RESEARCH ARTICLE

A Novel YOLO Algorithm Integrating Attention Mechanisms and Fuzzy Information for Pavement Crack Detection

Qingqing Li[1,2,3,4] · Tianshu Wu[1,4] · Tingfa Xu[2] · Jianmei Lei[1,4] · Jiu Liu[5]

Received: 18 March 2025 / Revised: 23 May 2025 / Accepted: 10 June 2025

Abstract
Pavement crack detection is widely spread over road maintenance, ensuring the longevity and safety of infrastructure. Traditional manual inspection methods are time-consuming, labor-intensive, and prone to errors. In response, automated crack detection systems based on deep learning have emerged, offering more efficient and accurate solutions. However, existing models often face challenges such as large model sizes, slow inference speeds, and limited applicability in real-time applications. In this paper, we propose a novel light-weight Crack Regional Segmentation method based on YOLOv11, which introduces attention mechanisms to address challenges in pavement images, such as varying crack sizes, occlusion, and irregular surface textures. By embedding a region-based attention mechanism into the YOLOv11 network, the method enhances the model's ability to focus on crack features. Specifically, the model network layers are progressively pruned to reduce the number of parameters and floating-point operations, thereby further improving operational efficiency and refining detection in the target regions. Furthermore, to tackle issues with blurred or indistinct crack boundaries, we present a fuzzy information-guided YOLOv11-based model, FIG-YOLO. This model integrates fuzzy logic and fuzzy membership functions to handle uncertainty in crack detection. The fuzzy membership functions are used to quantify the degree of crack features, allowing the model to better distinguish between crack and non-crack regions, especially in cases where crack boundaries are unclear. This approach significantly improves the accuracy of crack detection and segmentation. Extensive experiments demonstrate that our approach effectively addresses challenges such as complex backgrounds and blurred crack edges in pavement images. This research not only provides a novel solution for the automated detection of pavement cracks but also offers insights into the development of intelligent road maintenance systems. With the expansion of large-scale datasets and the advancement of deep learning models, pavement crack detection algorithms are expected to further enhance their accuracy and efficiency, offering significant support for road infrastructure management.

Keywords Pavement crack detection · Defect detection · Loss function

1 Introduction

The field of pavement crack detection has undergone significant technological evolution across three distinct generations. Early methodologies [1–5] relied on traditional computer vision techniques, such as edge detection using Canny and Sobel operators, and texture analysis via Gabor filters and wavelet transforms. While these approaches offered computational efficiency, they faced challenges in handling lighting variations and complex backgrounds [6–9]. Subsequently, the second generation introduced machine learning frameworks [10–12], which

Published online: 25 June 2025

integrated handcrafted features with classifiers like support vector machines (SVMs) and random forests. These methods demonstrated improved robustness compared to their predecessors but remained constrained by the limited capacity of manually designed feature representations. More recently, advancements have centered on deep learning architectures, particularly convolutional neural networks (CNNs) [10–12]. Notable progress in balancing speed and accuracy has been achieved with YOLO-series models [13], which have become widely adopted for real-time pavement crack detection due to their efficiency and performance.

Nevertheless, three persistent limitations continue to challenge current detection systems. Within automated road defect detection systems [10–12], the capturing and processing of high-resolution pavement images form the cornerstone for enabling both swift and precise evaluations. However, road images often present several challenges, such as variations in lighting, occlusion by objects such as vehicles or debris, and irregular surface textures. These issues complicate image preprocessing and defect analysis, increasing the risk of misclassification or missed detections, which can undermine the accuracy and reliability of defect detection systems. These limitations collectively result in unacceptable false negative rates for mission-critical infrastructure applications.

To address these challenges, this paper proposes a lightweight road defect detection method based on the YOLOv11 [13] model. By incorporating an attention mechanism, the model is able to focus on critical defect regions while minimizing interference from irrelevant background areas. Additionally, we introduce a gradual pruning method that continuously monitors the model's performance as parameters and floating-point operations are reduced. This process helps in obtaining a lightweight model without compromising detection accuracy. Experimental results demonstrate that our approach significantly improves the accuracy of road defect detection, providing a reliable foundation for the automation of road maintenance monitoring.

In real-world applications, road defects often exhibit irregular shapes and varying sizes, and their boundaries are not always well-defined [14, 15]. This can make the detection process particularly challenging. To improve the accuracy of defect segmentation, we propose a Fuzzy Information-Guided YOLOv11-based model, FIG-YOLO, which incorporates fuzzy logic to handle the uncertainty and imprecision inherent in road defect images. Specifically, fuzzy logic [16–18] allows the model to consider the degree of pixel membership in defect regions, rather than classifying pixels strictly as defective or non-defective. This approach enables the model to more effectively manage blurry or indistinct defect boundaries, leading to improved segmentation accuracy. Additionally, the model uses a fuzzy Jaccard Index loss function to optimize the similarity between predicted and ground truth segmentations.

Our main contributions are as follows:

1. We propose a novel Crack Regional Segmentation (CRS) method, which introduces an attention mechanism to focus specifically on the crack region. This approach effectively reduces interference from irrelevant background areas, improving the accuracy of segmentation in challenging dry-eye images.
2. We propose Fuzzy Information-Guided YOLOv11-based model, FIG-YOLO, to address the challenges posed by blurry and indistinct pavement crack boundaries. By incorporating fuzzy logic, this model can better handle the uncertainty and overlap in pavement crack areas, leading to more accurate detection and segmentation.
3. We propose a novel gradual pruning method that utilizes a lightweight YOLOv11-based model, which significantly reduces the number of floating-point operations and parameters, thereby achieving a balance between speed and accuracy to enable real-time detection.

2 Related Work

2.1 Pavement Crack Detection

Automated pavement crack detection systems have gained popularity in recent years. These systems typically involve the use of image capture devices to obtain road surface images, which are then analyzed using deep

learning models to detect the presence of cracks [10, 19, 20]. Deep learning-based approaches, on the other hand, offer a more automated and accurate solution by leveraging convolutional neural networks (CNNs) [1–5] and other advanced architectures to learn and identify cracks directly from raw image data [21–25]. This significantly reduces the need for manual intervention, enhances detection accuracy, and allows for real-time analysis, making deep learning a promising direction for pavement crack detection. However, these methods may still suffer from high computational costs and are often sensitive to variations in image quality.

2.2 Computer Aided Diagnosis

Computer-Aided Diagnosis (CAD) has become an essential tool in medical and engineering fields for improving diagnostic accuracy and efficiency [26–30]. In the context of pavement crack detection, CAD systems [31–33] have been increasingly adopted to automate the process of identifying and classifying road defects. These systems leverage machine learning and deep learning techniques to analyze road surface images [34–37], offering several advantages over traditional manual inspection methods, such as faster processing speeds, reduced human error, and enhanced accuracy. In pavement crack detection, CAD systems based on deep learning models, such as CNNs and other advanced architectures like YOLO, have been developed to detect cracks and defects in road surfaces. These models are trained on annotated datasets of road images, learning to distinguish between normal pavement conditions and defects such as cracks, potholes, or surface degradation. Recent advancements have focused on optimizing the architecture of these models to handle challenges such as variations in lighting, crack size, and background noise, ensuring that CAD systems can operate effectively under real-world conditions. The adoption of deep learning-based CAD systems in pavement crack detection has led to significant improvements in detection speed and accuracy. Moreover, CAD systems can be integrated into real-time monitoring frameworks to enable continuous surveillance of road conditions, eliminating the need for frequent manual inspections. By automating the detection workflow, these systems not only alleviate the workload for maintenance teams but also deliver a reliable, scalable solution for large-scale road network monitoring. With ongoing advancements in deep learning models and the availability of larger, more diverse datasets, the potential of CAD systems for automated pavement crack detection is poised to expand, ultimately ushering in smarter, more efficient infrastructure management practices.

2.3 Learning-Based Semantic Segmentation

Semantic segmentation is a fundamental task in computer vision, aiming to assign a label to every pixel in an image, allowing for the identification and localization of objects or regions of interest. In recent years, learning-based approaches, particularly those using deep neural networks, have become the dominant method for semantic segmentation due to their ability to learn hierarchical features and handle complex visual patterns. In the context of pavement crack detection, semantic segmentation [25] plays a crucial role by precisely identifying and delineating crack regions from the surrounding road surface. Traditional image processing techniques, such as edge detection and thresholding, often struggle to accurately segment cracks, especially in challenging conditions like varying lighting, complex backgrounds, or small-scale defects. However, deep learning-based semantic segmentation models, particularly Convolutional Neural Networks (CNNs) and their more advanced variants, have demonstrated remarkable success in handling these challenges. One of the most widely used architectures for semantic segmentation is the Fully Convolutional Network (FCN) [38], which replaces fully connected layers with convolutional layers to produce pixel-wise predictions. This architecture has been shown to outperform traditional methods in a variety of applications, from medical imaging to autonomous driving. The U-Net model [39], another notable architecture, has been extensively used in segmentation tasks, especially when high accuracy is required in segmenting small and fine structures. U-Net's encoder-decoder architecture, with its skip connections, enables the model to capture both high-level semantic features and fine-grained details, making it particularly suitable for tasks like road crack segmentation, where both large and small defects need to be identified. More recently, models like

DeepLab [40], SegNet [41], and PSPNet [42] (Pyramid Scene Parsing Network) have pushed the boundaries of semantic segmentation by incorporating advanced techniques such as atrous convolutions, pyramid pooling, and multi-scale feature extraction. These techniques help improve the model's ability to segment objects at different scales and in varying contextual settings, making them ideal for pavement crack detection, where cracks may vary in size and appearance. In pavement crack detection, learning-based semantic segmentation models have been employed to automatically segment crack regions from road surface images. These models, when trained on large annotated datasets, can accurately identify cracks and other road defects, even in the presence of noise, low resolution, or occlusion. The segmentation results can then be used to further analyze the severity and distribution of cracks, facilitating better decision-making for road maintenance. The key advantage of learning-based semantic segmentation is its ability to generalize well to unseen data. Once trained, these models can be applied to a wide range of road images captured under different conditions, providing a robust solution for large-scale road monitoring. As datasets grow and deep learning techniques continue to evolve, learning-based semantic segmentation is expected to remain a cornerstone of automated pavement crack detection, offering scalable, accurate, and efficient solutions for intelligent infrastructure management.

3 Method

3.1 Crack Region Segmentation

In pavement crack detection images, the primary region for identifying cracks is often the specific areas where the cracks are visible and have distinct boundaries. Other irrelevant areas in the image, such as background surfaces or objects, occupy a large proportion, leading to increased resource consumption for processing unnecessary information and potentially causing missed or false detections. To address this issue, we propose a **Crack Region Segmentation (CRS)** approach, which first performs image segmentation on the regions where cracks are likely to appear. Building on the YOLOv11 model, we enhance it by incorporating the CBAM (Convolutional Block Attention Module) attention mechanism [43], which improves the model's focus on crack regions in the image. This modified model enables the extraction of crack regions from pavement images while retaining crucial surface information, thereby reducing interference from unrelated areas. This allows for more accurate subsequent tasks such as crack contour detection, severity analysis, and other evaluations necessary for road maintenance. This method facilitates focused analysis and evaluation of crack regions, allowing for efficient crack extraction in various types of road images and demonstrating strong performance across different scenarios.

3.2 Architecture of CRS

Our CRS architecture, as shown in Fig. 1, is built upon the YOLOv11 [13] model. We selected YOLOv11 as our baseline model due to its optimal balance between real-time processing capability and detection accuracy among current object detection frameworks. The architecture offers multiple scalable variants with varying parameter counts and computational loads (FLOPs), enabling tailored solutions for different application requirements. Moreover, its mature ecosystem supports seamless conversion to various deployment formats (e.g., ONNX) through the Ultralytics library [13], significantly facilitating industrial implementation. To enhance feature extraction, we integrates the Convolutional Block Attention Module (CBAM) to perform feature extraction at both the spatial and channel levels. The spatial attention mechanism allows the model to prioritize and focus on the most relevant information in the feature map, while the channel attention mechanism aggregates spatial features, enhancing the model's ability to capture multi-dimensional information. By incorporating CBAM, the YOLOv11 model's performance is significantly improved. Furthermore, we employ a layer-by-layer pruning strategy and reduce the number of convolutional layers to optimize the model's efficiency. This approach not only streamlines the network architecture but also reduces computational overhead, making the model more lightweight and suitable

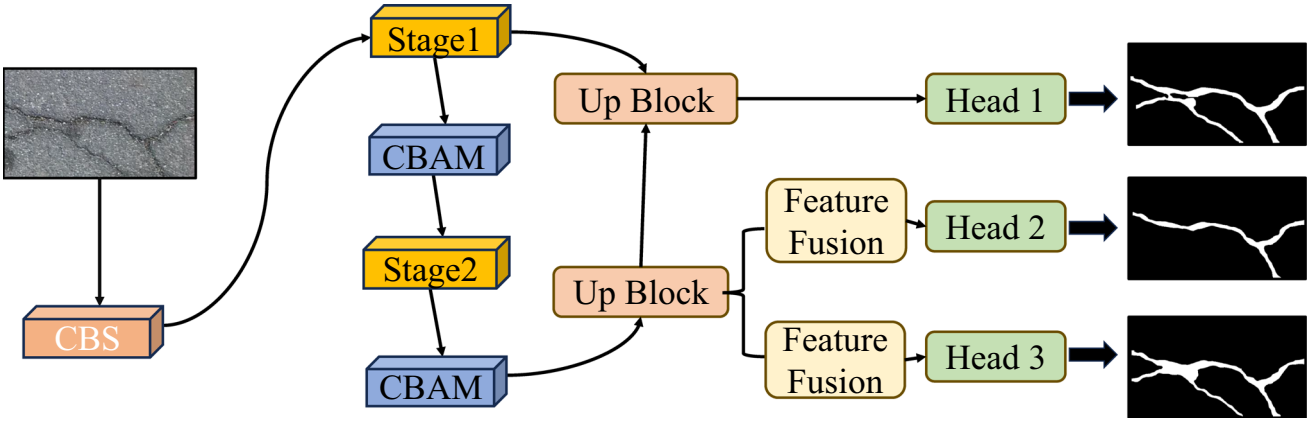

Fig. 1 Stage 1 begins with an Up Block for upsampling, followed by Head 1 for preliminary feature extraction. A CBAM is employed to enhance feature representation, and Feature Fusion integrates multi-scale information, with Head 2 generating intermediate outputs. Stage 2 follows a similar workflow but utilizes CBAM to deepen feature refinement. The final results are produced by Head 3, further optimized by CBAM for attention-aware feature enhancement. The hierarchical design emphasizes multi-stage feature strengthening, cross-scale fusion, and the critical role of attention mechanisms

for real-time applications. Despite the reduction in complexity, the model maintains high accuracy in detecting and segmenting pavement cracks, even in the presence of varying levels of noise and background interference. This combination of attention mechanisms and network optimization enables the model to effectively handle real-world pavement crack images, addressing significant challenges posed by complex backgrounds and varying image qualities.

3.3 Details of CRS

In our proposed architecture, we employ the Stage, drawn in Fig. 2, module and the CBAM (Convolutional Block Attention Module) twice in sequence to comprehensively extract both deep and shallow semantic features from pavement crack images. This dual application ensures that the model captures a rich hierarchy of features, ranging from fine-grained details to broader contextual information. Following the feature extraction process, we utilize the Up Block module to perform upsampling and feature fusion on the extracted deep and shallow semantic features. This step is crucial for integrating the detailed information from the deep features with the broader context provided by the shallow features, thereby enhancing the overall representational power of the feature maps. And then, one set of the extracted features is routed directly to Head 1, structure of the head is depicted in Fig. 3 , which is

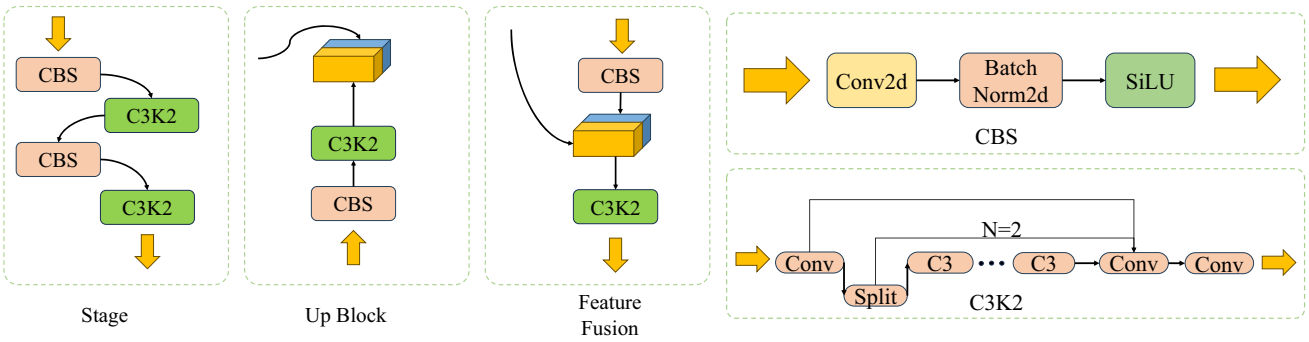

Fig. 2 The internal structure of the Stage, Up Block, and feature fusion modules. The Stage module consists of two standard convolutional blocks (CBS) and two C3K2 modules. The Up Block includes one standard convolutional block, one C3K2 module, and a tensor concatenation operation. The feature fusion module has the same components as the Up Block but with a different order of operations

 https://doi.org/10.1007/s44196-025-00894-5

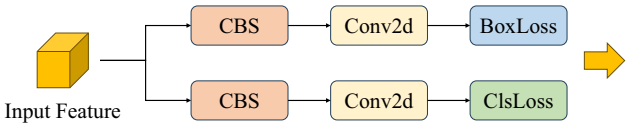

Fig. 3 Structure diagram of the object detection decoder, object detection head

responsible for the initial decoding process. Meanwhile, the other set of upsampled features is passed through two feature fusion modules. These modules further refine the feature maps by integrating complementary information, ensuring that the features are robust and informative for subsequent tasks. The outputs from the feature fusion modules are fed into two distinct heads (Head 2 and Head 3) for separate decoding processes. This parallel decoding strategy allows the model to generate multiple, specialized outputs that can address different aspects of the tear film segmentation task more effectively. By leveraging this multi-headed approach, we enhance the model's ability to capture diverse patterns and improve the accuracy and reliability of the segmentation results.

3.4 Core Improvement Module

Convolutional Block Attention Module, CBAM, is an attention module used in deep learning models, designed to enhance the network's focus on important features while suppressing the influence of irrelevant features. It is a compact and efficient attention mechanism commonly integrated into various layers of convolutional neural networks. It employs two sub-modules—spatial attention and channel attention—to optimize features along both dimensions. CBAM can be primarily divided into two modules: channel attention and spatial attention. Channel attention focuses on the feature channels that the model should prioritize. It extracts spatial information using global average pooling and global max pooling, and then learns the relationships between channels through shared network weights to generate a channel attention map, As shown in Fig. 4.

Channel Attention, Displayed in Fig. 5, focuses on the feature channels that the model should prioritize. It extracts spatial information using global average pooling and global max pooling, and then learns the relationships between channels through shared network weights to generate a channel attention map. Its formula is as follows:

$$\mathcal{M}_{\rfloor}(\mathcal{F}) = \sigma\left(MLP\left(\mathcal{A}(\mathcal{F})\right) + MLP\left(\mathcal{M}(\mathcal{F})\right)\right) \tag{1}$$

where $\mathcal{F}$ represents the input feature map. $\mathcal{A}(\mathcal{F})$ and $\mathcal{M}(\mathcal{F})$ denote average pooling and max pooling operations, respectively, applied to the feature map to extract global spatial information. All pooling operations use a 2×2 kernel size. MLP refers to the multilayer perceptron, which receives the pooled features as input, with each MLP using 1×1 kernels. Finally, the outputs of the two MLPs are weighted and combined using the Sigmoid function to generate the channel attention map $\mathcal{M}_{\rfloor}(\mathcal{F})$.

Spatial attention, as depicted in Fig. 6, focuses on identifying the most important locations within the feature map. It is achieved by compressing the channels, typically using the mean and maximum values across the feature map. This is followed by a convolutional layer that generates a spatial attention map. And its formula is as follows:

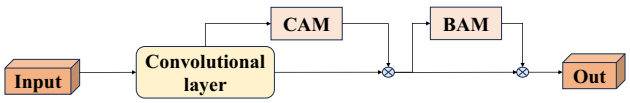

Fig. 4 The diagram of CBAM can be divided into two modules: channel attention and spatial attention

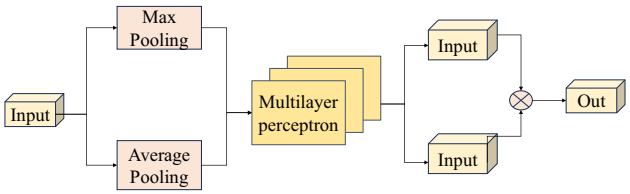

Fig. 5 Diagram of the channel attention mechanism

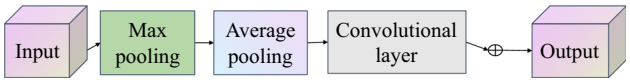

Fig. 6 An illustration of the mechanism of spatial attention

$$\mathcal{M}_f(\mathcal{F}) = \sigma\left(f^{7\times7}\left(\text{Concat}\left(\mathcal{A}(\mathcal{F}), \mathcal{M}(\mathcal{F})\right)\right)\right) \tag{2}$$

where $f^{7\times7}$ represents the convolution operation using a 7×7 kernel; σ is the sigmoid activation function; Concat denotes the concatenation of feature maps; $\mathcal{M}_f(\mathcal{F})$ is the learned spatial attention map. Average pooling and max pooling are commonly used pooling operations in convolutional neural networks, employed to reduce the dimensionality of feature maps while preserving important information. The formula for average pooling is shown as follows:

$$f_{\text{avg}}(\mathcal{X}) = \frac{1}{m \times n}\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \tag{3}$$

In the formula, $\mathcal{X}$ is the input feature subset within the pooling window; x_{ij} is an element in this subset; $f_{\text{avg}}(\mathcal{X})$ is the output value after the pooling operation. The formula for max pooling is shown as follows:

$$f_{\max}(X) = \max_{1\le i\le m, 1\le j\le n} x_{ij} \tag{4}$$

where $\mathcal{X}$ is the input feature subset within the pooling window; x_{ij} is an element in this subset; $f_{\max}(\mathcal{X})$ is the output value after the pooling operation.

The attention mechanism is a common application in object detection and image segmentation tasks. In object detection, the model must determine the position and category of objects within an image. By using a region-based attention mechanism, the model can focus on areas that are likely to contain the object, thereby improving both the accuracy and efficiency of the detection. In image segmentation, the region attention mechanism helps the model better understand the semantic information of different regions within the image, leading to more precise segmentation results. A key operation in the attention mechanism is calculating the attention weights for each image region. These weights are typically computed based on the features of each region to determine its importance for the current task. Common methods for computing these weights include using convolutional neural networks (CNNs) or fully connected layers to learn the attention weights or calculating the similarity between region features to derive the weights. Once the attention weights are obtained, they are applied to the feature representations in the image to produce a weighted aggregate representation. This allows the model to selectively focus on the regions that are most important for the task at hand, integrating the information from these regions into the overall feature representation.

In image segmentation, it is crucial to accurately segment only the areas of interest. The attention mechanism helps the model focus on these regions, improving its ability to perform accurate segmentation. Moreover, complex backgrounds and noise in the image can interfere with object segmentation. The attention mechanism enables the model to focus more on the target regions during segmentation, reducing the impact of background noise and enhancing segmentation accuracy. By incorporating the attention mechanism, the image segmentation model can dynamically adjust its focus on the image, effectively handling objects of different scales, shapes, and complexities, thereby improving segmentation accuracy and robustness. The model employed in this study is improved with a region-based attention mechanism, enabling targeted segmentation and extraction of the crack region.

3.5 Central Loss Function

The Attention Center Loss is a loss function used to measure the performance of the attention mechanism [43]. The definition of the Attention Center Loss function typically involves attention weights and target values. Attention weights are obtained by calculating the model's attention distribution over the input data, reflecting the model's focus on different parts of the input data during processing. The target value, on the other hand, represents the true attention distribution, reflecting how the model should ideally allocate its attention. The objective of the Attention Center Loss function is to minimize the difference between the model's attention weights and the target values. By optimizing the Attention Center Loss, the model's attention distribution can be made to more closely align with the target values, thereby improving the model's performance. The formula for the Attention Center Loss function is shown as follows:

$$L_A = \sum_{k=1}^{M} ||f_k - C_k||^2 \tag{5}$$

In the formula, $||\mathcal{V}||^2$ represents the Euclidean norm of the vector, also known as the L2 norm, and is defined as follows:

$$||v||_2 = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2} \tag{6}$$

The Attention Center Loss not only improves the model's performance in the attention mechanism but also enhances the model's ability to capture the semantic information within the image. This, in turn, increases the model's robustness against complex backgrounds and interference, to some extent.

3.6 Fuzzy Information-Guided YOLO Model

Pavement crack detection is crucial for assessing the condition of road surfaces. The detection of the first visible crack or defect after road exposure to stress is a key indicator of road degradation, and a rapid spread of cracks is a sign of advanced damage. With the development of automated detection systems, pavement crack detection has become a focus of research in infrastructure maintenance. Due to the complex nature of cracks, their boundaries are often blurred, with no clear edges in certain cases. These fuzzy boundaries make it challenging to accurately determine the exact location and shape of cracks. The aforementioned CRS method effectively guides the model to focus more intently on target segmentation areas, specifically crack pixels, thereby enhancing segmentation accuracy. Nevertheless, for naturally occurring cracks and those captured under uncontrolled imaging conditions, the distinction between crack pixels and intact pavement pixels often becomes ambiguous, resulting in boundary uncertainty issues. To address these challenges, this study employs the Fuzzy Information-Guided YOLOv11 model, FIG-YOLO, for pavement crack detection. Experimental results validate the feasibility and reliability of this method for accurately identifying and segmenting crack regions in pavement images, even under challenging conditions with blurred or indistinct crack edges.

3.7 Definition of Tear Film Fuzzy Boundary

Pavement crack detection faces several challenges, particularly when it comes to accurately identifying crack areas. These challenges manifest in three key aspects. First, the crack regions often exhibit irregular shapes and distortion, which are influenced by the underlying road surface and the crack's progression. Second, the similarity between the crack areas and their surrounding surface can be quite high, making it difficult to distinguish cracks from non-defective areas. Third, the boundaries of the cracks are often poorly defined, especially in cases where cracks are partially filled with debris or are small in scale. These issues complicate the process of accurately

detecting and segmenting cracks, and the uncertainty of these boundaries can prevent precise localization during detection, ultimately affecting the accuracy of the results.

In traditional detection models, such as those based on YOLOv11, lower-resolution features tend to be dominated by higher-resolution ones, leading to poor recognition of the blurred edges in crack images. This results in inadequately extracted crack boundaries, which are crucial for accurate detection and analysis. To address these issues, it is essential to integrate targeted models and algorithms. By combining YOLOv11's powerful feature extraction capabilities with fuzzy logic's ability to handle uncertainty, we can significantly improve the accuracy of pavement crack detection, offering a robust solution for road maintenance applications.

To this end, we propose a novel fuzzy logic method to handle the uncertainty and blurred boundaries of pavement cracks. With high detection accuracy for crack regions, the method effectively addresses the ambiguity in crack boundaries. The first step involves applying a fuzzification operation to the feature map, mapping it to the fuzzy domain to capture the uncertainty of the crack region. In the model, uncertain features are down-weighted, allowing the model to focus on more certain features, thus improving the accuracy of crack detection in the presence of fuzzy or unclear boundaries.

Next, fuzzy information is integrated into the YOLOv11 model to enhance its ability to process uncertain, fuzzy information. By introducing a fuzzy logic layer to YOLOv11, the model can efficiently handle the uncertainty present in the pavement crack images. Using a fuzzy Jaccard Index loss function, the model can precisely assess the differences between predicted and actual crack locations, leading to more accurate detection of crack regions and improving overall pavement maintenance diagnostics.

3.8 Fuzzy Information

Fuzzy logic has been successfully utilized to handle uncertainty in image processing. Fuzzy mean clustering has been successfully applied to image segmentation [59], and fuzzy clustering methods have outperformed non-fuzzy methods in terms of performance. Fuzzy mathematics is a mathematical approach to handling fuzziness and uncertainty. By introducing concepts and tools such as fuzzy sets, it allows for better description and handling of fuzzy and uncertain situations in the real world. Fuzzy mathematics can process fuzzy information, and through abstraction, generalization, synthesis, and reasoning, it can derive conclusions with a certain degree of precision. The key to applying fuzzy mathematics in model construction is to establish membership functions that align with real-world scenarios.

Membership is an important concept in fuzzy mathematics, used to describe the degree of membership or affiliation of an element within a fuzzy set. In traditional set theory, an element either belongs to a set or does not, with no intermediate state. In other words, it is certain whether an element belongs to a set, as represented in formulas:

$$\chi_A{:}U \rightarrow \{0, 1\} \tag{7}$$

$$x \rightarrow \chi_A(x) = \begin{cases} 1, x \in A \\ 0, x \notin A \end{cases} \tag{8}$$

In a fuzzy set, an element can belong to a set with a certain degree of membership, where the membership value can be any real number between 0 and 1. In fuzzy mathematics, the membership value ranges from [0, 1], where 0 indicates that the element does not belong to the fuzzy set, and 1 indicates that it completely belongs to the fuzzy set. For a specific element, the closer its membership value is to 1, the more completely it belongs to the fuzzy set; the closer its membership value is to 0, the less it belongs to the fuzzy set. Membership values between 0 and 1 represent varying degrees of membership in the fuzzy set. The formula is as follows:

$$\mu_A{:}U \rightarrow [0, 1] \tag{9}$$

$$x \rightarrow \mu_A(x) \in [0, 1] \tag{10}$$

Membership can be used to describe the degree of membership of elements within a fuzzy set, as well as the degree of correlation between elements in a fuzzy relation. In pavement crack detection, we consider a fuzzy set "pavement crack area" to describe the state of the road surface. For pavement images, a pixel can be described by its membership degree to indicate its affiliation with the "pavement crack area" fuzzy set. If the certainty that the pixel belongs to the crack region is high, its membership degree to the "pavement crack area" fuzzy set will be close to 1. Conversely, if the certainty that the pixel does not belong to the crack region is high, its membership degree to the "pavement crack area" fuzzy set will be close to 0.

3.9 FIG to Detection of Pavement Crack Detection

We propose a fuzzy information-guided pavement crack area detection model. The core of the model is a YOLOv11-based neural network integrated with fuzzy logic, and the loss function is the fuzzy Jaccard Index loss function. The fuzzy logic layer is used to extract uncertain information from the feature map, allowing the model to focus on extracting uncertain information in the image. During the network propagation process, uncertain feature information is subdued, while more certain feature information is emphasized. The fuzzy Jaccard Index loss function is primarily used to handle the uncertainty and fuzziness between the predicted results and the ground truth, addressing the uncertainty in the boundaries of the pavement crack area.

3.10 Architecture of FIG-YOLO

We uses YOLOv11 as the base model and introduces a fuzzy logic layer (Fuzzy Logic Block) and the fuzzy Jaccard Index loss function to enhance the model's understanding of fuzzy and uncertain features in the image, thereby improving the accuracy of object detection and semantic segmentation. The fuzzy logic layer follows the feature extraction and fusion modules, utilizing the feature information obtained from previous layers. It evaluates the membership degree of each pixel to a class using fuzzy set theory, and optimizes the network's end-to-end learning process with the fuzzy Jaccard Index loss function. The inclusion of the fuzzy logic layer not only enhances the model's ability to handle fuzzy boundaries but also improves the model's precision in recognizing various tear film rupture scenarios. The overall framework of the model is shown in Fig. 7. The pipeline of FIG-YOLO incorporates fuzzy logic processing to enhance feature extraction robustness. Beginning with an RGB input image ($H \times W \times 3$), the system first applies fuzzy logic operations to generate 64-channel feature maps then processes them through a backbone network consisting of convolutional layers, C2f modules, and SPPF spatial pyramid pooling to produce multi-scale features (20×20, 40×40, and $16 \times 16 \times 512$). These features are subsequently fused with fuzzy logic-enhanced maps in the neck section through concatenation and upsampling operations, refined by C3k2 modules, and finally fed into 3 detection heads for multi-scale prediction. The architecture demonstrates three key advantages: integration of fuzzy logic at both initial and feature fusion stages to handle boundary ambiguity in challenging conditions, maintained computational efficiency through optimized modules like C2f and SPPF, and effective multi-scale detection capability through hierarchical feature aggregation, making it particularly suitable for real-time detection tasks in complex environments such as infrastructure inspection or autonomous driving where illumination variations and image blur are common challenges.

3.11 Fuzzy Logic Layer

The core idea of the fuzzy logic layer is to allow for the presence of uncertainty and fuzziness when processing data, in contrast to traditional binary logic (where things are either true or false). In the detection of pavement crack, the fuzzy logic layer helps address issues of uncertainty and similarity within the image, thereby improving both the accuracy and robustness of recognition. The fuzzy logic layer enables the system to handle the uncertainty and complexity of image data in a more flexible and adaptive manner, thus enhancing the accuracy and reliability

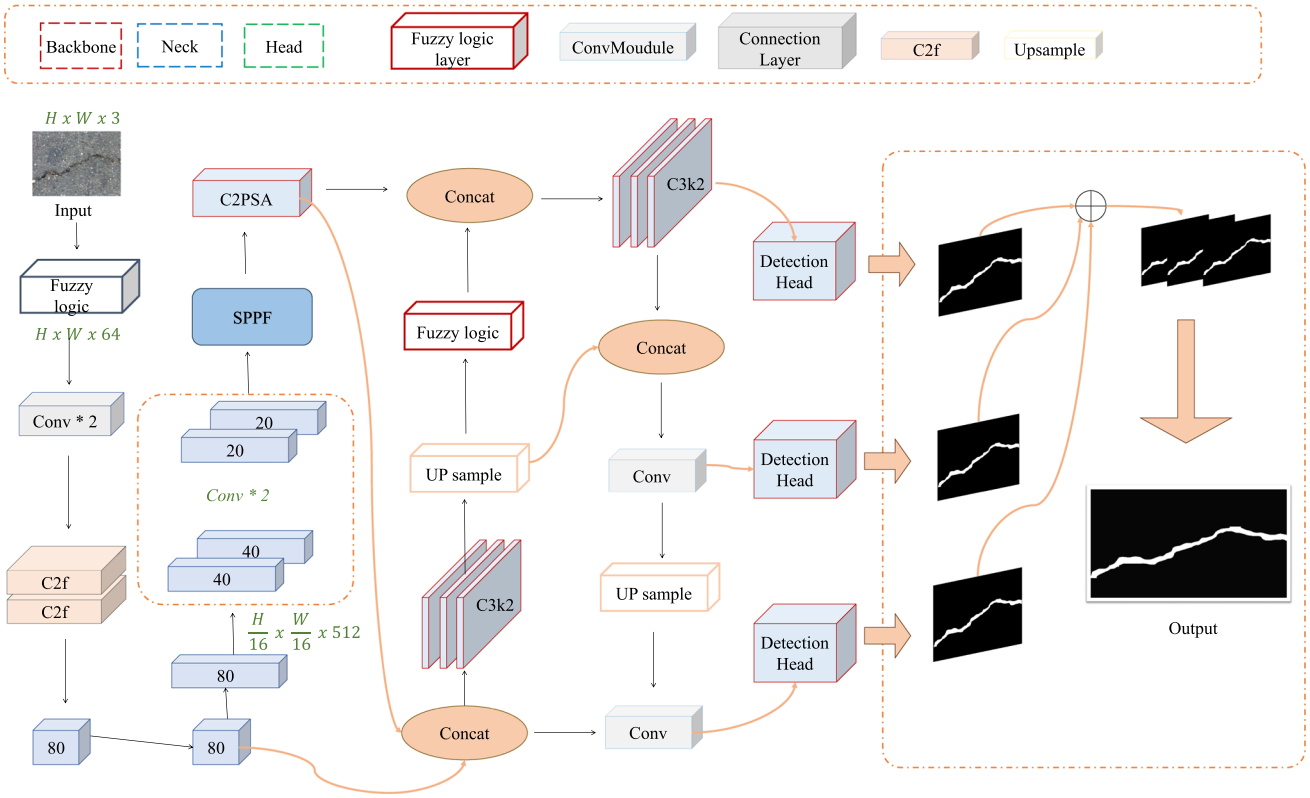

Fig. 7 The architecture of FIG-YOLO inherits the original style of the YOLO series but incorporates three additional fuzzy logic layers in the backbone during feature extraction to address the challenges associated with crack edge information

of pavement crack detection. This method is particularly useful in situations where the boundaries are unclear, categories overlap, or there is a high fuzziness.

By introducing fuzzy logic into the model's decision-making process, the fuzziness and uncertainty of pavement crack boundaries in the image can be effectively addressed. The fuzzy logic layer processes the feature map input, and by fusing fuzzy feature maps before output, the model becomes more robust when handling unclear pavement crack boundaries. This approach helps the model make more accurate judgments in areas with higher uncertainty.

In the first stage, fuzzification, the fuzzy logic layer converts the input feature map $X \in H \times W \times Ch$ into membership degrees in a fuzzy set, where H, W, and Ch represent the height, width, and number of channels of the feature map, respectively. The overall processing flow is shown in Fig. 7.

The fuzzy logic layer processes the input from the raw feature map X by calculating the fuzzy uncertainty. The fuzzy uncertainty of the raw feature map X is obtained by calculating the similarity between pixels, as shown in the formula below:

$$M_{i,j} = \sum_{m=-1}^{1} \sum_{n=-1}^{1} X_{i+m,j+n} \cdot K_{m,n} \tag{11}$$

In the formula, $M \in H \times W \times Ch$ is the feature map obtained by mapping the raw feature map to the fuzzy set. $X(i+m, j+n)$ is the pixel value at position $(i+m, j+n)$ in the input image, and $K(m, n)$ is the weight of the 3×3 convolutional kernel at the relative position (m, n). The indices m and n vary within the size range of the convolutional kernel. Fuzzy Entropy is a dimensionless measure used to assess the complexity of signal sequences. Based on fuzzy mathematical theory, it calculates the fuzzy entropy of a random variable. The definition of fuzzy entropy $H(X)$ is as follows: Let X be a random variable with a range of [0,1] and probability density function

$f(x)$.The fuzzy entropy $H(X)$ is defined by the formula below:

$$H(X) = -\int_0^1 f(x)\ln(f(x))dx \tag{12}$$

The degree to which a pixel belongs to different categories can be used to measure uncertainty. If a pixel contains similarities from different categories, it is difficult to assign it to one specific category. Fuzzy entropy reflects this situation. A pixel with high fuzzy entropy is defined as an uncertain pixel, while a pixel with low fuzzy entropy is defined as a certain pixel. After processing with fuzzy entropy, uncertain maps u' and certain maps u can be obtained. For the member vector i in u, the definition of fuzzy entropy is given by the formula below:

$$H(\mu_i) = -\frac{1}{\log C} \times \sum_{r=1}^{C} \mu_{ir} log \mu_{ir} \tag{13}$$

In the formula, C represents the category, and i_r denotes the probability that the i-th sample belongs to the r-th category. If the uncertainty μ_i is close to 1, the features of the pixel i generated in the convolutional block are uncertain. The features of uncertain pixels should have their weight reduced in the new feature map. These features will be replaced to reduce uncertainty.

After fuzzification, the fuzzy uncertainty μ_{ir} of the original image's pixels is transformed from the original feature map. To calculate the fuzzy uncertainty for all pixels, the input feature map is first transformed into a fuzzy uncertainty feature map. The calculation for each pixel is given by the formula, shown as follows:

$$\mu_{ir} = \frac{1}{1 + exp(a_{ir}x_i + \beta_{ir})} \tag{14}$$

In the formula, μ_{ir} is the fuzzy uncertainty of the target pixel; a_{ir} and β_{ir} are the model's hyperparameters, which are inferred during the model training process. In the experiments of this paper, we set a_{ir} and $\beta_{ir}$to 0.3 and 0.7, respectively. The membership of the feature map is obtained through two layers of 1×1 convolution operations. The specific formula is as follows:

$$\mu = Conv1 \times 1(Conv1 \times 1(M)) \tag{15}$$

In the formula, $\mu \in \mathbb{R}^{h \times w \times c}$ represents the membership feature map of the input feature map M; Conv1x1 refers to a 2D 1×1 convolution layer. In this paper, two layers of Conv1x1 are used, which allow the membership of each category to be more appropriately aligned with its corresponding class. $\mu_i \in \mu = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{iC}]$ is the membership vector of pixel i in the feature map, and the output is obtained through Softmax normalization. The formula is as follows:

$$Softmax(\mu_i) = \frac{e^{\mu_i}}{\sum_{j=1}^{C} e^{\mu_j}} \tag{16}$$

In the formula, μ_i represents the membership output value for the i-th pixel; C is the number of categories. The Softmax function is used to convert the membership output value of each pixel into a probability distribution that is within the range [0, 1] and sums to 1.

During the feature fusion stage, fuzzy logic is employed to enhance boundary information in the image and reduce noise, thereby improving detection accuracy. The primary method involves fusing the original feature map, the certain feature map, and the uncertain feature map. The formula for feature map fusion is as follows:

$$X' = X \otimes u' \oplus (Conv2D(X) \otimes u) \tag{17}$$

In the formula, $X' \in \mathbb{R}^{H\times W\times Ch}$ represents the output feature map; u is the certainty feature map; u' is the complement of u, representing the uncertainty feature map; $\otimes$ denotes element-wise multiplication of matrices; and $\oplus$ denotes element-wise summation of matrices. Through feature fusion, if the certainty of a pixel is high, the pixel's weight in the new feature map remains high. If the uncertainty of a pixel is high, the weight in the new feature map is reduced. This step helps diminish the uncertain information in the image, allowing the model to focus more on certain information.

The introduction of the fuzzy logic layer provides the model with a new dimension of learning, enabling it to recognize and handle the complexity and uncertainty in images. The addition of the fuzzy logic layer not only enhances the model's adaptability and robustness when facing fuzzy category boundaries but also increases the model's sensitivity to subtle changes in the image, thus contributing to improved overall detection and classification performance.

3.12 Fuzzy Loss Function

Considering the fuzzy boundaries and uncertainty, an effective loss function is the fuzzy Jaccard Index loss function. The fuzzy Jaccard Index loss function is a method used to measure the similarity between predicted and ground truth segmentations, and it is particularly suitable for addressing issues such as class imbalance and unclear boundaries. In pavement crack detection, because the boundary between rupture and non-rupture areas can be very fuzzy, using the fuzzy Jaccard Index loss function allows for better handling of uncertainty. The formula for the Jaccard Index loss function is as follows:

$$J_{\text{fuzzy}}\left(Y, \widehat{Y}\right) = 1 - \frac{1}{N}\sum_{i=1}^{N}\frac{\min(Y_i, \widehat{Y}_\iota)}{\max(Y_i, \widehat{Y}_\iota)} \tag{18}$$

In the formula, Y represents the ground truth labels; $\hat{Y}$ is the model's predicted result; N denotes the number of samples. The similarity between the prediction and the ground truth is measured by calculating the ratio between the minimum and maximum membership similarities.

The fuzzy Jaccard Index loss function is defined by its complement, which is then used for optimization. The specific formula is as follows:

$$L_{\text{fuzzy}}(Y, \widehat{Y}) = 1 - J_{\text{fuzzy}}(Y, \widehat{Y}) \tag{19}$$

The fuzzy Jaccard Index loss function accounts for uncertainty and fuzziness between the prediction and the ground truth by comparing the minimum and maximum membership degrees. It naturally addresses the issue of fuzzy boundaries in pavement crack detection. The pavement crack area and the non-rupture area are often imbalanced in terms of their quantity in the image. The fuzzy Jaccard loss function inherently tolerates sample quantity imbalance, allowing for a more fair evaluation of the model's performance.

4 Experiment and Analysis

To validate the effectiveness of the CRS method and the FIG-YOLO model for tear film break-up detection, we performed a series of comprehensive ablation experiments using the CRS method across multiple datasets. In addition, we compared the performance of the FIG-YOLO model with many classical and state-of-the-art segmentation models. The results consistently demonstrated the superior effectiveness of the CRS and FIG-YOLO model, as well as its ability to generalize across a variety of scenarios. The extensive experimental findings indicate that the proposed FIG-YOLO model not only achieves a competitive level of operational efficiency in terms of parameter count and floating-point operations but also demonstrates strong robustness when applied to

 https://doi.org/10.1007/s44196-025-00894-5

Table 1 Experimental environment configuration

Item	Version
Operating system	Windows 11
CPU	Intel(R) Xeon(R) Gold 5218 CPU @ 2.30 GHz
GPU	NVIDIA Quadro RTX 4000
	NVIDIA Quadro RTX 4000
Memery	128 GB
Pytorch	2.2.0+cu121
CUDA	12.2

different conditions and datasets. This suggests that the FIG-YOLO model is both efficient and adaptable, making it a promising solution for real-world applications in dry eye detection.

4.1 Implementation

All experiments in this study were conducted on two NVIDIA Quadro RTX 4000 graphics cards with a total of 16 GB of VRAM. The server used for these experiments is equipped with 128 GB of memory, providing ample space for efficient computation during model training with large datasets. The remaining experimental environment details are provided in the Table 1.

In this section, we detail the implementation of the proposed pavement crack detection model, including the training setup and key hyperparameters used in our experiments. We trained the model for 100 epochs. This number of epochs was chosen after preliminary experiments to ensure sufficient training without overfitting, allowing the model to learn the relevant features of pavement cracks effectively. The learning rate was set to 0.001 using the Adam optimizer. We found this learning rate to provide a good balance between convergence speed and stability. A learning rate schedule was employed to decrease the learning rate by a factor of 0.1 after every 10 epochs to fine-tune the model during later stages of training. To enhance the robustness of the model and improve generalization, several data augmentation techniques were applied, including rotation, scaling, flipping, and color jittering. These techniques helped the model generalize better across different types of pavement crack images and environmental conditions.

4.2 Dataset

For the algorithm proposed in this paper, we selected four well-established public datasets for pavement crack detection. These datasets are CRACK500 [44], CrackLS315 [45], CrackTree260 [46], and GAPS384 [47], each of which has unique characteristics and serves as a valuable resource for evaluating crack detection models.

CRACK500 [44]: The CRACK500 dataset consists of 500 high-resolution pavement images, each containing a variety of crack patterns under different environmental conditions. This dataset provides a diverse set of road surface conditions, allowing for comprehensive testing of crack detection models. It is widely used in the pavement crack detection community due to its well-annotated images and real-world applicability.

CrackTree260 [46]: CrackTree260 contains 260 images of pavement surfaces with a focus on cracks that exhibit tree-like branching patterns. This dataset is particularly useful for studying cracks that grow in irregular, branching forms, which are often more difficult to detect due to their non-linear structures. It helps evaluate the performance of crack detection algorithms in scenarios with more complex crack geometries.

GAPS384 [47]: The GAPS384 dataset consists of 384 images of road surfaces, with a focus on crack detection in low-contrast conditions and poorly illuminated environments. This dataset challenges models to detect subtle cracks against backgrounds with significant noise or low visibility, making it an important resource for testing the sensitivity and accuracy of detection algorithms under difficult conditions.

CrackLS315 [45]: The CrackLS315 dataset consists of images of road surfaces, with a focus on crack detection in low-contrast conditions and poorly illuminated environments. This dataset challenges models to detect subtle cracks against backgrounds with significant noise or low visibility, making it an important resource for testing the sensitivity and accuracy of detection algorithms under difficult conditions.

By using these four datasets, we compiled a diverse range of well-established datasets–namely CRACK500 [44], CrackLS315 [45], CrackTree260 [46], and GAPS384 [47]–into a unified dataset, which we refer to as the Combination Dataset. The dataset was split into training, validation, and test sets at a ratio of 7:1:2. Specifically, we first pooled all images from the four datasets into a single directory, then randomly allocated them to the respective subsets (train/val/test) using a randomized function. Furthermore, since the original annotations were provided as png mask files, we converted all mask images into the txt format compatible with Ultralytics training using the X-Anylabeling library [48]. This integration allowed us to evaluate our algorithm across a wide spectrum of crack detection scenarios, ensuring robustness and generalizability. We aim to evaluate the performance of the proposed algorithm across a wide range of scenarios, ensuring its robustness and generalizability for real-world pavement crack detection tasks.

4.3 Evolution Metrics

To comprehensively evaluate model performance and practical utility, we employed three primary metrics: accuracy, recall, and mean of average precision (mAP). Additionally, we quantified computational efficiency by measuring model parameters, floating-point operations (FLOPs), and inference time, thereby establishing a robust evaluation framework. These evaluation metrics are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

Additionally, in the data table, mAP_{50} refers to the mean average precision (mAP) calculated at an intersection-over-union (IoU) threshold of 0.5. In contrast, $mAP_{50:95}$ represents the average mAP computed across multiple IoU thresholds. Specifically, it evaluates mAP at 10 IoU thresholds within the range [0.5, 0.95], with a step size of 0.05, and then averages the results. A higher $mAP_{50:95}$ indicates more precise bounding box predictions, as it accounts for a wider range of IoU thresholds, particularly those with higher values.

4.4 Contrast Experiment

We conducted a comprehensive comparative analysis of our proposed method against multiple variants and iterations within the YOLO series, including YOLOv8 [13], YOLOv9 [49], YOLOv11 [13], and their respective derivatives. All models were trained on the training set of the aforementioned Combination Dataset. After each epoch, the validation set was used to evaluate the current epoch's performance, enabling real-time monitoring of the training process. Finally, the test set (which was never involved in training) was employed to compare the performance across all models. Our evaluation framework encompassed both baseline configurations and state-of-the-art improvements, focusing on key performance metrics such as precision, recall, inference speed, and robustness to environmental noise. This rigorous assessment aimed to provide a holistic view of how our algorithm performs relative to existing methods under various conditions. The comparative results are summarized in Table 2, offering a detailed overview of the performance metrics across different models. Additionally, Fig. 8 provides a visual representation of part of the training results on the Combination Dataset, highlighting the strengths and weaknesses of each model in practical scenarios. These findings collectively demonstrate the effectiveness of our proposed algorithm in addressing the challenges of pavement crack detection while maintaining high efficiency and accuracy.

 https://doi.org/10.1007/s44196-025-00894-5

Table 2 We compared the latest model in the YOLO series, including its variants at different scales

Model	Param.(M)	FLOPs(G)	$L + L_{post}(ms)$	Mask(P)	R	$mAP^{test}_{50:95}$	mAP^{test}_{50}
YOLO8n-seg	3.3	12.0	8.1 + 1.1	0.629	0.445	0.462	0.591
YOLO8n-seg-p6	5.1	12.0	10.4 + 1.0	0.635	0.447	0.466	0.601
YOLO11n-seg	2.8	10.2	14.3 + 1.4	0.691	0.471	0.479	0.612
CRS-n	3.2	10.8	12.3 + 1.1	0.689	0.455	0.475	0.595
FIG-YOLOn	2.9	10.3	10.8 + 1.0	0.679	0.456	0.471	0.599
YOLO8s-seg	11.8	42.4	9.3 + 1.2	0.729	0.462	0.478	0.622
YOLO8s-seg-p6	18.6	42.3	14.2 + 1.1	0.711	0.473	0.481	0.631
YOLO11s-seg	10.1	35.3	13.2 + 1.3	0.718	0.453	0.483	0.636
CRS-s	11.2	46.8	14.3 + 1.1	0.722	0.461	0.481	0.622
FIG-YOLOs	10.5	45.9	13.8 + 1.1	0.719	0.458	0.477	0.639
YOLO8m-seg	27.2	110.0	19.8 + 1.2	0.780	0.472	0.481	0.667
YOLO8m-seg-p6	46.4	114.1	25.2 + 1.2	0.762	0.477	0.496	0.651
YOLO11m-seg	22.3	123.0	21.2 + 1.5	0.798	0.467	0.505	0.652
CRS-m	23.6	125.2	22.3 + 1.2	0.781	0.478	0.503	0.658
FIG-YOLOm	22.6	123.6	21.3 + 1.3	0.778	0.476	0.492	0.649
YOLO8l-seg	45.9	220.1	32.1 + 1.2	0.789	0.452	0.488	0.666
YOLO8l-seg-p6	64.9	222.3	39.2 + 1.3	0.781	0.454	0.478	0.671
YOLO9c-seg	27.6	157.6	26.6 + 1.3	0.778	0.461	0.481	0.656
YOLO11l-seg	27.6	141.9	26.1 + 1.5	0.789	0.471	0.489	0.673
CRS-l	28.8	144.3	28.5 + 1.3	0.790	0.456	0.483	0.669
FIG-YOLOl	28.1	142.6	26.4 + 1.2	0.784	0.458	0.486	0.665
YOLO8x-seg	71.7	343.7	48.9 + 1.6	0.772	0.460	0.478	0.645
YOLO8x-seg-p6	101.4	347.1	60.1 + 1.4	0.778	0.467	0.481	0.654
YOLO9e-seg	59.7	244.4	49.6 + 1.3	0.762	0.451	0.485	0.656
YOLO11x-seg	62.0	318.5	45.7 + 1.3	0.781	0.457	0.479	0.677
CRS-x	63.8	321.9	46.8 + 1.2	0.784	0.461	0.482	0.671
FIG-YOLOx	62.5	319.5	46.1 + 1.3	0.776	0.458	0.478	0.668

Our model was trained on the Combination dataset and evaluated using the dataset's built-in test set. The performance of our model was then benchmarked against other YOLO series models. Additionally, in the term $L + L_{post}$, L represents the model inference time, while L_{post} denotes the post-processing time

4.5 Analysis of Contrast Experiment

We conducted a comprehensive comparison and analysis of the overall data as well as local models of the same scale. First, as shown in Table 2, the overall data indicate that the model's performance does not significantly improve beyond the medium (m) scale. In fact, the extra-large (x) scale model exhibits a slight decline in performance compared to the large (l) scale model, as illustrated in the Fig. 8. This trend may be attributed to overfitting on the Combination Dataset as the number of model parameters increases, leading to reduced performance on the test set. Conversely, the dataset may be too small to provide sufficient semantic and label information for models with larger parameter sizes, such as the l and x scales. To further investigate this, we plotted the $Param - mAP_{50:95}$ and $FLOPs - mAP_{50:95}$ curves, as shown in the Figs. 9 and 10.

Additionally, the data from local models of the same scale reveal that, from the nano (n) to medium (m) scale, the YOLOv11 model achieves the highest $mAP_{50:95}$ value at the smallest (n) scale. However, as the number of parameters and floating-point operations (FLOPs) increase, the guiding effect of the attention mechanism becomes more pronounced. Consequently, at the final medium (m) scale, the CRS model outperforms YOLOv11 across all metrics.

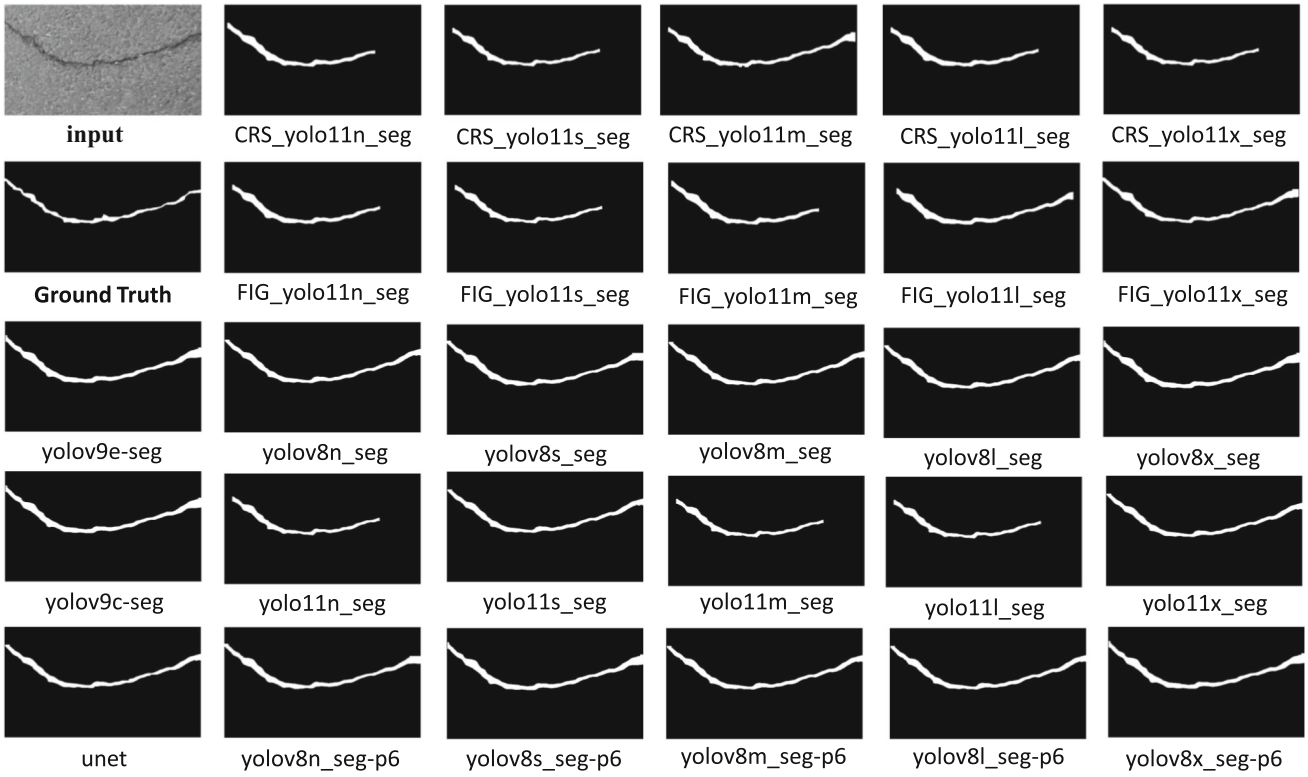

Fig. 8 A visual comparison of prediction results between different YOLO series models and our improved CRS series models a presented

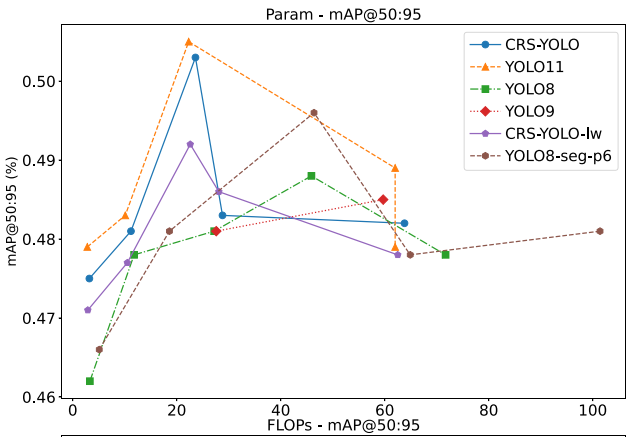

Fig. 9 Parameter and $mAP_{50:95}$ curves, $Param - mAP_{50:95}$, of YOLO series models and CRS series models

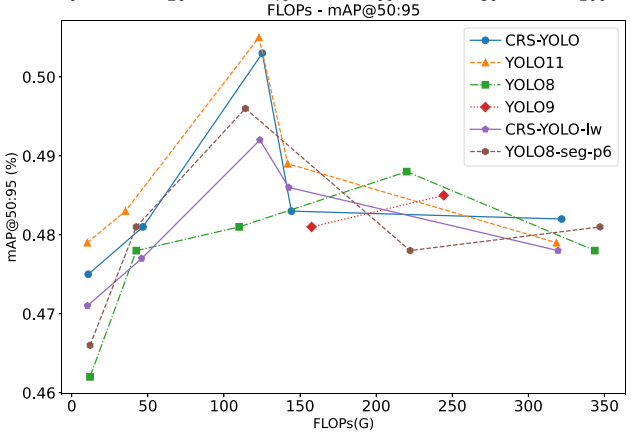

Fig. 10 FLOPs and $mAP_{50:95}$ curves, $FLOPs - mAP_{50:95}$, of YOLO series models and CRS series models

 https://doi.org/10.1007/s44196-025-00894-5

Table 3 The CRS-n-seg series incorporates attention mechanism modules at varying depths within the decoder to progressively evaluate the impact of these added attention mechanisms on the original YOLO model

Model	$Param.(M)$	$FLOPs(G)$	$L + L_{post}(ms)$	$Mask(P)$	R	$mAP^{test}_{50:95}$	mAP^{test}_{50}
CRS-n-a	2.9	10.3	8.1 + 1.1	0.629	0.445	0.462	0.591
CRS-n-b	3.0	10.5	9.4 + 1.0	0.635	0.447	0.466	0.601
CRS-n-c	3.2	10.8	10.3 + 1.1	0.691	0.471	0.479	0.612

All ablation experiment models were trained and validated on the Crack500 dataset

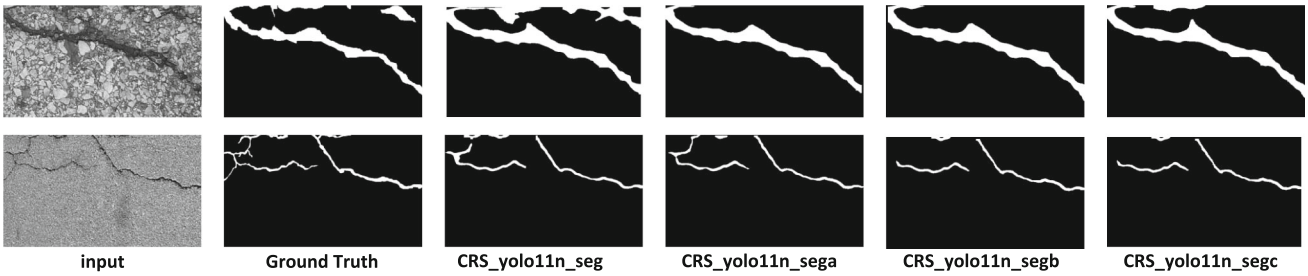

Fig. 11 Visualization of CRS algorithm ablation experiment. CRS-a, CRS-B, and CRS-C represent the replacement of the convolutional layer CRS layer at different sites, respectively

Table 4 We progressively reduced redundant convolutional modules at different locations, including standard convolutions and C3k2 modules, to investigate the impact of reducing varying numbers and types of convolutional modules on overall model performance

Model	$Param.(M)$	$FLOPs(G)$	$L + L_{post}(ms)$	$Mask(P)$	R	$mAP^{test}_{50:95}$	mAP^{test}_{50}
FIG-YOLOn-a	3.2	10.8	14.1 + 1.1	0.629	0.445	0.462	0.591
FIG-YOLOn-b	3.1	10.6	14.4 + 1.0	0.635	0.447	0.466	0.601
FIG-YOLOn-c	3.1	10.5	12.2 + 1.4	0.691	0.471	0.479	0.612
FIG-YOLOn-d	3.0	10.4	13.3 + 1.1	0.689	0.455	0.475	0.595
FIG-YOLOn-e	2.9	10.3	10.8 + 1.0	0.679	0.456	0.471	0.599

All experiments were trained and validated on the Crack500 dataset

4.6 Ablation Experiment

To further validate the performance of our proposed model, we conducted a series of comprehensive ablation studies. First, we investigated the impact of varying the insertion positions and numbers of attention mechanisms within the model architecture. Specifically, we experimented with different configurations to determine how the placement and quantity of attention modules affect overall performance. The results of these experiments are detailed in Table 3. Additionally, we selected several representative generated results for visual analysis to provide a more intuitive understanding of the model's behavior. These visual results are presented in Fig. 11.

Furthermore, we explored the effects of incorporating additional fuzzy logic layers in the FIG-YOLO model. This was done progressively to observe how each incorporation impacts the performance of both the original YOLO and the enhanced FIG-YOLO models. By systematically pruning convolutional layers, we aimed to identify the optimal balance between model complexity and performance. The experimental data summarizing these findings are provided in Table 4. The corresponding visualization results, which offer insights into the model's performance under different layer configurations, are shown in Fig. 12.

These ablation studies collectively provide a detailed understanding of the model's behavior and highlight the effectiveness of the proposed architectural modifications in enhancing performance while maintaining computational efficiency.

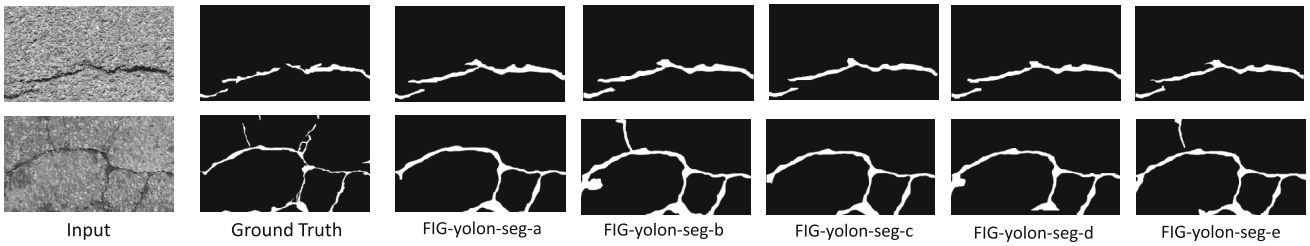

Fig. 12 Visualization of FIG-YOLO model ablation experiment. **a–e** Adding fuzzy membership functions to different sites respectively

4.7 Analysis of Ablation Experiment

In this section, we provide a detailed analysis of the ablation experiments conducted to validate the effectiveness and efficiency of our proposed model. The first set of ablation studies focused on the insertion of attention mechanism modules at varying depths within the decoder of the CRS-n model. As shown in Table 3, the incorporation of attention mechanisms led to improvements in both precision and recall metrics. Specifically, the model with the deepest attention module (CRS-n-c) achieved the highest $mAP_{50:95}$ of 47.9%, indicating that deeper attention mechanisms can enhance the model's ability to detect and segment objects accurately.

The second set of experiments aimed to evaluate the effects of incorporating additional fuzzy logic layers in the FIG-YOLOn model. As summarized in Table 4, the progressive incorporation of fuzzy logic layers led to varying impacts on model performance. For example, FIG-YOLOn-c achieved a higher $mAP_{50:95}$ of 47.9% compared to FIG-YOLOn-a, despite having fewer parameters and FLOPs. This indicates that the removal of redundant layers can enhance model efficiency without significantly compromising performance.

The visual results presented in Figs. 11 and 12, which provides further insights into the effectiveness of the architectural modifications. For instance, the attention mechanisms in CRS-n-seg models led to more accurate and complete segmentation masks, as evidenced by the improved precision and recall metrics. Similarly, the incorporation of fuzzy logic layers in FIG-YOLOn models resulted in more efficient inference times while maintaining reasonable segmentation accuracy.

The ablation experiments demonstrate that the incorporation of attention mechanisms and the incorporation of fuzzy logic layers can significantly enhance the performance and efficiency of the CRS models and FIG-YOLO. While attention mechanisms improve detection and segmentation accuracy, their placement and number should be carefully optimized to avoid excessive computational overhead. Similarly, incorporating additional fuzzy logic layers can enhance model efficiency, but an optimal balance must be maintained to prevent performance degradation. Future work will focus on further optimizing these architectural components to achieve better trade-offs between performance and efficiency.

4.8 Extended Experiment

To further validate and test the performance of CRS and FIG-YOLO, as well as its compatibility with similar domains, we utilized two additional defect detection datasets: MT [50] and DAGM2007 [51].

MT Dataset The MT dataset is a widely used benchmark for defect detection in industrial settings, featuring high-resolution images of surface defects such as cracks, scratches, and dents. It includes diverse scenarios with varying lighting conditions, material types, and defect sizes, making it suitable for evaluating model robustness and generalization capabilities.

DAGM2007 Dataset The DAGM2007 dataset is a synthetic dataset designed for defect detection in textured surfaces. It contains grayscale images with artificially generated defects, simulating real-world industrial inspection tasks. The dataset is divided into multiple classes, each representing a specific type of defect, and provides a challenging testbed for evaluating model precision and adaptability to complex textures.

 https://doi.org/10.1007/s44196-025-00894-5

Table 5 Extensive additional experiments were conducted using the MT dataset and the DAGM2007 dataset

Model	MT dataset					GADM2007 dataset			
	$L + L_{post}$ (ms)	$Mask(P)$	R	$mAP^{test}_{50:95}$	mAP^{test}_{50}	$Mask(P)$	R	$mAP^{test}_{50:95}$	mAP^{test}_{50}
YOLO8n-seg	15.6 + 1.7	0.559	0.253	0.266	0.583	0.995	0.2	0.242	0.557
YOLO8n-seg-p6	23.5 + 1.7	0.560	0.434	0.398	0.535	0.998	0.2	0.417	0.572
YOLO11n-seg	20.8 + 1.5	0.565	0.46	0.466	0.518	0.998	0.2	0.384	0.57
CRS-n	23.6 + 1.5	0.698	0.535	0.579	0.545	0.998	0.2	0.510	0.558
FIG-YOLOn	22.2 + 1.3	0.471	0.612	0.573	0.541	0.998	0.2	0.505	0.536
YOLO8s-seg	17.1 + 1.8	0.510	0.421	0.432	0.626	0.964	0.2	0.271	0.670
YOLO8s-seg-p6	21.2 + 1.4	0.561	0.482	0.483	0.573	0.993	0.2	0.240	0.555
YOLO11s-seg	20.5 + 1.5	0.563	0.409	0.374	0.647	0.997	0.2	0.486	0.531
CRS-s	24.9 + 1.5	0.543	0.456	0.492	0.638	0.998	0.2	0.465	0.633
FIG-YOLOs	23.0 + 1.5	0.648	0.525	0.571	0.644	0.998	0.2	0.445	0.621
YOLO8m-seg	26.3 + 1.5	0.556	0.498	0.398	0.612	0.998	0.2	0.486	0.622
YOLO8m-seg-p6	26.8 + 1.6	0.520	0.530	0.401	0.642	0.993	0.2	0.378	0.397
YOLO11m-seg	24.8 + 1.4	0.517	0.523	0.549	0.610	0.96	0.2	0.328	0.421
CRS-m	29.5 + 1.4	0.596	0.576	0.555	0.637	0.997	0.2	0.519	0.354
FIG-YOLOm	25.6 + 1.2	0.643	0.510	0.508	0.621	0.996	0.2	0.515	0.346
YOLO8l-seg	23.7 + 0.8	0.512	0.496	0.468	0.612	0.964	0.2	0.391	0.599
YOLO8l-seg-p6	30.0 + 0.6	0.520	0.303	0.513	0.542	0.993	0.2	0.322	0.554
YOLO9c-seg	30.8 + 0.8	0.456	0.406	0.486	0.519	0.991	0.2	0.283	0.514
YOLO11l-seg	34.7 + 1.4	0.359	0.422	0.374	0.592	0.996	0.2	0.417	0.589
CRS-l	43.2 + 1.4	0.399	0.492	0.401	0.572	0.996	0.2	0.436	0.578
FIG-YOLOl	34.7 + 1.2	0.508	0.474	0.395	0.513	0.994	0.2	0.344	0.523
YOLO8x-seg	27.2 + 0.9	0.502	0.456	0.415	0.566	0.964	0.2	0.391	0.519
YOLO8x-seg-p6	29.7 + 0.6	0.498	0.398	0.346	0.512	0.964	0.2	0.231	0.426
YOLO9e-seg	53.0 + 1.3	0.499	0.334	0.306	0.521	0.964	0.2	0.319	0.526
YOLO11x-seg	37.1 + 1.4	0.501	0.393	0.309	0.542	0.997	0.2	0.409	0.538
CRS-x	40.6 + 1.3	0.508	0.46	0.456	0.594	0.994	0.2	0.384	0.552
FIG-YOLOx	38.1 + 1.3	0.517	0.393	0.309	0.541	0.998	0.2	0.412	0.581

The metrics used in these experiments are consistent with those in the comparative experiments

The experiments were conducted using the same training and evaluation protocols as described in the previous sections. The models were trained on the combined dataset and evaluated using metrics such as $Precision(P)$, $Recall(R)$, $mAP_{50:95}$, and inference time. The results are summarized in Table5.

4.9 Analysis of Extended Experiment

On the MT dataset, the performance of our proposed CRS, FIG-YOLO models and the original YOLO series models is similar to that observed on road detection datasets. However, the results are not particularly stable, with minimal differences across different model variants. As illustrated in the Fig. 13. This further validates that, on small datasets, increasing model parameters or floating-point operations does not necessarily lead to significant performance improvements. Instead, it can substantially increase computational burden and costs.

In contrast, the GADM2007 dataset exhibited surprisingly high precision, with nearly every model achieving over 95% precision. However, this was accompanied by a significantly low recall rate. This situation may be related to class imbalance in the dataset. If the dataset contains far fewer positive samples (target regions) than negative samples (background regions), as shown in Fig. 14., this aligns with the data distribution of GADM2007.

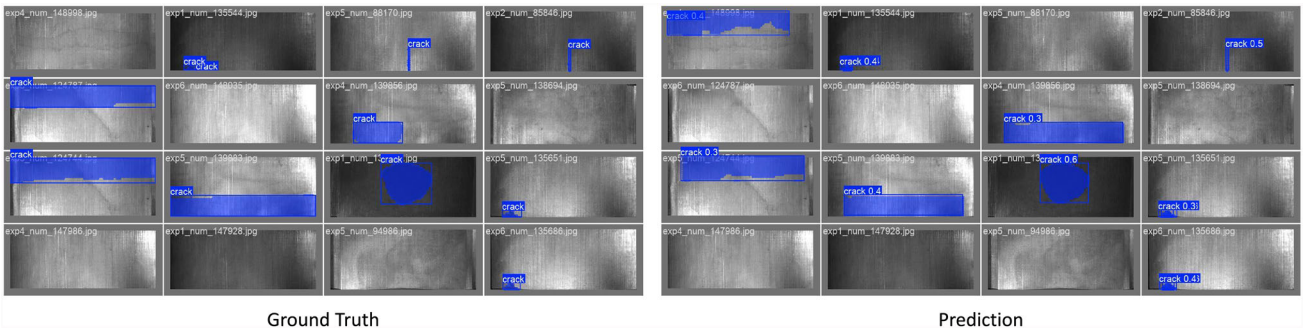

Fig. 13 In part by CRS prediction results and the corresponding real value label visualization

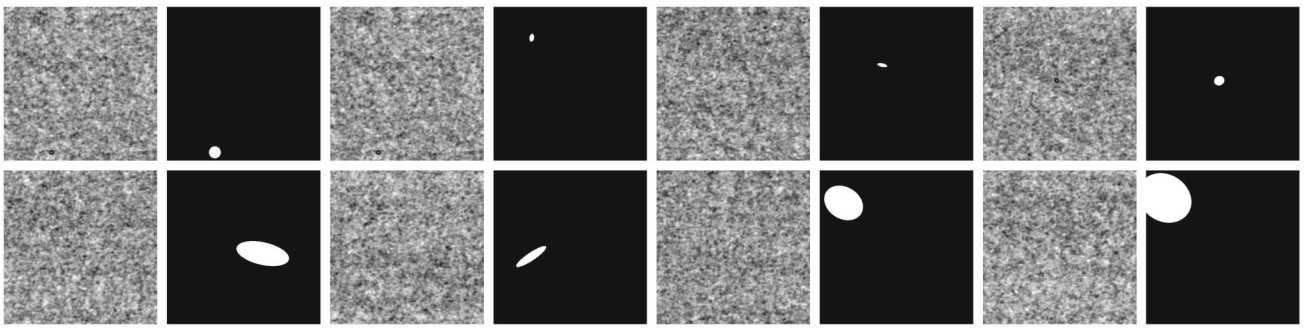

Fig. 14 Part of the GADM2007 dataset is visually displayed with its labels

In future research, we may address this issue by exploring modifications to the loss function to better handle class imbalance.

The experiments on the MT and DAGM2007 datasets further validated the effectiveness and robustness of the proposed CRS and FIG-YOLO models. The results demonstrated that these models can handle diverse scenarios and complex textures, making them suitable for real-world applications in defect detection. Future work will focus on further optimizing the models for real-time performance and exploring multi-modal approaches to enhance detection accuracy.

Discussion Despite the promising results of our proposed methods, several challenges remain in the field of pavement crack detection. First, the variability in crack appearance, including size, shape, and environmental conditions, continues to pose challenges for model generalization. While our CRS showed robust performance across multiple datasets, further improvements are needed to handle even more complex scenarios, such as cracks in heavily occluded or low-visibility environments. Second, the computational efficiency of the models is a critical factor for real-time applications. Although our CRS model achieved a good balance between accuracy and speed, further optimization is necessary to reduce the computational burden and enable deployment on edge devices with limited resources. This could involve exploring more efficient network architectures or leveraging model compression techniques such as pruning and quantization. What's more, the integration of additional contextual information, such as road surface type or environmental conditions, could further enhance the performance of crack detection models. Future work could explore multi-modal approaches that combine visual data with other sensor inputs to provide a more comprehensive understanding of road conditions. Finally, the development of larger and more diverse datasets is essential for improving model generalization and robustness. Current datasets, while valuable, may not fully capture the range of real-world scenarios encountered in pavement crack detection. Collaborative efforts to create more extensive and varied datasets could significantly advance the field. In conclusion, our study presents a novel and effective approach for automated pavement crack detection, leveraging attention mechanisms and fuzzy logic to address key challenges in this domain. The proposed methods demonstrate strong performance

and generalization capabilities, offering a promising direction for the development of intelligent road maintenance systems. Future work will focus on further optimizing model efficiency, exploring multi-modal approaches, and expanding dataset diversity to enhance the robustness and applicability of our methods in real-world scenarios.

5 Conclussion and Limitation

Conclusion In this study, we proposed a novel approach for pavement crack detection using a lightweight YOLOv11-based model with attention mechanisms and fuzzy logic integration. Our method, named CRS, incorporates a Crack Region Segmentation (CRS) approach to focus on relevant crack regions while reducing interference from irrelevant background information. Additionally, we introduced the Fuzzy Information-Guided YOLO (FIG-YOLO) model to handle the uncertainty and imprecision in crack boundaries, leveraging fuzzy logic to improve segmentation accuracy. Through extensive experiments on multiple datasets, our methods demonstrated superior performance in terms of accuracy, robustness, and computational efficiency compared to existing state-of-the-art models. First, We proposed a lightweight YOLOv11-based model with attention mechanism to focus on crack regions, significantly improving detection accuracy while maintaining real-time inference capabilities. And then, through comprehensive ablation experiments, we analyzed the impact of attention mechanisms and convolutional module optimization on model performance, providing insights into the optimal architecture for efficient and accurate crack detection. Our models demonstrated strong generalization capabilities across different datasets, including CRACK500, CrackLS315, CrackTree200, and GAPS384, as well as industrial defect detection datasets MT and DAGM2007.

Limitation In this study, we independently employ both CRS and FIG-YOLO models for road crack detection, each serving distinct purposes. The CRS model primarily functions to direct the model's attention to segmentation targets, while FIG-YOLO is specifically designed to address the challenge of ambiguous boundaries, thereby achieving superior segmentation results. However, our current work does not integrate these two models, as both are developed by introducing new modules to the YOLOv11 architecture. Although our newly added modules require minimal computational overhead, their incorporation inevitably increases the overall computational load compared to the original YOLOv11. To maintain comparable inference speed while preserving the enhanced accuracy from these modules, we had to remove certain native components of YOLOv11. Combining both CRS and FIG-YOLO would significantly increase computational demands, making it difficult to maintain low inference latency and meet real-time processing requirements. Therefore, we currently employ these algorithms separately. For future research, we aim to develop solutions that can maintain low latency while incorporating additional modules. This direction would allow us to improve model accuracy without compromising computational efficiency, achieving optimal performance without sacrificing processing speed.

Author Contributions Qingqing Li contributed to the conceptualization, methodology and wrote the final draft. Tianshu Wu and Tingfa Xu carried out experiments and wrote the original draft. The remaining authors contributed to validating the ideas, carrying out additional analyses and reviewing this paper. All authors read and approved the manuscript.

Funding This work was supported by the Technology Innovation and Application Development Projects of Chongqing (Grant No. CSTB2022TIAD-STX0001), and Chongqing Research Program of Basic Research and Frontier Technology (Grant No. cstc2021jcyj-msxmX0530).

Data Availability Data availability is not applicable to this article as no new data were created or analyzed in this study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Wang, H., Wang, B., Zhao, T.: Shuff-BiseNet: a dual-branch segmentation network for pavement cracks. SIViP **18**(4), 3309–3320 (2024)
2. Nasrallah, A.A., Abdelfatah, M.A., Attia, M.I., El-Fiky, G.S.: Positioning and detection of rigid pavement cracks using GNSS data and image processing. Earth Sci. Inf. **17**(2), 1799–1807 (2024)
3. Zhang, T., Wang, D., Lu, Y.: Benchmark study on a novel online dataset for standard evaluation of deep learning-based pavement cracks classification models. KSCE J. Civ. Eng. **28**(4), 1267–1279 (2024)
4. Zhang, S., Bei, Z., Ling, T., Chen, Q., Zhang, L.: Research on high-precision recognition model for multi-scene asphalt pavement distresses based on deep learning. Sci. Rep. **14**(1), 25416 (2024)
5. Zhang, J., Sun, S., Song, W., Li, Y., Teng, Q.: A novel convolutional neural network for enhancing the continuity of pavement crack detection. Sci. Rep. **14**(1), 1–20 (2024)
6. Guo, F., Qian, Y., Liu, J., Yu, H.: Pavement crack detection based on transformer network. Autom. Constr. **145**, 104646 (2023). https://doi.org/10.1016/j.autcon.2022.104646
7. Alkhedher, M., Alsit, A., Alhalabi, M., AlKheder, S., Gad, A., Ghazal, M.: Novel pavement crack detection sensor using coordinated mobile robots. Transp. Res. C Emerg. Technol. **172**, 105021 (2025). https://doi.org/10.1016/j.trc.2025.105021
8. Zhang, Y., Zhang, L.: Detection of pavement cracks by deep learning models of transformer and UNet. IEEE Trans. Intell. Transp. Syst. **25**(11), 15791–15808 (2024). https://doi.org/10.1109/TITS.2024.3420763
9. Cao, T., Wang, Y., Liu, S.: Pavement crack detection based on 3d edge representation and data communication with digital twins. IEEE Trans. Intell. Transp. Syst. **24**(7), 7697–7706 (2023). https://doi.org/10.1109/TITS.2022.3194013
10. Chen, J., Li, H., Zhao, Z., Hou, X., Luo, J., Xie, C., Liu, H., Ren, T., Huang, X.: Investigation of transverse crack spacing in an asphalt pavement with a semi-rigid base. Sci. Rep. **12**(1), 18079 (2022)
11. Ali, L., AlJassmi, H., Swavaf, M., Khan, W., Alnajjar, F.: Rs-net: residual sharp U-Net architecture for pavement crack segmentation and severity assessment. J. Big Data **11**(1), 116 (2024)
12. Zhang, Z., He, Y., Hu, D., Jin, Q., Zhou, M., Liu, Z., Chen, H., Wang, H., Xiang, X.: Algorithm for pixel-level concrete pavement crack segmentation based on an improved U-Net model. Sci. Rep. **15**(1), 6553 (2025)
13. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO. https://github.com/ultralytics/ultralytics
14. Zhang, J., Sun, S., Song, W., Li, Y., Teng, Q.: A novel convolutional neural network for enhancing the continuity of pavement crack detection. Sci. Rep. **14**(1), 1–20 (2024)
15. Qu, Z., Li, M., Yuan, B., Mu, G.: A method of hybrid dilated and global convolution networks for pavement crack detection. Multimedia Syst. **30**(4), 210 (2024)
16. Shilpi, A., Kumar, A.: Sensor node localization using nature-inspired algorithms with fuzzy logic in WSNs. J. Supercomput. **80**(19), 26776–26804 (2024). https://doi.org/10.1007/s11227-024-06464-4
17. Chen, Y., Liu, J., Zhou, J.: Design and application of fuzzy neural network systems optimized with hybrid algorithms. Trans. Inst. Meas. Control **46**(15), 3063–3070 (2024). https://doi.org/10.1177/01423312241266068
18. Zhou, J., Chen, Y.: Study on non-iterative algorithms for center-of-sets type-reduction of interval type-2 Takagi–Sugeno–Kang fuzzy logic systems. Int. J. Fuzzy Syst. (2024). https://doi.org/10.1007/s40815-024-01873-2
19. Aljohani, A.: Optimized convolutional forest by particle swarm optimizer for pothole detection. Int. J. Comput. Intell. Syst. **17**(1), 7 (2024)
20. Wang, X., Zhaozheng, H., Li, N., Qin, L.: Pavement crack analysis by referring to historical crack data based on multi-scale localization. PLoS ONE **15**(8), 0235171 (2020)
21. Shan, J., Jiang, W., Huang, Y., Yuan, D., Liu, Y.: Unmanned aerial vehicle (UAV)-based pavement image stitching without occlusion, crack semantic segmentation, and quantification. IEEE Trans. Intell. Transp. Syst. **25**(11), 17038–17053 (2024). https://doi.org/10.1109/TITS.2024.3424525
22. Li, G., Wan, J., He, S., Liu, Q., Ma, B.: Semi-supervised semantic segmentation using adversarial learning for pavement crack detection. IEEE Access **8**, 51446–51459 (2020). https://doi.org/10.1109/ACCESS.2020.2980086
23. Sun, X., Xie, Y., Jiang, L., Cao, Y., Liu, B.: DMA-Net: DeepLab with multi-scale attention for pavement crack segmentation. IEEE Trans. Intell. Transp. Syst. **23**(10), 18392–18403 (2022). https://doi.org/10.1109/TITS.2022.3158670

24. Han, C., Ma, T., Huyan, J., Huang, X., Zhang, Y.: CrackW-Net: a novel pavement crack image segmentation convolutional neural network. IEEE Trans. Intell. Transp. Syst. **23**(11), 22135–22144 (2022). https://doi.org/10.1109/TITS.2021.3095507
25. Wu, W., Zhou, X., Jin, Y., Fang, Z., Fan, X., Zhang, B., Zheng, R.: A method to detect pavement surface distress based on improved U-Net semantic segmentation network. In: 2023 4th International Conference on Computer Vision, Image and Deep Learning (CVIDL), pp. 625–630 (2023). https://doi.org/10.1109/CVIDL58838.2023.10165980
26. Güraksın, G.E., Kayadibi, I.: A hybrid LECNN architecture: a computer-assisted early diagnosis system for lung cancer using CT images. Int. J. Comput. Intell. Syst. **18**(1), 35 (2025)
27. Sarhan, A.M., Gobara, M., Yasser, S., Elsayed, Z., Sherif, G., Moataz, N., Yasir, Y., Moustafa, E., Ibrahim, S., Ali, H.A.: Knee osteoporosis diagnosis based on deep learning. Int. J. Comput. Intell. Syst. **17**(1), 241 (2024)
28. Obaid, A.M., Turki, A., Bellaaj, H., Ksantini, M.: Diagnosis of gallbladder disease using artificial intelligence: a comparative study. Int. J. Comput. Intell. Syst. **17**(1), 46 (2024)
29. Pérez-Cano, F.D., Parra-Cabrera, G., Jiménez-Delgado, J.J.: An approach to microscopic cortical bone fracture simulation: enhancing clinical replication. Int. J. Comput. Intell. Syst. **17**(1), 102 (2024)
30. Liu, J., Wang, Z., Wan, G., Liu, J.: A novel multi-modal sentiment analysis based on multiple kernel learning with margin-dimension constraint. Int. J. Comput. Intell. Syst. **17**(1), 207 (2024)
31. Moradi, M., Assaf, G.J.: Designing and building an intelligent pavement management system for urban road networks. Sustainability **15**(2), 1157 (2023)
32. Utami, R., Sandoval, C.O., Thom, N.: Conceptual framework for integrating building information modelling (BIM) with pavement management system (PMS). In: International Conference on Computing in Civil and Building Engineering, pp. 117–131. Springer (2024)
33. Oreto, C., Massotti, L., Biancardo, S.A., Veropalumbo, R., Viscione, N., Russo, F.: BIM-based pavement management tool for scheduling urban road maintenance. Infrastructures **6**(11), 148 (2021)
34. Mei, X., Gunaratne, M., Lu, J., Dietrich, B.: Neural network for rapid depth evaluation of shallow cracks in asphalt pavements. Comput. Aided Civ. Infrastruct. Eng. **19**(3), 223–230 (2004)
35. Sharma, M., Lim, J., Lee, H.: The amalgamation of the object detection and semantic segmentation for steel surface defect detection. Appl. Sci. **12**(12), 6004 (2022)
36. He, Y., Niu, X., Hao, C., Li, Y., Kang, L., Wang, Y.: An adaptive detection approach for multi-scale defects on wind turbine blade surface. Mech. Syst. Signal Process. **219**, 111592 (2024)
37. Luo, S., Xu, Y., Zhang, C., Jin, J., Kong, C., Xu, Z., Guo, B., Tang, D., Cao, Y.: LIDD-YOLO: a lightweight industrial defect detection network. Meas. Sci. Technol. **36**(1), 0161–5 (2024)
38. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2017). https://doi.org/10.1109/TPAMI.2016.2572683
39. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, pp. 234–241. Springer, Cham (2015)
40. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018). https://doi.org/10.1109/TPAMI.2017.2699184
41. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017). https://doi.org/10.1109/TPAMI.2016.2644615
42. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6230–6239 (2017). https://doi.org/10.1109/CVPR.2017.660
43. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision—ECCV 2018, pp. 3–19. Springer, Cham (2018)
44. Zhang, L., Yang, F., Daniel Zhang, Y., Zhu, Y.J.: Road crack detection using deep convolutional neural network. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3708–3712 (2016). https://doi.org/10.1109/ICIP.2016.7533052
45. Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., Wang, S.: DeepCrack: learning hierarchical convolutional features for crack detection. IEEE Trans. Image Process. **28**(3), 1498–1512 (2019). https://doi.org/10.1109/TIP.2018.2878966
46. Zou, Q., Cao, Y., Li, Q., Mao, Q., Wang, S.: CrackTree: automatic crack detection from pavement images. Pattern Recognit. Lett. **33**(3), 227–238 (2012). https://doi.org/10.1016/j.patrec.2011.11.004
47. Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stoeckert, U., Gross, H.-M.: How to get pavement distress detection ready for deep learning? A systematic approach. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2039–2047 (2017). https://doi.org/10.1109/IJCNN.2017.7966101
48. Wang, W.: Advanced Auto Labeling Solution with Added Features. Github (2023)
49. Wang, C.-Y., Yeh, I.-H., Mark Liao, H.-Y.: YOLOV9: learning what you want to learn using programmable gradient information. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision—ECCV 2024, pp. 1–21. Springer, Cham (2025)

50. Huang, Y., Qiu, C., Guo, Y., Wang, X., Yuan, K.: Surface defect saliency of magnetic tile. In: 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), pp. 612–617 (2018). https://doi.org/10.1109/COASE.2018.8560423
51. Weakly Supervised Learning for Industrial Optical Inspection. https://hci.iwr.uni-heidelberg.de/content/weakly-supervised-learning-industrial-optical-inspection (2017)

Authors and Affiliations

Qingqing Li[1,2,3,4] · Tianshu Wu[1,4] · Tingfa Xu[2] · Jianmei Lei[1,4] · Jiu Liu[5]

✉ Tianshu Wu
tshuwu@gmail.com

Qingqing Li
liqqoffice@163.com

Tingfa Xu
ciom_xtf1@bit.edu.cn

Jianmei Lei
leijianmei@caeri.com.cn

1 China Automotive Engineering Research Institute Co., Ltd., Chongqing 401122, China

2 School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China

3 State Key Laboratory of Intelligent Vehicle Safety Technology, Chongqing 130025, China

4 China Automobile Academy (Jiangsu) Automotive Engineering Research Institute Co., Ltd., Chongqing 401122, China

5 School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China