ConcreteGuard: A YOLOv8-based Web Application for Early Detection of Concrete Cracks

Mary Josephine R. Gamit
De La Salle University - Dasmariñas
Dasmariñas, Cavite
gmr0078@dlsud.edu.ph

Miguel Antonio D. Chavez
De La Salle University - Dasmariñas
Dasmariñas, Cavite
cmd0935@dlsud.edu.ph

John Carlo M. Gayoso
De La Salle University - Dasmariñas
Dasmariñas, Cavite
gjm0934@dlsud.edu.ph

Sheryl D. Kamantigue
De La Salle University - Dasmariñas
Dasmariñas, Cavite
sdkamantigue@dlsud.edu.ph

ABSTRACT

Concrete, a widely used building material, plays a crucial role in global construction. Detecting and repairing concrete damage is vital to maintain structural integrity and safety. Current detection methods involve manual inspection and non-destructive testing, which can be time-consuming and labor-intensive. To address these challenges, this project introduces a web application utilizing Computer Vision and Machine Learning to aid civil engineers in concrete crack detection and mapping.

ConcreteGuard employs the YOLOv8 model, trained on a diverse dataset with data augmentation. All built-in hyperparameters provided by YOLOv8 were used in their default setting. Evaluation metrics demonstrate a precision of 0.882, recall of 0.772, mAP50 of 0.868, and an F1 score of 0.823. OpenCV is used for contour detection and distance transform for crack width measurement, achieving an average difference of 0.765 mm. Challenges persist for cracks less than 2 millimeters wide.

Additionally, the system facilitates crack mapping in construction, offering a systematic approach to identify, locate, and document cracks in structures. In situations where rapid preliminary assessments are crucial, such as seismic-prone regions, ConcreteGuard's crack mapping functionality can prove indispensable for professionals in civil engineering and construction.

KEYWORDS

Crack mapping, YOLOv8, web application

1 INTRODUCTION

Concrete is a fundamental material in the construction industry due to its versatility and widespread use. Comprising cement, water, and aggregates like sand, gravel, and crushed stone, it forms the backbone of many structures. The global cement market was valued at USD 326.81 billion in 2021, with projections indicating growth to USD 481.73 billion by 2029 [10]. Despite a temporary decline of 3.6% in 2020 due to the COVID-19 pandemic, the market is rebounding, fueled by increased demand for residential and public infrastructure projects, including hospitals and healthcare centers.

However, despite its durability, concrete is susceptible to deterioration from various factors such as temperature changes, moisture, chemical reactions, and structural [1]. Detecting and addressing these issues promptly is crucial for maintaining structural integrity and ensuring the safety of building occupants.

Traditionally, professionals rely on manual visual inspections and non-destructive testing methods like ultrasound and ground-penetrating radar to identify concrete damage [11]. One visual concrete inspection method used is crack mapping. It allows engineers to identify the extent and severity of cracks within a structure [3]. By systematically documenting the location and size of cracks, engineers can assess the structural health and prioritize repairs accordingly. This information is crucial for preventive maintenance and ensuring the long-term durability of concrete infrastructure. While effective, this method is labor-intensive, time-consuming, and requires specialized equipment and expertise, adding complexity and cost to the inspection process.

In recent years, computer vision and Convolutional Neural Networks (CNNs) have emerged as promising tools for concrete crack detection. Unlike traditional methods, CNNs can automatically extract features from images without manual intervention, leading to more accurate and efficient detection [13].

Additionally, OpenCV, short for Open-Source Computer Vision Library, is a free and powerful software library for computer vision tasks and is used in this study [8].

Past research has demonstrated the efficacy of employing computer vision and deep learning techniques, such as convolutional neural networks (CNN), for concrete crack detection. Despite their success, these methods often grapple with small datasets, limiting their generalizability. This study builds upon this foundation by leveraging CNNs to develop a concrete crack detection web application, with an edge lying in its ability to generate comprehensive crack mapping reports. While previous approaches have achieved accuracies ranging from 84% to significant improvements in crack quantification, the focus of this study extends to practical application through an accessible digital platform, promising a new dimension in crack detection methodology [13, 2].

This study aims to develop "ConcreteGuard," a web application designed to enhance the accessibility and user-friendliness of concrete crack detection and documentation. ConcreteGuard facilitates the detection, classification, and measurement of cracks in concrete structures, allowing users to capture images and process them using the CNN model. Additionally, the application generates detailed reports to aid in comprehensive evaluation.

To achieve these objectives, the study utilized a dataset of images from Roboflow, a framework for creating computer vision models without hand-labeling images, supplemented by images captured by researchers from four public schools in Dasmarinas [7]. The dataset was used for training using the YOLOv8 model for crack detection and measurement, which was integrated into the web-based application for easy access and report generation.

However, it's important to note the study's limitations, including the exclusion of crack length measurement, internal flaw detection, and exploration of mobile applications. Furthermore, the application does not incorporate safety measures, and users are advised to adhere to safety guidelines during inspections and repairs. Moreover, YOLOv8 was chosen as the model due to its exceptional detection speed and its suitability for training with limited hardware resources [7]. This choice was reinforced by the procurement of 100 compute units from Google Colab, enabling uninterrupted training sessions despite the absence of GPU or similar high-performance hardware.

Despite these limitations, the study has the potential to significantly improve efficiency and accuracy in concrete structure inspection, thereby enhancing public safety and infrastructure longevity.

2 METHODOLOGY

2.1 Data Collection

The objective of this method is to compile a diverse dataset of concrete images, including examples of concrete cracks with various patterns. A total of 10,711 images were collected, with 711 sourced from four specific public schools in Dasmariñas, Cavite, and an additional 10,000 obtained from various public datasets within Roboflow. The primary dataset captures a broad spectrum of real life environmental and structural conditions, acquired using a Digital Single-Lens Reflex (DSLR) camera. Researchers took photographs from various angles and distances to ensure comprehensive coverage. These primary images are then uploaded to the project repository on Roboflow. Conversely, the secondary dataset was acquired by duplicating public dataset repositories into the project repository.

2.2 Data Processing

The primary dataset underwent annotation by researchers using Roboflow's annotation tool to segment concrete cracks. In contrast, the secondary dataset was pre-annotated, yet inaccuracies persist, prompting researchers to manually edit annotations as well. The images were resized to 640x640 and had their contrast adjusted using histogram equalization.

Moreover, data augmentation techniques were implemented to boost diversity and avoid underfitting. Calibration for varying lighting conditions was also conducted to maintain dataset consistency. Each training image generates three outputs through augmentation methods such as flipping, rotating by 90 degrees, shearing, adjusting saturation, brightness, exposure, applying blur, noise, and creating mosaics. An example of an original image is demonstrated by Figure 1, and its augmentations on Figure 2.

Figure 1: Original image of concrete crack.

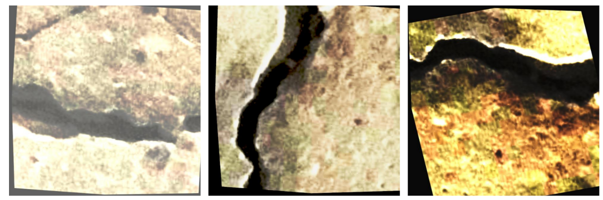

Figure 2: Augmented images of concrete crack.

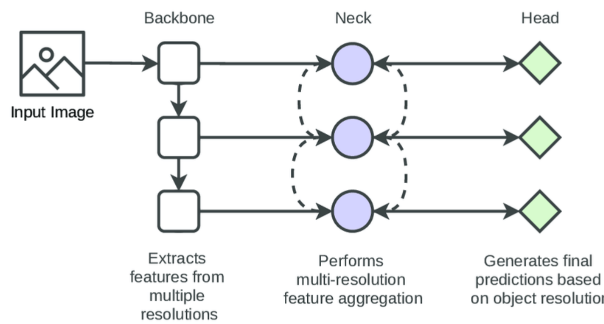

Figure 3: The YOLOv8 simplified model architecture.

2.3 Data Processing

The primary dataset underwent annotation by researchers using Roboflow's annotation tool to segment concrete cracks. In contrast, the secondary dataset was pre-annotated, yet inaccuracies persist, prompting researchers to manually edit annotations as well. The images were resized to 640x640 and had their contrast adjusted using histogram equalization.

Moreover, data augmentation techniques were implemented to boost diversity and avoid underfitting. Calibration for varying lighting conditions was also conducted to maintain dataset consistency. Each training image generates three outputs through augmentation methods such as flipping, rotating by 90 degrees, shearing, adjusting saturation, brightness, exposure, applying blur, noise, and creating mosaics. An example of an original image is demonstrated by Figure 1, and its augmentations on Figure 2.

2.4 Language, Model, Hosting Platforms

The web application was built using Gradio, a Python library that streamlines user interaction with machine learning models through intuitive web interfaces [14]. It simplifies the process of inputting data and presenting results, ensuring accessibility for users of varying technical backgrounds. On the other hand, Hugging Face serves as a central hub for the machine learning community, offering pre-trained models and hosting capabilities through "Spaces." These Spaces enable effortless deployment of the model, allowing users to interact with it via a web browser without the need for software installation [15].

Furthermore, YOLOv8 is a cutting-edge object detection and segmentation algorithm that utilizes a single convolutional neural network (CNN) to simultaneously predict bounding boxes, class probabilities, and pixel-level masks for multiple objects in an image. It is known for its fast inference speed, high accuracy, and ability to segment objects at the pixel level, making it suitable for real-time applications [12].

As seen on Figure 3, YOLOv8 breaks down an image into its key features using a pre-trained CNN backbone. It then combines these features from different resolutions to create a comprehensive understanding of the image. Finally, a specialized head predicts bounding boxes and class probabilities for objects directly, without needing pre-defined boxes, making it faster and more efficient.

2.5 Crack Width Measurement

$$W_r = \alpha W_p D C \tag{1}$$

The crack width measurement utilizes the OpenCV library, which is capable of measuring objects in images by employing contours and distance transform [9]. Contour detection serves the purpose of identifying shapes within the image, while the distance transform computes the distance of each pixel to the nearest edge. By measuring the distance and angle of the object relative to the camera lens, the real-world dimensions can be inferred from the number of pixels with the following equation, where W_r is width in millimeters, α is the angle relative to the lens, W_p is width in number of pixels across, D is distance to the lens, and C is the calibration factor. The calibration factor is the average ratio of widths measured by calipers and by the application over varying crack widths taken from 15 centimeters.

3 PRELIMINARY RESULTS

3.1 Model Performance

This study employed a YOLOv8 neural network to detect cracks in concrete images. The model's performance was evaluated on the combined primary and secondary dataset comprising 21259 training images, 2136 validation images, and 1688 test images.

Table 1: Model Training Experiments

Test	Images	Augmented	Epoch	Precision	Recall	mAP
1	10000	23376	50/50	0.875	0.775	0.879
2	711	170	35/50	0.591	0.354	0.389
3	200000	46752	22/50	0.782	0.691	0.718
4	10711	16710	45/100	0.862	0.752	0.827
5	10711	25083	52/150	0.882	0.772	0.868

Table 1 shows different experiments involved training the YOLOv8 model with varying amounts of images, augmentation, epochs, and achieved different precision, recall, and mAP scores for concrete crack segmentation. Experiment 1 had the highest mAP but it only consisted of secondary images. Experiment 5 had the second highest mAP, while also including the primary images gathered by the researchers, indicating better overall performance. The model produced was used for the web application. The metrics are as follows: Precision – 0.882; Recall – 0.772; mAP50 – 0.868; F1 Score – 0.8233.

Precision measures the accuracy of the positive predictions made by the model, whereas recall assesses the model's ability to identify all actual positive cases. mAP, or mean Average Precision, is a metric used to assess the accuracy of object detectors like YOLOv8 across different classes and threshold levels. It calculates the average precision for each class and then averages these scores, offering a comprehensive measure of model performance. The F1 score, which is 0.8233 combines both precision and recall into a single metric. It is particularly useful as it balances the trade-offs between precision and recall, providing a single score to measure the overall efficacy of the model at a set confidence threshold, in this case, at 0.459.

The results highlight the extensive training process of the YOLOv8 model, incorporating crucial aspects such as batch size selection and early stopping to prevent overfitting [5, 4].Conversely, to avoid underfitting, data collection strategies and augmentation techniques were employed to diversify the training set [6]. With these methods, the model's performance plateaus at epoch 32, suggesting saturation of its learning capacity [5, 4, 6]. However, evaluation metrics demonstrate the effectiveness of the mentioned methods. Overall, YOLOv8 consistently achieves high precision and recall rates for object detection tasks, along with exceptional detection speeds. These factors make YOLOv8 a viable choice for concrete crack detection and mapping.

3.2 Measurement Validation

Table 2: Measured and Analyzed Crack Widths

Image No.	Measured Width (mm)	Analyzed Width (mm)	Difference (mm)	Error (%)
1	2.7	3.02	0.32	11.85
2	3.1	3.09	0.01	0.32
3	2	1.96	0.04	2
4	1.3	3.7	2.4	184.62
5	1	1.7	0.7	70
6	5.6	6.64	1.04	18.57
7	5.3	5.76	0.48	9.06
8	8.5	7.16	1.34	15.76
9	9.8	9.1	0.7	7.14
10	3.4	2.78	0.62	18.24
Average Difference				0.765
Average Error (%)				33.756

Table 2 shows the difference between the measured crack width and the width analyzed by OpenCV contour detection and distance transform. The average difference is 0.765 mm, with an average error of 33.75%. Overall, the results show that the system is a promising tool for concrete crack segmentation and width measurement.

The above data shows a coefficient of -0.445 between width and percentage error, and the largest errors are from images of cracks with widths under 2 millimeters. It can be surmised that the thinner

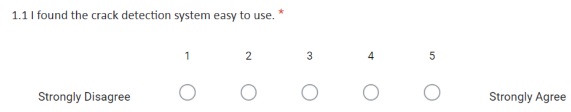

Figure 4: Example of user evaluation question.

the crack on the image, the more excess surface area the machine learning model segments from the image, contributing to the high percentage error for cracks with images under a certain threshold.

3.3 User Evaluation

3.3.1 Demographics. The study employed judgement sampling and concentrated on engaging participants comprising structural engineers, homeowners seeking to identify structural cracks in their residences, building proprietors, building managers, and engineering students. This deliberate selection of these distinct groups is intended to facilitate the acquisition of valuable insights and perspectives from individuals possessing pertinent expertise and hands-on experience in the realm of structural crack assessment. Of the 58 respondents, 25% of them state that they have working experience where their profession allows them to interact with concrete structures.

3.3.2 User Evaluation Survey. The quality of the web application was assessed by using a 5 – Point Likert scale. The ratings presented to respondents were qualitative and in increasing order with “Strongly Disagree” assigned to 1 point, and “Strongly Agree” assigned to 5 points, where 3 points indicates neutrality.

The system was assessed with questions fitting under the categories usability, functionality, and user satisfaction. The mean opinion score for questions under the usability category was 4.4, indicating that users found the system intuitive and easy to use. The mean opinion score for questions under the functionality category was 4.6, indicating that the users found that the features available to the users generally worked as intended. The mean opinion score for questions under the reliability category was 4.7, indicating that users experienced a low failure rate for the system’s features. User Satisfaction was scored at a mean opinion score of 4.6, with an average score of 4.7 for the item “I would highly recommend this crack detection to others” indicating that users were generally pleased with their experience with the application. In summary, the model’s performance and the application’s design generally led users to be generally satisfied with the system.

4 RECOMMENDATIONS FOR FURTHER WORK

In moving forward, the researchers recommend expanding the dataset used for YOLOv8 training by incorporating a larger volume of images to further enhance the model’s adaptability to diverse real-world scenarios. Specifically, a focus on including more instances of hairline cracks in the dataset will contribute to improving the model’s sensitivity to subtle variations in concrete surface conditions.

Additionally, future efforts should explore the incorporation of data that factors in varying angles and perspectives during image capture. Considering the impact of angle on crack measurements, this adjustment in the dataset will contribute to refining the model’s accuracy in assessing crack dimensions across different viewpoints. By addressing such nuances in data representation, we can further optimize the model’s performance in accurately detecting and measuring cracks in real-world applications. This comprehensive approach to dataset enrichment, with a specific emphasis on diverse crack types and angles, will undoubtedly contribute to the continued improvement of the YOLOv8-based concrete crack detection system.

REFERENCES

[1] Juan Camilo Avendaño. *Identification and quantification of concrete cracks using image analysis and machine learning.* 2020.

[2] Hyunjin Bae and Yun-Kyu An. “Computer vision-based statistical crack quantification for concrete structures”. In: *Measurement* 211 (Apr. 2023), p. 112632. DOI: http://dx.doi.org/10.1016/j.measurement.2023.112632.

[3] Kim Basham. *How to evaluate and troubleshoot concrete cracks - recommendations.* Aug. 2022. URL: https://www.forconstructionpros.com/concrete/equipment-products/repair-rehabilitation-products/article/21232603/kb-engineering-llc-how-to-evaluate-and-troubleshoot-concrete-cracks-recommendations.

[4] Jason Brownlee. *Difference between a batch and an epoch in a neural network.* Aug. 2022. URL: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/.

[5] Jason Brownlee. *Use early stopping to halt the training of neural networks at the Right Time.* Aug. 2020. URL: https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/.

[6] Muhammad Faizan. *Apply data augmentation on Yolov5/yolov8 dataset.* June 2023. URL: https://medium.com/red-buffer/apply-data-augmentation-on-yolov5-yolov8-dataset-958e89d4bc5d.

[7] Muhammad Faizan. *Roboflow.* Apr. 2022. URL: https://medium.com/red-buffer/roboflow-d4e8c4b52515.

[8] GeeksforGeeks. *Distance transformation in image - python opencv.* Jan. 2023. URL: https://www.geeksforgeeks.org/distance-transformation-in-image-python-opencv/.

[9] GeeksforGeeks. *Measure size of an object using python opencv.* Apr. 2023. URL: https://www.geeksforgeeks.org/measure-size-of-an-object-using-python-opencv/.

[10] Fortune Business Insights. *Cement market size, share, growth: Emerging trends [2032].* Apr. 2022. URL: https://www.fortunebusinessinsights.com/industry-reports/cement-market-101825.

[11] Byunghyun Kim and Soojin Cho. “Automated vision-based detection of cracks on concrete surfaces using a deep learning technique”. In: *Sensors* 18.10 (Oct. 2018), p. 3452. DOI: http://dx.doi.org/10.3390/s18103452.

[12] Alon Lekhtman. *Train yolov8 instance segmentation on your data.* Feb. 2023. URL: https://towardsdatascience.com/trian-yolov8-instance-segmentation-on-your-data-6ffa04b2debd.

[13] Majid Mirbod and Maryam Shoar. “Intelligent concrete surface cracks detection using computer vision, pattern recognition, and Artificial Neural Networks”. In: *Procedia Computer Science.* Vol. 217. Sept. 2020, pp. 52–61. DOI: http://dx.doi.org/10.1016/j.procs.2022.12.201.

[14] Ibrahim Abayomi Ogunbiyi. *How to deploy a machine learning model as a web app using gradio.* June 2022. URL: https://www.freecodecamp.org/news/how-to-deploy-your-machine-learning-model-as-a-web-app-using-gradio/.

[15] Miguel Rebelo. *What is hugging face?* May 2023. URL: https://zapier.com/blog/hugging-face/.