

Article

Improved Dual-Module YOLOv8 Algorithm for Building Crack Detection

Xinyu Zuo ¹, Ahmed D. Almutairi ^{2,*}, Muneer K. Saeed ² and Yiqing Dai ³¹ School of Architecture and Design, China University of Mining and Technology, Xuzhou 221000, China² Department of Civil Engineering, College of Engineering, Qassim University, Buraydah 51452, Saudi Arabia; m.abdo@qu.edu.sa³ College of Civil Engineering, Fuzhou University, Fuzhou 350108, China; yiqing.dai@fzu.edu.cn

* Correspondence: ah.almutairi@qu.edu.sa

Abstract

Building cracks are significant indicators of structural integrity. Conventional fracture detection methodologies, however, are characterized by extended durations, significant labor requirements, and limitations in both precision and operational effectiveness. Findings are also subject to subjective and technical constraints inherent in manual assessments. To overcome these challenges, this paper introduces an enhanced YOLOv8-based methodology for developing a building crack detection system, thereby achieving high precision, operational efficiency, and cost-effectiveness. Initially, classified and segmented datasets of building fractures were obtained from field photography, online image aggregation, and open-source databases, thereby providing the basis for training the experimental model. Subsequently, the Swin Transformer window multi-head self-attention mechanism was implemented to augment small-object recognition capabilities and reduce computational demands, thereby enabling the development of an enhanced image segmentation module. Utilizing the U-Net's segmentation capabilities, a rotated split method was implemented to quantify fracture width and derive geometric parameters from the segmented crack regions. In order to evaluate the effectiveness of the model, two experiments were conducted: one to demonstrate the performance of the classification category and the other to show the capabilities of the segmentation category. The result is that the proposed model has high accuracy and efficiency in the crack detection task. This approach effectively enhances fracture detection in structural safety evaluations of these buildings, providing technical support for relevant management decisions.

Keywords: building safety; YOLOv8; classification; segmentation; crack detection; deep learning



Academic Editor: Hugo Rodrigues

Received: 23 December 2025

Revised: 16 January 2026

Accepted: 20 January 2026

Published: 22 January 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

From 2020 to 2023, the total construction area under construction in China will remain at around 15 billion square meters [1]. This shows that the construction industry in our country is slowly shifting its focus to the management and optimization of existing real estate projects. As a result, more and more buildings are beginning to show serious aging and damage, such as large-scale breakage and cracking, which will lead to a significant increase in the demand for inspection, monitoring and maintenance services. With the increasing density of buildings in cities, especially high-rise buildings, traditional manual inspection methods are too slow, inefficient and dangerous to evaluate those unstable

buildings. These old methods cannot keep up with the future demand for building structure evaluation. Therefore, we urgently need to develop more reliable technical solutions to find the defects on the surface of buildings.

It should be noted that surface defects in building structures are not limited to cracks. Other common defects, such as spalling, delamination, corrosion-induced staining, and material degradation, also pose potential risks to structural durability and serviceability. However, among these defects, cracks are generally regarded as the most direct and sensitive indicators of structural deterioration, as they often appear at an early stage and may progressively evolve into more severe damage. Therefore, the accurate detection and analysis of cracks play a fundamental role in structural health assessment and preventive maintenance. For this reason, the present study focuses on crack detection and segmentation as a representative and critical subtask of surface defect inspection in civil infrastructure.

With the continuous progress of modern science and technology, the processing power of computers has been greatly improved. This makes computer-driven intelligent solutions come in handy in various engineering management fields, especially in automatic detection systems. The use of computer vision and digital image processing technology is increasingly replacing traditional manual detection methods because of their strong stability, high precision and strong adaptability, which has become the main research direction and attracted many people's attention. By using digital image processing to identify, check and evaluate structural cracks, managers can obtain key crack data quickly and accurately, so that they can evaluate and analyze structural conditions more comprehensively.

In the past few years, more and more scholars have begun to identify and locate building cracks with the computer vision and deep-learning methods, which can promote a more efficient automatic crack identification system [2]. However, there is still a big gap between the theoretical model and structural defect detection in practical engineering. Therefore, how to use deep learning and computer vision to automatically discover cracks, improve efficiency and accuracy, and reduce manpower and cost has become an important research direction in modern engineering management. Successfully solving this problem will greatly improve the safety standards and quality of engineering. When cracks appear in the structure, it is necessary to accurately find out the reasons, make detailed scientific analysis, and then put forward appropriate repair schemes [3]. Its objectives and significance can be summarized as follows.

In the past, it was time-consuming and labor-intensive to check whether buildings were damaged or not. It is different now that computer vision can automatically identify damage and reduce dependence on people [4]. In recent years, more and more studies use computer vision and deep learning to identify and evaluate the cracks in buildings, which makes crack detection more accurate and reliable. Therefore, this study uses the technologies of image processing, pattern recognition and machine learning to find out the best crack segmentation method. These improvements enable us to analyze the shape and severity of cracks more carefully and reduce misjudgments. The stability and accuracy of the algorithm are improved, and the health status of buildings can be evaluated more accurately. In fact, the defects of buildings are various, with different shapes, sizes, widths and colors. Therefore, it is very necessary for us to establish a set of crack detection technology which can deal with all kinds of cracks, including those that are small, complex and subtle. In this way, the application scope of crack identification is wider and more reliable, and it can work well in different environments. Our research will also focus on optimizing the speed and efficiency of the detection algorithm, so that it can be more widely used in the real world and can identify the cracks immediately. This method can make the vision-based structural health monitoring system really "live", quickly find the places that

may become fragile in the structure, and notify the relevant personnel in time. In this way, they can take correct remedial measures quickly.

Secondly, structural cracks are considered as an important sign of building aging. It is very important for us to find and identify these cracks in time to judge whether a building is strong or not [5]. By carefully observing the characteristics of cracks, such as their location and appearance, maintenance personnel can understand the health status of the building and warn of the possible hidden dangers in advance. In this way, we can make necessary repairs and reinforcement to avoid safety problems. Moreover, the regular inspection of cracks can also improve the efficiency of building maintenance. Finding cracks quickly and classifying them correctly is particularly critical for determining where and how to repair them. Adopting orderly and efficient maintenance methods can not only prolong the service life of buildings but also save long-term maintenance costs. Finding structural defects can also provide valuable experience for our future architectural and construction work. Studying the cracks in existing buildings can help us understand the causes of cracks and find potential problems in the planning and construction stages, thus improving the overall quality and reliability of buildings. This method can reduce the occurrence of structural cracks and ultimately improve the construction quality. Simply put, the progress of building crack detection technology can not only promote innovation in related engineering fields but also provide more sophisticated technical means for technicians and managers. Accurate crack detection is helpful to evaluate the citation of building structures [6]. By observing and analyzing the location and shape of cracks, supervisors can judge the integrity and stability of the structure, warn of potential risks in advance, and take corrective measures to eliminate potential safety hazards. In addition, continuous recording of cracks can also optimize the efficiency of building maintenance and supervision. Accurate identification and analysis of cracks is very important to determine the most effective repair strategy and technology. Effective maintenance and management methods can prolong the life of buildings and reduce the cost of maintenance and renovation. Recording and evaluating structural cracks can also provide important information to help improve architectural design and construction technology. Studying the cracks in existing buildings can help us find out the root of the problem, evaluate the defects in design or construction, enhance the firmness and stability of the building, reduce the possibility of new damage, and finally, improve the overall quality of the building. In addition, advanced building crack detection technology can also promote the progress of various fields of engineering management and provide more advanced tools and methods for professionals.

From a wider perspective, it is particularly important to use computer vision technology to detect structural cracks [7]. This direction has a wide influence, which can not only help us evaluate building safety, reduce manpower input, improve fairness and accuracy of evaluation but also speeds up data processing and analysis. In this way, it can also promote the further development of intelligent building maintenance and provide many useful ideas and support for actual project management.

However, although the research on crack detection has made some progress, there are still many problems and limitations, mainly focusing on the following three key aspects:

- Firstly, the standard method of checking building cracks mainly depends on traditional measurement technology [8]. Specifically, inspectors must measure the crack width manually at regular intervals or collect data with equipment such as a crack measuring instrument. Traditional crack detection methods are mainly based on digital image processing, including edge detection, threshold segmentation and wavelet transform. However, these traditional methods have average accuracy in identifying cracks. This process is not only slow and costly but also brings great security risks. Moreover,

traditional fracture measurements usually need to compare the fracture with the scale marks on the microscope, which often leads to great errors in manual reading. Therefore, many difficulties will be encountered in practical engineering applications.

- Secondly, although single-phase, two-phase and transformer-based networks have made good progress in many fields, there is still not enough research on fine-grained semantic detection.
- Thirdly, the main focus of research has been on crack detection technology, especially in the task of image classification or segmentation, and their cooperative use is often ignored. This combined method is particularly important for engineering managers to evaluate the severity of structural damage.

Ruggieri et al. [9] proposed an improved YOLO-based framework for structural defect detection. For instance, an enhanced YOLO architecture embedding attention mechanisms was proposed to support the inspection of reinforced concrete bridges, demonstrating improved precision and recall while maintaining real-time performance on constrained devices. By leveraging expert-labeled datasets and visual explanation techniques, such approaches highlight the effectiveness of attention-enhanced YOLO models in focusing on critical defect regions and improving the reliability of automated inspection systems. Despite these advances, accurately capturing fine-grained crack features and geometric characteristics under complex backgrounds remains challenging, motivating the development of more robust dual-module detection and refinement frameworks.

This study aims to take a comprehensive look at the development of fracture detection technologies and what the difficulties are. According to these findings, we suggest using the YOLOv8 model to make the effect better and solve these main problems. The main new ideas of our work include the following.

- The purpose of this paper is to emphasize the importance of detecting structural defects in buildings and take a closer look at the recent scientific progress and real-world applications in this field.
- This research deeply discusses the core principle, network structure and working steps of deep-learning systems, and lists the advantages and disadvantages of different methods. Considering the particularity of identifying structural fractures, we have laid a solid theoretical foundation for creating high-performance crack-recognition models.
- This work combines the advantages of Transformer and CNN and analyzes their respective advantages and disadvantages. We use imaging technology to collect fracture data and study various detection methods and parameter calculation methods. By using segmentation technology, we have established an image segmentation model, which can calculate the crack width and other geometric features, which can quickly and accurately separate the shape of the crack. This method provides an important foundation for promoting crack detection, repair, evaluation and inspection after repair.

The structure of this paper is as follows: Section 2 reviews and classifies the existing literature on crack detection technology in detail and divides them into deep-learning architecture and traditional image-based methods. In Section 3, we did a lot of structural analysis with YOLOv8, improved the Swin Transformer module in the Backbone, and worked out a more efficient classification and detection model. We also optimized the U-Net architecture of Neck and designed a new segmentation model for crack detection. Section 4 describes the experimental setup, including the datasets used in two independent experiments, control experiments and ablation studies. In Section 5, we carefully analyzed the limitations of the research, discussed the possible future direction, and pointed out the key points of the follow-up work. Finally, Section 6 summarizes all the results and findings of this experiment.

2. Literature Research

By reviewing and analyzing the research of crack detection, we find that there are two main methods in this field: one is to use a traditional image processing algorithm, and the other is to use a deep-learning network to identify cracks.

2.1. Traditional Image-Based Detection Methods

There are several modern image processing methods used to identify cracks, such as threshold method, edge detection, region recognition and matching algorithm. Among all kinds of image segmentation techniques, morphological filtering is the most commonly used and effective method. Atsushilto et al. [10] used high-resolution imaging to obtain detailed internal fracture data, and then used integrated image processing technology to automatically separate fracture features and measure displacement. This paper introduces a real-time monitoring system of highway bridges with digital images. It automatically identifies the cracks on the highway by analyzing the changes in surface texture and light intensity, combining adaptive filtering and morphological mathematics. This method can ensure that the measurement is accurate to a small part of the pixel, so it is convenient for us to automatically extract features and make quantitative evaluation. Subirats et al. [11] introduced a new method, using continuous wavelet transform (CWT) to automatically identify cracks in pavement images. When dealing with irregular signals, traditional multi-resolution wavelet analysis often produces false edges, or the details are not clear enough, which makes it difficult to find out the location and direction of cracks accurately. This special method firstly carries out continuous wavelet transform on the road image, and then establishes a complex coefficient map, with the purpose of capturing and retaining only the most important coefficients. Then, this process will find the peaks in these coefficients and convert the information into a binary representation, so that the cracks on the road surface can be clearly seen. Because this algorithm does not need to rely on the existing data, it is very accurate and runs very fast. Wang et al. [12] introduced a new interactive image segmentation method. This method uses something called “multi-layer graph constraint” to add the relationship between pixels and labels to the standard integration method, which improves the accuracy and processing speed of segmentation. This study first looks at the previous research in this field, points out some shortcomings of the current segmentation algorithm in dealing with multi-scale, unstable and noisy pictures, and then combines the two basic image processing methods to provide a complete solution. The experimental results show that this method is better than other existing methods for the segmentation effect. Peng et al. [13] used Sobel and Canny filters in eight different directions to identify cracks, and achieved good results when analyzing features from different angles.

Wang et al. [14] proposed a new interactive segmentation method. This method uses multi-layer graph constraints to add the relationship between pixels and labels to the traditional ensemble model, which makes the segmentation accuracy and performance better. The experimental results show that their method is more powerful than the existing technology and can obtain more effective segmentation results. Deng [15] solves a shortcoming of traditional image segmentation techniques, that is, they cannot handle the uneven gray level. He put forward a new method, which uses Cauchy distribution to segment pictures with inconsistent gray levels. Firstly, the original image is preliminarily processed, then a Gaussian kernel is added, and then a multi-layer frame is used to establish the relationship between pixels, so that a finer weight matrix can be generated. Finally, the segmentation is completed by minimizing the objective function by gradient descent. The experimental results show that this method greatly improves the segmentation accuracy and calculation speed of those irregular gray pattern images, thus obtaining better detection results.

Noh et al. [16] uses fuzzy C-means clustering to automatically identify cracks on the concrete surface. This method selects pictures related to cracks by evaluating the probability distribution of each pixel in the cluster and how close it is to the center of the cluster. Although this technology improves the accuracy of detection, it is easy to become stuck in the local optimal solution, which affects the effect. Oliveira et al. [17] introduced several clustering methods without instruction, which can be used to find cracks. Among these methods, the clustering method based on the Gaussian mixture model is particularly useful. These researchers also invented an algorithm called CrackIT, which combines the characteristics of “no need to be taught” and “need to be taught”. Abdellatif [18] integrates CrackIT algorithm and develops a new method, which can detect cracks at both block and pixel resolutions. This method can accurately find the edge of the crack and has strong anti-noise ability. Premachandra et al. [19] has developed an automatic crack detection system, which is specially designed for a single picture and uses DA technology to separate the crack-related features from the image data.

The crack detection technology based on traditional image processing is easily disturbed by the outside world, and often produces a lot of noise, which leads to poor results. Therefore, researchers began to use deep-learning methods to detect cracks.

2.2. Deep-Learning-Based Detection Methods

From a computer vision perspective, the automation of structural inspection can be addressed through a wide range of methodologies. A recent systematic review published by Di Mucci et al. [20] has shown that CV-based approaches for civil infrastructure management generally include image-level classification, object detection, pixel-level semantic segmentation, and hybrid frameworks that combine vision-based outputs with sensor data or predictive models. In addition, CV techniques are often integrated with structural health monitoring (SHM) systems and data-driven deterioration prediction models to support long-term performance and risk assessment. In addition, CV techniques are often integrated with structural health monitoring (SHM) systems and data-driven deterioration prediction models to support long-term performance and risk assessment.

Despite this methodological diversity, CV-based crack detection and segmentation remain the most mature and widely adopted techniques for surface-level damage assessment, due to their direct interpretability, low deployment cost, and suitability for large-scale visual inspection.

Wang et al. [21] wanted to make the tool YOLOv3 more capable of identifying cracks. They combined different levels of features to make the network better at this task. In addition, they also improved the loss function to make the detection more accurate. Compared with the original YOLOv3, their improved model has made obvious progress in crack identification. Wan et al. [22] proposed a lightweight computational model based on YOLOv5s. They replaced the YOLOv5s core network with the more compact ShuffleNetv2 architecture and incorporated an ECA attention module. To acquire richer feature data, they selected the BiFPN structure as an alternative to the traditional feature pyramid. The optimized algorithm not only accelerated crack detection but also successfully reduced the model size.

Dung et al. [23] put forward a new method. They used a deep network composed entirely of convolution layers to find cracks in concrete. In another study, Li et al. [24] also developed a method to identify cracks on the concrete surface by using a deep convolution neural network based on the AlexNet model. They tested this method on a dataset of concrete cracks collected by themselves. The experimental results show that their technology can effectively find defects on the concrete surface. Yang et al. [25] introduced a method called FPHBN to identify pavement cracks. This method uses a pyramid structure

to combine a wide range of background information with very small feature details. It also uses a weighted loss function, so that the contribution of different types of samples to the total loss is more balanced and the crack identification is more accurate. Through comparative analysis, they verified that this method is really effective. Liu et al. [26] used the structure UNet to detect cracks in concrete buildings. The test results show that UNet is more reliable and accurate than other CNN-based models, and it can perform well even with little data training. Al-Huda et al. [27] proposed a hybrid deep-learning framework based on knowledge transfer, which used the internal relationship between image classification and segmentation to conduct a comprehensive semantic analysis of pavement crack images. Their method uses a special classification network to generate fracture location maps and then inputs these maps into the encoder part of a segmentation network to optimize the description of fracture boundaries. This method has been verified in practice and has been proven to produce better crack detection results.

Li et al. [28] developed a neural network structure called U-CliqueNet, which combines the methods of U-Net and CliqueNet to separate cracks from the complex tunnel image background. In another study, Wu et al. [29] added a recursive attention mechanism to the U-Net structure, which improved the accuracy of crack detection and made the cracks on the pavement and bridge more accurate. Li et al. [30] put forward the optimization strategy of the YOLOv3 model, which greatly improved its verification performance by improving network design, bounding box evaluation, data enhancement technology and loss function calculation. Zhang et al. [31] designed the MobileNet V3-Large-CBAM model, embedded the CBAM attention module into the core of MobileNet V3-Large, and then comprehensively analyzed the classification and identification of cracks with the open fracture dataset.

The biggest difference between segmentation and classification is that segmentation needs to use deconvolution to change the classified pixels back to the original size of the picture [32]. Although the research on recognizing crack with deep learning has made a lot of progress, there are still some problems. First of all, the accuracy of the current method is not good enough, which can be seen from the low scores of those scoring indicators. They can recognize the big crack, but they cannot find the thin crack, and they are easily influenced by the noise in the picture. Secondly, the generalizability of these methods is not very good.

From a methodological perspective, existing deep-learning-based crack analysis approaches can be broadly divided into two categories: object detection-based methods and pixel-level segmentation-based methods.

Object detection methods (e.g., YOLO, Faster R-CNN) aim to localize cracks using bounding boxes and are generally more efficient and suitable for real-time inspection scenarios. In contrast, segmentation-based methods (e.g., U-Net and its variants) perform pixel-wise classification and are better suited for fine-grained crack morphology analysis, such as width and boundary extraction, but usually require higher computational costs and more precise annotations.

In this study, we primarily focus on object detection to achieve fast and robust crack localization in complex real-world environments. Segmentation is not treated as an independent task but is introduced as a post-detection refinement module to extract geometric crack information after detection.

This detection-first, segmentation-assisted strategy balances real-time performance with geometric accuracy and motivates the design of the proposed dual-module YOLOv8 framework.

3. Improved YOLOv8 Network Design

When checking cracks in buildings, it is very important to accurately classify crack pictures, so that we can calculate their accurate geometric parameters. Because the types of cracks have their own characteristics, each picture must be analyzed separately to find out the relevant characteristic parameters, so as to accurately detect and distinguish crack regions. The main purpose of classification is to detect and localize crack instances in the picture and to confirm that the detected features are really cracks [33]. For ordinary cameras, it is often necessary to take many photos to obtain enough information to take a complete image of a crack, which makes automatic segmentation particularly difficult. However, in practical application, cracks on the surface of buildings usually change greatly in shape and are disturbed by a lot of noise. Therefore, it is a challenging task to use traditional manual identification, measurement or image processing methods to achieve efficient and high-precision building crack detection, and it is difficult to meet the strict standards required by actual engineering management. To resolve this issue, a novel method integrating machine vision with network pattern recognition algorithms is proposed for the automated detection of surface fractures in buildings. Conversely, trained neural network models can precisely detect fissure information in images unaffected by intricate backgrounds and noise. Given the significant advantages of convolutional neural networks (CNNs) in handling unstructured data, their application to breach detection research in building structures has emerged as a new trend. Given the exceptional ability of CNNs to extract local features, their receptive fields exhibit strong locality. As a result, dependence exclusively on CNNs hampers the ability to realize explicit global and long-distance semantic interactions effectively. Moreover, the way to obtain the network structure is opaque, which makes it difficult to adjust the parameters. This often leads to incoherent prediction results because its “perception” range is limited. In addition, it is still difficult to find those thin and long cracks in a large picture, because the complex background will interfere with it. Transformers’ ability to capture long-distance information is crucial to solve this problem. In order to solve the problem that the crack detection network only using CNN cannot “see” far naturally, we decided to adopt the Swin Transformer structure in this study. Using its larger “field of view” and the concept of moving window, the feature extraction ability of the algorithm becomes stronger. The experimental results show that the improved method is better and more stable in all kinds of situations. Moreover, by adopting the single-stage detection method of the YOLOv8 model, we not only accelerate the detection speed but also improve the recognition accuracy, thus solving the problem of “performance degradation” hidden in the traditional neural network.

3.1. YOLOv8 Network Architecture

Ultralytics introduced YOLOv8, which is a member of their YOLO model family. This new edition is based on a well-designed architecture, which contains four core parts: input module, Backbone network, Neck component and Head unit [34]. The complete structure diagram of the YOLOv8 algorithm can be seen in Figure 1.

3.1.1. Input

In order to deal with the problem of different picture sizes, the YOLOv8 algorithm will use a preprocessing method to scale the input picture at a fixed ratio, which can improve the processing efficiency. This step allows the original images to be resized according to the same proportion, leaving as few black edges as possible [35]. The default size is 640×640 pixels. This module will also automatically calculate the anchor box and enhance it with mosaic data to help users choose the enhancement method that is the most suitable for their datasets. When training, the network will first generate a prediction box with an

initial anchor box, and then compare it with the real label to calculate the error. Then, the network parameters are adjusted repeatedly by the method of back propagation to make the error smaller and smaller.

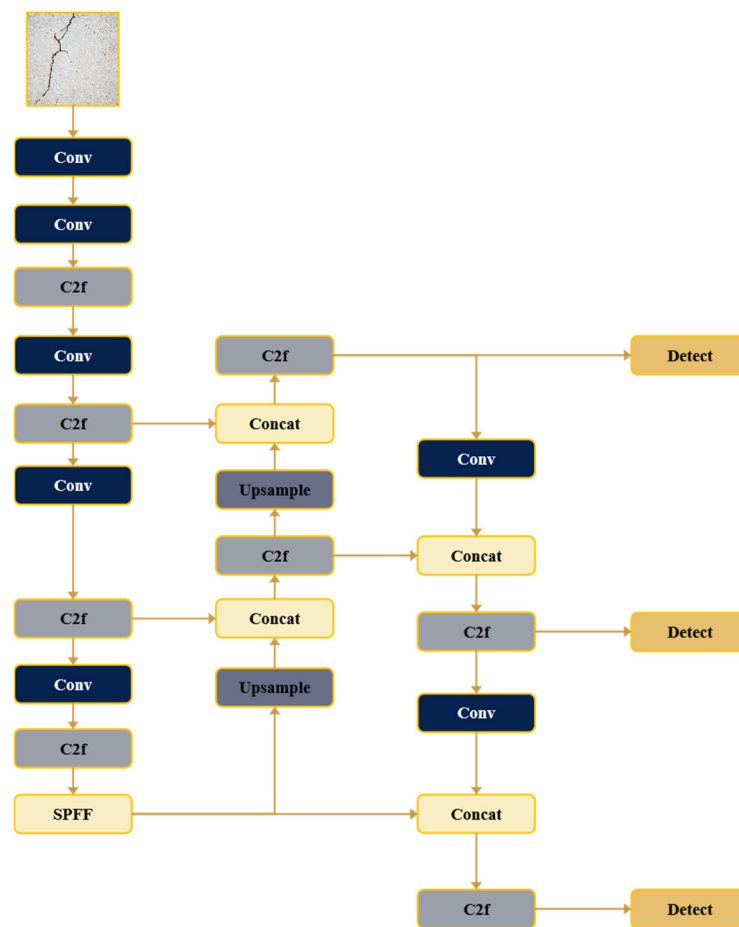


Figure 1. YOLOv8 framework diagram.

3.1.2. Backbone

In the YOLOv8 algorithm, the backbone network primarily extracts comprehensive object features. This component primarily comprises three modules: Conv, C2f, and SPPF.

1. Conv

The Conv module consists of three main parts: a 2D convolution operation (Conv2d), batch normalization (BN) and an SiU activation function. In order to deal with the edge filling of input data, the system uses an automatic filling strategy called autopad(kp).

2. C2f

The design of the C2f module is inspired by the C3 module of YOLOv5 and the ELAN module of YOLOv7. Through gradient segmentation, the developer merged the previously ordered Bottleneck components and optimized the module structure. This method can not only prevent the problem of slow convergence when the network becomes deeper but also keep the characteristics of small calculation of the model. Moreover, it also retains the complete gradient flow data, which greatly improves the performance of YOLOv8 model.

3. SPPF

Inspired by SPPNet, the SPPF module utilizes several sequential small pooling kernels. It substitutes the three convolutional kernels of varying sizes in the SPP module with

three 5×5 convolutional kernels, as two successive 5×5 kernels visually approximate a single 9×9 kernel. Similarly, three sequential 5×5 convolutions produce an equivalent effect to a single 13×13 convolution. Compared to employing large convolutions directly, cascading multiple smaller convolutions considerably decreases computational complexity and markedly enhance detection efficiency. Experimental results demonstrate that under identical input conditions, the outputs of the SPP module and SPPF module are consistent; however, the SPPF module functions at twice the speed of the SPP module. Figure 2 shows the structural settings of Conv, Bottleneck, C2f and SPPF.

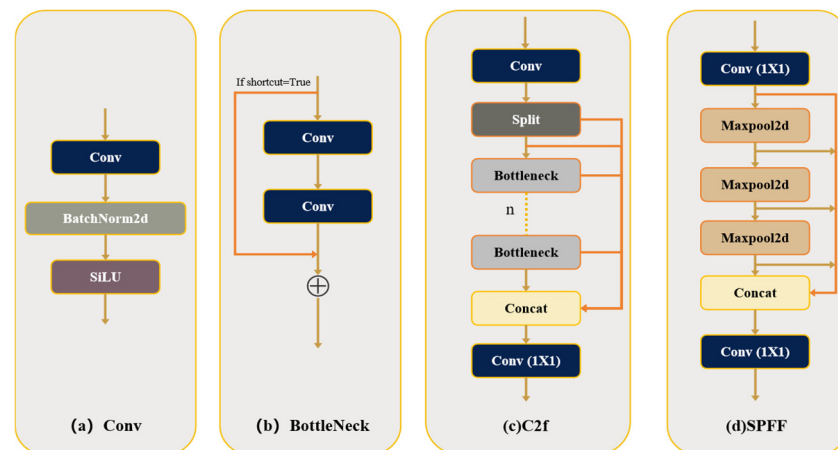


Figure 2. Structural diagram of the Conv (a), Bottleneck (b), C2f (c), SPPF (d) modules.

3.1.3. Neck

The neck design of YOLOv8 basically combines the two concepts of Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) to form a unified FPN + PAN structure. This Feature Augmentation Network is like an important bridge, connecting the backbone and prediction heads. By fusing the feature data of different scales, we can effectively detect objects. The practice of FPN is to combine the characteristics of high-resolution but weak semantic information with the characteristics of low-resolution but rich semantic information, so that it can generate feature maps that are very powerful at different scales. Different from other methods, PAN introduces adaptive feature pooling, which makes the information flow between layers of the network smoother, which enlarges the bottom details and makes the whole feature pyramid stronger. Moreover, PAN can effectively provide strong, bottom-up local features, and integrate these precise local details into the lower part of the hierarchy. This makes the path of information transmission from the lower layer to the upper layer extremely short, and also creates a flexible feature collector, which can efficiently transmit the useful data of each layer to the small network below. YOLOv8's Feature Augmentation Network not only replaces the C3 module with the C2f module but also completely avoids using convolution structure in the PAN-FPN sampling stage of YOLOv5.

3.1.4. Head

The detector of YOLOv8 algorithm adopts decoupled head design, which is a common method [36]. This system uses distributed focal length (DFL) loss to separate classification and detection tasks. The detection mechanism includes three different layers, and each layer corresponds to anchor boxes with different aspect ratios in the feature enhancement network. These layers are mainly responsible for object prediction and regression. Anchor boxes in YOLOv8 are not fixed; they will be dynamically adjusted and automatically

adapted to different datasets. In this model, the classification process uses variable Varifocal Loss (VFL), and the regression process combines complete Iouloss and Distributed Focal Loss (DFL).

1. CloU Loss: The YOLOv8 model uses CIoU instead of IoU, which can better detect the details of small objects and improve the recognition accuracy of small targets. The variable v represents the aspect ratio of the real bounding box and the predicted bounding box, respectively. The mathematical expression is this, as follows:

$$CIoU\ Loss = 1 - CIoU = 1 - \left(IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v \right) \quad (1)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (2)$$

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (3)$$

2. Varifocal Loss: Improve the detection accuracy by calculating the classification score (IACS) of IoU perception, which combines the object confidence and positioning accuracy. The VFL model will give priority to difficult positive samples, thus improving the overall object detection effect. This loss function treats positive samples and negative samples differently and will not be treated symmetrically. q here is the label: for positive samples, q represents the IoU value; for negative samples, q is set to 0. Therefore, this function uses Binary Cross-Entropy (BCE) for positive samples and Focal Loss for negative samples. The formula is as follows:

$$VFL(p, q) = \begin{cases} -q(q \log(p) + (1 - q) \log(1 - p)) & q > 0 \\ -\alpha P^\gamma \log(1 - p) & q = 0 \end{cases} \quad (4)$$

3.2. Improve the Swin Transformer Module

Transformers will encounter two main problems in natural language processing and image detection. When an object is transformed into binary representation from multiple angles, its visual characteristics will change more obviously. Vision Transformers may also perform poorly in many applications. In high-resolution pictures with many pixels, self-attention computations increase quadratically with pixel density, diminishing computational efficiency. The Swin Transformer proposes hierarchical architecture with sliding-window concepts to overcome these issues. The Swin Transformer addresses severe visual entity variations and computational complexity by combining sliding windows with a hierarchical architecture to model global relationships in computer vision tasks.

3.2.1. Swin Transformer Framework

In order to reduce the amount of calculation, Swin Transformer uses the sliding window method to limit the attention calculation to a single window. This method can also communicate between different windows, which is very important to establish a complete global representation. Its structure has four main modules: Multi-Layer Perceptron (MLP), window multi-head attention (W-MSA), shifted-window multi-head attention (SW-MSA) and Layer Normalization (LN). The schematic diagram of this model is shown in Figure 3.

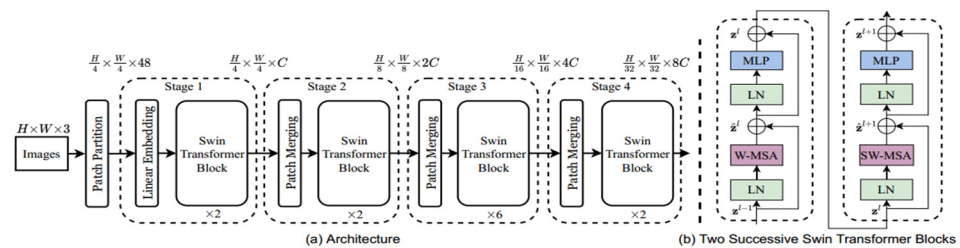


Figure 3. Architecture of Swin Transformer.

The backbone structure of Swin Transformer includes a Patch Partition module and four successive stages. It will initially process a three-channel RGB image with the size $H \times W$, and the Patch Partition module will divide the image into 4×4 non-overlapping small blocks. This step will reduce the image size and flatten the channel data, changing the input of $H \times W \times 3$ into $H/4 \times W/4 \times 48$. Then, the Linear Embedding layer will optimize these small blocks, project each channel linearly, and generate a tensor of $H/4 \times W/4 \times C$ for the Swin Transformer (ST) module. The second to fourth stages are similar, and each stage has a Patch Merging module and a set of corresponding Swin Transformer modules. Patch Merging module will merge adjacent small blocks into larger blocks, and then hand them over to the following Swin Transformer block for further processing.

3.2.2. Optimizing Self-Attention in Mobile Windows

Swin Transformer uses a new method. Instead of standard multi-head self-attention (MSA), it uses window multi-head self-attention (W-MSA) and sliding window multi-head self-attention (SW-MSA). This greatly reduces the amount of calculation. These attention-based modules are used in two consecutive ST modules. In addition, in order to train more stably, they added a normalization layer in front of MLP and each SW-MSA or W-MSA module, and also added residual connection. This particular building block can be represented by the following formula, and the variables in the formula represent the outputs of SW-MSA/W-MSA and MLP at Layer L , respectively:

$$\hat{z}^l = W - MSA\left(LN\left(z^{l-1}\right)\right) + z^{l-1} \quad (5)$$

$$z^l = MLP\left(LN\left(\hat{z}^l\right)\right) + \hat{z}^l \quad (6)$$

$$\hat{z}^{l+1} = SW - MSA\left(LN\left(z^l\right)\right) + z^l \quad (7)$$

$$z^{l+1} = MLP\left(LN\left(\hat{z}^{l+1}\right)\right) + \hat{z}^{l+1} \quad (8)$$

As illustrated in Figure 4, the working principle of the Swin Transformer attention mechanism can be explained step by step. Figure 4a shows the original crack image, which contains thin and locally distributed crack patterns. Figure 4b demonstrates the Window-based Multi-Head Self-Attention (W-MSA) mechanism. In this stage, the input image is evenly divided into multiple non-overlapping local windows. Self-attention is computed independently within each window, enabling efficient extraction of local features while significantly reducing computational complexity. However, since W-MSA limits attention to fixed windows, interactions between adjacent windows are restricted. To address this limitation, Sliding Window Multi-Head Self-Attention (SW-MSA) is introduced. Figure 4c illustrates the window partition after applying a half-window shift along both horizontal and vertical directions. The shifted image is conceptually divided into nine regions (labeled 0–8), which represent different spatial offsets resulting from the sliding operation. Figure 4d shows the re-grouping process in SW-MSA, where regions from adjacent windows are

merged to form new attention windows. This mechanism enables information exchange across neighboring windows while preserving the computational efficiency of local self-attention. By alternating W-MSA and SW-MSA layers, the Swin Transformer progressively expands its effective receptive field, allowing both fine-grained local crack details and broader contextual information to be captured. This property is particularly beneficial for detecting thin, discontinuous, and small-scale cracks in complex backgrounds.

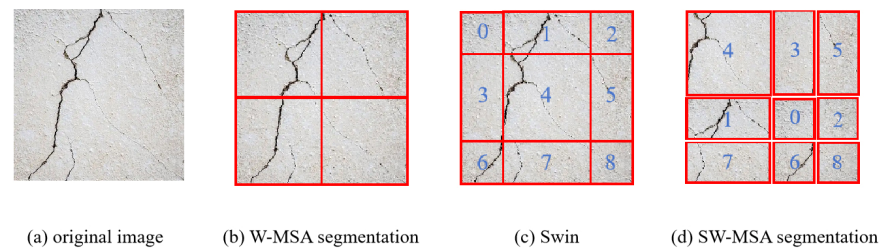


Figure 4. Mechanism diagram of the moving window.

As shown in Figure 4, the W-MSA part of the Swin Transformer module will first divide the input picture into many non-overlapping small blocks, the size of which is determined by the size of an MXM. Then, it only uses self-attention in these local windows, which can reduce the total calculation and memory occupation. However, in order to capture the global features similarly to Vision Transformer (ViT), Swin Transformer will move these windows in sequence, so that the positions of the two windows overlap. This sliding mechanism allows information to flow between adjacent small pieces, effectively strengthening the communication between different areas of the picture. Furthermore, Swin Transformer uses a multi-layer structure. By merging windows again and again and reducing their number, its “field of vision” gradually becomes larger, so that we can see the details clearly and grasp the whole. Therefore, SWMSA will move the window that was originally evenly divided by $(M/2, M/2)$ pixels in the vertical and horizontal directions, thus dividing the picture into nine different parts (see Figure 4c), which we call section 0 to section 8. In this process, section 5 and section 3 will be merged into a window, section 7 and section 1 will be merged, and then sections 8, 6, 2 and 0 will be grouped, and section 4 will be handled separately for circular shift. In this way, a new window arrangement is formed, as shown in Figure 4d. Because Swin Transformer’s attention mechanism only works in each local window, its computation will increase in proportion to the size of the input picture. At the same time, this sliding window method allows adjacent windows to “communicate” and maintains the connection between pixels, thus effectively representing the whole picture.

As explained in the previous section, YOLOv8 is faster than other models, and its accuracy is not bad, thanks to its optimized and efficient design. Swin Transformer uses the sliding window technology from Vision Transformers, combining Window Multi-head Attention (W-MSA) and Sliding Window Multi-Head Attention (SW-MSA), so that the object features can be almost completely integrated. This method focuses on the data of key objects and reduces the computational complexity from square to linear, so it is especially suitable for finding tiny cracks in low-resolution images. Therefore, this research combines the YOLOv8 framework with Swin Transformer’s sliding window attention module and develops a new algorithm, which can not only fully capture the details of cracks but also accurately locate small local semantic targets. This combination technology greatly improves the overall classification performance, accuracy and efficiency.

3.3. Improving the U-Net Model

An attention mechanism imitates the way our brains think. It will focus on the important information in feature maps and ignore the less important content. Spatial attention has been used in image classification and segmentation tasks, and the results have become better. These techniques can highlight the important spatial areas in the picture and weaken the irrelevant parts according to the current feature set, so as to improve the accuracy of prediction. Our work adds the attention module to the U-Net segmentation model, thus strengthening the structure of U-Net [37]. U-Net is a deep-learning framework specially designed for image segmentation. Its structure includes an encoder–decoder configuration with unique U-shape symmetry. The encoder uses CNNs to extract features and reduce the size of the input picture, while the decoder reconstructs these features through an upsampling operation and skip connections. This architectural design allows us to classify each pixel more accurately, so that we can segment the picture better. We used a specially designed U-Net framework in this study, which contains two “attention” organs: one is called Pixel-Level Attention (PA), and the other is called spatial multi-scale attention module (MSAM). Their appearances can be seen in Figure 5. The Pixel-level Attention component can help us see the important places in the feature map more clearly without losing the relevant information nearby. After we put PA into the encoder, it strengthens the connection between pixels in the segmented area, and makes the model find the target and enable it to be segmented more accurately. Then, before the final output results, we added a spatial multi-scale attention module (SA). The last feature map generated by the decoder will be adjusted to the most accurate size through the learned weight map and the integration of all other spatial dimensions, so that the characteristics of the target can be highlighted at the most appropriate scale. We will tell you about these two attention mechanisms in the next part.

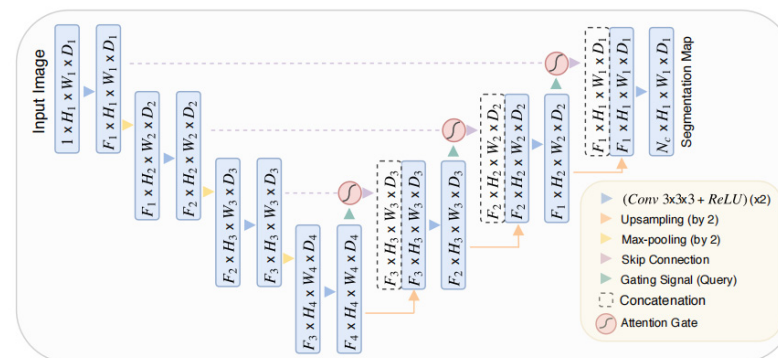


Figure 5. U-Net structure diagram.

3.3.1. Pixel-Level Attention

This PA module is designed to remember the important local features in the picture during the process of reducing it. In this way, it can better find out the most critical part of the picture. Its idea comes from the attention mechanism of CBAM. The PA module will do several operations on the input pictures at the same time: parallel convolution, global maximum pooling and global average pooling. This method can be clearly seen in Figure 6.

This study puts both addition and concatenation into the framework of CBAM. The addition process pays more attention to the “What” aspect, which can better integrate similar features, so that it is easier to analyze pictures later and uses less resources. On the contrary, the concatenation method emphasizes “Where” more, mainly to find out the specific position of key features in the picture. This method can make different features combine more naturally, while maintaining each part’s own characteristics. The concatenation function is used to enhance the addition method, which mainly focuses on “What”. Following the

integration of feature information, a two-layer MLP structure and multiplication operations are utilized to align feature sizes, as expressed in the subsequent formula:

$$y_1 = MLP(x_{cat}) \otimes x \quad (9)$$

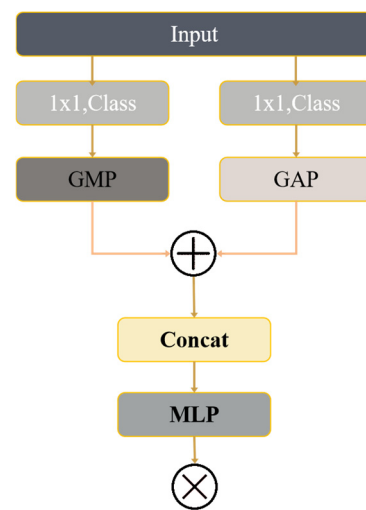


Figure 6. Pixel-level Attention Module.

3.3.2. Spatial Attention

The function of the SA module is to dynamically determine the spatial weight of each decoder stage feature map and then fuse it. In this module, we will select a feature map that is the most relevant to the current real-time picture, then adjust all other feature maps to the same size and finally combine them with the learned spatial weights.

Since the four feature map levels of the decoder possess varying resolutions and channel counts, U-Net utilizes upsampling and downsampling at three levels, whereas only downsampling is applied at the final level. For upsampling, the U-Net initially employs a 1×1 stride-convolution layer to reduce the feature map's channel dimension to match the previous level, followed by interpolation to enhance the resolution. In order to reduce the space size and the number of channels, we use a 3×3 convolution layer with a step size of 2 to downsample. The symbol f_i represents the adjusted feature map from the i -th stage, which has been modified to match the channel size and resolution of the feature map in the fourth stage. Finally, U-Net added an Atrial Spatial Feature Fusion (ASSF) component to merge the three processed feature maps with the feature maps in the fourth stage. The formula is articulated as follows:

$$y_2 = a_1 \otimes f_1 + a_2 \otimes f_2 + a_3 \otimes f_3 + a_4 \otimes F_4 \quad (10)$$

f_{1-3} corresponds to a feature map from the i layer, which has been modified to align with the number of channels and space size of the previous layer. a_{1-4} represents the spatial importance coefficients of feature maps assigned to four different network depths during adaptive training. The structure of U-Net has two limitations: $a_1 + a_2 + a_3 + a_4 = 1$ and $a_1, a_2, a_3, a_4 \in [0, 1]$. The formula is as follows:

$$a_i = \frac{e^{\lambda_i}}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4} \quad (11)$$

4. Experiment and Results

4.1. Experimental Environment and Training Configuration

All detection experiments were implemented using PyTorch 1.12.0 (Meta Platforms, Inc., Menlo Park, CA, USA) with CUDA 11.3 (NVIDIA Corporation, Santa Clara, CA, USA). The training was conducted on a workstation equipped with an NVIDIA RTX 3090 GPU (24 GB memory; NVIDIA Corporation, Santa Clara, CA, USA) and an Intel i5-13500H CPU (Intel Corporation, Santa Clara, CA, USA). The YOLOv8-based crack detection models were trained following the Ultralytics framework (Ultralytics Inc., Frederick, MD, USA). The specific hardware configuration is detailed in Table 1.

Table 1. Hardware configuration.

Hardware	Type
Graphics Card Model	RTX 3090
Graphics Memory	24 GB
CPU	i5-13500H

For crack detection, the YOLOv8n model was adopted as the baseline architecture. The optimizer was stochastic gradient descent (SGD) with a momentum of 0.937 and a weight decay of 5×10^{-4} . The initial learning rate was set to 0.0001 and scheduled using a cosine decay strategy with a warm-up phase during the first three epochs. The batch size was set to 16, and all models were trained for 200 epochs with an input image size of 640×640 pixels. Standard YOLOv8 data augmentation strategies were enabled during training, including Mosaic augmentation, random horizontal flipping, HSV color augmentation, and random scaling.

4.2. Crack Detection Experiments Based on YOLOv8

4.2.1. Evaluation Metrics for Crack Detection

The task addressed in Section 4.2 is multi-class crack object detection, rather than a simple image-level classification problem. The objective of the model is not only to determine the presence of cracks but also to localize crack regions using bounding boxes and classify them into different orientation categories, including horizontal, vertical, diagonal, and other cracks. Therefore, the evaluation metrics are defined and interpreted within the object detection framework.

To comprehensively assess the detection performance, this study adopts a combination of detection-oriented metrics and instance-level statistical indicators, including Accuracy, Precision, Recall, F1 score, Intersection over Union (IoU), Average Precision (AP), and Mean Average Precision (mAP). Among these metrics, mAP is considered the primary evaluation criterion, while the others are used as supplementary indicators to analyze detection quality and category-wise behavior.

Specifically, the confusion matrix is constructed at the detection-instance level, where a predicted bounding box is regarded as a True Positive (TP) if it matches a ground-truth box of the same crack category with an IoU greater than a predefined threshold. False Positives (FP) correspond to incorrect or unmatched detections, while False Negatives (FN) indicate missed ground-truth crack instances. True Negatives (TN) are not explicitly emphasized, as object detection focuses on positive instance matching rather than exhaustive background classification.

1. Accuracy represents the proportion of correctly detected crack instances among all detection outcomes. In the confusion matrix, True Positive (TP) denotes crack instances that are correctly detected and classified, False Positive (FP) denotes non-matching

or misclassified detections, and False Negative (FN) denotes missed crack instances. Accuracy can be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (12)$$

2. Recall measures the ability of the detection model to identify all actual crack instances. It is defined as the ratio of correctly detected crack instances to the total number of ground-truth crack instances. Its calculation formula is

$$Recall = \frac{TP}{TP + FN} \quad (13)$$

3. F1 score is the harmonic mean of Precision and Recall, providing a balanced measure of detection performance, particularly in scenarios with imbalanced crack categories. Its calculation formula is the following equation:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (14)$$

4. It is too simple to evaluate the model just by looking at the accuracy and recall in the experiment. We should use the comprehensive index AP to evaluate the detection ability of the model. MAP is the average value of these APS calculated according to the detection results of the model. Because the two indicators, accuracy and recall, will affect each other, we need to evaluate them more carefully. Therefore, mAP is an overall indicator used to comprehensively measure the performance of model detection. In this study, mAP represents the average detection accuracy of all building fractures, and the higher the value, the better the overall effect of the model. The calculation formula is

$$AP = \int_0^1 P(R)dr \quad (15)$$

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (16)$$

Although Accuracy, Precision, Recall, and F1 score provide intuitive insights into detection behavior, they are insufficient to fully characterize object detection performance. Therefore, Average Precision (AP) is employed to evaluate detection quality by integrating Precision–Recall curves for each crack category. Mean Average Precision (mAP) is then computed as the average of AP values across all crack categories, serving as an overall indicator of detection accuracy and robustness.

In this study, mAP reflects the model's comprehensive capability to localize and classify cracks of different orientations, and a higher mAP value indicates superior overall detection performance.

4.2.2. Dataset Creation

Training neural networks requires a lot of data, so it is important to prepare datasets. A comprehensive fracture dataset can make the model perform better and predict more accurately [38]. Therefore, before training the deep-learning model for detecting cracks, we must first establish a data set for crack detection.

In this paper, 9214 pictures of cracks in buildings from different environments were collected. Some of these pictures come from online and open-source datasets, while others are manually taken with drones and other shooting equipment. We randomly selected 5000 photos from these photos, created a fracture crack detection data set, and then divided

them into a training set and a verification set according to the ratio of 4:1. The training set is used to adjust the data of the model, while the verification set provides samples to evaluate the final generalization ability of the model. Both datasets contain horizontal, vertical and diagonal crack pictures, as well as various other crack types and crack pictures with complex backgrounds. These photos are shown in Figure 7.



Figure 7. Partial image.

Image preprocessing usually includes data normalization, size standardization and dataset enhancement. These steps can improve the generalization ability of the model, ensure that the input image meets the requirements of neural network, and expand the dataset to prevent over-fitting during training. By preprocessing the images in the dataset we improve the detection efficiency and model performance, optimize the use of computing resources and speed up the training process. The following are three image preprocessing operations that we have implemented on the experimental data set:

- **Dimension Standardization**
Neural network usually requires the input picture data to meet the preset size. Dimension standardization is an important step in neural network training, which aims to ensure the consistency of all inputs. This process will use scaling, cropping and other technologies to make all photos uniform in size, so as to meet the requirements of network structure and facilitate batch processing. This method can speed up the training, improve the accuracy, reduce the risk of over-fitting, and enhance the generalization ability of the model.
- **Data Normalization**
Data normalization is a very important preparation step before we train neural network. Its goal is to adjust the input data to a unified numerical range, such as 0 to 1, or -1 to 1. The advantage of this is that it can make the training more effective, and it can also make the neural network learn faster.
Firstly, standardizing the data can make the gradient descent algorithm more stable. When training the neural network, this algorithm reduces errors by adjusting the parameters of the model. However, if the range of input data is too large, the gradient may become too large or too small, so the training will be unstable.
Secondly, data standardization can help the model converge faster. In the unprocessed image data, the numerical range of different features may be much worse. This will cause some features to be updated too quickly, while others are updated too slowly.
- **Image Enhancement**
Image enhancement operation is a series of image processing methods to improve the image quality or highlight some details. Before training the neural network, these operations can increase the quality and quantity of our image dataset, so that the model can be better and more universal. Standard techniques include adjusting luminance and contrast, applying geometric transformations, reducing resolution via downsampling, and performing data augmentation. Due to the limited number of training images in this study, image augmentation techniques were employed on the collected dataset. A sequence of operations—including luminance modification, mirroring, flipping, and rotation—was applied to the dataset images, significantly

augmenting the quantity of fracture images. A schematic representation of the image enhancement process is depicted in Figure 8.

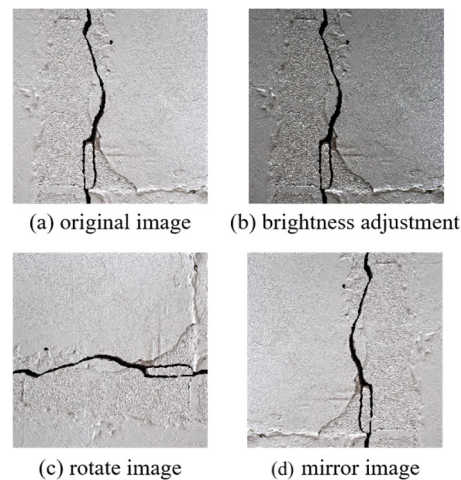


Figure 8. Image enhancement mode.

Image segmentation models perform pixel-level classification, where each pixel is assigned to a specific class. Accordingly, the crack segmentation dataset requires pixel-wise annotations indicating crack and non-crack regions. In the labeled images, crack pixels are assigned a value of 255, while background pixels are assigned a value of 0.

In this study, crack boundaries were manually annotated using the LabelMe tool, following a consistent annotation protocol. Crack pixels were defined as those enclosed by the annotated contours. To improve annotation reliability and reduce labeling noise, special attention was paid to ambiguous regions with low-contrast or complex backgrounds. Considering the limited availability of pixel-level annotations and the inherent uncertainty in crack boundaries, data augmentation techniques were applied to enhance model robustness and generalization. Moreover, segmentation is employed as a post-detection geometric refinement module rather than as an independent end-to-end task, which further reduces sensitivity to annotation noise and uncertainty.

4.2.3. Performance Evaluation and Effect Analysis

In order to verify how powerful this advanced model is, this part arranges a series of comparative studies to evaluate its ability to detect fractures from many aspects. The main contents include the following steps.

Step 1: Through comparative experiments and naked eye observation, it is proved that our improved model is better. Step 2: Find the best parameter settings of the model. Step 3: Test the detection ability of the model on a dataset built by ourselves, and prove that it has indeed become stronger, and also can perform some comparative tests. Step 4: Conduct an ablation experiment and confirm their respective contributions by removing the two modules one by one.

This series includes YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l and YOLOv8x [39], and their specific parameters can be seen in Table 2. Although they have the same basic structure, the depth and width of the networks are different. YOLOv8n is the lightest and fastest model in this series. Its depth and width are smaller, so it is easier to realize and more efficient to run. In contrast, YOLOv8s, YOLOv8m, YOLOv8l and YOLOv8x are larger models, which sacrifice some speed but improve accuracy. Although they may perform better than YOLOv8n, their larger size and higher computing requirements will make deployment more complicated. Therefore, this study chose YOLOv8n as the basic model.

The specific parameters of this basic model are detailed in the table. It should be noted that the Adam optimizer with a learning rate of 0.001 was exclusively used in the crack segmentation refinement experiment, rather than in the YOLOv8-based detection task. For the segmentation model, Adam was selected to facilitate stable convergence under limited pixel-level annotated data. The segmentation network was trained for 150 epochs with a batch size of 8, without Mosaic augmentation, while basic geometric and photometric augmentations were retained.

Table 2. YOLOv8 provides several different models.

Model	Depth_Multiple	Width_Multiple	Params (M)	FLOPs (G)	FPS, V100
YOLOv8n	0.33	0.25	3.2	8.7	120+
YOLOv8s	0.33	0.50	11.2	28.6	90+
YOLOv8m	0.67	0.75	25.9	78.9	60+
YOLOv8l	1.00	1.00	43.7	165.2	45+
YOLOv8x	1.00	1.25	68.2	257.8	30+

This paper evaluates the proposed model against several prominent algorithms outlined in the preceding section. The efficacy of this enhanced model demonstrates substantial improvements in bounding box prediction accuracy and crack detection precision. The recognition results are presented in Figure 9.

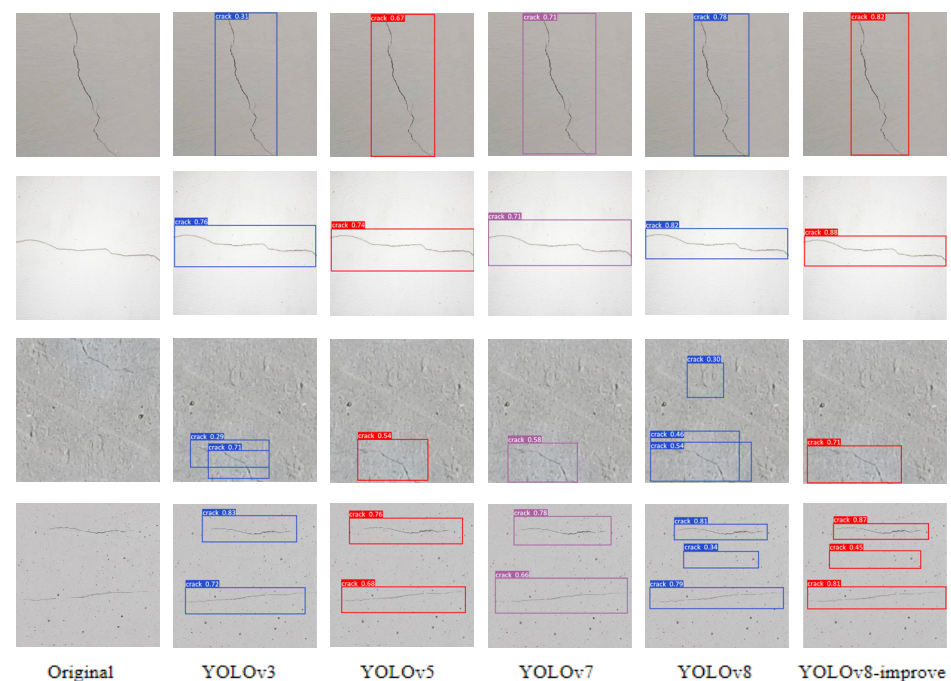


Figure 9. Experimental effect of building cracks.

4.2.4. Comparative Experiment

Table 3 compares the proposed model with representative YOLO versions (YOLOv3, YOLOv5, YOLOv7) and other mainstream architectures in terms of mAP for different crack orientations, rather than focusing solely on model size. In this study, we used the fracture crack detection dataset mentioned in this section to train our proposed model and see how powerful it is. In order to see if the improved model works well in the real world, we also compare it with several mainstream models, such as ViT, Faster R-CNN, YOLOV3 and YOLOV5, on the same dataset. On this dataset, our model performs very well, and its accuracy, recall and F1 score all exceed 90%, and the specific figures are 98.17%, 99.02% and

98.34%, respectively. Table 3 below lists the mAP values obtained by testing each model on this dataset detail.

Table 3. Comparison of detection performance across different models.

Model	mAP	Horizontal Cracks	Vertical Cracks	Diagonal Cracks	Other Cracks
Vit	90.58	91.34	90.15	89.37	89.16
Faster R-CNN	86.18	86.61	86.87	89.64	85.19
YOLOv3	85.27	86.38	86.17	85.97	83.64
YOLOv5	88.98	88.65	89.28	88.19	85.45
YOLOv7	92.17	93.48	93.81	92.37	89.77
proposed model	97.14	99.08	98.37	96.98	96.17

The model presented in this study exhibits superior recognition performance compared to other models. Further analysis indicates that this model attains mAP scores surpassing 99% for both horizontal and vertical fractures. In contrast, its mAP recognition performance for oblique and other crack types remains below optimal levels. In the test images, we seldom see such rare crack patterns as mesh fractures. Because there is too little training data on this kind of special fracture, the mean Average Precision scores of those uncommon fracture types will be lower. Our proposed model only takes about 20 milliseconds to process a picture on average. This shows that the sliding-window local attention mechanism of Swin Transformer is particularly good at recognizing the local features of tiny objects. In addition, YOLOv8 itself is very accurate, and it can fully meet the performance requirements of real-time crack analysis. Therefore, this method can quickly and accurately find and locate structural defects.

4.2.5. Ablation Experiments

In order to find out how important the two core parts, YOLOv8 and Swin Transformer, are to the model, we have performed ablation studies. In the experiment, we only remove one main part at a time to see what impact it will have. The performance data of these tests are recorded in Table 4.

Table 4. Ablation experiments.

YOLOv8	Swin Transformer	mAP
✓	✗	90.17
✗	✓	92.28
✓	✓	97.14

The test results show that adding Swin Transformer to YOLOv8 and using its sliding self-attention method can not only make the model collect more extensive context information but also optimize the processing requirements, thus greatly improving the accuracy of detection.

4.3. Improving the Segmentation Algorithm of U-Net

4.3.1. Performance Evaluation Metrics for Partitioning Algorithms

- In order to evaluate the segmentation effect, we have some standard measurement methods, such as the Dice coefficient, average symmetric surface distance (ASSD) and Intersection over Union (IoU). The Dice value is used to compare our predicted result with the correct answer, which is calculated by the following formula.

$$Dice = \frac{2|G_a \cap G_b|}{|G_a| + |G_b|} \quad (17)$$

where G_a and G_b indicate the predicted segmentation regions produced by the enhanced U-Net network and the corresponding ground truth segmentation, respectively.

- Average symmetric surface distance (ASSD). Its formula is as follows:

$$ASSD = \frac{1}{|T_a + S_b|} \times \left(\sum_{a \in T_a} d(a, S_b) + \sum_{b \in S_b} d(b, T_a) \right) \quad (18)$$

Here, T_a and S_b denote the sets of boundary points for the improved U-Net segmentation and the ground truth, respectively. $\sum_{a \in T_a} d(a, S_b)$ represents the minimum straight-line distance from point a in the set T_a to set S_b all points in set. $\sum_{b \in S_b} d(b, T_a)$ refers to the minimum linear distance from point b in set S_b to set T_a all points in the set.

- IoU
Intersection over Union (IoU) is a standard used to measure performance in computer vision and image processing. It is used to test the effects of various applications, such as object detection, image segmentation and location. IoU can calculate the degree of overlap between two regions, usually by comparing the region found by the algorithm with the real region. “Intersection” refers to the overlapping part of two areas (usually bounding box or division mask), while “union” refers to the total area of the two areas. The score range of IoU is 0 to 1, where 0 means no overlap at all and 1 means complete overlap. Generally speaking, the higher the IoU score, the more similar the two areas are, and the better the task performance is. Its calculation formula is

$$IoU(IntersectionoverUnion) = (Intersectionarea) / (Mergearea) \quad (19)$$

4.3.2. Dataset

Different from image-level classification or object detection tasks, crack segmentation requires pixel-level annotation to delineate crack regions with higher spatial precision. In crack segmentation datasets, each pixel is assigned a class label to distinguish crack and non-crack areas. Specifically, pixels corresponding to cracks are labeled with a value of 255, while background pixels are assigned a value of 0. Such pixel-wise annotations enable the extraction of geometric attributes, including crack width and local morphology, which supports subsequent crack information calculation.

In this study, pixel-level crack boundaries were annotated using the LabelMe v5.0.1 tool, following a consistent annotation protocol. Crack pixels were defined as those enclosed by manually delineated contours. The segmentation dataset consists of 1000 images, including 800 manually annotated samples and additional fully annotated images collected from publicly available datasets. Considering the limited scale of pixel-level annotations and the inherent uncertainty in crack boundaries, segmentation is employed as a post-detection geometric refinement module rather than as a standalone segmentation task. To further enhance robustness and generalization, data augmentation techniques were applied to the crack segmentation dataset, facilitating faster convergence and reducing sensitivity to annotation noise.

4.3.3. Performance Testing

The network was trained utilizing the Dice loss function, and evaluation was conducted on the highest-performing model across all validation set phases. The final evaluation was performed using 5-fold cross-validation.

The enhanced U-Net fracture segmentation model was trained on a dataset developed specifically for segmentation. The experimental results show that the Dice, ASSD and IoU values of the improved U-Net model in the crack segmentation task reached 91.95%, 0.5618 and 86.87%, respectively. These three performance indicators have achieved very good results. Therefore, the upgraded U-Net framework proposed in this study can effectively separate the cracked pixels in the image and accurately identify the cracks.

In order to compare the performance of the improved model, this study also uses the standard U-Net and Attention U-Net as evaluation benchmarks and adopts the same measurement standard for comparison. See Table 5 for comparison data.

Table 5. Comparative experiments.

Classification Network	Dice	ASSD	IoU
U-Net	86.28%	1.0814%	83.92%
Attention U-Net	89.17%	0.7698%	85.89%
Ours	91.95%	0.5618%	86.87%

This table shows that the improved U-Net model still performs well. Although Attention U-Net uses the attention mechanism, the method we study has a higher accuracy than it by adding PA and SA modules. Therefore, this crack segmentation algorithm can be drawn quickly and accurately.

4.3.4. Crack Information Calculation

In this study, segmentation is employed as a post-detection geometric refinement module rather than as a standalone crack segmentation task. Although pixel-level crack segmentation can provide fine-grained geometric details, object detection frameworks such as YOLO are more suitable for real-time inspection scenarios due to their efficiency and robustness. In this study, YOLOv8 is therefore adopted as the primary detection framework, while segmentation is introduced only as a post-detection module to refine crack geometry and extract quantitative crack parameters.

In this study, crack orientation is quantified using a geometric angle computed from the principal axis of the detected crack region.

Specifically, after crack segmentation, the skeleton or major axis of each crack is extracted, and its orientation angle θ with respect to the horizontal axis is calculated.

Based on this angle, crack orientation is discretized into engineering-oriented categories: horizontal cracks ($|\theta| \leq 15^\circ$), vertical cracks ($|\theta - 90^\circ| \leq 15^\circ$), diagonal cracks ($15^\circ < |\theta| < 75^\circ$), and other cracks for irregular or complex patterns. This quantitative criterion ensures that orientation labeling is objective and reproducible, rather than visually subjective.

The method of measuring crack information is completed by extracting crack pixels from an optimized U-Net segmentation model. In this study, crack orientation is discretized into engineering-oriented categories (horizontal, vertical, diagonal, and others) rather than modeled as a continuous angular parameter. In this paper, we introduce a new rotary cutting method, which uses the segmented pixel data to measure the width of cracks. This rotary cutting technology mainly has three key steps: image binarization, edge detection and rotary cutting to measure crack width. The specific operation steps are as follows:

- Use our proposed model to find out the crack part and obtain the segmented picture.
- De-noising the split crack image and removing the clutter but retaining the real crack pixels. Then, the Canny edge detection algorithm is used to extract the contour of the crack.

- The extracted crack width is calculated by rotating slice method. This method will traverse all pixels along the edge and divide the image into multiple slices at five intervals. Find the intersection with other edges and record the number of pixels between these points as w ($i = 1, 2, 3 \dots 30$). Take the minimum value among $w_1, w_2, w_3 \dots w_{30}$, and record it as $w_{\min}$, which represents the crack width of this position on the pixel map. By repeating these steps, the width information w of each crack pixel on the skeleton map can be obtained.

By using the enhanced U-Net model to segment the picture, and then following the above steps, we can remove the clutter, mark the edges of cracks, and measure their widths. According to the width data of each point on the crack, we can calculate the average width and maximum width of the crack.

5. Limitations and Future Work

The paper accomplished Crack images are detected by the building crack classification and segmentation methodology, enabling intelligent building health monitoring. However, personal limits, time, effort, and resource constraints limit this study's conclusions.

First, future studies may introduce new deep-learning algorithm models or computer vision methods. These changes will enhance the performance and stability of deep-learning models, thereby improving crack detection accuracy and operational efficiency. Multiple crack detection scenarios will be addressed and monitored in real time. Second, multi-modal data fusion research may continue. For more accurate fracture detection, data from pictures, LiDAR, and infrared sensors must be integrated. Research may improve real-time monitoring and prediction. Integrating BIM technology and digital twins allows real-time crack detection and monitoring, enabling timely repair procedures to maintain structural safety and reliability [40]. After identification and detection, robotics-based or autonomous crack repair systems might be researched to reduce labor expenses and enhance repair efficiency.

6. Conclusions

This paper builds a new YOLOv8 crack classification model to address challenges. This model detects building flaws quickly and accurately using modern high-performance computing technology. At the same time, this method improves detection accuracy and reduces labor and equipment costs. We use picture segmentation techniques to precisely extract fracture pixels and deliver geometric information for quantitative assessment. This can help engineering managers assess the health of buildings and make maintenance plans. The following are the main contributions.

We propose an improved YOLOv8 fracture classification network. In this study, YOLOv8 is used as the backbone network, because it is accurate, efficient and scalable, and can quickly and effectively identify crack photos. We combine Swin Transformer's window and multi-head attention method based on the shift window. After full training and verification, the model achieved 98.17% accuracy, 99.02% recall and a 98.34% F1 score on the dataset. At the same time, we have improved the U-Net model; we also measure the width of the crack by the rotating segmentation method and capture the geometric details, thus obtaining very accurate boundary information. The crack detection model based on U-Net has the following features: Dice is 91.95%, ASSD is 0.5618, and IoU is 86.87%. Moreover, the segmented results can be directly used for visual display. This method is verified by experiments, which proves that it is reliable in science and effective in practice. These results can provide more reliable data and analysis ideas for engineering professionals and help guide their strategic planning and specific work.

Author Contributions: Conceptualization, Y.D. and A.D.A.; methodology, X.Z.; software, Y.D., A.D.A., M.K.S. and X.Z.; formal analysis, X.Z. and M.K.S.; investigation, X.Z.; resources, X.Z., A.D.A., Y.D. and M.K.S.; data curation, X.Z.; writing—review and editing, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by the Deanship of Graduate Studies and Scientific Research at Qassim University, with financial support (QU-APC-2026).

Data Availability Statement: The data supporting the findings of this study are available from the first author, Xinyu Zuo, upon reasonable request.

Acknowledgments: The researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2026).

Conflicts of Interest: The authors declare that there are no conflicts of interest.

References

1. Zhang, D.; Wu, H. Smart City Construction and Urban Green Development: An Analysis from the Perspectives of Industrial Structure and Technological Progress. *Urban Plan. Constr.* **2025**, *3*, 63–78. [\[CrossRef\]](#)
2. Chen, K.; Reichard, G.; Xu, X.; Akanmu, A. Automated crack segmentation in close-range building façade inspection images using deep learning techniques. *J. Build. Eng.* **2021**, *43*, 102913. [\[CrossRef\]](#)
3. Alexander, M.; Beushausen, H. Durability, service life prediction, and modelling for reinforced concrete structures—Review and critique. *Cem. Concr. Res.* **2019**, *122*, 17–29. [\[CrossRef\]](#)
4. Wang, N.; Zhao, X.; Zhao, P.; Zhang, Y.; Zou, Z.; Ou, J. Automatic damage detection of historic masonry buildings based on mobile deep learning. *Autom. Constr.* **2019**, *103*, 53–66. [\[CrossRef\]](#)
5. Yao, Y.; Tung, S.-T.E.; Glisic, B. Crack detection and characterization techniques—An overview. *Struct. Control Health Monit.* **2014**, *21*, 1387–1413. [\[CrossRef\]](#)
6. Krishnan, S.S.R.; Karuppan, M.K.N.; Khadidos, A.O.; Khadidos, A.O.; Selvarajan, S.; Tandon, S.; Balusamy, B. Comparative analysis of deep learning models for crack detection in buildings. *Sci. Rep.* **2025**, *15*, 2125. [\[CrossRef\]](#)
7. 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. [\[CrossRef\]](#)
8. Jin, T.; Ye, X.-W.; Que, W.-M.; Wang, M.-Y. Automatic detection, localization and quantification of structural cracks combining computer vision and crowd sensing technologies. *Constr. Build. Mater.* **2025**, *476*, 141150. [\[CrossRef\]](#)
9. Ruggieri, S.; Cardellicchio, A.; Nettis, A.; Renò, V.; Uva, G. Using attention for improving defect detection in existing RC bridges. *IEEE Access* **2025**, *13*, 18994–19015. [\[CrossRef\]](#)
10. Abdel-Qader, I.; Abudayyeh, O.; Kelly Michael, E. Analysis of Edge-Detection Techniques for Crack Identification in Bridges. *J. Comput. Civ. Eng.* **2003**, *17*, 255–263. [\[CrossRef\]](#)
11. Subirats, P.; Dumoulin, J.; Legeay, V.; Barba, D. Automation of Pavement Surface Crack Detection using the Continuous Wavelet Transform. In Proceedings of the 2006 International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; pp. 3037–3040.
12. Wu, N.; Wang, Q. Experimental studies on damage detection of beam structures with wavelet transform. *Int. J. Eng. Sci.* **2011**, *49*, 253–261. [\[CrossRef\]](#)
13. Peng, B.; Jiang, Y.-s.; Pu, Y. Pavement Crack Detection Algorithm Based on Bi-Layer Connectivity Checking. *J. Highw. Transp. Res. Dev.* **2014**, *8*, 37–46. [\[CrossRef\]](#)
14. Wang, T.; Sun, Q.; Ji, Z.; Chen, Q.; Fu, P. Multi-layer graph constraints for interactive image segmentation via game theory. *Pattern Recognit.* **2016**, *55*, 28–44. [\[CrossRef\]](#)
15. Deng, H. Research on Segmentation Algorithm of Gray Inhomogeneous Image Based on Cauchy Distribution. In Proceedings of the 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT), Shenyang, China, 13–15 November 2020; pp. 287–291.
16. Noh, Y.; Koo, D.; Kang, Y.M.; Park, D.; Lee, D. Automatic crack detection on concrete images using segmentation via fuzzy C-means clustering. In Proceedings of the 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 13–17 May 2017; pp. 877–880.
17. Oliveira, H.; Correia, P.L. Automatic Road Crack Detection and Characterization. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 155–168. [\[CrossRef\]](#)
18. Abdellatif, M.; Peel, H.; Cohn, A.G.; Fuentes, R. Combining block-based and pixel-based approaches to improve crack detection and localisation. *Autom. Constr.* **2021**, *122*, 103492. [\[CrossRef\]](#)

19. Premachandra, C.; Waruna, H.; Premachandra, H.; Parape, C.D. Image Based Automatic Road Surface Crack Detection for Achieving Smooth Driving on Deformed Roads. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013; pp. 4018–4023.
20. Di Mucci, V.M.; Cardellicchio, A.; Ruggieri, S.; Nettis, A.; Renò, V.; Uva, G. Artificial intelligence in structural health management of existing bridges. *Autom. Constr.* **2024**, *167*, 105719. [\[CrossRef\]](#)
21. Wang, Q.; Mao, J.; Zhai, X.; Gui, J.; Shen, W.; Liu, Y. Improvements of YoloV3 for road damage detection. *J. Phys. Conf. Ser.* **2021**, *1903*, 012008. [\[CrossRef\]](#)
22. Zhijun, W.; Wenjing, L.; Liang, L.; Meng, Y. Low-Rate DoS Attacks, Detection, Defense, and Challenges: A Survey. *IEEE Access* **2020**, *8*, 43920–43943. [\[CrossRef\]](#)
23. Dung, C.V.; Anh, L.D. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58. [\[CrossRef\]](#)
24. Li, S.; Zhao, X. Image-based concrete crack detection using convolutional neural network and exhaustive search technique. *Adv. Civ. Eng.* **2019**, *2019*, 6520620. [\[CrossRef\]](#)
25. Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; Ling, H. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1525–1535. [\[CrossRef\]](#)
26. Liu, Z.; Cao, Y.; Wang, Y.; Wang, W. Computer vision-based concrete crack detection using U-net fully convolutional networks. *Autom. Constr.* **2019**, *104*, 129–139. [\[CrossRef\]](#)
27. Al-Huda, Z.; Peng, B.; Algburi, R.N.A.; Al-antari, M.A.; Al-Jarazi, R.; Zhai, D. A hybrid deep learning pavement crack semantic segmentation. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106142. [\[CrossRef\]](#)
28. Lin, Z.; Wang, H.; Li, S. Pavement anomaly detection based on transformer and self-supervised learning. *Autom. Constr.* **2022**, *143*, 104544. [\[CrossRef\]](#)
29. Wu, Z.; Lu, T.; Zhang, Y.; Wang, B.; Zhao, X. Crack Detecting by Recursive Attention U-Net. In Proceedings of the 2020 3rd International Conference on Robotics, Control and Automation Engineering (RCAE), Online, 5–8 November 2020; pp. 103–107.
30. Li, H.; Liu, L.; Du, J.; Jiang, F.; Guo, F.; Hu, Q.; Fan, L. An Improved YOLOv3 for Foreign Objects Detection of Transmission Lines. *IEEE Access* **2022**, *10*, 45620–45628. [\[CrossRef\]](#)
31. Zhang, X.; Fan, M.; Hou, M. Mobilenet V3-transformer, a lightweight model for image caption. *Int. J. Comput. Appl.* **2024**, *46*, 418–426. [\[CrossRef\]](#)
32. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [\[CrossRef\]](#)
33. Zakeri, H.; Nejad, F.M.; Fahimifar, A. Image Based Techniques for Crack Detection, Classification and Quantification in Asphalt Pavement: A Review. *Arch. Comput. Methods Eng.* **2017**, *24*, 935–977. [\[CrossRef\]](#)
34. Solimani, F.; Cardellicchio, A.; Dimauro, G.; Petrozza, A.; Summerer, S.; Cellini, F.; Renò, V. Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity. *Comput. Electron. Agric.* **2024**, *218*, 108728. [\[CrossRef\]](#)
35. da Silva Sobrinho, A.S.; Czeremuszkin, G.; Latrèche, M.; Wertheimer, M.R. Defect-permeation correlation for ultrathin transparent barrier coatings on polymers. *J. Vac. Sci. Technol. A* **2000**, *18*, 149–157. [\[CrossRef\]](#)
36. Hua, Z.; Aranganadin, K.; Yeh, C.C.; Hai, X.; Huang, C.Y.; Leung, T.C.; Hsu, H.Y.; Lan, Y.C.; Lin, M.C. A Benchmark Review of YOLO Algorithm Developments for Object Detection. *IEEE Access* **2025**, *13*, 123515–123545. [\[CrossRef\]](#)
37. Ye, Y.; Chen, Y.; Wang, R.; Zhu, D.; Huang, Y.; Huang, Y.; Liu, J.; Chen, Y.; Shi, J.; Ding, B.; et al. Image segmentation using improved U-Net model and convolutional block attention module based on cardiac magnetic resonance imaging. *J. Radiat. Res. Appl. Sci.* **2024**, *17*, 100816. [\[CrossRef\]](#)
38. Feng, S.; Zhou, H.; Dong, H. Using deep neural network with small dataset to predict material defects. *Mater. Des.* **2019**, *162*, 300–310. [\[CrossRef\]](#)
39. Bakirci, M. Advanced aerial monitoring and vehicle classification for intelligent transportation systems with YOLOv8 variants. *J. Netw. Comput. Appl.* **2025**, *237*, 104134. [\[CrossRef\]](#)
40. Ai, L.; Ziehl, P. Advances in Digital Twin Technology in Industry: A Review of Applications, Challenges, and Standardization. *J. Intell. Constr.* **2025**, *3*, 1–19. [\[CrossRef\]](#)

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.