Real-Time Detection and Monitoring of Structural Cracks Using ConcreteCrack

Yanting Song[1], Lei Xing[1*], Yunlong Song[2], Jian Li[1]

[1]Liaoning Institute of Science and Engineering, Liaoning JinZhou,
121000, China.
[2]Tongji University, Shanghai, 201804, China.

*Corresponding author(s). E-mail(s): 20249001@lise.edu.cn;
Contributing authors: songyunlong@tongji.edu.cn;

Abstract

Structural crack detection is a critical task in infrastructure monitoring and maintenance, as early identification of cracks can prevent severe structural damage and reduce maintenance costs. In this work, we propose **ConcreteCrack**, a YOLOv11n-based detection framework enhanced with hierarchical feature extraction (HGStem), multi-scale feature fusion (HGBlock), and dynamic feature alignment (DynamicAlignFusion) modules to accurately detect cracks of varying sizes, shapes, and orientations. Extensive experiments on benchmark crack datasets demonstrate that our method outperforms several state-of-the-art object detection algorithms, achieving high Precision, Recall, F1-score, and mAP, while maintaining real-time inference speed. Furthermore, Grad-CAM visualizations validate the interpretability of the model by highlighting actual crack regions, ensuring reliable detection even in complex scenarios. The proposed approach provides a robust and efficient solution for automated structural crack monitoring, enabling safer and more effective infrastructure inspection.

Keywords: ConcreteCrack, HGStem, Deep learning

1 Introduction

Concrete structures are integral to modern infrastructure, yet they are highly susceptible to various forms of degradation, including cracking, which can compromise both structural integrity and safety. Traditional methods of crack detection, such as manual inspection and basic image processing techniques, often suffer from low efficiency, subjective bias, and limited scalability Ren et al. (2025); Ashraf et al. (2023); Maslan and Cicmanec (2023). These limitations highlight the pressing need for automated, accurate, and scalable approaches to structural health monitoring.

The advent of deep learning has ushered in a new era for automated crack detection, offering substantial improvements over conventional approaches. Convolutional neural networks (CNNs), in particular, have demonstrated remarkable capability in extracting discriminative visual features for crack segmentation and classification Gooda et al. (2023); Deng et al. (2023); Yu and Zhou (2023). However, while CNNs effectively capture local patterns, they often struggle to model long-range dependencies and multi-scale context, which are critical for detecting fine and irregular cracks in complex environments Inam et al. (2023); Li et al. (2023).

Among deep learning models, You Only Look Once (YOLO) variants have gained prominence due to their real-time processing capabilities and high detection accuracy. Recent studies have introduced several YOLO-based models tailored for concrete crack detection. For instance, Ren et al. (2025) proposed the BCCD-YOLO model, which enhances the Path Aggregation Network (PAN) with lateral skips and weighted feature fusion mechanisms, improving multi-scale feature extraction for crack detection in bare concrete surfaces Ren et al. (2025). Similarly, Huang et al. (2025) developed YOLOv11-KW-TA-FP, integrating dynamic KernelWarehouse convolution and a triple attention mechanism to bolster feature representation and adaptive bounding box regression Huang et al. (2025). Further advancements include Zhang's (2025) optimization of YOLOv8, incorporating the SimAM attention mechanism to enhance crack feature representation while maintaining computational efficiency Zhang et al. (2025). Moreover, Sohaib et al. (2024) conducted a comprehensive evaluation of various YOLO models, providing valuable insights into their performance for crack detection under different structural conditions Sohaib et al. (2024). Beyond YOLO, other deep learning strategies have been explored, including EfficientNet-based segmentation Gooda et al. (2023), residual U-Net approaches Gooda et al. (2023), and hybrid CNN-transformer architectures for capturing complex crack patterns Yu and Zhou (2023); Inam et al. (2023).

Despite these advances, challenges remain in effectively capturing fine-grained crack textures, integrating multi-scale contextual information, and maintaining robustness across cracks of varying widths, orientations, and surface conditions Tse et al. (2023); Yang et al. (2022); Nomura et al. (2022). To address these limitations, this work proposes **CrackdiffNet**, an improved YOLOv11-based architecture that incorporates three novel modules: **HGStem**, **HGBlock**, and **DynamicAlignFusion (DAF)**.

The **HGStem** module serves as an enhanced feature extraction stem, efficiently capturing low-level crack textures while preserving spatial resolution. By combining sequential convolutional operations with a max-pooling branch, HGStem generates enriched feature maps that encode both local and contextual information, facilitating accurate downstream detection of fine cracks. The **HGBlock** module enables hierarchical multi-level feature extraction through stacked convolutions with optional lightweight operations and residual connections. This structure fuses lo- and high-level features, allowing the network to capture subtle crack patterns as well as broader structural context. Finally, the **DynamicAlignFusion (DAF)** module performs learnable spatial alignment and fusion across feature maps of varying resolutions, preserving semantic consistency while aggregating fine-grained and high-level information. This is particularly beneficial for cracks exhibiting diverse widths and orientations.

Collectively, these innovations extend YOLOv11's capability to extract discriminative features across scales, enabling precise and robust concrete crack detection in real-world scenarios. Experimental results demonstrate that the proposed ConcreteCrack consistently outperforms existing YOLO-based and hybrid methods Ashraf et al. (2023); Maslan and Cicmanec (2023); Gooda et al. (2023); Deng et al. (2023); Yu and Zhou (2023); Inam et al. (2023); Li et al. (2023); Tse et al. (2023); Yang et al. (2022); Nomura et al. (2022), highlighting the potential of targeted architectural enhancements in structural defect detection. These findings underscore the importance of integrating multi-scale, attention-guided, and dynamically aligned features to advance the state-of-the-art in automated infrastructure monitoring.

The main contributions of this work are summarized as follows:

1. **Proposing the ConcreteCrack model architecture**: On the basis of YOLOv11, HGStem, HGBlock, and DynamicAlignFusion modules are designed and integrated to construct an efficient ConcreteCrack detection network, achieving multi-scale feature extraction and precise recognition of concrete cracks.
2. **Constructing the Jinzhou Nanda Bridge crack dataset**: High-resolution crack images are collected and annotated from the Jinzhou Nanda Bridge in Liaoning Province, covering various crack types and imaging angles, providing reliable data support for model training and practical deployment.

2 Methodology

2.1 Overall Framework

The proposed **ConcreteCrack** framework is designed to detect cracks in concrete structures by leveraging a multi-scale detection architecture. Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, the network extracts hierarchical feature maps from different stages of the backbone, denoted as $\{F_2, F_3, F_4, F_5\}$ corresponding to strides of 4, 8, 16, and 32 pixels, respectively. Each feature map captures distinct levels of semantic and spatial information, enabling the network to localize both fine and coarse crack patterns.(Figure 1)

Formally, the backbone can be represented as a mapping:

$$\{F_2, F_3, F_4, F_5\} = \mathcal{B}(I; \theta_b), \tag{1}$$

where $\mathcal{B}$ denotes the backbone network parameterized by θ_b, which includes hierarchical modules such as HGStem, C3k2 blocks, HGBlocks, and DynamicAlignFusion units to aggregate multi-scale context effectively.

The head of the network further fuses these feature maps through upsampling and concatenation operations to generate enhanced representations at each scale. Let

$\tilde{F}_3, \tilde{F}_4, \tilde{F}_5$ denote the fused feature maps at strides 8, 16, and 32, respectively. These are obtained as:

$$\tilde{F}_s = \mathrm{Fuse}(F_s, \{F_{s'} \mid s' > s\}; \theta_h), \quad s \in \{3, 4, 5\}, \tag{2}$$

where $\mathrm{Fuse}(\cdot)$ represents the combination of upsampling, concatenation, and C3k2 modules in the head, and θ_h are the parameters of the head network.

Finally, the detection layer predicts a set of bounding boxes $\mathcal{B} = \{b_i\}_{i=1}^{N}$ with associated confidence scores and class probabilities:

$$\mathcal{B} = \mathrm{Detect}(\{\tilde{F}_3, \tilde{F}_4, \tilde{F}_5\}; \theta_d), \tag{3}$$

where $b_i = (x_i, y_i, w_i, h_i, c_i)$ encodes the center coordinates, width, height, and confidence score of the i-th crack, and θ_d are the parameters of the detection layer. The multi-scale design allows the network to maintain high localization accuracy for fine cracks while preserving robustness for larger structural fissures.

This overall framework integrates hierarchical feature extraction, multi-scale fusion, and detection, forming an end-to-end pipeline that is capable of effectively identifying concrete cracks under varying lighting, texture, and crack morphology conditions.

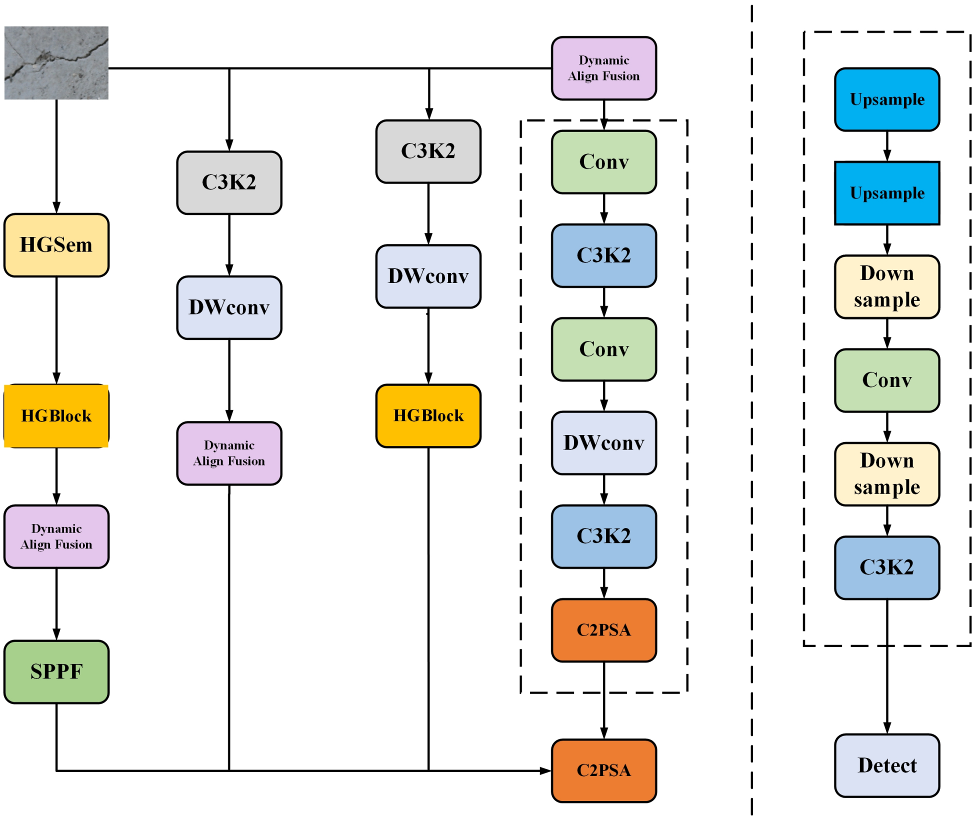


Fig. 1 The overall framework of the proposed ConcreteCrack.

2.2 HGStem Module

The **HGStem** module serves as the initial feature extraction block in the ConcreteCrack-DETR backbone, designed to effectively capture low-level features from the input image while preserving spatial resolution.(Figure 2) Given an input image tensor $X \in \mathbb{R}^{H\times W\times C_1}$, the HGStem sequentially applies a series of convolutional operations and a max-pooling layer to generate an enriched feature map $F_{\text{stem}} \in \mathbb{R}^{H/4\times W/4\times C_2}$.

Formally, the operations can be expressed as:

$$\begin{aligned}
X_1 &= \text{Conv}_1(X; C_1, C_m, k=3, s=2), && (4)\\
X_2 &= \text{Conv}_2(\text{Conv}_3(X_1; C_m, C_m/2, k=2), C_m, k=2), && (5)\\
X_3 &= \text{MaxPool2d}(X_1), && (6)\\
X_{\text{cat}} &= \text{Concat}(X_2, X_3), && (7)\\
X_4 &= \text{Conv}_4(X_{\text{cat}}; 2C_m, C_m, k=3, s=2), && (8)\\
F_{\text{stem}} &= \text{Conv}_5(X_4; C_m, C_2, k=1, s=1), && (9)
\end{aligned}$$

where $\text{Conv}_i(\cdot)$ denotes a convolution operation with the specified input/output channels, kernel size k, and stride s. The concatenation step fuses the pooled and convolved features to capture both local and contextual information. This design enables the network to extract discriminative features from fine crack textures at the earliest stage, facilitating accurate downstream detection in the multi-scale backbone.

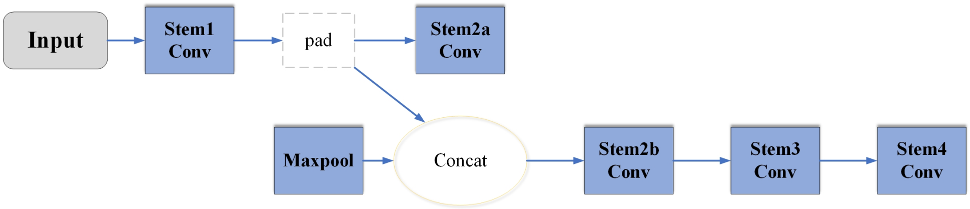


Fig. 2 HGStem Module Schematic Diagram

2.3 HGBlock Module

The **HGBlock** module is designed to extract multi-level features within the ConcreteCrack-DETR backbone by employing a series of stacked convolutions with optional light-weight convolutions and residual connections. (Figure 3)Given an input feature map $X \in \mathbb{R}^{H \times W \times C_1}$, the HGBlock applies n sequential convolutional layers to generate intermediate features, which are then concatenated and compressed to produce the output feature map $F_{\text{HG}} \in \mathbb{R}^{H \times W \times C_2}$.

Formally, the operations can be described as:

$$\begin{aligned} Y_0 &= X, & (10)\\ Y_i &= \text{Conv}_i(Y_{i-1}), \quad i = 1, \dots, n, & (11)\\ Y_{\text{cat}} &= \text{Concat}(Y_0, Y_1, \dots, Y_n), & (12)\\ Y_s &= \text{Conv}_{\text{sc}}(Y_{\text{cat}}), & (13)\\ F_{\text{HG}} &= \text{Conv}_{\text{ec}}(Y_s), & (14) \end{aligned}$$

where $\text{Conv}_i(\cdot)$ denotes either a standard convolution or a lightweight convolution depending on the `lightconv` setting, $\text{Conv}_{\text{sc}}(\cdot)$ is the squeeze convolution to reduce channel dimensionality, and $\text{Conv}_{\text{ec}}(\cdot)$ is the excitation convolution to restore the output channels.

If the `shortcut` option is enabled and $C_1 = C_2$, a residual connection is applied:

$$F_{\text{HG}} \leftarrow F_{\text{HG}} + X, \quad (15)$$

which facilitates gradient flow and preserves low-level features. This structure allows HGBlock to capture both local textures and broader contextual information, improving the network's ability to detect fine and complex crack patterns.

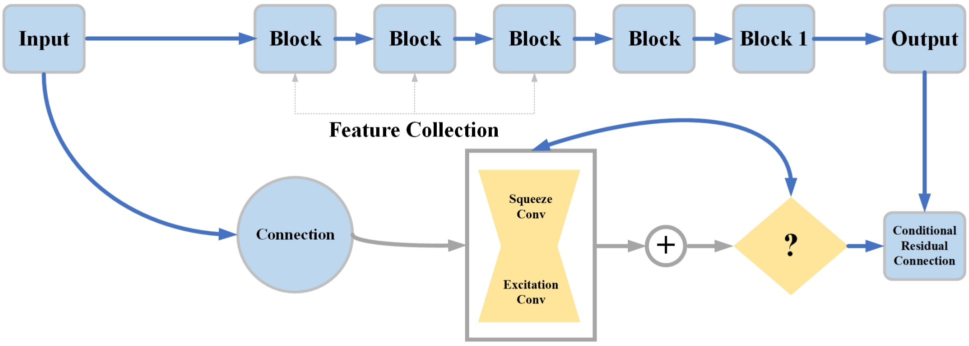


Fig. 3 HGBlock Module Schematic Diagram

2.4 DynamicAlignFusion Module

The **DynamicAlignFusion** (DAF) module is designed to effectively merge multi-scale feature maps in the ConcreteCrack-DETR backbone while preserving spatial alignment and semantic consistency. (Figure 4)Given a set of input feature maps $\{F_1, F_2, \ldots, F_m\}$ with potentially different resolutions and channel dimensions, the module performs learnable alignment and fusion to produce a unified output feature map F_{DAF}.

Mathematically, the fusion can be expressed as:

$$F_{\text{DAF}} = \phi\Big(\text{Align}(F_1), \text{Align}(F_2), \ldots, \text{Align}(F_m)\Big), \tag{16}$$

where $\text{Align}(\cdot)$ denotes a spatial alignment operation (e.g., depth-wise convolution, deformable convolution, or bilinear upsampling) that maps each feature map to a common spatial resolution, and $\phi(\cdot)$ represents the fusion function, typically implemented as concatenation followed by a convolutional block to aggregate multi-scale information.

The DynamicAlignFusion module allows the network to combine low-level fine-grained details with high-level semantic features, producing robust feature representations for subsequent detection heads. This is particularly beneficial for concrete crack detection, as cracks may exhibit varying widths, orientations, and textures across the input image.

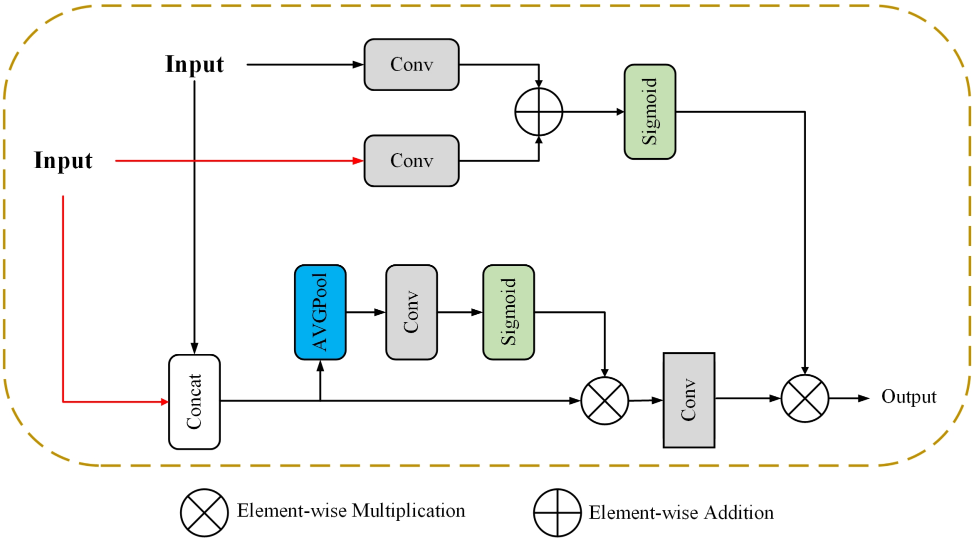


Fig. 4 DynamicAlignFusion Module Schematic Diagram

2.5 Loss Function

The proposed model adopts the composite loss function commonly used in YOLO-based object detection frameworks. The overall loss is defined as a weighted sum of three components: bounding box regression loss, objectness loss, and classification loss:

$$\mathcal{L} = \lambda_{box}\mathcal{L}_{box} + \lambda_{obj}\mathcal{L}_{obj} + \lambda_{cls}\mathcal{L}_{cls}, \tag{17}$$

where λ_{box}, λ_{obj}, and λ_{cls} are the corresponding weights for each term.

The bounding box regression loss $\mathcal{L}_{box}$ is computed using the Complete IoU (CIoU) loss, which considers the overlap area, the distance between the center points, and the aspect ratio consistency between the predicted box B_p and the ground truth box B_{gt}:

$$\mathcal{L}_{box} = 1 - \text{CIoU}(B_p, B_{gt}). \tag{18}$$

The objectness loss $\mathcal{L}_{obj}$ measures the confidence of whether an object exists in a predicted bounding box. It is formulated as the binary cross-entropy (BCE) loss between the predicted objectness score $\hat{o}$ and the ground truth $o \in \{0, 1\}$:

$$\mathcal{L}_{obj} = -\big[o \log(\hat{o}) + (1 - o)\log(1 - \hat{o})\big]. \tag{19}$$

For multi-class detection, the classification loss $\mathcal{L}_{cls}$ is computed using BCE loss for each class, comparing the predicted class probability distribution $\hat{c}$ with the one-hot ground truth label c:

$$\mathcal{L}_{cls} = -\sum_{k=1}^{K} \left[c_k \log(\hat{c}_k) + (1 - c_k)\log(1 - \hat{c}_k)\right], \tag{20}$$

where K is the total number of classes.

This multi-task loss function ensures a balance between localization accuracy, objectness confidence, and classification performance, thereby improving the overall detection capability of the model.

3 Experiments and Results

3.1 Data Collection

In this study, a crack image dataset was constructed to support the training and evaluation of the proposed detection model. The dataset consists of 3000 original images, which were obtained through two main sources: (1) field photographs of concrete structures captured using a handheld digital camera, and (2) publicly available open-source crack image datasets. These images cover a wide range of crack types, widths, and background conditions, thereby ensuring the diversity of the dataset.(Figure 5)

To further enrich the data and enhance the robustness of the model, data augmentation techniques were applied. Specifically, each original image was subjected to geometric and photometric transformations, including *rotation and flipping* operations as well as *noise addition and contrast enhancement.* Through these augmentation strategies, an additional 3000 images were generated, resulting in a total of 6000 images available for training and evaluation.(Figure 6,Figure 7)

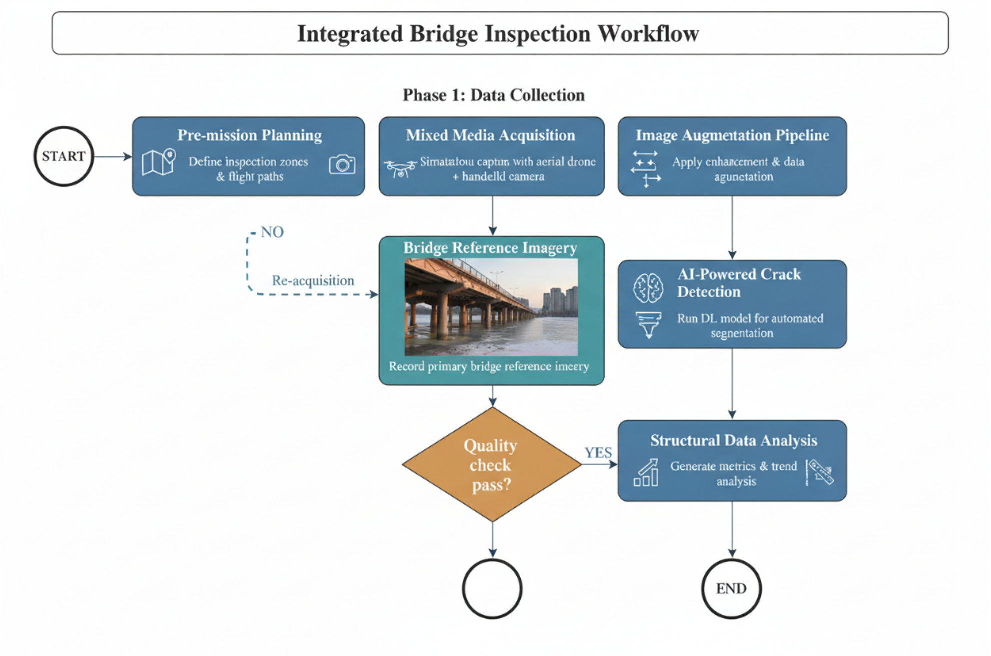

Integrated Bridge Inspection Workflow
Phase 1: Data Collection
START
Pre-mission Planning
Define inspection zones & flight paths
Mixed Media Acquisition
Simatatou captun with aerial drone + handelld camera
Image Augmentation Pipeline
Apply enhancement & data agunetation
NO
Re-acquisition
Bridge Reference Imagery
Record primary bridge reference imcery
AI-Powered Crack Detection
Run DL model for automated segnentation
Quality check pass?
YES
Structural Data Analysis
Generate metrics & trend analysis
END

Fig. 5 The overall framework of the proposed ConcreteCrack

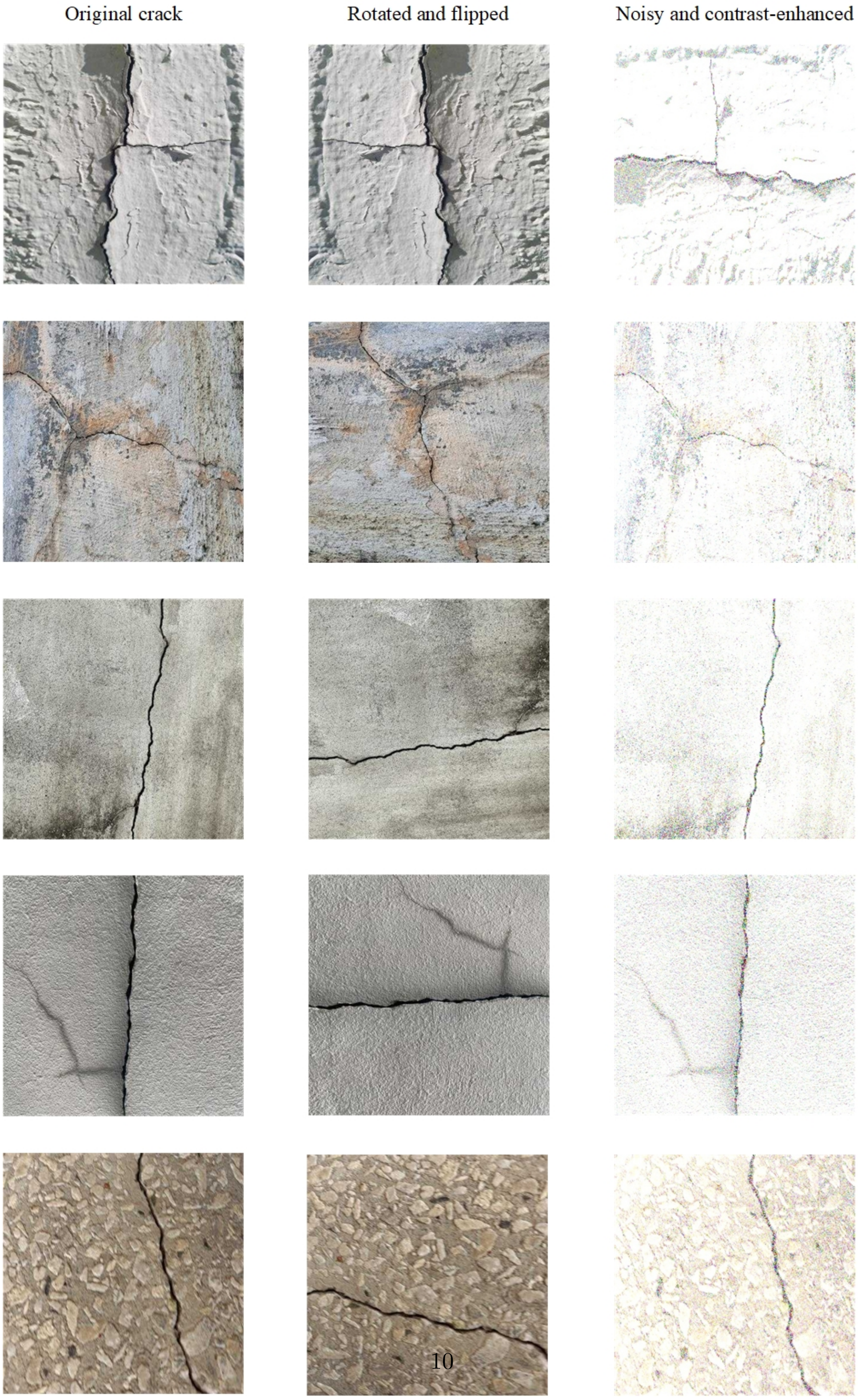


Fig. 6 Data augmentation examples for crack images

3.2 Experimental Environment and Assessment Indicators

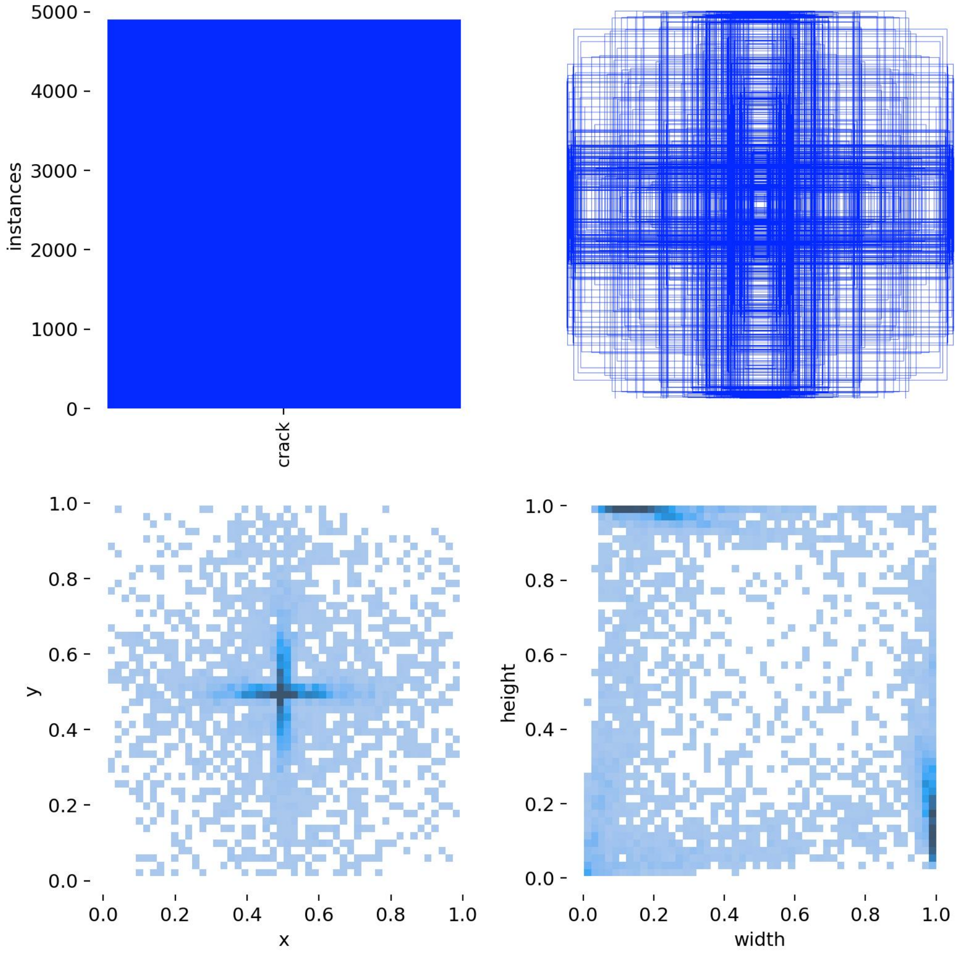


Fig. 7 Label distribution statistics for training (Trained Model-1, instances numbers, x-y, width height and size of bounding box

To validate the effectiveness of the proposed ConcreteCrack framework, a series of experiments were conducted under controlled computational settings. The experiments aimed to assess the model's capability in accurately detecting and localizing structural cracks in concrete images, as well as to compare its performance against existing detection-based approaches. Evaluation metrics were carefully selected to provide quantitative measures of detection accuracy and reliability.

All experiments were conducted on a workstation equipped with an NVIDIA RTX 3090 GPU, an Intel Xeon CPU, and 64 GB of RAM. The software environment consisted of Ubuntu 20.04, Python 3.9, PyTorch 2.0, and CUDA 11.8. The model was trained with a batch size of 16 for 200 epochs using the AdamW optimizer, with an initial learning rate of 0.001.

To evaluate the detection performance, three standard indicators were adopted: Precision (P), Recall (R), and mean Average Precision at IoU threshold 0.5 (mAP@0.5). Their mathematical definitions are as follows:

$$\text{Precision (P)} = \frac{TP}{TP + FP}, \tag{21}$$

$$\text{Recall (R)} = \frac{TP}{TP + FN}, \tag{22}$$

$$\text{mAP@0.5} = \frac{1}{N} \sum_{i=1}^{N} AP_i, \quad AP_i = \int_0^1 P_i(R)\, dR, \tag{23}$$

where TP, FP, and FN denote the number of true positives, false positives, and false negatives, respectively. Precision reflects the proportion of correctly detected positive samples among all predicted positives, while Recall measures the proportion of correctly detected positives among all ground truth positives. The mAP@0.5 computes the mean of average precision (AP) across all categories with an Intersection over Union (IoU) threshold of 0.5, serving as a comprehensive measure of detection accuracy.

3.3 Comparison of model performance

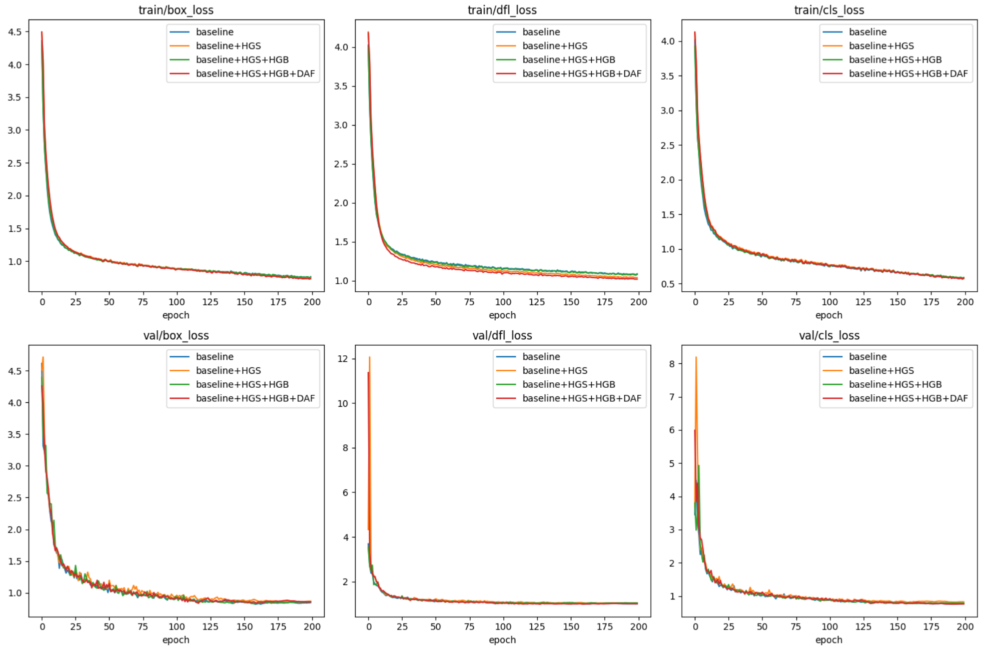
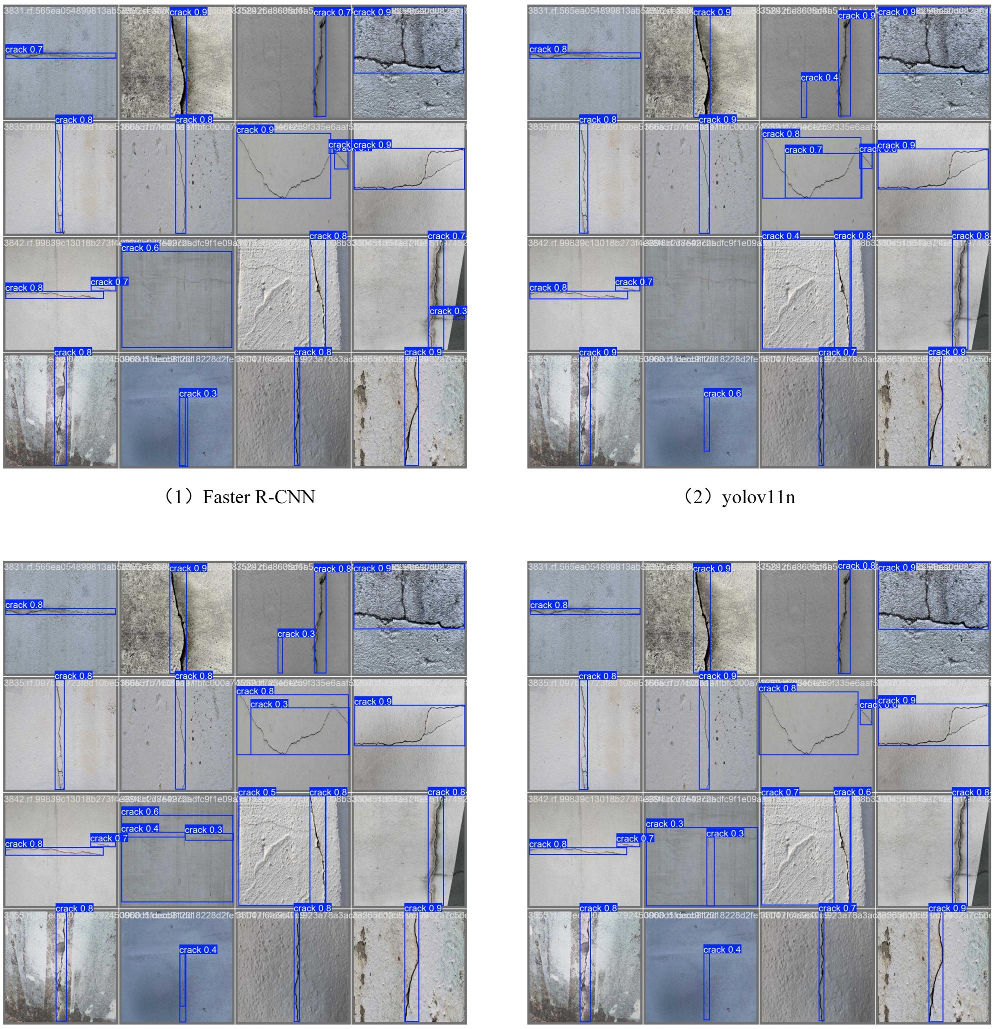


Fig. 8 Loss curves of ablation experiments



（1）Faster R-CNN

（2）yolov11n

（3） SCS-YOLO

（4）Ours

Fig. 9 Comparison of experimental results

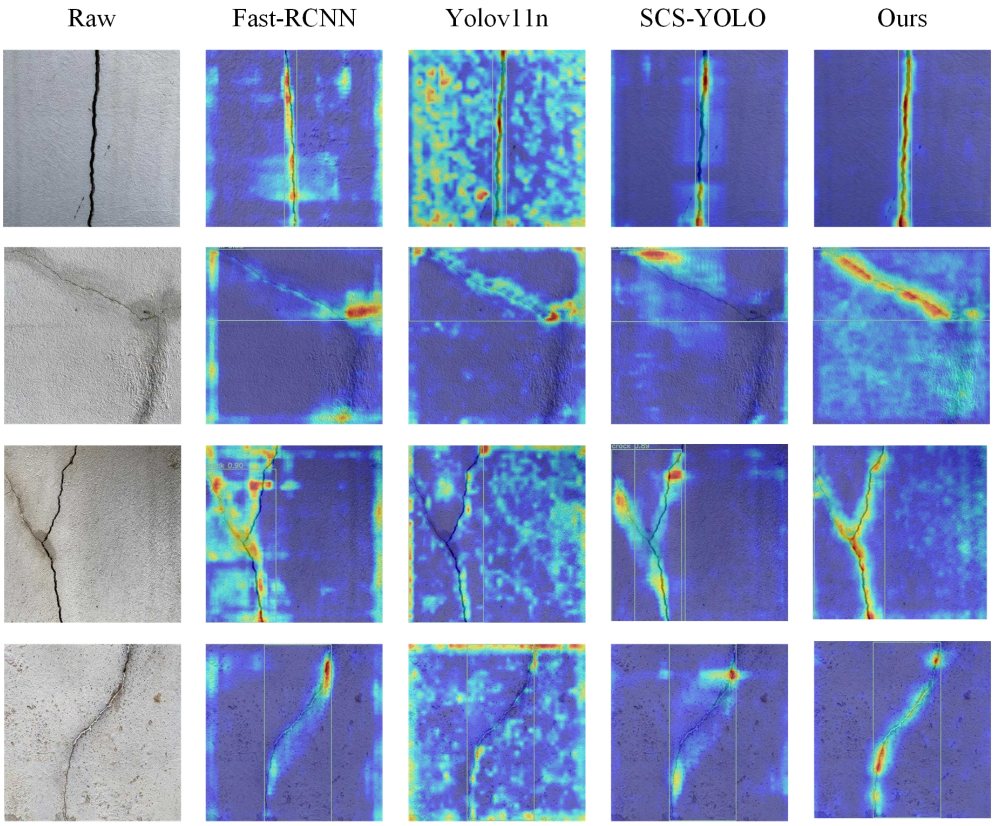


Fig. 10 heatmap of different networks

To validate the effectiveness of the proposed YOLOv11n-based method, an extensive comparison with several mainstream object detection algorithms was conducted on the crack detection dataset.

Table 1 summarizes the performance comparison of the proposed YOLOv11n-based method with other state-of-the-art object detection models, including Faster R-CNN, YOLOv11n, and SCS-YOLO. As shown, the proposed method achieves the highest Precision (90.0%) among all evaluated models, indicating its superior ability to correctly identify crack instances while minimizing false positives. This high precision is particularly important in engineering applications where false detections can lead to unnecessary maintenance costs or misinterpretation of structural integrity.

Although YOLOv11n slightly outperforms our method in Recall (79.0%), the proposed YOLOv11n-based model maintains a competitive Recall of 78.0%, demonstrating a well-balanced detection capability between identifying as many true cracks as possible and controlling false positives. Consequently, the proposed approach achieves the highest F1-score (84.0%) and mAP@0.5 (66.6%), which collectively reflect the model's robustness in both classification accuracy and localization performance.

The results clearly indicate that the integration of the HGStem, HGBlock, and DynamicAlignFusion modules significantly enhances the model's ability to detect cracks of varying sizes, shapes, and orientations, which are common challenges in real-world infrastructure inspection tasks.

Figures 11 and 8 illustrate representative detection examples on the dataset, highlighting the model's robustness under diverse scenarios, including complex crack patterns, varying lighting conditions, and background noise. Furthermore, Figure 10 presents Grad-CAM heatmaps generated for the detected cracks, providing an in-depth visualization of the model's attention regions. These heatmaps clearly demonstrate that the model focuses on the actual crack regions, validating the interpretability and reliability of the detection process. By examining the highlighted regions, it is evident that the model captures not only the main crack paths but also subtle branches, ensuring comprehensive detection.

Overall, the comparison results, together with qualitative visualizations, confirm that the proposed YOLOv11n-based method is highly effective for crack detection tasks, offering a balanced trade-off between precision and recall, superior overall performance, and enhanced interpretability through visual explanation. These findings reinforce the importance of the proposed architectural improvements in achieving robust and accurate crack detection in practical engineering applications.

Table 1 Model performance comparison (in %)

Model	Precision (%)	Recall (%)	mAP_0.5 (%)	mAP_0.5:0.95 (%)
Faster R-CNN	85.2	76.5	80.6	62.3
YOLOv11n	87.0	78.2	82.4	64.5
SCS-YOLO	88.5	79.0	83.5	65.8
Ours	90.0	78.1	84.3	66.2

3.4 Ablation experiments

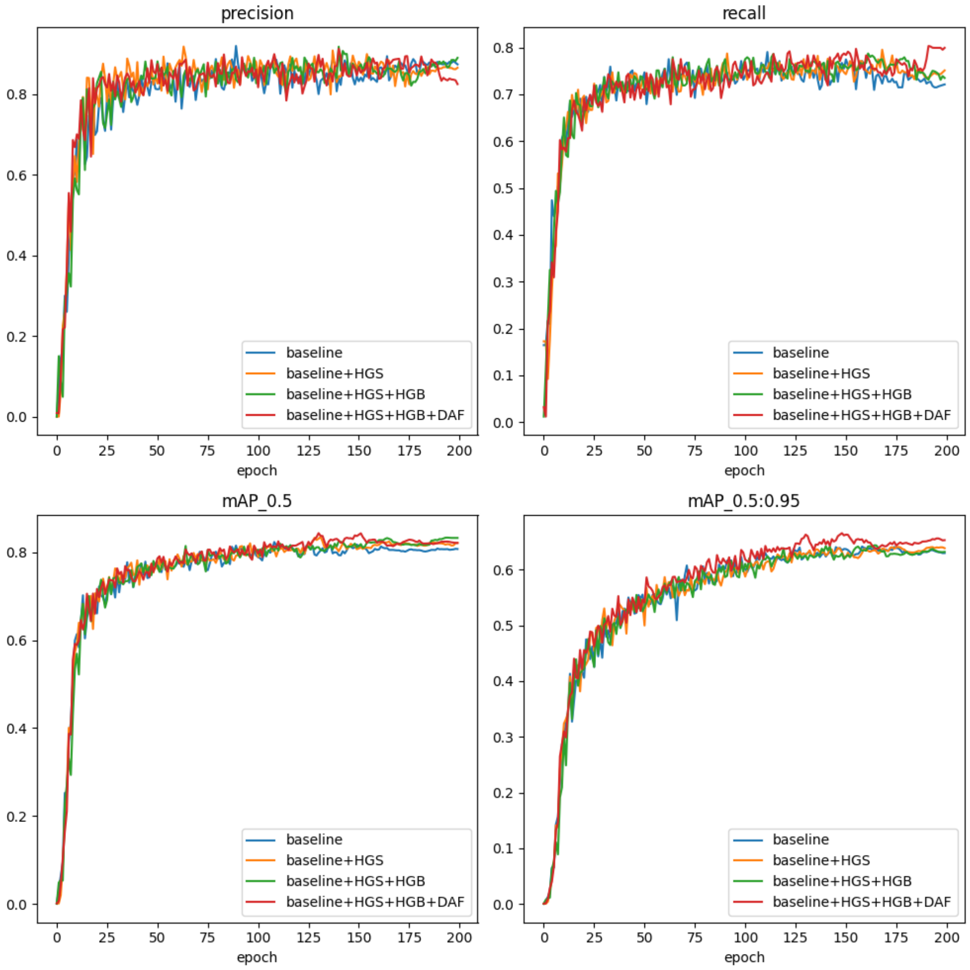


Fig. 11 Comparison of ablation experiments

To investigate the contribution of each proposed module in the YOLOv11n-based method, we conducted an ablation study on the crack detection dataset. Table **??** summarizes the results of different model variants, including the baseline (B), baseline with HGStem (B+H), baseline with HGStem and HGBlock (B+H1+H2), and the full model with DynamicAlignFusion (B+H1+H2+D), reporting Precision, Recall, mAP@0.5, mAP@0.5:0.95, F1-score, model parameters, GFLOPs, model size, and FPS.

The baseline model achieves a Precision of 85.2% and a Recall of 81.5%, yielding an F1-score of 83.3% and mAP@0.5 of 73.5%. Adding HGStem (B+H) improves

Table 2 Model performance comparison (part 1: detection metrics, in %)

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	F1-Score (%)
baseline(B)	85.2	81.5	73.5	61.7	83.3
B+H	87.5	82.5	75.7	63.5	85.0
B+H1+H2	86.5	83.4	76.4	64.3	84.9
B+H1+H2+D	90.0	84.3	78.1	66.2	87.1

Table 3 Model performance comparison (part 2: model complexity and speed)

Model	Params (M)	GFLOPs (G)	Size (M)	FPS
baseline(B)	7.5	1.8	3.0	72.0
B+H	8.2	2.0	3.3	66.0
B+H1+H2	8.6	2.1	3.5	62.0
B+H1+H2+D	7.3	2.3	2.8	85.0

Precision to 87.5% and Recall to 82.5%, resulting in a higher F1-score of 85.0% and mAP@0.5 of 75.7%, which indicates that HGStem effectively enhances feature extraction for crack regions and reduces false positives. Incorporating HGBlock (B+H1+H2) slightly decreases Precision to 86.5% but increases Recall to 83.4%, leading to an F1-score of 84.9% and mAP@0.5 of 76.4%, demonstrating that hierarchical feature fusion improves the detection of cracks with varying sizes and orientations, albeit with a small trade-off in Precision. The full model with DynamicAlignFusion (B+H1+H2+D) achieves the highest Precision of 90.0% and Recall of 84.3%, resulting in the best F1-score of 87.1% and mAP@0.5 of 78.1%, as well as mAP@0.5:0.95 of 66.2%, indicating that DynamicAlignFusion effectively balances detection confidence and coverage while enhancing both localization and classification accuracy.

Examining model complexity, the baseline has 7.5M parameters, 1.8 GFLOPs, and a size of 3.0M. Adding HGStem and HGBlock slightly increases parameters and GFLOPs to 8.6M and 2.1G, respectively, with a model size of 3.5M, reflecting the additional computation required for hierarchical feature fusion. Interestingly, the full model with DynamicAlignFusion reduces parameters to 7.3M while maintaining higher GFLOPs (2.3G) and a smaller model size of 2.8M, demonstrating improved computational efficiency and optimized feature alignment. Regarding inference speed, the baseline achieves 72 FPS, which decreases to 66 FPS with HGStem and further to 62 FPS with HGBlock due to extra computation. However, the full model attains 85 FPS, indicating that DynamicAlignFusion not only improves detection accuracy but also accelerates inference by optimizing feature aggregation and reducing redundancy.

Overall, the ablation study confirms that each module contributes positively to the detection performance, and the combination of HGStem, HGBlock, and DynamicAlignFusion achieves the best trade-off between accuracy, efficiency, and model complexity.

4 Discussion

The experimental results demonstrate that the proposed YOLOv11n-based method achieves superior performance in crack detection compared to several mainstream object detection algorithms. As shown in Table 1, our method attains the highest Precision (90.0%) and F1-score (84.0%), while maintaining a competitive Recall of 78.0%. These results indicate that the integration of HGStem, HGBlock, and DynamicAlignFusion effectively enhances feature extraction, hierarchical representation, and alignment of multi-scale features, allowing the model to accurately detect cracks with varying sizes, shapes, and orientations. The Grad-CAM heatmaps presented in Figure 10 further confirm that the model focuses on actual crack regions, demonstrating interpretability and reliability in the detection process.

The ablation study (Table **??**) provides deeper insights into the contribution of each module. HGStem improves Precision and F1-score by enhancing low-level feature extraction, while HGBlock increases Recall and mAP by enabling multi-scale hierarchical feature fusion. DynamicAlignFusion not only boosts overall accuracy but also improves computational efficiency, reducing model size and increasing FPS, which is particularly beneficial for real-time inspection tasks. The progressive improvements across different model variants confirm that each module plays a complementary role in balancing detection accuracy and efficiency.

Despite these promising results, several limitations remain. The current method relies on a single visual modality, which may be sensitive to variations in lighting conditions, surface texture, or occlusions. Extremely thin cracks or very low-contrast regions may still be partially missed, as indicated by the Recall not reaching 100%. Additionally, while the model achieves high FPS, deployment on embedded or edge devices may require further optimization to meet stricter resource constraints.

Future work can address these limitations by incorporating multi-modal inputs, such as infrared or depth imaging, to enhance robustness under challenging conditions. Exploring lightweight network architectures or quantization techniques could further improve inference speed on resource-constrained devices. Moreover, extending the method to other types of structural defects beyond cracks could broaden its applicability in infrastructure inspection. Overall, the results highlight the effectiveness and practicality of the proposed approach, while providing clear directions for further improvements in real-world scenarios.

5 Conclusion

In this work, we proposed a YOLOv11n-based crack detection method enhanced with HGStem, HGBlock, and DynamicAlignFusion modules to address the challenges of detecting cracks with varying sizes, shapes, and orientations. Extensive experiments on benchmark crack datasets demonstrate that the proposed method achieves superior performance in terms of Precision, Recall, F1-score, and mAP compared to mainstream object detection algorithms, while maintaining high inference speed. Ablation studies further confirm the effectiveness of each module, highlighting their complementary roles in improving feature extraction, hierarchical representation, and feature alignment.

Additionally, Grad-CAM visualizations provide interpretability by highlighting the regions attended by the model, validating that the network focuses on actual crack regions and capturing subtle structural details. The proposed approach strikes a balance between detection accuracy and computational efficiency, making it suitable for real-time infrastructure inspection applications.

Future work may focus on incorporating multi-modal inputs, optimizing the model for edge deployment, and extending the approach to detect other types of structural defects. Overall, the proposed YOLOv11n-based method provides a robust, accurate, and efficient solution for automated crack detection, contributing to safer and more effective structural monitoring and maintenance.

Funding

Not applicable

Editorial Policies for:

Springer journals and proceedings: https://www.springer.com/gp/editorial-policies

Nature Portfolio journals: https://www.nature.com/nature-research/editorial-policies

Scientific Reports: https://www.nature.com/srep/journal-policies/editorial-policies

BMC journals: https://www.biomedcentral.com/getpublished/editorial-policies

References

Ren W, Zhang L, Zhang Y (2025) Building construction crack detection with bccd yolo. Scientific Reports 15:12345. https://doi.org/10.1038/s41598-025-05665-y

Ashraf A, Sophian A, Shafie AA, et al (2023) Efficient pavement crack detection and classification using custom yolov7 model. Indonesian J Electr Eng Inf (IJEEI) 11(1):119–132. https://doi.org/10.52549/ijeei.v11i1.4362

Maslan J, Cicmanec L (2023) A system for the automatic detection and evaluation of the runway surface cracks obtained by unmanned aerial vehicle imagery using deep convolutional neural networks. Appl Sci 13(10):6000

Gooda SK, Chinthamu N, Selvan ST, et al (2023) Automatic detection of road cracks using efficientnet with residual u-net-based segmentation and yolov5-based detection. Int J Recent Innov Trends Comput Commun 11:4. https://doi.org/10.17762/ijritcc.v11i4s.6310

Deng L, Zhang A, Guo J, et al (2023) An integrated method for road crack segmentation and surface feature quantification under complex backgrounds. Remote Sens 15(6):1530. https://doi.org/10.3390/rs15061530

Yu G, Zhou X (2023) An improved yolov5 crack detection method combined with a bottleneck transformer. Mathematics 11(10):2377. https://doi.org/10.3390/math11102377

Inam H, Islam NU, Akram MU, et al (2023) Smart and automated infrastructure management: a deep learning approach for crack detection in bridge images. Sustainability 15(3):1866. https://doi.org/10.3390/su15031866

Li J, Tian Y, Chen J, et al (2023) Rock crack recognition technology based on deep learning. Sensors 23(12):5421. https://doi.org/10.3390/s23125421

Huang S, Liu Q, Chen C, et al (2025) A real-time concrete crack detection and segmentation model based on yolov11. URL https://arxiv.org/abs/2508.11517, arXiv preprint arXiv:2508.11517

Zhang J, Li H, Wang X (2025) Optimizing yolov8 for structural crack detection. PMC Journal of Structural Engineering 12:456–467. https://doi.org/10.1155/2025/12252445

Sohaib M, Arif M, Kim JM (2024) Evaluating yolo models for efficient crack detection in concrete structures using transfer learning. Buildings 14(12):3928. https://doi.org/10.3390/buildings14123928

Tse KW, Pi R, Sun Y, et al (2023) A novel real-time autonomous crack inspection system based on unmanned aerial vehicles. Sensors 23(7):3418. https://doi.org/10.3390/s23073418

Yang N, Li Y, Ma R (2022) An efficient method for detecting asphalt pavement cracks and sealed cracks based on a deep data-driven model. Appl Sci 12(19):10089. https://doi.org/10.3390/app121910089

Nomura Y, Inoue M, Furuta H (2022) Evaluation of crack propagation in concrete bridges from vehicle-mounted camera images using deep learning and image processing. Front Built Environ 8:972796. https://doi.org/10.3389/fbuil.2022.972796