AI Engineering
https://www.sciltp.com/journals/aieng



Article

Check for updates

Adaptive Mask-Guided Crack Attention Mask R-CNN for Automatic Crack Detection

Lihua Pan, Yuanyuan Ou, Wanyi Li *, Yuewei Ma, Shiyong Huang, Jinmei He, Daizheng Kuang and Qian Zhang

School of Computer Science and Artificial Intelligence, Guangdong University of Education, Guangzhou 510303, China
* Correspondence: luther1212@163.com

How To Cite: Pan, L.; Ou, Y.; Li, W.; et al. Adaptive Mask-Guided Crack Attention Mask R-CNN for Automatic Crack Detection. *AI Engineering* **2026**, *2*(2), 21. https://doi.org/10.53941/aieng.2026.100021

Received: 5 July 2026
Revised: 22 October 2026
Accepted: 26 October 2026
Published: 9 November 2026

Abstract: Crack detection plays an important role in infrastructure inspection; however, the utilization of existing pixel-level crack annotations for object detection remains challenging due to the absence of detection-oriented bounding-box labels. In this study, an Adaptive Crack-Guided Attention Mask R-CNN (Adaptive CGA-Mask R-CNN) framework is proposed for automatic crack detection. An adaptive mask-to-box generation strategy is developed to transform pixel-level crack masks into object detection annotations through background identification, morphological refinement, and connected-component analysis. Furthermore, an adaptive crack-guided attention module is introduced into the feature pyramid network of Mask R-CNN to enhance crack-related feature representation across multiple scales. The proposed framework preserves the original detection architecture while improving the sensitivity to thin, irregular, and complex crack structures. Experimental results demonstrate that the proposed method achieves competitive localization performance compared with conventional object detection approaches. Additional comparative and cross-dataset evaluations further verify the effectiveness and generalization capability of the proposed framework. The proposed method provides an efficient solution for exploiting existing segmentation annotations in automatic crack detection applications.

Keywords: crack detection; adaptive crack-guided attention; Mask R-CNN; mask-to-box generation; object detection; infrastructure inspection

1. Introduction

Traditional crack detection methods mainly relied on handcrafted image-processing techniques to identify crack regions from complex surface backgrounds [1]. Deep learning-based crack detection methods can generally be divided into object detection and image segmentation approaches [2]. Object detection methods, such as Faster R-CNN [3], SSD [4], and YOLO [5], have achieved remarkable progress in locating crack regions using bounding boxes. These approaches are computationally efficient and can be easily integrated into practical inspection systems. However, bounding boxes provide only coarse spatial information and cannot preserve detailed crack morphology. In contrast, segmentation-based methods directly predict pixel-level crack regions [6,7], allowing more comprehensive characterization of crack geometry, including shape, area, and continuity. Therefore, segmentation approaches have become increasingly important for detailed crack analysis.

Among existing segmentation frameworks, Mask R-CNN [8] has demonstrated strong capability by integrating object detection and instance segmentation into a unified architecture. It can simultaneously provide object categories, bounding boxes, and pixel-level masks, making it suitable for comprehensive crack analysis. However, cracks usually exhibit unique visual characteristics that differ significantly from common objects. Specifically, cracks are generally characterized by narrow structures, irregular distributions, weak contrast, and strong background interference. These

Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

characteristics make crack feature extraction challenging, and important crack-related information may be weakened during conventional feature representation and multi-scale feature fusion.

The feature pyramid network (FPN) [9] used in conventional Mask R-CNN improves multi-scale feature representation through hierarchical feature fusion. However, the original FPN treats different spatial regions and feature channels relatively equally, which may limit its ability to distinguish crack-related features from complex background information. For crack detection tasks, where the target regions are usually small and structurally irregular, insufficient feature enhancement may result in missed detections, inaccurate localization, or incomplete segmentation.

Attention mechanisms [10,11] have recently been introduced into deep learning models to enhance informative feature representations by adaptively weighting important spatial and channel information. Nevertheless, most existing attention strategies are designed for general visual recognition tasks and are not specifically optimized for crack characteristics. Therefore, an adaptive attention mechanism that can focus on crack-specific patterns while suppressing irrelevant background features is required to improve crack detection performance.

To address these challenges, an Adaptive Crack-Guided Attention Mask R-CNN (Adaptive CGA-Mask R-CNN) framework is proposed for automatic crack detection, as illustrated in Figure 1. In the proposed framework, an Adaptive Crack-Guided Attention (CGA) module is integrated into the feature pyramid network of Mask R-CNN. By adaptively recalibrating spatial and channel-wise feature responses, the proposed module enhances crack-related representations and improves the capability of the network to capture thin, irregular, and low-contrast crack structures. Furthermore, based on the multi-task prediction capability of Mask R-CNN, the proposed framework can simultaneously generate crack categories, bounding boxes, and pixel-level masks, providing both object-level localization and pixel-level morphological information.

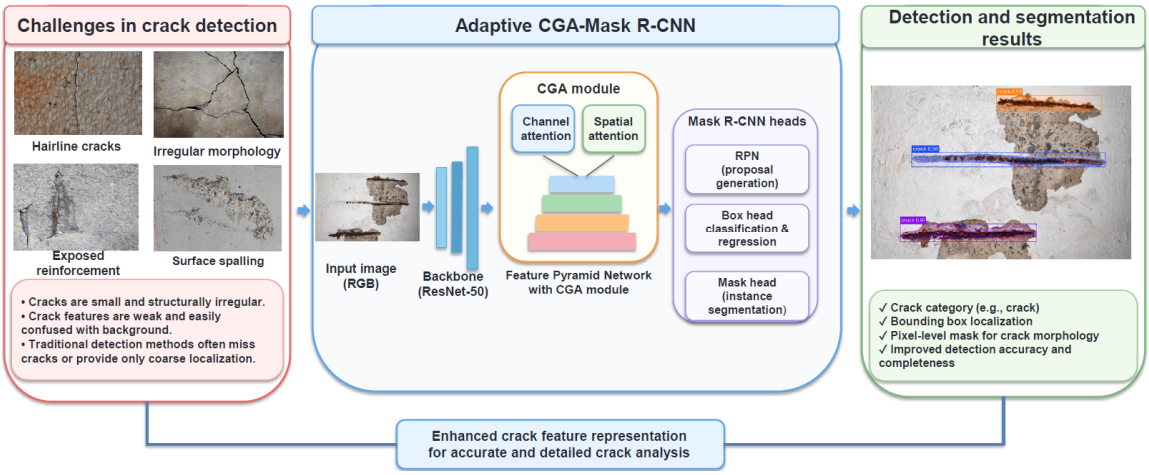


Figure 1. Motivation of the proposed Adaptive CGA-Mask R-CNN framework. (The concrete damage images are sourced from the publicly available HRCDS dataset [12] (CC BY 4.0). The detection and segmentation results were generated by the authors using the proposed model.).

The main contributions of this study are summarized as follows:

(1) An Adaptive Crack-Guided Attention module is proposed and integrated into Mask R-CNN to enhance crack-specific feature representation. The proposed module adaptively recalibrates spatial and channel features, enabling more effective extraction of thin, irregular, and low-contrast crack patterns under complex background conditions.

(2) An Adaptive CGA-Mask R-CNN framework is developed for multi-level crack analysis. Different from conventional object detection methods that only provide bounding-box localization, the proposed framework simultaneously predicts crack categories, bounding boxes, and pixel-level masks, allowing comprehensive characterization of crack location and morphology.

(3) A comprehensive experimental evaluation is conducted using quantitative comparisons, ablation studies, and cross-dataset validation. The effectiveness of the proposed framework is verified through comparisons with representative detection and segmentation approaches, demonstrating improved crack localization and segmentation performance.

2. Related Work

2.1. Deep Learning-Based Crack Detection

Deep learning-based methods [13–16] have been widely applied in automatic crack detection due to their strong capability in extracting discriminative features from complex images. Early studies mainly focused on crack classification, which can efficiently identify crack and non-crack regions but cannot provide accurate spatial localization.

To achieve detailed crack analysis, segmentation-based methods such as FCN [17] and U-Net [18] have been extensively explored. These approaches provide pixel-level crack regions and preserve morphological information. Fully convolutional architectures have also been specifically applied to concrete crack segmentation [7]. However, segmentation models usually require dense annotations and may suffer from high computational costs. Therefore, more efficient detection frameworks capable of providing accurate localization are still highly desirable.

2.2. Object Detection and Instance Segmentation for Crack Detection

Object detection methods, including Faster R-CNN, SSD, and YOLO, have been widely adopted for crack localization due to their efficiency and deployment advantages. R-CNN-based detection frameworks have also been investigated specifically for crack detection in concrete and bridge inspection applications [19–21]. Among them, Faster R-CNN with feature pyramid networks (FPN) provides strong multi-scale feature representation and has demonstrated effectiveness for detecting small and irregular defects. However, conventional object detection methods mainly rely on bounding boxes and cannot describe detailed crack morphology. Mask R-CNN extends Faster R-CNN by introducing an additional mask prediction branch, enabling simultaneous prediction of object categories, bounding boxes, and pixel-level masks. Therefore, it provides a suitable framework for comprehensive crack analysis.

Nevertheless, crack detection remains challenging because cracks usually exhibit thin structures, irregular shapes, weak contrast, and strong background interference. These characteristics may limit the feature representation capability of conventional Mask R-CNN.

2.3. Attention Mechanisms for Crack Feature Enhancement

Attention mechanisms have been introduced into deep neural networks to enhance feature representation by adaptively emphasizing important spatial regions and feature channels. Existing attention modules have demonstrated effectiveness in improving object detection and segmentation performance.

However, most attention mechanisms are designed for general visual tasks and do not specifically consider the characteristics of cracks. Due to the narrow geometry and low semantic response of cracks, conventional attention strategies may not sufficiently capture crack-related features. Transformer-based crack segmentation methods have further demonstrated the value of attention-based global feature modeling [11,22].

Therefore, an adaptive crack-oriented attention mechanism is required to enhance discriminative crack features while suppressing irrelevant background information. In this study, an Adaptive Crack-Guided Attention (CGA) module is proposed and integrated into Mask R-CNN to improve crack feature representation and achieve accurate crack localization and segmentation.

3. Methodology

3.1. Overall Framework

The proposed framework formulates crack localization as a single-class instance detection and segmentation problem. Each RGB image is paired with a pixel-level crack mask. Because the detection branch requires bounding-box annotations, the segmentation masks are first converted into instance-level bounding boxes and binary instance masks through a deterministic preprocessing pipeline.

The resulting targets are used to train a Mask R-CNN [8] detector equipped with a ResNet-50-FPN [9] backbone and an adaptive crack-guided attention module. The proposed framework consists of four main stages. First, the input image is normalized and resized by the standard transformation module of Mask R-CNN. Second, the ResNet-50 backbone extracts hierarchical features, and the feature pyramid network generates multi-scale feature representations. Third, the adaptive crack-guided attention module recalibrates the FPN features along the channel and spatial dimensions. Finally, the enhanced features are processed by the region proposal network and the RoI box and mask heads to generate crack bounding boxes, class labels, confidence scores, and instance masks. The architecture of Adaptive CGA-Mask R-CNN can be seen in Figure 2.

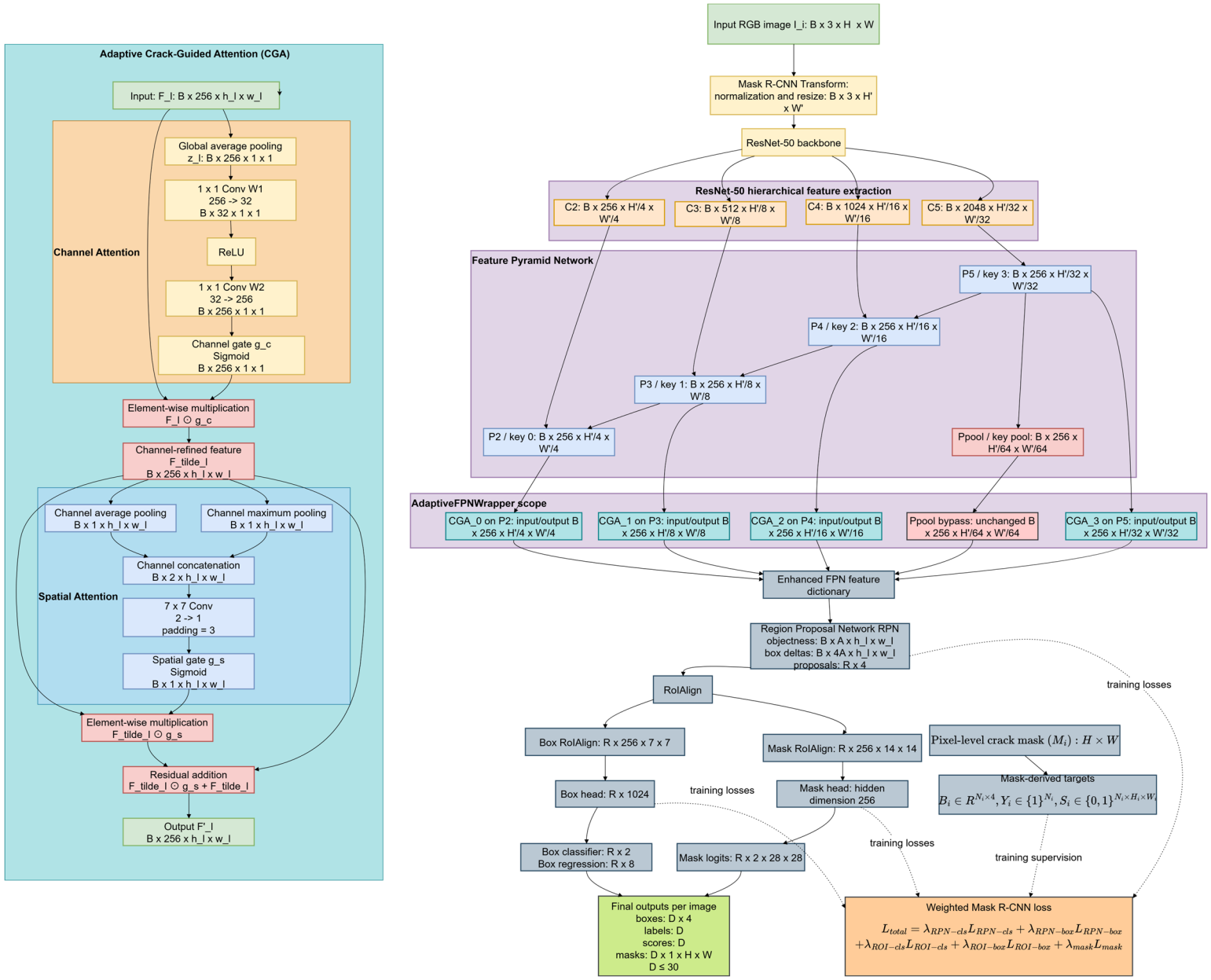


Figure 2. Overall architecture of the proposed Adaptive CGA-Mask R-CNN framework.

3.2. Mask-Derived Supervision

The original annotations are segmentation masks, whereas the detection branch requires bounding-box targets. To avoid introducing a separate annotation source, all training targets are generated directly from the segmentation masks. Three deterministic mask-processing strategies are considered: fixed-threshold binarization, dominant-background normalization, and dominant-background normalization followed by morphological closing. Previous studies have also shown that preprocessing strategies can influence the performance of R-CNN-based crack identification systems [20].

3.2.1. Fixed-Threshold Binarization

For the fixed strategy, the grayscale mask is converted into a binary foreground map using a predefined threshold t:

$$B_i^{\text{fix}}(x,y) = 1[M_i(x,y) > t] \tag{1}$$

where M_i denotes the mask of image i, B_i^{fix} is the resulting binary foreground map, and $1[\cdot]$ is the indicator function. The default configuration uses $t = 0$, meaning that all pixels with values greater than zero are treated as crack pixels. This strategy is appropriate when the mask encoding uses zero for the background and positive values for the crack region.

3.2.2. Dominant-Background Normalization

Because mask files may use different grayscale values or color encodings, a dominant-background normalization strategy is also used. The most frequent pixel value, or the most frequent RGB color for a color mask, is selected as the background value:

$$v_i^{\text{bg}} = \underset{v}{\text{argmax}}|\{(x,y): M_i(x,y) = v\}| \tag{2}$$

The binary foreground map is then obtained as:

$$B_i^{\text{dom}}(x,y) = 1\left[M_i(x,y) \neq v_i^{\text{bg}}\right] \tag{3}$$

This operation normalizes the mask encoding before connected-component extraction. It is treated as a deterministic annotation normalization procedure rather than as a new image-processing algorithm.

3.2.3. Morphological Closing

To evaluate the influence of filling small gaps and connecting nearby foreground regions, morphological closing is applied to the dominant-background binary mask:

$$B_i^{\text{morph}} = \left(B_i^{\text{dom}} \oplus K\right) \ominus K \tag{4}$$

where $\oplus$ and $\ominus$ denote dilation and erosion, respectively, and K is a rectangular structuring element. The implementation uses a 9×9 kernel for two iterations. This operation is applied only to the dominant_morph training variant. 3.2.4 Connected components and instance targets. The binary mask is decomposed into connected components using eight-connectivity. Components with an area smaller than $A_{\min} = 120$ pixels are removed. For each remaining component $C_{i,k}$, the tight axis-aligned bounding box is defined as:

$$b_{i,k} = \left[x_{\min}(C_{i,k}), y_{\min}(C_{i,k}), x_{\max}(C_{i,k}), y_{\max}(C_{i,k})\right], \quad \left|C_{i,k}\right| \geq A_{\min} \tag{5}$$

The same connected components are retained as instance masks for the Mask R-CNN mask branch. All retained regions are assigned to a single crack class. Therefore, the target of image i consists of: $\mathbf{B}_i \in \mathbb{R}^{N_i \times 4}$, $\mathbf{Y}_i \in \{1\}^{N_i}$, $\mathbf{S}_i \in \{0,1\}^{N_i \times H_i \times W_i}$, where N_i is the number of retained crack instances. For validation and testing, the same dominant-background rule without morphological closing is used to generate reference boxes for all training variants. Consequently, the evaluation targets remain fixed while the training annotation strategy changes. where N_i is the number of retained crack instances. For validation and testing, the same dominant-background rule without morphological closing is used to generate reference boxes for all training variants. Consequently, the evaluation targets remain fixed while the training annotation strategy changes.

3.3. Multi-Scale Feature Extraction and Adaptive Attention

3.3.1. ResNet-50-FPN Backbone

The detector adopts the torchvision implementation of Mask R-CNN with a ResNet-50 backbone and a feature pyramid network. The ResNet-50 backbone extracts feature maps at four spatial scales. Its internal channel dimensions are 256, 512, 1024, and 2048, respectively. The FPN projects these feature maps to a unified channel dimension of 256 and produces multi-scale feature representations. The spatial strides of the principal FPN levels are approximately 4, 8, 16, and 32 pixels relative to the input image. The additional pooled feature level has an approximate stride of 64 pixels. For an FPN feature at level l, the input to the attention module is $F_l \in \mathbb{R}^{256\times h_l\times w_l}$. The attention operation preserves both the spatial resolution and the channel dimension. The adaptive attention blocks are applied to the four principal FPN levels corresponding to keys 0, 1, 2, and 3. The additional pooled feature level is passed through unchanged.

3.3.2. Adaptive Crack-Guided Attention (CGA)

The adaptive crack-guided attention module contains a channel gate followed by a spatial gate. For an input feature F_l, global average pooling is first used to obtain a channel descriptor $z_l = \mathrm{GAP}(F_l)$ The descriptor is passed through two 1×1 convolutions. The intermediate dimension is 32, corresponding to a channel reduction ratio of 8:

$$g_l^c = \sigma\big(W_2\delta(W_1 z_l)\big), \quad W_1: 256 \to 32, \quad W_2: 32 \to 256 \tag{6}$$

where $\delta(\cdot)$ denotes the ReLU activation and $\sigma(\cdot)$ denotes the sigmoid activation. The channel-refined feature is calculated by element-wise multiplication:

$$\tilde{F}_l = F_l \odot g_l^c \tag{7}$$

The spatial gate then aggregates the channel-wise average and maximum responses:

$$g_l^s = \sigma\left(f^{7\times7}\left(\left[\mathrm{Avg}_c\left(\tilde{F}_l\right); \mathrm{Max}_c\left(\tilde{F}_l\right)\right]\right)\right) \tag{8}$$

where $f^{7\times7}$ denotes a 7×7 convolution and $[\cdot;\cdot]$ denotes channel-wise concatenation. The final enhanced feature is obtained using a residual connection:

$$F_l' = \tilde{F}_l \odot g_l^s + \tilde{F}_l \tag{9}$$

Each attention block contains 16,771 trainable parameters. Since four FPN levels are enhanced, the proposed attention module introduces 67,084 additional parameters.

The main contribution of CGA lies in its task-oriented feature-enhancement design for crack detection. Specifically, CGA employs global average pooling followed by two (1 × 1) convolutional layers to generate channel-wise weights, and subsequently applies spatial attention to recalibrate feature responses at different spatial locations. An explicit residual connection is introduced after spatial attention to preserve the channel-refined features alongside the spatially recalibrated features. This design aims to reduce the potential suppression of informative feature responses during spatial recalibration, which may be beneficial for detecting thin and low-contrast crack structures. Furthermore, CGA is integrated into four principal FPN feature levels of Mask R-CNN to enhance multiscale feature representations for subsequent detection and instance segmentation. These design choices constitute the main methodological contribution of the proposed framework.

3.4. Region Proposals and RoI Prediction Heads

The enhanced FPN features are supplied to the region proposal network. The RPN predicts objectness scores and bounding-box regression offsets for candidate regions at multiple scales. The resulting proposals are aligned using RoIAlign and then processed by two task-specific heads. The box head predicts two classes: background and crack. Its classification output has dimension $R \times 2$, where R is the number of sampled RoIs. The class-specific box regression output has dimension $R \times 8$, because four regression values are predicted for each of the two classes. The mask head uses a hidden dimension of 256 and produces class-specific mask logits with a nominal resolution of 28×28 for each RoI. After post-processing, the output of each image consists of boxes $\in \mathbb{R}^{D\times4}$, labels $\in \mathbb{R}^{D}$, scores $\in \mathbb{R}^{D}$, masks $\in \mathbb{R}^{D\times1\times H\times W}$, where D is the number of retained detections. The implementation limits the maximum number of detections per image to 30. Although the primary task is crack localization, the mask branch is retained because the implementation is Mask R-CNN rather than Faster R-CNN. The instance masks generated from the connected components are therefore used in the mask loss during training.

3.5. Weighted Training Objective and Optimization

The total training objective is defined as a weighted combination of the five standard Mask R-CNN losses:

$$L_{\text{total}} = \lambda_{\text{RPN-cls}} L_{\text{RPN-cls}} + \lambda_{\text{RPN-box}} L_{\text{RPN-box}} + \lambda_{\text{ROI-cls}} L_{\text{ROI-cls}} + \lambda_{\text{ROI-box}} L_{\text{ROI-box}} + \lambda_{\text{mask}} L_{\text{mask}} \quad (10)$$

where, $L_{\text{RPN-cls}}$ is the RPN objectness classification loss; $L_{\text{RPN-box}}$ is the RPN proposal-box regression loss; $L_{\text{ROI-cls}}$ is the RoI classification loss; $L_{\text{ROI-box}}$ is the RoI bounding-box regression loss; L_{mask} is the instance-mask loss. The coefficients control the relative contribution of the different objectives.

The coefficients $\lambda_{\text{RPN-cls}}$, $\lambda_{\text{RPN-box}}$, $\lambda_{\text{ROI-cls}}$, $\lambda_{\text{ROI-box}}$, and λ_{mask} are adjustable hyperparameters that control the relative contributions of the five loss components. In the initial configuration, all coefficients were set to 1.0:

$$\lambda_{\text{RPN-cls}} = \lambda_{\text{RPN-box}} = \lambda_{\text{ROI-cls}} = \lambda_{\text{ROI-box}} = \lambda_{\text{mask}} = 1 \quad (11)$$

This equal-weight configuration was used as the initial training setting. The coefficients can be adjusted in subsequent experiments according to validation performance or ablation results, and the selected values should remain fixed across all compared models.

3.6. Training Procedure

The training hyperparameters were treated as configurable variables rather than strictly fixed constants. The main parameters were selected within practical ranges according to the available computational resources and validation performance. The relevant parameters are shown in Table 1, and the process is shown in Figure 3.

Before model training, a pre-training annotation check is performed to verify the reliability of the generated targets. The pixel-value or color distribution of representative masks is first inspected, followed by statistical analysis of the generated bounding boxes, including the number of empty images, empty ratio, total number of boxes, and average number of boxes per image. In addition, the minimum connected-component area and morphological parameters are treated as configurable settings. This procedure is used to identify abnormal mask encoding or inappropriate preprocessing parameters before training and to reduce the risk of invalid automatically generated annotations.

Table 1. The training parameters.

Parameter	Adjustable Range	Initial Setting
Training epochs	30~100	50
Batch size	4~16	12
Learning rate	1×10^{-5}~5×10^{-4}	2×10^{-4}
Weight decay	1×10^{-5}~1×10^{-3}	1×10^{-4}
Gradient clipping norm	1–10	5
Minimum connected-component area	50–200 pixels	120 pixels
Morphological kernel size	3–11, odd values	9×9
Morphological iterations	1~3	2
Loss coefficients	0.5~2.0	1.0
DataLoader workers	2~8	4

At each iteration, the input images and their corresponding targets were passed through the model. The RPN classification loss, RPN box regression loss, RoI classification loss, RoI box regression loss, and instance-mask loss were combined into the total objective. When CUDA was available, automatic mixed-precision training was enabled. Gradient clipping was applied before the AdamW update to prevent excessively large parameter changes. The model was trained using the initial settings (50 epochs) listed in Table 1, with early stopping applied based on validation performance.

After each epoch, the model was evaluated on the validation set. The confidence threshold was searched within the range of 0.05–0.60 using the candidate values:

$$\{0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40, 0.50, 0.60\}$$

The checkpoint with the highest validation F1-score was selected as the final model. The corresponding confidence threshold was subsequently applied to the test set.

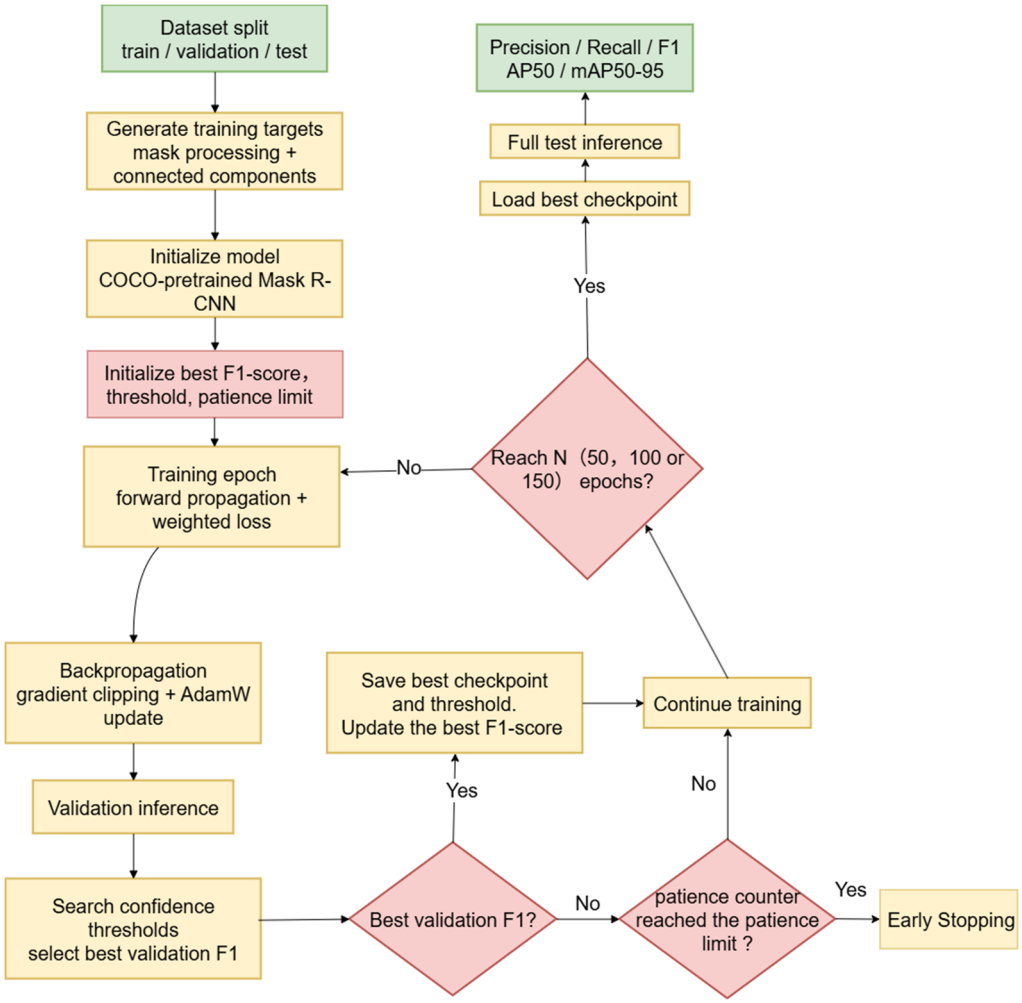


Figure 3. The training flowchart.

Several measures were used to reduce overfitting. First, the detector was initialized with COCO-pretrained weights [23]. Second, AdamW weight decay was applied within the range of $10^{-5}$to 10^{-3}. Third, model selection was based on the best validation F1-score rather than the final training epoch. Fourth, the confidence threshold was determined only from the validation set and was not adjusted using test data. Finally, all comparison variants used the same data split and reference targets. The code currently does not implement data augmentation, dropout, or learning-rate scheduling.

4. Experiments and Results

4.1. Experimental Setup

The experiments are primarily conducted on the HRCDS dataset [12] which provides high-resolution images with pixel-level damage annotations. The HRCDS dataset contains three subsets, including training, validation, and testing sets. The training set is used for model optimization, where crack annotations are generated from the corresponding segmentation masks. The validation set is used to monitor model performance and determine the optimal confidence threshold. The testing set is used for final evaluation. To ensure a fair comparison, all models use the same reference annotations in the validation and testing phases, which are generated from the original crack masks using a unified strategy.

The HRCDS dataset is pre-divided into three subsets, including the training set, validation set, and testing set. The training set is used for model optimization, the validation set is used for model performance monitoring and confidence threshold selection, and the testing set is used for final evaluation. Each subset contains pavement images and the corresponding crack segmentation masks, which are further processed to generate detection annotations. All experiments are conducted on a workstation equipped with a CUDA-enabled GPU. The proposed model is implemented using the PyTorch framework, and the detailed hardware and software configurations are listed in Table 2.

Table 2. Experimental hardware and software environment.

Component	Configuration/Version
CPU	Intel(R) Core(TM) Ultra 9 275HX
GPU	NVIDIA GeForce RTX 5080 Laptop GPU
GPU memory (VRAM)	16 GB GDDR7
System memory (RAM)	32 GB DDR5
Storage	8 TB SSD
Operating system	Windows 11 64 bit
Python	3.11.13
PyTorch	2.11.0+cu128
Torchvision	0.26.0+cu128
OpenCV	4.11.0
CUDA version	12.8
Computing device	CUDA

4.2. Evaluation Metrics

Evaluation Metrics: To quantitatively evaluate the crack detection performance, several commonly used metrics are adopted, including Precision, Recall, F1-score, Average Precision (AP50), mean Average Precision (mAP50-95), and mask-level segmentation metrics. Precision, Recall, and F1-score: Precision measures the proportion of correctly detected cracks among all predicted crack instances, while Recall evaluates the ability of the model to identify actual crack instances. They are defined as:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

where TP, FP, and FN represent true positives, false positives, and false negatives, respectively. To provide a comprehensive evaluation considering both detection accuracy and completeness, the F1-score is calculated as:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

In this study, a predicted bounding box is considered a true positive when its Intersection over Union (IoU) with the ground-truth box is greater than 0.5. The IoU threshold is set to 0.5 for Precision, Recall, and F1-score calculation. Average Precision (AP50) and mAP50-95. Following the commonly used object detection evaluation protocol, Average Precision at IoU threshold 0.5 (AP_{50}) and mean Average Precision over multiple IoU thresholds from 0.50 to 0.95 ($mAP_{50:95}$) are adopted:

$$AP_{50} = AP(IoU = 0.5) \tag{15}$$

where AP_{50} evaluates detection performance under a relatively relaxed localization criterion. The $mAP_{50:95}$ metric follows the COCO evaluation protocol and averages AP values over ten IoU thresholds:

$$mAP_{50:95} = \frac{1}{10}\sum_{i=0}^{9} A\,P_{0.50+0.05i} \tag{16}$$

Compared with AP_{50}, $mAP_{50:95}$ provides a stricter evaluation of localization accuracy. Mask-based Segmentation Metrics: Since the proposed framework performs instance segmentation, mask-level accuracy is additionally evaluated using Mask IoU and Dice coefficient. Mask IoU is defined as:

$$IoU_{mask} = \frac{M_p \cap M_{gt}}{M_p \cup M_{gt}} \tag{17}$$

where M_p and M_{gt} denote the predicted crack mask and ground-truth mask, respectively. The Dice coefficient is calculated as:

$$Dice = \frac{2\left|M_p \cap M_{gt}\right|}{\left|M_p\right| + \left|M_{gt}\right|} \tag{18}$$

These metrics measure the pixel-level overlap between predicted crack regions and reference annotations.

4.3. Comparison with Baseline Methods

To evaluate the effectiveness of the proposed Adaptive CGA-Mask R-CNN, five representative baseline models, including SSD [4], Faster R-CNN [3], Mask R-CNN [8], YOLOv8n [5], and YOLOv8-seg [5]are selected for comparison. The experiments are conducted on the HRCDS dataset [12] and an external CrackForest dataset [24] to evaluate both the in-domain detection performance and cross-dataset generalization capability.

It should be noted that SSD, Faster R-CNN, and YOLOv8n are object detection models that only generate bounding box predictions without pixel-level segmentation masks. Therefore, Mask IoU and Mask Dice metrics are not applicable to these methods. In contrast, Mask R-CNN, YOLOv8-seg, and the proposed Adaptive CGA-Mask R-CNN are instance segmentation models, and both detection and segmentation metrics are reported. To ensure a fair and consistent comparison, all models, including SSD, Faster R-CNN, Mask R-CNN, YOLOv8n, YOLOv8-seg, and the proposed method, were evaluated using a fixed confidence score threshold of 0.5. Specifically, predictions with confidence scores below 0.5 were filtered out during post-processing. In addition, a minimum connected-component area (Min-Area) of 80 pixels was adopted to maintain a consistent crack-instance filtering criterion. These settings were applied consistently across all comparative experiments to minimize evaluation discrepancies caused by different confidence thresholds or instance-filtering configurations. AP50 and mAP50-95 were computed on the full confidence-ranked predictions before threshold-based filtering.

As shown in Table 3, the proposed Adaptive CGA-Mask R-CNN achieves superior performance on the HRCDS dataset. The proposed model obtains a Precision of 0.8296, Recall of 0.8000, and F1-score of 0.8145, outperforming the other compared methods on most metrics. Furthermore, it achieves the highest AP50 (0.7899) and mAP50-95 (0.6359), demonstrating its strong capability in crack localization and detection accuracy.

For segmentation performance, the proposed method achieves the highest Mask IoU (0.8911) and Mask Dice (0.9384), exceeding Mask R-CNN and YOLOv8-seg. These improvements indicate that the introduced CGA module enhances crack feature representation and contributes to more accurate pixel-level crack segmentation.

To further investigate the generalization capability of the proposed method, a cross-dataset evaluation experiment is conducted on the external CrackForest dataset. As shown in Table 4, all models are trained using the HRCDS dataset and directly evaluated on CrackForest without additional training, fine-tuning, or parameter adjustment. Therefore, CrackForest serves as an unseen target domain to evaluate the robustness of the models under domain shifts.

Compared with the HRCDS results, all methods show certain performance degradation on CrackForest due to differences in crack morphology, background characteristics, illumination conditions, and image acquisition environments between the two datasets. Nevertheless, the proposed Adaptive CGA-Mask R-CNN maintains the best overall performance, achieving the highest F1-score (0.7747), AP50 (0.7521), and mAP50-95 (0.5948), indicating stronger cross-domain crack detection capability.

Table 3. Performance comparison of different crack detection models (detection and segmentation metrics, HRCDS Dataset).

Model	Precision	Recall	F1	AP50	mAP50-95	Mask_IoU	Mask_Dice	Inference (ms)	FPS	Params_M
SSD	0.8364	0.6571	0.7360	0.6461	0.3674	/	/	168.0426	5.9509	23.7459
Faster R-CNN	0.5851	0.7857	0.6707	0.7468	0.4902	/	/	251.9605	3.9689	41.2992
Mask R-CNN	0.6154	0.6857	0.6486	0.6825	0.4369	0.8323	0.8843	223.1511	4.4813	43.9224
YOLOv8n	0.8165	0.6357	0.7149	0.7152	0.5551	/	/	9.6251	103.8951	3.0058
YOLOv8-seg	0.5470	0.7071	0.6168	0.6647	0.5318	0.8523	0.9102	21.2293	47.1048	3.2638
Proposed	0.8296	0.8000	0.8145	0.7899	0.6359	0.8911	0.9384	194.1645	5.1503	43.9895

Note: Mask Dice represents the mean of per-instance Dice coefficients over all successfully matched prediction–ground-truth mask pairs, using confidence and mask IoU thresholds of 0.5.

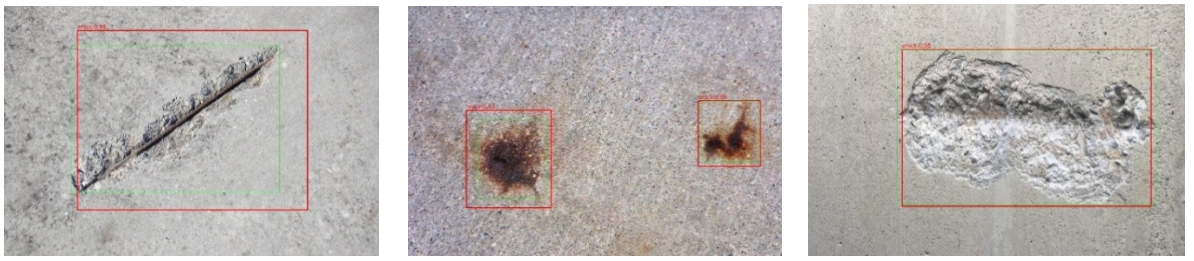
Table 4. Performance comparison of different crack detection models (detection and segmentation metrics, CrackForest Dataset).

Model	Precision	Recall	F1	AP50	mAP50-95	Mask IoU	Mask Dice	Inference (ms)	FPS	Params (M)
SSD	0.7815	0.6042	0.6816	0.5874	0.3228	/	/	171.3852	5.8367	23.7459
Faster R-CNN	0.5418	0.7165	0.6171	0.6812	0.4316	/	/	256.4728	3.8992	41.2992
Mask R-CNN	0.5726	0.6428	0.6057	0.6214	0.3925	0.7816	0.8524	226.7845	4.4098	43.9224
YOLOv8n	0.7584	0.5913	0.6644	0.6718	0.5126	/	/	9.8426	101.5964	3.0058
YOLOv8-seg	0.5127	0.6635	0.5782	0.6216	0.4867	0.8045	0.8796	21.8566	45.7526	3.2638
Proposed	0.7938	0.7564	0.7747	0.7521	0.5948	0.8617	0.9225	197.5268	5.0626	43.9895

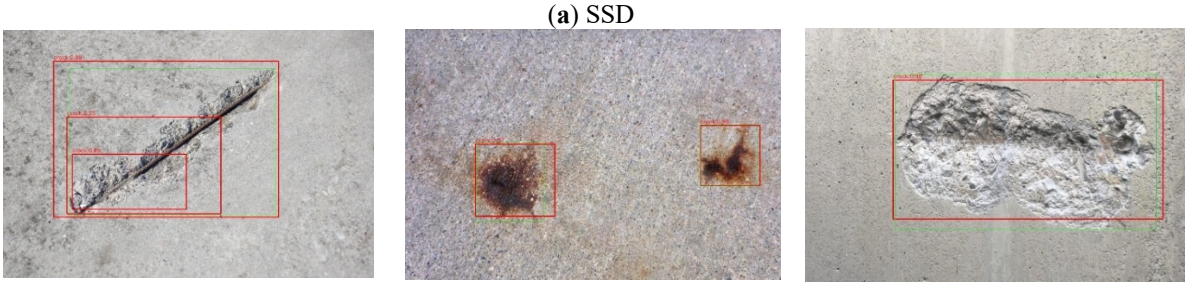
Note: Mask Dice represents the mean of per-instance Dice coefficients over all successfully matched prediction–ground-truth mask pairs, using confidence and mask IoU thresholds of 0.5.

Moreover, the proposed method achieves the highest segmentation performance on the external CrackForest dataset, with Mask IoU and Mask Dice values of 0.8617 and 0.9225, respectively. These results demonstrate that the proposed model can effectively preserve crack structural information and maintain robust segmentation performance when applied to unseen datasets.

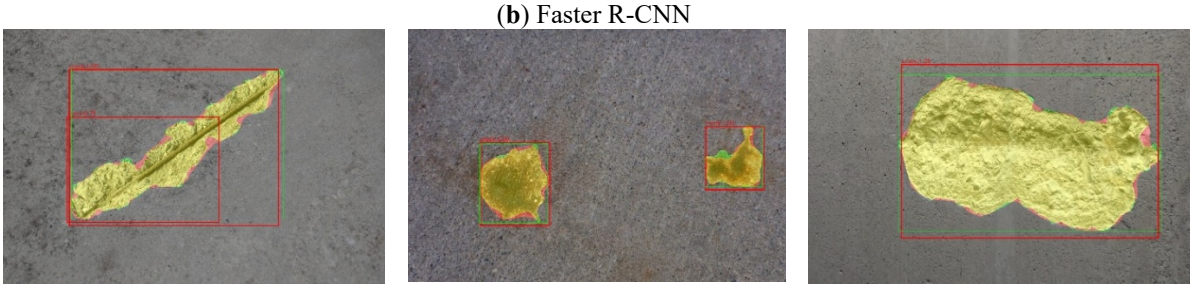
As shown in Figure 4, the proposed Adaptive CGA-Mask R-CNN demonstrates superior visual detection and segmentation performance on the HRCDS Dataset. For the three detection-only methods, SSD, Faster R-CNN, and YOLOv8n, obvious localization errors can be observed in several samples, where the predicted bounding boxes deviate noticeably from the ground-truth regions. In addition, although YOLOv8-seg is capable of generating pixel-level masks, missed detections can still be observed in some samples, resulting in incomplete representation of the defect regions. In comparison, the proposed method provides more accurate defect localization and more complete segmentation results. In the visualization, green represents the ground-truth defect regions, red represents the predicted regions, and yellow denotes the overlapping area between the prediction and the ground truth (Prediction ∩ Ground Truth). Therefore, a larger yellow region accompanied by fewer isolated red and green regions generally indicates better agreement between the predicted mask and the ground-truth annotation. As illustrated in Figure 4, the proposed method achieves a higher degree of overlap with the ground truth and better preserves the shape, continuity, and boundary details of building defects. These qualitative observations are consistent with the quantitative results reported in Table 3, further demonstrating the effectiveness of the proposed method in building defect detection and pixel-level segmentation.



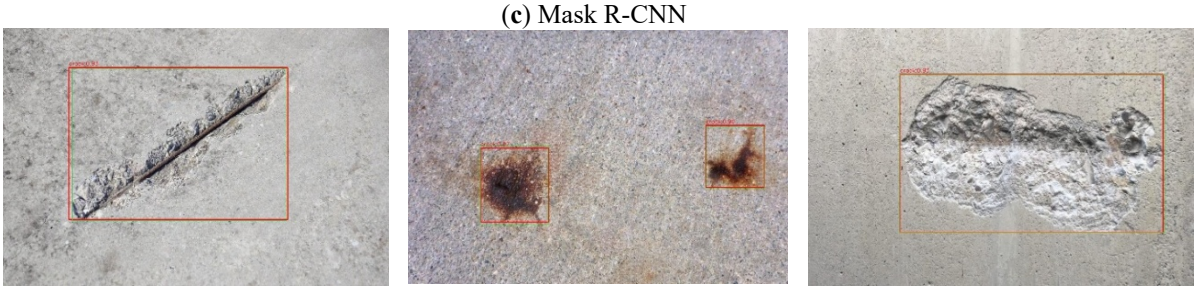
(**a**) SSD



(**b**) Faster R-CNN



(**c**) Mask R-CNN



(**d**) YOLOv8n

Figure 4. *Cont.*

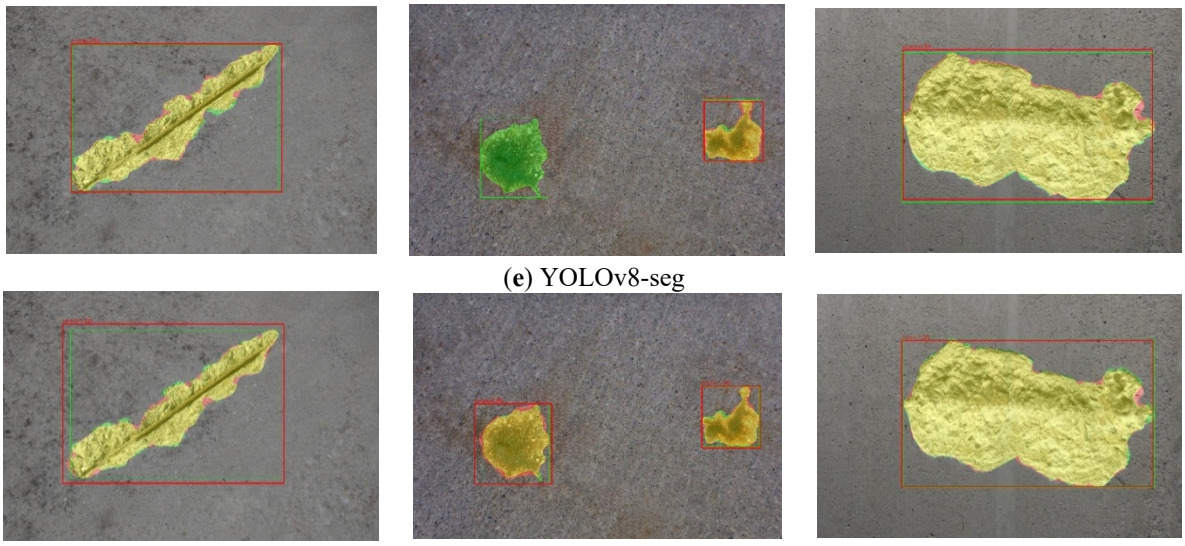


Figure 4. Visual comparison of crack detection and segmentation results.

4.4. Ablation Study

To investigate the effectiveness of the proposed Adaptive Crack-Guided Attention (CGA) module, ablation experiments were conducted by gradually introducing channel-wise and spatial attention branches into the Mask R-CNN framework. Specifically, different configurations were designed to evaluate the individual and combined contributions of the two attention branches. After each training epoch, the model was evaluated on the validation set, and the confidence threshold was selected from the candidate values of {0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40, 0.50, 0.60} based on the highest validation F1-score. In addition, a minimum connected-component area (Min-Area) of 120 pixels was consistently applied across all ablation variants. The same threshold selection strategy and evaluation settings were maintained for all ablation configurations to ensure a fair comparison of their performances.

As summarized in Table 5, the baseline model without any attention enhancement (CGA-none) achieved a Precision of 0.6146, Recall of 0.8429, and F1-score of 0.7108, with Mask IoU and Mask Dice values of 0.6667 and 0.8000, respectively. After introducing only the channel attention branch (CGA-channel), the F1-score increased to 0.7298, while Mask IoU and Mask Dice improved to 0.6775 and 0.8077. This improvement indicates that channel-wise feature recalibration enables the network to emphasize crack-related semantic responses and suppress less informative background features. Furthermore, when only the spatial attention branch was activated (CGA-spatial), the model achieved an F1-score of 0.7355, with Mask IoU and Mask Dice reaching 0.6534 and 0.7904, respectively. Compared with channel attention, spatial attention obtained larger improvements in pixel-level segmentation metrics, demonstrating its importance for capturing thin, discontinuous, and irregular crack structures.

Precision of 0.6146, Recall of 0.8429, and F1-score of 0.7108, with Mask IoU and Mask Dice values of 0.6667 and 0.8000, respectively. After introducing only the channel attention branch (CGA-channel), the F1-score increased to 0.7298, while Mask IoU and Mask Dice improved to 0.6775 and 0.8077. This improvement indicates that channel-wise feature recalibration enables the network to emphasize crack-related semantic responses and suppress less informative background features. Furthermore, when only the spatial attention branch was activated (CGA-spatial), the model achieved an F1-score of 0.7355, with Mask IoU and Mask Dice reaching 0.6534 and 0.7904, respectively.

The combination of both attention branches (CGA-full) achieved the best overall performance among all configurations. Compared with the CGA-none model, CGA-full improved Precision from 0.6146 to 0.8467 and F1-score from 0.7108 to 0.8375, representing increases of 23.21 percentage points and 12.67 percentage points, respectively. Meanwhile, the segmentation performance was significantly enhanced, with Mask IoU increasing from 0.6667 to 0.6992 and Mask Dice increasing from 0.8000 to 0.8230. In addition, the detection metrics were consistently improved, where AP50 increased from 0.7803 to 0.8048, and mAP50-95 increased from 0.5364 to 0.6466. These results demonstrate that the channel and spatial attention mechanisms provide complementary feature enhancement capabilities. The channel branch focuses on selecting discriminative crack-related feature channels, whereas the spatial branch captures the spatial distribution of crack regions. Their combination enables more effective feature representation for complex crack patterns with weak contrast and irregular morphology.

Overall, the proposed module provides substantial improvements in crack detection and segmentation accuracy while adding only 0.0671 M parameters (43.9895 M vs. 43.9224 M for Mask R-CNN), demonstrating a favorable balance between performance enhancement and model complexity.

Table 5. Ablation study of different CGA configurations on the HRCDS dataset.

Model	Channel Gate	Spatial Gate	Precision	Recall	F1	AP50	mAP50-95	Mask IoU	Mask Dice
CGA-none	×	×	0.6146	0.8429	0.7108	0.7803	0.5364	0.6667	0.8000
CGA-channel	√	×	0.7172	0.7429	0.7298	0.7254	0.4853	0.6775	0.8077
CGA-spatial	×	√	0.6706	0.8143	0.7355	0.7595	0.4902	0.6534	0.7904
CGA-full	√	√	0.8467	0.8286	0.8375	0.8048	0.6466	0.6992	0.8230

Note: √ indicates available, and × indicates unavailable. Mask Dice is calculated as 2 × Mask IoU/(1 + Mask IoU).

It should be noted that the ablation experiments adopted a validation-based confidence threshold selection strategy and a Min-Area of 120 pixels, whereas the comparative experiments in Table 3 used a fixed confidence threshold of 0.5 and a Min-Area of 80 pixels. These differences in experimental settings may lead to variations in the absolute evaluation results between the two tables. However, the evaluation settings were kept consistent among all variants within the ablation experiments, ensuring the validity of the internal comparisons.

5. Conclusions

In this study, an Adaptive Crack-Guided Attention Mask R-CNN (Adaptive CGA-Mask R-CNN) framework is developed for automatic crack detection and segmentation. Channel-wise and spatial attention mechanisms are integrated into the feature pyramid network to enhance crack-related feature representation and improve the extraction of thin, irregular, and low-contrast crack structures. Existing pixel-level crack annotations are converted into instance-level detection targets, allowing bounding-box localization and pixel-level segmentation to be performed simultaneously. Experimental results on the HRCDS Dataset show that the proposed method achieves an F1-score of 0.8145, AP50 of 0.7899, mAP50-95 of 0.6359, Mask IoU of 0.8911, and Mask Dice of 0.9384. The qualitative comparisons also show that more accurate crack localization, more complete mask prediction, and better preservation of crack morphology and boundary details are obtained by the proposed method.

To further evaluate the generalization capability of the proposed framework, cross-dataset experiments are conducted on the CrackForest Dataset. The models are trained only on HRCDS and are directly evaluated on CrackForest without additional training, fine-tuning, or parameter adjustment. The proposed method achieves an F1-score of 0.7747, AP50 of 0.7521, mAP50-95 of 0.5948, Mask IoU of 0.8617, and Mask Dice of 0.9225, indicating that stable performance is maintained under domain shift. In addition, the effectiveness of the CGA module is further verified through ablation experiments, where complementary improvements are provided by the channel and spatial attention branches. Overall, effective crack localization and pixel-level segmentation are achieved by the proposed framework, demonstrating its potential for automated inspection of cracks and structural defects in buildings and infrastructure.

Author Contributions

L.P.: conceptualization, methodology, investigation, writing—original draft; Y.O.: data curation, formal analysis, visualization; W.L.: supervision, project administration, funding acquisition, writing—review and editing; Y.M.: software, validation; S.H.: investigation, resources; J.H.: data curation, validation; D.K.: formal analysis, visualization; Q.Z.: methodology, software, validation. All authors have read and agreed to the published version of the manuscript.

Funding

This work is supported by the Undergraduate Innovation Training Project of Guangdong University of Education under Grant (No. 122, 2026), the Project for the Development of Guangdong Province Philosophy and Social Science (No. GD25CTY14), the Educational Science Planning Project of Guangdong Province (No. 2023GXJK125), the Guangdong Provincial Special Program in Key Areas for Higher Education Institutions (New Generation Electronic Information (Semiconductors)) (No. 2024ZDZX1040), the Guangdong Provincial Undergraduate Teaching Quality and Teaching Reform Development Project (Higher Education Teaching Reform Project, No. 905, 2025; Special Talent Training Program, No. 234, 2025), the Teaching Quality and Teaching Reform Project of Guangdong University of Education(No. 2025jxgg32, No. 2025jxgg31, No. 2024zxrcpyjh01),

and the Industry−Enterprise Course Resource Database Development Project of Guangdong University of Education (No. 54, 2024).

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The HRCDS dataset analysed in this study is publicly available in the Mendeley Data repository at https://doi.org/10.17632/6x4dzzrs2h.1. The CrackForest Dataset (CFD), associated with Shi et al. (2016), is publicly available at https://github.com/cuilimeng/CrackForest-dataset. The remaining data and the code are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this manuscript, the authors used ChatGPT and DeepSeek to check grammar and refine the wording of statements concerning the training epochs and cross-references to the tables and figures. After using these tools, the authors carefully reviewed and edited the content as necessary and take full responsibility for the content of the published article.

References

1. Fujita, Y.; Hamamoto, Y. A Robust Automatic Crack Detection Method from Noisy Concrete Surfaces. *Mach. Vis. Appl.* **2011**, *22*, 245–254. https://doi.org/10.1007/s00138-009-0244-5.
2. Hamishebahar, Y.; Guan, H.; So, S.; et al. A Comprehensive Review of Deep Learning-Based Crack Detection Approaches. *Appl. Sci.* **2022**, *12*, 1374. https://doi.org/10.3390/app12031374.
3. Ren, S.; He, K.; Girshick, R.; et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. https://doi.org/10.1109/tpami.2016.2577031.
4. Liu, W.; Anguelov, D.; Erhan, D.; et al. SSD: Single Shot MultiBox Detector. In *Computer Vision–ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Cham, Switzerland, 2016; pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.
5. Ultralytics YOLOv8. Available online: https://github.com/ultralytics/ultralytics (accessed on 31 October 2026).
6. Zhang, A.; Wang, K.C.P.; Fei, Y.; et al. Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces with a Recurrent Neural Network. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 213–229. https://doi.org/10.1111/mice.12409.
7. Dung, C.V.; Anh, L.D. Autonomous Concrete Crack Detection Using Deep Fully Convolutional Neural Network. *Autom. Constr.* **2019**, *99*, 52–58. https://doi.org/10.1016/j.autcon.2018.11.028.
8. He, K.; Gkioxari, G.; Dollar, P.; et al. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. https://doi.org/10.1109/iccv.2017.322.
9. Lin, T.Y.; Dollar, P.; Girshick, R.; et al. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. https://doi.org/10.1109/cvpr.2017.106.
10. Kang, D.H.; Cha, Y.J. Efficient Attention-Based Deep Encoder and Decoder for Automatic Crack Segmentation. *Struct. Health Monit.* **2022**, *21*, 2190–2205. https://doi.org/10.1177/14759217211053776.
11. Wang, J.; Zeng, Z.; Wang, J.; et al. Automatic Crack Segmentation Model Based on Multi-Branch Aggregation Transformer. *Adv. Struct. Eng.* **2024**, *27*, 2289–2302. https://doi.org/10.1177/13694332241266538.
12. Guo, P.; Bao, Y. HRCDS: A Benchmark Dataset for High-Resolution Concrete Damage Segmentation. Available online: https://doi.org/10.17632/6x4dzzrs2h.1 (accessed on 31 October 2026).
13. Kim, B.; Cho, S. Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique. *Sensors* **2018**, *18*, 3452. https://doi.org/10.3390/s18103452.

14. Rajadurai, R.S.; Kang, S.T. Automated Vision-Based Crack Detection on Concrete Surfaces Using Deep Learning. *Appl. Sci.* **2021**, *11*, 5229. https://doi.org/10.3390/app11115229.
15. Rao, A.S.; Nguyen, T.; Palaniswami, M.; et al. Vision-Based Automated Crack Detection Using Convolutional Neural Networks for Condition Assessment of Infrastructure. *Struct. Health Monit.* **2021**, *20*, 2124–2142. https://doi.org/10.1177/1475921720965445.
16. Cha, Y.J.; Choi, W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. https://doi.org/10.1111/mice.12263.
17. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. https://doi.org/10.1109/tpami.2016.2572683.
18. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015*; Springer: Cham, Switzerland, 2015; pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.
19. Xu, X.; Zhao, M.; Shi, P.; et al. Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN. *Sensors* **2022**, *22*, 1215. https://doi.org/10.3390/s22031215.
20. Zhao, M.; Shi, P.; Xu, X.; et al. Improving the Accuracy of an R-CNN-Based Crack Identification System Using Different Preprocessing Algorithms. *Sensors* **2022**, *22*, 7089. https://doi.org/10.3390/s22187089.
21. Gan, L.; Liu, H.; Yan, Y.; et al. Bridge Bottom Crack Detection and Modeling Based on Faster R-CNN and BIM. *IET Image Process.* **2024**, *18*, 664–677. https://doi.org/10.1049/ipr2.12976.
22. Wang, W.; Su, C. Automatic Concrete Crack Segmentation Model Based on Transformer. *Autom. Constr.* **2022**, *139*, 104275. https://doi.org/10.1016/j.autcon.2022.104275.
23. Lin, T.Y.; Maire, M.; Belongie, S.; et al. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2014; pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48.
24. Shi, Y.; Cui, L.; Qi, Z.; et al. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. https://doi.org/10.1109/tits.2016.2552248.