buildings

MDPI

Article

A Two-Stage Concrete Crack Segmentation Method Based on the Improved YOLOv11 and Segment Anything Model

Ru Zhang [1,2], Chaodong Guan [1], Yi Fang [1,2,*], Yuanfeng Duan [2] and Xiaodong Sui [1,2]

[1] Department of Civil Engineering, Hangzhou City University, Hangzhou 310015, China
[2] College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China
* Correspondence: yeefang@zju.edu.cn

Abstract

During long-term service, concrete structures are exposed to various adverse factors, which often lead to the formation of numerous surface cracks. These cracks pose serious threats to structural safety and durability. Therefore, accurately identifying crack characteristics is essential for evaluating the service performance of concrete structures. A two-stage concrete crack segmentation method is presented in this study. The crack is initially located by the improved YOLOv11 that integrates three novel modules, namely Multi-scale Edge Information Enhancement, Efficient-Detection, and P2-Level Feature Integration, to form the MEP-YOLOv11 model. Then, the detected region is taken as input prompts for Segment Anything Model (SAM) to achieve precise crack segmentation. This approach eliminates the need for manual prompting in SAM, enabling automatic crack feature identification. The average Accuracy, precision, and Intersection over Union (IoU) for crack segmentation are 95.98%, 92.60%, and 0.77, respectively. To further enhance the robustness of the two-stage segmentation method under non-uniform illumination conditions, a mask re-input strategy is introduced. The crack mask generated by SAM using bounding-box prompts is fed back into SAM to guide a second round of segmentation. Experimental results demonstrate that the improved method maintains high segmentation performance, with an average Accuracy of 92.38%, precision of 85.70%, and IoU of 0.64. Overall, the proposed method meets engineering requirements for high-precision and efficient crack detection and segmentation, showing strong potential for practical inspection tasks.

Keywords: crack segmentation; MEP-YOLOv11; segment anything model; concrete; non-uniform illumination

Check for updates

Academic Editor: Roberto Capozucca

Received: 14 January 2026
Revised: 4 February 2026
Accepted: 12 February 2026
Published: 14 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

In the course of long-term service, a variety of concrete structures tend to develop cracks due to adverse factors such as environmental corrosion, material aging, and natural disasters. The existence of cracks can accelerate material deterioration and significantly reduce the load-bearing capacity of the structure [1,2], thus increasing the risk of structural failure and even causing accidents [3]. Therefore, it is essential to conduct health monitoring and maintenance of concrete cracks. Conventional crack detection methods primarily depend on manually operated portable equipment, which often exhibit low efficiency and limited accessibility, making it challenging to guarantee the accuracy and reliability of the results. With the advancement of computer technology, image processing-based crack detection methods have been increasingly adopted [4–6]. However, classical image-processing pipelines require handcrafted preprocessing and sensitive parameter tuning,

and their performance often degrades under small cracks and non-uniform illumination—common conditions in real-world inspections [7].

The rise in deep learning has driven advancements in image processing technologies. Convolutional Neural Networks (CNNs) have shown significant advantages in crack detection due to their strong feature extraction capabilities, with methods mainly divided into bounding-box-based localization and pixel-level semantic segmentation [8–11]. The former is used for identification and classification, while the latter achieves precise pixel-level extraction. In localization research, models can be categorized into single-stage (e.g., YOLO series, which balance efficiency and accuracy) and two-stage (higher accuracy but slower speed) [5,12,13]. Among single-stage approaches, the YOLO family has been widely adopted due to its favorable balance between efficiency and accuracy. To improve YOLO-based crack detection under limited data or complex scenarios, researchers have proposed both training strategies and architectural enhancements. For example, Li et al. [14] proposed a transfer learning framework for road crack detection, reducing the training data required for YOLOv4 by 25% while improving detection accuracy by 26.66%. Wang et al. [15] introduced a Bidirectional Feature Pyramid Network (BiFPN) into the YOLOv5 architecture, which achieved only a 3.4% error between predicted and actual values in crack detection, with an average processing time of just 14.2 ms per image. With the advancement of research, increasing attention has been directed toward fine or small cracks, which are small in size and easily overlooked yet may indicate early-stage deterioration and thus require timely detection and treatment. To address this challenge, Hou et al. [16] incorporated the SimAM attention mechanism into the head network of the YOLOv5s model. The improved model achieved a 29.3% increase in average Accuracy for detecting small cracks on aircraft engine blades compared to the baseline model. Zhang et al. [17] introduced the multi-scale self-attention module, Swin Transformer, into the neck structure, significantly enhancing the model's ability to extract features of small target cracks. Despite these advances, concrete cracks remain particularly challenging for object detectors because they are typically small, elongated, and highly edge-dependent, often exhibiting low contrast and complex branching under cluttered textures, shadows, or stains. Under such conditions, detectors may miss subtle branches or confuse cracks with background patterns. Therefore, improving edge-sensitive multi-scale representations is crucial for robust crack localization.

In practical engineering applications, relying solely on crack localization and classification is insufficient to meet the requirements for assessing the load-bearing capacity of concrete structures. To obtain detailed geometric information about cracks, such as shape, orientation, and width distribution, corresponding crack segmentation methods are required. Although semantic segmentation networks like U-Net [18] and DeepLabv3+ [19] have achieved high accuracy, their training depends on large-scale, high-precision annotated datasets, and data re-annotation is still needed when detection conditions (e.g., lighting, shadows) change, limiting their practical deployment [20–23]. The foundational segmentation model, the Segment Anything Model (SAM), trained on over 1 billion masks and 11 million images, demonstrates strong generalization capabilities [24–26], offering the potential to reduce annotation costs and handle complex environments and lighting conditions. Huang et al. [27] validated SAM's performance on 53 medical datasets, showcasing its great potential in image segmentation. Carraro et al. [28] and Shan et al. [29] further demonstrated SAM's applicability in crop feature recognition and underwater rock segmentation, indicating that prompt-guided segmentation can improve efficiency while maintaining high accuracy. Teng et al. [30] proposed a concrete crack segmentation method based on SAM, using fractal dimension matrix prompts to effectively address environmental variations, achieving high segmentation accuracy. Preliminary studies sug-

gest that SAM's semi-supervised mode outperforms the fully automatic mode in crack identification. However, high-quality crack masks are often obtained in interactive settings that require manual prompts (e.g., clicks or boxes), which limit automation and throughput. In fully automated or weakly supervised modes, segmentation quality can degrade under complex backgrounds and especially under non-uniform illumination (e.g., shadows and overexposure), where ambiguous crack boundaries and texture interference reduce both accuracy and stability.

To achieve more real-time and accurate detection of concrete cracks, this paper proposes a two-stage concrete crack segmentation method by combining the strengths of YOLO and SAMs. The main contributions of this work are as follows:

(1) This study proposes an improved structure of the YOLOv11 model, which incorporates the novel Multi-scale Edge Information Enhancement (MSEIE), Efficient-Detection, and P2-level feature integration modules. The resulting model, named MEP-YOLOv11, significantly enhances the accuracy for crack localization.
(2) A two-stage crack segmentation method that combines the MEP-YOLOv11 model with the SAM is developed. The crack localization results generated by MEP-YOLOv11 are used as prompt information for SAM, enabling accurate crack segmentation without manual input and significantly enhancing the level of automation.
(3) A mask re-input method is proposed, where under non-uniform lighting conditions, a two-stage model can provide partial crack area localization information, enabling complete crack segmentation. This eliminates the need for retraining the neural network with images under different lighting conditions, thus reducing the reliance on expensive annotated datasets.

2. Methodology

The two-stage crack segmentation method proposed in this paper is primarily designed for visible open cracks on the surfaces of ordinary concrete bridges (such as bridge decks, beams, and piers). It demonstrates good performance in segmenting fine and branched cracks and can adapt to strong light, shadows, uneven lighting, and common background texture interference. The implementation process of the proposed two-stage concrete crack segmentation method is illustrated in Figure 1. First, the acquired crack images are fed into the MEP-YOLOv11 model to localize the crack regions. Then, the localized crack information is used as a bounding box prompt for the SAM to perform crack segmentation. To improve the segmentation accuracy under non-uniform illumination conditions, the crack mask output generated by the SAM using the initial bounding box prompt is re-fed into the model as a secondary input, guiding it to perform a refined segmentation of the crack regions.

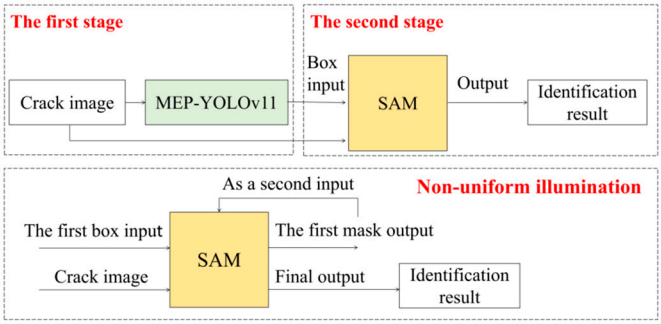


Figure 1. Implementation process of the two-stage crack segmentation method.

2.1. MEP-YOLOv11 Model-Based Crack Localization

YOLOv11 introduces several groundbreaking improvements in its network architecture compared to its predecessor models. In the backbone network, the model incorporates the C3K2 module, the C2PSA attention mechanism, and the Spatial Pyramid Pooling-Fast (SPPF) layer. The C3K2 module, with its core idea of branch processing and residual connections, along with convolution kernels of adjustable sizes, effectively extracts multi-scale feature information while maintaining the model's lightweight design. Compared to the previous version of the C3K module, the C3K2 module allows the selection of variable convolution kernels based on the feature extraction requirements at different scales, reducing unnecessary computational resource waste. The C2PSA module also employs branch processing and residual connections, combining the PSA module used in the backbone branch to achieve adaptive weighting of key regions. This reduces computational overhead while enhancing small object detection capabilities. In contrast to the single PSA module, which provides only a single-level spatial attention weighting, the C2PSA achieves collaborative optimization of local details and global context through cross-stage connections, gaining stronger multi-scale feature representation ability without significantly increasing the number of parameters [31]. The SPPF layer implements hierarchical progressive receptive field expansion in depth through repeated concatenated max-pooling operations and integrates feature dimensions in width via channel concatenation, thereby enhancing the model's ability to understand small objects and complex scenes. Compared to the classic SPP module, the SPPF replaces the multi-branch parallel structure with a serial recursive structure, achieving equivalent multi-scale features with less computational and memory overhead. This results in a significant increase in model inference speed and a simplified graph structure [32].

These modifications effectively reduce computational costs while significantly enhancing multi-scale feature fusion and robustness under complex scenarios. As a result, YOLOv11 achieves high accuracy and fast inference in conventional object detection tasks [33,34]. However, in the specific task of locating cracks in concrete bridges, the process heavily relies on edge information due to the small pixel area occupied by some cracks. This causes the YOLOv11 model to misclassify small, complex cracks as background textures, thereby limiting its applicability in this task. Therefore, this study improves the YOLOv11 architecture and proposes the MEP-YOLOv11 model, specifically designed for concrete crack detection and localization. The architecture of the model is shown in Figure 2. The improved model employs an enhanced C3K2 module with MSEIE (C3K2-MSEIE) in the backbone network to extract and retain richer edge features. Furthermore, the feature fusion layers (P2) in the neck are enhanced by upgrading the original three-scale detection to a four-scale detection structure. Finally, grouped convolution layers are introduced in the Efficient-Detection head to enhance local feature extraction while preserving inter-channel information exchange. With these improvements, the MEP-YOLOv11 model achieves accurate localization of slender and complex cracks in concrete structures.

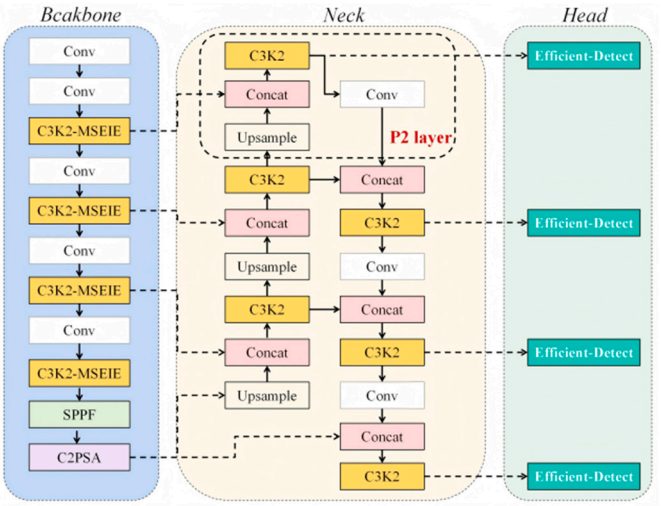


Figure 2. Architecture of the MEP-YOLOv11 model.

2.1.1. C3K2-MSEIE Module

Concrete cracks exhibit complex patterns and have low contrast with their background. Moreover, due to variations in shooting distance and angle, the size of cracks in the captured images can differ significantly. The original C3K2 module in the YOLOv11 model has limitations in multi-scale object detection and complex edge information extraction. This results in blurry crack edges in the generated feature maps, thereby affecting the model's detection performance and leading to problems such as missed detections and false positives. To address this issue, the MSEIE module is proposed, as shown in Figure 3. The module first divides the input feature map into four sub-feature maps at different scales using adaptive average pooling. Then, edge enhancement is applied to each sub-feature map. Finally, local features are concatenated with multi-scale outputs to improve the accuracy of crack edge detection. This method effectively enhances the model's recognition capability in complex scenarios.

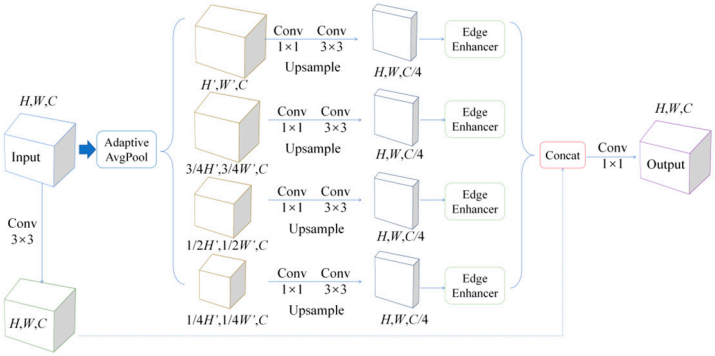


Figure 3. Structure of the C3K2-MSEIE module.

The core of the C3K2-MSEIE module is the adaptive average pooling operation and the edge enhancement submodule. The adaptive average pooling operation first fixes the output feature map size, and then adaptively determines the pooling window and stride based on the ratio between the input image size and the output image size. The input image is divided into several rectangular regions, and the average value of all elements in each sub-region is computed. After obtaining multi-scale feature maps, a 1×1 convolutional layer is used to reduce the number of channels to one-quarter of the original input, thereby improving computational efficiency. Finally, bilinear interpolation is used to restore high-

resolution image information, followed by preliminary edge enhancement of the multi-scale feature maps. The formula for adaptive pooling is as follows:

$$Y_{c,i,j} = \frac{1}{|R_{i,j}|} \sum_{(u,v)\in R_{i,j}} X_{c,u,v}, R_{i,j} \subseteq \{0,\ldots,H_{in}-1\} \times \{0,\ldots,W_{in}-1\} \quad (1)$$

$$R_{i,j} = \{(u,v)|u_{start}(i) \le u \le u_{end}(i), v_{start}(i) \le v \le v_{end}(i)\} \quad (2)$$

$$\begin{cases} u_{start}(i) = \left\lfloor \frac{i \cdot H_{in}}{H_{out}} \right\rfloor, \; u_{end}(i) = \left\lceil \frac{(i+1) \cdot H_{in}}{H_{out}} \right\rceil \\ v_{start}(j) = \left\lfloor \frac{i \cdot H_{in}}{H_{out}} \right\rfloor, \; v_{end}(j) = \left\lceil \frac{(i+1) \cdot H_{in}}{H_{out}} \right\rceil \end{cases} \quad (3)$$

$$\begin{aligned} s_h(i) &= k_h(i) = u_{end}(i) - u_{start}(i) \\ s_w(j) &= k_w(j) = v_{end}(j) - v_{start}(j) \end{aligned} \quad (4)$$

where $Y_{c,i,j}$ represents the average pooling output of the input image $R_{i,j}$ rectangular region, $|R_{i,j}|$ represents the number of pixels in the rectangular region $R_{i,j}$, $X_{c,i,j}$ represents the pixels in the rectangular region $R_{i,j}$ at the (u,v) position. $s_h(i) \times s_w(j)$ represents the size of the pooling window, while $k_h(i)$, $k_w(j)$ represent the strides of the pooling window in the vertical and horizontal directions, respectively.

To further enhance the recognition of concrete crack edge information, the multi-scale edge-processed feature maps are fed into the edge enhancement submodule, as shown in Figure 4. Crack images contain high-frequency edge information and low-frequency background information. Therefore, the feature image X undergoes a 3 × 3 average pooling operation to smooth high-frequency information while extracting low-frequency background information. Subsequently, the difference operation is applied to extract the high-frequency crack edge information from the image, emphasizing the edge features. For the binary classification problem between the crack edge region and the background region, a convolutional layer combined with a Sigmoid activation function is further applied to perform a nonlinear transformation on the feature map. This method compresses the pixel values of the crack edge regions, which are relatively large, towards 1, while compressing the pixel values of the background regions, which are relatively small, towards 0. This further enhances the contrast between the crack edges and the background, improving the saliency of the edge features. Finally, element-wise addition is performed to fuse the enhanced edge feature map with the original feature map X, strengthening the target region while retaining the original features. This allows the model to focus more on the crack edge details when processing the feature map. The formula for the edge enhancement submodule is as follows:

$$\begin{cases} X_{\mathrm{L}} = AvgPool2d_{3\times3}(X) \\ X_{\mathrm{H}} = X - X_{\mathrm{L}} \\ \sigma = \frac{1}{1+e^{-x}} \\ P = \sigma(C_{3\times3}(X_{\mathrm{H}})) \\ Y = X + P \end{cases} \quad (5)$$

In the equation, X represents the input feature map of the submodule, X_{L} denotes the low-frequency features, X_{H} represents the high-frequency features, σ is the Sigmoid activation function, $C_{3\times3}$ is a 3 × 3 convolution, P_{e} is the result after contrast enhancement of the edge regions, and Y is the result after edge information enhancement.

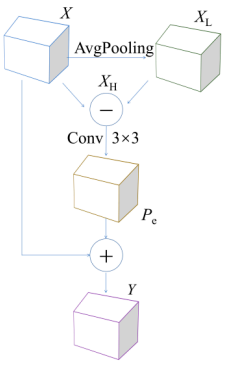


Figure 4. Structure of the edge enhancement sub-module.

2.1.2. The Improved Feature Fusion Layer

In the YOLO model architecture, the backbone network continuously downsamples the image from top to bottom, gradually reducing the scale of the feature maps. Although the final feature maps contain rich semantic information, spatial details are progressively lost during the downsampling process. To address this issue, the neck network recovers the feature map resolution through upsampling, as shown in Figure 5. The upsampled feature map is then concatenated along the channel dimension with shallow feature maps from the backbone network at the same scale, which contain rich spatial detail information. This enables precise localization and recognition of concrete cracks. The YOLOv11 base network contains only the P3, P4, and P5 layers, corresponding to feature maps at scales of 80×80, 40×40, and 20×20 for the original 640×640 image. To improve the model's ability to detect small concrete cracks, this paper adds additional upsampling and feature concatenation steps in the neck network, introducing a P2 level specifically for small object detection. The upsampled feature map at this level has a size of 160×160 and is concatenated with shallow feature maps from the backbone network that contain more detailed spatial information. This helps the model more accurately capture the subtle damage in bridge cracks.

The small-object layer P2 performs upsampling using bilinear interpolation. The bilinear interpolation method estimates the value of a target point by performing two one-dimensional linear interpolations, given the values of four adjacent known points. As shown in Figure 6, the target point R is located inside a rectangular region, which is defined by four adjacent known points: the top-left, top-right, bottom-left, and bottom-right points. First, linear interpolation is performed horizontally between the two points on the left and right, yielding two intermediate values. Then, a second linear interpolation is performed vertically on these two intermediate values to obtain the final estimated value for the target point, as shown in the following equation:

$$\begin{aligned} f(R_1) &= \frac{x-x_1}{x_2-x_1} f(P_4) + \frac{x_2-x}{x_2-x_1} f(P_3) \\ f(R_2) &= \frac{x-x_1}{x_2-x_1} f(P_2) + \frac{x_2-x}{x_2-x_1} f(P_1) \\ f(R) &= \frac{y-y_1}{y_2-y_1} f(R_2) + \frac{y_2-y}{y_2-y_1} f(R_1) \end{aligned} \tag{6}$$

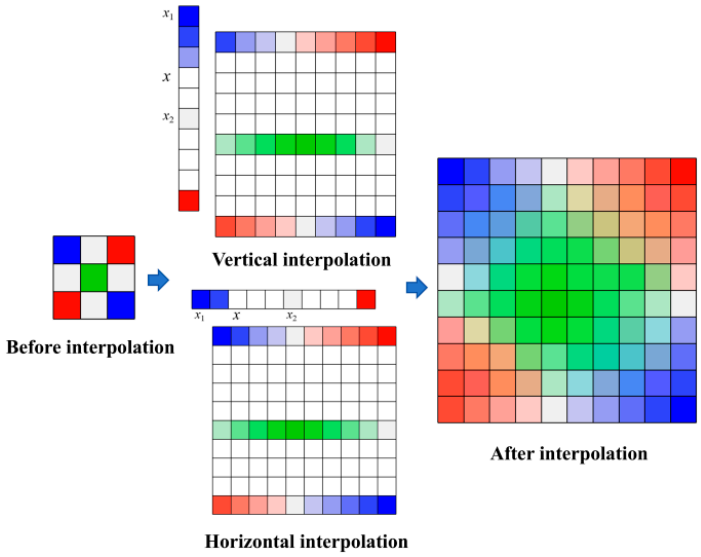


Figure 5. Upsampling schematic diagram.

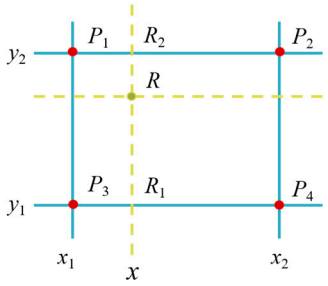


Figure 6. The schematic diagram of bilinear interpolation solution.

2.1.3. The Lightweight Efficient-Detection Module

In the original detection head of the YOLOv11 model, both depthwise convolution layers and standard convolution layers are used to adjust the feature channels, converting the number of input channels into the required number for predicting localization and classification losses. However, standard convolution layers compress information across all channels during channel adjustment, which makes the output susceptible to global features and leads to blurred edges and textures. In contrast, depthwise convolution layers apply separate kernels to each channel, which preserves individual channel features but cannot integrate information across channels, potentially leading to the loss of inter-channel correlations. To overcome the limitations of both convolution types in channel adjustment and to meet the real-time requirements for crack localization in concrete structures, a module called Efficient-Detection is proposed, as shown in Figure 7. It adopts a shared-parameter convolutional architecture that reduces the number of parameters compared with traditional models that rely on parallel processing. By applying group convolution, the input channels, output channels, and convolutional kernels are divided into g groups, with the number of parameters calculated as shown in Equation (7). As a result, the total parameters of the detection head are reduced to $1/g$ of the original, significantly improving detection efficiency and fulfilling the requirement for real-time crack damage detection. This group convolution method avoids the global feature interference seen in standard convolution and also prevents the lack of inter-channel integration associated with depthwise convolution.

$$\text{Number of parameters} = (h_1 \times w_1 \frac{C_{in}}{g}) \times \frac{C_{out}}{g} \times g \quad (7)$$

where h_1 and w_1 denote the height and width of the convolution kernel; C_{in} and C_{out} are the number of input and output channels, and g is the number of groups.

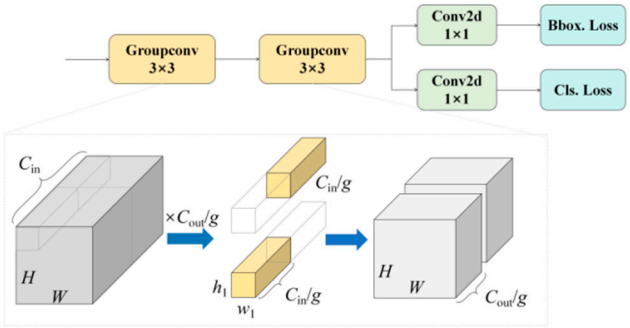


Figure 7. Structure of the lightweight Efficient-Detection module.

2.2. Crack Segmentation Using SAM

SAM is a general-purpose image segmentation foundation model developed by MATE AI, with pretraining conducted on the SA-1B dataset, which includes over one billion masks and 11 million images. Pretraining employs a multi-task learning strategy, combined with cross-entropy loss and Dice loss, to improve segmentation accuracy. At the same time, data augmentation techniques such as scaling, rotation, and random cropping are applied to enhance the model's robustness to various image transformations. This enables SAM to learn rich feature representations, and its zero-shot segmentation performance can rival that of fully supervised methods or even outperform them. The SAM consists of three core components: the image encoder, the prompt encoder, and the mask decoder. The image encoder in SAM uses Vision Transformer (ViT) as its backbone neural network. ViT directly models long-range dependencies, capturing global contextual information to improve the model's ability to understand complex scenes. The key components of the ViT backbone network include:

(1) Patch Embedding: The input image is divided into multiple fixed-size patches, and each patch is then flattened and linearly embedded into a high-dimensional space.

The input image $x \in R^{H \times W \times C}$ is divided into patches of size $P \times P$, and each patch is then embedded into a high-dimensional space through a linear transformation. The number of patches N and the patch embedding are given by the following formula:

$$N = \frac{H \times W}{P^2} \tag{8}$$

$$e_i = W_p P_i + b_p \tag{9}$$

where H and W represent the height and width of the image, C is the number of channels, e_i is the embedding vector after the linear transformation, Wp is the embedding weight, b_p is the bias term, and p_i denotes the i-th patch after segmentation.

(2) Transformer Encoder: The Transformer encoder in ViT processes the patch embeddings layer by layer using multi-head self-attention (MSA) and a feedforward network (FFN), enabling more efficient capture of both global and local image features. The operation on the patch embeddings at the l-th layer is as follows:

$$z_l = MSA(LN(z_{l-1})) + z_{l-1} \tag{10}$$

where LN represents layer normalization, and z_l denotes the output of the l-th layer.

(3) Position Embedding: Positional information is added to the patch embeddings to preserve the spatial relationships within the image. The combination of patch embed-

dings and position embeddings enhances the model's understanding of the overall structure of the image, enabling ViT to effectively extract rich visual features from the input image and perform various segmentation tasks more accurately.

$$z_0^i = e_i + E_{pos}^i,\ i = 1, 2, 3 \ldots N \tag{11}$$

where $E^i{}_{pos}$ represents the position embedding, which encodes the position of each patch in the original image.

The prompt encoder includes two types of prompts: sparse prompts (points, boxes, text) and dense prompts (masks), which guide the model's segmentation. Specifically, (1) point prompts are implemented by converting user-specified points into vectors, which not only represent the spatial location of the points but also indicate whether they are positive or negative samples. (2) Box prompts involve user-drawn bounding boxes, which are encoded to provide spatial constraints, helping the model determine the segmentation regions. (3) Text prompts leverage a pre-trained text model to convert the descriptive text provided by the user into feature vectors, enabling the model to understand the background information relevant to the segmentation task. (4) Mask prompts encode the initial segmentation masks provided by the user to refine or improve the existing segmentation results. These mask prompts are embedded into the model via convolution operations and added element-wise to the image embeddings, thereby enhancing the model's understanding of the segmentation task.

The mask decoder is responsible for integrating feature information from the ViT backbone network and the prompt encoder to generate precise segmentation masks. First, feature fusion is performed, where the image features extracted by the ViT backbone network are combined with the prompt features generated by the prompt encoder using an attention mechanism, ensuring that the final segmentation result accurately responds to the user's input prompts. Next, upsampling layers are used to gradually increase the resolution of the fused features until it matches the resolution of the original image. This step is crucial for generating high-resolution segmentation masks with rich details. Finally, a sigmoid function is applied to the upsampled output, converting the feature map into a binary mask that clearly defines the regions of the image to be segmented.

Currently, SAM offers two segmentation modes: fully automatic and semi-supervised modes. Examples of both modes applied to crack segmentation are shown in Figure 8. Currently, both the fully automatic and semi-supervised modes have limitations in terms of accuracy and efficiency, and neither fully meets the practical requirements of crack detection in engineering applications. In the fully automatic mode, the model segments all objects in the input image without any human intervention, but the segmentation accuracy is relatively low. In the semi-supervised mode, the user can guide the model to segment specific regions more accurately by providing four types of input prompts (clicks, bounding boxes, masks, and text). However, these four input methods currently require manual interaction, which limits detection efficiency. Therefore, this study proposes a novel crack segmentation method based on automatic bounding box input.

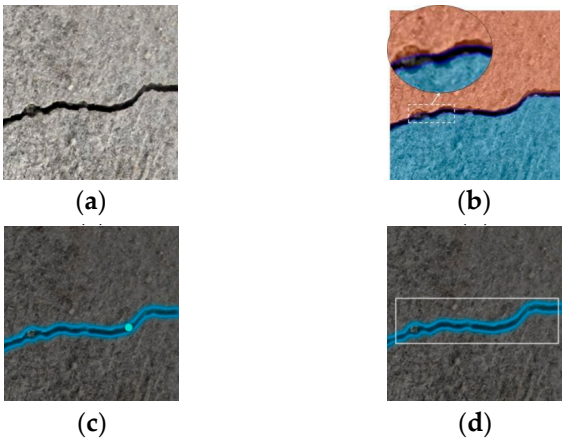


Figure 8. Crack segmentation results of SAM using different input modes. (**a**) Original images, (**b**) automatic segmentation, (**c**) point prompt segmentation, (**d**) box prompt segmentation.

2.2.1. Crack Segmentation Based on Automatic Bounding Box Input

The MEP-YOLOv11 model accurately localizes crack regions, and the bounding boxes generated by this model are used as bounding-box prompt inputs for the SAM, enabling fully automatic and high-precision crack segmentation. The detailed implementation process is shown in Figure 9. The method consists of the following four steps: (1) The crack image to be identified is fed into the fully trained MEP-YOLOv11 model to obtain the bounding box image and coordinate information of the crack region. (2) The center-format coordinates (cx, cy, w, h) output by the MEP-YOLOv11 model are converted into the two-point format (x_1, y_1, x_2, y_2) required by the SAM, which is then used as bounding-box prompts. The conversion formula is as follows:

$$\begin{cases} x_1 = (cx - w/2) \times w_o \\ y_1 = (cy - h/2) \times h_o \\ x_2 = (cx + w/2) \times w_o \\ y_2 = (cy + h/2) \times h_o \end{cases} \tag{12}$$

where (x_1, y_1) represents the coordinates of the top-left corner of the bounding box, and (x_2, y_2) represents the coordinates of the bottom-right corner. (cx, cy) denotes the normalized center coordinates of the box, while w and h represent its normalized width and height. wo and ho refer to the width and height of the input image. (3) The original image is fed into the ViT image encoder to extract image embeddings, while the converted two-point coordinates are input into the prompt encoder to generate the corresponding bounding-box prompt embeddings. (4) The mask decoder uses both the image embeddings and the bounding-box prompt embeddings to guide the model in generating the final crack segmentation mask.

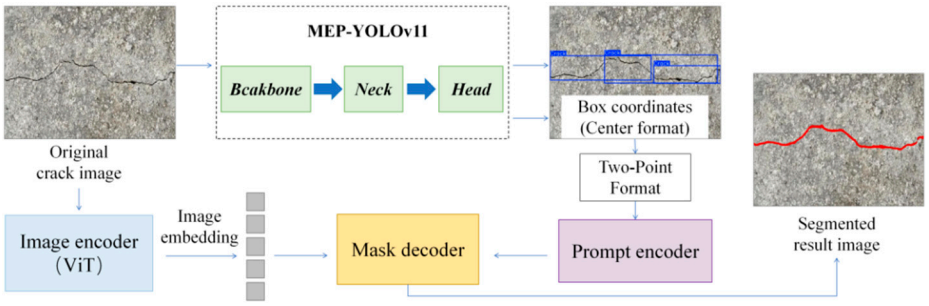


Figure 9. Crack segmentation process based on automatic bounding box input.

2.2.2. Failure Modes of the Crack Segmentation Method Based on Automatic Bounding Box Input

SAM performs segmentation based on bounding box input, which inherently imposes a spatial constraint, restricting the model to operate only within the specified region. Therefore, if the MEP-YOLOv11 model fails to accurately and fully locate the actual crack positions, the subsequent segmentation process by SAM will be limited, making it difficult to generate effective crack masks. The improved MEP-YOLOv11 model can accurately locate long and complex cracks in concrete, but when faced with untrained complex backgrounds, overexposure, shadows, and non-uniform illumination, issues such as missed detections and missed bounding boxes still occur, as shown in Figure 10. When the image as a whole suffers from overexposure and shadows, common image preprocessing methods such as exposure compensation and shadow removal can effectively improve the image quality, meeting the localization requirements of the MEP-YOLOv11 model, as shown in Figure 11. In normal scenarios, duplicate bounding boxes may also appear, but due to the strong segmentation capabilities of the SAM, duplicate bounding boxes have minimal impact on crack segmentation, as shown in Figure 12. Additionally, when inputting bounding box information into SAM, duplicate boxes with an overlap greater than 90% are removed, ensuring that the generated duplicate bounding boxes do not affect the segmentation results. However, when the model faces untrained complex backgrounds and non-uniform illumination, relying solely on image preprocessing cannot comprehensively improve the overall image quality or solve the issues of missed and false detections.

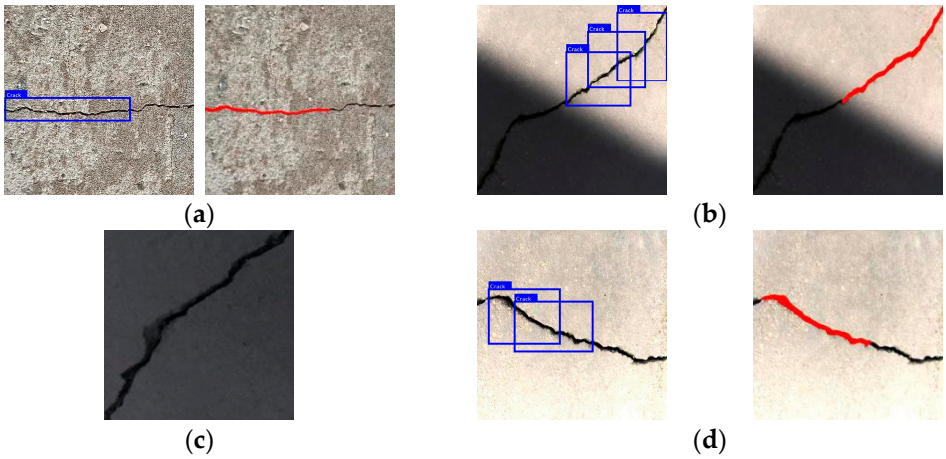


Figure 10. Failure scenarios of MEP-YOLOv11 model localization. (**a**) Untrained complex backgrounds (missed bounding boxes), (**b**) non-uniform illumination (missed bounding boxes), (**c**) shadows (missed detections), (**d**) overexposure (missed bounding boxes).



Figure 11. Model localization results after image preprocessing. (**a**) After shadow preprocessing. (**b**) After overexposure preprocessing.



Figure 12. Segmentation results in the case of duplicate bounding boxes.

2.2.3. Crack Segmentation Under Non-Uniform Illumination Conditions

Complex backgrounds often contain substantial high-frequency noise, which interferes with the model's ability to locate concrete crack edges. Under non-uniform illumination conditions, crack images often experience local overexposure and shadow occlusion, resulting in significant differences in brightness and contrast across different regions of the image [35]. This uneven lighting not only severely disrupts the model's ability to extract and understand features, but also leads to problems such as information loss, blurred edges, and shadow misguidance, thereby reducing the accuracy of crack segmentation. Therefore, when the MEP-YOLOv11 model faces untrained complex backgrounds and non-uniform lighting, if complete missed detection occurs, the training dataset needs to be expanded, and the model should be retrained. However, if the YOLO model can still recognize partial crack regions under untrained complex backgrounds and non-uniform lighting, the crack segmentation task can be completed using the mask re-input method proposed in this study. Crack segmentation under non-uniform lighting is much more challenging than in complex backgrounds, so this study uses non-uniform lighting conditions as an example to detail the process of the mask re-input method, as shown in Figure 13. First, the partial crack regions detected by the MEP-YOLOv11 model under non-uniform lighting are input into SAM's prompt encoder, while the original image is input into the image encoder, generating image embeddings and bounding box prompt embeddings, respectively. Next, a lightweight mask decoder is used to generate the initial crack mask, providing a preliminary segmentation of the corresponding crack regions. Then, the initial mask is used as a new prompt input to the prompt encoder, generating the mask prompt embedding. After combining this with the image embedding, it is input again into SAM to complete the full crack segmentation. This method fully utilizes the strong shape priors embedded in SAM's mask prompt inputs. Even if these inputs are incomplete (e.g., only covering part of the target), they still contain structural cues such as continuity, edge direction, and contour features, effectively constraining the model's output range and guiding it to complete the overall shape of the target, achieving a more comprehensive segmentation result.

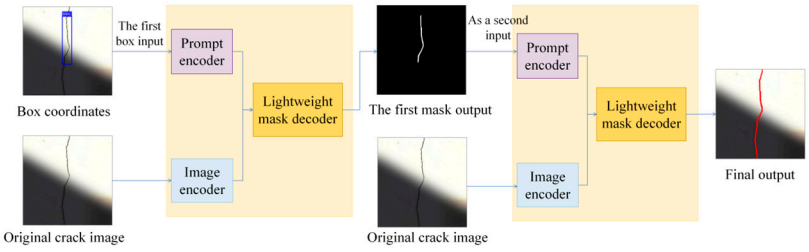


Figure 13. Crack segmentation process under non-uniform illumination conditions.

2.3. Evaluation Indicator

2.3.1. Evaluation Indicators for Localization Task

Precision (P_L), Recall (R), F1-Score ($F1$), and Average Precision (AP) are selected as evaluation metrics to assess the crack localization performance of the improved MEP-YOLOv11 model. The calculation formulas for each metric are presented as follows:

$$P_L = \frac{TP_L}{TP_L + FP_L} \tag{13}$$

$$R = \frac{TP_L}{TP_L + FN_L} \tag{14}$$

$$F1 = 2 \times \frac{P_L \times R}{P_L + R} \tag{15}$$

$$AP = \int_0^1 P_L(R)\mathrm{d}R \tag{16}$$

where TP_L, FP_L and FN_L are obtained by matching predicted bounding boxes with ground-truth boxes. Specifically, TP_L refers to the number of matched predicted boxes, FP_L is the number of unmatched predicted boxes, and FN_L is the number of missed ground-truth boxes.

2.3.2. Evaluation Indicators for Segmentation Task

Pixel-level Precision (P_S), IoU, and Accuracy are adopted to comprehensively evaluate the crack segmentation performance of the proposed two-stage model based on MEP-YOLOv11 and SAM. The metrics are defined as

$$P_S = \frac{TP_S}{TP_S + FP_S} \tag{17}$$

$$\mathrm{IoU} = \frac{\mathrm{area}(A_p \cap A_r)}{\mathrm{area}(A_p \cup A_r)} \tag{18}$$

$$\mathrm{Accuracy} = \frac{1}{2}\left(\frac{TP_S}{TP_S + FN_S} + \frac{TN_S}{TN_S + FP_S}\right) \tag{19}$$

where TP_S, TN_S, FP_S, FN_S are computed by pixel-wise comparison between the predicted mask and the ground-truth mask: TP_S refers to the number of crack pixels correctly predicted by the model as cracks, TN_S is the number of non-crack pixels correctly identified as non-cracks, FP_S is the number of non-crack pixels incorrectly predicted as cracks, and FN_S is the number of crack pixels incorrectly predicted as non-cracks. A_p represents the predicted region by the model, and A_r represents the true crack region.

3. Results and Discussion

3.1. Experimental Environment and Dataset

Both the MEP-YOLOv11 and SAM are executed under the same hardware and software environment. The hardware includes an Intel(R) Core(TM) i9-9900K @ 3.60GHz CPU, an NVIDIA GeForce RTX 4090 GPU, and 24 GB of RAM. The operating system is Windows 11, with Python 3.10.16 as the interpreter. The deep learning framework is PyTorch 2.2 with CUDA version 12.1. A total of 2108 concrete bridge crack images are collected from the public datasets Crack500, DeepCrack, and Concrete Crack Dataset, some of which are shown in Figure 14. The images are aggregated from multiple public datasets rather than using a single benchmark, because a single dataset is often insufficient in scale for training the modified detector reliably, and multi-source aggregation increases diversity in crack morphology, texture, and illumination. This helps reduce overfitting to one data

distribution and improves robustness when deploying the model to real inspection images. These images are used to train the MEP-YOLOv11 model, with each image having a resolution of 640 × 640 pixels. The dataset is randomly split into training (1688 images), validation (210 images), and test sets (210 images) in a ratio of 8:1:1. The detailed training parameters of the MEP-YOLOv11 model are listed in Table 1. The average precision at a threshold of 0.5 (*AP*50) on the validation set was used as the monitoring metric. If there was no significant improvement in *AP*50 for 10 consecutive epochs, training was stopped early. After comprehensive consideration of experimental tuning and the early stopping mechanism, the number of training epochs was finally set to 120. In training the MEP-YOLOv11 model, the initial learning rate is set to 0.01 and the stochastic gradient descent (SGD) optimizer with momentum is employed, where the momentum coefficient is set to 0.937. Momentum introduces "inertia" to make parameter updates not only respond to the current gradient but also retain 93.7% of the historical direction, preventing oscillation and divergence at high learning rates [36,37].



Figure 14. Some of the training samples.

Table 1. Training parameters of the MEP-YOLOv11 model.

Parameter	Value
Epoch	120
Batch size	32
Optimize	SGD
Learning rate	0.01
Momentum	0.937

3.2. Crack Localization Results

Ablation studies are conducted to investigate the sources of performance improvement and to understand the role and contribution of each component within the model. For the improved MEP-YOLOv11 model, the ablation study controls other variables to focus on the impact of the introduced C3K2-MSEIE, P2 layer, and Efficient-Detection on model prediction accuracy. This helps avoid blind modifications that may increase computational complexity or lead to overfitting. The results of the ablation study are shown in Table 2.

Table 2. Comparison results of modules ablation experiment.

Base	+C3k2-MSEIE	+P2	+Efficient-Detection	P_L	R	$F1$	$AP50$
√				0.5885	0.5924	0.5904	0.6049
√	√			0.5770	0.5995	0.5880	0.6120
√		√		0.5738	0.6093	0.5911	0.6061
√			√	0.6038	0.5902	0.5970	0.6229
√	√	√		0.6438	0.5680	0.6035	0.6353
√	√	√	√	0.5887	0.6391	0.6129	0.6362

As shown in Table 2, introducing the C3K2-MSEIE module, the P2 layer, or the Efficient-Detection module individually leads to varying degrees of improvement in $AP50$ compared to the baseline model. In terms of key metrics such as P_L, R, and $F1$, each module demonstrates different optimization tendencies: Specifically, the C3K2-MSEIE module and the P2 layer primarily contribute to improved recall, while the Efficient-Detection module mainly enhances precision. When the modules are combined, their synergistic effects further enhance the overall performance of the model. Therefore, with all proposed modules integrated, the model achieves comprehensive improvements across all evaluation metrics compared to the baseline. Specifically, P_L increases by 0.03%, R improves by 7.9%, $F1$ rises by 3.8%, and $AP50$ increases by 5.2%. These results strongly validate the effectiveness of the proposed improvements and module designs in enhancing crack localization accuracy for concrete structures.

3.3. Crack Segmentation Results

Figure 15 presents several representative examples of segmented concrete cracks. Four selected crack images are chosen for their diversity: the first shows a transverse crack with fine branches; the second is an inclined crack; the third is a slender and curved crack; and the fourth is a relatively straight vertical crack. The results demonstrate that the proposed two-stage segmentation approach, based on MEP-YOLOv11 and SAM, effectively handles various types of crack morphologies in concrete structures. To evaluate segmentation accuracy, 60 samples are selected for assessment. Manually annotated masks are used as ground-truth labels, while outputs from the two-stage model serve as predicted labels. Accuracy, P_S, and IoU are used as evaluation metrics. Experimental results show that the proposed method achieves an average Accuracy of 95.98%, an average P_S of 92.6%, and an average IoU of 0.77 on the test set.

To highlight the superiority of the proposed two-stage model, a comparative analysis is conducted against commonly used semantic segmentation models using the same evaluation metrics. Table 3 presents a performance comparison between the proposed method and DeepLabv3+, SegNet, and U-Net on the task of crack segmentation. The two-stage model does not require manually annotated segmentation datasets for training, and segmentation can be achieved using only the bounding boxes automatically generated by MEP-YOLOv11 as guidance. In contrast, the comparative semantic segmentation model requires training on segmentation datasets, which are costly to obtain. Therefore, we use a segmentation dataset of 100 images for transfer learning to train the comparative model. In terms of average Accuracy, the proposed two-stage model achieves the highest performance with a value of 95.98%, significantly outperforming other methods. DeepLabv3+ ranks second with an average Accuracy of 90.53%, while SegNet and U-Net show relatively weaker performance at 71.37% and 78.46%, respectively. For average P_S, the performance trend remains consistent with that of average Accuracy. The two-stage model again achieves the highest P_S of 92.60%, further confirming its reliability in crack segmentation. In terms of average IoU, both the proposed method and DeepLabv3+ achieve a score of 0.77, indicating

comparable segmentation performance. By contrast, SegNet and U-Net perform considerably worse, with IoU scores of 0.43 and 0.56, respectively. Figure 16 illustrates segmentation examples produced by each model. The results demonstrate that the two-stage model performs excellently in crack segmentation, producing detailed and clean crack boundaries without noise interference, thus reflecting high accuracy and robustness.

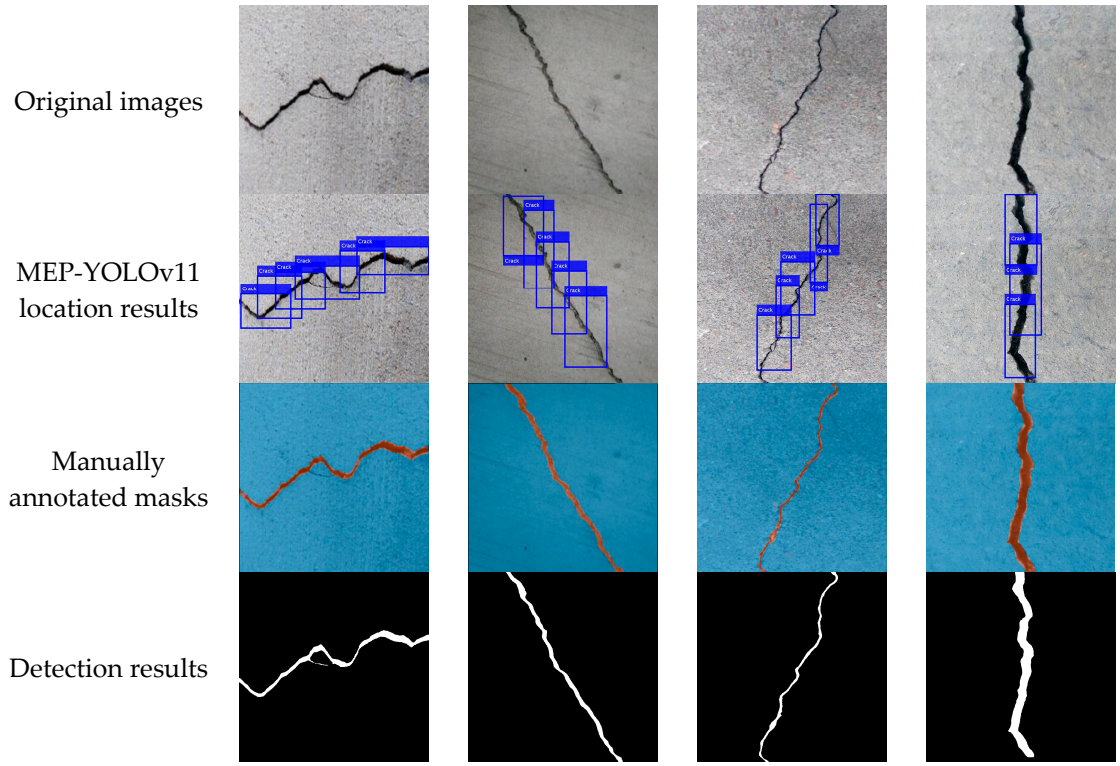


Figure 15. Example of crack segmentation process based on the two-stage model.

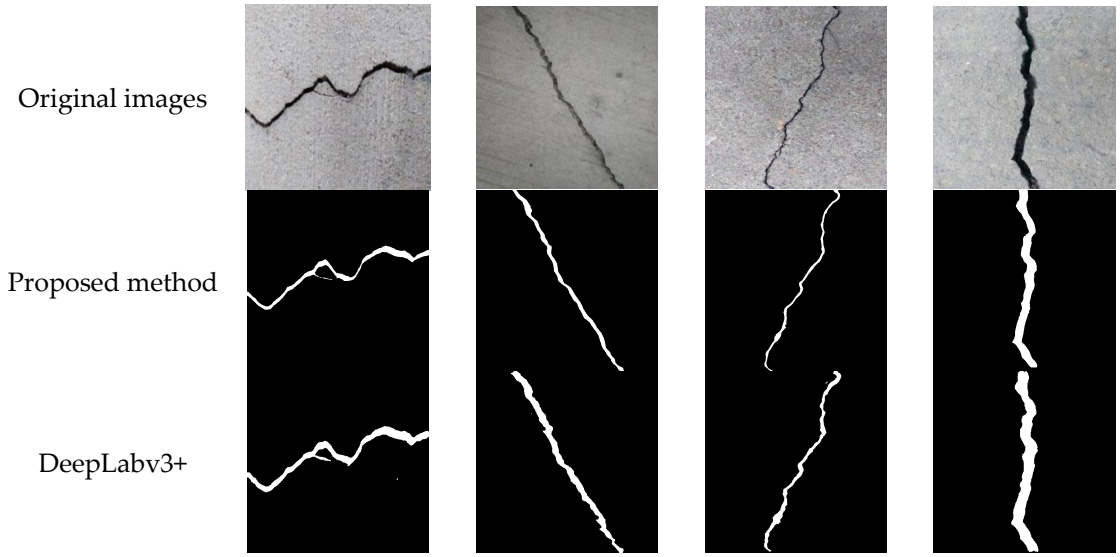


Figure 16. *Cont.*

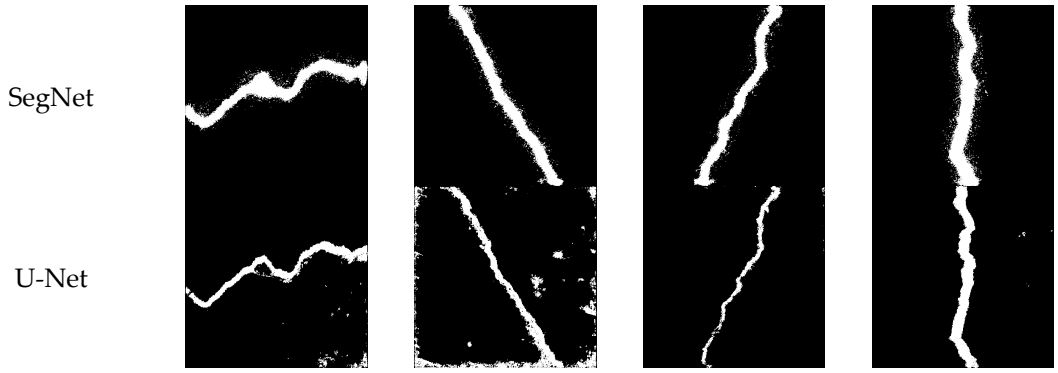


Figure 16. Crack segmentation results using different deep learning methods.

Table 3. Comparison of crack segmentation results among different methods.

Method	Evaluation Indicator		
	Average Accuracy	Average P_S	Average IoU
Proposed method	95.98%	92.60%	0.77
DeepLabv3+	90.53%	81.13%	0.77
SegNet	71.37%	42.75%	0.43
U-Net	78.46%	57.04%	0.56

3.4. Crack Segmentation Results Under Non-Uniform Illumination

Figure 17 shows the segmentation results under non-uniform illumination conditions and compares them with those of DeepLabv3+, the best-performing model from Section 3.3. The two-stage model that relies solely on box-based prompts is constrained by the crack localization performance of the MEP-YOLOv11 model. Under shadowed and artifact-prone areas, it struggles to provide accurate crack location information, resulting in incomplete boundaries or missed detections. DeepLabv3+, lacking exposure to samples with non-uniform illumination during training, exhibits poor adaptability to lighting variations. It tends to produce significant misclassifications in shadowed regions, often falsely identifying large shadow areas as cracks. In contrast, the mask re-input method uses the mask output generated from the initial box prompt as the second input to SAM. This approach enables complete crack segmentation without requiring the model to be retrained on images captured under non-uniform illumination conditions, and it avoids both missed and false detections.

The recognition accuracy under non-uniform illumination conditions is shown in Table 4, where "Without mask re-input" denotes the baseline two-stage inference using MEP-YOLOv11 box prompts and one-pass SAM, whereas "With mask re-input" denotes the second-pass SAM refinement guided by the initial mask prompt. The results indicate that without the use of mask re-input, the model's average Accuracy, average P_S, and average IoU significantly decrease compared to normal lighting conditions, failing to meet the practical requirements of crack recognition. Particularly, the DeepLabv3+ model experiences a steep performance decline under non-uniform illumination conditions. After introducing mask re-input, the model achieves an average Accuracy of 92.38%, an average P_S of 85.70%, and an average IoU of 0.64 under non-uniform illumination conditions. Compared to the scenario without this method, the average Accuracy improves by 3.02%, the average P_S increases by 5.02%, and the average IoU improves by 0.24.

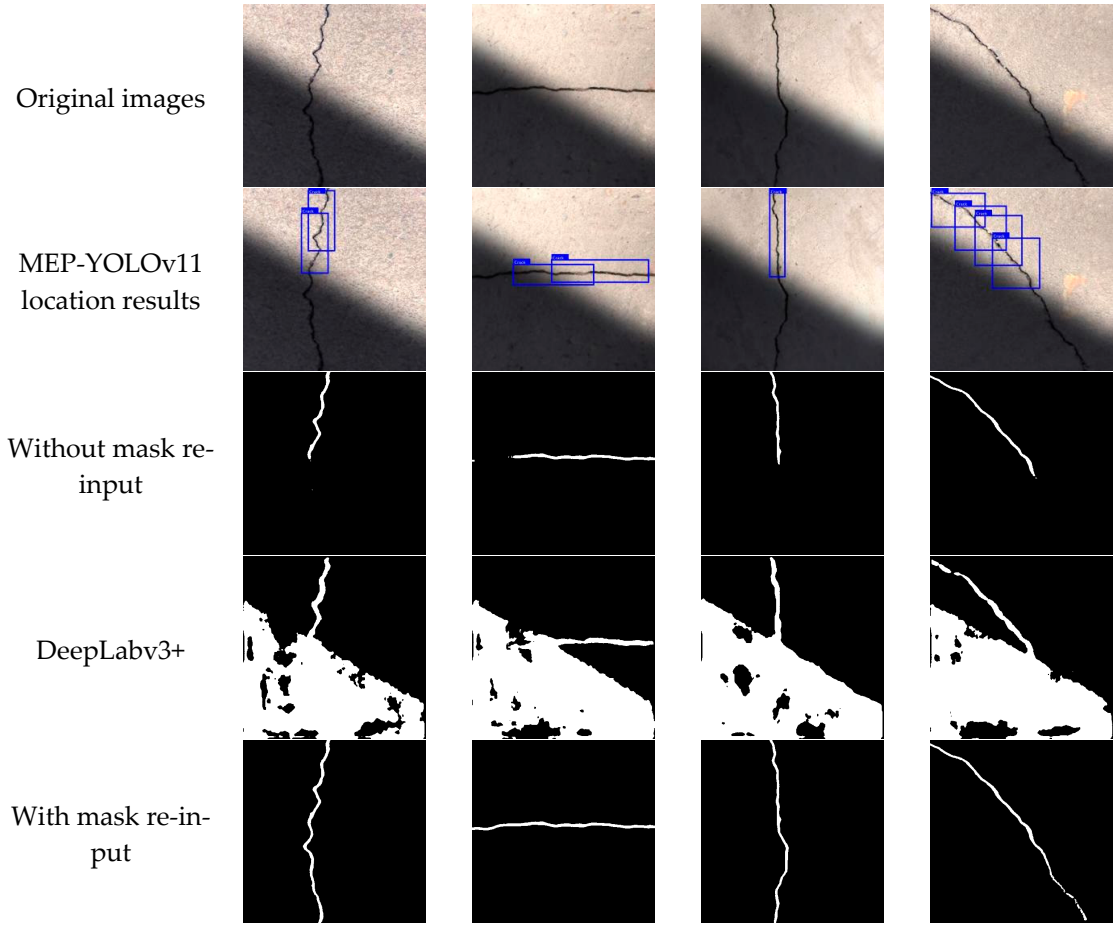


Figure 17. Examples of segmentation results from different methods under non-uniform illumination conditions.

Table 4. Comparison of segmentation results under non-uniform illumination conditions.

Method	Evaluation Indicator		
	Average Accuracy	Average P_S	Average IoU
Without mask re-input	89.36%	80.68%	0.40
With mask re-input	92.38%	85.70%	0.64
DeepLabv3+	53.13%	6.92%	0.07

Beyond the reported metric improvements, this study provides three practical findings regarding the application of foundation models in civil engineering. First, in a detector-to-foundation-model pipeline, the quality of detector-generated prompts (especially the recall on thin and branched cracks) directly governs the completeness of the final crack masks. This mechanism explains why the proposed MEP-YOLOv11, with its edge-sensitive multi-scale localization, is critical for benefiting downstream segmentation. Second, the proposed mask re-input refinement offers an efficient strategy to recover missed crack regions under non-uniform illumination, whereas a single-pass prompt-based segmentation often produces fragmented masks. Third, compared with fully supervised semantic segmentation pipelines, the proposed workflow reduces the dependency on costly pixel-wise annotations and eliminates manual prompting during inference, making it highly suitable for automated inspection scenarios (e.g., mobile platforms, robots, or UAV-based surveys). Furthermore, once a scale reference or camera calibration is available, the obtained masks

can support downstream crack characterization (e.g., crack trace continuity, length, and width distribution) for maintenance decision-making.

4. Conclusions

In this paper, a novel two-stage concrete crack segmentation method combining the improved MEP-YOLOv11 and the Segment Anything Model (SAM) is proposed to address the challenges of precise crack detection in complex engineering environments. The method firstly uses an improved MEP-YOLOv11 model to achieve more accurate crack localization and then uses the extracted positional information as prompts to guide the SAM in performing pixel-level segmentation of the crack regions. MEP-YOLOv11 automatically generates high-quality prompts, effectively overcoming SAM's heavy reliance on manual input and enabling full automation of the crack segmentation process. Compared with traditional semantic segmentation methods, this approach eliminates the need for large-scale and costly pixel-level annotated datasets, significantly reducing the time and resources required for model development and deployment while maintaining competitive segmentation accuracy. To improve the model's practicality under challenging field conditions—particularly non-uniform illumination—a mask re-input strategy is further introduced. This strategy leverages the complementary strengths of the two-stage model and enhances its robustness without additional training costs. Based on the research findings, the following conclusions can be drawn:

(1) The improved MEP-YOLOv11 model achieves comprehensive performance enhancement in the crack localization compared with the baseline model, with P_L, R, $F1$, and $AP50$ increasing by 0.03%, 7.9%, 3.8%, and 5.2%, respectively.
(2) The proposed two-stage model based on MEP-YOLOv11 and SAM performs excellently in the crack segmentation under normal lighting conditions, achieving an average Accuracy of 95.98%, average P_S of 92.60%, and average IoU of 0.77.
(3) By employing the mask re-input method, the two-stage model maintains reliable segmentation performance under non-uniform illumination conditions, with average Accuracy, average P_S, and average IoU of 92.38%, 85.70%, and 0.64, respectively.

Although the two-stage concrete crack segmentation method proposed in this paper demonstrates good performance in the segmentation of fine and branched cracks and can adapt to uneven lighting and common background texture interference, it still has the following limitations: First, it has not been verified whether the MEP-YOLOv11 model can still perform effectively on surfaces with excessively pronounced curvature, overly complex textures causing overlap and mixing with the crack's linear features, or situations where the background color and grayscale are similar to the crack's height. When the YOLO model cannot accurately locate cracks, SAM also struggles to perform effective segmentation. Although the mask re-input method can alleviate some false negatives, completely missed detections remain unresolved. Secondly, although the mask re-input method significantly improves segmentation performance under uneven lighting, performance metrics still show a notable decline compared to normal lighting conditions, indicating that lighting interference remains a major challenge in practical engineering applications. Future research will focus on the following directions: collecting and annotating more diverse crack images that reflect real-world engineering conditions, covering various levels of stain coverage, weather impacts, and combinations of different surface defects, to more comprehensively validate the robustness of the two-stage model. Future work should focus on improving the model's robustness, real-time performance, and generalization ability.

Author Contributions: Conceptualization, R.Z.; methodology, C.G.; software, C.G.; validation, C.G.; formal analysis, C.G.; investigation, C.G.; resources, R.Z. and Y.D.; data curation, C.G.; writing—

original draft preparation, C.G. and Y.F.; writing—review and editing, R.Z., C.G., Y.F. and X.S.; visualization, C.G.; supervision, R.Z., C.G. and Y.F.; project administration, R.Z.; funding acquisition, R.Z., Y.D. and X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Fund for National Natural Science Foundation of China (52578382, U24A20169), the Zhejiang Provincial Natural Science Foundation (LMS26E080033), and the Graduate Education Society of Zhejiang Province (2024-020).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Prasanna, P.; Dana, K.J.; Gucunski, N.; Basily, B.B.; La, H.M.; Lim, R.S.; Parvardeh, H. Automated crack detection on concrete bridges. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 591–599. [CrossRef]
2. Kompanets, A.; Duits, R.; Pai, G.T.; Leonetti, D.; Snijder, H.H. Loss function inversion for improved crack segmentation in steel bridges using a CNN framework. *Autom. Constr.* **2025**, *170*, 105896. [CrossRef]
3. Zhang, J.; Qian, S.R.; Tan, C. Automated bridge surface crack detection and segmentation using computer vision-based deep learning model. *Eng. Appl. Artif. Intel.* **2022**, *115*, 105225. [CrossRef]
4. Kheradmandi, N.; Mehranfar, V. A critical review and comparative study on image segmentation-based techniques for pavement crack detection. *Constr. Build. Mater.* **2022**, *321*, 126162. [CrossRef]
5. An, Y.H.; Kong, L.X.; Hou, C.C.; Ou, J.P. An adaptable rotated bounding box method for automatic detection of arbitrary-oriented cracks. *Struct. Health Monit.* **2024**, *23*, 3312–3335. [CrossRef]
6. Dorafshan, S.; Thomas, R.J.; Maguire, M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Constr. Build. Mater.* **2018**, *186*, 1031–1045. [CrossRef]
7. Ye, X.J.; Luo, K.H.; Wang, H.M.; Zhao, Y.H.; Zhang, J.W.; Liu, A.R. An advanced AI-based lightweight two-stage underwater structural damage detection model. *Adv. Eng. Inform.* **2024**, *62*, 102553. [CrossRef]
8. Kim, B.; Cho, S. Image-based concrete crack assessment using mask and region-based convolutional neural network. *Struct. Control Health Monit.* **2019**, *26*, e2381. [CrossRef]
9. Xiong, B.; Hong, R.; Wang, J.X.; Li, W.; Zhang, J.; Lv, S.T.; Ge, D.D. DefNet: A multi-scale dual-encoding fusion network aggregating Transformer and CNN for crack segmentation. *Constr. Build. Mater.* **2024**, *448*, 138206. [CrossRef]
10. Deng, J.H.; Hua, L.X.; Lu, Y.; Song, Y.; Singh, A.; Che, J.; Li, Y. Crack analysis of tall concrete wind towers using an ad-hoc deep multiscale encoder-decoder with depth separable convolutions under severely imbalanced data. *Struct. Health Monit.* **2025**, *24*, 3561–3579. [CrossRef]
11. Meng, S.Q.; Zhou, Y.; Jafari, A. Automatic high-precision crack detection of post-earthquake structure based on self-supervised transfer learning method and SegCrackFormer. *Struct. Health Monit.* **2024**, *23*, 3352–3370. [CrossRef]
12. Li, R.X.; Yu, J.Y.; Li, F.; Yang, R.T.; Wang, Y.D.; Peng, Z.H. Automatic bridge crack detection using Unmanned aerial vehicle and Faster R-CNN. *Constr. Build. Mater.* **2023**, *362*, 129659. [CrossRef]
13. Matarneh, S.; Elghaish, F.; Rahimian, F.P.; Abdellatef, E.; Abrishami, S. Evaluation and optimisation of pre-trained CNN models for asphalt pavement crack detection and classification. *Autom. Constr.* **2023**, *160*, 105297. [CrossRef]
14. Li, Y.S.; Che, P.Y.; Liu, C.L.; Wu, D.F.; Du, Y.C. Cross-scene pavement distress detection by a novel transfer learning framework. *Comput.-Aided Civ. Infrastruct. Eng.* **2021**, *36*, 1398–1415. [CrossRef]
15. Wang, S.K.; Dong, Q.; Chen, X.Q.; Chu, Z.P.; Li, R.Q.; Hu, J.; Gu, X.Y. Measurement of asphalt pavement crack length using YOLO V5-BiFPN. *J. Infrastruct. Syst.* **2024**, *30*, 04024005. [CrossRef]
16. Hou, X.Y.; Zeng, H.; Jia, L.; Peng, J.B.; Wang, W.X. MobGSim-YOLO: Mobile device terminal-based crack hole detection model for aero-engine blades. *Aerospace* **2024**, *11*, 676. [CrossRef]
17. Zhang, Y.X.; Lu, Y.; Huo, Z.J.; Li, J.L.; Sun, Y.R.; Huang, H. USSC-YOLO: Enhanced multi-scale road crack object detection algorithm for UAV image. *Sensors* **2024**, *24*, 5586. [CrossRef]
18. Liu, D.; Xu, M.D.; Li, Z.T.; He, Y.Y.; Zheng, L.; Xue, P.P.; Wu, X.D. A multi-scale residual encoding network for concrete crack segmentation. *J. Intell. Fuzzy Syst.* **2024**, *46*, 1379–1392. [CrossRef]
19. Fu, H.X.; Meng, D.; Li, W.H.; Wang, Y.C. Bridge crack semantic segmentation based on improved Deeplabv3+. *J. Mar. Sci. Eng.* **2021**, *9*, 671. [CrossRef]
20. Giulietti, N.; Revel, G.M.; Chiariotti, P. Automated vision-based concrete crack measurement system. *Measurement* **2025**, *242*, 115858. [CrossRef]

21. Al-Huda, Z.; Peng, B.; Algburi, R.N.A.; Al-antari, M.A.; AL-Jarazi, R.; Al-maqtari, O.; Zhai, D.H. Asymmetric dual-decoder-U-Net for pavement crack semantic segmentation. *Autom. Constr.* **2023**, *156*, 105138. [CrossRef]
22. Zhang, R.J.; Shi, L.; Cui, D.Y.; Wu, Y.F.; Ji, Z.L. GTRS-Net: A lightweight visual segmentation model for road crack detection. *Signal Image Video Process.* **2025**, *19*, 615. [CrossRef]
23. Zoubir, H.; Rguig, M.; El Aroussi, M.; Saadane, R.; Chehri, A. Pixel-level concrete bridge crack detection using Convolutional Neural Networks, gabor filters, and attention mechanisms. *Eng. Struct.* **2024**, *314*, 118343. [CrossRef]
24. Monteiro, G.A.A.; Monteiro, B.A.A.; dos Santos, J.A.; Wittemann, A. Pre-trained artificial intelligence-aided analysis of nanoparticles using the segment anything model. *Sci. Rep.* **2025**, *15*, 2341. [CrossRef]
25. He, P.P.; Hu, Y.; Song, H.Y.; He, W.; Zhao, X.J. A vision measurement method for ship hull plates based on multi-view stereo and image segmentation. *Measurement* **2024**, *238*, 115386. [CrossRef]
26. Zhao, Y.Q.; Ye, H.X. Crater detection and population statistics in Tianwen-1 landing area based on segment anything model (SAM). *Remote Sens.* **2024**, *16*, 1743. [CrossRef]
27. Huang, Y.H.; Yang, X.; Liu, L. Segment anything model for medical images? *Med. Image Anal.* **2024**, *92*, 103061. [CrossRef]
28. Carraro, A.; Sozzi, M.; Marinello, F. The Segment Anything Model (SAM) for accelerating the smart farming revolution. *Smart Agric. Technol.* **2023**, *6*, 100367. [CrossRef]
29. Shan, L.Q.; Liu, Y.C.; Du, K.; Paul, S.; Zhang, X.L.; Hei, X.L. Drilling rock image segmentation and analysis using segment anything model. *Adv. Geo-Energy Res.* **2024**, *12*, 89–101. [CrossRef]
30. Teng, S.; Liu, A.R.; Situ, Z.; Chen, B.C.; Wu, Z.H.; Zhang, Y.X.; Wang, J.L. Plug-and-play method for segmenting concrete bridge cracks using the segment anything model with a fractal dimension matrix prompt. *Autom. Constr.* **2025**, *170*, 105906. [CrossRef]
31. Hechkel, W.; Helali, A. Early detection and classification of Alzheimer's disease through data fusion of MRI and DTI images using the YOLOv11 neural network. *Front. Neurosci.* **2025**, *19*, 1554015. [CrossRef]
32. Khanam, R.; Hussain, M. YOLOv11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725. [CrossRef]
33. Sazak, H.; Kotan, M. Automated blood cell detection and classification in microscopic images using YOLOv11 and optimized weights. *Diagnostics* **2025**, *15*, 22. [CrossRef] [PubMed]
34. Rodríguez-Abreo, O.; Quiroz-Juárez, M.A.; Macías-Socarras, I.; Rodríguez-Reséndiz, J.; Camacho-Pérez, J.M.; Carcedo-Rodríguez, G.; Camacho-Pérez, E. Automatic detection of railway faults using neural networks: A comparative study of transfer learning models and YOLOv11. *Infrastructures* **2025**, *10*, 3. [CrossRef]
35. Parrany, A.M.; Mirzaei, M. A new image processing strategy for surface crack identification in building structures under non-uniform illumination. *Iet Image Process.* **2022**, *16*, 407–415. [CrossRef]
36. Khanam, R.; Asghar, T.; Hussain, M. Comparative performance evaluation of YOLOv5, YOLOv8, and YOLOv11 for solar panel defect detection. *Solar* **2025**, *5*, 6. [CrossRef]
37. Khan, A.T.; Jensen, S.M. LEAF-Net: A unified framework for leaf extraction and analysis in multi-crop phenotyping using YOLOv11. *Agriculture* **2025**, *15*, 196. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.