SDC-YOLO: A Method for Road Defect Detection

Man Wu, Ye Tao*

***Abstract*—To tackle the challenges associated with complex backgrounds, low-resolution images, and similar crack features that lead to missed detections, false positives, and low accuracy in pavement defect detection, we propose an enhanced pavement defect detection model named SDC-YOLO, which builds upon YOLOv8. Specifically, in the backbone and neck components, we replaced conventional convolution layers with SPDconv, an innovative approach that combines spatial-to-depth layers with non-strided convolution layers, thereby enhancing the model's capability to extract small-scale features. Additionally, in the neck component, we incorporated the lightweight dynamic upsampling module (Dysample) as a replacement for conventional upsampling techniques, enriching feature details while reducing computational overhead. Furthermore, we developed the efficient CSC2f module to minimize redundant features, enhance multi-scale feature extraction and fusion capabilities, and reduce the parameter count and computational load. On the RDD2022 dataset, compared to the baseline model, the mAP@0.5 of SDC-YOLO has increased by 2.3%, while the number of parameters and FLOPs has seen a slight rise. The model consistently outperforms other algorithms, thereby providing robust validation of its effectiveness and superiority.**

***Index Terms*—road defect detection, YOLOv8, upsampling, multi-scale features**

I. Introduction

ROADS , as a critical component of modern infrastructure, play an indispensable role in supporting national economic and social development. They are also essential for enhancing the quality of life and promoting balanced regional growth [1]. The causes of road defects are multifaceted: natural factors such as temperature variations, freeze-thaw cycles, precipitation-induced erosion, and geological movements can all contribute to road degradation; human factors include overloaded vehicles, poor driving practices, and inadequate maintenance; additionally, substandard materials or flawed design and construction processes are significant contributors to road damage [2]. Road deterioration can have substantial negative impacts on traffic efficiency, vehicle performance, socio-economic development, and public safety. Consequently, timely and effective road maintenance and repair are essential for maintaining uninterrupted traffic flow, reducing operational costs, and safeguarding public safety. With the ongoing progress in technology, road defect detection techniques have become increasingly important in the maintenance and management of modern transportation infrastructure [3].

Manuscript received December 27, 2024; revised March 16, 2025. This work was supported by National Natural Science Foundation of China (62272093), the Economic and Social Development Research Topics of Liaoning Province (2025-10146-244),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).

The essential role of pavement defect detection technology is to detect and assess different kinds of pavement issues, including cracks, potholes, and rutting, which in turn facilitates prompt maintenance actions. Advanced detection methods facilitate early-stage identification of issues, thus preventing road conditions from deteriorating to the point where extensive renovations are required. As a result, this method not only greatly enhances the lifespan of roads but also considerably cuts down on maintenance expenses in the long term. In contrast, traditional pavement defect detection methods primarily rely on manual visual inspections and simple measurement tools, which are labor-intensive, time-consuming, and inefficient. Moreover, the accuracy of detection results can be significantly influenced by the experience and subjective judgment of the operators, potentially leading to oversight or misassessment of defects [4]. Automated pavement defect detection technology leverages advanced sensors and image processing software [5], achieving substantial improvements in detection efficiency and accuracy. However, it also faces several limitations, including high equipment costs, suboptimal performance in complex environments, and stringent requirements for technical expertise. These factors have constrained its widespread adoption in practical applications. As a result, algorithms for detecting pavement defects based on deep learning have become a potential solution to tackle these challenges.

From the early 21st century onward, object detection algorithms utilizing deep learning have made remarkable progress and have been widely applied in numerous fields [6]. As deep learning technology continues to advance, object detection algorithms have been divided into two main categories: one-stage and two-stage approaches. These two categories differ in their detection modes, processing speeds, and accuracy levels. Representative two-stage algorithms include Faster R-CNN [7], Mask R-CNN [8], and R-FCN [9]. Liang et al. [10] developed a Faster R-CNN-based method capable of automatically identifying and precisely locating defects such as cracks, potholes, oil stains, and spots on road surfaces. Through comprehensive analysis and rigorous training, they refined an optimized Faster R-CNN model, which was validated for its accuracy and effectiveness through data comparison and experimental evaluations.

While two-stage algorithms provide greater accuracy, they compromise on speed and simplicity, which makes them less ideal for real-time applications and scenarios with limited resources. On the other hand, one-stage algorithms, such as SSD [11], RetinaNet [12], and YOLO [13], are widely adopted in high-real-time and resource-limited environments due to their efficiency and practicality. Among these, the YOLO series algorithms have demonstrated significant application potential across various fields, attributed to their rapid detection capabilities and excellent accuracy. Du et al. [14] proposed a pavement defect detection and classification strategy based on the YOLO network, addressing challenges

in pavement defect detection. This approach employs the YOLO network's object detection framework to forecast the positions and types of possible defects. Jiang et al. [15] designed a new road crack detection algorithm by combining the Transformer architecture with an explicit visual center, enabling it to capture long-range dependencies and integrate essential features. Despite improvements in detection performance, issues such as missed detections, false positives, and suboptimal multi-scale feature fusion, particularly in recognizing small cracks, still persist. Han et al. [16] presented an enhanced pavement defect recognition algorithm called MS-YOLOv8. This algorithm refines the YOLOv8 model through the integration of three innovative mechanisms, which improve detection precision and suitability for various pavement conditions. Sun et al. [17] introduced an enhanced YOLOv8 algorithm, where the conventional convolutions in the network backbone are substituted with a module consisting of spatial-to-depth layers and non-strided convolution layers. This modification enhances the detection of small road defects. Additionally, this algorithm enhances multi-scale feature extraction by fully fusing spatial and scale features using the ASF-YOLO neck. Wang et al. [18] designed an improved road defect detection algorithm named BL-YOLOv8. This algorithm incorporates the BiFPN concept to restructure the neck and adopts the SimSPPF module to lower the number of parameters and computational requirements, thereby enhancing processing efficiency. Moreover, the incorporation of a dynamic large convolutional kernel attention mechanism enlarges the model's receptive field, which improves the precision of target detection. These enhancements are designed to boost the accuracy of detecting pavement defects while maintaining real-time performance. Nevertheless, issues like missed detections, false positives, and the requirement for further enhancement of multi-scale feature fusion, particularly for small cracks, continue to remain.

To address the challenges in pavement defect detection, this paper selects YOLOv8 from the YOLO series as the foundational algorithm for research. Among the five variants, YOLOv8s offers a relatively balanced compromise between speed and accuracy. Nonetheless, there remains potential for enhancing its detection precision [19]. Thus, this paper concentrates on refining the YOLOv8s variant to enhance its detection accuracy. As a result, we introduce an enhanced pavement defect detection algorithm based on YOLOv8s, referred to as SDC-YOLO. The primary contributions of this paper are outlined as follows:

1. Introduction of SPDconv: We replace conventional convolution layers with SPDconv, which integrates spatial-to-depth layers and non-strided convolution layers, thereby minimizing the loss of fine-grained information and enhancing the extraction of small-scale features.

2. Application of a Lightweight Dynamic Upsampling Operator: This operator optimizes the upsampling process in the Neck component, thereby reducing the loss of fine-grained details in low-resolution images, enriching feature details, effectively capturing context information, and decreasing the number of parameters and computational burden.

3. Proposal of an Efficient CSC2f Module: This module minimizes redundant features, enhances multi-scale feature fusion capabilities, and significantly reduces computational overhead. By optimizing these aspects, the CSC2f module improves overall detection performance while maintaining efficiency.

The rest of this paper is structured as follows. Section 2 provides a review of the related works. Section 3 presents a detailed description of the proposed SDC-YOLO algorithm. Section 4 outlines the experimental setup and methodology. Finally, Section 5 presents the findings and conclusions.

II. Related Work

YOLOv8 carries forward the fundamental concept from the YOLOv5 series, treating object detection as a regression problem. This approach allows for efficient end-to-end detection. The network architecture mainly consists of three key components: Backbone, Neck, and Head. In the Backbone and Neck sections, YOLOv8 incorporates the ELAN design concept introduced in YOLOv7. It replaces the C3 structure of YOLOv5 with the C2f structure to promote a richer gradient flow. Moreover, the number of channels is dynamically adjusted based on different model scales [20].

The backbone network plays a role in extracting features from the input image and acts as the core foundation of the entire architecture. In YOLOv8, the backbone adopts a deeper and wider convolutional layer design to enhance feature representation capabilities. Positioned between the backbone and the detection head, the neck network utilizes the PAN-FPN (Path Aggregation Network - Feature Pyramid Network) structure to enable information exchange among multi-scale feature maps. This improves the integration of contextual information across different scales. The enriched information flow allows the model to capture target features more efficiently, thereby enhancing overall detection performance.

In contrast to YOLOv5, YOLOv8 incorporates two major enhancements in the head structure. First, it employs a decoupled head architecture that divides the classification and regression tasks while eliminating the objectness branch. Second, it shifts from a method that relies on anchors to one that is anchor-free. These modifications aim to simplify the model architecture, improve localization accuracy, enhance generalization capability, accelerate the training process, and boost deployment efficiency. As a result, these enhancements enable YOLOv8 to achieve superior performance in road defect detection tasks, particularly in scenarios requiring rapid and precise identification.

YOLOv8 employs the Task-Aligned Assigner based on the Best Online Sample (TOOD) for positive and negative sample matching. The Task-Aligned method implements a sophisticated fusion of classification scores and IoU, evaluating the consistency of tasks and precisely measuring the alignment extent at the anchor level for every individual instance. The loss computation consists of two key elements: classification and regression. For classification, Binary Cross-Entropy (BCE) loss is used to discriminate between categories and output confidence scores. To accommodate the anchor-free mode and enhance model generalization, the regression task adopts Dual-Focal Loss (DFL). DFL utilizes cross-entropy to optimize the probabilities of the two positions nearest to the label, allowing the network to concentrate more efficiently on the distribution of the target location and its neighboring regions. Additionally, Complete Intersection

over Union (CIoU) loss is utilized. CIoU, an advanced loss function in object detection, evaluates bounding box similarity by incorporating center point distance and aspect ratio differences based on IoU. By considering the location, scale, and geometry of bounding boxes, CIoU improves the precision of the model's regression performance.

III. Improved Model

A. SDC-YOLO

In order to tackle the issues of false alarms, missed detections, and low accuracy in road defect detection tasks while meeting real-time requirements, YOLOv8s was selected as the base model. Improvements were implemented in both the backbone and neck sections to enhance overall performance. The improved SDC-YOLO structure is illustrated in Figure 1. Firstly, conventional convolution layers were replaced with SPDconv, which integrates spatial-to-depth layers and non-strided convolution layers. This modification effectively captures fine-grained features, minimizes information loss, and enhances the detection capability for small-sized cracks. Secondly, an enhanced upsampling operator was introduced to minimize the loss of low-resolution image information, enrich feature details, and effectively capture contextual information. This operator also reduces the number of parameters and alleviates computational burden.Finally, an efficient CSC2f module was proposed to eliminate redundant features, enhance feature representation capabilities, and simultaneously reduce computational costs. These enhancements collectively result in the outstanding performance of SDC-YOLO in detecting road defects.

B. SPDconv moudle

Convolution is a fundamental building block in constructing YOLOv8, serving key functions such as feature extraction, feature fusion, and dimensionality adjustment of input features. In object detection tasks, traditional convolutions generally perform well for high-resolution images and large target objects. However, their effectiveness diminishes when applied to small-sized targets and low-resolution images. In the context of road defect detection, image resolution is often suboptimal, and the road background is complex, which can interfere with crack detection. Cracks exhibit significant irregularity in size and shape, with some being extremely fine. During the learning process, small cracks may be overshadowed by larger cracks, leading to missed detections. In these particular circumstances, conventional convolutions are prone to losing detailed information, which impedes the model's capacity to fully extract and learn the pertinent features.

In order to tackle the limitations of conventional convolution in road defect detection tasks, we introduce SPDconv [21] to replace certain traditional convolution layers. This replacement is intended to avoid information loss and strengthen the network's capability to extract small-scale features. SPDconv is an innovative spatial encoding method that improves the performance of models by efficiently processing image data. It comprises two key components: a spatial-to-depth layer and a non-strided convolutional layer.

The spatial-to-depth layer transforms the spatial dimensions of the feature map into the depth dimension, thereby increasing the number of channels to preserve more detailed information. The subsequent non-strided convolutional layer maintains the spatial resolution while reducing the number of channels, ensuring that fine-grained features are retained. By integrating these components, SPDconv effectively mitigates information loss and enables the network to capture more refined features, thereby enhancing performance in road defect detection tasks. We will provide a detailed illustration of this mechanism through specific examples below.

Assume that there exists a feature map X with dimensions $L \times L \times C_1$. Sub-feature maps are extracted from X using a stride of S, where S denotes the step size for segmentation:

$$\begin{aligned} f_{0,0}{=}X[0:L{:}s,0:L:s], f_{1,0}{=}X[1:L{:}s,0:L{:}s],\ldots, \\ f_{s-1,0} = X[s-1:L{:}s,0:L{:}s]; \\ f_{0,1}{=}X[0:L{:}s,1:L:s], f_{1,1}{=}X[1:L{:}s,1:L:s],\ldots, \\ f_{s-1,1} = X[s-1:L{:}s,1:L{:}s]; \\ \ldots \\ f_{0,s-1}{=}X[0:L{:}s,s-1:L:s], f_{1,s-1},\ldots, \\ f_{s-1,s-1} = X[s-1:L{:}s,s-1:L:s] \end{aligned} \tag{1}$$

Where $f_{x,y}$ consists of all elements $X_{(i,j)}$ that satisfy the condition "both $i+x$ and $j+y$ are divisible by L". For each subgraph, the feature map X is downsampled by a factor of s. Specifically, when $s = 2$, downsampling X results in four subgraphs: $f_{0,0}$, $f_{1,0}$, $f_{0,1}$, and $f_{1,1}$, each with dimensions $(L/2, L/2, C_1)$. These four subgraphs are then combined along the channel dimension and subjected to an element-wise addition operation, yielding a new feature map with $4C_1$ channels and spatial dimensions reduced by half. Following the spatial-to-depth transformation, a convolutional layer with no stride conducts a convolution operation using a stride of 1. This process reduces the number of channels to C_2 while preserving the spatial resolution at half the original size. The primary rationale for selecting non-strided convolution is to retain the maximum amount of distinctive feature information. Conversely, using a 3×3 filter with a stride set to 2 or 3 would result in significant loss of critical feature details during the downsampling procedure.

C. Dysample module

The upsampling operator refers to a set of operations aimed at recovering the spatial resolution of feature maps. Its main objective is to transform low-resolution feature maps into high-resolution ones, enabling more effective capture and representation of detailed information. In the YOLOv8 model, the upsampling process employs the nearest neighbor interpolation (Nearest Neighbor Upsampling) technique. This method increases the size of the feature map by selecting the pixel value closest to the target pixel, thereby improving its spatial resolution.

In the context of road surface defect detection, cracks exhibit irregular sizes and shapes; some may be extremely small or have complex geometries, making accurate detection challenging for the model. Traditional upsampling methods often result in images that have irregular edges or appear blurry, which can lead to the loss of critical detail information and weaken the model's ability to detect features effectively.

To address these limitations, we introduce the Dysample operator, which enhances the upsampling process while

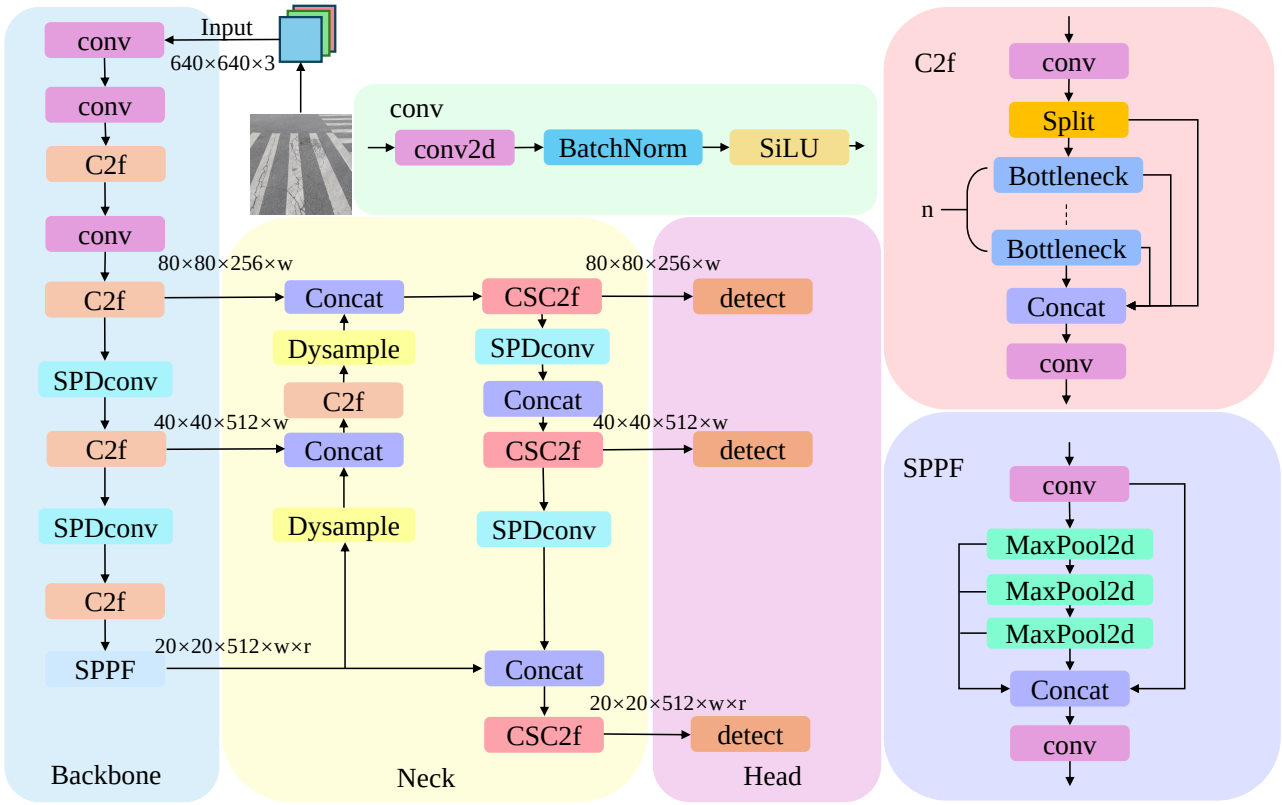


Fig. 1: The architecture of the SDC-YOLO network

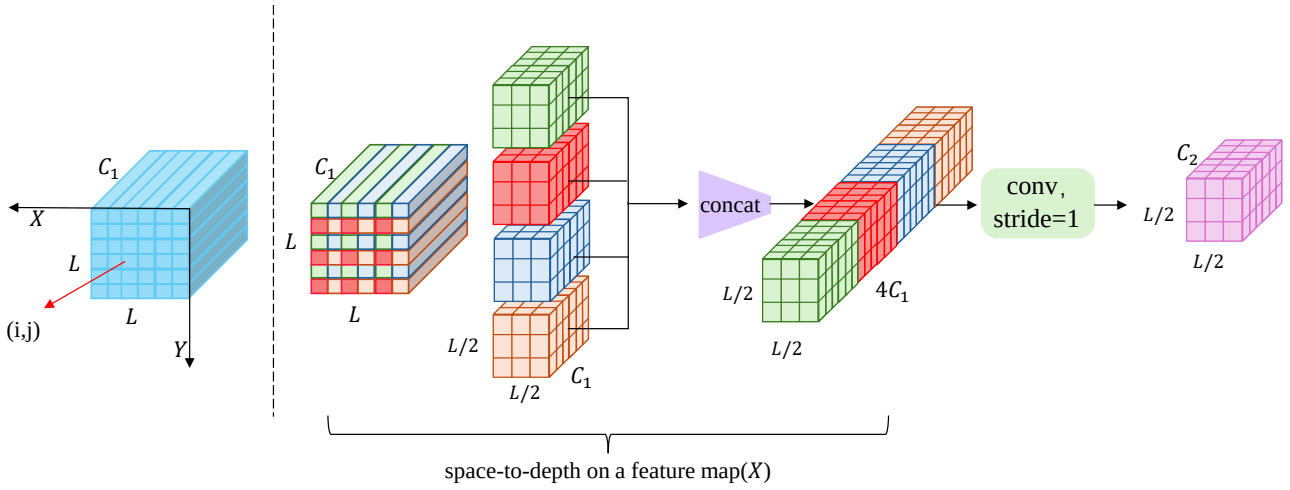


Fig. 2: The structure of the SPDconv module

enriching feature representation. Dysample adaptively selects the optimal upsampling strategy, effectively preserving key details. This approach not only reduces computational resource consumption but also improves the model's accuracy in detecting small-sized targets.

Dysample is a lightweight dynamic upsampling operator that can not only efficiently capture fine-grained details but also produce more precise high-resolution feature maps, thereby significantly improving the overall performance of the model [22]. Compared to other upsampling methods, Dysample achieves higher accuracy while significantly reducing computational costs.

The detailed procedure for Dysample's upsampling operation is as follows: Initially, the input feature map is processed by a sampling point generator to produce a set of sampling points. Subsequently, grid sampling is applied to re-sample the feature map based on the positions specified in the sampling set, resulting in a high-resolution feature map. This process ensures that critical details are preserved during upsampling, leading to improved detection accuracy. The sampling method employed by Dysample is illustrated in Figure 3.

For a feature map M with dimensions $C \times H \times W$ and an upsampling scale factor α, a linear transformation is utilized,

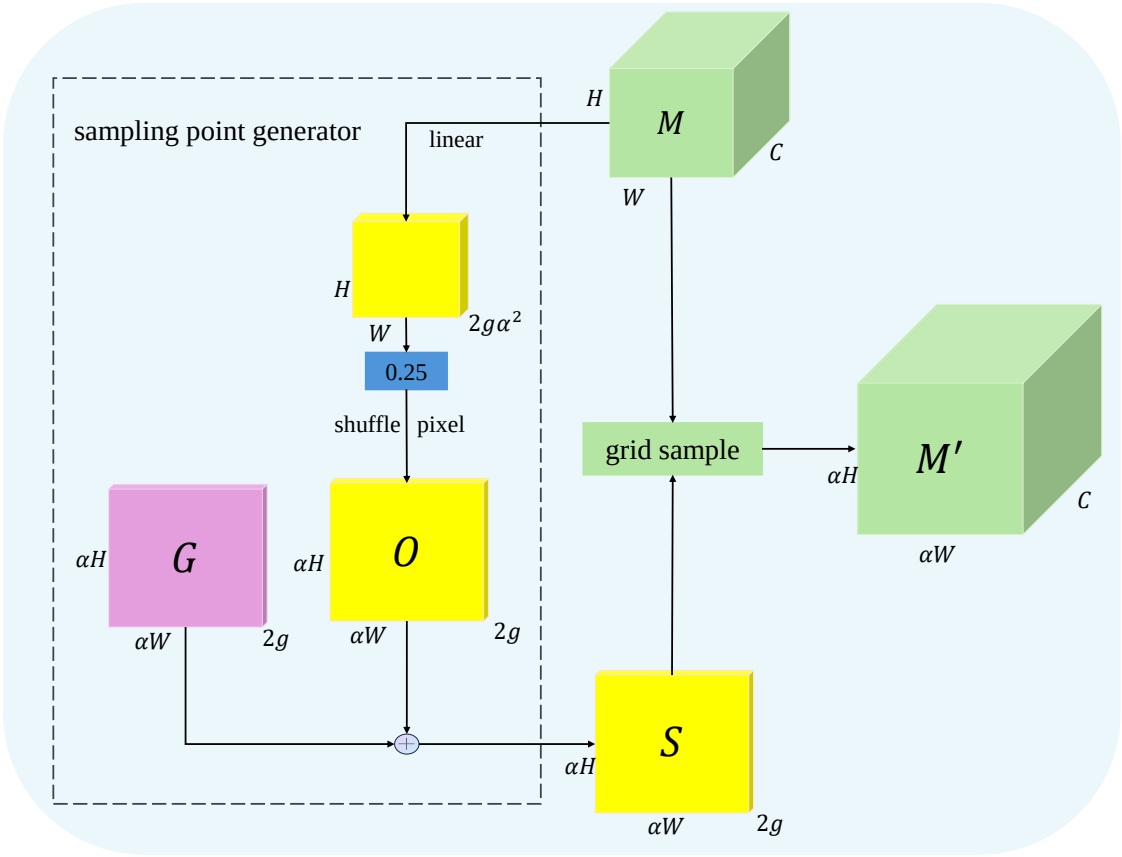


Fig. 3: The structure of the Dysmple module

with the input channel size being C and the output channel size configured as $2\alpha^2$. This is followed by pixel shuffling to generate an offset O of size $2 \times \alpha H \times \alpha W$. The original grid G is then added to the offset O to obtain the final sampling set S, as described by the following formula:

$$\begin{aligned} O &= 0.25 \text{ linear } (X), \\ S &= G + O \end{aligned} \tag{2}$$

Herein, 0.25 is the range factor, which restricts the offset range of the sampling positions to alleviate their overlap. Finally, the original feature M and the sampling set S generate a new feature M' through grid sampling, and the formulation is as follows:

D. CSC2f module

The C2f module plays a key role in the YOLOv8 network model, functioning as a convolutional neural network (CNN) module with residual connections. Its primary role is to perform feature extraction and fusion. The efficiency and effectiveness of the C2f module play a significant role in shaping the overall performance of the YOLOv8 model. Road surface cracks are often small, irregularly shaped, and embedded in complex backgrounds, further complicated by external factors such as weather and lighting conditions. These characteristics pose significant challenges for the C2f module, potentially leading to the capture of excessive irrelevant information, resulting in feature redundancy and degraded model performance.

In order to tackle these limitations, we introduce the CSC2f module, illustrated in Figure 4. This module optimizes the C2f module by incorporating the Spatial and Channel Reconfigurable Convolution Block (SCCB), which replaces the original Bottleneck structure. The core innovation lies in the introduction of the spatial and channel reconfigurable convolution (SCConv) [23]. The distinctive architecture and algorithm of SCConv improve the model's capacity to extract and process complex feature information with greater efficiency.

Additionally, SCConv optimizes the utilization of computational resources, thereby improving both the model's performance and operational efficiency. This improvement offers substantial assistance for the model's successful operation in identifying road defects, ensuring better accuracy and reliability in challenging environments.

SCConv is an efficient convolutional module designed to minimize spatial and channel redundancies within convolutional neural networks (CNNs). Its primary objective is to minimize computational resources and enhance performance by optimizing the feature extraction process.

The SCConv module consists of a Spatial Reconstruction Unit (SRU) and a Channel Reconstruction Unit (CRU). The SRU mitigates spatial redundancy through a separation and reconstruction approach, effectively preserving critical spatial information. Meanwhile, the CRU employs a segmentation-transformation-fusion strategy to alleviate channel redundancy, ensuring that only the most relevant features are retained. The specific structure of these units

is depicted in Figure 5. By integrating these innovative components, SCConv not only reduces computational overhead but also improves the model's ability to capture and process complex feature information accurately.

Given an input feature $X \in R^{C\times H\times W}$, where C denotes the number of channels, H represents the height, and W indicates the width, the input first passes through the SRU. In the SRU, X undergoes group normalization to reduce scale differences among different feature maps. The formula for calculating the normalization-related weights is provided in Figure 5, where γ is a trainable parameter that reflects variations in spatial information. By applying a series of weights to the features, mapping them through a Sigmoid activation function, and setting a threshold, the method distinguishes between information weights W_1 and non-information weights W_2. The specific formula is as follows:

$$W = \text{ Threshold (Sigmoid } (W_\gamma(GN(X))) \text{)} \tag{3}$$

Subsequently, the initial feature X is multiplied element-wise by W_1 and W_2 to obtain X_1^w, which contains a higher information content, and X_2^w, which contains a lower information content. X_2^w is typically considered redundant information. To decrease spatial redundancy, a reconstruction process is performed by adding the information-rich feature X_1^w and the information-dense feature X_2^w in a cross-reconstruction manner. The reconstructed features X^{w1} and X^{w2} are then concatenated to form the final feature X, which has had spatial redundancy removed. The specific process is detailed as follows:

$$\begin{aligned} X_1^w &= W_1 \times X, \\ X_2^w &= W_2 \times X, \\ X_{11}^w + X_{22}^w &= X^{w1}, \\ X_{21}^w + X_{12}^w &= X^{w2}, \\ X^{w1} \cup X^{w1} &= X^w \end{aligned} \tag{4}$$

Where $\cup$ denotes concatenation. Although feature X^w has reduced spatial redundancy, it still contains redundant features in the channel dimension. Subsequently, the CRU performs further operations on the refined feature X^w that has been processed by the SRU. First, the channels of X^w are split into αC channels and $(1 - \alpha)C$ channels, where α represents the partition ratio and $0 \leq \alpha \leq 1$. In order to improve computational efficiency, 1×1 convolution is utilized to reduce the number of channels, while a compression ratio is defined to regulate the channel count. After completing the division and compression operations, feature X^w is split into the left part X_l and the right part X_r. High-level information is obtained from the feature-rich X_l through GWC and PWC, which helps to decrease computational costs. The results of these operations are then aggregated to produce the refined feature Y_1. The equation for the left transformation stage is presented as follows:

$$Y_1 = M^G X_l + M^{P_1} X_l \tag{5}$$

In this context, M^G and M^{P_1} represent the trainable weight matrices corresponding to GWC and PWC. During the right-stage processing, a 1×1 PWC is utilized on the feature X_r, and the result is then concatenated with the original X_r, without introducing extra computational expenses. This operation yields Y_2, capturing shallow hidden detail features. The equation for this procedure is presented below:

$$Y_2 = X_r \cup M^{P_2} X_r \tag{6}$$

In this context, M^{P_2} represents the trainable weight matrix for PWC, while $\cup$ signifies the concatenation operation. After the exchange stage, $Y_1^{'}$ and $Y_2^{'}$ are merged through a weighted summation by utilizing global average pooling and SoftMax activation along the channel dimension, resulting in the refined feature Y. Upon going through the CRU, the refined feature Y further diminishes redundancy in the channel dimension. The CRU effectively extracts informative and distinctive features while keeping the computational cost minimal.

IV. Experiment

A. Experimental Environment and Parameters

The experiment was carried out on a Windows 10 platform, utilizing an Intel(R) Xeon(R) Silver 4210R CPU and an NVIDIA GTX 3090 GPU equipped with 24GB of video memory. The PyTorch framework was employed, using CUDA 12.0 as the GPU accelerator and Python 3.8 for programming. The input images were adjusted to a size of 640×640 pixels.

The training configuration included an initial learning rate of 0.01, weight decay of 0.0005, a batch size of 64, and momentum of 0.937. Training proceeded for 100 epochs, during which all parameters remained constant. This setup ensured consistent and reproducible experimental conditions.

B. Experimental Data Set

We utilized the RDD2022 dataset for our experiments, which includes images from six distinct countries. The dataset comprises a total of 47,420 images and covers nine types of pavement defects: D00 (longitudinal cracks), D01 (construction joint areas), D10 (transverse cracks), D11 (construction joint areas), 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 images were split into training and validation sets in a ratio of 8:2. In detail, the training dataset includes 20,911 images, while the validation dataset contains 5,227 images. This partitioning guarantees a well-balanced allocation of data for both robust model training and effective evaluation.

C. Evaluation Measures

For assessing the overall performance of the SDC-YOLO model, we adopted a wide range of metrics, including mean Average Precision (mAP), Precision, Recall, number of parameters (Params), Gigaflops (GFlops), and Frames Per Second (FPS).

Precision indicates the ratio of true positive predictions to all positive predictions, showcasing the accuracy of the model's positive classifications. Recall evaluates the model's capability to accurately identify all genuine positive instances, reflecting its effectiveness in detecting true positives. These two metrics provide insights into the model's detection

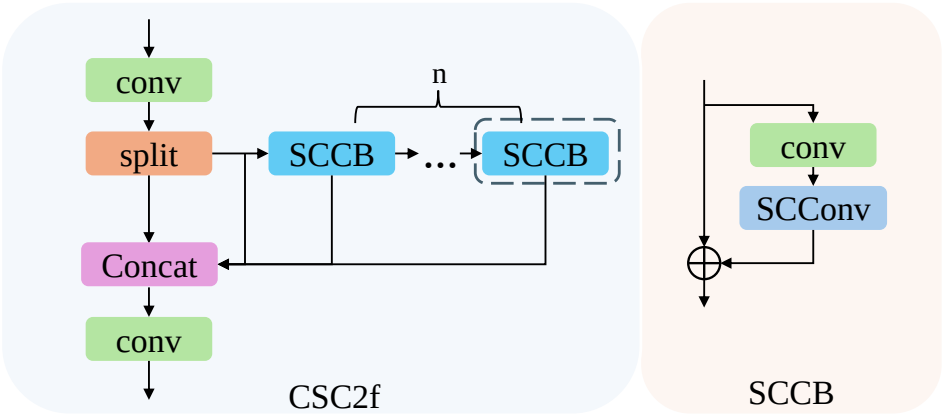


Fig. 4: The structure of the CSC2f module

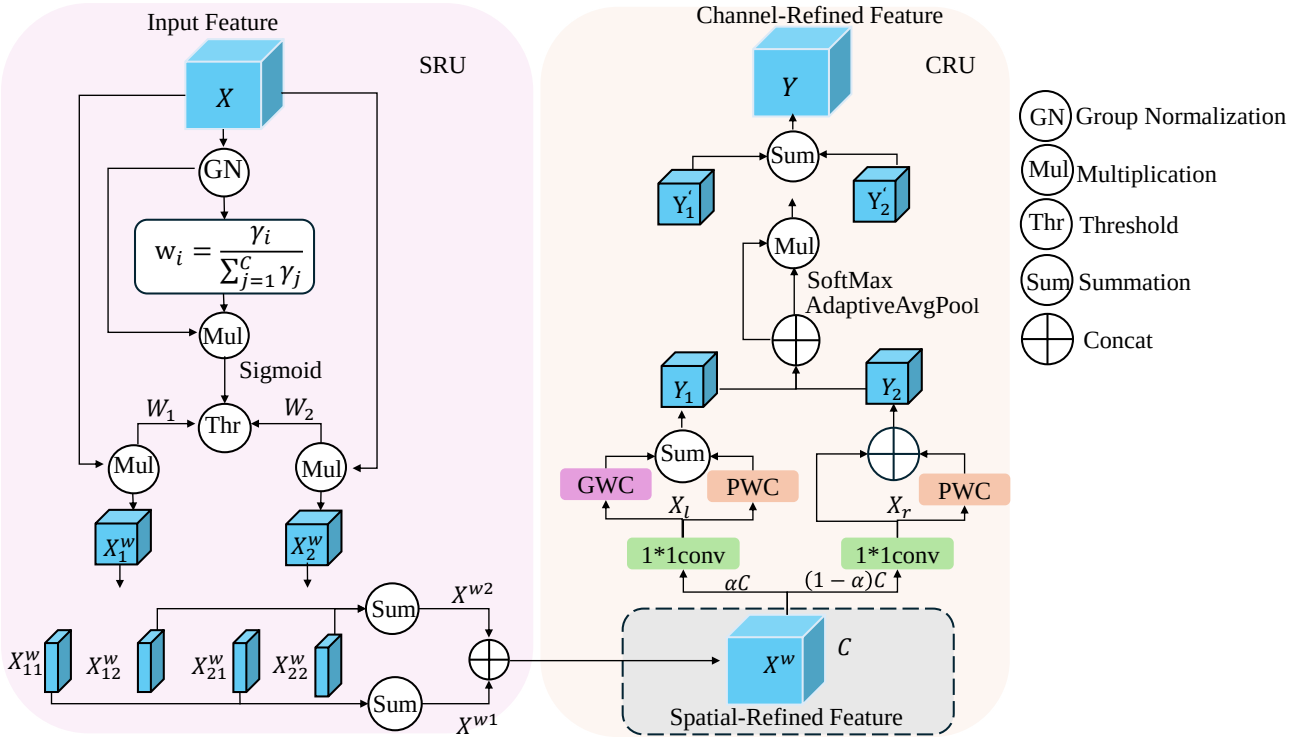


Fig. 5: The structure of SCConv

accuracy and coverage. The equations for precision and recall are presented below:

$$P = \frac{TP}{FP + TP}$$
$$R = \frac{TP}{TP + FN} \tag{7}$$

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

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

In these metrics, n indicates the number of classes, while AP corresponds to the precision for a single class. Params indicates the total amount of trainable parameters within the model, FLOPs reflects the computational quantity necessary for the model to carry out a single forward reasoning, and FPS measures the number of samples that the model can handle per second.

D. Visualization Analysis

To provide a clearer and more intuitive visualization of SDC-YOLO's detection performance, Figure 6 provides a comparative evaluation of the detection outcomes achieved by YOLOv8s and SDC-YOLO. The upper portion of Figure 6 displays the detection outcomes of the YOLOv8s algorithm, while the lower portion illustrates the results obtained by the SDC-YOLO algorithm.

By comparing these images, it is evident that the improved SDC-YOLO model demonstrates enhanced detection accu-

Fig. 6: Comparison of the detection results between YOLOv8s and SDC-YOLO

TABLE I: Ablation experiment

YOLOv8s	SPDconv	Dysample	CSC2f	P(%)	R(%)	mAP@0.5(%)	Params(M)	FLOPs(G)	FPS
√				67.2	56.5	62.1	11.1	28.7	107
√	√			64.8	60.4	62.9	13.3	31.3	100
√		√		69.3	55.1	63.3	11.1	27.5	108
√			√	63.1	58.7	63	10.5	27.4	112
√	√	√		71.6	53.2	63.8	13.3	30.1	101
√	√	√	√	75.1	56.5	64.4	12.7	29.2	105

racy and can identify targets that were previously undetected by the original YOLOv8s model. Specifically, SDC-YOLO significantly reduces missed detections of minor cracks and improves overall detection precision. This enhancement is particularly notable in complex road defect scenarios where small and irregularly shaped cracks are common.In summary, SDC-YOLO outperforms YOLOv8s by effectively addressing the limitations of the original model, thereby providing more reliable and accurate detection results.

The overall performance of the proposed SDC-YOLO model is assessed by means of the precision-recall (P-R) curves of SDC-YOLO and YOLOv8s. For specific details, please refer to Figure 7. An in-depth analysis of these P-R curves reveals the performance trends of each model, providing a more accurate reflection of their ability to identify positive examples and offering robust support for model evaluation and optimization.

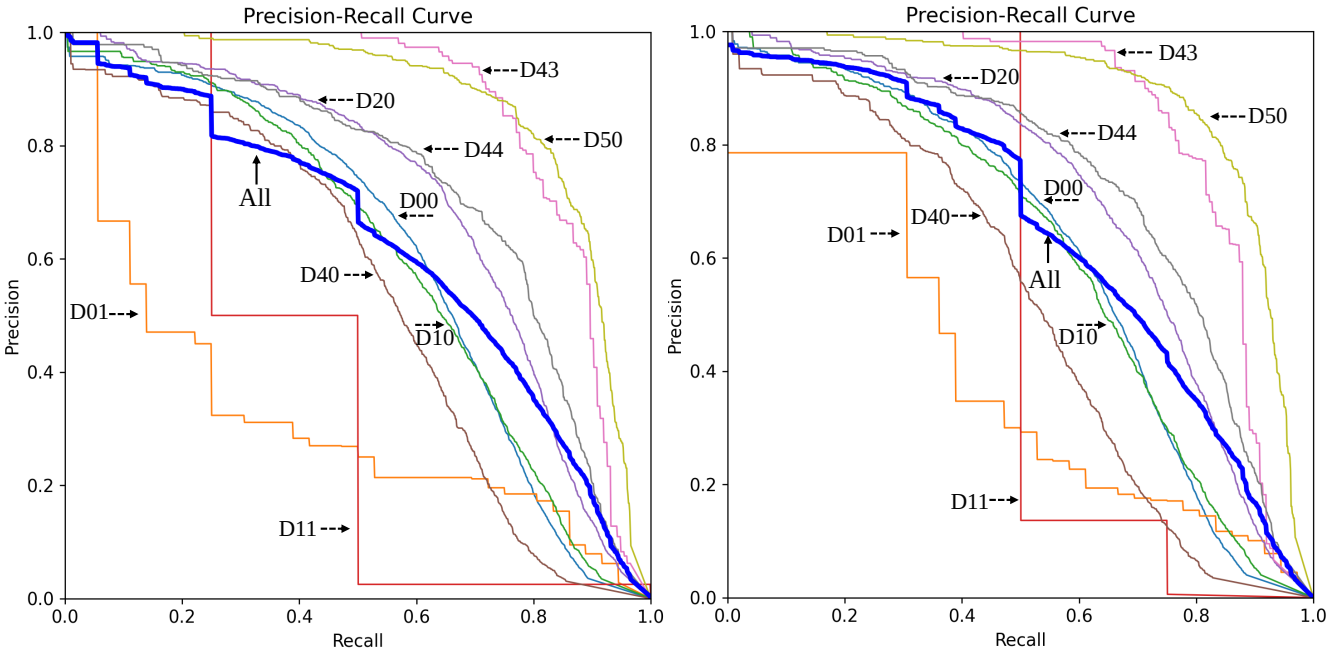
Specifically, Figure 7(a) illustrates the P-R curve for the YOLOv8s algorithm, while Figure 7(b) shows the P-R curve for the SDC-YOLO algorithm. The thicker blue curve represents the overall mAP@0.5, while the thinner curves in different colors correspond to the mAP@0.5 for individual categories. Notably, the mAP@0.5 for YOLOv8s is 62.1%, compared to 64.4% for SDC-YOLO, representing an improvement of 2.3%.

This enhancement highlights the improved efficiency of the SDC-YOLO algorithm, especially in its capacity to identify small and irregularly shaped road defects more accurately. The rise in mAP@0.5 suggests that SDC-YOLO attains greater accuracy while preserving an improved equilibrium between precision and recall, thus enhancing its reliability for real-world applications.

E. Ablation Study

The SDC-YOLO model is developed based on the YOLOv8s architecture and integrates three primary optimization strategies. To systematically evaluate the effectiveness of each measure, an ablation study was conducted to compare their individual and combined impacts on detection performance. Table I summarizes the ablation results as follows:

1. In the baseline model, traditional convolutional layers were replaced with SPDconv to enhance feature extraction



(a) P-R curve of YOLOv8s (b) P-R curve of SDC-YOLO

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

TABLE II: Performance comparison of mainstream algorithms

Model Name	P(%)	R(%)	mAP@0.5(%)	Params(M)	FLOPs(G)	FPS
Faster R-CNN [7]	69.1	65.2	66.8	38.2	47.3	53
YOLO-LRDD[24]	61	57.8	59.5	19.8	17.4	87
YOLOv5s	58.4	55.6	57.2	9.1	23.8	95
YOLOv6s [25]	50.1	56	56.4	16	44	72
YOLOv7-tiny [26]	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
YOLOv9S [27]	69.5	53	62.7	7.1	26.2	113
YOLOv10s [28]	70.8	58.5	63.3	7	21.4	115
ours	75.1	56.5	64.4	12.7	29.2	105

capabilities.

2. The ordinary upsampling operator was replaced with Dysample to enhance spatial resolution and better preserve details.

3. The standard C2f module in the baseline model was substituted with the CSC2f module to enhance feature fusion and minimize redundancy.

4. Two of the aforementioned measures were combined in the baseline model to evaluate their synergistic effects.

5. All three measures were jointly integrated into the baseline model to assess their comprehensive impact on overall performance.

As presented in Table I, the introduction of SPDconv resulted in a 0.8% increase in mAP@0.5, allowing the network to extract more detailed and nuanced features and thereby enhancing detection accuracy. More specifically, the integration of SPDconv enhanced the model's capacity to identify small and complex details, which in turn resulted in improved overall performance.

By replacing the nearest-neighbor interpolation upsampling operator with Dysample, mAP@0.5 increased by 1.2%, while FLOPS decreased by 1.2G and FPS improved slightly. This substitution not only boosted the model's accuracy but also significantly reduced computational overhead, making the model more efficient.

Additionally, the CSC2f module contributed to a 0.9% improvement in mAP@0.5, reduced the number of parameters by 0.6M, decreased FLOPs by 1.3G, and increased FPS by 5 frames per second. These optimizations effectively enhanced feature extraction, reduced resource consumption, and improved overall performance.

When all three measures were integrated, it was observed that the combination of Dysample and CSC2f mitigated the increase in parameters and FLOPs while further improving accuracy. In contrast to the original YOLOv8s model, SDC-YOLO attained greater detection accuracy while preserving a nearly identical detection speed. The experimental outcomes clearly highlight the efficacy of the proposed enhancements.

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

In order to confirm the superiority of the SDC-YOLO algorithm over other state-of-the-art object detection models, comparative experiments were carried out on the RDD2022 dataset. As shown in Table II, our SDC-YOLO model was benchmarked against several advanced algorithms, including Faster R-CNN [7], YOLO-LRDD [24] , YOLOv5s, YOLOv6s [25], YOLOv7tiny [26], YOLOv8n, YOLOv8s,

YOLOv9s [27], and YOLOv10s [28]. Using evaluation metrics such as mAP@0.5, Params, FLOPS, and FPS, we performed a comprehensive and precise assessment of the model's performance.

Specifically, while some algorithms achieved higher mAP@0.5 scores, our SDC-YOLO model demonstrated superior performance in terms of computational efficiency, featuring lower Params, reduced FLOPS, and improved FPS. Notably, compared with YOLOv9s and YOLOv10s, our model not only outperformed them in detection accuracy but also achieved a well-balanced compromise between accuracy and efficiency.

A holistic analysis of all indicators reveals that although our model has slightly slower detection speed compared to certain algorithms, it achieves a significant improvement in accuracy. This highlights the efficacy and efficiency of the proposed SDC-YOLO algorithm, rendering it especially appropriate for real-world applications where both accuracy and resource management are essential.

V. Conclusion

In order to tackle the problems of missed detections, false alarms, and low accuracy in road defect detection tasks, we introduce a new road defect detection model called SDC-YOLO, which is built upon the YOLOv8 algorithm. By incorporating SPDconv, the model significantly reduces information loss during feature extraction and enhances its capability to represent small-scale features. The lightweight dynamic upsampling module not only improves upsampling performance but also preserves critical details while reducing computational overhead. Additionally, the CSC2f module minimizes redundant features, optimizes the extraction process, and lowers overall complexity. Comprehensive experiments carried out on the RDD2022 dataset show that, in comparison with other leading-edge algorithms, the SDC-YOLO model attains higher performance with respect to both accuracy and efficiency. Specifically, it outperforms existing models in detecting small and intricate road defects, while maintaining a balanced trade-off between precision and resource utilization.The research presented in this paper not only advances the development of pavement defect detection technology but also establishes a scalable framework applicable to small target detection and real-time detection tasks. This work holds significant practical value across various domains, including intelligent transportation systems and automated road maintenance.

References

[1] S. Chatterjee, P. Saeedfar, S. Tofangchi, and L. M. Kolbe, "Intelligent road maintenance: a machine learning approach for surface defect detection." in *ECIS*, 2018, p. 194.

[2] O. M. V. OCCUPANTS, "National highway traffic safety administration (nhtsa) notes," *Annals of Emergency Medicine*, vol. 54, no. 2, 2009.

[3] A. Du and A. Ghavidel, "Parameterized deep reinforcement learning-enabled maintenance decision-support and life-cycle risk assessment for highway bridge portfolios," *Structural Safety*, vol. 97, p. 102221, 2022.

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

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

[6] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3708–3712.

[7] 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.

[8] 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.

[9] 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.

[10] L. Song and X. Wang, "Faster region convolutional neural network for automated pavement distress detection," *Road Materials and Pavement Design*, vol. 22, no. 1, pp. 23–41, 2021.

[11] 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.

[12] T. Lin, "Focal loss for dense object detection," *ArXiv Preprint ArXiv:1708.02002*, 2017.

[13] J. Redmon, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016.

[14] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, "Pavement distress detection and classification based on yolo network," *International Journal of Pavement Engineering*, vol. 22, no. 13, pp. 1659–1672, 2021.

[15] Y. Jiang, H. Yan, Y. Zhang, K. Wu, R. Liu, and C. Lin, "Rdd-yolov5: road defect detection algorithm with self-attention based on unmanned aerial vehicle inspection," *Sensors*, vol. 23, no. 19, p. 8241, 2023.

[16] Z. Han, Y. Cai, A. Liu, Y. Zhao, and C. Lin, "Ms-yolov8-based object detection method for pavement diseases," *Sensors (Basel, Switzerland)*, vol. 24, no. 14, p. 4569, 2024.

[17] 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.

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

[19] X. Zhang and Y. Tian, "Traffic sign detection algorithm based on improved yolov8s," *Engineering Letters*, vol. 32, no. 1, pp. 168–178, 2024.

[20] 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.

[21] R. Sunkara and T. Luo, "No more strided convolutions or pooling: A new cnn building block for low-resolution images and small objects," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2022, pp. 443–459.

[22] W. Liu, H. Lu, H. Fu, and Z. Cao, "Learning to upsample by learning to sample," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6027–6037.

[23] J. Li, Y. Wen, and L. He, "Scconv: Spatial and channel reconstruction convolution for feature redundancy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6153–6162.

[24] F. Wan, C. Sun, H. He, G. Lei, L. Xu, and T. Xiao, "Yolo-lrdd: A lightweight method for road damage detection based on improved yolov5s," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, p. 98, 2022.

[25] 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.

[26] 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.

[27] 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.

[28] 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.