AI-Enhanced Building Safety: Real-Time Crack Detection with YOLO

Mrs.R.Aruna [1], M.Umar Farook [2], P.Vignesh [3], S.Santhosh Kumar [4]

[1] Assistant Professor, Department of IT , Kongunadu College of Engineering and Technology
[2] Student, Department of IT , Kongunadu College of Engineering and Technology
[3] Student, Department of IT , Kongunadu College of Engineering and Technology
[4] Student, Department of IT , Kongunadu College of Engineering and Technology

---

Abstract - - In an increasing demand for safe and more reliable infrastructure, early detection of structural defects in buildings is now considered to prevent a catastrophic failure. This paper discusses an AI-enhanced real-time crack detection system using the YOLO algorithm, one of the deep learning algorithms which can detect an object from the given images or video feeds in real time. The proposed system makes use of YOLO's object detection capability to detect and classify cracks in images or video feeds. This system aims to support the efficient and proactive monitoring of structure health by building managers and engineers. Delivering fast and accurate crack detection, this AI solution surpasses the traditional approaches that would normally raise the costs of maintenance and minimize the overall safety level of a building. Implementation of this methodology would help result in safer buildings and cost-efficient building management.

***Key Words*:**AI-enhanced crack detection ,YOLO algorithm Real-time monitoring ,Structural health, Building safety.

1. INTRODUCTION

Building safety is one of the significant concerns, especially in aged structures and areas prone to extreme weather or seismic activity. Traditional monitoring methods such as manual inspections are time-consuming and full of mistakes. This paper presents an AI-powered real-time crack detection system using the YOLO deep learning model. The developed system automatically detects and classifies various types of cracks which will provide building managers with aproactive solution towards issues that may arise before they deteriorate, therefore reducing repair costs and safety risks. Real-time alerts are integrated in the system to alert stakeholders by email, SMS, or application when there is a crack found. Cost-effectiveness and scalability are offered with the help of open-source tools like OpenCV for image processing and TensorFlow or PyTorch for running YOLO. This combines object detection, cloud storage, and automated reporting to monitor the site continuously and making buildings safer overall. When a crack is found, the system immediately notifies stakeholders by email, SMS, or app, allowing for timely action. It guarantees affordability and scalability by utilizing open-source tools such as TensorFlow/PyTorch for YOLO and OpenCV for image processing. It improves overall building safety by offering continuous monitoring through the integration of object detection, cloud storage, and automated reporting.

2. METHODOLOGY

Safety in buildings is essential, particularly in older buildings that are subject to harsh weather conditions. Manual inspections are laborious and prone to mistakes. With the help of YOLO, this study presents an AI-powered crack detection system that provides real-time alerts via email, SMS, or apps for prompt action. YOLO improves maintenance and lowers repair costs by achieving over 90% accuracy in real-time crack detection.

2.1 Related Works

[1] An AI driven crack identification technique for stone masonry that operates without labeled data was presented by Agrafiotis et al. For historic structures where manual checks are difficult, this is ideal.
[2] Golding et al. achieved great accuracy in identifying concrete cracks by using deep learning. Their CNN-based model performs better than conventional techniques, resulting in quicker and more accurate inspections.
[3] To detect pavement cracks, Hu and Zhao used Local Binary Patterns (LBP). Their method is excellent at managing various surface conditions and sunlight, opening the door for contemporary AI crack detection.
[4] To improve building façade defect identification, Interlando et al.'s study makes use of several deep learning models. Their method works very well for monitoring cities on a broad scale, which helps to make infrastructure management safer.
[5] Medina et al. presented a hybrid approach for detecting road surface cracks that incorporates 2D and 3D images. This integration improves accuracy and drastically lowers false positives, making it extremely dependable for practical uses.
[6] For accurate pavement distress identification, Mathavan et al. examined 3D imaging methods as laser scanning and stereo vision. These methods are popular for monitoring infrastructure because of their great accuracy.
[7] To effectively detect road cracks, Oliveira and Correia created an automated method that combines edge recognition and morphological picture processing. Their method offers a scalable and efficient way to evaluate transportation infrastructure.
[8] In order to improve crack detection accuracy while preserving computing efficiency,Ren et al. integrated attention mechanisms into deep learning models. Their research yields encouraging outcomes for practical implementations.
[9] Sarhadi et al. used an enhanced SWIN U-Net model with

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 02 | Feb - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

attention mechanisms to optimize the identification of concrete cracks. Because of its great accuracy and dependability, this approach is appropriate for monitoring concrete infrastructure on a broad scale.

[10] For fracture identification, Shu et al. presented an active learning technique using a difficulty learning mechanism. By giving complicated samples priority, our method lowers annotation costs while enhancing model performance overall.

[11] A deep learning model created especially for identifying cracks in marble surfaces was created by Vrochidou et al. Supporting robotic applications for automated resin application in the marble industry is the goal of their work.

[12] Varadharajan et al. talked about computer vision methods for road inspection that are powered by AI. Their model effectively detects pavement cracks and other road issues by using image processing.

[13] A deep residual U-Net model for pavement fracture identification was presented by Yang et al. To improve segmentation accuracy, their method makes use of deep feature extraction and skip connections.

[14] CNN-based deep learning models for concrete surface crack detection were compared by Zadeh et al. With its great accuracy in detecting structural flaws, their research demonstrates AI's expanding potential in infrastructure monitoring

2.2. System Architecture

The software system automatically detects and analyzes building flaws by combining cloud, AI, and computer vision technologies. Building managers can promptly resolve problems, stop additional damage, and lower maintenance

.

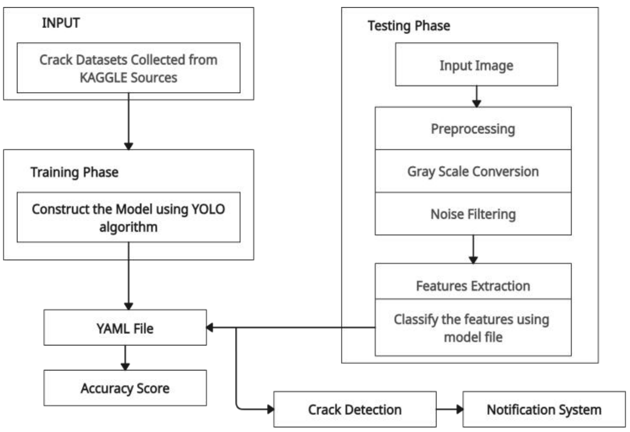

Fig 2.2.1.System Architecture

A. Dataset Collection

Gathering a collection of crack photos is the initial stage in the training phase. These datasets, which include tagged photos of cracks and non-cracks, are taken from Kaggle or other repositories. To increase the resilience of the model, the dataset should be varied, including pictures with various surfaces, angles, and lighting conditions. The dataset is divided into training and validation sets, with the model being trained on the training set and its performance assessed on the validation set. To increase dataset variability, data augmentation methods such as flipping, rotation, and contrast modifications can also be used.

B. Model Construction Using YOLO Algorithm

The entire image is processed in a single pass by the realtime object detection model known as the YOLO (You Only Look Once) method. YOLO creates bounding boxes, separates an image into grid cells, then categorizes items inside the boxes.YOLO is trained to identify crack patterns and differentiate them from non-crack areas in order to detect cracks. The number of epochs, batch size, and learning rate are among the parameters that are set up for the model. Convolutional layers are used to refine the network architecture in order to efficiently extract spatial data, guaranteeing excellent detection accuracy while preserving processing speed in real time.

C. YAML File Generation

Following training, the YOLO model creates a YAML (Yet Another Markup Language) file that serves as a structured configuration file with all the necessary settings for deployment and testing. In order for the trained model to be effectively reused without retraining and be able to categorize fresh images with high accuracy, this file is essential. By organizing important model metadata, it facilitates the trained model's integration into other machine learning frameworks and practical applications. The dataset configuration, which specifies paths for training, validation, and test datasets, is the first of several essential elements included in the YAML file. The number of object classes (nc) and their labels—typically ['crack'] in the context of crack detection—are also specified. It also includes anchor box definitions, which are bounding boxes that have been designed to help detect cracks of various sizes and forms, hence increasing the accuracy of detection. Hyperparameter storage, which includes parameters like learning rate, batch size, and momentum that affect the model's stability and performance, is another crucial area of the YAML file. The YAML file bridges the gap between training and practical application by standardizing these configurations, guaranteeing consistency, portability, and smooth deployment of the trained model in diverse situations.

D. Accuracy Score Evaluation

Accuracy measures are used to assess the model's performance after training. Predicted labels and ground truth labels are compared to calculate the accuracy score. Model performance is measured using metrics like Intersection over Union (IoU), F1-score, precision, and recall. When a model has a high accuracy score, it can identify cracks with few false positives and false negatives.In order to increase performance, hyperparameters are adjusted and more training data is added if the accuracy score is not adequate. During the testing phase, the accuracy score aids in determining whether the model is prepared for deployment.

E. Input Image

A fresh image is supplied as input for crack identification during the testing phase. This picture might have been captured by a camera or an industrial inspection tool. The image is used as the raw data, which will go through a number of preprocessing stages before the trained model analyzes it. The accuracy of detection is greatly influenced by the quality of the input image. Clear contrast and high definition photos enhance the model's capacity to detect cracks. The final detection findings could be impacted by any distortion, blurriness, or noise in the input image.

F. Prepossessing

The image is preprocessed to improve features and eliminate extraneous parts before the model processes it. To increase detection accuracy, preprocessing entails scaling the image to fit the input dimensions of the model and normalizing pixel values. Additional methods like sharpening filters and contrast augmentation can also be used. By following these procedures, you may be sure that the image is in the best possible format for feature extraction. Better detection results are achieved by minimizing the effects of illumination, texture, and background noise fluctuations while assisting the model in concentrating on crack-related variables through appropriate preprocessing.

G. Gray Scale Conversion

A crucial preprocessing step that converts a color image to a black-and-white (grayscale) format is grayscale conversion. This preserves crucial fracture structural information while lowering computational complexity. The contrast between crack regions and non-crack surfaces is improved by converting the image to grayscale because cracks are usually depicted by changes in pixel intensity rather than color. This improves the model's ability to identify minute details in the picture. Grayscale conversion makes sure that the texture and pattern of the cracks are the main emphasis by removing extraneous color information.

H. Noise Filtering ,Feature Extraction and Classification Using Model File

Noise Filtering: Uses bilateral, median, or Gaussian filtering to eliminate dust, shadows, and reflections. This increases the accuracy of fracture detection, decreases false positives, and improves image clarity .Key crack patterns, such as edges, lines, and textures, are identified by feature extraction. In order to distinguish genuine cracks from dirt, scratches, or surface flaws, the YOLO model examines various areas.Classification: Detected patterns are given a likelihood score by the trained YOLO model. The mechanism verifies the existence of a crack if the score rises above a predetermined level. Model weights that are stored (from the YAML file) provide precise and effective classification of fresh photos.

I. Crack Detection Output and Notification System

The system logs the results for analysis after detecting a fracture and marks it with bounding boxes. It is used to automate inspections and reduces damage in buildings, bridges, and roadways. The notification system provides location, severity, and crack details to maintenance staff in real time, via software, emails, or SMS. This makes it possible to act quickly, avoiding expensive repairs and guaranteeing safety.

2.3. Workflow

A. Image Input

An input image is first sent to the system for analysis. Usually, cameras or drones are used to take this picture in order to inspect the infrastructure. It ensures that the detection process begins with appropriate and lucid visual data by acting as the basis for further processing

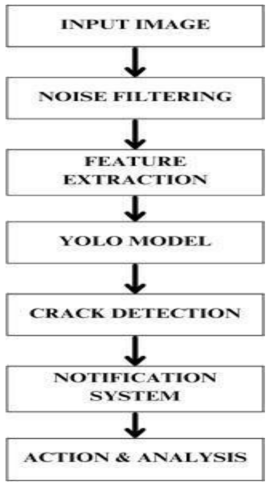

Fig 2.3.1: Architecture diagram for Workflow

B. Filtering out noise

Dust, shadows, and reflections are examples of undesired components that can obstruct detection. By eliminating these distortions while maintaining crucial fracture information, noise filtering techniques like Gaussian and median filtering improve the image. Better accuracy in subsequent processing is ensured by this phase.

C. Extraction of Features

Important visual elements are taken out of the picture, including textures, edges, and lines. This procedure aids in differentiating fractures from comparable formations such as surface imperfections or scratches. The features that have been retrieved are used as inputs for the classification model.

D. The YOLO Model

By examining the extracted features, the YOLO (You Only Look Once) model ascertains whether a crack is present. Using trained data, it rapidly processes the image in real-time, identifying areas as crack or non-crack.

E. Crack Identification

The technique uses bounding boxes to highlight any cracks that are found. Maintenance personnel can find the problem more easily thanks to this visual marker.

F. Notification System

When a crack is found, the notification system notifies users, guaranteeing prompt action. Depending on the integration, notifications may be provided by text message, email, or software alerts. These notifications assist maintenance personnel prioritize important cases by providing crucial information like the location, severity, and kind of the fracture .The solution improves efficiency, decreases human monitoring efforts, and stops infrastructure faults before they worsen by providing real time notifications.

G. Analysis & Action

Maintenance personnel examine the detection data and take corrective action as the last step. This contributes to the safety and durability of infrastructure by preventing structural failures.

2.4. Experimental Analysis

Fig.2.4.1 Input Preprocessing

Through step-by-step image analysis in above Figure 2.4.1,this AI-based method finds building fractures. Before enhancing contrast to highlight cracks, the image is first converted to grayscale to eliminate color distractions. It then uses binary conversion, which isolates the cracks by making them look as black lines on a white background, and segmentation. AI evaluates these fissures to determine the extent of structural damage, allowing for prompt repair and early identification. By preventing malfunctions and ensuring prompt repairs through real-time notifications, this increases building safety and makes structures stronger and more resilient.

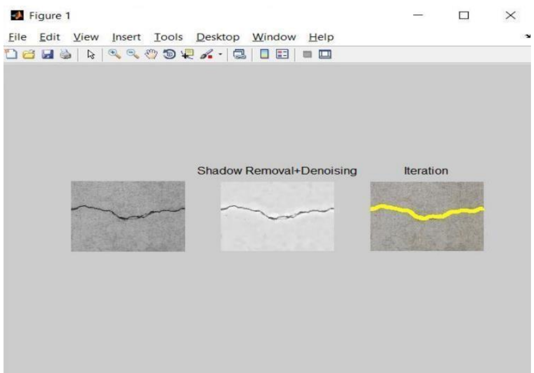

Fig.2.4.2 Shadow Removal and Denoising Iteration

The above Figure 2.4.2 demonstrates the methodical steps involved in AI-based crack detection for building security. Although the first image shows the original grayscale surface, it is difficult to see the crack because of noise and shadows. The crack is accentuated and made more noticeable in the second image by denoising and shadow removal. In the final image, the AI precisely detects the crack by highlighting it in yellow. This procedure facilitates real-time monitoring, permits prompt repairs, and averts significant structural breakdowns.

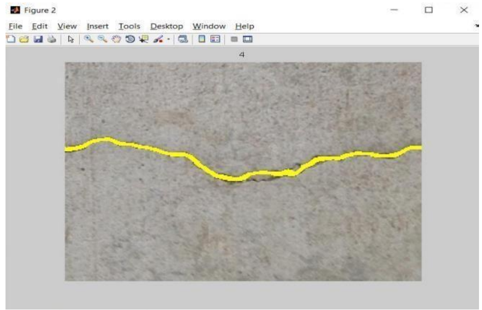

Fig.2.4.3 Crack Detection

The last phase of AI fracture identification on a concrete surface is depicted in the above Figure 2.4.3. When the crack is indicated in yellow, the AI has successfully located it. The procedure entails eliminating shadows, cutting down on noise, and clearly identifying the crack. This aids in prompt repairs, halting additional harm and maintaining the safety of buildings.

```
Converged at 500
Iteration = 0
Iteration = 25
Iteration = 50
Iteration = 75
Iteration = 100
Iteration = 125
Iteration = 150
Iteration = 175
Iteration = 200
Iteration = 225
Iteration = 250
Iteration = 275
Iteration = 300
Iteration = 325
Iteration = 350
Iteration = 375
Particular #    Mean     Intensity     C.Area     Diameter
# 1             0.6      56136.0       1334.3     267.3
```

Fig.2.4.4 Experimental Results

The above Figure 2.4.4 result is demonstrates how an AI system gradually identifies cracks. The crack is successfully detected after 500 attempts. The system monitors development and gets better with time. Crack details such as confidence (0.6), intensity (56136.0), area (1334.3), and diameter (267.3) are measured in the final result. This aids in preventative maintenance to ensure the safety of buildings.

3. CONCLUSIONS

For detecting structural fractures, the YOLO-based AI crack detection system offers a quick, dependable, and real time solution. With little human labor, it guarantees great accuracy through the use of deep learning and computer vision. YOLO is scalable and effective when used with Python, PyTorch, and OpenCV. This lowers inspection costs and permits early maintenance to stop significant damage. To ensure prompt response, the system has an enhanced alert mechanism that sends administrators emails, SMS, and messages when cracks are found. Engineers may make well-informed judgments with data logging and real-time monitoring. A completely automated infrastructure monitoring system may result from upcoming improvements like cloud storage, predictive analytics, and IoT integration. By making building maintenance more proactive, secure, and economical, this innovation ensures long-term durability with little work.

ACKNOWLEDGEMENT

I sincerely thank my mentor, research team, and technical experts for their invaluable guidance, feedback, and dedication throughout this project. Special appreciation goes to the developers of the YOLO algorithm and AI technologies. I also acknowledge my family's support in this endeavor.

REFERENCES

[1] Agrafiotis, P., Doulamis, A., & Georgopoulos, A. "Unsupervised Crack Detection on Complex Stone Masonry Surfaces." arXiv preprint arXiv:2303.17989, 2023.

[2] Golding, Vaughn Peter, et al. "Crack Detection in Concrete Structures Using Deep Learning." Sustainability 14.13 (2022): 8117.

[3] Hu, Y., and C. Zhao. "A Local Binary Pattern Based Methods for Pavement Crack Detection." Journal of Pattern Recognition Research 5.1 (2010): 140–147.

[4] Interlando, M., Pacifico, M. G., Novellino, A., & Pastore, V. P. "Ensembles of Deep Neural Networks for the Automatic Detection of Building Facade Defects from Images." IEEE Access, 2024.

[5] Medina, R., J. Llamas, E. Zalama, and J. Gomez Garcia Bermejo. "Enhanced Automatic Detection of Road Surface Cracks by Combining 2D/3D Image Processing Techniques." In Proceedings of IEEE International Conference on Image Processing, 2014, pp. 778–782.

[6] Mathavan, S., K. Kamal, and M. Rahman. "A Review of Three-Dimensional Imaging Technologies for Pavement Distress Detection and Measurements." IEEE Transactions on Intelligent Transportation Systems 16.5 (2015).

[7] Oliveira, H., and P. L. Correia. "Automatic Road Crack Detection and Characterization." IEEE Transactions on Intelligent Transportation Systems 14.1 (2013): 155–168.

[8] Ren, Junhua, et al. "Automatic Pavement Crack Detection Fusing Attention Mechanism." Electronics 11.21 (2022): 3622.

[9] Sarhadi, A., Ravanshadnia, M., Monirabbasi, A., & Ghanbari, M. "Optimizing Concrete Crack Detection: An Attention-Based SWIN U-Net Approach." IEEE Access, 2024.

[10] Shu, Jiangpeng, et al. "An Active Learning Method with Difficulty Learning Mechanism for Crack Detection." Smart Structures and Systems 29.1 (2022): 195-206.

[11] Vrochidou, Eleni, et al. "Towards Robotic Marble Resin Application: Crack Detection on Marble Using Deep Learning." Electronics 11.20 (2022): 3289.

[12] Varadharajan, S., S. Jose, K. Sharma, L. Wander, and C. Mertz. "Vision for Road Inspection." In Proceedings of 2014 IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 115–122.

[13] Yang, Yalong, et al. "Research on Pavement Crack Detection Algorithm Based on Deep Residual Unet Neural Network." Journal of Physics: Conference Series 2278.1 (2022). IOP Publishing.

[14] Zadeh, S. S., Birgani, S. A., Khorshidi, M., & Kooban, F. "Concrete Surface Crack Detection with Convolutionalbased Deep Learning Models." arXiv preprint arXiv:2401.07124, 2024.

[15] **R. Sathya,** V. C. Bharathi, S. Ananthi, K. Vaidehi and S. Sangeetha, "Intelligent Home Surveillance System using Convolution Neural Network Algorithms," 2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2023, pp. 1683-1688, doi: 10.1109/ICESC57686.2023.10193402.

[16] **R. Aruna**, M. Prabu, S. Ananthi, V. C. Bharathi, R. Sathya and B. Suchithra, "Vision based Cassava Plant Leaf Disease Classification using Machine Learning Techniques," *2023 2nd* International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 959-964, doi: 10.1109/ICACRS58579.2023.10404468.

[17] R. Sathya, M. Mythili, S. Ananthi, R. Asitha, V. N. Vardhini and M. Shivaani, "Intelligent Video Surveillance System for Real Time Effective Human Action Recognition using Deep Learning Techniques," *2023 2*nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp.18261831,doi:10.1109/ICACRS58579.2023.10404670.