

CCD-YOLO: A Lightweight Method for Road Defect Detection

Man Wu, Ye Tao*

Abstract—Striking an ideal balance between precision and real-time efficiency in detecting road defects remains a major challenge in the field. The substantial parameter size of existing models further exacerbates this issue, particularly limiting their deployment on edge devices with constrained computational resources, which consequently restricts their practical applicability. To address these limitations, we propose CCD-YOLO, an innovative lightweight road defect detection model based on YOLOv8s architecture. Our approach incorporates three key innovations: a lightweight Cross-Scale Feature Fusion Network (CFFN) that effectively integrates multi-scale detailed features with contextual information while significantly reducing both parameter count and computational complexity; a novel Cross Stage Partial PConv (CSPPC) module designed to minimize feature map redundancy across different channels, thereby reducing computational overhead and memory access costs; and a Dynamic detection (Dydetect) mechanism that employs multiple attention mechanisms to enhance the detection head's representational capacity. Comprehensive experimental evaluations demonstrate that CCD-YOLO achieves substantial improvements over the original model, including a 32 % reduction in parameter count, a 6.3 G decrease in FLOPs, a 9 FPS increase in processing speed, and a 0.9% improvement in mAP@0.5. These substantial improvements allow CCD-YOLO to successfully address the stringent demands for both precision and real-time capability in detecting road defects, thus confirming the effectiveness of our lightweight architectural design strategy.

Index Terms—road defect detection, lightweight, YOLOv8s, Dynamic detection

I. INTRODUCTION

ROADS constitute a fundamental pillar of modern infrastructure, serving as critical arteries for the movement of people and goods while playing an indispensable role in regional economic development. Their significance extends beyond mere transportation, as they enhance urban-rural connectivity, foster industrial collaboration, and contribute substantially to social progress and economic prosperity. However, the inevitable deterioration of road surfaces through prolonged use presents significant challenges [1]. This degradation stems from multiple interrelated factors: primarily, the continuous increase in traffic volume and operation beyond design capacity leads to structural damage, manifesting as cracks,

potholes, and other surface irregularities. Furthermore, climate change exacerbates these issues, with extreme temperature fluctuations, frequent precipitation, and snowfall imposing both physical and chemical stresses on road materials, thereby accelerating their degradation [2]. Compounding these challenges, the quality of construction materials and the standards of construction techniques significantly influence road durability, where subpar materials or inadequate practices can result in premature aging, cracking, or even structural failure.

Given the critical role of roads in societal infrastructure and the multifaceted nature of their deterioration, the development of efficient and accurate road surface defect detection algorithms has become imperative. Such advanced detection systems would enable the timely identification and resolution of road surface issues, thereby ensuring optimal road functionality and enhancing traffic safety. Moreover, these systems would contribute to extending road lifespan, reducing maintenance costs, and supporting sustainable societal and economic development [3].

Pavement defect detection technology has evolved as a critical tool for identifying and evaluating various types of road surface damage and deficiencies. Historically, this field has progressed through three primary methodologies: manual inspections, sensor-based systems, and image processing techniques. Traditional manual inspection methods, while foundational, are plagued by significant limitations, including inefficiency, reliance on subjective expertise, inability to provide real-time monitoring, elevated safety risks, and vulnerability to environmental disturbances. These methods are inherently time-consuming and labor-intensive, making comprehensive coverage of extensive road networks impractical. Moreover, the subjective nature of manual assessments often leads to inconsistent evaluations, misjudgments, and oversight of critical defects, resulting in delayed responses to emerging issues [4].

Sensor-based detection methods, though offering improved precision over manual techniques, present their own set of challenges. These systems are extremely responsive to environmental factors like temperature and humidity, which can have a substantial impact on their precision. The high costs associated with sensor-based technologies pose economic barriers to large-scale implementation, while their limited sensitivity to minor defects may result in the oversight of subtle but critical issues. Additionally, the installation, calibration, and maintenance of these systems require specialized technical expertise, further complicating their widespread adoption [5].

In response to these limitations, deep learning-based pavement defect detection technology has emerged as a

Manuscript received February 7, 2025. revised April 29, 2025. This work was supported by National Natural Science Foundation of China (62272093), the Economic and Social Development Research Topics of Liaoning Province (2025lslybwzkt-100), and Postgraduate education and teaching reform research project of Liaoning Province (LNYJG2024092).

Man Wu is a graduate student of the School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China. (e-mail: 13889718670@163.com).

Ye Tao is an Associate Professor of the School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China. (Corresponding author to provide phone: +8613304224928; e-mail: taibeijack@163.com).

transformative solution for road maintenance and management. This advanced approach offers several key advantages: the capacity to precisely detect and categorize pavement defects, providing reliable data for timely and targeted repairs [6]; high-precision detection capabilities that ensure even minor defects are identified, significantly enhancing result reliability; robust generalization abilities that enable effective performance across diverse and complex real-world scenarios; a high degree of automation that reduces human intervention, minimizes labor costs, and improves operational efficiency; and the capacity for continuous learning and optimization, allowing for ongoing improvements in detection accuracy and adaptability. These features collectively position deep learning-based methods as a superior alternative for modern pavement defect detection and management.

Since the beginning of the 21st century, deep learning has transformed the domain of object detection, propelled by improvements in computing hardware and the rapid expansion of big data technologies. These advancements have allowed deep learning methods, especially Convolutional Neural Networks (CNNs) [7], to attain remarkable success in image recognition, greatly improving the accuracy and efficiency of object detection systems. Early object detection algorithms primarily relied on manually engineered feature extraction methods, such as the Histogram of Oriented Gradients (HOG) [8]. While these traditional approaches were capable of meeting basic detection requirements, they exhibited notable limitations in complex scenarios, particularly when handling multi-scale and multi-pose objects, where their accuracy and robustness were often inadequate.

On the other hand, object detection algorithms based on deep learning have revolutionized the field by automatically acquiring distinctive image features, leading to higher detection accuracy. A key development in this area was the introduction of the R-CNN series, which implemented a region proposal mechanism [9] and utilized CNNs for feature extraction and classification, representing a substantial improvement in detection performance. Subsequent innovations, such as Fast R-CNN [10] and Faster R-CNN [11], further refined this framework, optimizing detection speed and reducing computational overhead. Notable two-stage algorithms, including Mask R-CNN [12] and Region-based Fully Convolutional Networks (R-FCN) [13], have also demonstrated exceptional performance in various detection tasks. For example, He et al. [14] proposed a pavement defect detection approach based on Mask R-CNN, demonstrating outstanding defect recognition performance. Nevertheless, the model's effectiveness was limited by the small number of defect samples in the training data, and its slower detection speed made it less appropriate for real-time use cases.

Although two-stage detection algorithms achieve superior accuracy, their high computational demands and longer processing times can often impede real-time implementation. Consequently, there is a growing demand for more efficient algorithms that can effectively balance accuracy and speed, particularly in practical scenarios where real-time performance is critical. This requirement

has driven the creation of single-stage detectors and lightweight architectures, which seek to balance high accuracy with computational efficiency.

In recent years, the advent of single-stage detection algorithms, notably YOLO [15] and SSD [16], has significantly advanced real-time object detection technology. These algorithms can handle a significant number of image frames per second (from tens to hundreds) while preserving high accuracy, rendering them highly appropriate for use in autonomous driving, security surveillance, and intelligent transportation systems. Concurrently, research in road defect detection using deep learning has witnessed remarkable progress, with numerous innovative approaches being proposed.

For example, Huang et al. [17] created a lightweight road defect detection model utilizing an improved YOLOv7 architecture. Through the integration of a grouped spatial pyramid pooling module and the application of Ghost convolution techniques, this model successfully reduces both the parameter count and computational complexity to a great extent. To counteract any potential accuracy loss resulting from the lightweight design, the model integrates a spatial channel attention mechanism, thereby improving detection precision. Likewise, Du et al. [18] introduced a lightweight object detection approach built upon the YOLO algorithm, where the detection performance is improved via enhanced feature extraction. More specifically, this approach utilizes a bidirectional feature pyramid network (BiFPN) architecture to enhance feature extraction and incorporates a zoom-loss function to tackle sample imbalance problems, thereby substantially improving the accuracy of road defect detection.

Further advancements include the work of Luo et al. [19], who introduced an enhanced lightweight network architecture by incorporating a feature extraction enhancement module. This module significantly improves the network's feature representation capabilities. Additionally, the network utilizes a vertically connected bidirectional feature pyramid structure to facilitate the effective reuse of feature information across different levels, thereby fully leveraging multi-scale context information. Zhang et al. [20] introduced a lightweight single-stage object detection network called AAL-Net. This network combines a lightweight feature extraction module (LF) and integrates a normalized attention module (NAM) to maintain high precision and reliability during the detection process. Wan et al. [21] designed a lightweight road damage recognition model named YOLO-LRDD. This model incorporates a new backbone network referred to as Shuffle-ECANet and combines a BiFPN feature pyramid network. In order to improve detection performance further, they adjusted the localization loss function to Focal-EIOU, attaining an ideal trade-off between detection accuracy and speed.

In conclusion, although lightweight models have achieved substantial progress in enhancing real-time performance, the challenge of maintaining computational efficiency without sacrificing accuracy continues to be a key focus for future research. Continued innovation and optimization are essential to address this challenge and advance the field of real-time road defect detection.

To address the persistent challenges in road defect detection, this paper selects YOLOv8 from the YOLO series as the foundational algorithm for in-depth investigation. Among the five variants of YOLOv8, YOLOv8s provides a fairly balanced compromise between speed and accuracy, while still possessing considerable potential for additional optimization. Building on this foundation, this study focuses on implementing targeted enhancements to YOLOv8s, aiming to significantly improve its overall performance. As a result, we propose an advanced road defect detection algorithm named CCD-YOLO. The main contributions of this paper can be summarized as follows:

1. **Lightweight Cross-Scale Feature Fusion Network (CFFN):** We design a novel lightweight cross-scale feature fusion network that effectively integrates multi-scale detailed features and contextual information. This advancement significantly cuts down on the number of parameters and computational complexity, all while preserving high detection accuracy.

2. **Cross Stage Partial PConv (CSPPC) Module:** We introduce the CSPPC module to minimize redundancy among feature maps across different channels. This module significantly lowers computational costs and memory access requirements, enhancing the model's efficiency without compromising performance.

3. **Dynamic Detection Module (Dydetect):** A dynamic detection module is integrated, utilizing multiple attention mechanisms to strengthen the representational capabilities of the detection head. This module improves the model's overall detection performance by enabling more precise and robust feature extraction.

Together, these contributions push the boundaries of road defect detection, providing a more efficient and precise solution suitable for real-world applications. Through tackling the limitations of current models, CCD-YOLO showcases substantial enhancements in both computational efficiency and detection accuracy, laying the foundation for more effective road maintenance and management systems.

II. RELATED WORK

YOLOv8 marks a substantial progression in object detection, adhering to the central concept of the YOLO series by treating object detection as a regression problem. This approach facilitates efficient end-to-end detection. The structure of YOLOv8 consists of three key elements: the backbone network, the neck network, and the detection head. Compared to its predecessors, YOLOv8 has introduced substantial improvements across these components, enhancing both performance and versatility.

The backbone network functions as the core component for extracting features from input images. It utilizes deeper and wider convolutional layers to strengthen the model's capacity to capture and extract varied visual details, which in turn enhances feature representation. The neck network, located between the backbone and the detection head, is essential for merging multi-scale feature maps. This integration ensures that the model can capture rich semantic details across multiple levels, which is essential for accurate object detection. Notably, YOLOv8 replaces the C3 module used in YOLOv5 with the C2f structure, which provides a

more robust gradient flow and adjusts the number of channels according to the specific model scale [22].

In the development of the detection head, YOLOv8 incorporates two significant advancements compared to YOLOv5. Firstly, it implements a decoupled head structure, which divides the classification and localization tasks. This departure from the traditional unified target branch structure enhances the model's adaptability to diverse task requirements. Secondly, YOLOv8 moves from an anchor-based system to an anchor-free method. This change streamlines the overall structure, enhances localization precision, reinforces generalization abilities, speeds up training, and supports more efficient deployment.

For positive and negative sample matching, YOLOv8 employs a Task-Aligned Assigner. This assigner quantifies task alignment by integrating higher-order combinations of classification scores and Intersection over Union (IoU), ensuring precise matching of positive and negative samples and optimizing training efficiency. In terms of loss function design, YOLOv8 integrates both classification and regression losses. More specifically, the classification task uses Binary Cross-Entropy Loss (BCE) to assess category confidence, whereas the regression task applies Dual-Focal Loss (DFL) and Complete Intersection over Union (CIoU). These selections are especially advantageous for the anchor-free detection approach, as they strengthen the model's robustness and enhance the precision of the regression task.

In summary, YOLOv8 showcases excellent performance in terms of object detection accuracy, speed, and adaptability, establishing itself as a leading-edge solution for various applications [23].

III. IMPROVED MODEL

A. CCD-YOLO

In order to attain an ideal trade-off between accuracy and real-time performance, while minimizing model parameters for efficient deployment on edge devices, we introduce the CCD-YOLO model. The proposed model is based on YOLOv8s and includes specific improvements in both the neck and head modules, as shown in Figure 1. First, we present a lightweight cross-scale feature fusion network (CFFN). This network boosts the model's ability to handle scale variations and enhances small-object detection by integrating features across different scales, while significantly cutting down on parameters and computational requirements. Second, we introduce the CSPPC lightweight module, which minimizes redundancy within channels by eliminating similar or repetitive information. This approach decreases computational complexity and memory access requirements. Additionally, we incorporate a dynamic detection head (Dydetect) that integrates three attention mechanisms. By focusing on various dimensions, this head improves its expressive power, thereby enhancing detection performance and efficiency. These innovations collectively enable CCD-YOLO to achieve a superior balance between accuracy and real-time performance, making it highly suitable for practical applications in road defect detection and other real-world scenarios.

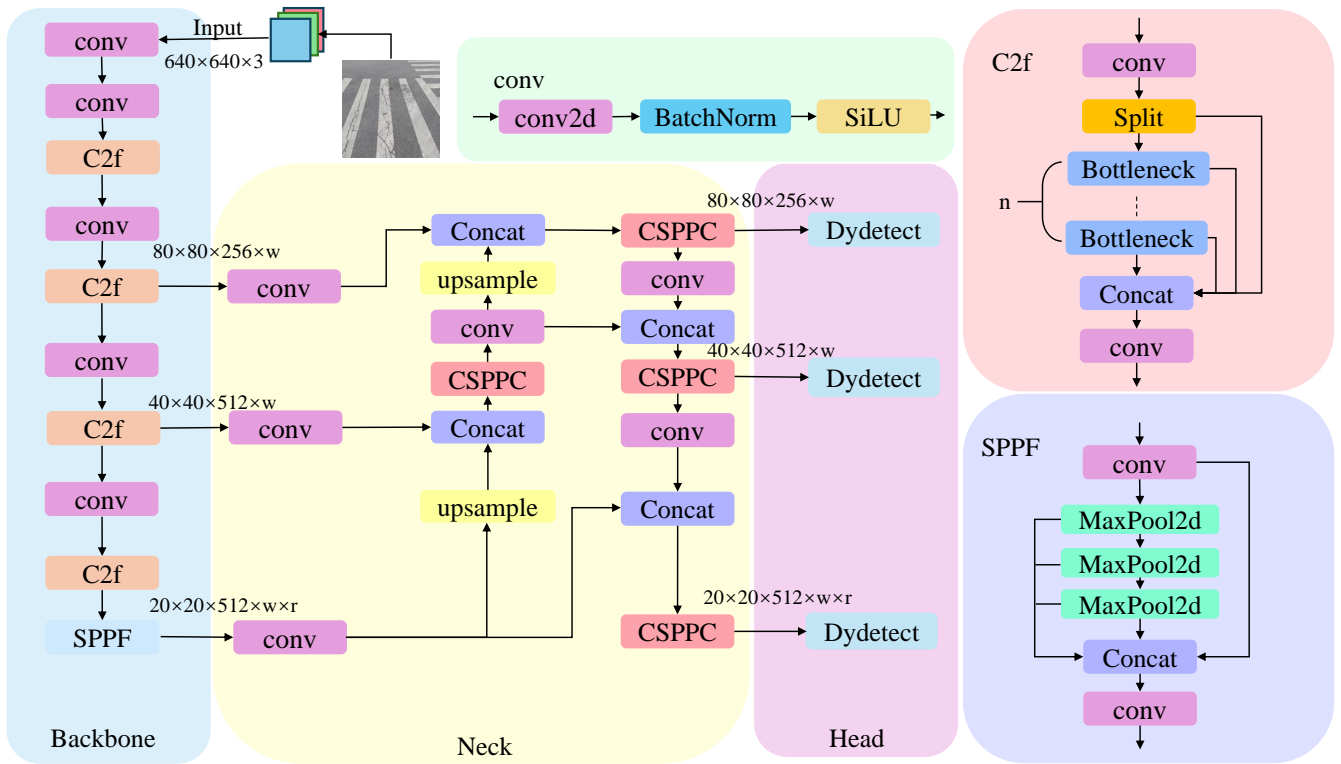


Fig. 1: The architecture of the CCD-YOLO network

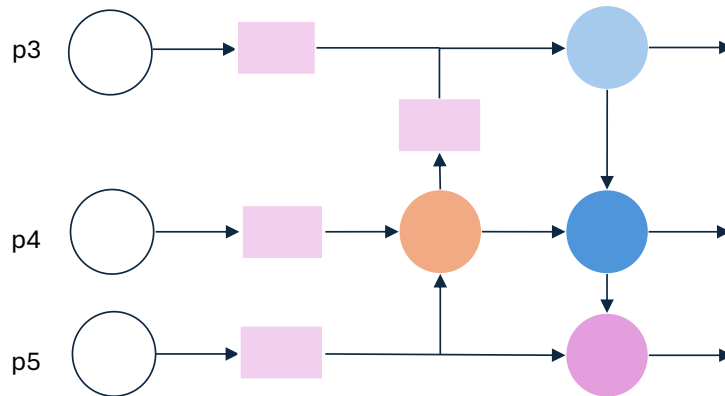


Fig. 2: The structure of CFFN

B. Cross-scale Feature Fusion Network (CFFN)

The neck structure of YOLOv8 mainly utilizes a Path Aggregation Network (PAN-FPN) that combines the Feature Pyramid Network (FPN) [24, 25]. This design facilitates multi-scale feature fusion by aggregating information from different levels of the network. This architecture leverages the complementary advantages of FPN and PAN to merge features from multiple scales, thus strengthening the model’s ability to detect objects at various scales. Through the use of both top-down and bottom-up pathways for feature integration, PAN-FPN successfully consolidates information across different network levels, which enhances detection precision for smaller objects and in intricate scenarios. The primary objective of this mechanism is to optimize feature representation, enabling the network to capture multi-scale

and high-level semantic information more efficiently. However, the incorporation of fusion mechanisms like PAN-FPN can increase computational complexity and memory usage, potentially leading to slower inference speeds, particularly with high-resolution images. This trade-off may compromise real-time performance, which is critical in many practical applications.

In road defect detection tasks, maintaining a balance between high accuracy and real-time performance is crucial. These scenarios often involve complex backgrounds and are influenced by multiple environmental factors. Road defects typically exhibit irregular shapes and sizes, necessitating a feature fusion network capable of effectively integrating multi-scale information. Building on the Cross-Channel Feature Fusion (CCFF) introduced in RT-DETR [26], we propose the Cross-scale Feature Fusion Network (CFFN). To address the trade-off between

accuracy and real-time performance, the CCD-YOLO algorithm adopts CFFN as its neck network. The detailed structure of CFFN is illustrated in Figure 2.

The CFFN is designed to reduce the computational overhead associated with traditional fusion methods while maintaining or even improving detection accuracy. Through the use of CFFN, CCD-YOLO enables effective feature fusion across multiple scales, ensuring reliable performance in identifying road defects with diverse sizes and shapes. This method not only strengthens the model's capacity to manage complex backgrounds and environmental conditions but also guarantees efficient computation, rendering it highly suitable for practical applications like road defect detection. The integration of CFFN allows CCD-YOLO to achieve an ideal trade-off between precision and real-time capabilities, overcoming the constraints of conventional fusion techniques and enhancing its overall applicability.

The Cross-scale Feature Fusion Network (CFFN) is a lightweight yet powerful architecture designed for robust feature extraction and integration across multiple scales. It features a lower number of parameters, reduced computational requirements, and quicker inference times, which makes it especially appropriate for road defect detection tasks. The CFFN architecture is designed to tackle scale variations and improve the detection of small objects by efficiently merging features from various scales.

The Cross-scale Feature Fusion Network (CFFN) is a lightweight architecture designed for robust feature extraction and integration across multiple scales. It features fewer parameters, lower computational costs, and faster inference speeds, making it ideal for road defect detection. CFFN addresses scale variations and enhances small target detection by effectively fusing features from different scales. CFFN combines bottom-up and top-down propagation pathways to integrate detailed features with contextual information. The bottom-up pathway refines high-resolution features to capture fine details, while the top-down pathway enriches low-resolution features with broader context. A key innovation in CFFN is the use of supplementary convolutional operations, which enhance multi-scale feature extraction. By applying multi-level convolutional processing, the model improves its ability to capture and represent features at different scales. In conclusion, the lightweight structure of CFFN and its effective feature fusion capabilities strike an excellent balance between precision and computational efficiency. This makes it especially well-suited for practical applications, including the detection of small and irregularly shaped objects like road defects.

The system utilizes input features from levels 3 to 5, denoted as $P_i^{in} = (P_3^{in}, P_4^{in}, P_5^{in})$, where P_i^{in} represents the input feature at the i -th level. This notation describes the feature fusion mechanism of the CFFN as illustrated in Figure 2:

$$\begin{aligned} P_3^{out} &= \text{conv}(P_3^{in} + \text{Resize}(P_4^{td})), \\ P_4^{td} &= \text{conv}(P_4^{in}) + \text{Resize}(P_5^{in}), \\ P_4^{out} &= \text{conv}(P_4^{td}) + \text{Resize}(P_3^{out}), \\ P_5^{out} &= \text{conv}(P_5^{in}) + \text{Resize}(P_4^{out}) \end{aligned} \quad (1)$$

In this context, P_i^{out} denotes the output feature at the

level, while P_4^{td} represents the intermediate feature at the 4th level. The Resize operation encompasses both upsampling and downsampling processes, and conv refers to the convolution operation. By utilizing its distinctive cross-scale feature fusion mechanism, CFFN substantially improves the model's ability to identify objects at different scales while also minimizing computational costs and parameter complexity. These advantages provide a substantial competitive edge in practical applications, particularly in scenarios demanding real-time performance and high accuracy.

C. CSPPC module

Within YOLOv8, the C2f module is crucial for improving the model's feature representation and effective use of multi-scale information. It achieves cross-layer information fusion, ensuring the preservation of feature details across different scales while optimizing the transmission of contextual data. However, the C2f module's complex architecture and multi-level fusion mechanism increase computational complexity, which can lead to higher computational costs and memory consumption in road damage detection tasks. These drawbacks negatively impact inference speed and real-time performance, posing challenges for practical applications.

To address these limitations while maintaining robust feature extraction and fusion capabilities, we propose the Cross Stage Partial PConv (CSPPC) module, as shown in Figure 3. The CSPPC module enhances the capabilities of the C2f module by integrating residual connections and featuring a significant innovation: substituting the conventional Bottleneck structure with the PConv Bottleneck (PCB) module. The PCB module leverages partial convolution (PConv) technology [27], which significantly reduces computational redundancy while preserving feature integrity. Overall, the CSPPC module represents a balanced approach to feature fusion, combining efficiency with robust feature representation to address the challenges posed by complex detection tasks.

Partial Convolution (PConv) is an efficient convolution technique designed to enhance the inference speed of neural networks by optimizing computational efficiency and reducing memory access overhead. The core principle of PConv lies in minimizing redundant calculations and decreasing memory access frequency by leveraging the inherent redundancies present in feature maps. In numerous situations, some channels within feature maps show high similarity to others, suggesting that processing these redundant features during forward propagation does not add significant information and instead raises computational demands and memory access costs.

To tackle this problem, PConv performs standard convolution operations for spatial feature extraction on a selected subset of input channels while keeping the remaining channels unchanged, as shown in Figure 4. By focusing computations on fewer channels, PConv substantially decreases the number of floating-point operations (FLOPs), thus reducing overall computational complexity. For example, when the partial rate is set to 1/4, the computational cost of PConv is reduced to just 1/16th

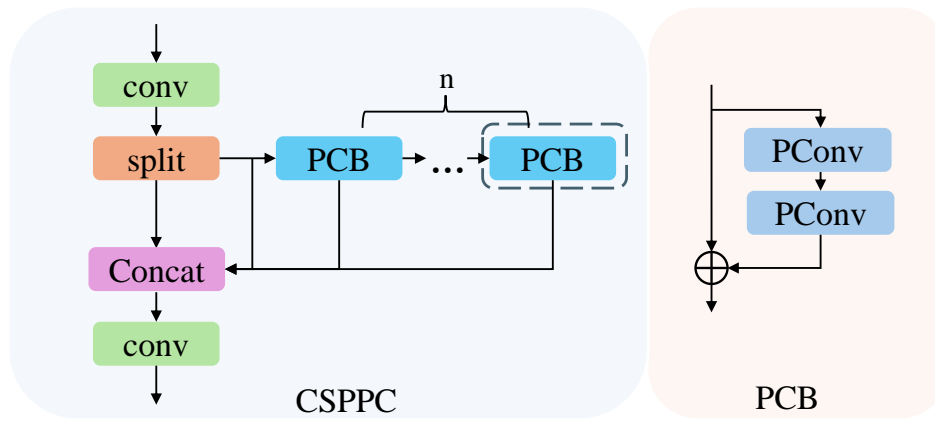


Fig. 3: The structure of the CSPPC module

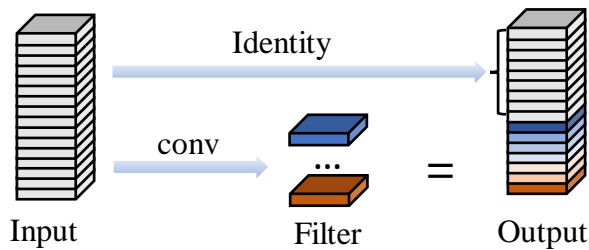


Fig. 4: The structure of the PConv module

of that of traditional convolution methods. This relationship can be mathematically expressed as:

$$PConv_{FLOPs} = h \times w \times k^2 \times c_p^2 \quad (2)$$

Here, h stands for the height of the feature map, w represents its width, k refers to the convolution kernel size, and c_p signifies the number of channels undergoing partial convolution. Compared to standard convolution techniques, PConv significantly reduces memory access requirements, a feature that is particularly beneficial for devices with limited input/output (I/O) resources.

D. Dydetect module

Within YOLOv8, the detection head is essential for deriving the final object detection outcomes from the feature maps produced by the network. These results include object categories, bounding box coordinates, and confidence scores. Through a series of convolutional operations, the detection head computes attributes for each predicted box, such as classification probabilities, localization adjustments, and object confidence levels, enabling accurate object recognition and localization. Its primary function is to translate high-level features extracted by the network into concrete detection outputs, ensuring precise predictions even in complex and diverse scenarios.

However, in road defect detection tasks, several challenges arise due to low image resolution, complex backgrounds, and the inherent characteristics of defects. Cracks, for instance, often exhibit irregular sizes and shapes, with some being extremely fine, making them difficult to detect. Additionally, while lightweight feature fusion networks and modules improve computational efficiency, they may inadvertently reduce recognition accuracy, posing a trade-off between speed and precision.

To address these challenges, we introduce the DyHead block to enhance the capabilities of the detection head, as shown in Figure 5. The DyHead block leverages dynamic attention mechanisms to achieve higher precision and flexibility [28], significantly improving detection accuracy, speed, and adaptability to complex environments. This improvement increases the model's robustness and reliability for real-world road defect detection, especially when dealing with a variety of defect types. By incorporating the DyHead block, the model achieves more stable and accurate results, ensuring its suitability for practical applications where both efficiency and precision are paramount.

The Dynamic Head (DyHead) block combines three specific attention mechanisms: scale-aware, spatial-aware, and task-aware. As shown in Figure 6, the input to the object detection head is expressed as a three-dimensional tensor, with dimensions corresponding to hierarchy, space, and channels. Each dimension is enhanced by its respective attention mechanism, significantly improving both the performance and efficiency of object detection.

Scale-Aware Attention Mechanism: This mechanism dynamically modifies the network's attention weights according to object size, allowing the model to adaptively concentrate on objects of varying scales. By prioritizing features relevant to objects of varying sizes, it significantly improves detection accuracy in multi-scale scenarios, ensuring robust performance across diverse object dimensions. **Spatial-Aware Attention Mechanism:** This mechanism dynamically modulates attention weights in the spatial domain, intensifying focus on critical regions while reducing emphasis on background or irrelevant areas. By optimizing the spatial distribution and extraction of features, it enhances the model's robustness in complex backgrounds and improves its ability to localize objects

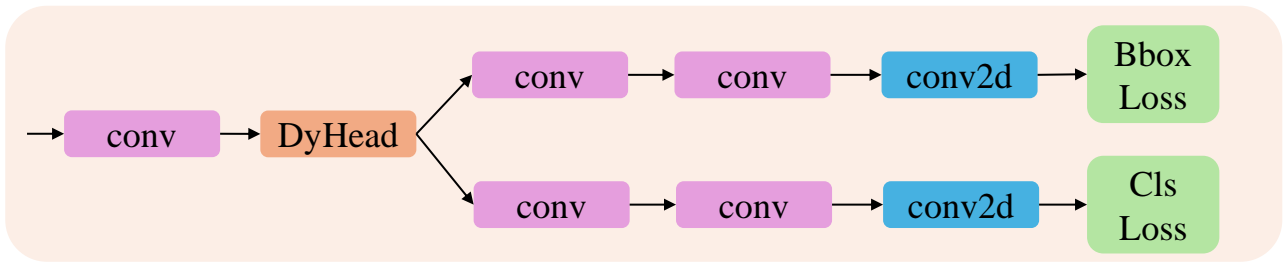


Fig. 5: The structure of the Dydetect module

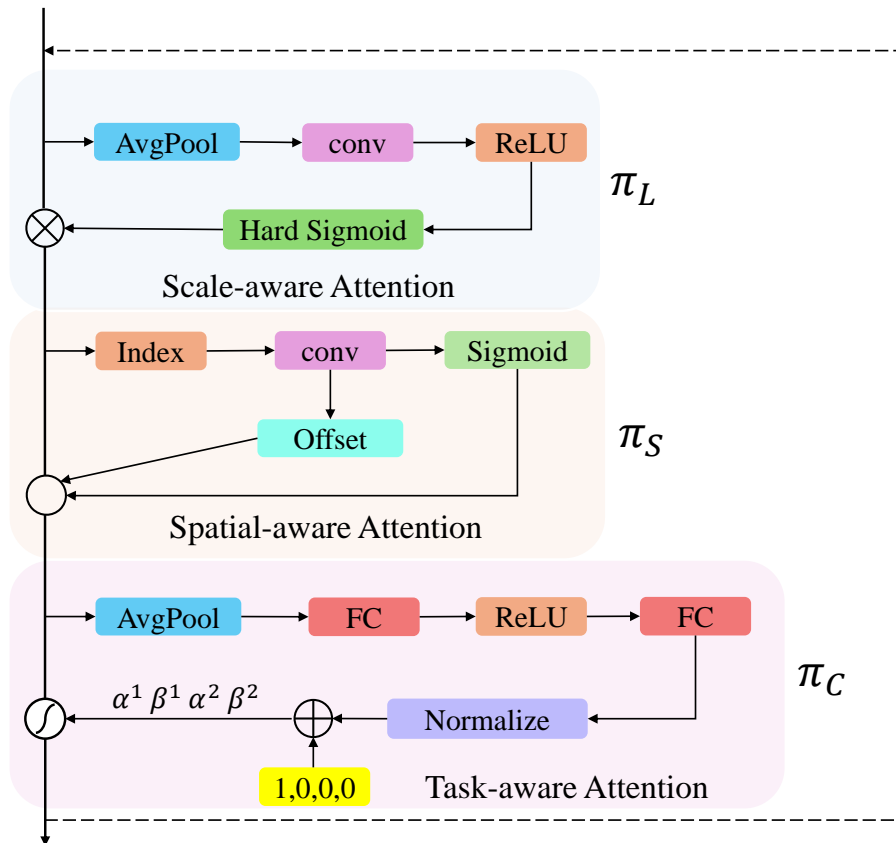


Fig. 6: The structure of the DyHead block

accurately. **Task-Aware Attention Mechanism:** This mechanism dynamically adjusts attention weights according to specific task requirements, such as classification or detection. By precisely regulating the attention given to task-relevant features, it significantly boosts the network's performance in specific tasks, making it more accurate and efficient in processing task-related information.

Together, these three attention mechanisms work synergistically to enhance the effectiveness and adaptability of the DyHead block. Through the combination of scale-aware, spatial-aware, and task-aware attention, the DyHead block achieves excellent performance in various object detection scenarios. This makes it a highly effective tool for applications demanding high precision and efficiency, such as road defect detection.

By uniformly normalizing the feature pyramid to a consistent scale, we assume an input represented as a three-dimensional tensor $F \in R^{L \times S \times C}$, where L denotes

the hierarchical level, S represents the spatial dimension, and C indicates the channel dimension. The equations for the three types of attention mechanisms are as follows:

$$W(F) = \pi_C(\pi_S(\pi_L(F) \cdot F) \cdot F) \cdot F \quad (3)$$

In this scenario, $\pi_L(\cdot)$, $\pi_S(\cdot)$, and $\pi_C(\cdot)$ denote the attention functions corresponding to three different dimensions. More specifically, the scale-aware attention mechanism, which focuses on the hierarchical feature dimension (L), can be expressed as:

$$\pi_L(F) \cdot F = \sigma \left(f \left(\frac{1}{SC} \sum_{S,C} F \right) \right) \cdot F \quad (4)$$

In this setting, $f(\cdot)$ represents a linear function, while $\sigma(\cdot)$ stands for the Hard-Sigmoid activation function. The spatial attention mechanism, which focuses on the spatial dimension (S), can be expressed as:

$$\pi_S(F) \cdot F = \frac{1}{L} \sum_{l=1}^L \sum_{k=1}^K w_{l,k} \cdot F(l; p_k + \Delta p_k; c) \cdot \Delta m_k \quad (5)$$

In this scenario, K denotes the total number of sampling points, $p_k + \Delta p_k$ represents the position after applying the spatial offset, and Δm_k corresponds to a scalar value at position p_k . The task-aware attention mechanism, which concentrates on the channel dimension and dynamically modulates channel activations to satisfy various task demands, can be expressed as follows:

$$\pi_C(F) \cdot F = \max(\alpha^1(F) \cdot F_c + \beta^1(F), \alpha^2(F) \cdot F_c + \beta^2(F)) \quad (6)$$

Here, F_c denotes the feature map of the c th channel, and $[\alpha^1, \beta^1, \alpha^2, \beta^2]^T$ is a parameter vector designed to learn and adjust the activation threshold. The procedure begins with reducing dimensions using global average pooling. Next, the features are processed by two fully connected layers and then passed through a normalization layer. Lastly, the output values are transformed into the $[-1, 1]$ range using a shifted sigmoid function.

IV. EXPERIMENT

A. Experimental Environment and Parameters

The experiments were performed on a Windows 10 system, equipped with an Intel(R) Xeon(R) Silver 4210R CPU and an NVIDIA GTX 3090 GPU featuring 24 GB of video memory. The PyTorch framework was used for implementation, leveraging CUDA 12.0 for GPU acceleration and Python 3.8 as the programming language. Input images were resized to a uniform resolution of 640×640 pixels to ensure dataset consistency. The training setup involved an initial learning rate of 0.01, weight decay set at 0.0005, a batch size of 64, and momentum configured to 0.937. The model was trained over 100 epochs, with all hyperparameters kept constant throughout the experiment to ensure consistency and reproducibility.

B. Experimental Data Set

In our experiments, we employed the RDD2022 dataset, which comprises images collected from six countries. This dataset includes a total of 47,420 images and covers nine categories of pavement defects: D00 (longitudinal cracks), D01 (construction joints), D10 (transverse cracks), D11 (construction joints), D20 (alligator cracks), D40 (potholes), D43 (blurred pedestrian crossings), D44 (blurred white lines), and D50 (manhole covers). Following the removal of unlabeled images, the remaining data was split into training and validation sets in an 8:2 ratio. Specifically, the training set consists of 20,911 images, while the validation set includes 5,227 images. This partitioning ensures a robust evaluation of the model's performance while maintaining a sufficient volume of data for effective training.

C. Evaluation Measures

For a comprehensive assessment of the CCD-YOLO model's performance, we utilized a collection of well-established metrics: mean Average Precision (mAP), Precision, Recall, number of parameters (Params), Gigaflops (GFlops), and Frames Per Second (FPS). Precision measures the proportion of true positive predictions out of all positive predictions, indicating the model's accuracy in detecting positive instances. This is computed as:

$$P = \frac{TP}{FP + TP} \quad (7)$$

$$R = \frac{TP}{TP + FN}$$

Here, TP represents true positives, FP stands for false positives, and FN indicates false negatives. mAP serves as a metric to assess the model's average detection performance across various categories. The formula for calculating mAP is presented below:

$$mAP = \frac{1}{n} \sum_{k=1}^n AP_k = \frac{1}{n} \sum_{k=1}^n \int_0^1 P(R) dR \quad (8)$$

In this set of metrics, n represents the total number of classes, while AP corresponds to the precision for an individual class. Params signifies the overall number of parameters in the model, FLOPs provides insight into the model's computational complexity, and FPS evaluates the speed of model inference.

D. Visualization Analysis

For a thorough assessment of the CCD-YOLO model introduced in this study, Figure 7 illustrates a detailed comparison of the Precision-Recall (P-R) curves between CCD-YOLO and YOLOv8s. These curves offer significant insights into the performance patterns of the models, accurately reflecting their capability to detect positive instances. Additionally, they provide a solid basis for evaluation and optimization processes. Specifically, Figure 7(a) illustrates the P-R curve for YOLOv8s, while Figure 7(b) depicts the P-R curve for CCD-YOLO. In both figures, the thicker blue curve represents the overall mAP@0.5, while the thinner curves in various colors correspond to the mAP@0.5 for each individual category.

Notably, YOLOv8s achieves an mAP@0.5 of 62.1 %, whereas CCD-YOLO attains an mAP@0.5 of 63 %, representing a 0.9 % improvement. This enhancement is reflected in the upward shift of the P-R curve, indicating that, at the same recall rate, CCD-YOLO achieves higher precision. Moreover, CCD-YOLO showcases substantial improvements in computational efficiency, with a 32% reduction in the number of parameters and a decrease of 6.3G in FLOPs when compared to the original YOLOv8s algorithm. These enhancements highlight the ability of CCD-YOLO to achieve a lightweight architecture while preserving or improving detection performance. Overall, the results highlight the model's ability to balance accuracy and efficiency, making it a promising solution for practical applications in road defect detection.

For a more intuitive illustration of CCD-YOLO's detection performance, Figure 8 offers a comparative analysis of the

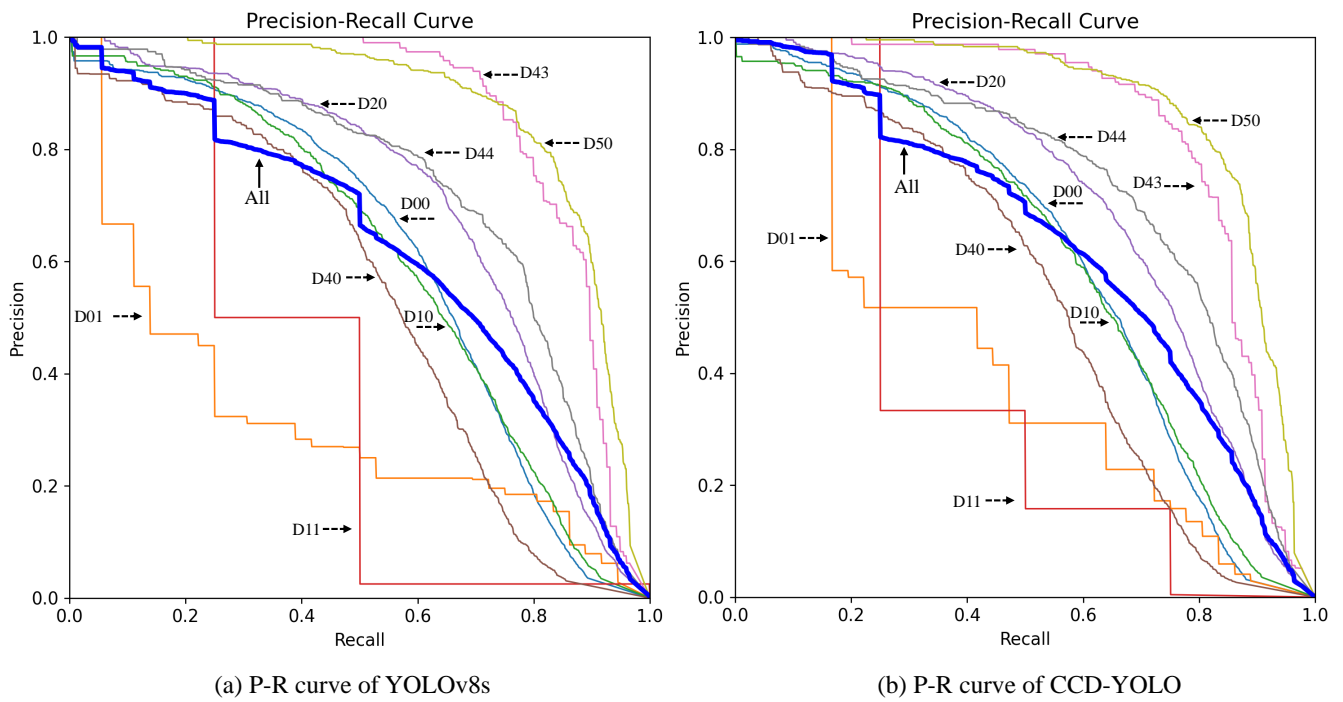


Fig. 7: The P-R curves of YOLOv8s and CCD-YOLO

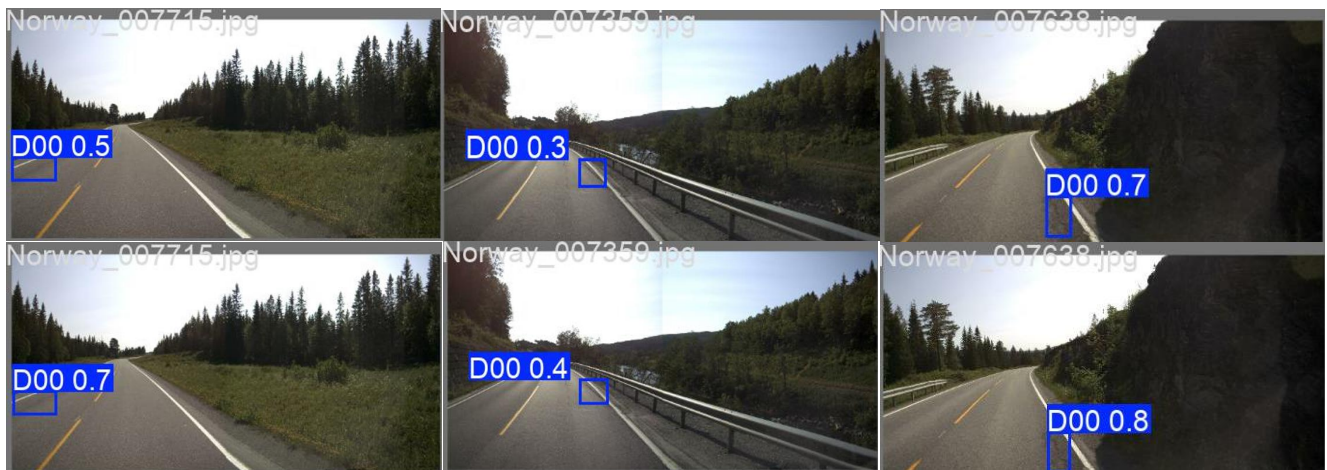


Fig. 8: A Comparison of the Detection Results between YOLOv8s and CCD-YOLO

detection results from YOLOv8s and CCD-YOLO. In this figure, the top portion illustrates the detection outputs of the YOLOv8s algorithm, whereas the bottom portion highlights those of the CCD-YOLO algorithm. A thorough examination of these images reveals that the optimized lightweight model not only preserves the original detection accuracy but also attains significant enhancements in performance.

Specifically, CCD-YOLO demonstrates enhanced detection precision compared to YOLOv8s, particularly in identifying small and irregularly shaped defects. Additionally, CCD-YOLO achieves these improvements while significantly reducing the number of parameters and computational requirements, underscoring its efficiency and practicality. These results validate the effectiveness of the proposed optimizations and highlight the model’s potential for deployment in resource-constrained environments.

E. Ablation Study

In order to evaluate the effectiveness of the three optimization techniques in the CCD-YOLO model, this study performed an ablation analysis to systematically examine how each technique influences detection performance. A comprehensive overview of the ablation experiments is presented in Table 1, which outlines the following configurations:

1. Substitution of the conventional feature pyramid in the original YOLOv8s model with the CCFN module.
2. Substitution of the C2f module in the neck of the original architecture with the CSPPC module.
3. Replacement of the standard detection module in the baseline model with the Dydetect module.
4. Concurrent application of both the CCFN and CSPPC optimizations.

TABLE I: Ablation experiment

YOLOv8s	CCFN	CSPPC	Dydetect	P(%)	R(%)	mAP@0.5(%)	Params(M)	FLOPs(G)	FPS
✓				67.2	56.5	62.1	11.1	28.7	107
✓	✓			61.9	56.4	60.9	7.2	23.1	118
✓		✓		59.7	63.7	62.1	9.4	24.9	112
✓			✓	62.3	55.6	61.4	10.8	28.1	107
✓	✓	✓		59.8	62.8	62.7	6.9	21.6	121
✓	✓	✓	✓	61.3	62.1	63	7.5	22.4	116

TABLE II: Performance comparison of mainstream algorithms

Model Name	P(%)	R(%)	mAP@0.5(%)	Params(M)	FLOPs(G)	FPS
Faster R-CNN [11]	69.1	65.2	66.8	38.2	47.3	53
YOLO-LRDD[21]	61	57.8	59.5	19.8	17.4	87
YOLOv5s	58.4	55.6	57.2	9.1	23.8	95
YOLOv6s [29]	50.1	56	56.4	16	44	72
YOLOv7-tiny [30]	64.2	57.6	58.7	6.3	13.6	127
YOLOv8n	57.2	57.8	56.7	3	8.1	135
YOLOv8s	67.2	56.5	62.1	11.1	28.7	107
YOLOv9-S [31]	69.5	53	62.7	7.1	26.2	113
ours	61.3	62.1	63	7.5	22.4	116

5. Simultaneously implement the three aforementioned optimizations.

The results presented in Table 1 reveal the following key findings: Introducing the CCFN module led to a reduction of 3.9 million parameters, a decrease of 5.6 billion FLOPs, and a significant increase in FPS. Nevertheless, this optimization led to a minor decrease in accuracy. Integrating the CSPPC module kept mAP@0.5 at the same level while decreasing the number of parameters by 1.7 million and FLOPs by 3.8 billion, along with a 5-frame boost in FPS. This enhancement maintained the model’s accuracy while significantly reducing computational demands. The Dydetect module cut the parameter count by 0.3 million and FLOPs by 0.6 billion, further enhancing efficiency. When both the CCFN and CSPPC modules were combined, there was an increase in mAP@0.5 as well as notable reductions in parameters and computational requirements. Implementing all three optimizations together led to a further improvement in detection accuracy by integrating the Dydetect module. In comparison to the original model, CCD-YOLO reduced the number of parameters by 32 % while enhancing detection accuracy. The experimental results indicate that the lightweighting strategies employed in this study are highly efficient, leading to substantial improvements in the model’s performance across various aspects, such as accuracy, computational efficiency, and inference speed. The CCD-YOLO model thus represents a balanced and optimized solution for real-world applications such as road defect detection.

F. Comparison with Other State-of-the-Art Object Detection Algorithms

In order to demonstrate the superiority of the CCD-YOLO algorithm compared to other state-of-the-art algorithms, this study performed a comparative analysis using the RDD2022 dataset. Table 2 presents a detailed comparison with existing models, such as Faster R-CNN [11], YOLO-LRDD [21], YOLOv5s, YOLOv6s [29], YOLOv7-tiny [30], YOLOv8n, YOLOv8s, and YOLOv9s [31]. The performance of each model was evaluated using key metrics such as mAP@0.5, parameter count (Params),

and computational cost measured by floating-point operations (FLOPs).

Notably, the CCD-YOLO model demonstrated superior performance in terms of mAP@0.5, achieving the highest accuracy among the compared models. Although it might be slightly less competitive in some aspects, like parameter count or FLOPs, CCD-YOLO successfully attains an ideal balance between accuracy and real-time efficiency. This balance is particularly critical for practical applications, where both detection precision and computational efficiency are paramount. The findings confirm the efficacy of CCD-YOLO, emphasizing its capacity to achieve high accuracy while preserving computational efficiency. This makes it a strong and reliable choice for practical applications, including road defect detection.

V. CONCLUSION

In response to the challenges of balancing accuracy and real-time performance in road defect detection, as well as the deployment limitations caused by excessive parameters on edge devices with constrained computational resources, this paper introduces a lightweight road defect detection model called CCD-YOLO, which is based on YOLOv8s. The primary improvements are as follows: CCFN efficiently fuses detailed features and contextual information across multiple scales, substantially decreasing the parameter count and computational complexity while ensuring strong feature representation capabilities. Additionally, the CSPPC Module is incorporated into the model. This module minimizes redundancy among feature maps across different channels, thereby lowering computational costs and memory access requirements. By leveraging partial convolution technology, it achieves efficient feature extraction with reduced resource consumption. Introduction of Dydetect. By integrating multiple attention mechanisms, Dydetect strengthens the representational capabilities of the detection head, thereby enhancing the model’s capacity to identify objects with diverse sizes and shapes. This leads to improved overall detection performance. Extensive experiments and analyses were carried out on the RDD2022 dataset, where CCD-YOLO was evaluated against other leading algorithms. The results demonstrate that

CCD-YOLO not only achieves an optimal balance between accuracy and real-time performance in road defect detection tasks but also can be efficiently deployed on edge devices with limited computing resources. The results thoroughly validate the effectiveness and advantages of CCD-YOLO, establishing it as a viable and efficient choice for practical applications. These advancements position CCD-YOLO as both an academically innovative and practically viable solution for large-scale road maintenance systems, with potential applications in smart cities and autonomous infrastructure inspection. Future work will explore further optimization for low-power embedded systems.

REFERENCES

- [1] M. A. Benallal and M. S. Tayeb, "An image-based convolutional neural network system for road defects detection," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 2, p. 577, 2023.
- [2] A. Hosseini, A. Faheem, H. Titi, and S. Schwandt, "Evaluation of the long-term performance of flexible pavements with respect to production and construction quality control indicators," *Construction and Building Materials*, vol. 230, p. 116998, 2020.
- [3] P. Kumar, A. Sharma, and S. R. Kota, "Automatic multiclass instance segmentation of concrete damage using deep learning model," *IEEE Access*, vol. 9, pp. 90330–90345, 2021.
- [4] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016.
- [5] Q. Chen, Y. Huang, H. Sun, and W. Huang, "Pavement crack detection using hessian structure propagation," *Advanced Engineering Informatics*, vol. 49, p. 101303, 2021.
- [6] Z. Sun, L. Zhu, S. Qin, Y. Yu, R. Ju, and Q. Li, "Road surface defect detection algorithm based on yolov8," *Electronics*, vol. 13, no. 12, p. 2413, 2024.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [8] 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.
- [9] 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.
- [10] R. Girshick, "Fast r-cnn," *ArXiv Preprint ArXiv:1504.08083*, 2015.
- [11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.
- [12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.
- [13] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [14] 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.
- [15] J. Redmon, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016.
- [16] 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.
- [17] P. Huang, S. Wang, J. Chen, W. Li, and X. Peng, "Lightweight model for pavement defect detection based on improved yolov7," *Sensors*, vol. 23, no. 16, p. 7112, 2023.
- [18] F.-J. Du and S.-J. Jiao, "Improvement of lightweight convolutional neural network model based on yolo algorithm and its research in pavement defect detection," *Sensors*, vol. 22, no. 9, p. 3537, 2022.
- [19] H. Luo, C. Li, M. Wu, and L. Cai, "An enhanced lightweight network for road damage detection based on deep learning," *Electronics*, vol. 12, no. 12, p. 2583, 2023.
- [20] C. Zhang, G. Li, Z. Zhang, R. Shao, M. Li, D. Han, and M. Zhou, "Aal-net: A lightweight detection method for road surface defects based on attention and data augmentation," *Applied Sciences*, vol. 13, no. 3, p. 1435, 2023.
- [21] 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.
- [22] H. Li and J. Wu, "Lsod-yolov8s: A lightweight small object detection model based on yolov8 for uav aerial images," *Engineering Letters*, vol. 32, no. 11, pp.2073-2082, 2024.
- [23] S. Guo, N. Zhao, X. Ouyang, and Y. Ouyang, "Rbl-yolov8: A lightweight multi-scale detection and recognition method for traffic signs," *Engineering Letters*, vol. 32, no. 11, pp.2180-2190, 2024.
- [24] 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.
- [25] 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.
- [26] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16965–16974.
- [27] 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. 12021–12031.
- [28] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7373–7382.
- [29] 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.
- [30] 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.
- [31] 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.