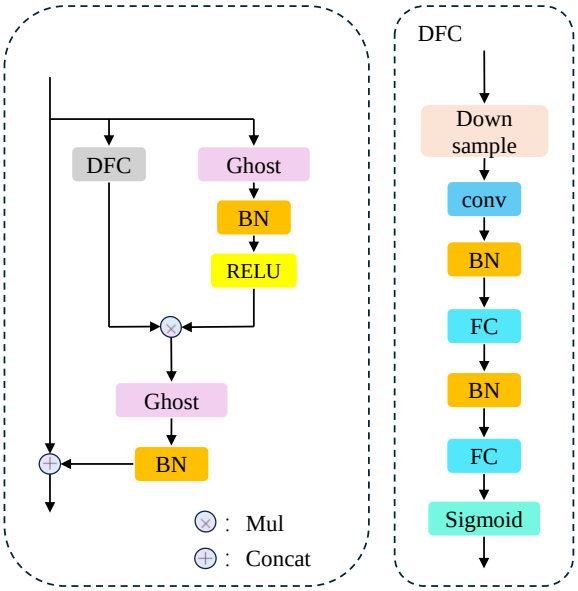


Fig. 3: The structural diagrams of GhostNetV2 (redrawing based on [15])

The GhostNetV2 bottleneck, as shown in Figure 3, replaces the original bottleneck of the C2f module in the C2f_GhostNetV2 module. GhostNetV2, a lightweight convolutional neural network architecture and an upgraded version of GhostNet . By integrating the C2f_GhostNetV2 module, the model achieves enhanced performance while maintaining a relatively low computational complexity and parameter count.

The GhostNetV2 comprises a Ghost module and Decoupled Full Convolution (DFC) attention. GhostNet, a lightweight model designed for efficient inference on mobile devices, consists of these key elements. The Ghost module is a core component that generates more feature maps with fewer computational costs compared to traditional convolution methods. In a typical scenario, a Ghost module takes an input feature $X \in \mathcal{R}^{H\times W\times C}$with height H, width W, and channels C, and replaces it through two steps. Firstly, a 1 × 1 convolution is employed to create the eigenfeature.

$$Y' = X * F_{1\times 1} \tag{10}$$

Here, * denotes the convolution operation. $F(1 \times 1)$ represents pointwise convolution, and $Y' \in \mathcal{R}^{H\times W\times C_{\text{out}'}}$ is the intrinsic feature, which is generally smaller in size than the original output feature. Inexpensive operations are then employed to create additional features derived from the original intrinsic feature. The two feature parts are concatenated along the channel dimension, the specific operations are as follows:,

$$Y = \text{Concat}\left([Y', Y' * F_{dp}]\right), \tag{11}$$

Where F_{dp} signifies the depthwise convolution filter, $Y \in R^{(H\times W\times C_{\text{out}})}$ represents the output feature. While the Ghost module significantly reduces computational costs, it may compromise representational power.

The GhostNetV1 bottleneck is structured by combining two Ghost modules as residual blocks. The first Ghost module functions as an expansion layer, enlarging the output channel size. The next Ghost module reduces the channel size to align with the shortcut path. In contrast, GhostNetV2 incorporates attention mechanisms to enhance the output feature information, thereby facilitating the capture of long-range dependencies among various spatial pixels. For this purpose, features are input through two branches. In one branch, the Ghost module generates output features Y, while the other branch utilizes the DFC module to create attention maps A in both horizontal and vertical . This configuration can be succinctly represented as follows:

$$\begin{aligned} \boldsymbol{a}'_{hw} &= \sum_{h'=1}^{H} F^{H}_{h,h'w} \odot \boldsymbol{z}_{h'w}, h = 1, 2, \cdots, H, w = 1, 2, \cdots, W, \\ \boldsymbol{a}_{hw} &= \sum_{w'=1}^{W} F^{W}_{w,hw'} \odot \boldsymbol{a}'_{hw'}, h = 1, 2, \cdots, H, w = 1, 2, \cdots, W, \end{aligned} \tag{12}$$

Here, $F_{\text{H}}^{\top}$ and $F_{\text{W}}^{\top}$ denote the transformation weights, and $\odot$ represents element-wise multiplication. Z is the original feature input, and $A = \{a_{11}, a_{12}, \ldots, a_{H-W}\}$ is the resulting attention map.The C2f_GhostNetV2 module replaces the traditional bottleneck of the C2f module with a GhostNetV2 bottleneck. This approach enhances the model's performance while keeping the computational complexity and parameter count low.

D. Pconv

Various CNN variants are tailored to solve specific issues by modifying convolutional operations or network structures, aiming to boost the network's performance and efficiency. Among these, MobileNet, ShuffleNet, and GhostNet leverage depthwise convolution and group convolution to capture feature information. However, despite efforts to decrease FLOPs, operators often encounter heightened memory access issues, which can impede performance. Another notable

variant is PConv , a convolutional operation that efficiently captures spatial feature feature information by minimizing redundant computations, thereby refining the computational efficiency of neural networks. The central concept of PConv involves integrating a mask into the convolution operation; this enables the distinction between damaged and intact regions in the image, amplifying the model's responsiveness to these areas. Consequently, the selection of the most suitable CNN variant in practical scenarios can significantly enhance outcomes for particular tasks. Moreover, this mechanism aids in advancing the model's overall performance.

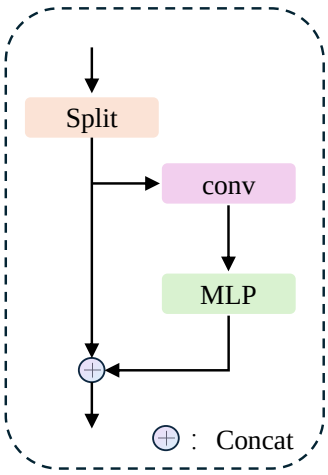


Fig. 4: The design of the PConv (redrawing based on [16])

By applying filters to only a subset of the input channels, PConv offers a fast and efficient alternative to standard convolution operations by decoupling channel and spatial dimensions, as shown in Figure 4. This approach results in fewer floating point operations (FLOPs) compared to regular convolution; however, it incurs more FLOPs than depth-wise or grouped convolution methods. Specifically, for spatial feature extraction, PConv employs regular convolution on selected input channels, ensuring that the rest remain unaffected.

IV. EXPERIMENTS

A. Database Preparation

This study utilizes the Road Damage Detection dataset (RDD2022) to evaluate the effectiveness of proposed improvements on YOLOv8n algorithm performance. RDD2022 was established to facilitate the Road Damage Detection Challenge (CRDDC2022) and provide resources for road damage detection tasks. The dataset comprises a total of 47,420 road images sourced from six countries: China, India, Czech Re public, Norway, the United States and Japan. These images are meticulously annotated and feature four distinct types of road damage: D00 (longitudinal crack), D10 (transverse crack), D20 (alligator crack), and D40 (pothole). Out of the 36,000 annotated images, around 14,700 were deemed empty due to the absence of identifiable objects. Consequently, two specific subsets, China and the United States, were selected. These subsets represent images captured from various viewpoints, including motorcycles, drones, and cars, providing a diverse range of perspectives for the analysis. To facilitate experimentation, the dataset is segmented into a training set and a validation set using a 9:1 split. The training set comprises 7,434 images, while the validation set consists of 826 images.

B. Experimental Environment and Parameter Configuration

We utilized YOLOv8n as the baseline network model. The detailed configuration of the experimental environment is provided in Table I.

TABLE I: Experimental environment configuration

Environmental Parameter	Value
Operation platform	Ubuntu18.04
Deep learning framework	Pytorch
programming language	Python3.8
GPU	RTX 3090
CPU	Intel(R) Xeon(R) Platinum 8255C
RAM	32GB

Hyperparameters were used consistently in the training of all experiments. Table II lists the specific hyperparameters utilized in the training process.

TABLE II: Hyperparametric configuration

Hyperparameters	Value
Learning Rate	0.01
Image Size	640 × 640
Momentum	0.937
Batch Size	64
Epoch	300
Weight Decay	0.0005

C. Evaluation Index

To validate whether the proposed improvements can enhance the performance of the YOLOv8n algorithm, specific evaluation criteria are employed. These evaluation metrics include accuracy, recall rate, mAP, parameter count, FLOPs, and FPS. Accuracy is a crucial metric that assesses how well the model predicts positive classes. It measures the ratio of true positives among the samples that the model predicts as positive. The formula for calculating accuracy is provided in equation (15).

$$\text{Precision Score } = \frac{TP}{(FP+TP)} \tag{13}$$

Where TP represents the number of targets correctly detected. FP represents the number of background or negative class predictions erroneously classified as targets. However, in addition to TP and FP, recall is another crucial metric. The formula for recall, which measures the model's ability

to successfully identify positive examples from all actual positive instances, is given by:

$$\text{Recall } = \frac{TP}{TP + FN} \tag{14}$$

Where FN represents the number of undetected target samples. Mean Average Precision (mAP) is a comprehensive evaluation metric [30]. The calculation formula is as follows:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \tag{15}$$

Where AP_i represents the average precision of a certain class, and n represents the number of classes. FLOPs is an indicator used to measure model complexity and computational requirements. The number of parameters is directly proportional to FLOPs, meaning that more parameters require more computational resources. FPS is used to measure the processing speed of the model. Specifically, FPS reflects the time required for the algorithm to process each frame of an image [31].

D. Experimental Results

To illustrate the performance changes of the model, Precision-Recall (P-R) curves were employed, with precision plotted on the horizontal axis and recall on the vertical axis. This graphical representation not only visually showcases the model's variations in precision and recall but also conveys its recognition performance on positive examples. Additionally, the area under the curve serves as a metric for assessing model performance, with a larger area indicating better performance. After conducting experiments on the RDD2022 dataset [18], Precision-Recall (P-R) curves were generated and depicted in Figure 6. Figure 6(a) shows the P-R curve for the original algorithm, while Figure 6(b) depicts the P-R curve for the improved algorithm. In the figures, the blue curve denotes the mAP@0.5 for all categories, while the curves of other colors represent the mAP@0.5 values for individual categories. Comparing the results, the mAP@0.5 for the blue curve increased from 73.1% in Figure 6(a) to 75.2% in Figure 6(b), indicating a 2.1% improvement. Furthermore, the enhanced YOLO8n algorithm achieved a reduction of 24% in parameter count and 1.6G in FLOPs compared to the original algorithm. These data demonstrate the effectiveness of the enhanced YOLOv8n algorithm presented in this study.

E. Ablation Study

For the optimization of the YOLOv8n algorithm, this paper implements three key measures. Ablation experiments were conducted to showcase the impact of each measure on the original algorithm. Firstly, the ConvNeXt V2 backbone network was utilized. Next, all C2f blocks in the Neck part of the original algorithm were replaced with C2f_GhostNet V2 blocks. Additionally, PConv was integrated into the original algorithm. Furthermore, an integration approach was employed by combining the ConvNeXt V2 backbone network and C2f_GhostNet V2 block. Finally, the ConvNeXt V2 backbone network, C2f GhostNet V2 block, and PConv were collectively applied in the YOLOv8n algorithm to assess the overall effectiveness of integrating each module. The ConvNeXt V2 backbone structure improved mAP@0.5 by 1.7%, reduced parameters by 0.3M, decreased FLOPs by 0.7G, and increased FPS by 2. The addition of the C2f GhostNet V2 block boosted mAP@0.5 by 1.4%, cut down parameters by 0.5M, decreased FLOPs by 0.7G, and increased FPS by 7. Similarly, the integration of PConv raised mAP@0.5 by 1.9%, reduced parameters by 0.1M, decreased FLOPs by 0.2G, and increased FPS by 10. The experimental data clearly illustrates that the inclusion of the C2f_GhostNet V2 block notably reduces model complexity. Furthermore, introducing PConv in conjunction with the three improvement modules in the original algorithm mitigates the decline in mAP@0.5 observed when the first two modules are combined. The enhanced model, in comparison to the original one, exhibits fewer parameters, decreased complexity, and enhanced detection performance. The experimental results are presented in Table III, further validating the effectiveness of the algorithm developed in this study.

This study achieved a 2.1% improvement in mAP@0.5 while also reducing the parameter count by approximately 24%. Despite a slight increase in FLOPs, the model's detection performance remained unaffected, showcasing the efficacy of reducing the parameter count. This demonstrates that the proposed improvements strike a balance between detection accuracy and real-time performance.

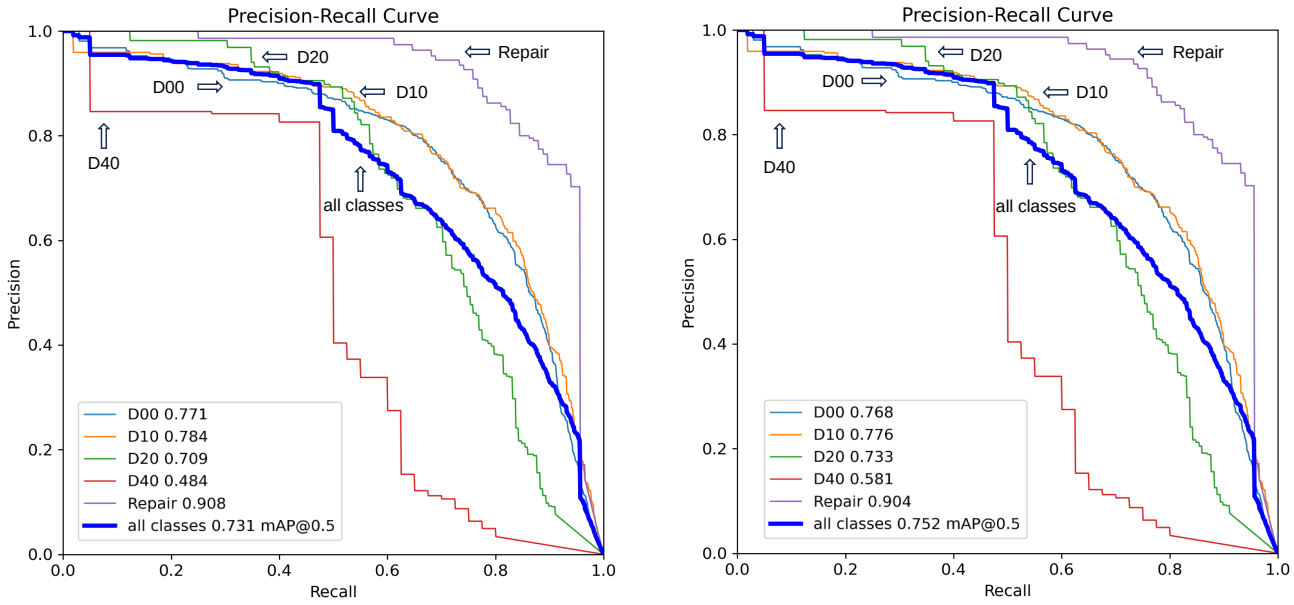
To provide more direct evidence of the effectiveness of different improvement modules, as illustrated in Figure 7, four images were randomly selected. Heatmaps were then used to display the performance changes of the algorithm before and after the improvements. It can be observed that the baseline model extracts features across a wide range, rather than focusing specifically on road damage targets. By incorporating the ConvNeXt V2 module, significant improvement is seen in addressing the issue of wide feature extraction. Further enhancement in performance is attained by integrating the C2f_GhostNet V2 block with the bias convolution, reaching optimal effectiveness.

F. Comparison with Mainstream Algorithms.

We conducted experimental validation on the RDD2022 dataset. The evaluation process involved comparisons with several algorithms, namely Fast-RCNN, YOLOv4-tiny [32], YOLOv5s, YOLOv6s [33], YOLOX [34], YOLOv7-tiny [35], YOLOv8s, and YOLOv8n. The proposed model outperforms other algorithms in terms of mAP@0.5 and recall rate, as summarized in Table IV. Although the proposed model exhibits slightly lower accuracy compared to certain other algorithms, it benefits from a reduced number of parameters. In fact, there is evidence to suggest a slight improvement in performance.

G. Detection on Random Images

Figure 8 illustrates the detection results of the YOLOv8n and the enhanced YOLOv8n algorithms applied to the RDD2022 dataset. The first three images within this figure depict the detection outcomes generated by the original YOLOv8n algorithm. In contrast, the latter three images exhibit the results produced by the enhanced YOLOv8n algorithm. The comparative analysis of these images demonstrates that the enhanced YOLOv8n algorithm exhibits superior detection performance.



(a) YOLOv8n-P-R curve before improvement

(b) YOLOv8n-P-R curve after improvement

Fig. 5: Comparison of P-R curves for the YOLOv8n algorithm before and after improvement on the RDD2022 dataset

TABLE III: ABLATION EXPERIMENT OF YOLOv8n ALGORITHM ON RDD2022 DATASET

YOLOv8n	ConvNeXt V2	C2fGhostNet V2	PConv	P(%)	R(%)	mAP@0.5(%)	Params(M)	FLOPs(G)	FPS
√				78.5	65.5	73.1	3	8.2	126
√	√			79.1	67.5	74.8	2.7	7.5	128
√		√		82.3	67.9	74.5	2.5	7.5	133
√			√	77.5	69.7	75	2.9	8	136
√	√	√		78.4	68.6	74.6	2.3	6.8	129
√	√	√	√	80.4	67.3	75.2	2.3	6.6	130

V. Conclusion

This paper introduces an enhanced road damage detection algorithm that builds on the YOLOv8n framework to achieve superior performance compared to existing mainstream object detection algorithms. The primary innovation is the introduction of a novel ConvNeXt V2 backbone network structure, which significantly improves the network's capacity to extract contextual feature information while simplifying its complexity. This structural modification enhances detection accuracy and real-time performance. Additionally, we have integrated a novel C2f_GhostNetV2 block structure within the network's neck, which amplifies feature representation and reduces computational burdens, facilitating efficient model inference on portable devices. Furthermore, the incorporation of PConv enhances the extraction of spatial features, reduces redundant computations, and strengthens the model's detection performance and robustness. As a result of these enhancements, our algorithm outperforms several prevalent object detection algorithms in terms of detection precision. Furthermore, lightweight optimizations have been implemented to ensure real-time performance, thereby making this algorithm particularly suitable for practical applications.

References

[1] X. Wang, H. Gao, Z. Jia, and Z. Li, "Bl-yolov8: an improved road defect detection model based on yolov8," *Sensors*, vol. 23, no. 20, p. 8361, 2023.

[2] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo: From yolov1 to yolov8 and beyond," *arXiv preprint arXiv:2304.00501*, 2023.

[3] C. M. Richard, K. Magee, P. Bacon-Abdelmoteleb, J. L. Brown *et al.*, "Countermeasures that work: A highway safety countermeasure guide for state highway safety offices, 2017," United States. Department of Transportation. National Highway Traffic Safety ..., Tech. Rep., 2018.

[4] M. E. Torbaghan, W. Li, N. Metje, M. Burrow, D. N. Chapman, and C. D. Rogers, "Automated detection of cracks in roads using ground penetrating radar," *Journal of Applied Geophysics*, vol. 179, p. 104118, 2020.

[5] G. M. Hadjidemetriou, P. A. Vela, and S. E. Christodoulou, "Automated pavement patch detection and quantification using support vector machines," *Journal of Computing in Civil Engineering*, vol. 32, no. 1, p. 04017073, 2018.

[6] T. S. Nguyen, S. Begot, F. Duculty, and M. Avila, "Free-form anisotropy: A new method for crack detection on

TABLE IV: PERFORMANCE COMPARISON WITH MAINSTREAM ALGORITHMS

Model Name	P(%)	R(%)	mAP@0.5(%)	Params(M)	FLOPs(G)	FPS
Faster R−CNN [11]	71.8	52.7	63.83	28.3	47.4	24
YOLOv4−tiny [32]	74.6	55.3	64.7	5.9	6.8	147
YOLOv5s	88.5	52.7	74.5	7	16.5	119
YOLOv6s [33]	88.9	56.8	74.9	16	45	91
YOLOX [34]	86.3	58.4	75.4	5	15.4	79
YOLOv7−tiny [35]	79.1	57.4	74.43	6.02	13.2	101
YOLOv8s	81.7	66.3	75.1	11.1	28.7	105
YOLOv8n	78.5	65.5	73.1	3	8.2	126
ours	80.4	67.3	75.2	2.3	6.6	130

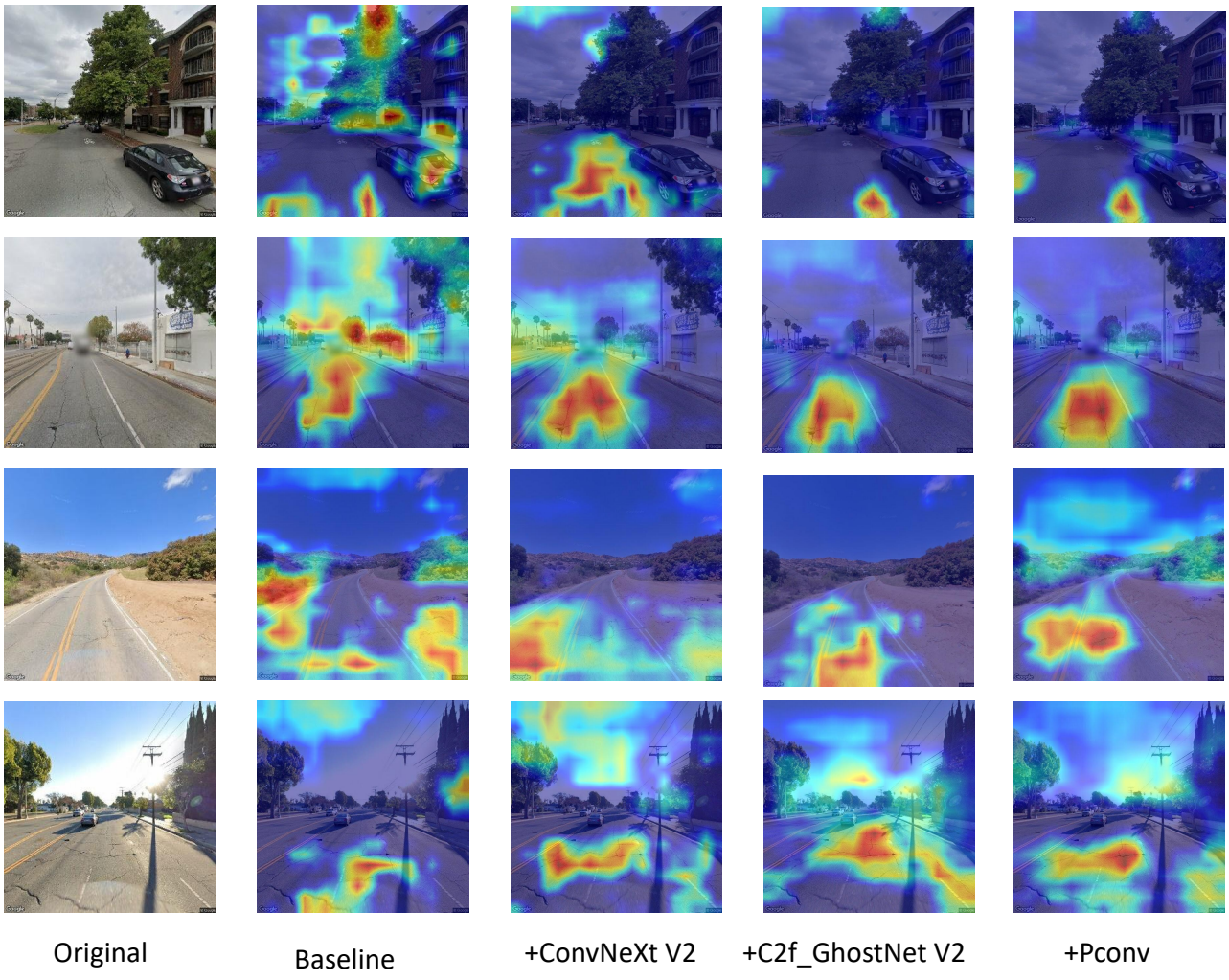


Fig. 6: Heat map comparisons after adding different improvement modules

pavement surface images," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 1069–1072.

[7] H. Nguyen, L. Nguyen, and D. N. Sidorov, "A robust approach for road pavement defects detection and classification," *Journal of Computational and Engineering Mathematics*, vol. 3, no. 3, pp. 40–52, 2016.

[8] N. Safaei, O. Smadi, A. Masoud, and B. Safaei, "An automatic image processing algorithm based on crack pixel density for pavement crack detection and classification," *International Journal of Pavement Research and Technology*, vol. 15, no. 1, pp. 159–172, 2022.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and

Fig. 7: Comparison of detection results between YOLOv8n and the improved YOLOv8n algorithm

semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[10] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

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

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[14] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 133–16 142.

[15] Y. Tang, K. Han, J. Guo, C. Xu, C. Xu, and Y. Wang, "Ghostnetv2: enhance cheap operation with long-range attention," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9969–9982, 2022.

[16] 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," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 021–12 031.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[20] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *2008 IEEE conference on computer vision and pattern recognition*. Ieee, 2008, pp. 1–8.

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

[22] S. Shim, J. Kim, S.-W. Lee, and G.-C. Cho, "Road damage detection using super-resolution and semi-supervised learning with generative adversarial network," *Automation in Construction*, vol. 135, p. 104139, 2022.

[23] S. Naddaf-Sh, M.-M. Naddaf-Sh, A. R. Kashani, and H. Zargarzadeh, "An efficient and scalable deep learning approach for road damage detection," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 5602–5608.

[24] 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 Advances in Signal Processing*, vol. 2022, no. 1, p. 98, 2022.

[25] L. Haohan, F. Yiming, H. Huaiqing, and H. Kanghua, "Improved yolov7-tiny's object detection lightweight model." *Journal of Computer Engineering & Applications*, vol. 59, no. 14, 2023.

[26] 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*, 2017, pp. 2117–2125.

[27] 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*, 2018, pp. 8759–8768.

[28] R. King, "Brief summary of yolov8 model structure· issue# 189· ultralytics/ultralytics," 2023.

[29] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[30] S. Li and W. Liu, "Small target detection model in aerial images based on yolov7x+." *Engineering Letters*, vol. 32, no. 2, pp. 436–443, 2024.

[31] X. Zhang and T. Ying, "Traffic sign detection algorithm based on improved yolov8s," *Engineering Letters*, vol. 32, pp. 168–178, 2024.

[32] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[33] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[34] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[35] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7464–7475.