Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.ijamce.com

A Real-Time Pavement Crack Detection Method Based on an Improved Lightweight Yolov8 Model

Zeqi Liu[a], Hui Yao[a], Xiaoyue Zhong[a], Zhaopeng Deng[a, b*]

[a] *College of Information and Control Engineering, Qingdao University of Technology, Qingdao 266525, China*

[b] *Innovation Institute for Sustainable Maritime Architecture Research and Technology, Qingdao University of Technology*

ARTICLE INFO

Article history:
Received 14 May 2024
Accepted 22 July 2024
Available online 3 August 2024

Keywords:
Crack detection
Lightweight network
YOLOv8
Attention mechanism

ABSTRACT

The automatic detection of pavement cracks plays a crucial role in road maintenance. However, existing detection methods often suffer from large model sizes and slow detection speeds, limiting their application in pavement crack detection systems. To address this, this paper proposes a pavement crack detection method based on an improved lightweight YOLOv8 model. First, the ShuffleNetV2 lightweight network is used to replace the C2f module in YOLOv8, improving the model's inference speed and thus meeting the requirements for real-time detection. Moreover, the Convolutional Block Attention Module (CBAM) is integrated to improve the model's ability to detect small cracks by enhancing focus on both channel and spatial features. Finally, the head structure of YOLOv8 is optimized to better handle challenges posed by complex pavement textures and variations in lighting conditions. The experimental results demonstrate that the size of the improved model is 3.8MB, representing a reduction of 74%, 39%, and 34% compared to YOLOv5, YOLOv8, and YOLOv10, respectively. Additionally, the number of parameters in the model is reduced by 43% compared to the original YOLOv8 model, and this smaller parameter size significantly enhances the detection speed of the model. Overall, the model demonstrates significant advantages in crack detection speed and model size, highlighting its potential for practical applications.

1. Introduction

Pavement cracks are a typical manifestation of early damage to roads, which not only affect driving comfort but also pose a significant threat to traffic safety. With the increase in traffic volume and the extended lifespan of roadways, the formation and propagation of cracks have gradually become a critical issue in road maintenance. Timely and accurate detection and assessment of pavement cracks are essential for formulating maintenance strategies, ensuring driving safety, and prolonging the service life of infrastructure. Traditional crack detection methods primarily rely on manual inspections, which are not only time-consuming and labor-intensive but also susceptible to subjective biases. As a result, these methods fail to meet the efficiency and accuracy demands of modern road maintenance.

To enhance the automation of crack detection, image processing-based detection methods have garnered widespread attention in recent years (Kheradmandi et al., 2022; Yang et al., 2022). Traditional image processing techniques include edge detection, threshold segmentation, and morphological operations (Chen et al., 2022; Dai et al., 2020). These methods can recognize pavement cracks to some extent. However, due to the complex and variable nature of the pavement environment, challenges such as texture interference, uneven lighting, and diverse crack shapes often hinder the effectiveness of traditional approaches. Particularly when dealing with small or irregularly shaped cracks, the detection accuracy of these methods significantly decreases, leading to serious issues of missed detections and false detections.

With the rapid development of deep learning technology, object detection methods based on Convolutional Neural Networks have gradually become a research hotspot in the field of image recognition (Dhillin et al., 2020; Sun et al., 2021). The YOLO (You Only Look Once) series of models, as a representative algorithm, is widely applied due to its real-time capabilities and efficiency (Jiang et al., 2022; Diwan et al., 2023). The YOLO model approaches the object detection task as a regression problem, enabling the prediction of object locations and classes through a single forward pass. Wang and others proposed an improved YOLOv5 model that incorporates a Vision Transformer (ViT) module into the YOLOv5 architecture.

* Corresponding author.
E-mail addresses: dengzhaopeng@qut.edu.cn (Z. Deng)
Digital Object Identifiers: https://doi.org/10.62953/IJAMCE.188520

This model demonstrates high accuracy and speed in detecting vertical, horizontal, and fatigue cracks (Wang et ai., 2023). Xing and others built upon the YOLOv5 network framework by integrating a Swin Transformer structure and a Bidirectional Feature Pyramid Network (BiFPN) to enhance feature extraction capabilities. They achieved real-time pixel-level detection of pavement cracks (Xing et al., 2023). Zhou and others proposed integrating a squeeze excitation network into the YOLOv5s model and aligning the anchor points, calculated using the K-means clustering algorithm, closely with the fracture dataset, thereby improving the model's recall rate (Zhou et al., 2024). Hu et al. enhanced crack focus within the YOLOv5 network framework by incorporating a weighted attention mechanism and optimized the training process using the Silu activation function and CIoU loss function. This approach effectively resolved the issues of blurred small cracks and incomplete information extraction from vehicle-mounted images (Hu et al., 2024). Ma et al. proposed a novel end-to-end task-integrated convolutional neural network architecture, YOLO-Crack, which enables the simultaneous output of crack object detection and image segmentation results (Ma et al., 2023).

Although many scholars have conducted extensive research and analysis on various types of pavement crack images, some advanced pavement detection algorithms often introduce complex network structures to improve detection accuracy. This, however, leads to increased computational resource demands and higher computational costs, ultimately affecting real-time performance. Large deep learning models typically consume significant memory and GPU resources, especially on mobile or embedded devices, where limited hardware resources can hinder real-time detection. Therefore, optimizing model size and reducing memory usage is a major challenge in achieving real-time detection.

To address the aforementioned challenges, this paper presents a pavement crack detection method based on an improved YOLOv8 model. In this approach, a lightweight network is employed to replace the C2f module in the original YOLOv8 model, reducing the model's parameter count and computational complexity, thereby shortening inference time. The model is further enhanced by integrating a Convolutional Block Attention Module to improve its ability to focus on detailed crack features (Fu et al., 2021). By calculating channel and spatial attention in parallel, CBAM more effectively extracts key region features, enhancing the model's detection accuracy for fine cracks. Additionally, the head structure of YOLOv8 is optimized to better address challenges posed by complex pavement textures and lighting variations. Experiments on publicly available pavement crack datasets validate the superiority of the improved model in terms of detection speed and model size.

2. Methods

2.1 Framework of Pavement Crack Detection Algorithm Based on Improved YOLOv8

Compared to previous versions, YOLOv8 introduces improvements in model architecture, loss functions, and training strategies, leading to enhanced detection accuracy and speed (Sohan et al., 2024). Its network structure can be divided into four main components: Backbone, Neck, Head, and Output. The Backbone is responsible for extracting multi-level features, typically through a convolutional neural network (CNN) that progressively downsamples the image to capture high-level semantic information. The Neck enhances features through multi-scale feature fusion (FPN or PANet), improving the detection of small objects. The Head simultaneously performs object classification and localization, predicting bounding boxes and their confidence scores. This unified design, characteristic of the YOLO series, further boosts computational efficiency. The Output layer applies Non-Maximum Suppression (NMS) to eliminate redundant boxes, ultimately outputting the object locations and categories, optimizing the balance between detection accuracy and speed.

However, as the complexity increases, the model's demand for hardware resources and training time also rises significantly, making it potentially unsuitable for all application scenarios. In addition, the original design of the model is primarily aimed at general object detection tasks, and it still has limitations in terms of detecting small objects and recognizing targets in complex backgrounds. To address these issues, this paper proposes an improved lightweight YOLOv8 model, as illustrated in Figure 1. The model retains the framework structure of YOLOv8, consisting of the Input layer, Backbone, Neck, and Head. In the Backbone section, a Conv_maxpool module is employed to reduce the parameter count and computational cost of subsequent layers. Additionally, three ShuffleNetV2 basic modules are stacked to further enhance the lightweight performance of the model. Meanwhile, CBAM is introduced in the feature maps after multi-scale feature extraction at the fourth layer, to enhance attention to key information while suppressing irrelevant features and noise, thereby improving the discriminative ability of the features. Subsequently, the high-scale features are deeply extracted through three repeated ShuffleNetV2 modules. In the Head section, an additional detection head specifically designed for small objects has been incorporated, expanding the model's detection range and significantly improving its perception and detection accuracy for small targets.

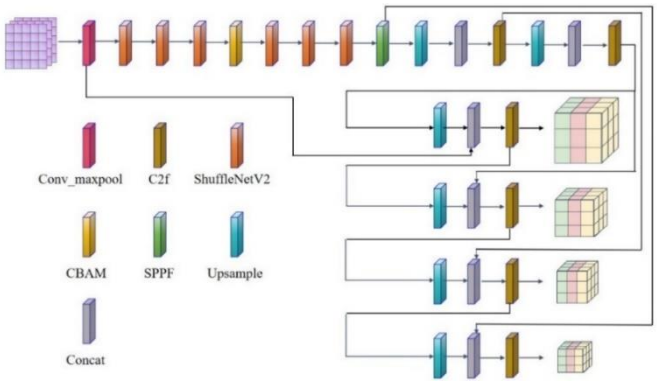


Fig. 1. Improved YOLOv8 network structure

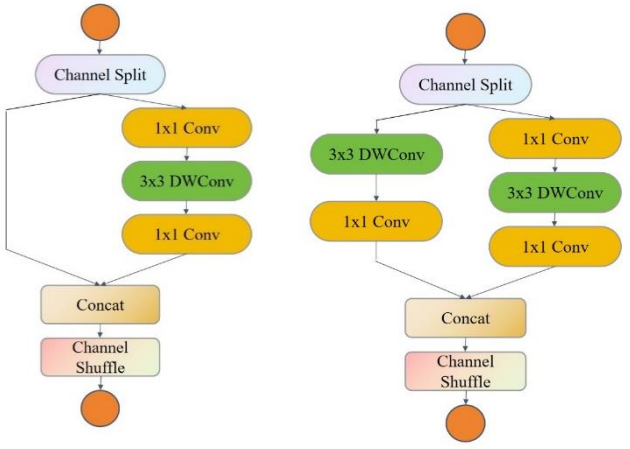
2.2 ShuffleNetV2 lightweight

As the network deepens, the number of channels in the feature maps increases significantly, which may lead to redundancy due to the presence of duplicated or similar information across different channels. To address this issue, ShuffleNetV2 introduces an efficient channel shuffle mechanism (Hao et al., 2022). By rearranging the channels of the feature maps after convolution, the channel shuffle ensures interaction between features from different groups, allowing each layer of the network to fully utilize the information from all channels while maintaining computational efficiency and reducing redundancy.

This paper draws on the design principles of ShuffleNetV2 and proposes an improved structure to replace the C2f module in YOLOv8, aiming to achieve a lightweight design and improve computational efficiency. The architecture of ShuffleNetV2 is illustrated in Figure 2. At the beginning of the module, the input feature map is divided into two parts through Channel Split, where the two branches are designed with different structures according to the design guidelines. One of the branches remains unchanged and is passed directly, while the other branch consists of three convolutional layers, where the number of input and output channels is kept consistent. The principle of grouped convolution is to divide the input feature channels into multiple groups and perform convolution operations on each group independently, thereby reducing computational complexity. The computation formula for standard convolution is shown in Equation 1.

$$Y = W * X \tag{1}$$

where X represents the input feature map, W is the convolution kernel, and Y is the output feature map. After the convolution operation, the output feature maps from the two branches are concatenated to maintain the original number of channels. This process does not involve element-wise addition. The concatenated feature map undergoes a channel shuffle operation to ensure effective communication between the different branches. These modules are then repeatedly stacked to construct the entire ShuffleNetV2 network. By adhering to these design principles, ShuffleNetV2 demonstrates high efficiency in improving both computational performance and model accuracy. In this paper, the architecture of the YOLOv8 model is optimized by integrating the efficient computational modules of ShuffleNetV2, successfully achieving a lightweight design and significantly enhancing the model's runtime efficiency while maintaining accuracy.



(a) Basic module (b) Downsampling module

Fig. 2. ShuffleNetv2 structure diagram

2.3 CBAM attention mechanism

In YOLOv8, the backbone network employs a multi-scale feature fusion strategy, where features extracted from different layers (p3, p4, p5) are concatenated with features of the same scale in the detection head for further processing. While this fusion strategy effectively utilizes multi-scale feature information to enhance detection accuracy, direct concatenation as the number of channels increases can lead to redundancy. This, may cause the model to focus on irrelevant features, negatively impacting overall performance.

To address this issue, this paper introduces the CBAM, whose structure is shown in Figure 3. CBAM adaptively assigns weights to both the channel and spatial dimensions of the feature map, effectively filtering out more critical features while suppressing irrelevant or redundant information. Specifically, CBAM first applies the Channel Attention Module (CAM) to globally weight each channel of the feature map. The corresponding formulas are shown in equations (2) and (3).

$$M_c = \sigma[W_1(W_0 F_{avg}) + W_1(W_0 F_{\max})] \tag{2}$$

$$F_1 = M_c \odot F \tag{3}$$

where σ represents the activation function, W_1 is the matrix that restores the channel dimensions, and W_0 denotes the dimensionality reduction matrix. F_{avg} and F_{max} refer to the aggregated information for each channel, obtained through average pooling and max pooling, respectively. M_c represents the attention weight for each channel. F is the input feature map, $\odot$ denotes element-wise multiplication, and F_1 is the weighted feature map after applying attention. Next, the weighted features are fed into the spatial attention module, which assigns weights to each spatial location to measure the importance of different positions within the feature map. The mathematical formulation is shown in equation (4). This process allows the model to more effectively extract key information, thereby enhancing detection accuracy.

$$F_2 = M_s \odot F_1 \tag{4}$$

where, M_s represents the spatial attention map, F_2 denotes the result of applying spatial weighting to the input feature map. In this study, the CBAM is embedded after the fourth layer feature map (P4/16) in the multi-scale feature extraction process. The output feature map from this layer has undergone multiple downsampling operations, containing richer semantic information, and its channel count is 232.

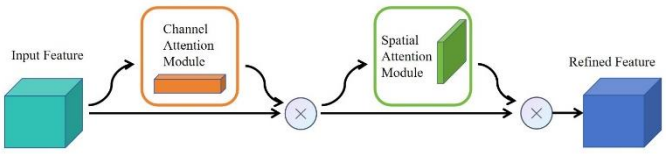


Fig. 3. CBAM structure diagram

2.4 Small target detection head

To enhance the multi-scale feature extraction capability of the model for pavement crack detection, this study optimized the head structure of YOLOv8 (Dai et al., 2021; Zhu et al., 2023). The original YOLOv8 head utilizes a Feature Pyramid Network (FPN) for multi-scale feature fusion, but it exhibits certain limitations when handling complex crack patterns. To better

capture crack features across different scales, the following improvements were made based on the original architecture.

First, by introducing more refined up-sampling and down-sampling operations between the P2, P3, P4, and P5 feature layers, this study achieves multi-level fusion of feature maps across different layers. Specifically, during each down-sampling stage, convolutional layers (Conv) are used to reduce the size of the feature maps, while the feature maps from the previous layer are concatenated (Concat) with those of the current layer, thereby enhancing feature representation. By incorporating the C2f module, the concatenated feature maps undergo further processing to enhance feature extraction capabilities. This module reinforces the model's ability to perceive details of cracks through repeated convolution operations. The improved detection head structure is illustrated in Figure 4.

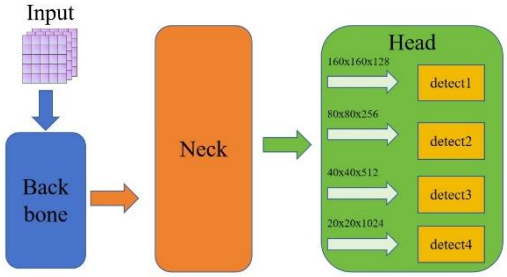


Fig. 4. Improved detection head structure

Ultimately, the improved YOLOv8 head effectively integrates multi-scale information from P2, P3, P4, and P5, enabling the model to perform exceptionally well in detecting cracks of various sizes and shapes. This optimization strategy has led to a significant enhancement in the model's accuracy in crack detection tasks.

3. Experimental Results and Analysis

3.1 Datasets and experimental environment

To validate the effectiveness of the crack detection algorithm proposed in this study, we selected a subset of pavement crack images from the CrackForest and SDNET2018 databases as the experimental dataset. To ensure sample balance, we extracted the training, validation, and test sets in a ratio of 7:2:1. The CrackForest dataset is a widely used standard dataset for pavement crack detection, providing high-quality image data for training and testing. This dataset contains 118 real-world road images, each with a resolution of 480 × 320 pixels, showcasing the distribution of cracks in various actual road scenarios. SDNET2018 is a large standard dataset designed for structural crack detection, created by the Structural Health Monitoring Research Team at the University at Buffalo. It focuses specifically on crack detection tasks related to materials such as concrete and brick walls. This dataset offers a substantial collection of high-resolution images, covering crack scenarios found in real-world construction materials. The hardware environment for this experiment utilizes the Ubuntu operating system based on Linux, paired with an NVIDIA GeForce RTX 4090 graphics card that has 24 GB of VRAM. The software stack comprises PyTorch 2.3.0 and Python 3.11.9. The corresponding parameter settings are as follows: the number of epochs is set to 300, and the batch size is set to 8.

3.2 Evaluation indicators

In object detection tasks, evaluating a model's performance relies not only on its detection accuracy but also on various comprehensive metrics. To thoroughly assess the performance of the improved YOLOv8 model, this study introduces several commonly used performance indicators for quantitative evaluation.

Firstly, precision (P) and recall (R) are fundamental metrics for measuring object detection performance, reflecting the model's accuracy and coverage in detecting targets, respectively. In practical applications, a model not only needs to achieve a high precision to avoid false positives but also requires a high recall to ensure that as many true targets as possible are detected. The calculation formulas for P and R are presented in Equations (5) and (6), respectively.

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

Additionally, the F1-Score (F_1) provides a comprehensive evaluation metric for the model's overall performance by balancing precision and recall. The calculation formula for the F_1 is presented in Equation (7). To further evaluate the detection performance of the model, we employed the widely used Mean Average Precision (mAP). mAP represents the average precision and recall values across multiple IoU thresholds, providing a comprehensive assessment of the model's performance under varying detection difficulties. In this study, mAP@0.5 was utilized as the primary evaluation criterion, with a focus on analyzing the model's detection accuracy under different IoU requirements. The calculation formula for mAP is presented in Equation (8).

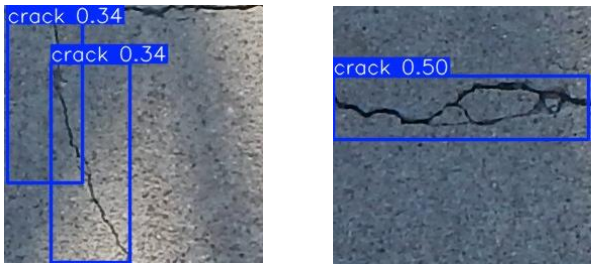
$$F_1 = 2 \times \frac{PR}{P + R} \tag{7}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{8}$$

In practical applications, the detection speed and resource consumption of the model are equally important factors to consider. Therefore, this study also evaluated the model's inference time (T), number of parameters (N), FLOPs (F), and model size (S). These metrics provide insight into the model's efficiency and its suitability for deployment in real-time scenarios. Inference time determines the model's applicability in real-time detection tasks, while the number of parameters, FLOPs, and model size are closely related to the model's complexity and computational overhead. By conducting a comprehensive analysis of these metrics, we can provide a more objective evaluation of the adaptability and practicality of the improved YOLOv8 model in various scenarios.

3.3 Pavement crack detection results

In this study, we applied the improved YOLOv8 model for pavement crack detection tasks and evaluated its performance on the SDNET2018 dataset. As shown in Figure 5, a portion of the crack detection results are presented. In the visualized detection outcomes, the improved YOLOv8 model demonstrates enhanced accuracy in identifying various types of cracks, including longitudinal crack, transverse crack, block crack, and intersecting crack. Notably, in areas with complex backgrounds, the improved model shows a better capability to recognize crack boundaries, effectively reducing the occurrence of false positives and missed detections.



(a) longitudinal crack (b) transverse crack

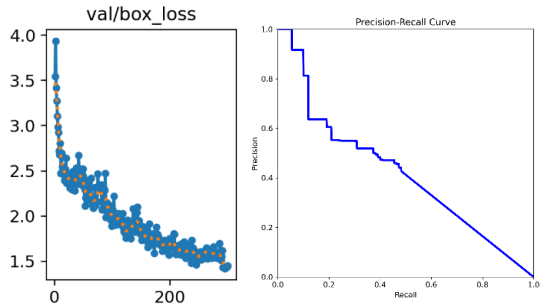
(c) block crack (d) intersecting crack

Fig. 4. Pavement crack detection results

In addition, to quantitatively assess the detection accuracy of the proposed neural network, this study introduces *P*, *R*, F_1, and *mAP* for validation. Table 1 summarizes the scores for various evaluation metrics of the proposed model. This study deals with a highly challenging dataset, which includes numerous small targets and complex backgrounds. Despite the difficulty of the task, the proposed model achieved an mAP@0.5 of 45%, a *P* of 43%, a *R* of 48%, and an F_1 of 47% on the validation set, demonstrating its effectiveness in handling complex detection tasks.

Tab. 1. Improve model scoring.

Model	P/%	R/%	F_1/%	mAP@0.5/%
Proposed	43	48	47	45



(a) Loss function curve (b) Precision-Recall curve

Fig. 4. Model training curve

Figure 6 presents the model's loss function curve and the Precision-Recall curve. As shown in Figure 6(a), the loss gradually decreases during the training process, indicating that the model is converging. Meanwhile, Figure 6(b) illustrates the Precision-Recall curve, which reflects the trade-off between precision and recall at different confidence thresholds. As recall increases, precision gradually decreases, demonstrating the inverse relationship between these two metrics. In the high-recall region, the precision of the model decreases, indicating an increase in false positives as more targets are detected. However, in the lower-recall region, the model maintains a higher precision, suggesting that the detection results are more accurate when the model operates at higher confidence levels. This trade-off reflects the model's ability to balance between detecting more instances and minimizing false detections.

3.4 Comparison of model training results

To validate the performance improvement of the lightweight YOLOv8 model in crack detection tasks, this section conducts a series of comparative experiments. The primary objective of these experiments is to assess how the model achieves significant optimization in terms of the number of parameters, inference speed, computational complexity, and model size while maintaining detection accuracy. By comparing with the original YOLOv8 and other mainstream detection models, we can comprehensively assess the practicality of the improved model (Jocher et al., 2022; Wang et al., 2024).

Table 2 presents a comparison of various parameters between the proposed improved YOLOv8 model and other detection models. It can be observed that the parameter count of the proposed model is 1.7, while the parameter counts of the compared mainstream models, YOLOv5, YOLOv8, and YOLOv10, are 7.0, 3.0, and 2.7, respectively. The parameter count of the proposed model is significantly lower than that of these models. Thanks to the reduced parameter count, the proposed model demonstrates a faster processing speed during inference, achieving 0.4 ms. In contrast, other models exhibit slower inference speeds, with the slowest being 1.3 ms. This advantage makes the proposed model more suitable for resource-constrained environments.

The proposed improved model in this study has undergone a lightweight optimization in terms of parameter and architecture design, resulting in a model size of 3.8 MB. In comparison, other deep learning models range from a maximum size of 14.4 MB to a minimum size of 5.8 MB, representing reductions of 74% and 34%, respectively. This decrease in model size not only enhances deployment efficiency but also alleviates storage pressure on edge devices or embedded systems. Furthermore, this model has a FLOPs value of 9.1 G, which represents a 30% reduction in computational complexity compared to YOLOv5, which has a FLOPs of 15.8 G. In summary, the proposed model demonstrates significant advantages in terms of speed, computational resources, model parameter count, and model size. The model is not only suitable for large-scale pavement crack monitoring tasks but also exhibits outstanding potential in real-time performance, lightweight deployment, and adaptability across multiple scenarios, providing a reliable, fast, and efficient solution for crack detection in practical applications.

Tab. 2. Comparison of model parameters

Model	N/M	S/MB	T/ms	F/G
YOLOv5	7.0	14.4	1.3	15.8
YOLOv8	3.0	6.2	0.9	8.1
YOLOv10	2.7	5.8	0.3	8.2
Proposed	1.7	3.8	0.4	9.1

4. Summary

This paper addresses the need for real-time detection of pavement cracks on highways by proposing a detection method based on an improved YOLOv8 model, effectively overcoming the limitations of existing technologies concerning model size and detection speed. By incorporating a lightweight network to replace the C2f module in YOLOv8, the inference speed is significantly enhanced, enabling

real-time detection capabilities. Furthermore, the integration of the Convolutional Block Attention Module improves the model's ability to recognize fine cracks in complex scenarios. Optimizations to the YOLOv8 head structure further enhance the model's adaptability and robustness under challenging pavement conditions. Experimental results demonstrate that the improved model achieves comprehensive advancements in speed, parameter count, and model size while maintaining detection accuracy, thereby validating its potential and practicality for automatic pavement crack detection. This research provides a new technological pathway for the future of intelligent road maintenance.

Acknowledgements

The authors would like to acknowledge financial support provided by the National Natural Science Foundation of China (62001263), the Key research projects of Qingdao Science and Technology Plan (22-3-3-hygg-30-hy), and the Natural Science Foundation of Shandong Province (ZR2021MF024).

References

Kheradmandi, N., Mehranfar, V., 2023, A critical review and comparative study on image segmentation-based techniques for pavement crack detection. J. Construction and Building Materials. 321: 126162.

Yang, Z., Ni, C., Li, L., et al., 2022, Three-stage pavement crack localization and segmentation algorithm based on digital image processing and deep learning techniques. J. Sensors. 22(21): 8459.

Chen, C., Seo, H., Jun, C H., et al., 2022 A potential crack region method to detect crack using image processing of multiple thresholding. J. Signal, Image and Video Processing. 16(6): 1673-1681.

Dai, C., Jiang, K., Wang, Q,. 2020, Recognition of tunnel lining cracks based on digital image processing. J. Mathematical Problems in Engineering. 2020(1): 5162583.

Dhillon, A., Verma, G.K., 2020, Convolutional neural network: a review of models, methodologies and applications to object detection. J. Progress in Artificial Intelligence. 9(2): 85-112.

Sun, X., Wang, P., Wang, C., et al., 2021, PBNet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery. J. ISPRS Journal of Photogrammetry and Remote Sensing. 173: 50-65.

Jiang, P., Ergu, D., Liu, F., et al., 2022, A Review of Yolo algorithm developments. J. Procedia computer science. 199: 1066-1073.

Diwan, T., Anirudh, G., Tembhurne, J.V., 2023, Object detection using YOLO: Challenges, architectural successors, datasets and applications. J. multimedia Tools and Applications. 82(6): 9243-9275.

Wang, S., Chen, X., Dong, Q., 2023, Detection of asphalt pavement cracks based on vision transformer improved YOLO V5. J. Journal of Transportation Engineering, Part B: Pavements. 149(2): 04023004.

Xing, J., Liu, Y., Zhang, G.Z., 2023, Improved yolov5-based uav pavement crack detection. J. IEEE Sensors Journa. 23(14): 15901-15909.

Zhou, S., Yang, D., Zhang, Z., et al., 2025, Enhancing autonomous pavement crack detection: Optimizing YOLOv5s algorithm with advanced deep learning techniques. J. Measurement. 240: 115603.

Hu, H., Li, Z., He, Z., et al., 2024, Road surface crack detection method based on improved YOLOv5 and vehicle-mounted images. J. Measurement. 229: 114443.

Ma, T., Han, C., Huyan, J., et al., 2023, End-to-end task integrated pavement crack image detection and segmentation based on the improved YOLO network. M. Advances in Functional Pavements. CRC Press, 2023: 134-138.

Fu, H., Song, G., Wang, Y., 2021, Improved YOLOv4 marine target detection combined with CBAM. J. Symmetry. 13(4): 623.

Sohan, M., Sai, Ram.T., 2024, Reddy R, et al. A review on yolov8 and its advancements. C. International Conference on Data Intelligence and Cognitive Informatics. Springer, Singapore. 2024: 529-545.

Hao, L.I., Weigen, Q.I.U., Lichen, Z., 2022, Improved ShuffleNet V2 for Lightweight Crop Disease Identification. J. Journal of Computer Engineering & Applications. 58(12).

Jocher, G., Chaurasia, A., Stoken, A., et al., 2022, ultralytics/yolov5: v6. 2-yolov5 classification models, apple m1, reproducibility, clearml and deci. ai integrations. J. Zenodo.

Wang, A., Chen, H., Liu, L., et al., 2024, Yolov10: Real-time end-to-end object detection. J. arxiv preprint arxiv:2405.14458, 2024.

Dai, X., Chen, Y., Xio, B., et al., 2021 Dynamic head: Unifying object detection heads with attentions. C. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 7373-7382.

Zhu, G., Zhu, F., Wang, Z., et al., 2023, Small target detection algorithm based on multi-target detection head and attention mechanism. C. 2023 IEEE 3rd International Conference on Digital Twins and Parallel Intelligence (DTPI). IEEE. 2023: 1-6.

Zeqi Liu is currently pursuing a master's degree in Information and Control Engineering at Qingdao University of Technology. He obtained a bachelor's degree from Qingdao University of Technology in 2022. His main research areas are computer vision, image processing, and deep learning.

Hui Yao is currently studying for a master's degree in information and control engineering at Qingdao University of Technology. Bachelor degree from Qingdao University of Technology in 2024. The main research fields are image recognition, image mosaic, feature detection, borehole panoramic image recognition and so on.

Xiaoyue Zhong is currently studying information and control engineering at Qingdao University of Technology. Her main area of research is robotics and computer vision.

Zhaopeng Deng received his Ph.D from Shandong University of Science and Technology, China in 2019. His major research interests include image processing and machine vision. Currently he is a teacher in Qingdao University of Technology. His major research interests include intelligent control, image processing, machine vision and pattern recognition. He has published over 6 research papers in international journals, and conferences.