A Lightweight Method for Road Damage Detection Based on Improved YOLOv8n

Xudong Li, Yujun Zhang*

Abstract—**Current road damage detection models encounter substantial challenges in striking an optimal balance between precision and processing speed. Furthermore, owing to the extensive number of parameters, these models present significant challenges for effective deployment on edge devices with constrained computational resources. In order to tackle these challenges, this paper introduces a lightweight road damage detection model, BSE-YOLO, which is founded on the optimization of the YOLOv8n architecture. Initially, we reformulate the feature fusion network by integrating the concept of BiFPN to minimize both the parameter count and computational overhead. Subsequently, a novel SC2f module is introduced and integrated into the feature fusion network, thereby further minimizing both the parameter count and computational requirements. Additionally, this study presents the SEAHead module, which makes use of limited computational resources to obtain vital information, consequently improving both the efficiency and precision of detection tasks while reducing computational costs. The experimental results indicate that, in comparison to the original model, the BSE-YOLO algorithm achieves a 40% reduction in parameters, a decrease of 2.2G in FLOPs, an increase of 3 in FPS, and only a 0.1% decline in mAP@0.5. This model effectively fulfills the accuracy and real-time processing demands for road damage detection tasks on mobile edge devices.**

Index Terms—**road damage detection, lightweight, YOLOv8n, edge devices**

I. Introduction

WITH the ongoing development of Chinese society, investments in highway construction have been progressively increasing each year, positioning it as one of the world's foremost road networks. As a cornerstone of sustainable economic development [1], the road network not only meets daily travel needs but also facilitates the smooth operation of commerce and industry. However, the aging infrastructure is gradually showing signs of wear and tear, leading to various road surface defects. To foster the stable growth of the national economy, it is imperative to enhance the maintenance and safeguarding of transportation infrastructure. Among the various types of road damage, pavement cracks are particularly common. They not only severely impact road safety but also diminish driving comfort. Therefore, ensuring the maintenance of high-quality road conditions is a key task in safeguarding road safety. The detection of road damage is crucial for mitigating pavement degradation and ensuring traffic safety [2].

Manuscript received July 12, 2024; revised November 20, 2024. The work was supported by the Intelligent Construction Internet of Things Application Technology Key Laboratory of Liaoning Province, China. The project number is 2021JH13/10200051.

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

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

Initially, road damage assessment typically relied on manual methods, where evaluators conducted on-site inspections to visually identify and record road damage. This approach required significant expertise and experience from the evaluators, necessitated their physical presence, was cumbersome and costly, and entailed safety risks [3]. Additionally, since manual inspections often required road closures for on-site evaluations, they frequently disrupted traffic flow. Given these limitations, traditional manual inspection techniques proved insufficient for large-scale road damage assessments. With the advancement of technology, semi-automatic detection techniques have progressively replaced traditional manual approaches, becoming the mainstream method for road damage assessment. In semi-automatic detection methods, high-speed vehicles outfitted with specialized equipment traverse the roads, automatically capturing images of the roadway. These images are then analyzed by professional technicians for the identification and recording of various types of road damage [4]. Although this method reduces traffic disruption, it remains associated with certain drawbacks, such as cumbersome data processing, limited detection targets, and high equipment costs.

The emergence of machine learning has created new avenues for the implementation of road damage detection and pavement crack classification [5]. The swift advancement of deep learning technology has garnered significant attention across various domains, including object detection. Deep learning architectures have achieved remarkable speed and precision in tasks related to object detection, highlighting their strong performance and ability to generalize. By eliminating manual feature extraction and sophisticated feature segmentation techniques, deep learning diminishes the likelihood of misclassifying or omitting critical target features during the pre-sampling phase of feature extraction. Object detection methodologies leveraging deep learning can typically be classified into two distinct categories: two-stage algorithms and one-stage algorithms. Prominent examples of two-stage algorithms encompass R-CNN [6], Fast R-CNN [7], Faster R-CNN [8] and SPP-Net [9]. Xu et al. [10] integrated the training strategies of Faster R-CNN and Mask R-CNN, achieving remarkable detection performance with a limited number of crack images and yielding favorable detection results. Nevertheless, the method exhibits a comparatively slow detection speed, and its generalization capability requires further enhancement. He et al. [11] introduced a method for road damage detection that integrates Mask R-CNN with transfer learning, demonstrating notable accuracy in detecting damages. Nevertheless, the constrained magnitude of damage persists in presenting challenges for a comprehensive evaluation of the model's efficacy, while the detection speed remains insufficient to fulfill real-time detection criteria. Although two-stage algorithms demonstrate advantages in detection accuracy, their slower speed renders them impractical for real-time detection needs. Single-stage object detection algorithms encompass the You Only Look Once (YOLO) series [12–17], Single Shot MultiBox Detector (SSD) [18] and Retinanet [19]. Nonetheless, the limited degree of damage still poses difficulties for a comprehensive evaluation of the model's effectiveness, and the detection speed is insufficient to meet real-time detection standards. Consequently, single-stage algorithms have garnered heightened interest within the domain of road damage detection. Lu et al. [20] implemented an optimized SSD network for crack detection, aiming to enhance detection accuracy while simultaneously adhering to real-time detection standards. However, in complex scenarios, multi-scale crack detection continues to require enhancement. Ma et al. [21] developed a network that integrates PCGAN and YOLO-MF for crack detection and tracking, capable of delivering accurate and real-time performance when deployed on devices. Wang et al. [22] presented a lightweight crack detection framework that employs a bidirectional network, effectively establishing an optimal equilibrium between inference speed and detection accuracy. While the detection speed has been enhanced, there is a corresponding decline in accuracy, and the complexity of the network structure renders it unsuitable for implementation on mobile devices. Guo et al. [23] introduced an enhanced YOLOv5 framework for road damage detection, employing the lightweight MobileNetV3 as the backbone network to mitigate model complexity. They also implemented cooperative attention mechanisms to augment the network's capability for precise localization of damaged targets, thereby enhancing the accuracy of damage detection. Xia Yu et al. [24] incorporated an attention module into YOLOv7, adjusting the weights of visual features while minimizing the impact of irrelevant features. Tu Chengfeng et al. [25] integrated the lightweight architectures ShuffleNetv2 and GhostNet within the YOLOv5 framework, thus creating an efficient object detection system.

YOLOv8, developed by Ultralytics, the company behind YOLOv5, is an architecture consisting mainly of a backbone and head, with the neck incorporated into the head. The architecture of YOLOv8 incorporates the concept of the CSP module, wherein the C2f module supersedes the original C3 module to further diminish the model's complexity. It retains the Conv and SPPF modules from YOLOv5, but the size of the first convolution's kernel has been reduced. The neck preserves the architecture of the feature pyramid network (FPN) and the path aggregation network (PANet) [26, 27]. Compared to YOLOv5, YOLOv8 eliminates the 1x1 downsampling layer. The head employs an anchor-free design and features a decoupled structure to independently manage classification and regression tasks. This architecture enables each branch to concentrate on its specific task, thereby minimizing the conflict between classification and regression while enhancing the capability to detect irregular objects, ultimately improving the model's overall accuracy. In classification tasks, the binary cross-entropy (BCE) loss function is employed, whereas for regression tasks, both the dual-focal loss (DFL) and complete intersection over union (CIOU) loss functions are utilized.

This paper extends the significant achievements of the YOLOv8 object detection algorithm, aiming to further enhance its performance. While YOLOv8 demonstrates outstanding performance in detection accuracy and inference speed, the subsequent rise in model parameters and the accompanying computational burden have constrained its deployment in real-world applications. In response to these challenges, a lightweight road defect detection model named BSE-YOLO has been developed, which is founded on YOLOv8n and seeks to minimize computational complexity and parameter size while preserving detection accuracy. The enhancements include:

1. Introducing the BiFPN to improve feature extraction capabilities while reducing computational and parameter overhead during feature fusion.
2. The SC2f module is introduced, integrating the StarV2 and C2f modules into a novel feature extraction network within the bidirectional feature pyramid, aimed at enhancing detection efficiency while reducing both parameters and computational requirements.
3. Introducing the SEAHead detection module to further reduce computational and parameter requirements in the head section.

II. IMPROVED MODEL

In Chapter 2, this paper provides an in-depth explanation of the proposed enhancements to the model. Section 2.1 describes the overall structure of the BSE-YOLO model, establishing the basis for the detailed analysis in subsequent sections. Section 2.2 introduces the BiFPN (Bi-directional Feature Pyramid Network), which plays a crucial role in improving lightweight performance by facilitating efficient feature fusion. Following this, Section 2.3 discusses in detail the SC2f lightweight module, a novel combination of the C2f and StarV2 modules. Finally, Section 2.4 explains the implementation and relevance of the new detection head, SEAHead, within the improved model. Each section systematically elaborates on key components, thereby improving the clarity and coherence of the proposed model.

A. BSE-YOLO

This section introduces the overall structure of the proposed algorithm built on YOLOv8n, as shown in Fig. 1. Initially, the BiFPN is introduced to effectively integrate features from various levels, thereby enhancing feature representation and utilization efficiency [28]. The BiFPN facilitates dual fusion in both top-down and bottom-up directions, thereby enriching the feature set. This bi-directional feature fusion strategy improves the model's recognition accuracy while simultaneously reducing the parameter size. Next, the SC2f lightweight module is incorporated into the BiFPN to strengthen feature representation, reduce computational complexity, and decrease parameter count, thereby increasing detection speed. Ultimately, the SEAHead detection module is utilized to reduce both the computational load and parameter size of the head while maintaining detection efficiency.

B. BiFPN

YOLOv8 employs a feature pyramid architecture to extract multi-scale features, constructing a Feature Pyramid Network (FPN) that utilizes feature maps at various scales to facilitate the detection of objects with differing sizes. Specifically,

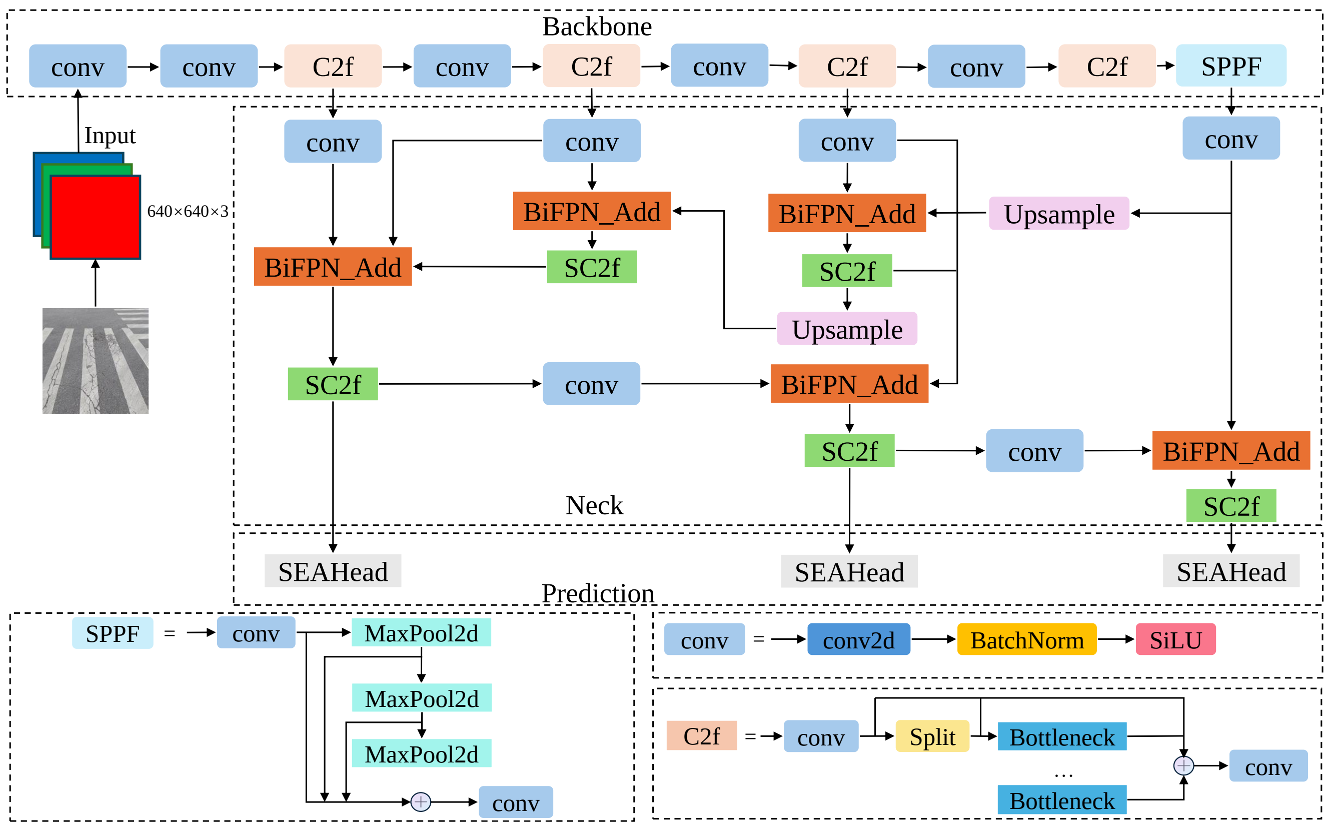
Backbone
conv
C2f
SPPF
Input
640×640×3
Neck
BiFPN_Add
SC2f
Upsample
SEAHead
Prediction
SPPF = conv MaxPool2d conv
conv = conv2d BatchNorm SiLU
C2f = conv Split Bottleneck ... Bottleneck conv

Fig. 1: Network architecture of the BSE-YOLO

YOLOv8 incorporates the bottom-up PANet architecture as its neck network and integrates it with a top-down FPN to enable the prediction layer to access both high-level semantic information and low-level spatial details. However, in the context of road damage detection, road cracks are typically characterized by their elongated and discontinuous nature, with the damaged areas often being small and exhibiting a slender shape. Thus, the network requires strong feature extraction capabilities to efficiently manage these feature complexities. BiFPN, a neural network architecture designed for object detection, enhances feature extraction and is therefore more suitable for road damage detection tasks. To achieve improved feature fusion, the BSE-YOLO algorithm introduces BiFPN and incorporates modifications to the BiFPN connection structure, optimizing the selection of connection pathways, as shown in Fig. 2.

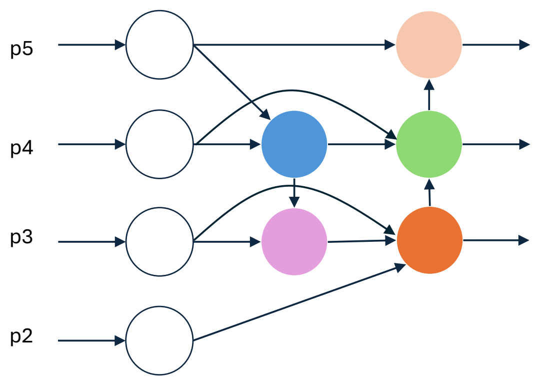

Fig. 2: BiFPN network architecture

BiFPN is a type of FPN architecture developed to optimize feature fusion in object detection architectures. Fig. 2 illustrates the network architecture of BiFPN. It utilizes two primary pathways to combine features across different scales: the top-down path transmits high-resolution feature maps from the top of the pyramid downward to enrich contextual information for lower-resolution features. The bottom-up pathway consolidates information from low-resolution feature maps and propagates it upward to refine the details in high-resolution features, thereby further enhancing the efficacy of object detection. BiFPN addresses the limitations of traditional FPN by introducing information exchange paths both top-down and bottom-up, enabling more efficient feature aggregation and contextual information flow. Through adaptive feature fusion and selection, BiFPN effectively mitigates issues such as information bottlenecks and feature distortion inherent in feature pyramid networks.

The fundamental concept underlying the BiFPN architecture lies in its innovative approach to integrating bidirectional information flow with rapid normalization fusion. In the process of merging low-level and high-level features, BiFPN employs learnable weights to assess the significance of various input features, rather than merely executing a summation or concatenation. This methodology effectively leverages the integration of bidirectional cross-scale connections alongside fast normalization fusion within BiFPN. Subsequently, we illustrate the fusion process of two features at level 4 in the

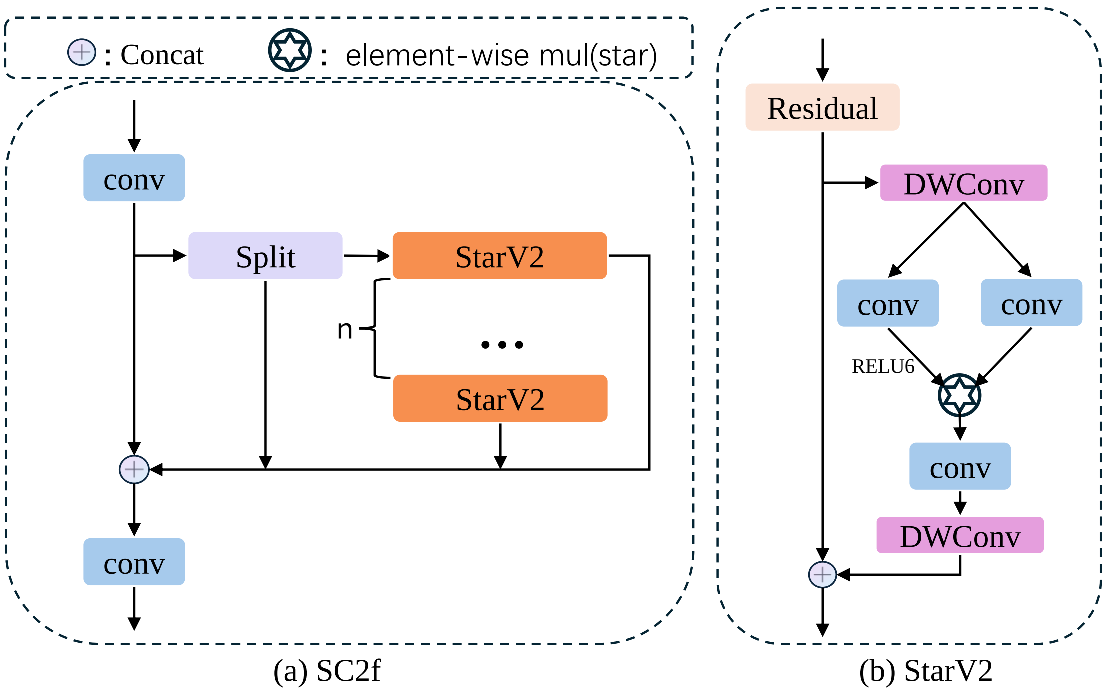

Fig. 3: The SC2f module and the StarV2 module

BiFPN framework, as depicted in Fig. 2:

$$P_4^{mid} = \mathcal{C}onv\left(\frac{w_1 \cdot P_4^{in} + w_2 \cdot \mathcal{R}esize\left(P_5^{in}\right)}{w_1 + w_2 + \epsilon}\right)$$

$$P_4^{\text{out}} = \mathcal{C}onv\left(\frac{w_1' \cdot P_4^{in} + w_2' \cdot P_4^{mid} + w_3' \cdot \mathcal{R}esize\left(P_3^{\text{out}}\right)}{w_1' + w_2' + w_3' + \epsilon}\right) \tag{1}$$

Where P_4^{mid} denotes the intermediate feature at the 4th level of the top-down pathway, and P_4^{out} signifies the output feature at the 4th level of the bottom-up pathway. w^i denotes a learnable weight. $\epsilon = 0.0001$ serves as a small constant to mitigate numerical instability. The $Resize$ operation commonly encompasses upsampling or downsampling to facilitate resolution alignment, whereas $Conv$ typically denotes convolutional operations employed for feature extraction.

C. *SC2f module*

The C2f module serves as a crucial element within the YOLOv8 neural network architecture, specifically engineered for feature extraction and information fusion. The design of the C2f module is inspired by the strengths of ResNet and DenseNet, augmenting the network's expressive power and performance through improved feature fusion and information flow. Through the integration of multiple convolution operations and feature fusion, the C2f module extracts multi-scale, multi-level features, thereby enriching feature representation. By leveraging feature fusion and residual connections, the C2f module enhances information flow, ensuring smoother communication between layers and aiding in the maintenance of gradient flow while mitigating gradient vanishing. Nevertheless, in the context of road damage detection, the intricate nature of road backgrounds across various regions presents significant challenges. Cracks often display elongated and discontinuous features, while damages tend to be slender and small; additionally, varying weather conditions, such as overcast skies and intense sunlight, further complicate crack detection. These factors present challenges for the effective deployment of the model on edge devices.

In order to facilitate lightweight road damage detection, the BSE-YOLO model integrates the SC2f module. This module utilizes the StarV2 component, which, optimized from the original Star module [29], substitutes fully connected layers with 2D convolutions and employs the SiLU activation function in place of the GELU activation function, as illustrated in Fig. 3 (b). The SC2f module replaces the original Bottleneck in C2f with the StarV2 component. The central design concept of the StarV2 component is derived from the Star module in StarNet, as depicted in Fig. 3 (a).

The SC2f module efficiently diminishes both the computational burden and the parameter count while preserving the model's capacity for expression.StarNet is a neural network model that maps inputs to high-dimensional nonlinear feature spaces using element-wise multiplication (called star-shaped operation) to promote efficient and compact network structures. The Star module constitutes the core component of the StarNet architecture. Element-wise multiplication integrates features from different subspaces through element-by-element multiplication, serving as an implicit high-dimensional space mapping operation. This operation is termed the "star operation" because the symbol for element-wise multiplication resembles a star. Traditional deep learning methods increase network depth to map input features from low to high dimensions, which significantly raises model complexity and computational load. Unlike traditional neural networks that expand network width (channels), the star operation functions similarly to a kernel function by performing pairwise feature multiplication across different channels, particularly resembling polynomial kernels. When implemented in neural networks and arranged in multiple layers, each layer facilitates an exponential increase in implicit dimensional complexity. Even with only a few layers of star-shaped operations, it can attain nearly infinite dimensions within a compact feature space. The distinctive advantage of star-shaped operations resides in their capacity to execute computations within a compact feature space while leveraging implicit high-dimensional representations. By utilizing the star operation, StarNet can surpass several meticulously designed efficient models, including MobileNetv3 [30], EdgeViT [31], and FasterNet [32].

In the context of a neural network's single layer, the asterisk operation is commonly represented as $(W_1^T X + B_1) * (W_2^T X + B_2)$, indicating the element-wise multiplication of two features obtained from linear transformations. The weight matrix and bias may also be combined into a unified entity, identified as $W = [W, B]^T$, and likewise referred to as $X = [X, 1]^T$, thus enabling the creation of the asterisk operation $(W_1^T X) * (W_2^T X)$. To simplify our analysis, we consider a scenario with a single-output channel transformation and a single-element input.Specifically, let w_1, w_2, $x \in \mathbb{R}^{(d+1)\times 1}$ be defined, with d denoting the quantity of input channels. This formula can be expanded to support multiple output channels W_1,$W_2 \in \mathbb{R}^{(d+1)\times(d'+1)}$, while utilizing X $\in\mathbb{R}^{(d+1)\times n}$ to process various feature elements. Typically, the STAR operation can be expressed as follows:

$$\begin{aligned} w_1^{\mathrm{T}} x * w_2^{\mathrm{T}} x &= \left(\sum_{i=1}^{d+1} w_1^i x^i\right) * \left(\sum_{j=1}^{d+1} w_2^j x^j\right) \\ &= \sum_{i=1}^{d+1}\sum_{j=1}^{d+1} w_1^i w_2^j x^i x^j \end{aligned} \tag{2}$$

This operation facilitates the fusion of features through element-wise multiplication, effectively capturing interactions between different feature elements and enabling the network to map inputs to high-dimensional, nonlinear feature spaces efficiently.

D. SEAHead module

In the YOLOv8 model, the detection layer plays a pivotal role, as it directly influences the model's capacity to accurately identify and localize various objects within an image. Specifically, this layer produces predicted bounding boxes along with their associated class probabilities. This process involves complex calculations, including extracting information from feature maps and converting it into the final output format.

In road damage detection tasks, particularly with the RDD2022 dataset, data is collected from various countries using different devices and under diverse conditions, leading to substantial variations in image quality. Some images may be of low resolution or blurry, while others may depict deteriorated road conditions, which directly impact the algorithm's recognition and classification capabilities. Additionally, some images exhibit interference from elements such as sand or snow on the road, along with challenging weather conditions like overcast skies or intense sunlight, which can further constrain the model's detection performance.

In order to tackle these challenges, the BSE-YOLO model incorporates the SEAHead attention detection head, which integrates attention mechanisms across spatial positions and output channels. By employing attention mechanisms, the model integrates information from the input data more effectively, enabling a better understanding of contextual elements within the images. This allows the model to emphasize areas of road damage while de-emphasizing background regions. This method proves to be especially effective in addressing the intricate shapes and contexts of objects within road damage images. The SEAHead enhances the stability of road damage detection and assists the network in rapidly focusing on target areas, thereby avoiding redundant calculations in non-relevant regions. The configuration of this module is illustrated in Fig. 4 (a).

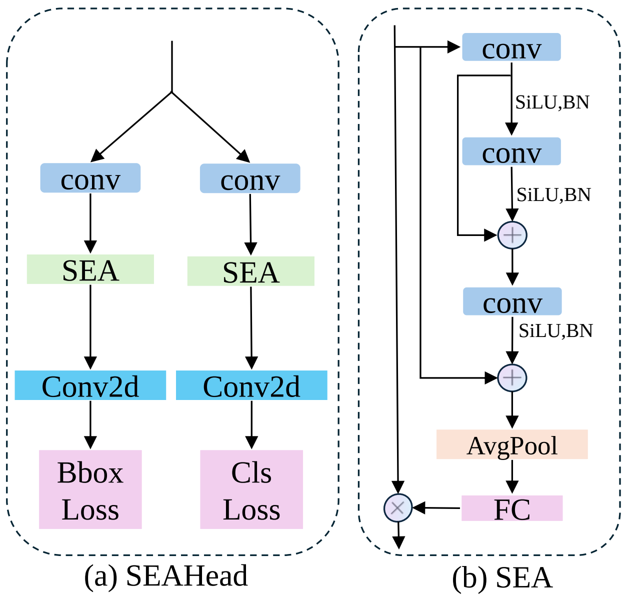

Fig. 4: The structural of SEAHead and SEA

The SEA module, inspired by the SEAM design concept [33], substitutes the original multi-head attention mechanism with 2D convolution, depthwise separable convolution, and point convolution. It also employs the SiLU activation function, as illustrated in Fig. 4 (b). In 2D convolution, each kernel processes information across all input channels simultaneously, capturing complex inter-channel relationships and dependencies more effectively. This unified approach prevents potential spatial information loss that may arise when convolution operations are decomposed into multiple steps. Capturing inter-channel interactions in a single operation enhances the network's feature extraction capabilities. This method is more straightforward and eliminates the need for additional steps to combine results from different convolutions, thereby simplifying the computational process. The SiLU activation function is preferred over GELU due to its simplicity, superior performance, broad applicability, and robustness. These attributes render SiLU particularly suitable for road damage detection tasks.

In conclusion, the SEA attention mechanism significantly enhances feature representation and localization accuracy, effectively minimizing redundant computations while augmenting the model's generalization capability and detection performance for small targets. The SEAHead detection head effectively reduces the computational and parameter load, facilitating the lightweight implementation of road damage detection.

III. Experiments and Analysis

A. Introduction to Database

To assess the efficacy of the proposed BSE-YOLO model, an experimental validation was performed utilizing the RDD2022 road damage dataset. The RDD2022 dataset was developed specifically for the Challenge on Road Defect Detection (CRDDC2022) and is carefully crafted to enhance deep learning approaches aimed at detecting road defects. This dataset comprises 47,420 annotated road images, encompassing more than 55,000 instances of road damage from six countries: Japan, India, the Czech Republic, Norway, the United States, and China. The dataset includes nine categories of road damage: D00 (longitudinal cracks), D01 (construction joint part), D10 (transverse cracks), D11 (construction joint part), D20 (alligator cracks), D40 (potholes), D43 (crosswalk blur), D44 (white line blur), and D50 (manholes). For this study, images from Japan, India, the Czech Republic, and Norway were selected. The dataset is partitioned into a training set and a validation set in an 8:2 ratio, consisting of 16,832 images designated for training and 4,208 images allocated for validation.The selected images cover diverse road damage scenarios, offering a comprehensive test bed for assessing the performance of the improved YOLOv8n algorithm.

B. Experimental Environment and Parameter Configuration

The experimental framework was developed on Ubuntu 18.04 in conjunction with the PyTorch deep learning library, employing YOLOv8n as the core model. A comprehensive configuration of this experimental environment is detailed in Table 1.

TABLE I: Setup and training environment

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

The same hyperparameters were consistently applied across all training experiments. Table II presents the specific hyperparameters employed in the training process.

C. Evaluation Index

To evaluate the effectiveness of the proposed enhancements in improving the performance of the YOLOv8n algorithm, specific evaluation metrics were utilized. The evaluation metrics encompass accuracy, recall, mean average precision (mAP), parameter count, floating-point operations per second (FLOPs), and frames per second (FPS). Accuracy

TABLE II: Hyperparametric configuration

Hyperparameters	Value
Learning Rate	0.01
Image Size	640 × 640
Momentum	0.937
Batch Size	64
Epoch	200
Weight Decay	0.0005

serves as a crucial metric for assessing a model's effectiveness in predicting positive classes. It quantifies the ratio of true positive samples to the total number of positive predictions generated by the model. The calculation of accuracy is expressed in Equation (3).

$$\text{Precision Score } = \frac{TP}{(FP+TP)} \tag{3}$$

Where TP signifies the number of true positives, referring to instances that the model has accurately identified as targets. FP represents false positives, which are individuals erroneously classified as targets by the model due to either background classes or misclassified negative predictions. Additionally, recall rate is a vital metric closely linked with both TP and FP; it quantifies the model's effectiveness in identifying all relevant instances, as demonstrated in Equation (4).

$$\text{Recall } = \frac{TP}{TP+FN} \tag{4}$$

Where FN denotes the count of false negatives, referring to instances that the model failed to identify. The mean Average Precision (mAP) serves as a metric for average precision and is commonly employed to assess the detection performance of the model across various categories. It is a comprehensive metric for performance evaluation. The formula for calculating mAP is shown below:

$$mAP = \frac{1}{n}\sum_{k=1}^{n} AP_k \tag{5}$$

Where AP_i denotes the average precision for a specific category, while n represents the total number of categories. FLOPs are utilized to assess the computational complexity of the model. The relationship between the number of parameters and FLOPs is positively correlated, indicating that an increase in parameter count necessitates greater computing resources. FPS (frames per second) serves as a metric for evaluating the real-time performance of object detection algorithms. Specifically, FPS indicates the number of images that can be processed and detected by the model each second.

D. Experimental Results

In order to visually assess the performance enhancement of the BSE-YOLO model, we conduct an analysis of its accuracy and recall rates through the P-R curve. Precision is represented on the x-axis, while recall is positioned on

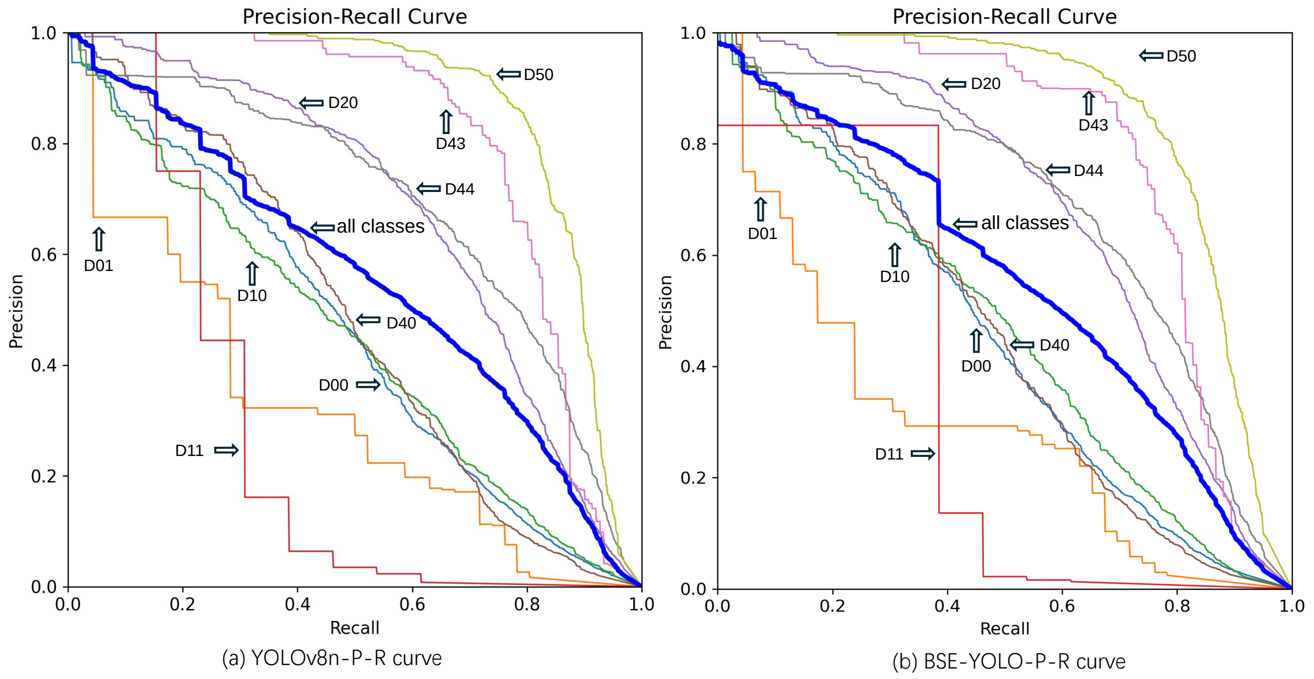

Fig. 5: Compare the P-R curves of YOLOv8n and BSE-YOLO on the RDD2022 dataset

TABLE III: Ablation experiment

YOLOv8n	BiFPN	SC2f	SEAHead	P(%)	R(%)	mAP@0.5(%)	Params	FLOPs(G)	FPS
√				60.7	50.5	55.5	3,007403	8.1	126
√	√			58.3	53.5	55	1,993079	7.1	128
√		√		56.5	53.3	55.6	2,704657	7.2	127
√			√	54.6	54.8	55.7	2,819243	7	127
√	√	√		60.2	51.4	55.7	1,963799	7	130
√	√	√	√	59	52.1	55.4	1,775639	5.9	129

the y-axis, thereby providing a clear depiction of changes in model performance and its capacity to detect positive instances. Furthermore, the area under the P-R curve serves as a significant metric for evaluating the model's effectiveness, with a larger area signifying superior performance. Fig. 5 illustrates the P-R curves derived from experiments performed on the RDD2022 dataset. Fig. 5 (a) presents the P-R curve of the original YOLOv8n algorithm, whereas Fig. 5 (b) demonstrates the P-R curve of the BSE-YOLO algorithm. The blue curve illustrates the average precision across all categories at a threshold of mAP@0.5. Other colored curves indicate the mAP@0.5 for individual classes. In Fig. 5 (a), the blue curve presents an mAP@0.5 of 55.5%, whereas in Fig. 5 (b), the mAP@0.5 is 55.4%. Despite a minor reduction of 0.1% in mAP@0.5, the parameter count was reduced by 40%, and FLOPs decreased by 2.2 GFLOPs. This illustrates the efficiency of the lightweight design implemented in the proposed BSE-YOLO algorithm.

E. Ablation Study

The BSE-YOLO algorithm introduces three key optimization measures to enhance the YOLOv8n algorithm. To demonstrate the effectiveness of each optimization measure on the original algorithm, the following ablation experiments were conducted:

1.Incorporating the BiFPN feature fusion network into the foundational algorithm.

2.Replacing all C2f modules in the Neck section of the foundational algorithm with SC2f modules.

3.Incorporating the SEAHead detection head into the original algorithm.

4.Combining the BiFPN Feature Pyramid Network and SC2f modules into the original algorithm.

5.Applying the BiFPN Feature Pyramid Network, SC2f modules, and SEAHead detection head simultaneously in the YOLOv8n algorithm to assess the overall effectiveness of these modules integrated into the original network.

The proposed BiFPN Network reduced the number of parameters by approximately 1 million, decreased mAP@0.5 by 0.5%, decreased FLOPs by 1 billion, and increased

TABLE IV: Performance comparison of mainstream algorithms

Model Name	P(%)	R(%)	mAP@0.5(%)	Params	FLOPs(G)	FPS
Faster-RCNN [8]	62.7	59.3	60.2	38.3	47.4	51
YOLO-LRDD [34]	59	58.2	57.6	19.8	17.4	86
YOLOv5s	61.9	56.5	56.7	9	17.5	90
YOLOv6s [17]	63.6	53.4	57.9	16	44.5	66
YOLOv7-tiny [35]	59.2	51.6	56.3	6.1	13.2	101
YOLOv8s	60.4	57.7	58.2	11.1	28.7	80
YOLOv8n	60.7	50.5	55.5	3	8.1	126
YOLOv9-T [36]	57.2	53	56.2	2.6	10.7	130
YOLOv10n [37]	60.5	54.5	54.5	2.2	6.5	137
FasterNet-YOLOv8n	55.6	48.5	52.3	1.7	5.1	133
BSE-YOLO(ours)	59	52.1	55.4	1.7	5.9	129

FPS by 1. Incorporating the SC2f module into the original algorithm increased mAP@0.5 by 0.14%, reduced the number of parameters by approximately 0.3 million, decreased FLOPs by 0.9 billion, and increased FPS by 2. Integrating the SEAHead detection head into the algorithm increased mAP@0.5 by 0.2%, reduced the number of parameters by 0.2 million, decreased FLOPs by 1.1 billion, and increased FPS by 1. Experimental data indicate that incorporating the BiFPN Feature Pyramid Network significantly reduces model complexity. The incorporation of the SC2f module and SEA-Head detection head significantly mitigated the decline in mAP@0.5 attributed to the BiFPN feature pyramid network, and upon fully integrating these three enhanced modules into the original algorithm, both the parameter count and computational load were reduced. In comparison to the initial model, the BSE-YOLO algorithm features fewer parameters, reduced complexity, and maintains nearly identical detection performance. The experimental results presented in Table III substantiate the effectiveness of the proposed enhancements to the original algorithm.

In conclusion, this study achieved a reduction of approximately 40% in parameter count and 2.2 billion FLOPs compared to the original algorithm, with a slight reduction of 0.1% in mAP@0.5. This suggests that the suggested enhancements had a negligible effect on the model's detection capabilities, while considerably decreasing both the number of parameters and computational demands. These modifications not only lowered the model's parameter count and processing requirements but also preserved detection accuracy, achieving an effective balance between real-time performance and precision in detection.

F. Compared with the Performance of Advanced Object Detection Algorithms

To validate the advantages of the proposed BSE-YOLO algorithm for road damage detection, we conducted experiments using the RDD2022 dataset. The evaluation compared BSE-YOLO with several algorithms, including Faster R-CNN [8], YOLO-LRDD [34], YOLOv5s, YOLOv6s [17], YOLOv7-tiny [35], YOLOv8s, YOLOv8n, YOLOv9-T [36] and YOLOv10n [37]. Additionally, one lightweight models based on YOLOv8n were FasterNet-YOLOv8n. The performance of these models was assessed by evaluating detection accuracy, recall rate, model parameter count, floating-point operations per second (FLOPs), mean average precision (mAP) at an IoU threshold of 0.5, and frames per second (FPS). The detailed results are presented in Table IV. While YOLOv9 and YOLOv10 represent the latest iterations of the model, their enhancements yield superior performance when assessed against consistent evaluation metrics. As shown in Table 4, the proposed model demonstrates superior performance compared to other algorithms regarding parameters, computational complexity, and frames per second (FPS). While its accuracy is marginally lower than that of certain other algorithms, the detection capability of the model remains largely unaffected.

G. Detection on Random Images

Fig. 6 illustrates the detection results of both YOLOv8n and the BSE-YOLO algorithm on the RDD2022 dataset. The top three images in Fig. 6 display the detection results of the YOLOv8n algorithm, while the bottom three images present the results of the BSE-YOLO algorithm. The comparison of these images indicates that the BSE-YOLO algorithm achieves similar detection performance to the original YOLOv8n algorithm.

IV. Conclusion

This study introduces a lightweight algorithm for detecting road damage, referred to as BSE-YOLO, which is built upon the YOLOv8n framework and exhibits enhanced overall performance when compared to existing mainstream object detection algorithms. The principal advancements introduced in this research encompass the incorporation of a BiFPN feature pyramid network, which enables more efficient feature aggregation and context propagation. This enhancement enhances the model's recognition capabilities and accuracy while concurrently reducing the number of parameters. Implementing

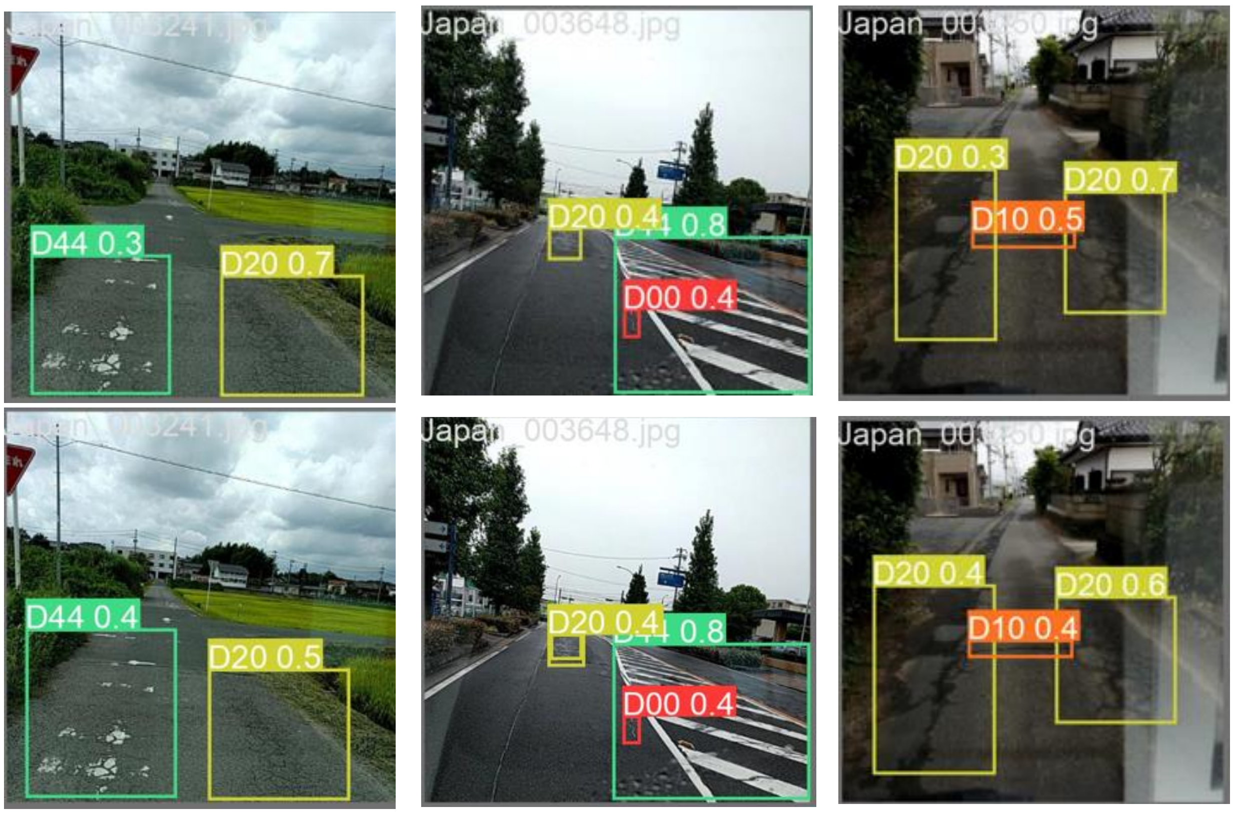

Fig. 6: Different detection results between the YOLOv8n and the BSE-YOLO algorithms

a new SC2f module in the neck section enhances feature representation while minimizing computational and parameter overhead, thereby enabling efficient inference on mobile devices. Integrating the SEAHead detection head improves feature representation and localization accuracy, enhances model generalization and small object detection, reduces redundant computations, and decreases computational and parameter demands in the detection head. These lightweight improvements result in the BSE-YOLO algorithm surpassing some current mainstream object detection algorithms in comprehensive performance. By achieving lightweight optimization and enhancing real-time performance, the model's detection capability is effectively preserved.

References

[1] S. C. Radopoulou and I. Brilakis, "Detection of multiple road defects for pavement condition assessment," *Eindhoven, The Netherlands*, 2015.

[2] S. A. Hosseini and O. Smadi, "How prediction accuracy can affect the decision-making process in pavement management system," *Infrastructures*, vol. 6, no. 2, p. 28, 2021.

[3] A. Chatterjee and Y.-C. Tsai, "A fast and accurate automated pavement crack detection algorithm," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2140–2144.

[4] Y. Du, X. Zhang, F. Li, and L. Sun, "Detection of crack growth in asphalt pavement through use of infrared imaging," *Transportation Research Record*, vol. 2645, no. 1, pp. 24–31, 2017.

[5] A. Daniel and V. Preeja, "Automatic road distress detection and analysis," *International Journal of Computer Applications*, vol. 101, no. 10, 2014.

[6] 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*, 2014, pp. 580–587.

[7] R. Girshick, "Fast r-cnn in proceedings of the ieee international conference on computer vision (pp. 1440–1448)," *Piscataway, NJ: IEEE.[Google Scholar]*, vol. 2, 2015.

[8] 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.

[9] 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.

[10] X. Xu, M. Zhao, P. Shi, R. Ren, X. He, X. Wei, and H. Yang, "Crack detection and comparison study based on faster r-cnn and mask r-cnn," *Sensors*, vol. 22, no. 3, p. 1215, 2022.

[11] Y. He, Z. Jin, J. Zhang, S. Teng, G. Chen, X. Sun, and F. Cui, "Pavement surface defect detection using mask region-based convolutional neural networks and transfer learning," *Applied Sciences*, vol. 12, no. 15, p. 7364, 2022.

[12] 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.

[13] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[14] J. Redmon, "Yolov3: An incremental improvement," *ArXiv Preprint ArXiv:1804.02767*, 2018.

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

[16] J. Glenn, "Yolov5 release v6. 1. 2022," *Source:¡ https://github.com/ultralytics/yolov5/releases/tag/v6*, vol. 1.

[17] 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.

[18] 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.

[19] 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*, 2017, pp. 2980–2988.

[20] G. Lu, X. He, Q. Wang, F. Shao, J. Wang, and Q. Jiang, "Bridge crack detection based on improved single shot multi-box detector," *Plos One*, vol. 17, no. 10, p. e0275538, 2022.

[21] D. Ma, H. Fang, N. Wang, C. Zhang, J. Dong, and H. Hu, "Automatic detection and counting system for pavement cracks based on pcgan and yolo-mf," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22 166–22 178, 2022.

[22] W. Wang and C. Su, "Deep learning-based real-time crack segmentation for pavement images," *KSCE Journal of Civil Engineering*, vol. 25, no. 12, pp. 4495–4506, 2021.

[23] G. Guo and Z. Zhang, "Road damage detection algorithm for improved yolov5," *Scientific Reports*, vol. 12, no. 1, p. 15523, 2022.

[24] Y. Xia, M. Nguyen, and W. Q. Yan, "A real-time kiwifruit detection based on improved yolov7," in *International Conference on Image and Vision Computing New Zealand*. Springer, 2022, pp. 48–61.

[25] C. Tu, A. Yi, T. Yao, and W. He, "High-precision garbage detection algorithm of lightweight yolov5n," *Comput. Eng. Appl*, vol. 59, pp. 187–195, 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] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 781–10 790.

[29] X. Ma, X. Dai, Y. Bai, Y. Wang, and Y. Fu, "Rewrite the stars," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5694–5703.

[30] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[31] J. Pan, A. Bulat, F. Tan, X. Zhu, L. Dudziak, H. Li, G. Tzimiropoulos, and B. Martinez, "Edgevits: Competing light-weight cnns on mobile devices with vision transformers," in *European Conference on Computer Vision*. Springer, 2022, pp. 294–311.

[32] 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.

[33] Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu, and X. Wang, "Yolo-facev2: A scale and occlusion aware face detector," *Pattern Recognition*, p. 110714, 2024.

[34] 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.

[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.

[36] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European Conference on Computer Vision*. Springer, 2025, pp. 1–21.

[37] 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.