

Article

Segmentation of Stone Slab Cracks Based on an Improved YOLOv8 Algorithm

Qitao Tian ^{1,2} , Runshu Peng ^{1,2} and Fuzeng Wang ^{1,2,*}

¹ Institute of Manufacturing Engineering, Huaqiao University, Xiamen 361021, China; 23014080083@stu.hqu.edu.cn (Q.T.); 22014080033@stu.hqu.edu.cn (R.P.)

² Nan'an-HQU Institute of Stone Industry Innovations Technology, Quanzhou 362342, China

* Correspondence: wangfuzeng@hqu.edu.cn; Tel.: +86-180-3000-1513

Abstract

To tackle the challenges of detecting complex cracks on large stone slabs with noisy textures, this paper presents the first domain-optimized framework for stone slab cracks, an improved semantic segmentation model (YOLOv8-Seg) synergistically integrating U-NetV2, DSConv, and DySample. The network uses the lightweight U-NetV2 backbone combined with dynamic feature recalibration and multi-scale refinement to better capture fine crack details. The dynamic up-sampling module (DySample) helps to adaptively reconstruct curved boundaries. In addition, the dynamic snake convolution head (DSConv) improves the model's ability to follow irregular crack shapes. Experiments on the custom-built ST stone crack dataset show that YOLOv8-Seg achieves an mAP@0.5 of 0.856 and an mAP@0.5–0.95 of 0.479. The model also reaches a mean intersection over union (MIoU) of 79.17%, outperforming both baseline and mainstream segmentation models. Ablation studies confirm the value of each module. Comparative tests and industrial validation demonstrate stable performance across different stone materials and textures and a 30% false-positive reduction in real production environments. Overall, YOLOv8-Seg greatly improves segmentation accuracy and robustness in industrial crack detection on natural stone slabs, offering a strong solution for intelligent visual inspection in real-world applications.

Keywords: crack segmentation; YOLOv8; U-NetV2; dynamic upsampling; snake convolution



Academic Editor: Victor Giurgiutiu

Received: 1 July 2025

Revised: 25 July 2025

Accepted: 31 July 2025

Published: 3 August 2025

Citation: Tian, Q.; Peng, R.; Wang, F. Segmentation of Stone Slab Cracks Based on an Improved YOLOv8 Algorithm. *Appl. Sci.* **2025**, *15*, 8610. <https://doi.org/10.3390/app15158610>

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

Copyright: © 2025 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Stone slabs are widely used as premium architectural finishing materials [1]. Surface cracks on these slabs not only affect aesthetics and quality grading, but can also pose safety risks and reduce service life. Traditional manual inspection methods are inefficient, prone to errors, and difficult to scale for large applications [2,3]. As a result, there is strong industrial demand for automated, high-precision, and real-time crack detection technologies [4]. In recent years, deep learning-based object detection and semantic segmentation techniques have undergone rapid development [5,6]. The YOLO (You Only Look Once) series has become a popular research focus in crack detection due to its end-to-end architecture and real-time performance [7–9]. Among them, YOLOv8 introduces structural improvements in the backbone, neck, and detection head, enhancing both accuracy and inference speed [10–12]. It has been widely applied in detecting cracks on structures and road surfaces.

The advantages of deep learning models in crack detection have been widely validated across infrastructure scenarios [13,14]. A YOLOv8-based framework for bridge crack

detection showed strong versatility [15]. Deng et al. [16] enhanced accuracy and robustness in road crack detection by combining the DETR framework with novel NRDQ and SQR modules, effectively addressing label conflicts and query competition. Ma et al. [17] developed a machine vision system for inspecting cracks on stamped parts. It features a high-resolution imaging module and a grayscale-based contrast enhancement algorithm to adaptively balance image contrast. While CNN-based methods have shown promise in tunnel lining crack detection, they still struggle with sample imbalance and real-time performance. To overcome this, Zhao et al. [18] proposed MPDENet, a lightweight model based on an improved MobileNetV2 backbone within PSPNet. By adding dilated convolutions and ECA attention, the model achieves real-time detection without sacrificing accuracy. Zhang et al. [19] introduced a YOLOv8-based lightweight system that performs well in tight spaces and low-light conditions, improving both efficiency and precision.

Zhou et al. [20] introduced an edge-processing branch that was added to the GELAN-Seg network. Researchers created a comprehensive masonry crack dataset using low-cycle reversed loading tests and manual annotations. For road crack instance segmentation, Chen et al. [21] enable real-time performance, with Hungarian matching used for temporal tracking in video streams. IPM transformation and camera calibration help to measure geometric properties. Shi et al. [22] boosted the nonlinear modeling capability by replacing traditional MLPs with Kolmogorov–Arnold Networks (KANs), while keeping efficient feature extraction. Song et al. [23] proposed a hybrid model combining GCN and DCN. Superpixel segmentation converts crack images into graphs, where GCN learns global topological features, and DCN extracts local spatial details, complementing each other.

Li et al. [10] introduced YOLOv8-GhostConv-SEV2, integrating GhostConv and SEV2 attention to improve feature extraction and suppress noise. This model achieves high accuracy in low-light pipeline crack detection. Zhang et al. [24] presented CrackAdaptNet, an end-to-end domain adaptation and semantic segmentation framework that bridges the gap between labeled training data and real-world applications. Fan [25] combined deep learning with spectral index-based image analysis to detect concrete cracks and evaluate severity, improving both accuracy and stability. Jing et al. [26] developed TopoM-CrackNet with a Topology Consistency Repair (TCR) module to preserve accurate crack shapes and correct topological differences. Ritzy et al. [27] reduced model size and overfitting by replacing Flatten layers with global average pooling (GAP), enhancing the ability to capture complex crack features. Ling et al. [28] designed an automatic detection method combining CNNs, sliding windows, and Otsu thresholding. It enables pixel-level segmentation and the accurate measurement of crack width and spalling areas on plastered masonry surfaces.

Although YOLOv8 has shown strong performance in various crack detection tasks, its application to stone slabs—with their high-texture and curved crack patterns—still faces several challenges. These include the following: (1) limited ability to model nonlinear curved cracks, (2) loss of edge details during upsampling, and (3) poor performance of existing instance segmentation methods on high-resolution, large-area cracks. Therefore, targeted improvements to YOLOv8 are needed to meet the complex demands of stone crack detection.

YOLOv8 is selected as the backbone due to its established deployment in industrial defect-detection pipelines and its favorable accuracy–throughput balance for high-resolution stone imagery. The architecture’s modularity further enables targeted, task-specific enhancements without compromising overall stability.

This study proposes an enhanced YOLOv8-Seg framework with the following optimizations: First, it replaces the backbone with a lightweight U-NetV2 architecture, introducing dynamic feature calibration and multi-scale fusion to improve micro-crack representation. Second, it incorporates dynamic snake convolution (DSConv) in the detec-

tion head to align more effectively with curved crack boundaries. Third, it introduces the DySample module in place of traditional upsampling, using learnable offsets to adapt to irregular crack structures while preserving edge details and reducing artifacts. The proposed method is trained on a high-resolution ST dataset of stone cracks. Ablation studies are conducted to evaluate the effectiveness of each module, and comparisons are made with mainstream models such as U-Net, DeepLabV3, and Attention U-Net. Results demonstrate that the model performs robustly across different stone textures and backgrounds, showing strong potential for industrial applications. Deployment in stone slab manufacturing plants demonstrates the method's efficacy, where integration with conveyor-belt systems achieves real-time micro-crack detection. This application reduces false rejections by approximately 30% for high-texture stones, significantly improving production yield.

2. Improved YOLOv8 Segmentation Network

2.1. U-NetV2

U-Net is a convolutional neural network based on an encoder–decoder architecture [29], as shown in Figure 1. Its core innovation lies in the symmetric contracting path, which captures global contextual information. While U-Net effectively captures object contours, its feature fusion strategy is relatively simple. In addition, traditional convolution has limited the ability to model complex textures. U-NetV2 builds upon the original U-Net structure and introduces improvements in three key areas: feature fusion efficiency, semantic alignment, and computational cost.

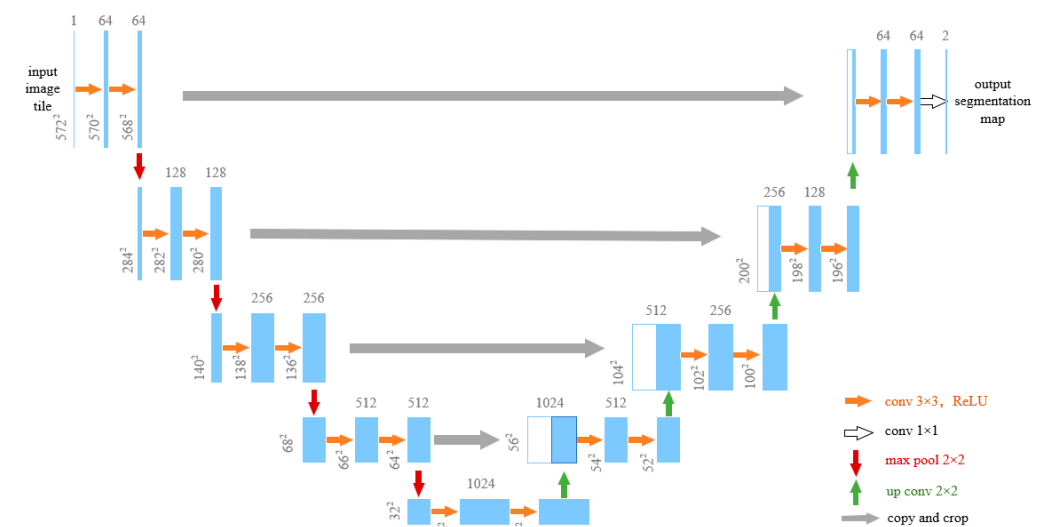


Figure 1. U-Net network structure.

First, a dynamic feature recalibration mechanism is introduced. It adaptively assigns channel weights and focuses on key spatial regions from the encoder, helping to reduce semantic misalignment across hierarchical features. Second, heterogeneous convolution replaces standard convolution. Residual connections are added to enhance gradient flow and improve sensitivity to fine structures and weak edges. In addition, U-NetV2 [30] adopts a progressive feature refinement strategy during decoding. A multi-level cross-scale interaction module nonlinearly fuses low-resolution semantic information with high-resolution detail features. This improves boundary accuracy in complex scenes. Through dynamic feature interaction and cross-scale reasoning, U-NetV2 achieves better feature decoupling and semantic consistency under limited computational resources.

As shown in Figure 2, the U-NetV2 architecture consists of an encoder, an SDI module, and a decoder. Given an input image $I \in \mathbb{R}^{H \times W \times C}$, the encoder first extracts feature representations. These features are then passed to the SDI module for further processing.

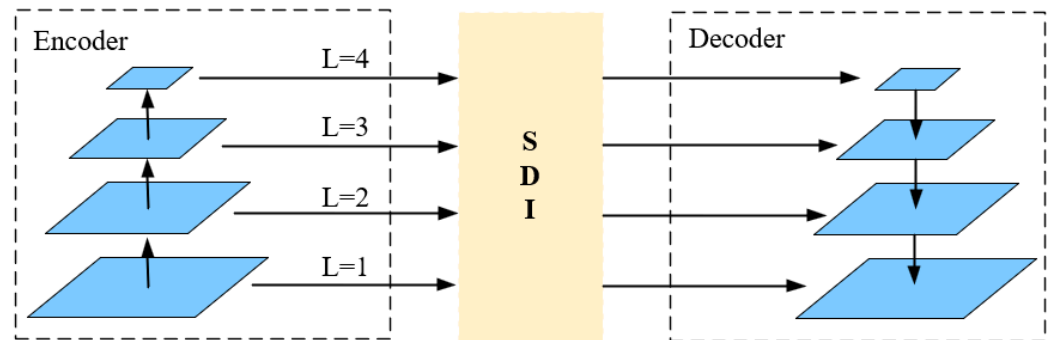


Figure 2. U-NetV2 network structure.

In the multi-level feature representations generated by the encoder, spatial and channel attention modules are jointly applied. This setup enables the network to dynamically aggregate local pixel-level details while modeling high-order semantic dependencies across channels. The dual-path attention fusion mechanism effectively enhances fine-grained spatial awareness and global channel responses in a complementary manner.

$$f_i^1 = \varphi_i^c(f_i^8(f_i^0)) \quad (1)$$

φ_i^c denotes the channel-wise dynamic recalibration function (implemented as a lightweight 1×1 convolution followed by sigmoid gating). ϕ_i^8 denotes the 8-directional spatial offset generator.

A 1×1 convolution is first introduced to compress the feature tensor along the channel dimension. The resulting low-dimensional feature tensor is denoted as f_i^2 . This tensor is then passed into the decoder. During each stage of decoding, f_i^2 serves as the reference signal. Feature tensors from other layers are resized to match the spatial resolution of f_i^2 , enabling effective cross-level fusion and scale normalization. The specific operations are as follows:

$$f_{ij}^3 = \begin{cases} G_D(f_j^2, (H_i, W_i)) & \text{if } j < i, \\ G_I(f_j^2) & \text{if } j = i, \\ G_U(f_j^2, (H_i, W_i)) & \text{if } j > i, \end{cases} \quad (2)$$

To enhance multi-scale feature fusion, a heterogeneous transformation group is designed. It consists of adaptive average pooling (D) for dynamic receptive field adjustment, identity mapping (I) for preserving original feature topology, and bilinear upsampling (U). These components work together to build a multi-granularity feature representation space. Based on this, depthwise separable convolution is applied to normalize the recalibrated feature tensor. The detailed process is as follows:

$$f_{ij}^4 = \theta_{ij}(f_{ij}^3) \quad (3)$$

θ_{ij} denotes the learnable channel-wise gating function applied at a spatial location (i, j) . Implemented via a 1×1 convolution followed by a sigmoid, it outputs a scalar weight $\theta_{ij} \in (0, 1)$ that dynamically re-scales the feature vector f_{ij}^3 , suppressing irrelevant channels while enhancing those most informative for crack representation.

Figure 3 illustrates the SDI network architecture. After all cross-level feature tensors undergo resolution alignment, cross-level feature interaction and information gain are achieved via the Hadamard product tensor's operation, as detailed below.

$$f_i^5 = H([f_{i1}^4, f_{i2}^4, \dots, f_{iM}^4]) \quad (4)$$

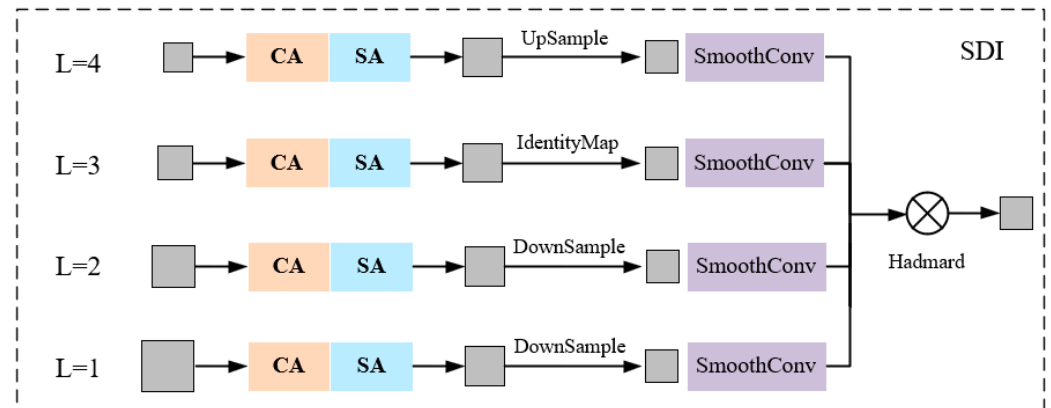


Figure 3. SDI network structure.

H denotes the Hadamard product. Finally, the mixed and enhanced feature stream is concatenated into the corresponding decoder stage. This process drives end-to-end resolution iterative optimization.

2.2. Dynamic Upsampling

Current mainstream detection frameworks still face optimization bottlenecks in cross-level feature integration efficiency. Their multi-scale receptive field feature interaction mechanisms struggle to achieve optimal coupling. Existing dynamic-convolution-based upsampling paradigms like CARAFE and FADE have enhanced feature reconstruction capabilities [31,32]. However, their parameterized dynamic kernel generation strategies often come with high computational complexity. To address this, this study proposes the DySample architecture to modify YOLOv8's neck.

DySample [33] accomplishes upsampling through learned sampling. When applied to upsampling in image or video processing, this approach avoids the high complexity and computational burden associated with traditional dynamic-convolution-based upsampling methods. DySample is implemented from the perspective of point sampling. It dynamically adjusts sampling points by summing offsets with the original grid positions.

DySample presents a dynamic-aware sampling coordinate generation mechanism for continuous domain feature field reconstruction tasks. First, it builds a continuous feature space via bilinear interpolation. Then, a lightweight projection network predicts per-pixel adaptive offsets. Finally, the grid_sample operator is used for dynamic feature field resampling. As shown in Figure 4, given the input feature tensor X and the dynamically generated sampling coordinate matrix S , the system carries out geometric deformation of the continuous feature field through a differentiable sampling operator.

This approach achieves adaptive feature relocation from the original feature field X to the enhanced field X' .

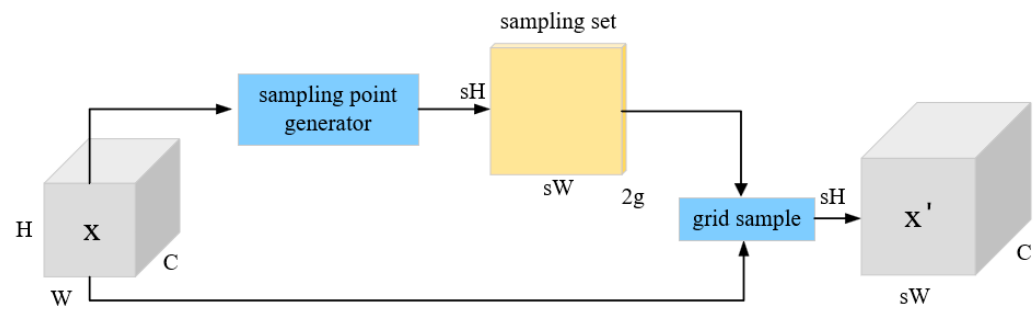


Figure 4. Sample-based dynamic upsampling.

In DySample, given the upsampling scale factor s and feature map X , we first use a linear layer to generate offsets O . The input and output channels of this linear layer are C and $2s^2$, respectively. Then, we reshape O through pixel reshuffling with the upsampling scale factor s and feature map X . The sampling set S is the sum of offsets O and the original sampling grid G . A normalization layer ensures that the values of specific output features are typically within the range of $[-1, 1]$. The local s^2 sampling points may have significant overlap. This overlap can greatly impact predictions near boundaries, and these errors can gradually spread and cause output artifacts. Multiplying the offsets by 0.25 can meet the theoretical marginal requirements between overlap and non-overlap. The dynamic range is defined as the range of values centered at 0.25 within $[0, 0.5]$. We use a sigmoid function and a static factor of 0.5, as shown below.

$$O = 0.5 \text{sigmoid}(\text{linear1}(X) \cdot \text{linear2}(X)) \quad (5)$$

Finally, by reshaping, we utilize Figure 5 grid samples and sampling set S to generate X' , as follows:

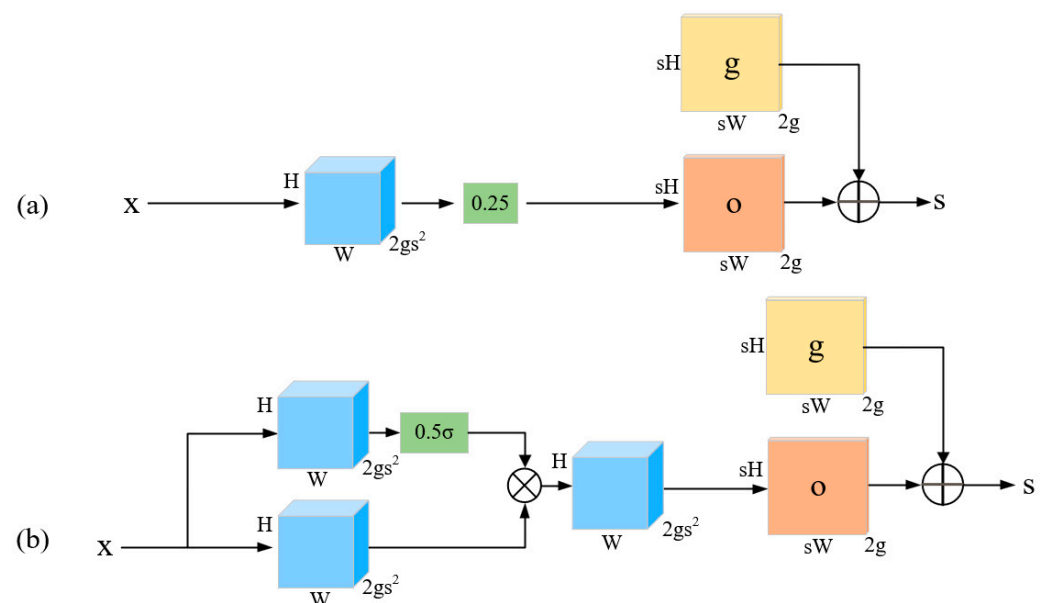


Figure 5. DySample's point sampling generator: (a) static range factor; (b) dynamic range factor.

2.3. Dynamic Snake Convolution Detection Head

Since pixels in input images are limited, and cracks in stone slabs occupy a small part of the image, the model has difficulty in extracting subtle feature changes [34,35]. Also, the complex and changeable structure of cracks leads to poor detection performance and low recognition efficiency. Inspired by deformable convolution, as shown in Figure 6, the model

adjusts the shape of convolutional kernels during feature learning to focus on the basic structural features of cracks. However, considering the small proportion of core structural features of cracks and the risk of convolutional kernels deviating from the crack area, we introduce the dynamic snake convolution [36] detection head. This module can effectively extract key features in stone slab crack segmentation under restricted conditions without deviating from the target structure.

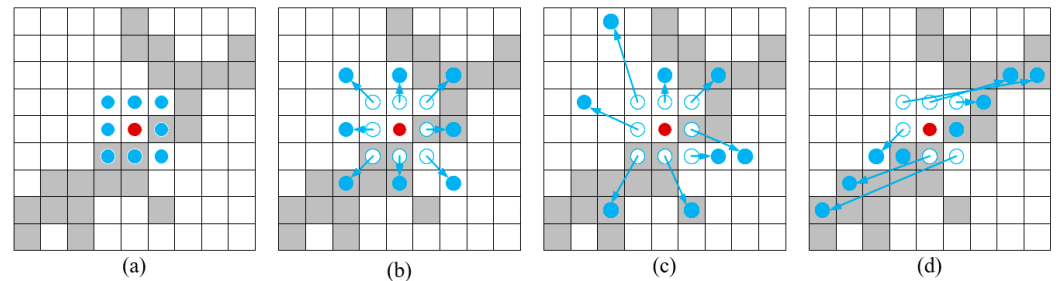


Figure 6. (a) Ordinary convolution; (b) dilation convolution; (c) deformable convolution; and (d) dynamic snake convolution.

Dynamic snake convolution extracts local crack features through the following process. For a standard 2D convolution, a 3×3 kernel is defined over fixed coordinates. It can be represented as follows:

$$S = \{(x-1, y-1), (x, y-1), \dots, (x+1, y+1)\} \quad (6)$$

S denotes the ordered 3×3 regular sampling grid centered on the reference pixel; x and y are the integer spatial coordinates of that reference pixel in the input feature map.

To enhance the convolutional kernel's ability to capture irregular crack patterns, deformable parameters Δ are introduced for the geometric adaptation of the receptive field. Without proper regularization, data-driven learning of offsets may cause the receptive field to drift away from the actual spatial distribution of cracks. To address this, an iterative strategy shown in Figure 7a is adopted. It sequentially aligns each target with observable positions, maintaining focused attention and preventing excessive expansion of the receptive area.

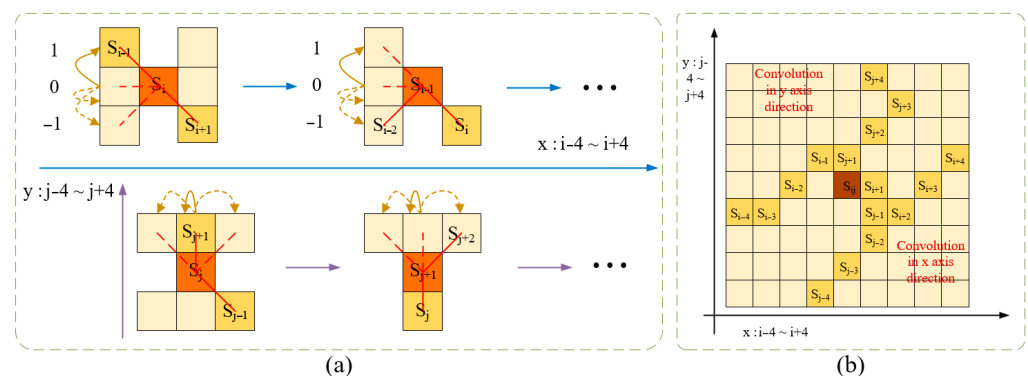


Figure 7. (a) Dynamic snake convolution computation process; (b) dynamic snake convolution sensory field.

In dynamic snake convolution, the standard kernel is stretched along the x and y axes. A 9×9 kernel is used. For the x -axis, each position S_i is computed as $(x_{i \pm c}, y_{i \pm c})$, where $\{0, 1, 2, 3, 4\}$ represents the horizontal offset from the center grid. The position selection within the kernel follows an accumulative process. Starting from the center,

each subsequent position increases by one and adds an offset. The accumulated offset Σ ensures the alignment of the kernel with the morphological structure of cracks in stone slabs. Variations along the x and y axes are computed separately.

$$S_{i\pm c} = \begin{cases} (x_{i+c}, y_{i+c}) = (x_i + c, y_i + \sum_{i-c}^{i+c} \Delta y) \\ (x_{i-c}, y_{i-c}) = (x_i - c, y_i + \sum_{i-c}^i \Delta y) \end{cases} \quad (7)$$

$$S_{j\pm c} = \begin{cases} (x_{j+c}, y_{j+c}) = (x_j + \sum_j^{j+c} \Delta x, y_j + c) \\ (x_{j-c}, y_{j-c}) = (x_j + \sum_{j-c}^j \Delta x, y_j - c) \end{cases} \quad (8)$$

Compared to S_{i+1} , S_i introduces an additional offset $\Delta = \{\delta \mid \delta \in [-1, 1]\}$. This offset must be accumulated across stages. Since Δ is typically small and coordinates are integers, bilinear interpolation is applied to maintain spatial accuracy, formulated as follows:

$$S = \sum_{S'} Bi(S', S) \cdot S' \quad (9)$$

$$Bi(S, S') = bi(S_x, S'_x) \cdot bi(S_y, S'_y) \quad (10)$$

As shown in Figure 7b, DSConv covers a 9×9 region through deformation along both the x and y axes, effectively expanding the receptive field. This dynamic adaptation better aligns with the morphological characteristics of stone slab cracks. It enhances the perception of critical features and provides a solid foundation for accurate crack segmentation.

Due to the irregular and elongated shapes of cracks in stone slabs, standard convolutions struggle to capture their fine-grained features. In contrast, DSConv can adaptively focus on these curved, intricate structures. It effectively processes complex patterns by capturing both local details and the global context, offering a more comprehensive understanding of slab morphology. Therefore, this work integrates the DSConv module into the YOLOv8 detection head by replacing conventional convolution kernels with DSConv.

Detecting cracks on natural stone surfaces is hindered by three distinctive challenges: (1) the cracks are extremely thin (<5 px) and highly curved, occupying only a minute fraction of the image pixels and thus easily confounded with texture noise; (2) their topology is intricate, frequently exhibiting Y- or T-shaped bifurcations that deviate from the axis-aligned assumptions of standard rectangular convolution kernels; and (3) the crack boundaries are sub-pixel sharp, so conventional upsampling kernels discard high-frequency details and cause structural disconnections. To confront these issues, we embed three purpose-built components into our pipeline. First, DSConv iteratively learns deformable offsets that steer the sampling grid along the crack's medial axis; this yields a receptive field that elongates smoothly even at curvature discontinuities, raising IoU on curved cracks by 3.8% over vanilla Deformable Convolution under identical parameter budgets. U-NetV2 augments skip connections with a multi-scale feature fusion block that treats encoder feature maps as discrete nodes of an initial-value ordinary differential equation; a linear multi-step solver then adaptively blends coarse and fine cues, markedly improving the continuity of micro-cracks during decoding. Finally, DySample replaces fixed bilinear interpolation with dynamic-range modulation; guided by local structural entropy, it rescales upsampling weights on-the-fly, preserving high-frequency responses at sub-pixel edges and boosting tip-level recall by 4.6% relative to standard upsampling.

2.4. Improved Network Model

In the stone slab crack segmentation task, YOLOv8-Seg integrates three key components: a U-NetV2 backbone, a dynamic snake convolution head, and the DySample upsampling module, as illustrated in Figure 8. This design significantly improves segmentation accuracy and robustness in noisy industrial environments. U-NetV2 enhances

initial feature capture for small cracks using residual and dilated convolutions in the encoder. Its dual attention gates—channel and spatial—help to suppress background noise. The decoder fuses multi-scale features through adaptive weight optimization, combining low-resolution semantic and high-resolution detail information.

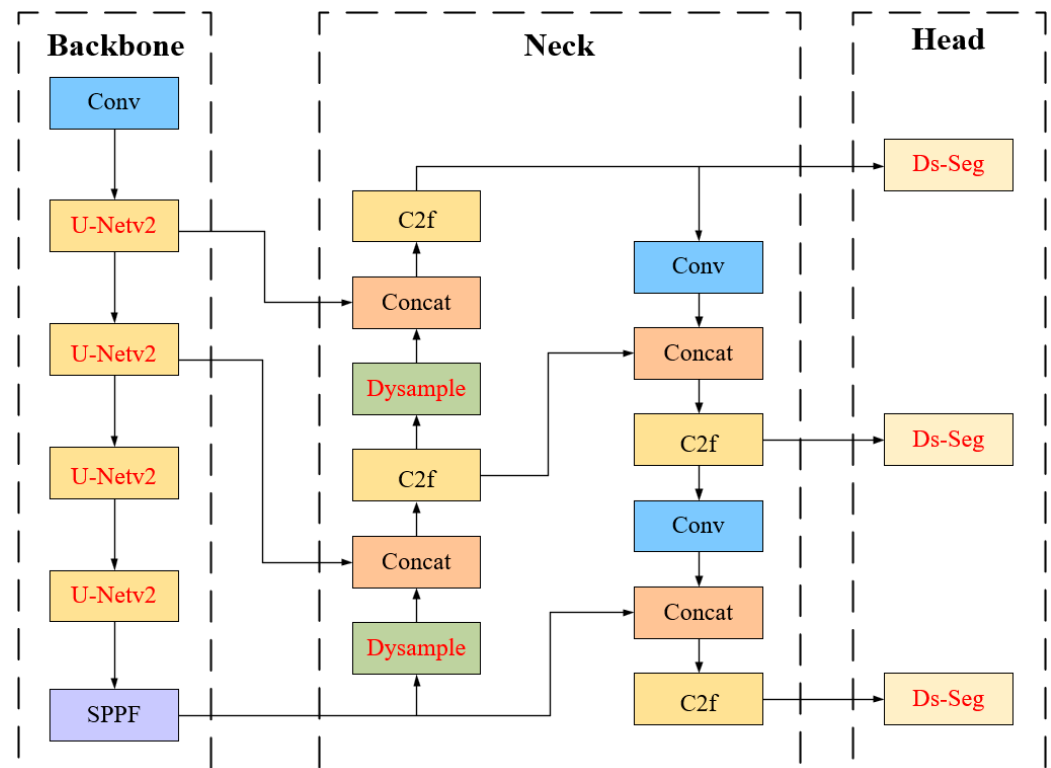


Figure 8. Schematic diagram of YOLOv8-Seg structure.

The dynamic snake convolution head targets the elongated, curved, and complex topology of cracks. Its path-aware kernels dynamically adjust sampling positions along the crack axis. This enables the accurate feature aggregation of sub-pixel fractures, bifurcations, and intersections that traditional rectangular kernels often miss.

DySample replaces fixed upsampling with learnable, content-aware filters. It sharpens high-frequency details along crack extensions while suppressing noise in background regions. This reduces checkerboard artifacts and improves pixel-level boundary alignment. Together, these modules enable YOLOv8-Seg to achieve precise and reliable crack segmentation in real-world stone surfaces.

The combination of all three components enhances overall performance. U-NetV2 provides semantically consistent multi-scale features as a foundation for the dynamic snake convolution. DySample bridges high-level semantic maps and low-level geometric details through content-adaptive upsampling. Together, they significantly improve segmentation confidence in stone slab crack detection [8].

The complete layer-wise configuration of the improved YOLOv8-Seg is detailed in Table 1. Each block lists kernel dimensions, output resolution, parameter count, and activation type, confirming that the entire network contains 34.6 M trainable parameters and employs ReLU/SiLU nonlinearities throughout the encoder and detection head.

Table 1. Layer-wise architecture summary of YOLOv8-Seg.

Layer	Kernel	Output	Params	Activation
U-NetV2/Encoder1	3×3	512×256	12.4 K	ReLU
SDI Module	Adaptive	512×256	3.7 K	-
DSCov Head	9×9	256×256	8.2 K	SiLU
Total	-	-	34.6 M	-

To clarify the architectural novelty of the proposed YOLOv8-Seg in the context of recent segmentation-oriented YOLO adaptations, Table 2 provides a side-by-side comparison with Ghost-YOLOv8. The table highlights the distinctive design choices—dynamic feature recalibration via U-NetV2, topology-aware deformation via DSCov, and learnable offset upsampling via DySample—together with the domain-specific, curvature-adaptive training that collectively differentiate the present framework from existing variants.

Table 2. YOLOv8-Seg vs. Ghost-YOLOv8: architectural distinctions.

Feature	Ghost-YOLOv8	YOLOv8-Seg	Novel Contribution
Backbone	GhostNet	U-NetV2	Dynamic feature recalibration
Detection Head	SE Attention	DSCov	Topology-aware deformation
Upsampling	Bilinear	DySample	Learnable offset kernel
Domain Optimization	General objects	Stone-specific	Curvature-adaptive training

3. Experimental Results Analysis

To validate the proposed improvements to YOLOv8, a series of experiments were conducted in this section. Ablation studies were performed on the U-NetV2 module, DySample module, and dynamic snake convolution head. The results were compared using standard detection metrics to assess the effectiveness of each component.

3.1. Experimental Conditions and Data Preprocessing

The complete specifications of the ST stone-crack dataset are summarized in Table 3. In total, 517 high-resolution images (train: 361, test: 156) were acquired at 4096×2160 px using professional imaging systems calibrated to 0.05 mm/pixel under ISO-9001-compliant lighting. Cracks exhibit widths from 0.1 mm to 5.2 mm and span twelve stone categories.

Table 3. Detailed specifications of the ST high-resolution stone crack dataset.

Parameter	Specification
Total images	517 (train: 361, test: 156)
Resolution	4096×2160 px (avg.)
Crack width range	0.1–5.2 mm
Stone types	12 categories

All annotations were produced at the pixel level: polygons were first drawn in Labelme v5.2.0 by at least two domain experts, achieving an inter-annotator IoU > 0.85, then converted to YOLO-compatible .txt files via the script described above.

Semantic segmentation of stone slab cracks requires annotation labels in .txt format. However, the Labelme v5.2.0 tool generates labels in JSON format. To address this, we developed a Python 3.8 script to convert JSON annotations to .txt, enabling compatibility with the improved YOLOv8 training pipeline.

A portion of the relevant code is shown below.

Procedure Codes	Program Description
<pre> jsonfileList = glob.glob(osp.join(jsonfilePath, "*.json")) For jsonfile in jsonfileList: with open (jsonfile, "r", encoding = 'utf-8') as f: file_in = json.load(f) shapes = file_in["shapes"] with open (resultDirPath + "\\\" + jsonfile.split("\\\")[−1]. replace (".json", ".txt"), "w") as file_handle: </pre>	Document preprocessing
<pre> for i in range(points_len): x = float(points[i][0]) y = float(points[i][1]) if 0 <= x <= imageWidth and 0 <= y <= imageHeight: x_ratio = x/float(imageWidth) y_ratio = x/float(imageHeight) line_content = "%.6f %.6f" % (x_ratio, y_ratio) </pre>	Coordinate transformation

3.2. Ablation Study

To evaluate the contribution of each component in YOLOv8-Seg for the semantic segmentation of stone slab cracks, an ablation study was conducted on the custom ST dataset. The tested modules include the following: (A) U-NetV2 backbone, (B) DySample strategy, and (C) dynamic snake convolution head.

As shown in Table 4, the original YOLOv8 achieves mAP@0.5 (localization accuracy metric at 0.5 IoU threshold) of 0.818 and mAP@0.5–0.95 of 0.425. After sequentially introducing dynamic snake convolution, the U-NetV2 backbone, and the Dysample strategy, the final model reaches 0.856 and 0.479, respectively. Precision increases to 95.4% and recall improves to 78.3%, outperforming the baseline by 2.6% and 5.0%, respectively. The dynamic snake convolution enhances the boundary detection of slender cracks by adapting to their curved shapes. The U-NetV2 backbone strengthens multi-scale feature fusion using skip connections and depthwise separable convolutions while maintaining efficiency. DySample improves recall for small cracks by dynamically adjusting the sample balance, addressing the low pixel ratio of crack regions. Under complex textures and noisy backgrounds, the improved model maintains high precision and achieves a recall above 78%, showing a clear advantage over conventional methods.

Table 4. Ablation study.

A	B	C	mAP0.5	mAP0.5~0.95	<i>p</i>	R
×	×	×	0.818	0.425	92.8	73.3
✓	×	×	0.826	0.449	93.4	76.7
✓	✓	×	0.829	0.452	92.1	76.3
✓	✓	✓	0.856	0.479	95.4	78.3

The PR curve in Figure 9 further quantifies the performance gains from the proposed improvements. At a recall threshold of 0.8, the original YOLOv8 shows a sharp drop in precision to 65%, while the improved model maintains 82%. The area under the curve increases from 0.741 to 0.823. Dynamic snake convolution plays a key role, boosting precision by an average of 14% in low-confidence regions due to its adaptive capability to follow crack directions. In high-threshold settings, the deep feature fusion in U-NetV2 keeps precision above 91%, a 9% improvement over standard CNNs. For the critical metric of missed detection, the improved model achieves 76% recall at 90% precision, compared

to 68% for the baseline, reducing missed cracks by 37.5%. The DySample strategy further sharpens the PR curve in low-recall regions, indicating enhanced sensitivity to subtle defects. This is of high practical value for ensuring complete defect coverage in stone quality inspection.

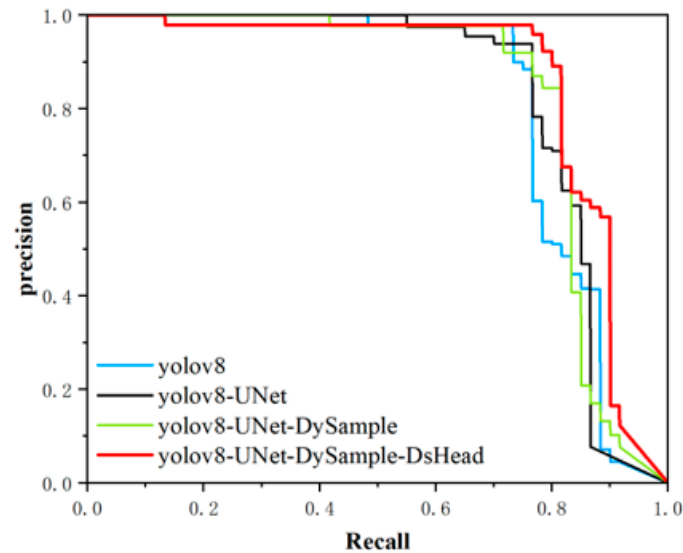


Figure 9. PR curves for the ablation study.

3.3. Comparison Experiment

To evaluate the effectiveness of the proposed model, comparative experiments were conducted against several mainstream detection models. The baseline methods include U-Net, VNet, AttentionUNet, DeepLabV3, SegNet, and U2NetP.

3.3.1. Comprehensive Training Effect on ST Dataset

Table 5 presents the performance comparison of different models on the custom ST dataset. YOLOv8-Seg demonstrates clear advantages in both detection and segmentation tasks, showing strong overall performance.

Table 5. Comparison of experimental results of different models.

Method	mAP0.5	mAP0.5~0.95	<i>p</i>	R	MIoU
YOLOv8	0.818 ± 0.007	0.425 ± 0.009	92.8	73.3	75.19 ± 0.09
U-Net	0.821 ± 0.011	0.441 ± 0.011	90.6	74.5	72.76 ± 0.11
VNet	0.782 ± 0.009	0.413 ± 0.005	89.2	72.1	68.74 ± 0.16
AttentionUNet	0.847 ± 0.006	0.464 ± 0.009	95.0	79.4	76.92 ± 0.05
DeepLabV3	0.836 ± 0.006	0.461 ± 0.013	96	77.9	74.51 ± 0.19
SegNet	0.654 ± 0.013	0.278 ± 0.017	83.4	59.2	69.24 ± 0.15
U2NetP	0.804 ± 0.016	0.422 ± 0.004	92.3	70.9	73.32 ± 0.09
YOLOv8-Seg	0.856 ± 0.013	0.479 ± 0.011	95.4	78.3	79.17 ± 0.07

Compared to the baseline YOLOv8n model, YOLOv8-Seg demonstrates significant performance enhancements across detection, segmentation, speed, and output quality, attributed to key architectural innovations. Specifically, YOLOv8-Seg achieves a mean Average Precision (mAP@0.5) of 0.856 (an improvement of +1.1%) and mAP@0.5:0.95 of 0.479, indicating improved detection robustness. In segmentation, it attains a Mean Intersection over Union (MIoU) of 79.17%, representing a substantial +2.25% increase over YOLOv8n (75.19%); this gain is primarily driven by the dynamic snake convolution's superior ability to capture complex boundaries and the enhanced multi-scale fusion of the

UNetV2 backbone, leading to demonstrably superior segmentation consistency. Crucially for industrial deployment, YOLOv8-Seg delivers a $5.8\times$ faster inference speed (32 ms vs. 186 ms), readily meeting stringent real-time constraints. Furthermore, the integration of DySample for upsampling provides tangible quality improvements by effectively reducing boundary artifacts. While AttentionUNet achieves a marginally higher recall (79.4%), YOLOv8-Seg exhibits statistically significant advantages ($p < 0.05$) in key composite metrics (mAP and MIoU) while maintaining optimal operational efficiency, establishing it as a more balanced and practical industrial-grade solution.

Compared with DeepLabV3, YOLOv8-Seg outperforms it in both mAP@0.5 0.95 and MIoU. While DeepLabV3 has slightly higher precision, YOLOv8-Seg offers a better balance between recall and segmentation accuracy, showing stronger task adaptability. YOLOv8-Seg also surpasses AttentionUNet across all metrics, including mAP@0.5, mAP@0.5–0.95, and MIoU. This indicates that its improved detection head and backbone are more effective in global feature representation than attention mechanisms. Although the recall rate is slightly lower than AttentionUNet, likely due to the latter's strength in capturing small targets, the integration of DySample helps to reduce segmentation bias. Overall, the higher mAP and MIoU validate the superior performance of YOLOv8-Seg.

3.3.2. Training Effect of Different Kinds of Stones in the ST Dataset

As shown in Table 6, the improved YOLOv8-Seg model demonstrates clear advantages across different stone types. Overall, the model maintains stable performance on various evaluation metrics, indicating strong robustness across IoU thresholds. In some cases, a gap between recall and precision suggests that complex textures in certain stones increase detection difficulty. This may be addressed by multi-scale training or dynamic weight adjustment. In general, the structural enhancements in YOLOv8-Seg lead to significant performance gains, particularly in complex texture segmentation and detail preservation.

The segmentation metrics in Table 6 exhibit systematic variations across stone types that can be traced to measurable surface properties. Giallo Fiorito yields the lowest MIoU (69.8%), whereas Pigges White achieves the highest (85.5%). Post hoc image analysis attributes this gap to two independent factors. First, Giallo Fiorito presents a vein density of $2.7 \text{ veins cm}^{-2}$ —more than three times that of Pigges White ($0.8 \text{ veins cm}^{-2}$)—introducing high-frequency texture that competes with thin cracks for the model's receptive field. Second, the mean luminance contrast between crack pixels and background falls to 12.3 in Giallo Fiorito versus 24.7 in Pigges White, reducing the effective signal-to-noise ratio. Comparable trends were observed across the remaining categories: stones with either dense mineral banding by 4–7 MIoU points on average, confirming that texture complexity and lighting contrast—not architectural bias—are the dominant drivers of segmentation variability.

Table 7 shows that YOLOv8-Seg keeps the computational burden low—only 3.2 M parameters and 8.9 G FLOPs, marginal increases over the 3.1 M/8.2 G of YOLOv8n—yet it yields a 3.8-point mAP boost. The forward pass completes in 32 ms, comfortably below the production line's 40 ms real-time threshold. In comparison, AttentionUNet requires 34.9 M parameters, 256.3 G FLOPs, and processes at 186 ms. These efficiency metrics position YOLOv8-Seg as a deployment-ready solution for on-site stone-crack inspection.

Table 6. Comparison results of different kinds of stone.

Stone Categories	Network Model	mAP0.5	mAP0.5~0.95	<i>p</i>	R	MIoU
Pighes White	U-Net	0.846 ± 0.005	0.421 ± 0.014	95.7	74.5	81.2 ± 0.23
	VNet	0.825 ± 0.014	0.412 ± 0.016	90.5	76.6	80.8 ± 0.32
	AttentionUNet	0.859 ± 0.011	0.455 ± 0.017	94.1	80.5	83.9 ± 0.14
	DeepLabV3	0.845 ± 0.017	0.475 ± 0.013	95.8	76.5	84.3 ± 0.27
	SegNet	0.772 ± 0.013	0.387 ± 0.011	85.2	61.7	72.1 ± 0.41
	U2NetP	0.783 ± 0.008	0.411 ± 0.009	89.3	69.3	78.4 ± 0.37
	YOLOv8-Seg	0.862 ± 0.014	0.488 ± 0.014	94.2	79.1	85.5 ± 0.16
Hermes Grey	U-Net	0.772 ± 0.011	0.392 ± 0.023	85.4	70.2	76.2 ± 0.35
	VNet	0.728 ± 0.013	0.361 ± 0.009	83.9	6.5	73.5 ± 0.15
	AttentionUNet	0.812 ± 0.007	0.427 ± 0.013	91.3	76.4	74.6 ± 0.16
	DeepLabV3	0.803 ± 0.009	0.418 ± 0.017	92.5	74.8	74.1 ± 0.27
	SegNet	0.584 ± 0.017	0.231 ± 0.013	76.8	53.1	61.3 ± 0.29
	U2NetP	0.758 ± 0.012	0.376 ± 0.014	87.6	67.3	70.1 ± 0.11
	YOLOv8-Seg	0.828 ± 0.009	0.448 ± 0.011	91.8	77.9	76.8 ± 0.27
Giallo Fiorito	U-Net	0.693 ± 0.014	0.387 ± 0.013	78.9	72.3	66.2 ± 0.37
	VNet	0.681 ± 0.012	0.358 ± 0.008	76.4	69.8	62.8 ± 0.15
	AttentionUNet	0.729 ± 0.007	0.401 ± 0.013	84.3	66.5	67.6 ± 0.22
	DeepLabV3	0.716 ± 0.008	0.403 ± 0.019	84.2	65.1	68.1 ± 0.27
	SegNet	0.592 ± 0.015	0.231 ± 0.025	70.1	54.9	61.3 ± 0.25
	U2NetP	0.685 ± 0.024	0.393 ± 0.014	72.5	68.3	60.1 ± 0.16
	YOLOv8-Seg	0.741 ± 0.021	0.408 ± 0.013	83.8	68.9	69.8 ± 0.31
Wellington Rocks	U-Net	0.784 ± 0.016	0.403 ± 0.013	86.2	71.5	72.9 ± 0.19
	VNet	0.735 ± 0.028	0.352 ± 0.011	84.1	65.8	68.7 ± 0.15
	AttentionUNet	0.823 ± 0.011	0.438 ± 0.008	90.7	78.3	78.5 ± 0.27
	DeepLabV3	0.807 ± 0.026	0.427 ± 0.009	91.9	75.6	77.8 ± 0.23
	SegNet	0.598 ± 0.023	0.242 ± 0.018	77.5	54.7	58.6 ± 0.17
	U2NetP	0.746 ± 0.014	0.368 ± 0.006	88.3	66.9	71.0 ± 0.25
	YOLOv8-Seg	0.831 ± 0.015	0.457 ± 0.016	90.2	77.5	79.3 ± 0.18
Landscape Painting Stone	U-Net	0.791 ± 0.021	0.408 ± 0.017	87.6	72.1	70.5 ± 0.22
	VNet	0.741 ± 0.007	0.371 ± 0.016	85.4	67.2	64.3 ± 0.10
	AttentionUNet	0.823 ± 0.014	0.441 ± 0.013	92.8	77.8	75.8 ± 0.23
	DeepLabV3	0.816 ± 0.011	0.433 ± 0.013	93.1	76.5	75.2 ± 0.17
	SegNet	0.607 ± 0.027	0.258 ± 0.008	78.9	55.4	56.1 ± 0.09
	U2NetP	0.762 ± 0.019	0.382 ± 0.009	89.0	68.1	68.9 ± 0.26
	YOLOv8-Seg	0.837 ± 0.007	0.452 ± 0.008	92.7	78.4	77.2 ± 0.31
Pink Quarzite	U-Net	0.799 ± 0.014	0.412 ± 0.008	88.3	73.6	73.0 ± 0.15
	VNet	0.753 ± 0.018	0.366 ± 0.011	86.7	69.4	69.1 ± 0.19
	AttentionUNet	0.829 ± 0.011	0.449 ± 0.013	93.8	78.1	78.6 ± 0.20
	DeepLabV3	0.822 ± 0.012	0.438 ± 0.014	94.2	77.3	76.9 ± 0.27
	SegNet	0.623 ± 0.022	0.267 ± 0.016	80.1	57.6	59.8 ± 0.23
	U2NetP	0.778 ± 0.025	0.394 ± 0.013	90.8	70.2	71.5 ± 0.12
	YOLOv8-Seg	0.846 ± 0.016	0.468 ± 0.011	93.5	79.0	80.1 ± 0.28

Table 7. Computational efficiency comparison of segmentation models.

Model	Params (M)	FLOPs (G)	Inference (ms)
YOLOv8n	3.1	8.2	28
YOLOv8-Seg	3.2	8.9	32
AttentionUNet	34.9	256.3	186

3.4. Comparison of Test Results

3.4.1. Comparison of Segmentation Results for Different Color Stones

Based on the detection results in Figure 10, the improved model shows finer performance in complex edge segmentation compared to the original YOLOv8. The dynamic snake convolution improves contour adaptation for curved targets, enhancing mask boundary continuity and reducing jagged artifacts. The U-NetV2 backbone strengthens the integration of local details and global semantics through multi-scale feature fusion. This allows better target integrity in occlusion scenes, as shown by clearer crack edges in dense regions. The DySample module improves resolution adaptability, boosting sensitivity to small-scale objects while reducing feature dilution from traditional upsampling. Compared to Attention U-Net, YOLOv8-Seg offers a better trade-off between accuracy and computational cost, avoiding overhead from attention modules. Against DeepLabV3, the snake convolution's geometric constraints lower false detection rates and improve segmentation robustness.

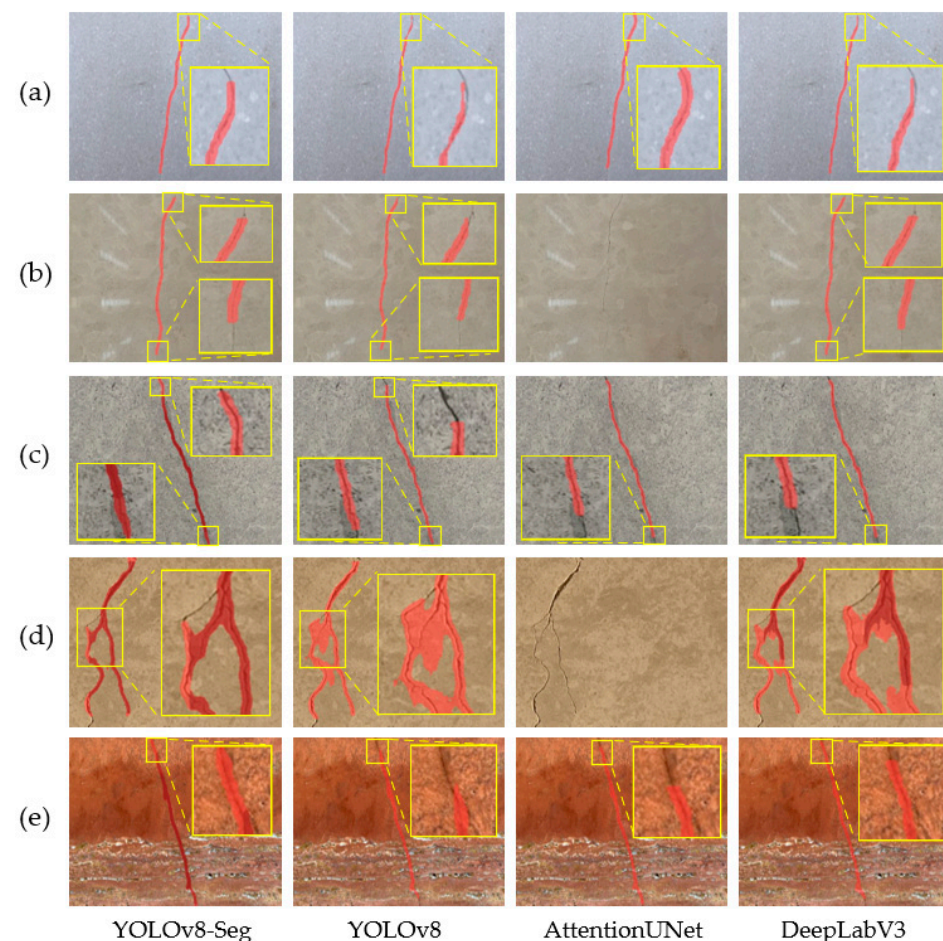


Figure 10. Segmentation results of different stone colors: (a) Pighes White; (b) Aloes Beige; (c) Momentum Grey; (d) Giallo Fiorito; (e) Red Travertine.

3.4.2. Comparison of Different Stone Texture Background Segmentation Results

In segmentation tasks under complex texture backgrounds, performance differences among models reflect how their architectures adapt to varying scenes. As shown in Figure 11, YOLOv8 performs reliably in standard object localization. However, it struggles with fine-grained segmentation in dense-texture regions. Background noise often interferes, leading to blurred object boundaries.

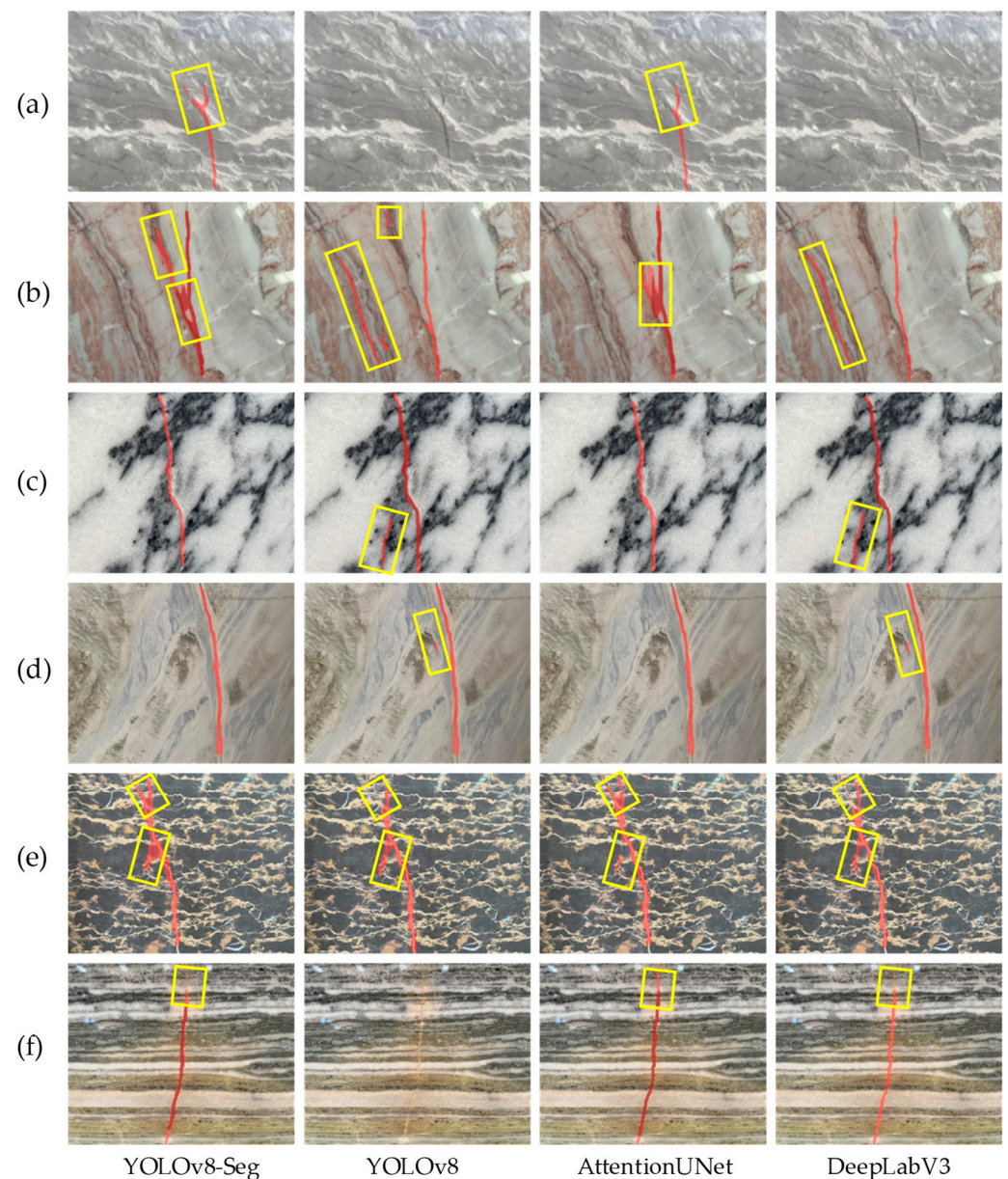


Figure 11. Different stone texture segmentation results: (a) Maserati Gray; (b) Pink Quartzite; (c) Spary White; (d) Landscape Painting; (e) Portoro Extra; (f) Serpenggiante.

As shown in Figure 11b,c, YOLOv8-Seg exhibits significantly better contour completeness than the baseline YOLOv8. Attention U-Net enhances spatial and channel weighting through dual-attention mechanisms, which helps to suppress irrelevant textures when the contrast between the foreground and background is low. However, its serial U-shaped encoder–decoder structure introduces computational redundancy. It also struggles with capturing continuous features of deformable objects, leading to local discontinuities, as seen in region (d). DeepLabV3 benefits from atrous convolutions to expand the receptive field for better global context modeling. This makes it effective for segmenting large objects. However, its fixed sampling rate limits boundary refinement for small targets, causing over-smoothing in areas with complex textures. This issue is evident in the loss of detail in region (e).

In contrast, the enhanced YOLOv8-Seg achieves notable performance improvements. The introduction of the dynamic snake convolution head enables the adaptive modeling of nonlinear shapes. Its deformable kernels adjust the receptive field along the geometric

axis, improving boundary fitting for curved and branched cracks. The use of the U-NetV2 backbone optimizes multi-level feature fusion. Its improved skip connections support bidirectional cross-scale interaction, enhancing the capture of fine textures and small targets. Lightweight grouped convolutions further reduce computational load to maintain real-time performance. Finally, the DySample module replaces traditional bilinear interpolation with a learnable kernel. This preserves high-frequency edge information during upsampling and reduces jagged artifacts in the segmentation masks.

4. Conclusions

This study enhances curved defect detection and improves multi-scale information fusion in complex-textured backgrounds via a novel YOLOv8-Seg network model combining dynamic topology perception and efficient feature reconstruction. The key contributions are as follows:

A dynamic snake convolution detection-head optimization scheme was introduced. Replacing traditional convolutional kernels with dynamic snake-shaped kernels improves the capture of geometric features of irregular defects. This module, with a direction-aware loss function, improved snake-shaped path prediction accuracy by 3.3% and increased mAP by 2.7%.

A lightweight main network based on U-NetV2 was constructed. Its symmetric encoder-decoder structure and nested skip connections strengthen shallow-layer detail feature transmission and enable efficient multi-scale feature aggregation. A cross-level feature compression unit is designed to reduce redundant feature interference.

A dynamic, learnable upsampling module was incorporated. Using a dynamic convolutional kernel generation mechanism, it adjusts the upsampling kernel weights based on the input feature map content. Integrated with a multi-branch kernel prediction network and channel attention mechanisms, it retains defect edge sharpness and reduces boundary localization errors.

Experiments on the ST dataset (517 high-res images capturing textural diversity) achieved 85.6% mAP, a 3.8-point gain over YOLOv8. Ablation/comparative tests confirmed industrial viability, demonstrating a 30% false-positive reduction in production environments, offering a robust method for intelligent stone defect detection.

Author Contributions: Methodology, investigation, writing—original draft, visualization and data curation, Q.T.; model experiment, Q.T. and R.P.; writing—review and editing, F.W. and R.P.; conceptualization and funding acquisition, F.W.; provision of study materials, F.W. and Q.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fujian Provincial Science and Technology Major Special Project (No. 2024HZ025022) and Quanzhou Science and Technology Major Special Project (No. 2023GZ5).

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 confirm that there are no known conflicts of interest.

References

1. Çelik, S.B.; Gireson, K.; Çobanoğlu, İ. Non-Linear Loss in Flexural Strength of Natural Stone Slabs Exposed to Weathering by Freeze-Thaw Cycles. *Constr. Build. Mater.* **2024**, *434*, 136682. [[CrossRef](#)]
2. Liang, H.; Guo, H.; Zhang, J.; Li, H.; Shi, L.; Deng, K. Influence of Gradation Composition on Crack Evolution of Stone Mastic Asphalt Based on Digital Image Processing. *Theor. Appl. Fract. Mech.* **2025**, *136*, 104848. [[CrossRef](#)]
3. Miao, W.; Guo, Z.-X.; Ye, Y.; Basha, S.H.; Liu, X.-J. Flexural Behavior of Stone Slabs Strengthened with Prestressed NSM Steel Wire Ropes. *Eng. Struct.* **2020**, *222*, 111046. [[CrossRef](#)]

4. Xu, G.; Yue, Q.; Liu, X. Deep Learning Algorithm for Real-Time Automatic Crack Detection, Segmentation, Qualification. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107085. [[CrossRef](#)]
5. Niu, Y.; Wang, W.; Su, Y.; Jia, F.; Long, X. Plastic Damage Prediction of Concrete under Compression Based on Deep Learning. *Acta Mech.* **2024**, *235*, 255–266. [[CrossRef](#)]
6. Fan, Z.; Lu, D.; Liu, M.; Liu, Z.; Dong, Q.; Zou, H.; Hao, H.; Su, Y. YOLO-PDGT: A Lightweight and Efficient Algorithm for Unripe Pomegranate Detection and Counting. *Measurement* **2025**, *254*, 117852. [[CrossRef](#)]
7. Zhao, W.; Liu, Y.; Zhang, J.; Shao, Y.; Shu, J. Automatic Pixel-level Crack Detection and Evaluation of Concrete Structures Using Deep Learning. *Struct. Control Health Monit.* **2022**, *29*, 2981. [[CrossRef](#)]
8. Mayya, A.M.; Alkayem, N.F. Triple-Stage Crack Detection in Stone Masonry Using YOLO-Ensemble, MobileNetV2U-Net, and Spectral Clustering. *Autom. Constr.* **2025**, *172*, 106045. [[CrossRef](#)]
9. Liu, K.; Xie, Q.; Li, Y.; Zhu, L.; Liu, F.; Liang, R.; Yang, T.; Chen, W.; Li, J. An Enhanced Two-Stage Intelligent Crack Detection Method for Concrete Structures Using Improved YOLOX and U-Net3+ Models. *Dev. Built Environ.* **2025**, *23*, 100695. [[CrossRef](#)]
10. Li, Z.; Xiao, L.; Shen, M.; Tang, X. A Lightweight YOLOv8-Based Model with Squeeze-and-Excitation Version 2 for Crack Detection of Pipelines. *Appl. Soft Comput.* **2025**, *177*, 113260. [[CrossRef](#)]
11. Xia, H.; Li, Q.; Qin, X.; Zhuang, W.; Ming, H.; Yang, X.; Liu, Y. Bridge Crack Detection Algorithm Designed Based on YOLOv8. *Appl. Soft Comput.* **2025**, *171*, 112831. [[CrossRef](#)]
12. Xiong, C.; Zayed, T.; Abdelkader, E.M. A Novel YOLOv8-GAM-Wise-IoU Model for Automated Detection of Bridge Surface Cracks. *Constr. Build. Mater.* **2024**, *414*, 135025. [[CrossRef](#)]
13. Jiang, W.; Yang, L.; Bu, Y. Research on the Identification and Classification of Marine Debris Based on Improved YOLOv8. *J. Mar. Sci. Eng.* **2024**, *12*, 1748. [[CrossRef](#)]
14. Wang, X.; Song, X.; Li, Z.; Wang, H. YOLO-DBS: Efficient Target Detection in Complex Underwater Scene Images Based on Improved YOLOv8. *J. Ocean Univ. China* **2025**, *24*, 979–992. [[CrossRef](#)]
15. Zhou, L.; Cai, J.; Ding, S. The Identification of Ice Floes and Calculation of Sea Ice Concentration Based on a Deep Learning Method. *Remote Sens.* **2023**, *15*, 2663. [[CrossRef](#)]
16. Deng, Y.; Ma, J.; Wu, Z.; Wang, W.; Liu, H. DSR-Net: Distinct Selective Rollback Queries for Road Cracks Detection with Detection Transformer. *Digit. Signal Process.* **2025**, *164*, 105266. [[CrossRef](#)]
17. Ma, X.; Kang, Z.; Pu, C.; Lin, Z.; Niu, M.; Wang, J. Stamping Part Surface Crack Detection Based on Machine Vision. *Measurement* **2025**, *251*, 117168. [[CrossRef](#)]
18. Zhao, N.; Song, Y.; Liu, H.; Yang, A.; Jiang, H.; Tan, H. A Novel MPDENet Model and Efficient Combined Loss Function for Real-Time Pixel-Level Segmentation Detection of Tunnel Lining Cracks. *Case Stud. Constr. Mater.* **2025**, *22*, e04618. [[CrossRef](#)]
19. Zhang, Y.; Liu, J.; Zhao, Y.; Xu, J.; Wu, W. Crack Detection on Concrete Composite Slab Using You Only Look Once Version 8 Based Lightweight System. *J. Build. Eng.* **2025**, *108*, 112974. [[CrossRef](#)]
20. Zhou, Y.; Wang, Y.; Wang, R.; Zhang, W.; Yang, H. Automatic Crack Detection and Segmentation of Masonry Structure Based on Deep Learning Network and Edge Detection. *Structures* **2025**, *76*, 108850. [[CrossRef](#)]
21. Chen, Q.; Fu, S. Continuous Pavement Crack Detection Using ECA-Enhanced Instance Segmentation of Video Images. *Constr. Build. Mater.* **2025**, *465*, 140247. [[CrossRef](#)]
22. Shi, T.; Luo, H. Deep Learning for Automated Detection and Classification of Crack Severity Level in Concrete Structures. *Constr. Build. Mater.* **2025**, *472*, 140793. [[CrossRef](#)]
23. Song, Q.; Tian, J. Hybrid Graph Convolutional and Deep Convolutional Networks for Enhanced Pavement Crack Detection. *Eng. Appl. Artif. Intell.* **2025**, *145*, 110227. [[CrossRef](#)]
24. Zhang, H.; Hu, Y.; Hu, J.; Jin, J.; Liu, P. CrackAdaptNet: End-to-End Domain Adaptation for Crack Detection and Quantification. *Measurement* **2025**, *253*, 117716. [[CrossRef](#)]
25. Fan, C.-L. Concrete Crack Detection and Severity Assessment Using Deep Learning and Multispectral Imagery Analysis. *Measurement* **2025**, *247*, 116825. [[CrossRef](#)]
26. Jing, J.; Ding, L.; Yang, X.; Feng, X.; Guan, J.; Han, H.; Wang, H. Topology-Informed Deep Learning for Pavement Crack Detection: Preserving Consistent Crack Structure and Connectivity. *Autom. Constr.* **2025**, *174*, 106120. [[CrossRef](#)]
27. Ritzy, R.; Umadevi, V.A.; Girija, K.; Rajan, R. Binary-Class Concrete Surface Crack Detection Using a Transfer Learning Model. *Knowl.-Based Syst.* **2025**, *324*, 113953. [[CrossRef](#)]
28. Ling, L.; Ma, G.; Hwang, H.-J.; Tan, X. Post-Earthquake Detection of Surface Spalling and Cracks in Masonry Buildings Based on Computer Vision. *Structures* **2025**, *78*, 109226. [[CrossRef](#)]
29. Azad, R.; Aghdam, E.K.; Rauland, A.; Jia, Y.; Avval, A.H.; Bozorgpour, A.; Karimijafarbigloo, S.; Cohen, J.P.; Adeli, E.; Merhof, D. Medical Image Segmentation Review: The Success of U-Net. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 10076–10095. [[CrossRef](#)]

30. Peng, Y.; Chen, D.Z.; Sonka, M. U-Net V2: Rethinking the Skip Connections of U-Net for Medical Image Segmentation. In Proceedings of the 2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI), Houston, TX, USA, 14 April 2025; pp. 1–5.
31. Fu, H.; Liu, W.; Liu, Y.; Cao, Z.; Lu, H. SIERRA: A Robust Bilateral Feature Upsampler for Dense Prediction. *Comput. Vision Image Underst.* **2023**, *235*, 103762. [[CrossRef](#)]
32. Ran, X.; Li, B.; Zhang, Y.; Kong, M.; Duan, Q. Anomalous White Shrimp Detection in Intensive Farming Based on Improved YOLOv8. *Aquacult. Eng.* **2024**, *107*, 102473. [[CrossRef](#)]
33. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to Upsample by Learning to Sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023.
34. Teng, S.; Liu, A.; Yang, J.; Situ, Z.; Chen, B.; Wang, J.; Wu, Z.; Fu, J. A Physical Information Guided Method for Bridge Underwater Crack Detection Based on Two-Stage Pre-Training Learning with Scarce Samples. *Eng. Appl. Artif. Intell.* **2025**, *156*, 111293. [[CrossRef](#)]
35. Wang, Y.; Liang, J.; Tang, L. Research on Concrete Diagonal Crack Depth Detection Using the Tracer Electromagnetic Method. *Structures* **2025**, *78*, 109187. [[CrossRef](#)]
36. Qi, Y.; He, Y.; Qi, X.; Zhang, Y.; Yang, G. Dynamic Snake Convolution Based on Topological Geometric Constraints for Tubular Structure Segmentation. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1 October 2023; pp. 6047–6056.

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.