Article

Detection and Evaluation of Construction Cracks through Image Analysis Using Computer Vision

Alexandre Almeida Del Savio *, Ana Luna Torres, Daniel Cárdenas Salas, Mónica Alejandra Vergara Olivera and Gianella Tania Urday Ibarra

Scientific Research Institute (IDIC), Universidad de Lima, Lima 15023, Peru; aluna@ulima.edu.pe (A.L.T.); danielcardenas70@gmail.com (D.C.S.); mvo945@gmail.com (M.A.V.O.); g.urday97@gmail.com (G.T.U.I.)

* Correspondence: delsavio@gmail.com

Abstract: The introduction of artificial intelligence methods and techniques in the construction industry has fostered innovation and constant improvement in the automation of monitoring and control processes at construction sites, although there are areas where more studies still need to be conducted. This paper proposes a method to determine the criticality of cracks in concrete samples. The proposed method uses a previously trained YOLOv4 neural network to identify concrete cracks. Then, the region of interest, determined by the bounding box resulting from the neural network model classification, is extracted. Finally, the extracted image is converted to negative grayscale to quantify the number of white pixels above a certain threshold, automatically allowing the system to characterize the fracture's extent and criticality. The classification module reached a veracity between 98.36% and 99.75% when identifying five concrete crack types of failures in 1132 images. A qualitative analysis of the results obtained from the characterization module shows a promising alternative to evaluate the criticality of concrete cracks.

Keywords: artificial intelligence; neural networks; YOLO; construction; construction failures; cracks; concrete; computer vision

1. Introduction

In recent years, the construction industry has sought to develop new methodologies to improve its processes, where innovation plays a substantial role supported by new equipment and the permanent and rapid development of technologies. Innovative methodologies improve the construction processes and guide the automation of the monitoring and control of construction works, achieving a better-quality final product.

In line with the construction process automated monitoring, relevant information can be collected by installing conventional and thermographic cameras (sensitive to heat) and generating images and videos that can be used to detect cracks, microcracks, and other construction problems related to concrete [1], which is one of the most used materials in construction [2]. Although there have been previous studies using the YOLO family to detect and classify cracks [3], they are limited and do not consider the analysis of the detected cracks in concrete to measure the criticality of the failure. Based on this, it is proposed in this work to detect and evaluate construction cracks through image analysis using computer vision.

State of the Art

Several researchers have evaluated the use of artificial intelligence in the construction industry. Refs. 4,5 evaluated the YOLOv4 object identification and classification algorithm to identify objects of interest in construction sites based on images from drones and static cameras. Eight classes of objects were identified in 1000 drone images and 1046 static camera images of a construction site, with an accuracy ranging from 78.8% to 82.8% and 73.56% to 93.76%, respectively.
An enhanced convolutional neural network can detect objects in real-time. The database used by [6] consists of 10,000 images classified into “worker” and “excavator.” The results were measured by mean average precision (mAP); a 91% precision was obtained for the class “worker” and 95% for “excavator.”

A detection algorithm proposal was developed by [7]: Contrastive Res-YOLOv5 Network, or CORY-Net. This algorithm was compared to YOLOv3, Faster R-CNN, and Improved-YOLOv5 to verify the application’s detection and classification accuracy. The generated database contains 22 classes and 12,000 images obtained from a construction site from the “Safety-Helmet-Wearing-Dataset” database, of which 7146 were manually classified as part of the training and testing process. Mean average precision (mAP) values of 63.7%, 69.7%, 70.2%, and 74.2% were obtained in the YOLOv3, Faster R-CNN, Improved YOLOv5, and the proposed CORY-Net algorithms, respectively.

On the other hand, ref. [8] proposed a training process executed on a neural network to ensure an adequate social distance policy in a construction environment to identify people inside construction sites. They identified people performing activities in an upright and a crouched position using the YOLOv4 algorithm with an accuracy of 77.98%.

More focused on controlling structural failures, ref. [9] proposed using convolutional neural networks to detect and classify cracks in concrete from images. The database consists of 40,000 images with a 256 × 256 pixels resolution. This number of images started from 277 for the training and validation processes out of 332 raw images, and they were taken between 1.0 and 1.5 m from the objects of interest and with different lighting variations. The result reached 98% accuracy for detecting and classifying cracks.

Also, ref. [10] proposed a framework based on the fusion of convolutional neural networks and the Naive Bayes algorithm, called NB-CNN, to detect cracks in underwater metallic surfaces in multiple video frames. The final database consists of 147,344 images with cracks and 149,460 without cracks. The proposed merger achieved a TPR (number of true positives among total positives) of 99.9% and a hit rate of 98.3%. This framework can detect small cracks with low contrast and a considerable variation in brightness; however, the neural network requires a large amount of data for training, at least 100,000, to avoid overfitting.

Another research by [11] presented a model based on the Fast-RCNN architecture, Crack Deep Network (CrackDN), to detect sealed and unsealed cracks in complex road bottoms and compare it with the performance of the Faster-RCNN and SSD300. Images for the database were collected from different devices, smartphones, and mounted cameras to obtain a final amount of 12,000 images with a resolution of 500 × 356 pixels. As a result, CrackDN had a higher mAP in detecting sealed and unsealed cracks of 97.48% and 97.56%, while Faster-RCNN obtained 96.41% and 95.46%, and SSD300 obtained 89.97% and 89.47%, respectively.

A method based on a fully convolutional network (FCN) to detect image cracks is shown in [12]. The neural network was trained with a database of 500 manually classified images of 227 × 227 pixels, obtaining an average accuracy of 90%.

A convolutional neural network to recognize cracks in asphalt surfaces is proposed by [13]. The database consists of 5000 manually classified 3D images of different pavement sections. The proposed neural network was evaluated on the training and test information, obtaining 96.32% and 94.29% accuracy, respectively.

On the other hand, ref. [3] studied the efficiency of using a deep neural network (YOLOv5) to identify concrete fractures, specifically in bridges, using drone images. The authors focused on obtaining high precision (mAP = 0.976) without sacrificing identification speed using focal loss, pruning, and data scaling techniques.

Regarding the use of thermal imaging cameras in the construction industry, ref. [14] developed the BIA (Building Insulation Analyzer) software to measure the differences between the interior and exterior temperatures at three points on the façade of a building, whether openings such as window corners or doors. Two images were taken with a thermal
camera and one with a reflex camera to evaluate temperatures. A variation of 2 °C was obtained in the openings and 5 °C in the insulated and thick walls.

A method based on the analysis of thermal images that consists of three phases is proposed by [15]: collecting original and thermal images, estimating the position of the captured image, and updating the 3D model of the construction site in the BIM software. The proposal was applied to a building under construction in Suzhou, China. In conclusion, when processing the original images, it failed to identify the elements of the building due to low image quality, poor light conditions, and image noise. In the case of the mixture of the original and thermal images, a high percentage of object detection (95%) was obtained.

On the other hand, ref. [16] evaluated the application of machine learning and image analysis algorithms to analyze metallic structures from thermographic images. It was found that minor defects can be detected when they are closer to the surface. Therefore, it was suggested to use higher resolution images to obtain more pixels per fault image.

Regarding the characterization of fractures, there is extensive literature on the subject. Still, we can highlight the seminal work of [17], who sought to provide a quantitative description of the fracture surface in polymer-based concrete and concluded that the geometry of a surface fracture was dependent on the scale of observation. More recent works, such as that of [18], sought to understand and characterize the evolution of a concrete fracture process using the Digital Image Correlation technique; they concluded that the size of the fracture area or zone in a piece of concrete is related to the size of the constructive element. On the other hand, ref. [19] used an X-ray scanner to generate computed tomography images to assess the evolution of damage in the internal structure of the concrete under stress situations. Finally, ref. [20] unified the techniques of fractal dimensions and neural networks (YOLOv5, among others) to segment the fractures. However, although the computational complexity of the chosen method was relatively low, they were unable to measure or estimate the size of the fracture.

In this line of research, the development of mechanisms for the automation of the control and monitoring of construction processes is proposed based on the use of artificial intelligence and the capture of images from conventional cameras (PTZ and Bullet) and infrared cameras. The aim is to develop an alert and prediction model that automatically and constantly monitors the construction process during its execution, with the ability to detect deficiencies or failures (cracks) in the concrete.

2. Materials and Methods
2.1. General Operation

The research’s general methodological process is shown in Figure 1. As a first step, the types of failures in concrete structures that fall within the scope of the investigation were determined, as well as the predominant characteristics that allow the detection of such failures by various methods.

The second step was an experimental stage, in which concrete blocks were created considering certain thresholds in the variables analyzed (inputs, process, temperature, times, among others) that have a high probability of generating the failures. For this stage, images were taken in a controlled environment. The purpose of this stage was to generate images for a subsequent analysis that allowed the manual evaluation of the essential characteristics of the failures to be identified in an automated way. Once the visual attributes to be searched for were determined, the artificial intelligence model that identifies the distinctive features of the images was selected.

If the selected artificial intelligence model belongs to the type of supervised learning model, a training stage is necessary based on the images obtained from the concrete specimens (steps three to four). This experimental stage includes a sub-process called focal loss (step five) to reduce the potential imbalance in the dataset between positive and negative images. Since detection speed is not considered a relevant variable in this study, the pruning technique will not be applied. Data augmentation techniques (like brightness modification or images of various sizes) will be used, as proposed by [3]. Once good
indicators are obtained (e.g., mAP > 70%), it can proceed to the operational phase, where images of actual construction to be analyzed are obtained using the same thermographic camera and the previously trained model. Finally, as the last step, it is necessary to validate if the results obtained by the system are within acceptable thresholds.

Figure 1. General methodology.

2.2. YOLOv4

The YOLO (“You Only Look Once”) neural network model, version 4, was used for this research. YOLO was presented in 2016 as a single convolutional network for detecting and classifying objects in images and videos in real-time [21]. Using the entire image, it can simultaneously generate multiple bounding boxes with their respective object classification accuracy scores. This results in a higher processing algorithm speed without compromising the detection accuracy. During the training phase, the model was configured according to the number of classes to be trained; for the first training process, two classes were used, so it was set with 4000 batches, based on the formula “n° classes × 2000”, while for the rest of the training five classes were used, so 10,000 batches based on the previous work by [4].

2.3. Instruments

The images needed to apply the methodology proposed were obtained from the thermal camera in Table 1 and from the videos obtained by the video surveillance equipment in Table 2.

2.4. Training

A neural network is trained with the objects and classes that must be identified. This stage is executed once per project. Figure 2 shows the objects to be identified within the images obtained from the construction site.
### Table 1. Instruments used.

<table>
<thead>
<tr>
<th><strong>Thermal Camera</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Detector resolution</td>
<td>320 × 240 or 640 × 480</td>
</tr>
<tr>
<td>Touch screen</td>
<td>3.5 inches LCD of 640 × 480</td>
</tr>
<tr>
<td>Digital zoom</td>
<td>2× and 4×</td>
</tr>
<tr>
<td>Integrated digital camera</td>
<td>5 megapixels</td>
</tr>
<tr>
<td>Temperature measurement range</td>
<td>−20 °C to +1200 °C</td>
</tr>
<tr>
<td>Precision</td>
<td>±2 °C or 2%</td>
</tr>
<tr>
<td>Thermal sensitivity</td>
<td>≤0.05 °C to 30 °C</td>
</tr>
<tr>
<td>Battery life</td>
<td>3–4 h per battery</td>
</tr>
<tr>
<td>Color alarm</td>
<td>High temperature, low temperature, and isotherms</td>
</tr>
<tr>
<td>Operating temperature</td>
<td>−10 °C to +50 °C</td>
</tr>
<tr>
<td>Storage temperature</td>
<td>−20 °C to +50 °C</td>
</tr>
<tr>
<td>Size (H x W x L)</td>
<td>27.7 × 12.2 × 16.7 cm</td>
</tr>
<tr>
<td>Weight (battery included)</td>
<td>1.04 kg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Motorized PTZ IP Camera</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>SD10A848WA-HNF</td>
</tr>
<tr>
<td>Maker</td>
<td>DAHUA</td>
</tr>
<tr>
<td>Sensor</td>
<td>1/1.8” CMOS</td>
</tr>
<tr>
<td>Lenses</td>
<td>5.7 mm−275 mm</td>
</tr>
<tr>
<td>Zoom</td>
<td>48×</td>
</tr>
<tr>
<td>Image sensor</td>
<td>1/1.8” CMOS</td>
</tr>
<tr>
<td>Effective pixels</td>
<td>2.42 MP</td>
</tr>
<tr>
<td>Max. Resolution</td>
<td>1920 (H) × 1080 (V)</td>
</tr>
<tr>
<td>ROM</td>
<td>8 GB</td>
</tr>
<tr>
<td>RAM</td>
<td>2 GB</td>
</tr>
<tr>
<td>Electronic shutter speed</td>
<td>1/3 s–1/100,000 s</td>
</tr>
<tr>
<td>Scanning system</td>
<td>Progressive scanning</td>
</tr>
<tr>
<td>Colour</td>
<td>0.001 Lux@F1.4</td>
</tr>
<tr>
<td>Min. Illumination</td>
<td>B/W: 0.0001 Lux@F1.4</td>
</tr>
<tr>
<td></td>
<td>0 Lux (IR light on)</td>
</tr>
</tbody>
</table>

![Image of a thermal camera](image-url)  

**Figure 2.** Example of classes (cracks) to classify.

A frame extraction algorithm was used in the videos where the objects of interest were found to obtain images with the elements to be identified. Then, the neural network was
trained to identify the desired features. The training process is only carried out with the classes necessary for the domain under study; in this case, five classes.

Thermal Camera

Figures 3–5 show images of the elements of interest within the construction work of a building located in Lima, Peru.

Figure 3. Original image of a column and the same column with thermographic elements.

Figure 4. Original image of a beam and the same beam with thermographic elements.

Figure 5. Original image of a slab and the same slab with thermographic elements.
2.5. Model Development

2.5.1. Frame Extraction

The frame extraction process has as input information the videos obtained from video surveillance cameras. A total of 2929 images were obtained from the video compilation process, including 1611 beams and 1318 test tubes. A total of 1132 images contained concrete cracks. The resulting images shown in Figure 6 have a dimension of 3840 × 2160 pixels, with a resolution of 96 dpi. Next, YOLOv4 reset image sizes to 416 × 416 pixels in its initial processing stages. Lower input image resolutions can cause fuzzy features in small cracks, which is not recommended [3]. For the execution of the Algorithm 1 [24], the IDLE development environment with the Python programming language version 3.7 was used.

Algorithm 1. Frame extraction algorithm

1. Reading the configuration parameters
   Video file reading path
   a. Results in storage path
   b. Image filter (for example, “*.MP4”)
   c. Interval between frames
2. Get a list of files to be processed
3. For each file
   a. Create a directory with the video file name
   b. Get the total list of frames
   c. Filter frames to process by interval
   d. For each frame
      ○ Generate JPG image
      ○ Update result file images.txt.
      ○ Update image counter

Configuration and results:
Libraries used: decord, cv2, numpy, pathlib, os
Total generated images: 2929

Figure 6. Example of images resulting from the frame extraction process.

2.6. Manual Image Classification

LabelImg was used for manual classification. LabelImg is an open-source graphical image tool on GitHub that provides image annotation mechanisms by labeling bounding boxes of objects. This tool is available for Linux, Windows, and macOS operating systems. It is written in Python and uses Qt as a graphical interface [25].

According to [26], cracks can be classified according to their origin and moment of appearance:
• Cracks originated in the plastic state.
• Cracks caused by plastic contraction. The leading causes from which this type of crack usually originates are hydraulic shrinkage during setting and excessive vibration or trowel.

• Cracks originated from the plastic settlement due to four factors: little coating and excessive diameters in the steel, changes in consistency in continuous pours, displacement of the formwork, and deformation of the supporting ground.

• Cracks originated in the hardened state, among which cracks originated by spontaneous movements caused by: contraction due to carbonation and thermal shrinkage; numbness due to thermal expansion, excessive oxidation of reinforcing steel, or excess expansive in cement; and alkali-aggregate reaction.

Another case of cracks in the hardened state is produced by loads caused by compression, traction, bending, shear, and torsion stresses. Figure 7 shows the types of compression failures that can be generated in concrete.

Figure 7. Types of cracks in concrete.

The images obtained from the process indicated in Section 2.5.1 were saved in a folder with the classes.txt file containing the classified classes. The manual image classification process was conducted with 1132 images and five classes (Beam_bending_failure, Column_type2_failure, Column_type3_failure, Column_type4_failure, and Column_type5_failure) to classify the types of cracks in beams and test tubes within the images. Table 2 shows the five classified classes and the region of interest where the compression failures were generated. Figure 8 shows how the region of interest was selected using the software LabelImg.

2.7. Neural Network Training

All training processes were performed locally on a computer with an Intel Core i7 processor, 64 GB RAM, and NVIDIA GeForce RTX 2080 graphics card.

The manually classified images and their .txt files were distributed randomly in two folders for training and validation. The training folder contained 70% of the images, and the validation folder included 30%.

In the first instance, initial training was carried out where 200 images were used, so the amount in the training folder was 140, while the validation folder contained 60. The training was carried out with only the necessary classes for the domain under study, “Crack” and “No_crack.”

The training process for the 200 images lasted approximately 20 h, obtaining 4 “.weight” files. The yolov4_custom_best.weights file had the best mAP (Mean Average Precision) result, with 89.77%. The yolov4_custom_best.weights file was used for evaluation on two images, IMG75 and IMG175, for the detection and classification of manually classified classes “Crack” and “No_crack.”

Figure 9 shows the classification and detection of the classes “No_Crack” and “Crack” in images (a) and (b), respectively. The first class was detected with 98% accuracy and the second with 99% accuracy.
Table 2. Classes for manual classification.

<table>
<thead>
<tr>
<th>Type</th>
<th>Image</th>
<th>Area Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 2</td>
<td><img src="image1.png" alt="Type 2 Image" /></td>
<td><img src="image2.png" alt="Type 2 Area Detail" /></td>
</tr>
<tr>
<td>Type 3</td>
<td><img src="image3.png" alt="Type 3 Image" /></td>
<td><img src="image4.png" alt="Type 3 Area Detail" /></td>
</tr>
<tr>
<td>Type 4</td>
<td><img src="image5.png" alt="Type 4 Image" /></td>
<td><img src="image6.png" alt="Type 4 Area Detail" /></td>
</tr>
<tr>
<td>Type 5</td>
<td><img src="image7.png" alt="Type 5 Image" /></td>
<td><img src="image8.png" alt="Type 5 Area Detail" /></td>
</tr>
<tr>
<td>Beam</td>
<td><img src="image9.png" alt="Beam Image" /></td>
<td><img src="image10.png" alt="Beam Area Detail" /></td>
</tr>
</tbody>
</table>
Four training sessions were carried out for training with the total number of images (1132 files). All images and corresponding .txt files were distributed among the training (792 images) and validation (340 images) folders.

The training was processed only with the classes necessary for the domain under study. In the case of this research, five classes were worked on: Beam_bending_failure, Column_type2_failure, Column_type3_failure, Column_type4_failure and Column_type5_failure. This process was performed with Darknet, an open-source neural network model written in C and CUDA [27].

The best mAP (mean average precision) result from the training process was n°3 with 99.75%, as shown in Table 3.

The yolov4_custom_best.weights file from training n°3 was used to evaluate the detection and classification of manually classified objects in Section 2.5 in two images, IMG903 and IMG1076.

Figure 10 shows the classification and detection of the objects “Column_type4_failure” and “Beam_bending_failure” in images (a) and (b), respectively. Both objects were detected with 100% accuracy.
Table 3. Results of the neural network training.

<table>
<thead>
<tr>
<th>Training</th>
<th>Training Images (70%)</th>
<th>Validation Images (30%)</th>
<th>Best mAP</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>n°1</td>
<td>792</td>
<td>340</td>
<td>98.36%</td>
<td>0.96</td>
<td>0.99</td>
</tr>
<tr>
<td>n°2</td>
<td>792</td>
<td>340</td>
<td>99.49%</td>
<td>0.96</td>
<td>0.97</td>
</tr>
<tr>
<td>n°3</td>
<td>792</td>
<td>340</td>
<td>99.75%</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>n°4</td>
<td>792</td>
<td>340</td>
<td>96.62%</td>
<td>0.95</td>
<td>0.97</td>
</tr>
</tbody>
</table>

The training was processed only with the classes necessary for the domain under study. In the case of this research, five classes were worked on: Beam_bending_failure, Column_type2_failure, Column_type3_failure, Column_type4_failure and Column_type5_failure. This process was performed with Darknet, an open-source neural network model written in C and CUDA [27].

The best mAP (mean average precision) result from the training process was n°3 with 99.75%, as shown in Table 3.

The yolov4_custom_best.weights file from training n°3 was used to evaluate the detection and classification of manually classified objects in Section 2.5 in two images, IMG903 and IMG1076.

Figure 10 shows the classification and detection of the objects “Column_type4_failure” and “Beam_bending_failure” in images (a) and (b), respectively. Both objects were detected with 100% accuracy.

2.8. Fracture Characterization

Static images were considered to identify fractures in the concrete since it was not intended to evaluate the dynamics of the fracture. Unlike previous fracture characterization works that use fractal dimensioning [18] or X-ray devices [20], in this work, the classification results from the neural network were based on the coordinates in the X and Y axes of the “bounding box.” Then, using computer vision functions, the image that contained only the previously detected fracture was extracted, which was converted to grayscale and then to negative so that the fracture was established with pixels with a certain tendency to white. Then, the number of pixels whose chromatic values in gray were above an arbitrary value (in this case, 160, where 255 represents the white color) were counted in such a way that the greater the number of “white” pixels, the greater the estimated area of the fracture and, consequently, its criticality.

Figures 11–13 show the classified image resulting from the neural network with the bounding box (a) and their classification accuracy value, 100% of accuracy for the detection and correct classification of the classes “Column_type3_failure”, “Beam_bending_failure” and “Column_type4_failure”, respectively. The fracture section extracted from the bounding box is in (b), and the resulting image converted to grayscale and negative in (c). The number of pixels above the arbitrarily defined threshold (160) was 650 in Figure 11c, 62 in Figure 12c, and 7257 in Figure 13c.
detection and correct classification of the classes "Column_type3_failure", "Beam_bending_failure" and "Column_type4_failure", respectively. The fracture section extracted from the bounding box is in (b), and the resulting image converted to grayscale and negative in (c). The number of pixels above the arbitrarily defined threshold (160) was 650 in Figure 11c, 62 in Figure 12c, and 7257 in Figure 13c.

Figure 11. (a) Original image with bounding box. (b) Extracted section corresponding to the bounding box. (c) Image converted to negative grayscale.

Figure 12. (a) IMG903 with the bounding box. (b) Extracted section corresponding to the bounding box. (c) Image converted to negative grayscale.
was not used, in addition to a relatively limited dataset of 1132 images, with 792 used for training. Since the results obtained are (mAP between 96% and 99%) above the minimum expected threshold (mAP = 0.70), they demonstrate that the use of a model with better performance characteristics, such as YOLOv5, YOLOv6, or YOLOv7, together with a higher amount and variability of training images can achieve better model performance. The possibility that the current model suffers from underfitting (high bias and low variation) may cause the lower performance of the model; however, it can be overcome either by modifying the parameters of the model (for example, by increasing the number of epochs or stages) or by increasing the training data.

Regarding the characterization of the fracture, in which a simple count of pixels with a chromatic value greater than 160 has been used, the results indicate the feasibility of use in certain conditions. First, the general “whiteness” of the negative greyscale image affects the pixel count, as can be inferred from Figure 13c, which contained several white pixels comparatively higher than the other two samples (7257 in Figure 13c versus 650 and 62 pixels in Figures 11c and 12c, respectively). This evidences the need to calculate the average “white” value to determine the threshold. Second, while the image size was not a variable in this study, it can affect the results if images of different sizes are compared against each other.

4. Conclusions

This study demonstrates the feasibility of utilizing a specialized neural network model, YOLOv4, to effectively detect and assess various types of cracks and fractures in concrete.

Figure 13. (a) IMG1076 with the bounding box. (b) Extracted section corresponding to the bounding box. (c) Image converted to negative grayscale.
beams and columns. The achieved precision, measured by mean average precision (mAP), ranged from 96.62% to 99.75%. The model was trained on a modest set of high-resolution images (792 images, 3840 × 2160 pixels, 96 dpi).

To further enhance the neural network’s performance, the following recommendations are proposed:

- Consider adopting more recent YOLO model iterations, such as YOLOv6 or YOLOv7.
- Fine-tune the model’s operational parameters to prioritize performance gains over shorter training times.
- Apply pruning to reduce the computational cost of the neural network, as well as other preprocessing and data augmentation techniques.
- Expand the training dataset significantly, encompassing over 2000 images.
- Introduce greater diversity to the images, incorporating different concrete compositions, varied angles, and lighting conditions.

Concurrently, an alternative mechanism demanding minimal computational resources has been suggested for evaluating fracture criticality. This approach involves a grayscale conversion and pixel chromatic value comparison to gauge the significance of a fracture based on pixels near the target in a converted grayscale image. A comprehensive analysis is recommended to validate the proposed method’s precision, contrasting automatic pixel-based assessments with manual or visual evaluations.

However, it is important to note some limitations of the proposed method that could be part of relevant future research:

- The method is limited to static evaluations and does not account for dynamic fracture changes over time.
- Factors like the size of the analyzed structural element, lighting conditions, porosity of the building material, and camera quality and positioning influence the outcomes. Additional experimentation with these variables is essential to establish standardized fracture criticality scales.
- The implemented methodology is limited to applying the pre-existing YOLOv4 algorithm in a customized database for detecting and evaluating cracks in construction, so preprocessing techniques such as pruning have not been applied.

In conclusion, the proposed mechanism is not designed to offer an exact quantitative assessment of crack size. Instead, its primary purpose is to identify regions warranting closer in situ examination. This application holds particular significance for hard-to-reach structures like bridges or dams, where automated preliminary identification via drones proves advantageous.


Funding: This article was developed as part of the “Detection of construction failures by analyzing frames using AI” research project, funding number AC.06.016.2022, of the Scientific Research Institute (IDIC) of Universidad de Lima.

Data Availability Statement: The data supporting the findings of this study are available within the article.

Acknowledgments: The authors thank the Universidad de Lima for supporting this research.

Conflicts of Interest: The authors declare no conflict of interest.


**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.