

Article

Real-Time Post-Earthquake Structural Crack Segmentation Using a Quadrupedal Robotic Inspection Platform

Kemal Hacıfendioglu ^{1,2,*} , Volkan Kahya ^{1,2} , Ali Motamedi ² and Ayşecan Bostan ² 

¹ Earthquake and Structural Health Monitoring Research Center, Karadeniz Technical University, Trabzon 61080, Türkiye; volkan@ktu.edu.tr

² Department of Civil Engineering, Karadeniz Technical University, Trabzon 61080, Türkiye; ali.motamedi00@yahoo.com (A.M.); aysecanbostan61@gmail.com (A.B.)

* Correspondence: kemalhacifendioglu@ktu.edu.tr

Abstract

Post-earthquake structural inspections are critical for public safety and recovery, yet traditional manual assessments are slow, hazardous, and resource-intensive. This paper proposes a novel system that integrates a quadrupedal robot with a deep learning (DL) vision model to rapidly detect structural cracks and damage in the aftermath of earthquakes. A Unitree Go2 quadruped robot, equipped with cameras and sensors, is paired with a YOLOv8 instance segmentation network for near-real-time crack detection and localization. The approach addresses key limitations of manual post-disaster inspections by enabling operator-supervised, near-real-time visual crack screening in hazardous or hard-to-reach areas. The YOLOv8 model is trained on a curated dataset of crack and damage images to support crack detection and segmentation performance, and its advanced segmentation capabilities allow precise delineation of damaged regions. The integrated system is validated on a laboratory-scale concrete–steel frame with simulated damage. Preliminary results demonstrate that the Unitree Go2 quadruped robot can navigate and inspect structural elements while the AI model identifies and segments cracks and surface damage under near-real-time laboratory operating conditions. This work highlights the potential of combining advanced legged robotics and state-of-the-art DL for structural health monitoring (SHM), offering a preliminary visual screening tool that can support operator awareness and help prioritize areas requiring expert structural inspection.

Keywords: crack segmentation; YOLOv8; robotic inspection; structural health monitoring; instance segmentation; deep learning; damage detection



Academic Editor: Francesco Zirilli

Received: 8 April 2026

Revised: 2 May 2026

Accepted: 4 May 2026

Published: 10 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Earthquakes can inflict severe damage on built infrastructure, necessitating prompt structural assessments to guide emergency response and repairs [1]. However, post-earthquake damage evaluation is traditionally performed via visual inspections by engineers, a process that is laborious, time-consuming, and often dangerous due to unstable structures [2,3]. Manual inspections also require specialized expertise and are prone to human error, with limited frequency and reach under disaster conditions [4,5]. These constraints motivate the development of automated approaches [6]. Recent advances in robotics and computer vision offer promising alternatives, as automation can overcome the inefficiencies of manual assessments [7,8]. In particular, deploying autonomous robots to survey damaged buildings can improve safety by reducing human exposure to collapse-prone areas and speed up the collection of damage information [9,10]. Post-earthquake

inspection can also be interpreted as a time-critical risk management task rather than a purely visual recognition problem. Decisions are often made under uncertainty, with incomplete access to damaged zones and rapidly changing safety conditions. From this perspective, robotic perception systems are valuable insofar as they improve situational awareness, reduce inspector exposure, and provide traceable evidence that supports triage and subsequent detailed assessment.

Legged robots, such as Unitree Go2 quadruped robot, have emerged as capable platforms for navigating complex and debris-strewn environments [9]. Quadruped robots (e.g., Boston Dynamics Spot or Unitree Go2) can traverse uneven terrain and stairs, making them suitable for post-disaster inspection tasks [8,9]. Researchers have demonstrated that these robots, equipped with appropriate sensors, can assist in search and rescue missions by exploring hazardous post-disaster areas and providing visual data from dangerous zones [11]. By mounting cameras and deploying artificial intelligence (AI) on such mobile platforms, it becomes feasible to automatically detect structural damage in real time as the robot patrols a disaster site [2,12]. This integration of field robotics with real-time computer vision offers a promising pathway toward rapid post-earthquake structural screening, particularly in environments where mobility and perception constraints are tightly coupled [9]. At the same time, practical deployments remain constrained by sensing conditions and onboard resources, including illumination variability, motion-induced blur, and strict latency and power budgets. These operational factors motivate model choices and system designs that balance accuracy with responsiveness.

Concurrently, the field of computer vision has seen significant progress in automated crack detection and damage recognition on civil structures [6,12]. Image-based crack detection has gained considerable attention as a non-destructive evaluation technique for infrastructure [4,13]. In particular, deep learning (DL) methods, including convolutional neural networks (CNNs), have been widely adopted for image-based damage detection because they can learn hierarchical visual features directly from data, reducing reliance on hand-crafted descriptors [4,12,13]. Nevertheless, reported performance is often contingent on imaging conditions and data distributions, and sensitivity can degrade under domain shift, low contrast, or motion blur. These constraints are particularly relevant for crack detection, where targets are thin, fragmented, and sometimes visually confounded with the surface texture. Numerous studies have applied DL to tasks like classifying damage types in buildings and bridges, with encouraging results that underscore the potential of AI for automated damage inspection [2,12,14]. One such approach is the “You Only Look Once (YOLO)” family of object detectors, which perform fast one-shot detection [15]. For instance, a recent study showed that a YOLOv5 model could automatically detect defects on bridge images with high effectiveness [4], illustrating the viability of one-stage detectors in structural inspection [16]. Building upon these advances, the latest YOLOv8 model extends capability beyond bounding-box detection to instance segmentation, enabling pixel-level delineation of cracks and spalls [17]. YOLOv8 features a one-stage architecture that achieves both high accuracy and real-time performance, offering competitive accuracy while preserving inference efficiency compared with heavier two-stage segmentation pipelines, which can be advantageous for time-constrained inspection settings [18,19]. Its superior inference speed and accuracy make YOLOv8 well-suited for real-time damage detection applications where timely and precise identification of cracks is essential [20].

While both legged robotic inspection and deep-learning-based crack detection have been studied extensively, fewer works have emphasized an implementation-level coupling between quadrupedal mobility and instance segmentation under near real-time constraints. In practice, the value of segmentation is realized only when it can be executed responsively during inspection and communicated in an operator-interpretable manner. Motivated by

this gap, we develop and evaluate an integrated quadruped-and-segmentation workflow for crack delineation during traversal. This study combines the mobility of a Unitree Go2 quadruped robot with the intelligence of a state-of-the-art vision algorithm to create a fast robot-assisted post-earthquake crack detection workflow [21]. A Unitree Go2 quadruped robot was integrated with an onboard YOLOv8-based instance segmentation model to enable near-real-time detection of structural cracks and other types of damage [22]. The robotic platform carries cameras that feed live video to the AI model, which in turn identifies and segments cracks or damaged regions on structural components as the robot traverses a site [9]. To ensure robust performance, the YOLOv8 model was trained on a high-quality dataset of crack images and annotated structural damage examples [23]. The quality of training data is crucial, as DL segmentation networks require extensive, well-labeled datasets to achieve reliable accuracy [4,6,12]. The proposed system was evaluated in a laboratory setting using a scaled concrete–steel frame structure with induced cracks and damage scenarios. During the experiments, the quadrupedal robot navigated around the test structure under operator supervision, while the perception pipeline processed the incoming camera stream to detect and segment visible cracks and surface anomalies. These preliminary trials suggest that the integrated robot and vision system can delineate damage regions with practical near-real-time responsiveness under laboratory conditions, supporting the feasibility of the proposed inspection workflow [24].

This study offers three practical contributions. First, we present an end-to-end inspection workflow that couples a quadrupedal robot with YOLOv8 instance segmentation for pixel-level crack delineation during traversal. Second, we describe an integration strategy (camera streaming, inference, and visualization) that can be adapted to similar mobile SHM deployments. Third, we report a laboratory-scale validation that illustrates both the potential benefits and current limitations of near real-time segmentation for post-event screening. The present study focuses on laboratory-scale feasibility and does not yet claim full autonomy or broad field generalization across diverse post-disaster conditions.

2. Methodology

2.1. System Overview and Hardware Integration

The proposed system is built around a quadruped robotic inspection platform equipped with imaging sensors and onboard computational resources. The Unitree Go2 quadruped robot (Unitree Robotics, Hangzhou, China), a commercially available quadrupedal robot, was selected due to its high mobility, payload capacity, and ability to maintain stability while traversing uneven terrain and stairs. These capabilities are particularly critical for post-earthquake inspection scenarios, where environments are often cluttered with rubble and debris [8].

The robot is equipped with a built-in camera that captures images of structural surfaces during inspection tasks. For onboard processing, the EDU expansion dock of the Unitree Go2 platform was used. This onboard computing unit is based on an NVIDIA Orin Nano 8 GB platform (NVIDIA Corporation, Santa Clara, CA, USA) and includes a 6-core ARMv8 CPU operating up to approximately 1.5 GHz, 8 GB of RAM, and an NVIDIA Ampere GPU. The computing unit was used to support image-stream handling and AI-based damage detection during the laboratory deployment. The entire hardware configuration was designed to operate in an untethered manner and was controlled wirelessly, enabling the robot to access damaged or structurally unstable areas while allowing the human operator to remain at a safe distance [25]. The detailed power consumption of the onboard inference unit was not separately measured in the present study and will be considered in future field-oriented evaluations.

Software integration is achieved using the Robot Operating System (ROS 2 Humble) framework in conjunction with the Unitree Software Development Kit (SDK). The ROS environment enables efficient communication between the robot's locomotion subsystem and the perception and inference modules. Dedicated ROS nodes were developed to handle camera stream acquisition, motion command transmission, and AI model inference deployment. While the Unitree Go2 quadruped robot's native control interface is utilized for low-level navigation, the system architecture allows for the integration of higher-level autonomy, such as predefined inspection or scanning paths.

During operation, the live video stream captured by the robot's camera is transmitted to the onboard computing unit, where each frame is processed by the YOLOv8-based instance segmentation model to identify cracks and other forms of structural damage. When potential damage is detected, its spatial location is marked in software and immediately communicated to the operator. An overview of the complete robotic inspection setup and the overall system workflow is presented in Figure 1.

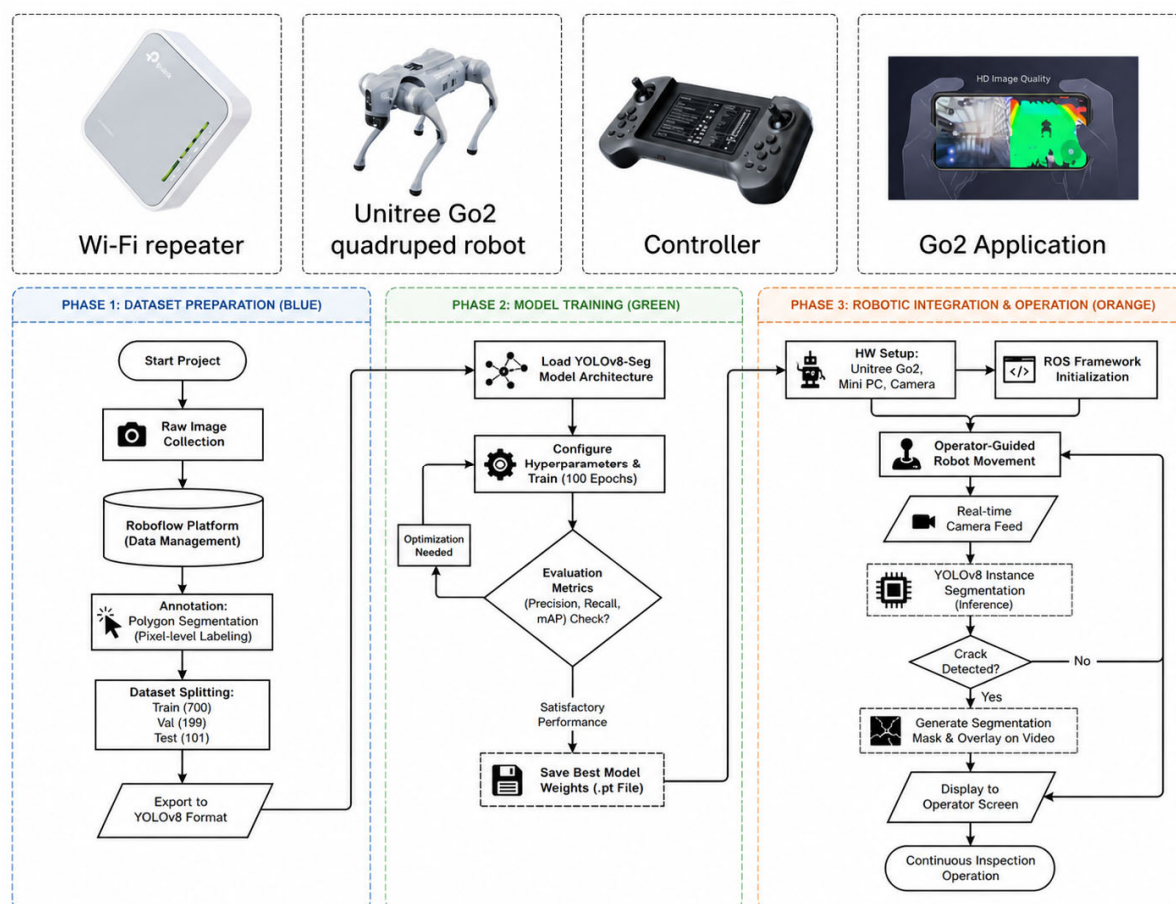


Figure 1. Unitree Go2 quadruped robot setup and overall system workflow.

2.2. YOLOv8 Segmentation Model for Damage Detection

Conventional object detection methods are primarily limited to localizing objects within an image and representing them using bounding boxes. In contrast, instance segmentation operates at the pixel level, enabling precise separation of object pixels from the background and from other objects, thereby allowing accurate delineation of object shapes [26,27]. This fine-grained representation supports a broader range of applications compared to traditional object detection approaches, particularly in scenarios requiring detailed structural analysis.

Building on the success of YOLO-based object detection frameworks, YOLOv8 extends these capabilities to instance segmentation while maintaining high accuracy and computational efficiency [28]. The YOLOv8 segmentation model demonstrates robust performance across objects of varying scales, closely spaced objects, and challenging viewpoints, while also offering relatively high inference speeds suitable for real-time and near-real-time applications. Its compact model architecture further facilitates deployment on platforms with limited computational resources, such as mobile and robotic systems [15,18,29].

Another key advantage of YOLOv8 lies in its efficient and user-friendly implementation framework provided by Ultralytics. The intuitive application programming interface enables seamless integration of segmentation models into practical systems, supporting rapid development and deployment in real-world scenarios. As illustrated in Figure 2, the YOLOv8 instance segmentation architecture employed in this study is specifically configured to enable pixel-level crack and damage detection, making it well-suited for structural inspection tasks.

Model	File Names	Task	Inference	Validation	Training	Export
YOLOv8	yolov8n.pt	Detection	✓	✓	✓	✓
	yolov8s.pt					
	yolov8m.pt					
	yolov8l.pt					
	yolov8x.pt					
YOLOv8-seg	yolov8n-seg.pt	Instance Segmentation	✓	✓	✓	✓
	yolov8s-seg.pt					
	yolov8m-seg.pt					
	yolov8l-seg.pt					
	yolov8x-seg.pt					

Figure 2. Architecture of the YOLOv8 instance segmentation model employed for pixel-level crack detection.

From an architectural perspective, YOLOv8 incorporates recent advances in deep learning, including convolutional neural networks and multi-scale feature extraction mechanisms. In particular, the use of feature pyramid representations allows the model to effectively capture both global contextual information and fine-grained details, which is critical for accurate segmentation in complex scenes [30]. This multi-scale representation enhances boundary localization and contributes to the generation of high-quality segmentation masks, thereby improving overall damage detection performance.

In this study, YOLOv8 was selected because the main objective was to develop and evaluate a robot-integrated crack segmentation workflow with efficient inference and practical deployment capability. Compared with heavier two-stage instance segmentation models such as Mask R-CNN, YOLO-based segmentation models offer practical advantages for resource-constrained robotic applications, particularly in terms of inference efficiency and implementation simplicity. YOLOv5-seg and Mask R-CNN are relevant baseline approaches for future comparison; however, the present study focuses primarily on system integration and laboratory feasibility rather than comprehensive model benchmarking. Future work should include direct quantitative comparison using the same training and test splits to further evaluate the trade-off between segmentation accuracy and deployment efficiency.

2.3. Training Dataset and Annotation

The dataset used in this study was organized and curated using the Roboflow platform, which provides an integrated workflow for dataset management, preprocessing, annotation, and export for computer vision applications. Roboflow enables controlled dataset versioning and facilitates deployment-oriented data handling through its interface and software tools.

To increase visual diversity and support more robust learning, the dataset was assembled from multiple sources, capturing variations in surface texture, illumination, and crack morphology. All images were collected and used in accordance with their respective licenses and institutional guidelines. To reduce the risk of information leakage, the training, validation, and test splits were created at the image level, and we verified that near-duplicate images were not shared across subsets.

Although the dataset was curated to include variations in crack morphology, illumination, and surface texture, it was not specifically composed of verified post-earthquake field images. Therefore, the present dataset provides a suitable basis for evaluating the proposed crack segmentation workflow, while future studies should expand the dataset with real post-earthquake field images to further assess generalization under disaster-site conditions.

Image annotation was performed using Roboflow Annotate, which supports pixel-level labeling for instance segmentation tasks. Given the thin, fragmented, and often irregular geometry of cracks, annotation followed a standardized polygon protocol to promote consistent boundary definition. A two-stage quality control procedure was applied. First, annotations were reviewed for polygon ordering, closure, and evident boundary errors. Second, a subset of samples was re-checked to ensure that visually ambiguous regions, such as stains, texture edges, or surface artifacts, were not systematically mislabeled as cracks. While residual label noise is difficult to eliminate in practice, these steps aimed to improve annotation consistency and reduce systematic bias.

The final dataset comprises 1000 crack images, each annotated in YOLOv8 instance segmentation format to capture crack shape and location. Although the dataset contained 1000 image files, the total number of annotated crack instances was approximately 1600 because several images contained more than one separated or fragmented crack region. Therefore, the image count and the instance count represent two different dataset levels: the former refers to the number of images, while the latter refers to the number of individual crack annotations used for instance segmentation.

For model development and evaluation, the dataset was partitioned into three subsets: 700 images for training, 199 images for validation, and 101 images for independent testing. The training set was used to optimize model parameters, the validation set supported model selection and training monitoring, and the test set was reserved for final performance assessment [18]. Representative examples of images and corresponding polygon-based annotations are shown in Figure 3.

All annotations were generated in accordance with the YOLOv8 segmentation specification. Each label file begins with a class identifier, followed by a sequence of normalized coordinate pairs (x, y) defined on the [0, 1] range with respect to image dimensions. The ordered coordinate pairs form a polygon that outlines the object contour, enabling the model to learn pixel-level boundary information. Compared with conventional bounding-box annotation, this polygonal representation better captures irregular geometries and supports precise delineation of crack boundaries during inference [26].

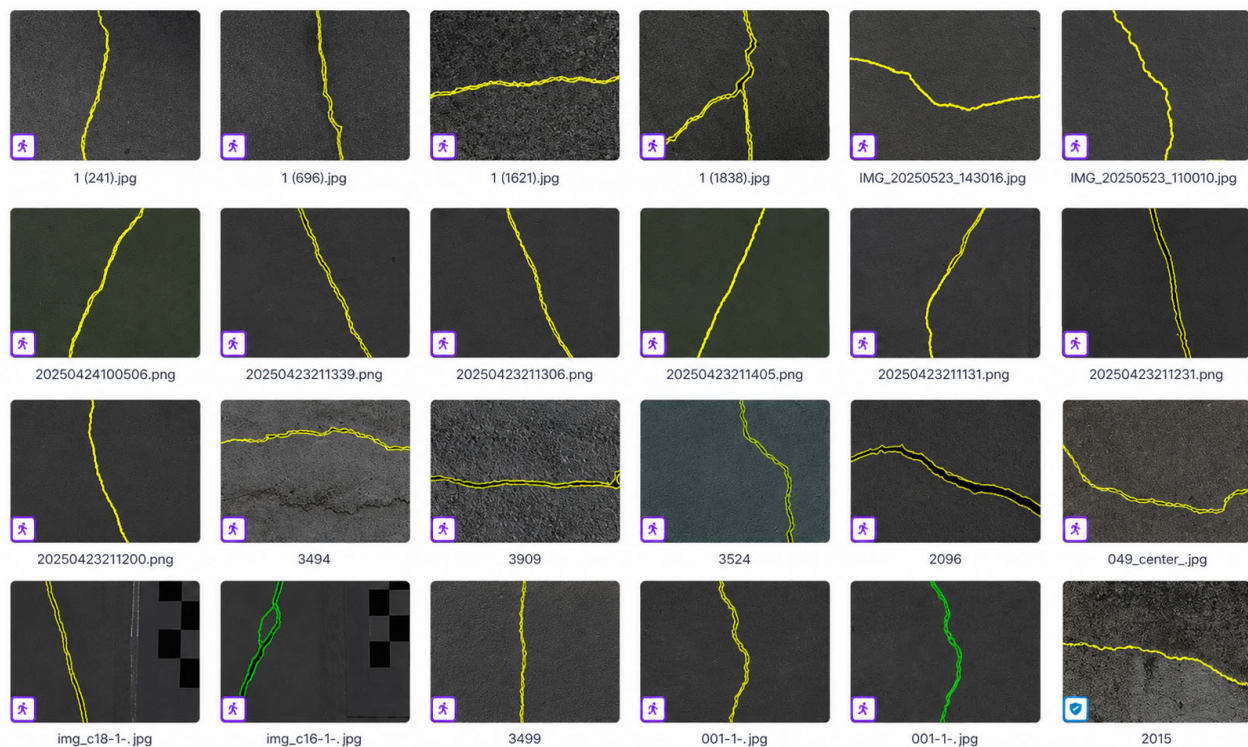


Figure 3. Representative examples of crack images with corresponding YOLOv8 instance segmentation annotations used in the training dataset.

2.4. Model Training Procedure

The YOLOv8 model was trained for segmentation purposes. The code block required to initiate the training process was obtained from the official webpage of the Ultralytics YOLOv8 application. After making adjustments to the code in accordance with the project requirements, the training process was initiated. The training was planned for a total of 100 epochs and was completed in approximately 0.962 h. The reported training duration is presented as an indicator of computational feasibility and should not be interpreted as direct evidence of segmentation accuracy. Pixel-level delineation performance was evaluated using the training and validation loss curves, precision, recall, mAP values, and qualitative segmentation mask outputs.

2.5. Operational Assumptions and Failure Modes

The current prototype assumes sufficient illumination and a camera-to-surface distance that preserves crack visibility. The Unitree Go2 platform is equipped with a front lighting option that can support visual inspection under low-illumination conditions. However, during the laboratory experiments, ambient illumination was sufficient, and this lighting function was not required. Performance may still degrade for extremely low-contrast cracks, strong specular reflections, or significant motion blur during fast maneuvers. These limitations are partly mitigated by controlled robot speed during close-range inspection, but they remain relevant for field deployments and motivate future work on robustness and uncertainty-aware reporting. Therefore, low-light robustness remains a topic for future quantitative evaluation.

Similarly, although robot speed was controlled during close-range inspection to reduce motion blur, a separate quantitative analysis of the relationship between robot speed and detection accuracy was not conducted in the present study and will be addressed in future experiments.

3. Results and Discussion

3.1. Training and Validation Results

As shown in Figure 4, both the training and validation loss curves exhibit a steady downward trend throughout the training process, indicating stable convergence of the model. The close alignment between training and validation losses suggests that the model learned representative features from the dataset without exhibiting significant divergence between the two phases.

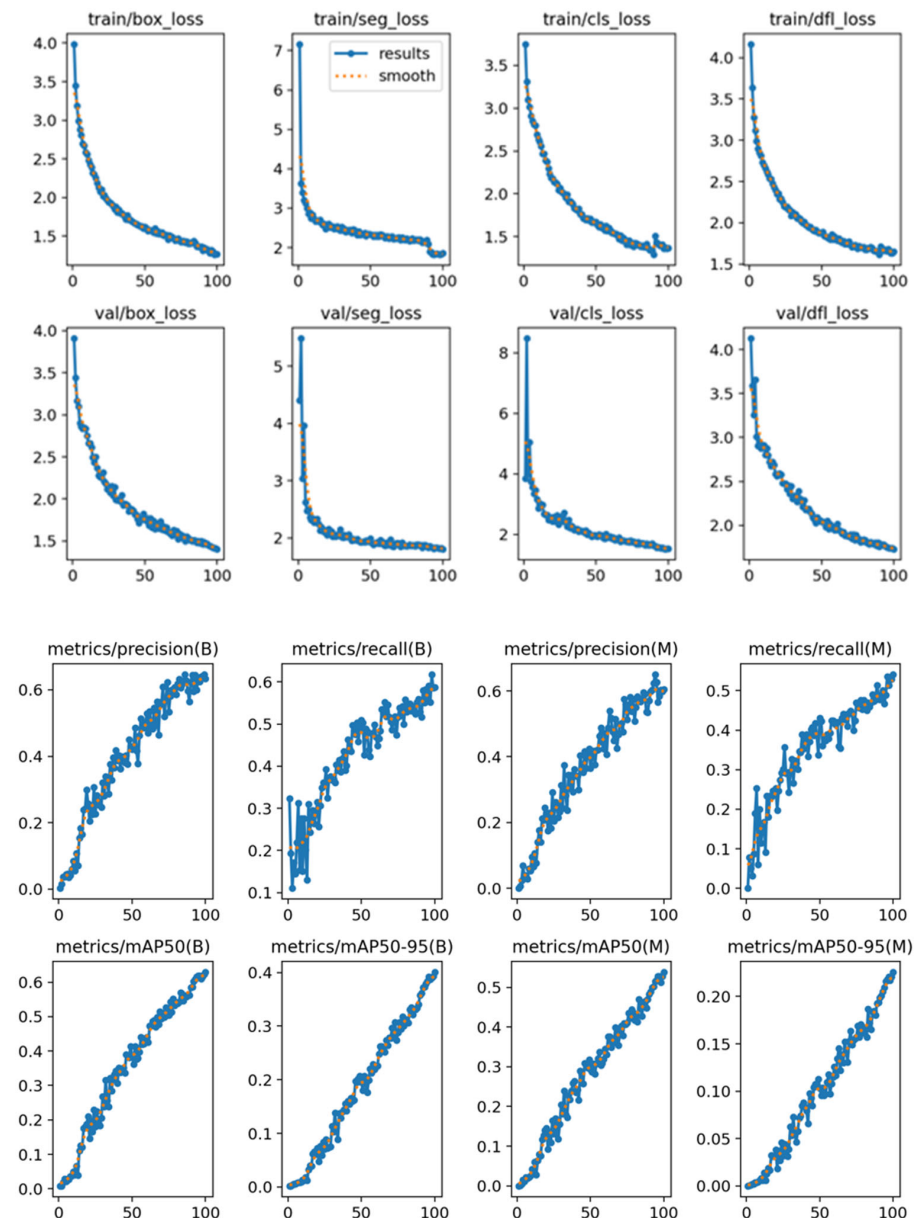


Figure 4. Training and validation loss curves and performance metrics of the YOLOv8 segmentation model across epochs, including box and mask precision, recall, and mAP scores.

In addition, the absence of a clear divergence between training and validation trends suggests that severe overfitting was not observed; however, the moderate recall and mAP values indicate that further dataset expansion and model refinement are still required to reduce the risk of underfitting to diverse crack appearances.

Regarding performance metrics, consistent improvements are observed in precision, recall, mAP@50, and mAP@50–95 for both bounding box (B) and segmentation mask (M)

predictions [31]. The increase in precision and recall reflects enhanced detection reliability, characterized by a higher true positive rate and a reduction in false positive predictions. In addition, the continuous growth of mAP values, particularly at stricter evaluation thresholds such as mAP@50–95, demonstrates improved localization and segmentation accuracy across varying levels of overlap.

Overall, the combined trends in loss reduction and performance metric improvement indicate that the YOLOv8 segmentation model was trained effectively. The balanced progression of training and validation curves, together with the steady enhancement of detection and segmentation metrics, suggests that the model achieved stable training behavior and practical segmentation performance under the evaluated dataset conditions.

3.2. Confusion Matrix Analysis

Figure 5 presents the normalized confusion matrix summarizing the classification behavior of the model for crack and background classes [32]. The results indicate that the model exhibits a strong ability to correctly identify background regions, with all true background samples being correctly classified. This suggests a high true negative rate, reflecting the model's effectiveness in distinguishing non-damaged areas.

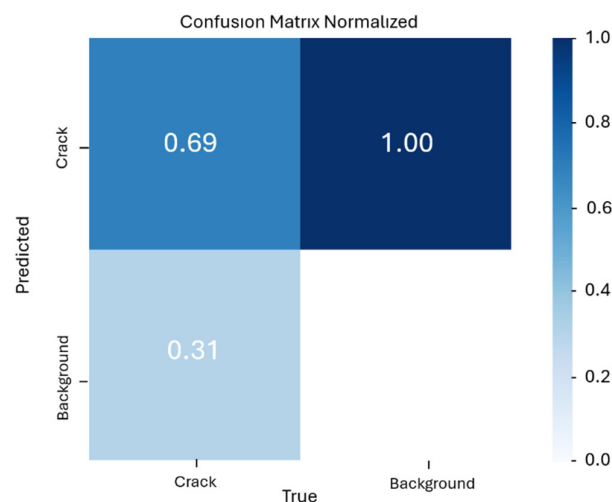


Figure 5. Normalized confusion matrix illustrating classification performance for crack and background classes.

In contrast, the performance for the crack class is comparatively lower. Approximately 69% of true crack samples were correctly classified as crack, while the remaining 31% were misclassified as background. This level of misclassification indicates that the model occasionally fails to detect cracks, which can be attributed to factors such as thin crack structures, low contrast between cracks and the surrounding surface, or visual similarities between crack patterns and background textures.

Overall, the confusion matrix highlights an imbalance in classification performance between the two classes, with background detection being more reliable than crack identification. This limitation suggests that further improvements in crack sensitivity are required. Potential strategies include increasing the diversity and quantity of crack samples in the training dataset, applying targeted data augmentation techniques, or refining the model architecture and loss weighting to better emphasize crack-related features.

This performance disparity should be considered when interpreting the reliability of the proposed system for post-earthquake inspection. The high background recognition rate indicates that the model can effectively distinguish non-crack regions under the tested conditions. However, the lower crack recognition rate shows that some crack patterns may still be

missed, particularly when cracks are thin, low-contrast, or visually similar to the surrounding surface. Therefore, the current results demonstrate the feasibility of operator-supervised crack screening, while also indicating the need for further improvement in crack recall before broader field deployment. In practical inspection scenarios, the model outputs can support inspectors by highlighting potential crack regions, but they should be interpreted together with expert judgment and, where possible, repeated observations from different viewpoints.

As a future extension, the visual segmentation output of the proposed system could be complemented by additional sensing technologies. For example, visual crack localization from the quadruped platform could be combined with stress, strain, pressure, or vibration-based sensing to support a more comprehensive multimodal assessment of structural condition. Such integration may help reduce the uncertainty associated with visual-only inspection, particularly in cases where cracks are low-contrast, partially occluded, or difficult to interpret from images alone.

3.3. Dataset Distribution and Correlation Analysis

The visualizations presented in Figure 6 illustrate the spatial distribution and size characteristics of the annotated crack instances within the training dataset. The dataset contains a single target category (crack), comprising approximately 1600 annotated instances. While class imbalance is not applicable in a single-class setting, the analysis focuses on the spatial and geometric distribution of annotations, as these factors can significantly influence model learning behavior and generalization performance [33].

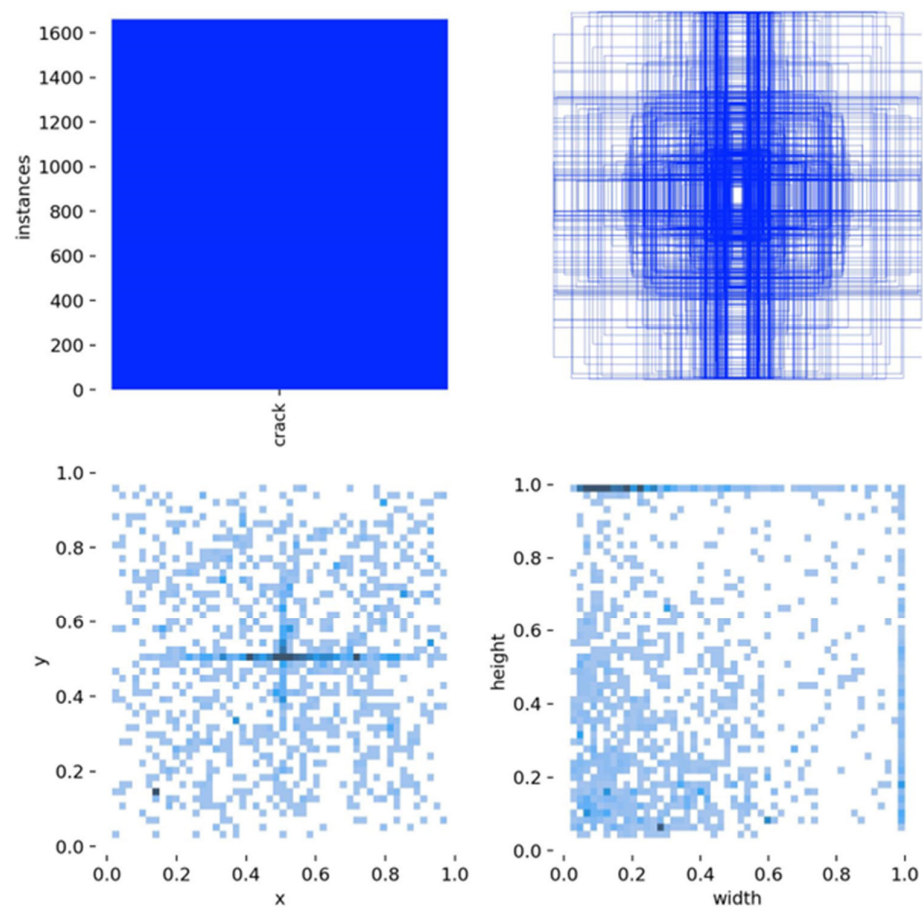


Figure 6. Dataset distribution and correlation analysis showing spatial concentration and size characteristics of annotated crack instances.

Although the proposed approach is based on instance segmentation, bounding box statistics are analyzed in this section as a complementary representation to assess potential spatial and scale-related biases in the dataset. Bounding box parameters provide an effective and widely adopted means of summarizing object location and size distributions, which are otherwise more difficult to quantify directly from pixel-level segmentation masks.

The overlay of all annotated bounding boxes reveals a pronounced concentration of instances in the central regions of the images, with a gradual decrease toward the image boundaries. This pattern is further supported by the density distributions of the bounding box center coordinates, which show clustering around the central area. Such spatial bias suggests that the model is more frequently exposed to centrally located crack instances during training, which may reduce sensitivity to cracks occurring near image edges [34]. This issue was considered as a dataset-related limitation, and future dataset expansion should include more boundary-adjacent crack samples and targeted augmentation strategies, such as translation, cropping, and mosaic-based augmentation, to improve edge-crack detection capability.

In addition, the distributions of bounding box width and height indicate a higher density toward smaller sizes, reflecting the inherently thin and elongated nature of cracks. This observation is consistent with the physical characteristics of crack structures; however, it also implies that the dataset predominantly represents small-scale objects. Consequently, while the model may perform well on narrow crack patterns, its ability to generalize to larger or more complex damage structures may be limited [35].

The cross-correlation analysis shown in Figure 7 further examines the relationships between bounding box center coordinates and size parameters. The marginal distributions of the x and y coordinates indicate a tendency toward central placement, with the y -axis exhibiting slightly stronger central clustering. In contrast, the width and height distributions are more uneven. Notably, a subset of annotations exhibits height values close to 1.0, suggesting the presence of vertically elongated cracks that span most of the image height [36].

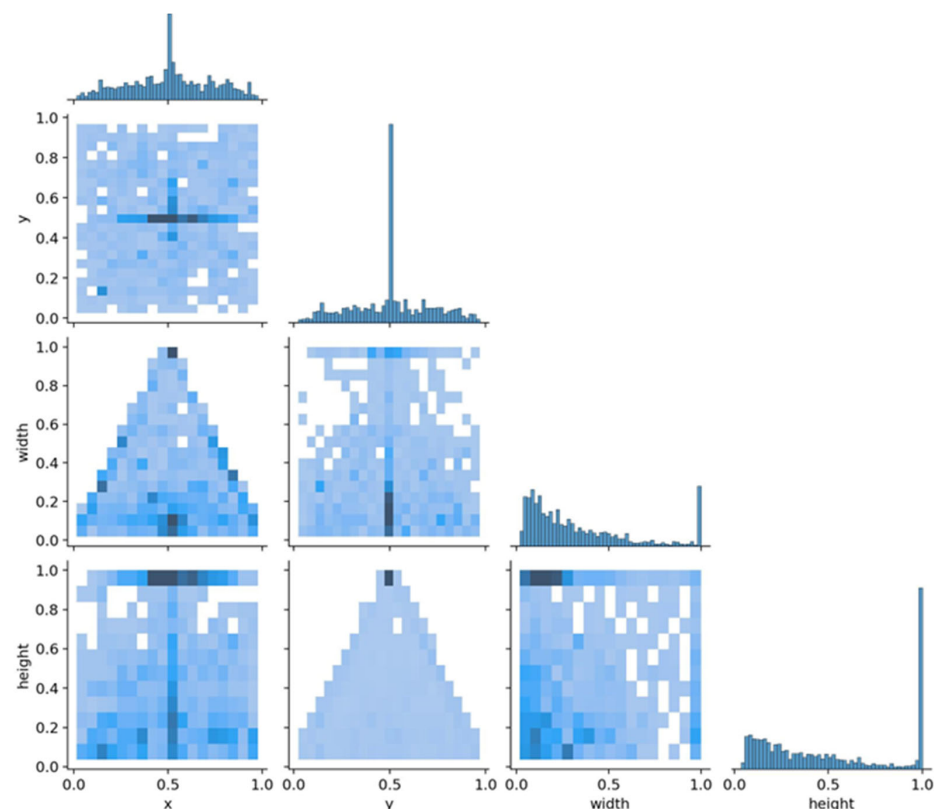


Figure 7. Cross-correlation analysis of bounding box parameters (center coordinates, width, and height) for crack annotations.

The bivariate density plots reveal limited but meaningful correlations among certain parameter pairs, such as width versus height and x -coordinate versus width. The presence of dense clusters and sparsely populated regions indicates constrained variability in object size and placement, which may influence the model's robustness when exposed to crack instances with uncommon geometries or spatial configurations.

Overall, the analysis highlights a non-uniform distribution of crack instances in terms of both spatial location and size. While this distribution supports effective learning for centrally located and small-scale cracks, it may introduce bias against boundary-adjacent or less frequently represented crack patterns. To mitigate these effects, future dataset enhancements could include targeted data augmentation, inclusion of samples with greater positional and scale diversity, and adaptive sampling strategies during training [37].

3.4. Confidence-Based Performance Evaluation

The F1–Confidence curve presented in Figure 8 illustrates the relationship between the F1 score and the prediction confidence threshold for the crack class. The confidence threshold determines the minimum probability required for a prediction to be considered valid, thereby directly influencing the balance between precision and recall. As shown in the figure, the F1 score increases from low confidence levels, reaches a maximum value of approximately 0.61 at a confidence threshold of 0.316, and subsequently declines as the threshold increases further. This behavior indicates that the model achieves its best balance between precision and recall at moderate confidence levels. At lower thresholds, the inclusion of low-confidence predictions increases recall but reduces precision, whereas higher thresholds suppress false positives at the cost of missing true crack instances. Therefore, excessively high confidence thresholds reduce the number of detected cracks, leading to a decline in the F1 score. The F1–Confidence curve thus provides a practical guideline for selecting an operating threshold that optimally balances detection sensitivity and reliability [38].

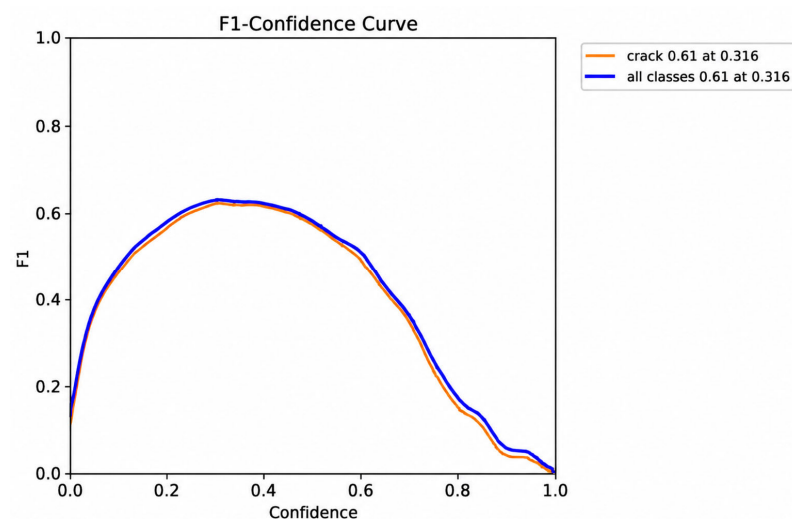


Figure 8. F1–Confidence curve.

However, the confidence threshold of 0.316 should not be interpreted as a universal operating threshold for safety-critical inspection. This value represents the dataset-specific threshold at which the F1 score reached its maximum in the present evaluation. In post-earthquake screening, the operating threshold may need to be adjusted according to the inspection objective. For example, if minimizing missed cracks is prioritized, a lower confidence threshold may be preferred, although this would increase false-positive detections and require additional expert review. Therefore, threshold selection should be treated as an application-dependent decision rather than a fixed model property.

The Precision–Confidence curve shown in Figure 9 further confirms this trade-off. Precision increases monotonically with higher confidence thresholds, reaching a value of 1.00 at a confidence level of approximately 0.908. This result indicates that predictions retained at very high confidence levels are highly reliable and associated with a minimal false positive rate. However, such operating points typically correspond to reduced recall, as a significant number of true crack instances are filtered out.

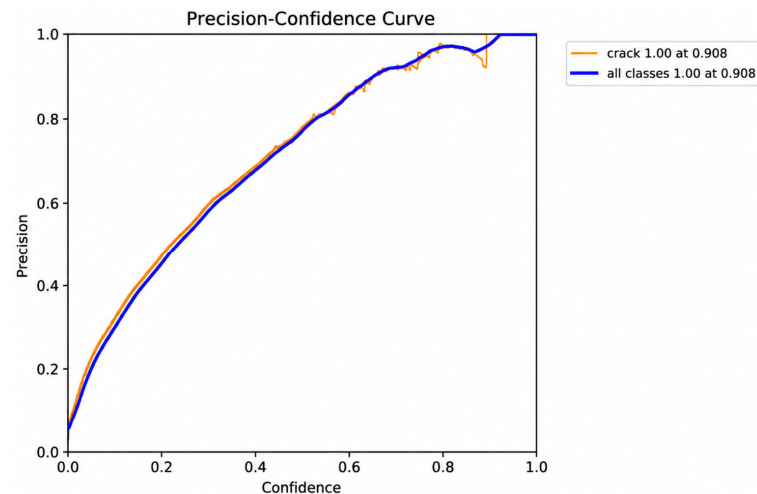


Figure 9. Precision–Confidence curve.

The precision–recall trade-off is explicitly visualized in the Precision–Recall curve presented in Figure 10. As recall increases, precision gradually decreases, reflecting the inherent compromise between detecting a larger proportion of cracks and maintaining prediction reliability. The area under this curve corresponds to an mAP@0.5 value of 0.629, which is achieved both overall and specifically for the crack class. In the context of instance-segmentation-based crack detection, this value indicates a moderate but consistent detection performance across varying confidence thresholds and supports the feasibility of the proposed robot-vision integration under the tested conditions. From an application perspective, this level of performance is promising for preliminary crack localization, rapid visual documentation, and supporting inspector awareness during post-earthquake screening. Nevertheless, for safety-critical structural assessment, the model outputs should be interpreted as decision-support information and complemented by expert evaluation and, where possible, additional observations or sensing methods.

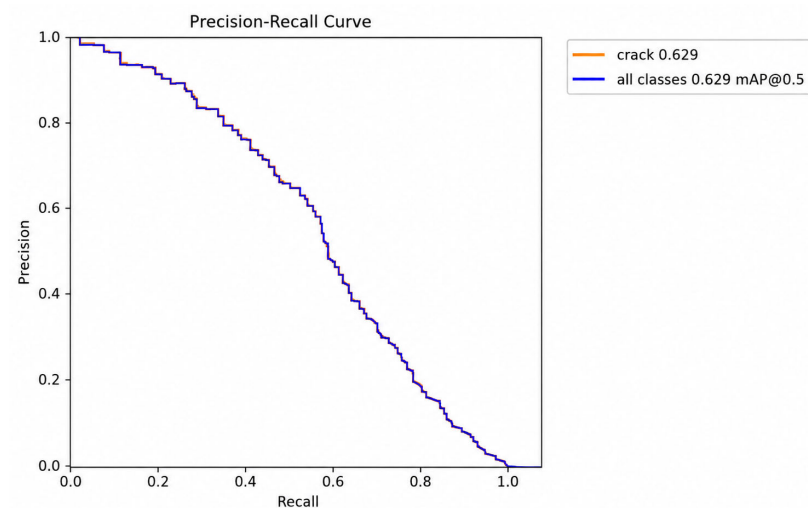


Figure 10. Precision–Recall curve.

Overall, the confidence-based evaluation demonstrates that while very high confidence thresholds yield highly precise predictions, a moderate confidence setting provides a more balanced performance by preserving recall without excessively sacrificing precision. These findings highlight the importance of confidence threshold selection in practical deployment scenarios, particularly for safety-critical inspection tasks where missed detections can be as detrimental as false alarms.

3.5. Qualitative Results on Training and Validation Batches

The batch visualization shown in Figure 11 provides qualitative insight into the diversity of training samples and the spatial distribution of annotations used during model learning [39,40]. The training images include cracks observed on different surface types and under varying lighting, texture, and environmental conditions. Such visual diversity is beneficial for improving the robustness and generalization capability of the model [41]. The alignment between the annotated regions and the visible crack structures indicates that the annotations are generally consistent with the underlying crack geometry.

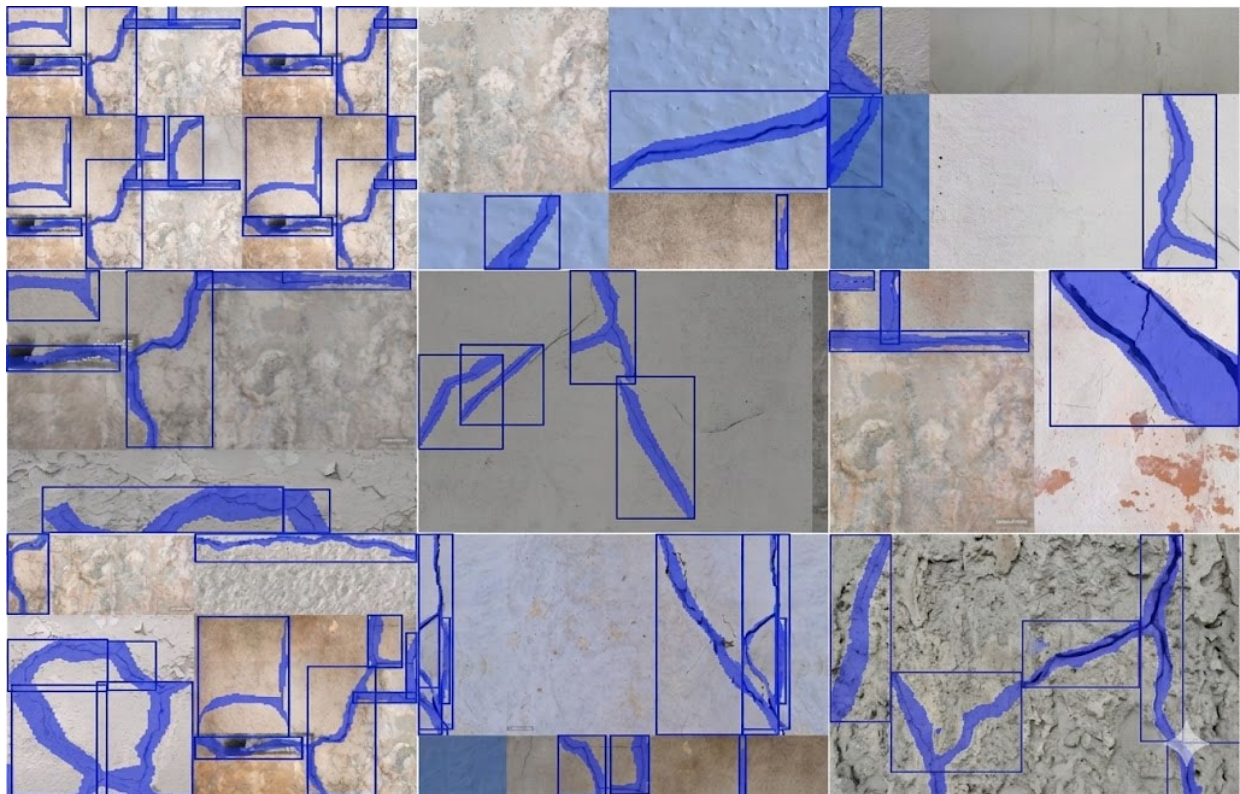


Figure 11. Visualization of a training batch showing crack annotations and image diversity used during model learning.

In several samples, however, a dense concentration of overlapping or closely spaced annotations can be observed [42,43]. This mainly results from the fragmented and irregular morphology of crack patterns and the fine-grained instance-level annotation strategy used during dataset preparation. When visually separated crack branches or discontinuous segments appeared in the same image, they were annotated as separate crack instances. This approach was adopted to preserve detailed crack geometry for instance segmentation, although future dataset refinement may further improve annotation consistency.

The qualitative validation results presented in Figure 12 illustrate the model's inference behavior on unseen data. Overall, the model demonstrates a strong capability to detect cracks and to delineate their spatial extent with reasonable accuracy, despite the thin and

elongated nature of crack structures. Predicted regions with confidence scores in the approximate range of 0.6 to 0.8 generally exhibit good spatial correspondence with ground truth annotations, indicating reliable detection performance at moderate confidence thresholds.

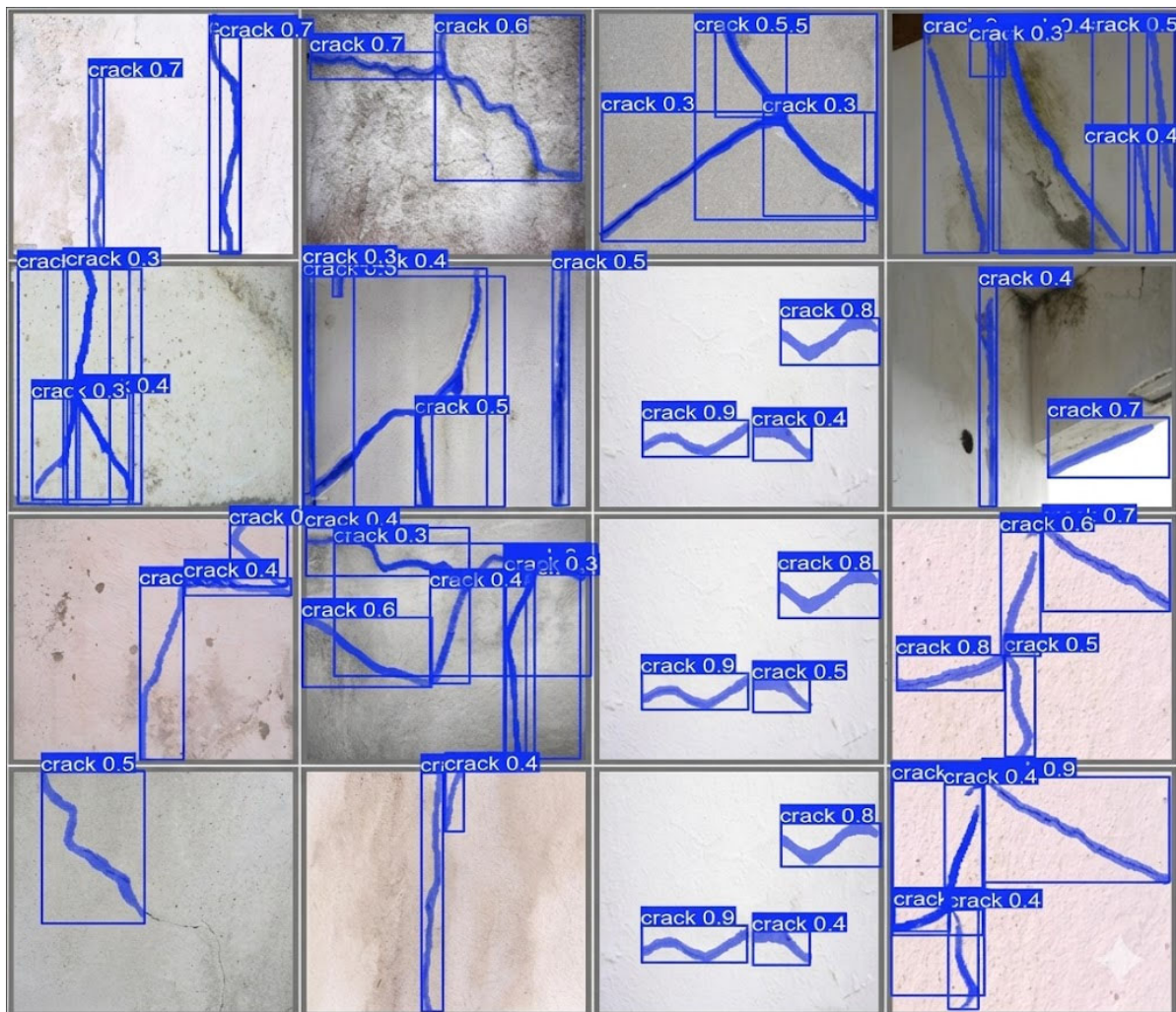


Figure 12. Validation batch results illustrating predicted crack segmentation masks and confidence scores.

The visualization also reveals that some correctly localized crack detections are assigned relatively low confidence scores, suggesting that the model is able to capture subtle crack patterns under uncertain visual conditions. However, lowering the confidence threshold excessively may lead to an increased number of false-positive detections. In certain cases, the model fails to detect cracks entirely or captures them only partially, with long or inclined cracks being fragmented into multiple segments or incompletely delineated. This behavior indicates reduced sensitivity to extended, low-contrast, or obliquely oriented crack patterns.

Additionally, the presence of multiple overlapping detections around the same crack suggests a degree of redundancy in the predictions. This observation implies that further optimization of post-processing steps, such as non-maximum suppression parameters or confidence-based filtering, may improve detection consistency and reduce duplicate predictions. Overall, the qualitative results highlight both the strengths of the proposed approach in capturing diverse crack patterns and the remaining challenges associated with complex crack geometries and confidence calibration.

3.6. Laboratory Robotic Deployment Results

To evaluate the practical applicability of the proposed YOLOv8-based crack segmentation model, a series of laboratory robotic deployment experiments was conducted using an integrated robotic inspection system. The primary objective of these experiments was to assess the model's ability to detect and delineate structural cracks at the pixel level under laboratory operating conditions, beyond controlled training and validation environments.

A laboratory-scale reinforced concrete–steel frame specimen was used as the test structure, providing a representative laboratory surface for evaluating segmentation-based crack detection under controlled experimental conditions. The specimen, shown in Figure 13, was designed to include multiple observation surfaces and realistic crack patterns suitable for experimental inspection. The reinforced concrete specimen was constructed using a concrete strength class of C25/30 and reinforcing steel of class B420C, ensuring sufficient structural integrity and durability. The overall dimensions of the specimen were $550 \times 1700 \times 300$ mm (width $\times$ length $\times$ height), with a total structural height of 1850 mm. These dimensions provided adequate surface area and geometric diversity for evaluating crack detection performance from different viewpoints.

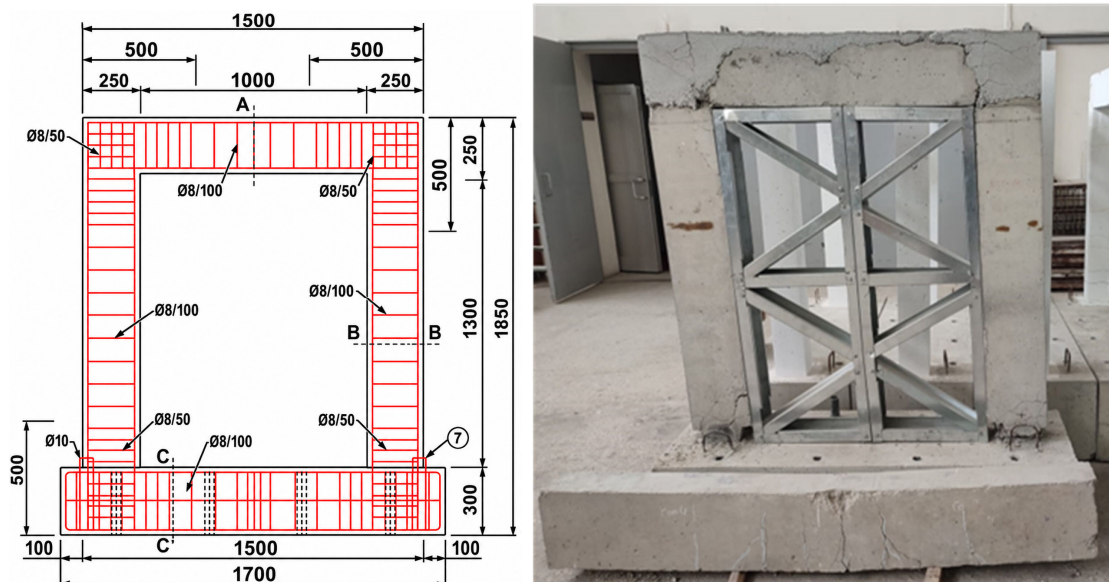


Figure 13. Laboratory-scale reinforced concrete–steel frame specimen used for laboratory crack detection experiments.

It should be noted that surface texture can significantly influence segmentation accuracy. Although the C25/30 reinforced concrete specimen provided a controlled and repeatable laboratory surface, real post-earthquake environments may include dust, stains, spalling, exposed aggregates, moisture, shadows, and irregular textures. These conditions may reduce the visual contrast between cracks and the surrounding surface and may increase the risk of both missed detections and false positives. Therefore, the laboratory results should be interpreted within the surface conditions tested in this study.

During the experiments, the Unitree Go2 quadrupedal robot was activated and operated via remote control, as illustrated in Figure 14. The robot's onboard camera continuously captured images of the specimen surfaces while the robot was maneuvered around the structure. The captured image stream was transmitted to the onboard computing unit, where it was processed in near real time by the YOLOv8 instance segmentation model.



Figure 14. Activation and operation of the Unitree Go2 quadrupedal robot during experimental inspection tests.

The frame rates observed during the laboratory robotic deployment were approximately 3.28–3.43 FPS, as shown in Figure 15. This observed frame rate reflects the end-to-end operation of the system, including image-stream transmission, onboard processing, instance segmentation inference, and visualization. Although the camera/sensor hardware is capable of operating at higher frame rates, the effective deployment FPS was influenced by the overall processing and communication pipeline. Under the slow, operator-supervised close-range inspection conditions used in this study, this near-real-time performance was sufficient to demonstrate the integrated workflow, while further optimization of hardware acceleration, network configuration, input resolution, and model size may improve deployment speed in future work.

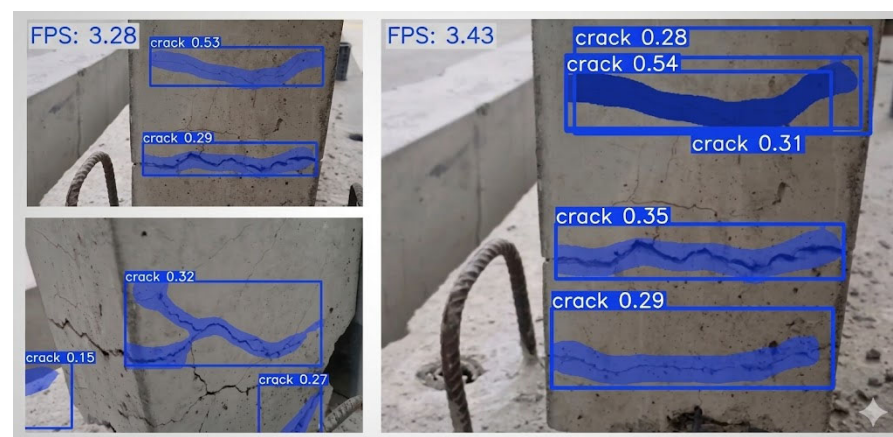


Figure 15. Near-real-time crack segmentation results obtained during laboratory testing, showing pixel-level delineation of detected cracks.

The integration between the robotic platform and the segmentation model was implemented using a Python 3.10.12-based processing pipeline. This pipeline handled image acquisition, preprocessing, model inference, and visualization of segmentation outputs. Operating synchronously with the robot's movement, the system enabled inspection from multiple viewing angles, thereby increasing surface coverage and improving the likelihood of detecting cracks with varying orientations and appearances.

The near-real-time segmentation results obtained during laboratory testing are presented in Figure 15. The model successfully detected cracks and delineated their boundaries at the pixel level, even for thin and irregular crack patterns. The generated segmentation masks clearly indicate the spatial extent and geometry of the detected cracks, demonstrating the model's capability to transfer learned crack features from the training dataset to the tested laboratory structural surface.

Overall, the experimental results demonstrate that the proposed YOLOv8-based segmentation approach can be integrated with a mobile robotic platform for laboratory-based

crack detection tasks. The successful integration of the model with the quadrupedal robot and its reliable performance under laboratory conditions highlight the potential of the proposed system for structural health monitoring and post-event inspection applications.

The laboratory robotic experiment complements the quantitative model evaluation by demonstrating the practical operation of the integrated system during operator-supervised inspection. In the present study, precision, recall, and mAP were evaluated using the independent test dataset, while the robotic deployment focused on verifying image acquisition, onboard inference, and visualization under laboratory conditions. Further work can extend this validation by annotating frames from robotic inspection videos and calculating frame-level detection and segmentation metrics during robot operation.

4. Concluding Remarks

This study presented an integrated robotic inspection framework that combines a quadrupedal mobile robot with a YOLOv8-based instance segmentation model to support operator-supervised crack detection in post-earthquake structural assessment scenarios. By leveraging the mobility of a quadrupedal robotic platform and the pixel-level delineation capability of modern deep-learning-based vision models, the proposed system addresses key limitations of traditional manual inspections, including safety risks, limited accessibility, and time constraints.

Comprehensive quantitative and qualitative evaluations demonstrated that the YOLOv8 segmentation model achieved stable convergence during training and exhibited consistent detection performance across validation and test datasets. The model effectively identified crack patterns of varying sizes and orientations, while confidence-based analyses highlighted the importance of appropriate threshold selection to balance precision and recall in safety-critical inspection tasks.

The proposed approach was further evaluated through laboratory robotic deployment using a reinforced concrete–steel frame specimen. The deployment of the segmentation model on the Unitree Go2 quadruped robot demonstrated the feasibility of near-real-time crack detection and pixel-level boundary delineation under operator-supervised laboratory conditions. These results suggest that the model can transfer learned crack features to the tested laboratory specimen, while further validation is required to assess generalization across more complex post-earthquake field conditions.

Overall, the findings indicate that the integration of instance-segmentation-based DL models with mobile robotic platforms offers a promising assistive approach for preliminary crack screening and visual documentation in post-earthquake structural inspection.

Since the experimental validation was conducted on a laboratory-scale specimen under controlled conditions, the findings should be interpreted primarily as evidence of system feasibility, while broader validation in complex post-earthquake environments remains an important direction for future work.

While the current study focused on laboratory-scale validation, future work will aim to extend the system toward higher levels of autonomy, incorporate multi-modal sensing, and evaluate performance in large-scale and field-deployed post-disaster environments. Such advancements have the potential to further enhance the reliability and operational impact of robotic AI-assisted inspection systems in real-world emergency response applications.

Author Contributions: Methodology, K.H. and A.B.; Software, K.H.; Formal analysis, A.M.; Investigation, A.B.; Resources, K.H. and A.B.; Data curation, V.K.; Writing—original draft, V.K. and A.M.; Writing—review & editing, K.H., V.K. and A.M.; Visualization, K.H. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Karadeniz Technical University Scientific Research Projects Coordination Unit, with support and funding from Research Grant No: FBA-2024-11049, Focused Research Universities Support Program Project (BAP13).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to acknowledge sincere thanks to Karadeniz Technical University Scientific Research Projects Coordination Unit. During the preparation of this manuscript, the authors used ChatGPT (OpenAI, GPT-5.5 Thinking) for minor language editing, grammar correction, and improving clarity of expression. No generative AI tool was used for data analysis, results generation, or scientific interpretation. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Jia, J.; Ye, W. Deep learning for earthquake disaster assessment: Objects, data, models, stages, challenges, and opportunities. *Remote Sens.* **2023**, *15*, 4098. [\[CrossRef\]](#)
2. Cha, Y.J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 731–747. [\[CrossRef\]](#)
3. Koch, C.; Georgieva, K.; Kasireddy, V.; Akinci, B.; Fieguth, P. A review on computer vision-based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Adv. Eng. Inform.* **2015**, *29*, 196–210. [\[CrossRef\]](#)
4. Hacıfendioğlu, K.; Altunışık, A.C.; Abdioğlu, T. Deep learning-based automated detection of cracks in historical masonry structures. *Buildings* **2023**, *13*, 3113. [\[CrossRef\]](#)
5. Hacıfendioğlu, K.; Demir, S. Deep learning-based automated crack detection for post-earthquake damage assessment in reinforced concrete structures. *Adv. Civ. Eng.* **2026**, *2026*, 6845779. [\[CrossRef\]](#)
6. Zhou, H.; Xu, C.; Tang, X.; Wang, S.; Zhang, Z. A review of vision-laser-based civil infrastructure inspection and monitoring. *Sensors* **2022**, *22*, 5882. [\[CrossRef\]](#)
7. Freimuth, H.; König, M. Planning and executing construction inspections with unmanned aerial vehicles. *Autom. Constr.* **2018**, *96*, 540–553. [\[CrossRef\]](#)
8. Halder, S.; Afsari, K.; Chiou, E.; Patrick, R.; Hamed, K.A. Construction inspection and monitoring with quadruped robots in future human–robot teaming: A preliminary study. *J. Build. Eng.* **2023**, *65*, 105814. [\[CrossRef\]](#)
9. Lee, A.J.; Song, W.; Yu, B.; Choi, D.; Tirtawardhana, C.; Myung, H. Survey of robotics technologies for civil infrastructure inspection. *J. Infrastruct. Intell. Resil.* **2023**, *2*, 100018. [\[CrossRef\]](#)
10. Surmann, H.; Daun, K.; Schnaubelt, M.; von Stryk, O.; Patchou, M.; Böcker, S.; Wietfeld, C.; Quenzel, J.; Schleich, D.; Behnke, S.; et al. Lessons from robot-assisted disaster response and recovery. *J. Field Robot.* **2023**, *41*, 782–797. [\[CrossRef\]](#)
11. Cruz Ulloa, C.; del Cerro, J.; Barrientos, A. Mixed-reality for quadruped-robotic guidance in SAR tasks. *J. Comput. Des. Eng.* **2023**, *10*, 1479–1489. [\[CrossRef\]](#)
12. Dong, C.-Z.; Catbas, F.N. A review of computer vision-based structural health monitoring at local and global levels. *Struct. Health Monit.* **2021**, *20*, 692–743. [\[CrossRef\]](#)
13. Ye, X.-W.; Jin, T.; Chen, P.-Y. Structural crack detection using deep learning-based fully convolutional networks. *Adv. Struct. Eng.* **2019**, *22*, 3412–3419. [\[CrossRef\]](#)
14. Andrushia, A.D.; Anand, N.; Neebha, T.M.; Naser, M.Z.; Lubloy, E. Autonomous detection of concrete damage under fire conditions. *Autom. Constr.* **2022**, *142*, 104364. [\[CrossRef\]](#)
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [\[CrossRef\]](#)
16. Islam, M.M.; Kim, J.-M. Vision-based autonomous crack detection of concrete structures using a convolutional neural network. *Sensors* **2019**, *19*, 4251. [\[CrossRef\]](#)
17. Kang, D.; Benipal, S.S.; Gopal, D.L.; Cha, Y.-J. Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning. *Autom. Constr.* **2020**, *118*, 103291. [\[CrossRef\]](#)
18. Sapkota, R.; Ahmed, D.; Karkee, M. Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments. *Artif. Intell. Agric.* **2024**, *13*, 84–99. [\[CrossRef\]](#)

19. Savino, P.; Tondolo, F. Civil infrastructure defect assessment using pixel-wise segmentation based on deep learning. *J. Civ. Struct. Health Monit.* **2023**, *13*, 35–48. [\[CrossRef\]](#)
20. Zarski, M.; Wójcik, M.; Książek, M.V.; Miszczak, J. Finicky transfer learning—A method of pruning convolutional neural networks for cracks classification on edge devices. *Comput.-Aided Civ. Infrastruct. Eng.* **2022**, *37*, 500–515. [\[CrossRef\]](#)
21. Spencer, B.F., Jr.; Hoskere, V.; Narazaki, Y. Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering* **2019**, *5*, 199–222. [\[CrossRef\]](#)
22. Dung, C.V.; Anh, L.D. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58. [\[CrossRef\]](#)
23. Dorafshan, S.; Thomas, R.J.; Maguire, M. SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data Brief* **2018**, *21*, 1664–1668. [\[CrossRef\]](#)
24. Cubero-Fernandez, A.; Rodriguez-Lozano, F.J.; Villatoro, R.; Olivares, J.; Palomares, J.M. Efficient pavement crack detection and classification. *EURASIP J. Image Video Process.* **2017**, *2017*, 39. [\[CrossRef\]](#)
25. Tian, Y.; Chen, C.; Sagoe-Crentsil, K.; Zhang, J.; Duan, W. Intelligent robotic systems for structural health monitoring: Applications and future trends. *Autom. Constr.* **2022**, *139*, 104273. [\[CrossRef\]](#)
26. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969. [\[CrossRef\]](#)
27. Zhang, A.; Wang, K.C.P.; Li, B.; Yang, E.; Dai, X.; Peng, Y.; Fei, Y.; Liu, Y.; Li, J.Q.; Chen, C. Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 805–819. [\[CrossRef\]](#)
28. Yang, H.; Yang, L.; Wu, T.; Meng, Z.; Huang, Y.; Wang, P.S.-P.; Li, P.; Li, X. Automatic detection of bridge surface crack using improved YOLOv5s. *Int. J. Pattern Recognit. Artif. Intell.* **2022**, *36*, 2250047. [\[CrossRef\]](#)
29. Zhu, G.; Liu, J.; Fan, Z.; Yuan, D.; Ma, P.; Wang, M.; Sheng, W.; Wang, K.C.P. A lightweight encoder–decoder network for automatic pavement crack detection. *Comput.-Aided Civ. Infrastruct. Eng.* **2024**, *39*, 1743–1765. [\[CrossRef\]](#)
30. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125. [\[CrossRef\]](#)
31. Pan, Z.; Guan, J.; Yang, X.; Fan, K.; Ong, J.C.H.; Guo, N.; Wang, X. One-stage 3D profile-based pavement crack detection and quantification. *Autom. Constr.* **2023**, *153*, 104946. [\[CrossRef\]](#)
32. Hoang, N.-D. A novel method for asphalt pavement crack classification based on image processing and machine learning. *Vis. Comput.* **2019**, *35*, 487–498. [\[CrossRef\]](#)
33. Buda, M.; Maki, A.; Mazurowski, M.A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **2018**, *106*, 249–259. [\[CrossRef\]](#)
34. Fabbri, S.; Papadopoulos, S.; Ntoutsi, E.; Kompatsiaris, I. A survey on bias in visual datasets. *Comput. Vis. Image Underst.* **2022**, *223*, 103552. [\[CrossRef\]](#)
35. Model, I.; Shamir, L. Comparison of data set bias in object recognition benchmarks. *IEEE Access* **2015**, *3*, 1953–1962. [\[CrossRef\]](#)
36. Kalfarisi, R.; Wu, Z.Y.; Soh, K. Crack detection and segmentation using deep learning with 3D reality mesh model for quantitative assessment and integrated visualization. *J. Comput. Civ. Eng.* **2020**, *34*, 04020026. [\[CrossRef\]](#)
37. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [\[CrossRef\]](#)
38. Wenkel, S.; Alhazmi, K.A.; Shoud, S.A. Confidence score: The forgotten dimension of object detection performance evaluation. *Sensors* **2021**, *21*, 4350. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Alomar, K.; Aysel, H.I.; Cai, X. Data augmentation in classification and segmentation: A survey and new strategies. *J. Imaging* **2023**, *9*, 46. [\[CrossRef\]](#)
40. Mumuni, A.; Mumuni, F. Data augmentation: A comprehensive survey of modern approaches. *Array* **2022**, *16*, 100258. [\[CrossRef\]](#)
41. Gholinavaz, S.; Saeedi, N.; Gharehveran, S.S. Robustness analysis of YOLO and Faster R-CNN for object detection in realistic weather scenarios with noise augmentation. *Sci. Rep.* **2025**, *15*, 44888. [\[CrossRef\]](#)
42. Qiu, S.; Wen, G.; Deng, Z.; Liu, J.; Fan, Y. Accurate non-maximum suppression for object detection in high-resolution remote sensing images. *Remote Sens. Lett.* **2018**, *9*, 237–246. [\[CrossRef\]](#)
43. Wang, W.; Li, X.; Lyu, X.; Zeng, T.; Chen, J.; Chen, S. Multi-Attribute NMS: An enhanced non-maximum suppression algorithm for pedestrian detection in crowded scenes. *Appl. Sci.* **2023**, *13*, 8073. [\[CrossRef\]](#)

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