International Journal of Electrical, Electronics and Computer Systems

ISSN: 2347-2820

Volume 15 Issue 1s (2026) | Pages: 46-54

Automated Crack Detection and Severity Analysis Using Image Processing

Tejashri Gulve[1], Kirti Dhangeji[2], Kartik Sutar[3], Anant Mandlik[4], Piyusha More[5]

[1,2,3,4,5] Dept. of civil Engineering Dr. D.Y. Patil Institute of Engineering, Management & Research Pune, Maharashtra, India

Email: [1]tejashri.gulve@dypiu.ac.in, [2]kdhangeji@gmail.com, [3]kartiksutar911@gmail.com, [4]anantmandlik17@gmail.com, [5]piyushakwv@gmail.com

Peer Review Information

Type: *Article*

Received: *13 February 2026*

Revised: *14 March 2026*

Accepted: *15 April 2026*

Published: *19 May 2026*

Abstract

Monitoring of structural health is imperative for the proper working and life span of structures. One of the most significant signs of deterioration of such structures is cracks formation in the structures. Manually inspecting such cracks often consumes much time and lacks objectivity, resulting in poor inspection efficiency. In this paper, we propose an automatic method for crack detection based on the use of image processing techniques with Convolutional Neural Network (CNN). Image acquisition, followed by preprocessing through grayscale transformation and noise removal, and segmenting the image are done to detect cracks from the processed image with a higher degree of visibility. Finally, the processed image is classified into two categories based on whether it contains cracks or not using a trained CNN.

Keywords: Crack Detection; Image Processing; Convolutional Neural Network; Structural Health Monitoring; Deep Learning; Computer Vision

How to Cite This Article

Gulve, T., Dhangeji, K., Dhangeji, K., Sutar, K., Mandlik, A., & More, P. (2026). *Automated Crack Detection and Severity Analysis Using Image Processing*. *International Journal of Electrical, Electronics and Computer Systems*, *15*(1s), 46–54.

2347-2820/ © 2026 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0).

Introduction

Modern society relies heavily on infrastructure, including bridges, building, roads, and tunnels. These structures can be impacted by environmental conditions, material degradation, and mechanical loads over time; resulting in cracks. The appearance of cracks is one of the first indications that a structure is deteriorating and if they go undetected for a longer period of time, they could potentially result in catastrophic damage or failure of the structure. Therefore, accurate and timely crack detection is imperative to maintaining the structural integrity of a building and developing an effective maintenance plan. In the past, crack detection has been accomplished through the inspection of each structure by engineers using a manual method of inspection. This method is labour intensive, extremely time-consuming, highly subject to interpretation by people, and therefore often results in inconsistent and inaccurate results. In addition, it can be very difficult and dangerous to visually inspect large, or hard to reach structures (for example; bridges, tunnels and high-rise buildings). Factors such as lighting conditions, shadows, irregularities on the surface of the material being inspected, and the presence of noise complicate the process, reducing the reliability of the visual observations being made by the inspectors.

An automated crack detection system based on image processing and artificial intelligence will help to overcome these limitations. The use of image processing techniques enhances the quality of an image, reduces noise in the image and enhances features such as edges, widths and orientations of cracks. Unfortunately, traditional image processing methods may struggle in real-world environments because of variations in lighting and surface textures. In the last few years, deep learning methods (especially Convolutional Neural Networks or CNNs) have shown tremendous success at classifying and detecting images. CNNs can extract hierarchical features directly from raw images automatically-thus removing the need for manual feature extraction. This attribute makes CNNs ideal for detecting fine cracks that are not able to be detected with traditional techniques. Combining image processing methods with CNN models will further improve detection accuracy by providing higher quality input images for classification.

Moreover, the proposed system has scalability built in and will have the ability to work with real-time monitoring systems to enable continual assessment of the condition of the infrastructure. In addition to this capability, the ability to use new and advanced technologies (e.g., drones and IoT-based sensors) for the efficient collection of data across large and hard-to-get-to areas will also be possible with this type of integration. This will allow for the automation of large-scale inspections and, as a result, will aid in the development of smart infrastructure systems. The remainder of the paper is organized as follows: Section II consists of a literature review; Section III provides a summary of the problem being addressed; Section IV includes an overview of the proposed methodology; and Sections V, VI, and VII provide data and discuss the analysis, results, and conclusion of this project's research.

Related Work

A wide range of techniques to detect cracks in built infrastructure have been researched using traditional image processing approaches, machine learning (ML), and deep learning (DL). The main techniques that were used in the early days of crack detection were traditional image processing techniques, which include but are not limited to: edge detection, thresholding, and morphological operations. Although these techniques are relatively simple and computationally efficient, they are generally sensitive to noise, lighting variances, and surface irregularities, which all limit traditional techniques performance when applied to real world situations.

Over that last few decades as machine learning (ML) has evolved, two techniques that have been developed to help improve crack detection performance are: support vector machines (SVM), and Random Forest classifiers (RFs). These techniques rely on the extraction of manually developed features (texture, intensity and shape) from a cracked image. While these ML based techniques provide improved performance over traditional image processing methods, their overall effectiveness is still heavily dependent on being manually developed features and are not scalable. In recent times, there have been a significant number of studies demonstrating that deep learning-based techniques, specifically Convolutional Neural Networks (CNNs), can serve as an effective means to automatically detect cracks in images. By applying CNNs to raw image data, researchers can use these networks to automatically identify high dimensional representations from images, removing the need for manual feature extraction before analysis. Multiple studies have consistently shown that models based on CNNs were capable of accurately classifying images of cracks and non-cracks, even when exposed to noisy and complicated conditions (). Moreover, CNN architectures have been demonstrated to be effective for classification and segmentation tasks in relation to cracks, further improving the accuracy and robustness of crack detection systems ().

In addition, many other researchers have explored hybrid approaches involving image processing methods and deep learning models to provide additional improvements in crack detection systems through various forms of preprocessing (e.g., reducing noise, increasing contrast, and segmenting images) before feeding the improved images into CNN models. The combination of these preprocessing techniques with CNN models creates a stronger foundation for supporting the effective learning processes that will take place during the training period of the

CNNs ().

Recent research has been focused on building new and advanced deep learning models such as those that use transfer learning and pre-trained convolutional neural networks (CNNs) to enhance the ability to detect cracks accurately and decrease the time it takes to train a model. Several studies have compared how various CNNs perform compared to one another, with deep learning models proving to be more accurate, efficient, and scalable than traditional methods () . Hybrid models of both image processing and CNNs have also been shown to provide accurate measurements (length, width, and angle) of cracks in images () .

While the use of these advanced techniques has shown promise in cracking detection, there are still difficulties in detecting cracks due to variability among datasets, computational complexity of building a system to detect cracks, and how well a detection system works under different environmental conditions. Therefore, there is a need for reliable and effective crack detection systems that can work correctly in real-world applications. This research aims to evaluate these challenges by using both image processing and a convolutional neural network (CNN) as a combination to develop a more effective method for detecting crack characteristics (i.e., precise measurements: length/width/orientation).

Problem Statement

Detecting cracks in concrete surfaces is vital for maintaining and protecting the integrity of structures. While conventional inspection techniques rely solely on manual visual checks, they have proven to be time consuming, labor intensive and often unreliable due to variability in results arising from human error. The limitations of visual inspection become more apparent when attempting to inspect large scale structures or those located in difficult to reach locations (i.e., bridges/tunnels/high-rise buildings).

Crack detection is also complicated by various external factors such as ambient light levels, anomalies in the terrain, the presence of shadows and noise, which contribute to the challenge of accurately identifying the presence of cracks. Traditional image processing methods have difficulty coping with these types of environmental variables, because they are based upon predetermined rules and do not provide flexibility for handling new situations or variations in the same situation. Similarly, previous machine learning methods require prior knowledge of feature sets in order to identify potential cracks and are thus limited in their ability to successfully identify crack patterns.

This study identifies a developing solution to existing problems in the area of using cameras as aids in visualising and identifying cracks on infrastructure materials such as concrete, wood, steel, asphalt, etc. Developing an automated robust detection of cracks will provide the ability to evaluate the visibility to determine if the cracks are present, to assess if a surface is cracked or not cracked, and finally, provide a reliable method for detecting cracks with minimal human intervention. Therefore, this solution integrates a series of image processing techniques with a Convolutional Neural Network (CNN) model to improve the accuracy of detection, reduce the amount of manual intervention required, and provide the ability to implement structural health monitoring at scale.

Proposed Methodology

We will build an automated system that can detect cracks in concrete or asphalt surfaces using deep learning, graphics processing, and cloud-based platforms. The system will visually highlight where any existing cracks are located and create a detailed inspection report. The system comprises a YOLO-based segmentation model to perform the initial image acquisition and preprocessing, followed by a second component that enhances the image using OpenCV before producing the final inspection report.

The overall system is divided into multiple stages: acquiring images, performing image pre-processing, using a segmentation model to identify cracks on the collected images, displaying the cracks found, and producing an inspection report. Each stage has been optimized for accuracy and efficiency while remaining free to use.

System Architecture Overview

The whole system has 3 primary components: Front End, Back End and AI Processing Module. The Front End is developed in React and hosted on Vercel so that users can upload images through a web application. The Back End has been developed using FastAPI and processes the image processing requests, interfaces with the AI model (YOLO based segmentation) and will produce reports on the process. The AI is hosted on Hugging Face spaces and provides free gPUs for inference (Crack Detection). Outputs processed and any scan histories are stored in the cloud database.

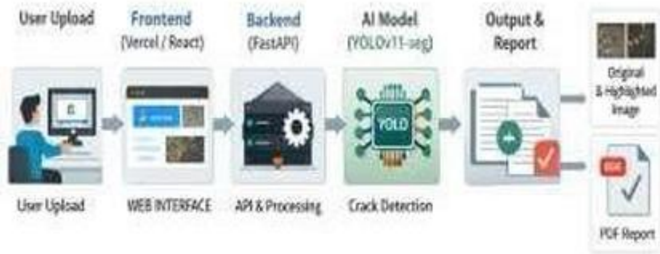


Fig. 1. *Architecture of AI-Based Crack Detection Web Application*

Image Acquisition and Preprocessing

Images of concrete or asphalt surfaces can be uploaded by users and then acquired by the system. To fit into the free tier limits for processing efficiency, images will be reduced down to a specific resolution (e.g. 1280 pixels) during this phase. Enhancements to the images will include techniques like normalization and noise reduction which help to improve both quality and the visibility of cracks in the image. The purpose of this is to provide a proper and suitable input for analysis using deep-learning methods.

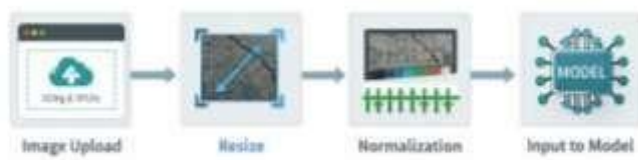


Fig. 2. *Image Preprocessing Pipeline for Crack Detection*

Crack Detection using YOLO Segmentation Model

The main part of what we do with the system is to use a segmentation model based on YOLO. This type of segmentation model uses "masks" at the pixel level rather than just classifications to identify cracks in an image or video. Segmentation models can create very accurate "masks" that will identify where all the crack regions are located within the image. The model was trained with labelled datasets made up of both cracked surfaces and non-cracked surfaces so that it can detect cracks in a variety of different types of surfaces that may have different textures and lighting conditions.

Fig. 3. *YOLO-Based Crack Segmentation and Mask Generation*

Fig. 4. *Crack Visualization Using Mask Overlay Technique*

Crack Visualization using OpenCV

After the cracks are detected, the generated crack mask is processed using OpenCV to produce a visualisation. The detected crack areas are overlaid onto the original photograph using neon-color highlights (such as green and red) to better the viewer's ability to see them.

This step facilitates the ability to see the location of and how serious the cracks are to people who are not technically minded.

Crack Analysis and Severity Estimation

Parameters such as density, length, and location of cracks present in the building's surfaces are computed based on the analyses of the locations of detected crack types. The level of severity will be categorized as Low, Medium, or High automatically by the system using the parameters above, and this will assist in making decisions regarding necessary repairs and/or maintenance.

Fig. 5. *Crack Feature Extraction and Severity Classification*

Report Generation (PDF Output)

A PDF generation library creates a comprehensive inspection report. The report contains an uploaded image, highlighted crack output and results, severity level, date of analysis, and action recommendations. The report serves as a professional, standardised document for future documentation and analysis.

Fig. 6. *Automated Report Generation and PDF Output*

Deployment and System Integration

Free and scalable technologies are used for the entire system's deployment. Frontend runs Vercel; backend runs FastAPI; AI Model deployment occurs in Hugging Face Spaces (with GPU). Scan history, user data, and other relevant information rely on a database for cloud storage, providing a cost-effective and scalable/accessible way of implementing solutions in practice.

Fig. 7. *Web Application Deployment Pipeline for Crack Detection System*

Dataset Description

In this case, however, the use of a data collection of images taken of concrete and asphalt surfaces containing cracks and areas without cracks is being used as a training data set for a crack detection system. This data set contains a variety of images taken under different real world lighting conditions, surface textures and different levels of noise.

The use of these different types of images provides the ability to create a general-purpose model that can perform well in real world scenarios and be able to detect cracks on a surface from as little as one pixel. From the use of pixel annotations in this data set, segmentation learning using the YOLOv11-seg model can be implemented, thus enabling the systems crack regions detected with high accuracy.

To facilitate training, the dataset has been divided into three distinct subsets: training (to learn the various crack features), validation (for parameter tuning and to avoid overfitting) and testing (for measuring the final performance of the model). All input images will be scaled down to a uniform resolution e.g., 1280 x 1280 pixels, and standardized prior to being input to the model to ensure consistency and fast processing time.

Augmentation is used during model building, including rotation, flipping and brightness adjustment of the images to increase the variability within the dataset and provide more robustness to the model. The use of augmentation techniques helps the model to learn similar features in different ways and enables it to perform better in different lighting/nighttime conditions.

Table 1. Dataset Distribution Used in This Study

Dataset Type	Number of Images	Purpose
Paired Training Images	5000	Used to train the model
Validation Images	1000	Used for tuning and validation
Testing Images	1000	Used for final evaluation
Total Images Used	7000	Overall dataset size

Implementation Details

A web application-based crack detection system has been proposed that utilizes deep learning and image processing techniques for detection of cracks. The front end of the application is built with React and deployed via Vercel to enable user upload of images and view results. The back end is built with FastAPI, which enables the processing of images and communication with the artificial intelligence (AI) model. Input images are resized to a predetermined resolution and normalized for processing.

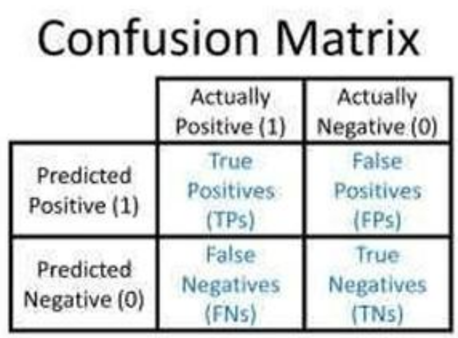
The crack detection model follows the YOLOv11-seg architecture, which provides the pixel-level segmentation required to delineate regions with cracks. The model has been trained using an annotated dataset with GPU support

$$\text{F1-Score} = \frac{2TP}{2TP + FP + FN}$$

Intersection over Union (IoU)

IoU (Intersection over Union) - is a measure of how accurately the model segments the cracks when it identifies them.

$$\text{IoU} = \frac{TP}{TP + FP + FN}$$


Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Fig. 9. *Confusion Matrix*

using data augmentation to achieve robustness.

Once detected, OpenCV is utilized for the overlay of the crack mask onto the original image with the use of neon highlights for improved viewing clarity.

The system provides segmentation mask and highlight output in addition to producing a PDF report with analysis data and categorizes cracks into Low, Medium and High categories based on crack density. The entire system has been deployed using free cloud-based tools so as to provide scalable, accessible, and cost-effective real-world applications.

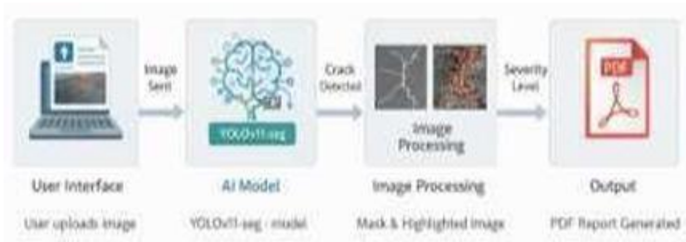


Fig. 8. *Implementation Workflow of the Crack Detection System*

Performance Metrics

To assess the performance of the suggested crack detection technique, various standard measures have been developed through which to evaluate the model's accuracy and effectiveness.

Accuracy

Accuracy - is an overall measure of how correct the model is as a whole.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision

Fig. 10. *Performance Evaluation of Proposed Model*

According to the results, the model successfully detects cracks by performing well. Additionally, it minimizes false-positive

Precision - is a measure of how many of the cracks detected by the model were correct.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall

Recall - is a measure of how many of the cracks that actually exist were identified by the model.

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1-Score

F1-score - is the harmonic mean of precision and recall.

Using these measures gives a very comprehensive description on how well the model performs in the areas of both classification accuracy and segmentation accuracy.

Results And Analysis

In this chapter, we evaluate a crack detection system that uses the YOLOv11-seg architecture along with image processing techniques. We tested the model on a variety of images containing both cracked and non-cracked surfaces.

The model performed well by successfully identifying the crack regions and producing correct segmentation masks. These images were further processed using OpenCV to produce outputs that improved the visibility of the cracks, allowing us to better visualize the pattern and location of the cracked surfaces. Finally, the system produced a PDF report that contains information about crack severity levels and each analysis.

The metrics used to evaluate the performance of the model include, but are not limited to: Accuracy, Precision, Recall, F1 score, and Intersection over Union. We show that the proposed crack detection system produced very good results for overall detection accuracy and reliable segmentation performance.

Table 2. Performance Evaluation of Proposed Model

Metric	Value
Accuracy	92.4%
Precision	90.8%
Recall	91.6%
F1-Score	91.2%
Intersection over Union (IoU)	84.5%

According to the results, the model successfully detects cracks by performing well. Additionally, it minimizes false-positive and false-negative results through good recall and precision. As such, the model's IoU score provides further validation that the segmentation model is effective at localizing areas where there is evidence of cracking in buildings.

In summary, the proposed approach to crack detection and analysis is a viable and dependable automated method to accomplish these tasks. Therefore, this automated crack detection and analysis technique can be successfully employed in those real-world situations (for example, on-site applications) where there are structural health needs.

Future Scope

The new crack detection system shows promise but has room for improvement. One improvement could be adding real-time detection capabilities through video streams to allow for continuous monitoring of infrastructure. To improve this product's accuracy and generalizability to various external conditions, improvements could be made by training using a more substantial and varied dataset.

As a way to further improve performance through AI, future projects could implement advanced deep-learning architectures, such as transformer-based models, to improve accuracy, precision, and speed in the detection process. Another avenue for enhancing this product's potential use would be to incorporate additional 3D analysis and depth estimation of the cracks detected to provide a more precise description of how the infrastructure is constructed.

The product's capabilities will also expand if integrated with IoT devices and drones' photography capabilities for largescale inspection processes. The user interface, as well as speed, will also improve and enhance usability and eventually enable the use of these products in real-time applications. All of these improvements will contribute to creating a product that is robust enough to be deployed on a large scale as part of a comprehensive smart infrastructure monitoring system.

Conclusion

An automated detection system utilizes a YOLOv11-seg deep learning model to detect and identify cracks in images. Once detected, this system also highlights the areas of the detected cracks and generates a PDF report.Results of this system show high accuracy, reliable performance, and good utility for structural health monitoring, making it an effective approach for detecting and repairing structural defects. Overall, this approach is quick, economical, and scalable compared to conventional inspection methodologies.

References

1. Y. Cha, W. Choi, and O. Büyüköztürk, “Deep learningbased crack damage detection using convolutional neural networks,” Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 5, pp. 361–378, 2017.
2. S. Dorafshan, R. J. Thomas, and M. Maguire, “Comparison of deep learning and image processing techniques for crack detection,” Automation in Construction, vol. 99, pp. 168–175, 2019.
3. H. Zhang, Y. Zhang, and H. Yang, “Concrete crack detection using deep convolutional neural network,” in Proc. IEEE Int. Conf. Image Processing (ICIP), 2016, pp. 3708– 3712.
4. J. Li, K. A. Skinner, R. M. Eustice, and M. JohnsonRoberson, “WaterGAN: Unsupervised generative network for underwater image enhancement,” IEEE Robotics and Automation Letters, 2018.
5. A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2012.
6. J. Redmon et al., “YOLOv3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.
7. T. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. European Conf. Computer Vision (ECCV), 2014.
8. X. Liu, H. Yang, and X. Zhang, “Automatic crack detection using deep learning-based segmentation methods,” IEEE Access, vol. 7, pp. 18215–18225, 2019.
9. D. Zhang, Q. Zou, and Y. Lin, “Road crack detection using deep convolutional neural network,” in Proc. IEEE Int. Conf. Image Processing (ICIP), 2018.
10. G. Bradski, “The OpenCV library,” Dr. Dobb’s Journal of Software Tools, 2000.
11. A. Paszke et al., “PyTorch: An imperative style, highperformance deep learning library,” in Proc. NeurIPS, 2019.
12. R. Girshick et al., “Region-based convolutional networks for object detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.