Pavement Damage Recognition Based on Deep Learning

Mingbo Ning
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 450156598@qq.com

Shengquan Yang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: xaitysq@126.com

***Abstract*—Road surface disease detection is a vital component of road maintenance. Traditional deep learning-based detection methods face challenges such as low detection accuracy, high false alarm rates in complex scenarios, and significant missed detection rates for small targets like potholes. To address these limitations, this paper proposes an improved pavement disease detection algorithm based on RT-DETR. First, a lightweight backbone network named LMBANet is constructed by integrating DRB and ADown modules. This network enhances feature extraction capabilities without increasing computational overhead during inference, preserving local details of low-level features while expanding the receptive field to capture long-range semantic information and reduce false detection of diverse defects in complex scenes. Second, an small-target enhanced feature pyramid network is designed using SPDConv and OmniKernel. By feeding large-scale feature maps extracted by the backbone into a feature fusion layer and enhancing multi-scale feature representation through EFKM, this network resolves the high missed detection rate of small targets in the original model. Experimental results demonstrate that on the RDD2020 dataset, the improved network achieves an mAP of 69.2%, representing a 2.1 percentage point improvement over the original network, while simultaneously reducing parameters and computational costs.**

Keywords-Deep Learning; Road Surface Disease Detection; RT-DETR; Lmbablock; STEP

I. INTRODUCTION

Roads are critical components of the transportation system, with highway construction playing a particularly vital role in infrastructure development. Highway transportation significantly facilitates public travel and accelerates socioeconomic progress. However, pavement health issues can severely impact traffic safety. If maintenance is delayed until obvious pavement damage occurs, repair costs will escalate dramatically. Therefore, early detection and repair of potholes and cracks using intelligent inspection technologies are essential for ensuring transportation safety and reducing long-term maintenance expenses.

Early pavement damage identification and assessment methods primarily relied on manual inspections conducted by road maintenance workers. These workers would patrol the road network, visually inspecting and manually measuring various damage parameters to evaluate the overall pavement deterioration. Although this human-based approach offers simplicity and relatively high accuracy, it suffers from several significant drawbacks: the labor-intensive process is time-consuming and inefficient, often causing urban traffic congestion during inspections, which adversely impacts transportation efficiency and poses potential safety hazards. Consequently, manual inspections have gradually been replaced by specialized pavement inspection vehicles equipped with professional Charge-Coupled Device (CCD) cameras. These vehicles enable quantitative assessment of road defects through continuous video recording without disrupting normal traffic flow. However, they still require manual image processing for damage analysis, and their high operational costs fail to resolve the substantial consumption of human and financial resources.

With the remarkable success of deep learning technology, computer vision approaches have been widely adopted for pavement damage detection tasks. Current mainstream object detection models, however, struggle to balance computational complexity with detection performance. Models with high computational complexity face

deployment challenges in real-world scenarios, while lightweight models with reduced computations often exhibit insufficient detection accuracy, particularly showing susceptibility to false positives and missed detections under complex environmental conditions. These limitations hinder their ability to meet practical engineering requirements. To address these challenges, this paper proposes an enhanced model based on RT-DETR (Real-Time Detection Transformer), aiming to optimize both computational efficiency and detection reliability in pavement damage identification.

II. RELATED WORK RESEARCH

In recent years, with advancements in artificial intelligence and computer hardware technologies, scholars have progressively applied object detection models such as Faster R-CNN, YOLO, and DETR to pavement damage detection. These algorithms enable automatic identification of damaged road areas through single-image input while achieving satisfactory detection performance. Li et al. [1] employed Faster R-CNN to analyze 5,966 road defect images captured from diverse angles and distances. Experimental results demonstrated the model's robust detection capability under varying illumination conditions, effectively recognizing five categories of road defects: transverse cracks, longitudinal cracks, potholes, alligator cracks, and manhole-related defects.

The YOLO series of algorithms achieve extremely fast inference speeds while maintaining high detection accuracy, and their robust real-time detection capabilities have made them widely adopted in pavement damage recognition. Joseph Redmon et al. [2] introduced a feature pyramid network in YOLOv3 to leverage multi-scale feature maps for improving recognition accuracy of targets of varying sizes. Duan et al. [3] further enhanced cross-scale feature extraction by integrating a Bi-directional Feature Pyramid Network (BiFPN).

The success of Transformer models in natural language processing has demonstrated the exceptional capability of attention mechanisms in integrating global contextual semantic information. Researchers began exploring their applications in computer vision. Dosovitskiy et al. [4] proposed the Vision Transformer (ViT), a deep learning model specifically designed for computer vision tasks using self-attention mechanisms. ViT processes input images by dividing them into patchembeding, learning global contextual information through self-attention, and subsequently passing these features to fully connected layers for classification or regression tasks. However, ViT's global attention mechanism requires computing pairwise relationships between all image patches, resulting in quadratic computational complexity ($O(N^2)$) that poses challenges for high-resolution images and large-scale datasets.

Facebook AI [5] introduced DETR (Detection Transformer) in 2020 as an end-to-end global detection framework. DETR employs a CNN backbone for feature extraction followed by Transformer encoder-decoder layers for prediction. It replaces anchor generation with learnable object queries and utilizes a bipartite matching-based loss function to enforce one-to-one prediction matching, eliminating non-maximum suppression (NMS). Building upon DETR, Zhu et al. [6] proposed Deformable Attention to address the $O(N^2)$ complexity of standard attention, resolving slow convergence and high feature map dependency. Chen et al. [7] developed Group DETR, which employs multiple object queries to retain end-to-end inference advantages while accelerating convergence through one-to-many supervision. DINO [8] enhances detection robustness via contrastive denoising to reduce anchor dependency and improve occluded object recognition. Co-DETR [9] implements a collaborative hybrid training scheme with auxiliary detectors like ATSS and Faster R-CNN, enriching supervision signals for small object detection. MFDS-DETR [10] introduces a hierarchical semantic FPN (HS-FPN) to optimize multi-scale feature fusion, significantly boosting small target detection accuracy.

In 2023, Baidu's PaddlePaddle team [11] introduced RT-DETR (Real-Time Detection Transformer), a highly practical industrial-grade detector featuring an efficient hybrid encoder. This architecture combines an Attention-based Intra-scale Feature Interaction Module for contextual refinement and a CNN-based Cross-scale Feature-

fusion Module for multi-level integration, achieving real-time performance through computational redundancy reduction while maintaining detection precision.

Figure 1. Road surface damage dataset under different conditions

However, most existing approaches primarily predict pavement crack defects under conventional conditions, demonstrating limited robustness in complex environmental scenarios. As illustrated in Figure 1, these challenging scenarios include shadow interference, rainy conditions, color segmentation ambiguities, dense defect distributions, and pothole clusters. Current object detection algorithms generally suffer from three critical limitations: Similarity between defect features and background textures frequently causes false positives; The spatial continuity and linear characteristics of cracks often lead to misclassification alligator cracks as other defect types; Significant scale variations between defects result in frequent missed detections of small targets like potholes. To address these challenges while maintaining real-time detection capabilities, this paper proposes an enhanced RT-DETR-based model.

III. METHODS

A. RT-DETR Network

RT-DETR is a Transformer-based real-time object detection model that employs an HybriDencoder to reduce computational redundancy through decoupled intra-scale interactions and cross-scale fusion, while maintaining detection accuracy. By eliminating post-processing operations like non-maximum suppression (NMS), the algorithm achieves enhanced inference efficiency and fully leverages end-to-end advantages. Given the requirements for low computational overhead and high real-time performance in pavement defect detection tasks, this paper selects the relatively lightweight RT-DETR-r18 as the baseline model. The overall network architecture is illustrated in Figure 2.

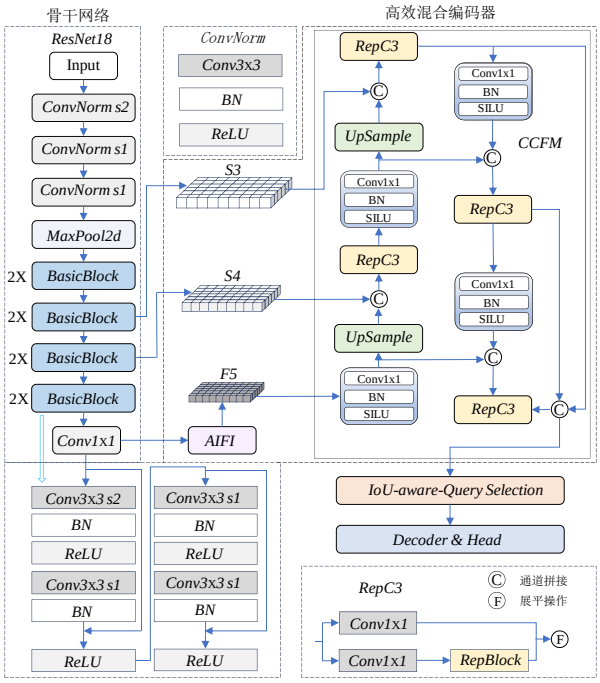

Figure 2. RT-DETR-r18 model structure

The model comprises three core components: Backbone, HybridEncoder, TransformerDecoder. RT-DETR adopts ResNet18 [12] as its backbone - a classical deep residual network characterized by shallow architecture and robust performance. Through residual blocks implementing cross-layer connections, ResNet18 effectively mitigates vanishing gradient issues. The hybrid encoder consists of two specialized modules: the Attention-based Intra-scale Feature Interaction (AIFI)

module and the CNN-based Cross-scale Feature Fusion (CCFM) module.

The input image first undergoes multi-scale feature extraction through the backbone network. High-level semantic features from the S5 layer are then flattened and processed by the AIFI module with positional encoding. Multi-head attention mechanisms execute intra-scale feature interactions within AIFI, with the output subsequently reshaped into 2D features (denoted as F5) for cross-scale fusion. The CCFM module inserts convolutional Fusion Blocks into the fusion path to integrate adjacent-scale features. Finally, IoU-aware queries select fixed-length features from the encoder's output sequence as initial object queries for the decoder. These queries are optimized through auxiliary pre-detection heads to generate final class predictions and bounding boxes. The representation process is:

$$Q = K = V = Flatten(S5) \tag{1}$$

$$F5 = reshape\left(Attn(Q,K,V)\right) \tag{2}$$

$$Output = CCFM\left(\{S3,S4,F5\}\right) \tag{3}$$

Among them, flatten denotes the flattening operation, Attn refers to multi-head self-attention, and reshape represents the process of restoring features to the same shape as S5.

B. Improving RT-DETR Network

The improved model utilizes a more lightweight network compared to ResNet18 for shallow feature extraction, achieving a larger effective receptive field to capture long-range semantic information. The input image generates four-scale feature maps S2, S3, S4, and S5 through the backbone network. Among them, the S5 feature is encoded into F5 within the original model's AIFI module. S2, S3, S4, and F5 are then fed into an enhanced small-object feature pyramid fusion network. The upsampled F5 feature map is concatenated with the S4 feature map along the channel dimension. The resulting output is upsampled again and concatenated with the S2 feature map processed by SPDConv along the channel dimension. The final output undergoes EFKM processing to generate a feature map containing small-scale information. Through a series of multi-scale feature fusions, the model ultimately produces a comprehensive feature map with effective information across all scales, which is then input into the decoder.

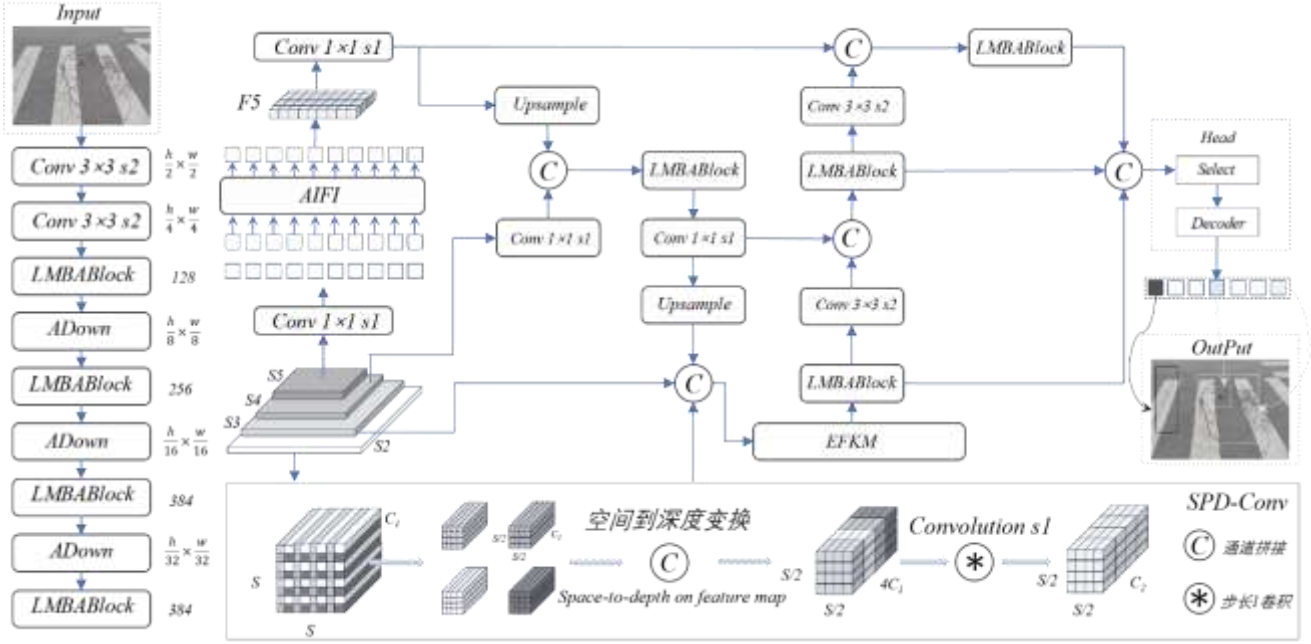

Figure 3. Improved RT-DETR model structure

C. LMBANet

The coexistence of multiple pavement defects often leads to model misdetections across various damage types. For instance, in complex scenarios, there exists significant similarity between alligator cracks and transverse cracks, as illustrated in Figure 4. Such cases may cause misclassification between crack types, subsequently affecting maintenance crews' root cause analysis and targeted repair strategies. To address this challenge, we integrate GELAN with Dilated convolution principles to design a Long-range feature extraction backbone network.

Figure 4. Diagrams of different types of cracks

GELAN [13] is an efficient aggregation network combining CSPNet architecture with gradient path optimization, enabling effective propagation and integration of multi-level feature information. The network partitions input feature tensors into two streams: one preserves original features through identity mapping, while the other undergoes multi-layer convolutional operations to extract higher-level abstractions. These streams are concatenated through multi-stage channel-wise fusion.

The Dilated Re-param Block (DRB) [14] enhances feature representation through a re-parameterization mechanism based on dilated convolutions. During training, the module employs a 7×7 non-dilated convolution layer parallel with three dilated convolutional branches {kernel sizes=5,3,3, dilation rates=1,2,3}. Outputs from these branches are batch-normalized and aggregated additively. During inference, re-parameterization converts the entire structure into an equivalent single non-dilated convolution layer, eliminating computational overhead from auxiliary branches.

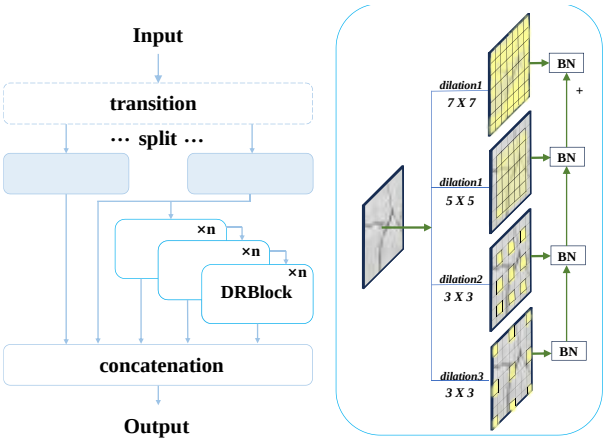

Figure 5. Structure of LMBA module

We integrate DRB into GELAN's branch pathways to create a Long-Road Multi-branch Aggregation Block (LMBABlock), as detailed in Figure 5. Replacing original feature extraction modules, DRB-enhanced branches capture multi-receptive-field features. The aggregated multi-scale features from parallel branches enable long-range semantic understanding. The input features first undergo channel and spatial dimension adjustment through a convolutional layer, before being processed by the LMBABlock to extract multi-scale features with large receptive fields. These features are subsequently downsampled through the Adown [14] module - an innovative downsampling component that splits the input features into two parallel paths: one path employs stride-3 convolution to preserve original structural information, while the other utilizes max pooling to extract salient features. Through the stacked configuration of LMBABlock and ADown modules, the complete backbone network architecture is constructed, as shown in the left portion of Figure 5.

D. STEP

Potholes, as typical small-scale targets in pavement damage detection, often suffer from information degradation during feature propagation from shallow to deep layers. Due to the inherent locality of feature mapping and varying receptive field scales across network depths, fine-grained details in abstract feature maps are progressively weakened, leading to frequent missed detections of small targets. Figure 6(a) illustrates the original cross-scale fusion network in RT-DETR, which

constructs top-down and bottom-up feature pyramid pathways for multi-scale interactions. However, this interaction initiates from the P3 detection layer, inherently limiting the model's capacity to preserve small-scale semantic information. Traditional improvement approaches, as shown in Figure 6(b), address this by adding a P2 small-target detection layer, but inevitably introduce excessive computational overhead. To resolve this dilemma, we propose an small-target enhanced feature pyramid architecture specifically optimized for small targets, depicted in Figure 6(c). The P2 feature map first undergoes SPDConv [15] to enrich small-target representations, then employs our improved EFKM (Efficient Full Kernel Module) derived from OmniKernel [16] Module for efficient feature consolidation while maintaining computational efficiency.

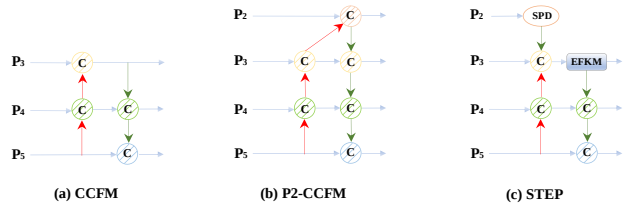

Figure 6. Comparison of feature Pyramid net

The SPDConv module comprises a Space-to-Depth (SPD) layer followed by a non-strided convolution layer, with its architectural details illustrated in the lower section of Figure 3. The SPD layer reduces the spatial dimensions while expanding the channel dimensions of the input feature map, effectively preserving spatial information without loss. After processing through SPDConv, the resulting P2-level feature maps undergo cross-scale fusion with P3 and P4 features within the EFKM to integrate multi-resolution representations.

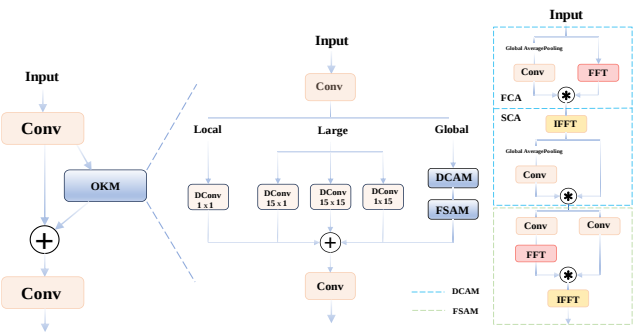

Figure 7. Structure of EFKM module

The EFKM (Efficient Full Kernel Module) architecture is illustrated in Figure 6. Given input features $X \in RC \times H \times W$ from the OKM (Omni-Kernel Module), the features undergo 1×1 convolutional processing before being distributed to three parallel branches: the local branch, large kernel branch, and global branch, which collectively enhance multi-scale representations. The outputs from these branches are aggregated through element-wise summation and subsequently modulated by another 1×1 convolution.

The large kernel branch employs a computationally efficient large-kernel depthwise convolution ($K \times K$) to capture extensive receptive fields. Complementing this, parallel $1 \times K$ and $K \times 1$ depthwise convolutions are utilized to extract strip-shaped contextual information. To address the limitation of large kernels in achieving global coverage, the global branch incorporates a Dual-domain Channel Attention Module (DCAM) and a Frequency-based Spatial Attention Module (FSAM). For input features $X_{Global} \in R^{C \times H \times W}$, the DCAM first applies Frequency Channel Attention (FCA), expressed as:

$$X_{FCA} = IF\left(F\left(X_{Global}\right)\right) Conv \{ \left\{GAP\left(X_{Global}\right)\right\} \quad (4)$$

Where F and IF denote Fast Fourier Transform (FFT) and its inverse, respectively. The operator

$\odot$ represents element-wise multiplication, while GAP and Conv indicate global average pooling and 1×1 convolution. Optimized features from FCA are then fed into the Spatial Channel Attention (SCA) module as described in equation:

$$X_{DCAM} = X_{FCA} \{ Conv\left\{GAP\left(X_{FCA}\right)\right\} \quad (5)$$

Here, XDCAM represents the output of DCAM. Following channel-wise enhancement, FSAM performs fine-grained spectral refinement in the spatial dimension through frequency-based attention mechanisms, formally defined as:

$$X_{FSAM} = IF(W1 \{ W2) \quad (6)$$

$$W1 = F\left(Conv\{ X_{DCAM} \}\right) \quad (7)$$

$$W2 = Conv\left\{ X_{DCAM} \right\} \quad (8)$$

Where W1 and W2 derive from frequency-domain and spatial-domain transformations of XDCAM, respectively. This enables the module to prioritize frequency components carrying critical semantic information. In addition to the large kernel branch for extended receptive fields and the global branch for full-scale coverage via dual-domain processing, a lightweight local branch supplements local detail preservation through a simple 1×1 depthwise convolution.

IV. EXPERIMENTS

A. Experimental Environment

Table I shows the experimental environment in this paper, which is based on the Ubuntu 18.04 operating system, the graphics card model is RTX4090D, and the memory is 24GB. The experiment basically uses the parameters recommended by RT-DETR, builds the model based on Python3.9 and Pytorch1.13.1 framework, and uses the standard SGD optimizer, with batch-size set to 8 and epochs set to 150.

TABLE I. EXPERIMENTAL ENVIRONMENT

Experimental environment	Version
CPU	Intel Xeon Platinum 8352V
GPU	NVIDIA GeForce RTX4090D
Language	Python3.9
Deep Learning Framework	Pytorch1.13.1
CUDA	11.6.0

B. Dataset

In this experiment, we utilized the publicly available RDDC2020 [17] dataset provided by the Global Road Damage Detection Challenge. The original RDD2020 dataset comprises 26,336 road images collected from India, Japan, and the Czech Republic. To better align with domestic road surface environments, a subset of 9,600 images demonstrating similar characteristics to Chinese pavement conditions was carefully selected for our study. Following standard experimental protocols, the dataset was partitioned into training and testing sets, with 80% allocated for training purposes and the remaining 20% reserved for testing. The quantitative distribution of different damage category labels is systematically presented in Table 2, illustrating the sample statistics across various defect types.

TABLE II. DISEASE CATEGORY

Category	Train Set	Test Set
D00(Longitudinal cracks)	7419	876
D10(Transverse cracks)	5702	636
D20(Alligator cracks)	6244	689
D40(Potholes)	2316	248

C. Evaluation Metrics

In this study, the following evaluation metrics were adopted: precision (P), recall (R), average precision (AP), mean average precision (mAP), model parameter count, and computational complexity measured in Giga Floating-point Operations Per Second (GFLOPs). The mAP metric, one of the most widely used benchmarks for object detection performance, is derived from the precision-recall relationship. Its calculation procedure follows the equations below [18]:

$$P = TP/(TP+FP) \tag{9}$$

$$R = TP/(TP+FN) \tag{10}$$

$$AP = \int_0^1 P(R)d(R) \tag{11}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{n} APi \tag{12}$$

Where TP denotes true positives (correctly detected positive samples), FP represents false positives (negative samples erroneously classified as positive), FN indicates false negatives (positive samples misclassified as negative), N is the total number of damage categories, and APi denotes the detection accuracy for the ii-th category, calculated through precision-recall integration.

Parameter count quantifies model size, while computational complexity (GFLOPs) evaluates the arithmetic operations required during inference.

Models with lower parameter counts and computational demands are prioritized for lightweight deployment scenarios, as they reduce hardware resource requirements while maintaining detection efficacy.

D. Algorithm verification results

The detection performance comparison between RT-DETR and its improved variant on the test set is systematically summarized in Table 3.

TABLE III. COMPARISON BEFORE AND AFTER IMPROVEMENT

Algorithm	Pars/M	FLOPS/G	FPS/f/s	mAP/%
RT-DETR	19.8	57.3	69	67.1
Improved RT-DETR	14.6	45.2	60	69.2

As evidenced by the quantitative results, the enhanced model demonstrates superior detection accuracy across all damage categories, achieving a 3.8 percentage point improvement for small-target D40 potholes, along with 3.2 and 2.2 percentage point gains for easily confounded D10 and D20 defects under complex scenarios. The overall mean average precision (mAP) shows a marked enhancement, while model parameter count and computational complexity are reduced by 29% and 10%, respectively, compared to the baseline. Although the frames per second (FPS) slightly decreases from 69 to 60, this operational speed remains well above the 30 FPS threshold required for practical road damage detection systems deployed on vehicular or drone platforms. Although the frames per second (FPS) slightly decreases from 69 to 60, this operational speed remains well above the 30 FPS threshold required for practical road damage detection systems deployed on vehicular or drone platforms.

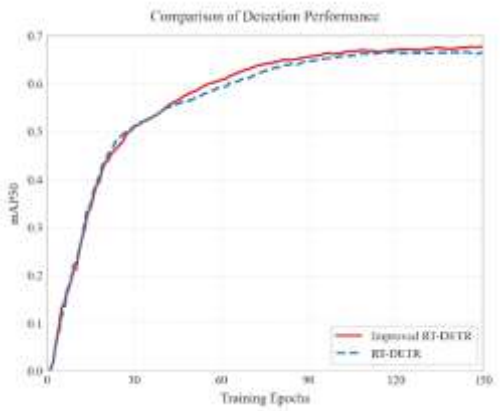

Figure 8. Comparison chart of mAP during training

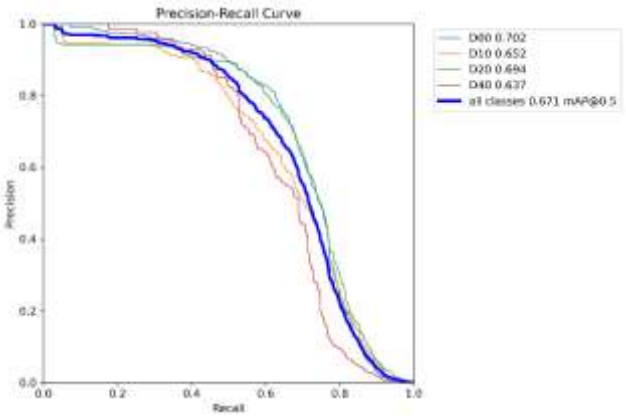

Figure 9. Average precision of each label in RT-DETR

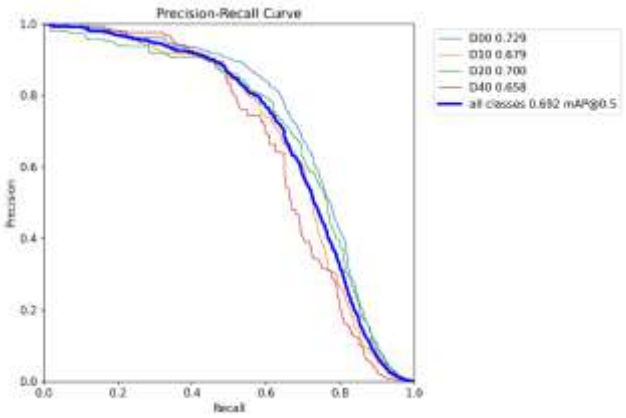

Figure 10. Average precision of each label in Improved RT-DETR

Figures 8-10 provide detailed performance analyses: Figure 8 contrasts the mAP evolution during training between the original and improved models, while Figures 9 and 10 visualize their precision-recall characteristics on the test set. The

baseline RT-DETR's suboptimal detection of transverse cracks and potholes stems from its limited receptive field, which frequently misclassifies transverse cracks as reticular counterparts. In contrast, the enhanced architecture strategically integrates local texture patterns with global semantic contexts through multi-scale feature fusion, thereby acquiring significant advantages in small-target recognition and spatial relationship modeling.

Figure 11 presents the detection outcomes of the algorithm before and after improvement in different scenarios of the selected dataset. From left to right, the scenarios are normal conditions, color interference, dense diseases, dense small targets, and low - light conditions. As can be seen from Figure 11, the algorithm improved by introducing the enhanced small-object feature pyramid network managed to identify the tiny potholes that RT - DETR failed to detect in the dense small - target scenario. Moreover, in the dense - disease and color - interference scenarios, the improved algorithm did not mix up transverse cracks with networked cracks.

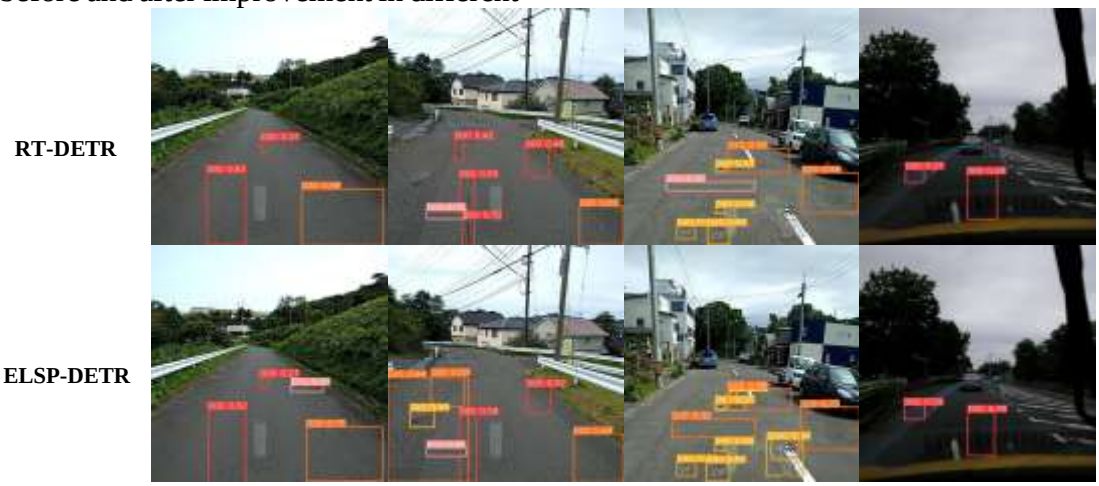

Figure 11. Visual comparison of test results

E. Ablation experiment

TABLE IV. COMPARISON BEFORE AND AFTER IMPROVEMENT

Experiments	LMBAN	STEP	Pram/M	FLOPs/G	mAP/%
I			19.8	57.3	67.1
II	✓		12.8	41.9	68.3
III		✓	20.5	59.5	68.9
IV	✓	✓	14.6	45.2	69.2

The model improvement is based on the RT-DETR architecture. To validate the effectiveness of each modification, ablation experiments evaluating detection accuracy and computational resource consumption were conducted using the dataset adopted in this study with results presented in Table 4.

The original RT-DETR model's performance metrics are shown in the first experimental configuration. Replacing its backbone network improved model accuracy by 1.2 percentage points while reducing parameters by 35% and computational cost by 26%, demonstrating efficiency gains without sacrificing detection capability. Substituting the original CCFM structure with STEP increased mAP by 1.8 percentage points compared to the baseline, indicating enhanced representation of small-scale features despite higher computational requirements. Combining both modifications achieved 2.1 percentage point mAP improvement over the original model while reducing parameters by 26% and computational cost by 21%.

F. Comparison experiment

To further validate the superiority of the improved algorithm for pavement disease detection, comparative experiments were conducted between

the proposed algorithm and conventional object detection algorithms. All experiments were performed under identical software and hardware environments using the same dataset, with results presented in Table 5.

Table 5 demonstrates that the improved algorithm achieves the highest accuracy among all compared methods. [19-20] Meanwhile, its parameter count and computational cost are significantly lower than those of other mainstream algorithms, enabling better adaptability of the model in edge device environments with limited computational resources.

TABLE V. COMPARISON BEFORE AND AFTER IMPROVEMENT

Algorithm	Pars/M	FLOPS/G	FPS/s/f	mAP/%
RT-DETR	19.8	57.3	69	67.1
Yolov11m	20.1	68.0	107	67.9
Fast-RCNN	136.5	370.2	21	50.2
Improved RT-DETR	14.6	45.2	60	69.2

V. COPYRIGHT FORMS AND REPRINT ORDERS

This paper addresses the issues of high false detection rates in complex road damage detection scenarios and missed detection of potholes by improving the RT-DETR network model. We propose an efficient backbone network for long-range semantic feature extraction to reduce computational overhead and mitigate false detections in complex environments. Additionally, a feature pyramid network incorporating Full Kernel modules and SPDConv is introduced to small-target enhanced feature pyramid network, specifically addressing the problem of missing tiny potholes. A series of experiments have demonstrated the effectiveness of the proposed algorithm. While the improved model shows enhanced detection performance, there remains room for optimization as it still exhibits relatively high computational complexity and parameter volume, along with decreased FPS compared to the original RT-DETR. Future work will focus on optimizing the model scale and improving detection speed.

REFERENCES

[1] Hao S, Shao L, Wang S. A Faster RCNN Airport Pavement Crack Detection Method Based on Attention Mechanism [J]. Academic Journal of Science and Technology, 2022, 4(2): 129-132.

[2] Redmon J, and Farhadi A. YOLOv3: An Incremental Improvement [J]. CoRR, 2018, 1804: 02767.

[3] Wu L, Duan Z, Liang C. Research on asphalt pavement disease detection based on improved YOLOv5s[J]. Journal of Sensors, 2023, 2023(1): 2069044.

[4] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16x16 words: Transformers for image recognition at scale [J]. arXiv preprint arXiv:2010.11929, 2020.

[5] CARIONN, MASSAF, SYNNAEVE G, et al. End-to-end object detection with transformers[C]// Proceedings of the 2020 European Conference on Computer Vision. Cham: Springer International Publishing, 2020: 213-229.

[6] Zhu X, Su W, Lu L, et al. Deformable detr: Deformable transformers for end-to-end object detection[J/OL]. arXiv preprint arXiv, 2020[2024-11-18]. https://doi.org/10.48550/arXiv.2010.04159

[7] CHEN Q, CHENX, WANGJ, et al. Group detr: Fast detr training with group-wise one-to-many assignment [C]//Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2023: 6633-6642.

[8] Zhang H, Li F, Liu S, et al. Dino: Detr with improved denoising anchor boxes for end-to-end object detection [J]. arXiv preprint arXiv:2203.03605, 2022.

[9] Zong Z, Song G, Liu Y. Detrs with collaborative hybrid assignments training [C]//Proceedings of the IEEE/CVF international conference on computer vision. 2023: 6748-6758.

[10] Chen Y, Zhang C, Chen B, et al. Accurate leukocyte detection based on deformable-DETR and multi-level feature fusion for aiding diagnosis of blood diseases[J]. Computers in Biology and Medicine, 2024, 170: 107917.

[11] Zhao Y, Lv W, Xu S, et al. Detrs beat yolos on real-time object detection [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 16965-16974.

[12] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[13] Wang C Y, Yeh I H, Mark Liao H Y. Yolov9: Learning what you want to learn using programmable gradient information [C]//European conference on computer vision. Cham: Springer Nature Switzerland, 2024: 1-21.

[14] Ding X, Zhang Y, Ge Y, et al. UniRepLKNet: A Universal Perception Large-Kernel ConvNet for Audio Video Point Cloud Time-Series and Image Recognition [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 5513-5524.

[15] Sunkara R, Luo T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects [C]//Joint European conference on machine learning and knowledge

discovery in databases. Cham: Springer Nature Switzerland, 2022: 443-459.

[16] Cui Y, Ren W, Knoll A. Omni-Kernel Network for Image Restoration [C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 38(2): 1426-1434.

[17] Arya D, Maeda H, Ghosh S K, et al. RDD2020: An annotated image dataset for automatic road damage detection using deep learning [J]. Data in brief, 2021, 36: 107133.

[18] Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (voc) challenge[J]. International journal of computer vision, 2010, 88: 303-338.

[19] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks [J]. IEEE transactions on pattern analysis and machine intelligence, 2016, 39(6): 1137-1149.

[20] Khanam R, Hussain M. Yolov11: An overview of the key architectural enhancements [J]. arXiv preprint arXiv:2410.17725, 2024.