Learning and Nonlinear Models - Journal of the Brazilian Society on Artificial Intelligence (SBIA), Vol. 23, Iss. 2, pp. 46–58, 2025
https://doi.org/10.21528/lnlm-vol23-no2-art4

A DEEP LEARNING MODEL FOR AUTOMATED CRACK DETECTION AND CHARACTERIZATION ON BUILDING ELEMENTS

Leticia M. G. Morais
Laboratory for Optimization, Analysis and Design of Structures and Materials - LOADS
Federal University of Semi-Arid Region
leticia.morais30463@alunos.ufersa.edu.br

Heitor C. Dantas
Laboratory for Optimization, Analysis and Design of Structures and Materials - LOADS
Federal University of Semi-Arid Region
heitor.dantas@alunos.ufersa.edu.br

Paulo H. A. Bezerra
Federal University of Semi-Arid Region, Department of Engineering and Technology
paulo.bezerra@ufersa.edu.br

Rosana C. B. Rego
Federal University of Semi-Arid Region, Department of Engineering and Technology
rosana.rego@ufersa.edu.br

Abstract – Detecting civil construction defects, such as material deterioration and cracks on masonry or structural members, is crucial for building safety and durability. Among all defects, cracks – whether visible or hidden – are critical issues that can compromise the integrity of buildings, bridges, roads, and other infrastructure elements. Artificial intelligence (AI) approaches, such as deep learning algorithms, can assist in the early identification and characterization of cracks, facilitating preventative actions to avoid future problems. In this study, we explore the application of deep learning with image segmentation techniques for crack detection and characterization in civil construction elements. We implemented a residual neural network capable of detecting cracks either in isolation or by mapping their distribution across surfaces such as concrete, bricks, steel, and wood. Additionally, we integrated the segmentation model SAM to improve the precision of crack segmentation in images. Through simulations and comparative analysis, we evaluated the performance of the models in accurately identifying and delineating cracks in civil infrastructure. The proposed model achieved an accuracy of 100% and an intersection over union of 0.95. Despite these high-performance metrics, there remains room for error analysis to further refine the approach, particularly in complex or edge-case scenarios. These results demonstrate the efficacy of the proposed approach in achieving accurate crack detection.
Keywords – Deep Learning, crack detection, artificial intelligence, civil engineering, image segmentation.

1 Introduction

Cracks in structural elements may lead to significant damage, therefore, crack detection on building elements is crucial for keeping the structural integrity, safety, and durability of buildings and infrastructure [1, 2]. The importance of automated crack detection systems extends beyond technical considerations to encompass significant economic, safety, and social implications. Structural failures resulting from undetected or improperly assessed cracks can lead to catastrophic consequences, including building collapses, bridge failures, and infrastructure disasters that may result in loss of life, injuries, and substantial economic losses [1]. Crack characterization is important for comprehending the nature, causes, and potential impacts of cracks in building materials; furthermore, diagnosing construction issues relies on identifying the origin causes of cracks, whether they stem from material fatigue, foundation settlement, thermal expansion, or structural overload [3, 4]. Early detection and timely repairs can prevent minor issues from escalating, thereby prolonging the lifespan of construction elements and ensuring their safe use [5, 6]. Effective repair and maintenance strategies are dependent on proper crack characterization. This understanding is essential for designing appropriate remedial actions and assessing the severity and potential impact of the cracks on the structural integrity of the building, since different types of cracks require specific repair techniques [7, 8]. However, traditional methods of crack detection and characterization often depend on visual inspection, which can be time-consuming, subjective, and predisposed to human error [3].

Manual inspection of this vast infrastructure network is not only labor-intensive and expensive but also inherently limited in its ability to detect hairline cracks or defects in inaccessible areas. Various inspection methods and technologies are employed to identify cracks. These approaches include direct physical examination of the structure to spot visible cracks, as well as sophisticated non-destructive testing techniques. However, cracks can sometimes appear in concealed or hard-to-see areas, making them difficult to detect through standard visual inspections. Artificial intelligence (AI) techniques, including deep learning algorithms,

https://doi.org/10.21528/lnlm-vol23-no2-art4

can aid visual inspections by detecting cracks at their initial stages and characterizing them by analyzing their various attributes such as size, shape, orientation, and severity. Automated crack detection systems powered by AI offer the potential to conduct more frequent, consistent, and comprehensive inspections at reduced costs, enabling proactive maintenance strategies that can significantly extend the service life of structures.

Motivated by recent advances in deep learning for crack detection (as reviewed in Section 2), in this work, we implemented a ResNet with 50 layers with a segmentation technique designed for crack detection and characterization on building elements, as proposed by [9]. ResNet was chosen due to its superior performance compared to various deep learning models – including Inception, MobileNet, VGG, and U-Net – for crack detection on civil infrastructure surfaces, as highlighted in previous work [9]. Moreover, we proposed a crack characterization procedure to understand the potential impacts of cracks in building materials. The main contributions of this work are:

i) Propose a framework that integrates the design and training of deep learning architectures for crack detection, segmentation, and characterization.

ii) Develop a characterization method to evaluate specific features of the detected cracks, such as their slope and whether they are isolated or interconnected with others.

Table 1 summarizes the main symbols and notations used throughout this paper.

Table 1: Summary of symbols and notations.

Symbol/Notation	Description
$\mathbf{x}$	Input to a residual block
$\mathbf{y}$	Output of a residual block
$\mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\})$	Residual mapping with weights $\mathbf{W}_i$
$\sigma(\mathbf{x})$	Sigmoid activation function
$\mathcal{L}(p, q)$	Binary cross-entropy loss function
y_i	Label (0 or 1) for sample i
$p(y_i)$	Predicted probability for the negative class
$q(y_i)$	Predicted probability for the positive class
N	Number of samples
IoU	Intersection over Union metric
$\mathbf{P}$	Set of pixels in predicted mask
$\mathbf{G}$	Set of pixels in ground truth mask
p_t	Model's estimated probability for the true class
α_t	Weighting factor for class imbalance (focal loss)
γ	Focusing parameter (focal loss)
p_i	Predicted intensity for pixel i (Dice loss)
g_i	Ground truth intensity for pixel i (Dice loss)
FL(p_t)	Focal loss
DL	Dice loss
SAM	Segment Anything Model
ReLU	Rectified Linear Unit activation function
RMSprop	Root Mean Square Propagation optimizer
CNN	Convolutional Neural Network
FCN	Fully Convolutional Network

The paper is divided as follows: In section 2, related works on deep learning approaches for crack detection are reviewed. In section 3, the datasets used for training and test the models are presented. In section 4, the ResNet background is recalled, and the proposed model is presented with the procedure used to detect cracks. In section 5, the proposed crack characterization algorithm is presented. In section 6, the results are presented. Finally, in section 7, the conclusions are discussed.

2 Related Works

Detecting surface cracks presents several challenges influenced by factors like surface complexity, changing environmental conditions, and the demand for high precision. To overcome these obstacles, researchers have been developing advanced algorithms and technologies aimed at better interpreting material properties and surface features. In this context, deep learning models have been employed by various studies to support and enhance surface crack detection. Deep learning models, such as convolutional neural networks (CNNs), which are built for image and video analysis, can be employed for detecting and characterizing cracks in construction materials. Many CNN architectures exist, including VGG (Visual Graphics Group) [10, 11], Inception [12], ResNet (Residual Network) [13, 14], EfficientNet [15], Densely Connected Convolutional Networks [16, 17], SqueezeNet [18], among others.

https://doi.org/10.21528/lnlm-vol23-no2-art4

Several recent studies have proposed deep learning-based approaches for crack detection and segmentation. Table 2 presents a comparative overview of recent deep learning-based approaches for crack detection. For instance, Liu et al. [11] developed an enhanced Fully Convolutional Network (FCN) combined with Deeply-Supervised Nets (DSN) for crack segmentation. Their model aggregates hierarchical features from 13 convolutional layers inspired by the VGG-16 architecture. This approach achieved state-of-the-art performance on the proposed dataset, obtaining a mean Intersection over Union (IoU) of 85.9% and a best F1-score of 86.5%. In another study, Fei et al. [17] proposed CrackNet-V, an improved version of CrackNet, aimed at pixel-level crack detection in 3D asphalt pavement images. Their deep learning model demonstrated a precision of 84.31%, a recall of 90.12%, and an F1-score of 87.12%. Yang et al. [19] explored a transfer learning strategy involving multiple convolutional neural networks for crack classification. Their approach yielded a high accuracy of 99.50%, highlighting the effectiveness of using pre-trained models for crack detection tasks. Differently, Tabernik et al. [20] proposed a convolutional neural network-based segmentation method for detecting surface defects in industrial semi-finished products. They trained their model on the Kolektor Surface-Defect Dataset and achieved an accuracy of 99.2%. Bae et al. [21] introduced SrcNet (Super-resolution crack network), a model designed to improve automated crack detection in civil infrastructure, particularly in situ bridge inspections. By leveraging super-resolution techniques, SrcNet enhances the visual quality and detection accuracy of fine cracks.

Building on the potential of deep learning in different contexts, Cao et al. [22] introduced an enhanced framework based on YOLOv9c for underwater crack detection using sonar imagery. Their approach was benchmarked against several state-of-the-art models, including YOLOv9c, YOLOv8, YOLOv5, YOLOX, RT-DETR, and Faster R-CNN. The proposed model achieved the best performance, with a mean Average Precision (mAP) of 0.835 and an F1-score of 0.773, highlighting its robustness in complex underwater environments. Wang et al. [23] employed the ResNet50 model alongside a large and diverse dataset to detect cracks on concrete surfaces. To enhance the interpretation of the results, the authors developed a 3D model that virtually visualizes the detected cracks, offering a more intuitive and detailed understanding of structural damage. Similarly, Wang et al. [24] explored the effectiveness of deep learning models in detecting cracks on building surfaces by comparing ResNet50 and YOLOv8. To improve real-world applicability, the authors integrated these models into Unmanned Aerial Vehicles (UAVs), enabling automated and scalable inspection processes. A comprehensive dataset was collected, containing both cracked and non-cracked surface imagery. The study concluded that classification models such as ResNet50 generally outperform object detection models like YOLOv8 in terms of detection accuracy and reliability for this application. In [25], the authors proposed an ensemble-based deep learning approach for automatic classification of building facade defects. Specifically, they combined recent and complex neural network architectures – Vision Transformers (such as ViT B16 and Swin B) and ConvNeXts – into two ensemble strategies. They introduced a new open-source dataset called Facade Building Defects, containing labeled images of four defect classes: cracks, spalling, stains, and vegetation.

Table 2: Performance metrics of some deep learning models for crack detection.

Work	Dataset	Model	Accuracy	F1-score
[11]	Custom dataset	FCN + DSN (VGG-16)	0.9860	0.8650
[17]	3D Asphalt	CrackNet-V	–	0.8712
[19]	CCIC [26], SDNET [27], BCD [28]	CNNs + Transfer Learning	0.9950	–
[20]	Kolektor Surface	CNN-based segmentation	0.9920	–
[21]	DIV2K [29]	SrcNet (Super-Resolution)	0.9617	–
[23]	CCIC [26], SDNET [27], CFD [30], Crack500 [31], CrackTree200 [32], DeepCrack [11]	ResNet50	0.9700	–
[22]	Generated dataset based on a sonar simulation platform	improved YOLOV9c	–	0.773
[24]	CCIC [26], SDNET [27], CFD [30], Crack500 [31], CrackTree200 [32], DeepCrack [11]	ResNet50 and YOLOv8	0.99	–
[25]	PEER Hub ImageNet and custom Facade Building datasets	ViT B16, Swin B, and ConvNeXts	0.9090	–

3 Dataset

The dataset for training the proposed classification model includes 40,000 images, with both positive and negative cases of crack presence [26]. It is divided into 20,000 images for the positive class and 20,000 images for the negative class. Additionally, we used the Bricks surface crack dataset presented by [33], the steel surface crack dataset proposed by [34], and the wood surface defect dataset presented by [35] to test and evaluate the model with cracks in different material surfaces.

The bricks dataset contains a relatively balanced number of images with and without cracks, totaling 400 images. In contrast, the steel defect dataset is unbalanced, with 12,568 images, of which 6,666 contain defects, and 5,902 do not. Similarly, the wood surface defects dataset is heavily skewed, comprising 20,276 images, of which 18,284 have defects, and 1,992 are defect-free. To standardize the input data and reduce computational load, all images were resized to 96×96 pixels and converted from RGB

https://doi.org/10.21528/lnlm-vol23-no2-art4

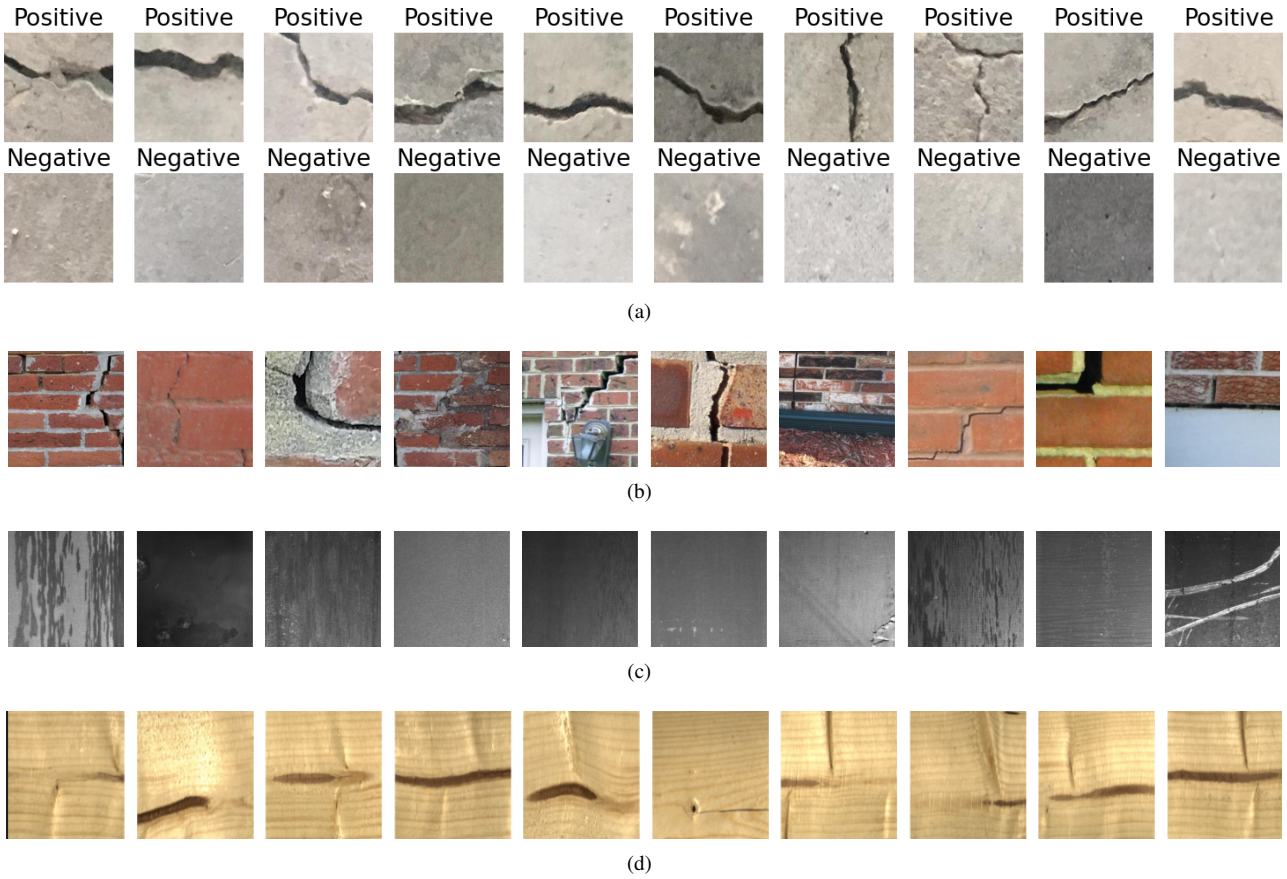
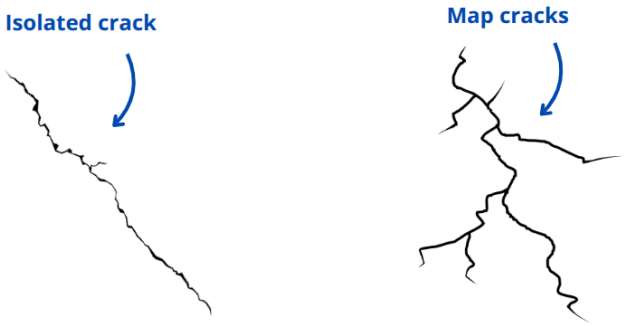


Figure 1: (a) concrete surface crack [26], (b) bricks surface crack [33], (c) steel surface crack [34], and (d) wood surface defect [35] .

to grayscale before being used in the experiments. Although the original images were in color, grayscale versions were adopted to simplify processing. Moreover, the imbalances were addressed during training using data augmentation and class weighting strategies to mitigate bias and improve model performance.

These cracks can be formally categorized into two main types based on their spatial distribution and morphological patterns: isolated cracks and map cracks. Figure 2 illustrates an isolated crack and maps cracks. Isolated cracks are singular, discontinuous fissures that occur independently of other cracks. They typically manifest due to localized phenomena, such as drying shrinkage, differential settlement, thermal gradients in a confined area, or impact loads. These cracks are generally linear or slightly curved and do not intersect with other fissures. Their presence often suggests a limited structural issue, usually restricted to the immediate vicinity of the crack, and may not indicate a more generalized failure mechanism within the material or structure. In contrast, map cracks – also referred to as networked cracks, pattern cracking, or crazing – form an interconnected network of multiple cracks, typically resembling a map or a web. These cracks intersect at various angles and are distributed across a broader surface area. Map cracking is commonly associated with systemic or widespread structural issues, such as thermal incompatibility between material layers, alkali-silica reaction in concrete, plastic shrinkage in large surface areas, or excessive loading and load redistribution. This type of cracking is often an indicator of material degradation over time, pointing to a loss of cohesion or internal stress redistribution within the structural element.

https://doi.org/10.21528/lnlm-vol23-no2-art4



(a) (b)

Figure 2: Examples of crack types in structural elements: (a) Isolated crack – a single, distinct crack, and (b) map cracks – a network of interconnected cracks.

4 Deep learning model

We implemented ResNet-50 for crack surface classification. In contrast to conventional neural networks, ResNet addresses the vanishing gradient issue encountered as network depth increases by employing residual blocks [9]. These blocks enable the network to learn residual mappings, where gradients are preserved and allow for effective training across deep layers. The architecture of ResNet can be formulated as

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\}) + \mathbf{x}, \tag{1}$$

where $\mathbf{x}$ is the input to the residual block, $\mathbf{y}$ is the output, $\mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\})$ represents the residual mapping to be learned through stacked layers with weights $\mathbf{W}_i$, and the addition operation $\mathbf{x} + \mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\})$ denotes the residual connection [13].

In the output layers, we set the sigmoid activation function $\sigma(\mathbf{x})$, which is defined by equation (2):

$$\sigma(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}}}. \tag{2}$$

The sigmoid function is frequently utilized in binary classification tasks. It compresses the neural network's output to a range between 0 and 1, allowing the output to be interpreted as the probability of belonging to either the positive or negative class.

Deep learning models require a loss function to evaluate their performance during training. Given that the problem addressed in this work involves binary classification (1 or 0, crack or no crack), the binary cross-entropy loss function was chosen. Therefore, the loss function is given by:

$$\mathcal{L}(p, q) = -\frac{1}{N}\left[\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(q(y_i))\right], \tag{3}$$

where y_i represents the label (0 indicating the presence of a crack and 1 indicating the absence of a crack). The term $p(y_i)$ denotes the predicted probability of the absence of a crack, while $q(y_i) = 1 - p(y_i)$ signifies the predicted probability of the presence of a crack, for each of the N samples.

The proposed model utilizes transfer learning with the ResNet50 architecture pre-trained on ImageNet, serving as a fixed feature extractor by freezing its convolutional layers. A custom classification head is appended, consisting of a dense layer with 1024 ReLU-activated units, a dropout layer with a rate of 0.5 to mitigate overfitting, and a final dense layer with two units. We employed the RMSprop (Root Mean Square Propagation) algorithm to compute the gradient of the loss function. The initial learning rate was set to 0.0001. To optimize the model's performance, we experimented with different values for this learning rate and other hyperparameters, such as activation functions. We tested learning rates of 0.01, 0.001, and 0.0001, with 0.0001 yielding the best performance. Additionally, we evaluated different activation functions, ReLU, Tanh, and Sigmoid, in the dense layers. The optimal configuration used ReLU activation for the hidden dense layer and Sigmoid activation for the output layer. We manually fine-tuned these hyperparameters based on the model's performance on the validation set. This manual tuning process involved iterating over different configurations to find the best setup that minimizes the loss function while ensuring the model's generalization to unseen data.

4.1 Segmentation

We applied the Segment Anything Model (SAM), as proposed by [33], to the task of crack segmentation. SAM is specifically designed to effectively segment a diverse range of objects with varying shapes and sizes. It uses a dynamic masking algorithm that adjusts the receptive fields of convolutional filters based on object boundaries [33].

https://doi.org/10.21528/lnlm-vol23-no2-art4

The prediction of segmentation masks is supervised using a combination of focal loss and Dice loss. The focal loss is defined as:

$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t), \tag{4}$$

where $p_t \in [0, 1]$ is the model's estimated probability for the true class, α_t is a weighting factor for class imbalance, and γ is a focusing parameter that reduces the loss contribution from easy examples while emphasizing hard ones [36].

The Dice loss is given by:

$$\mathrm{DL} = 1 - \frac{2 \sum_{i=1}^{N} p_i \cdot g_i}{\sum_{i=1}^{N} p_i^2 + \sum_{i=1}^{N} g_i^2}, \tag{5}$$

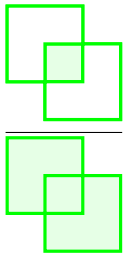
where p_i and g_i are scalar values representing the predicted and ground truth intensities for pixel i, and N is the total number of pixels in the image. In the case of multi-channel images, each pixel i may include one value per channel, and the loss is computed per channel and then aggregated.

To evaluate the segmentation performance of the SAM model, we utilized the Intersection over Union (IoU) metric, defined as:

$$\mathrm{IoU} = \frac{|\mathbf{P} \cap \mathbf{G}|}{|\mathbf{P} \cup \mathbf{G}|}, \tag{6}$$

where $\mathbf{P}$ is the set of pixels in the predicted mask, $\mathbf{G}$ is the set of pixels in the ground truth mask, $|\mathbf{P} \cap \mathbf{G}|$ denotes the number of pixels in their intersection, and $|\mathbf{P} \cup \mathbf{G}|$ denotes the number of pixels in their union. A clear and intuitive representation of the mathematical concept can be given by

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{}{}, \tag{7}$$



where the overlap area shows the common area between rectangles, and the union area includes the entire space occupied by both shapes.

We implemented the SAM model combined with the ResNet network with 50 layers as proposed in our previous work [9]. Figure 3 shows the SAM-ResNet50 model. The process begins with the input image. This image is passed through the image encoder, which extracts meaningful features. The encoder generates a representation of the input in the form of a high-dimensional feature map. The embeddings are then processed using a convolutional layer. This operation refines the feature map, highlighting patterns relevant for crack detection, such as edges and textures. The convolutional operation reduces spatial dimensions while preserving essential features, making the embeddings more compact and information-rich. The prompt encoder encodes additional contextual information (referred to as prompts) to guide the network in interpreting the feature map. The mask decoder generates segmentation masks, predicting the exact regions in the image that correspond to cracks. The masks are color-coded to represent different segmentation outputs, likely indicating probabilities or categories of detected regions. The masks are evaluated, and scores are computed, which quantify the confidence or quality of the segmentation. These scores ensure that only the most accurate predictions are retained. The segmentation output is then passed through the ResNet50.

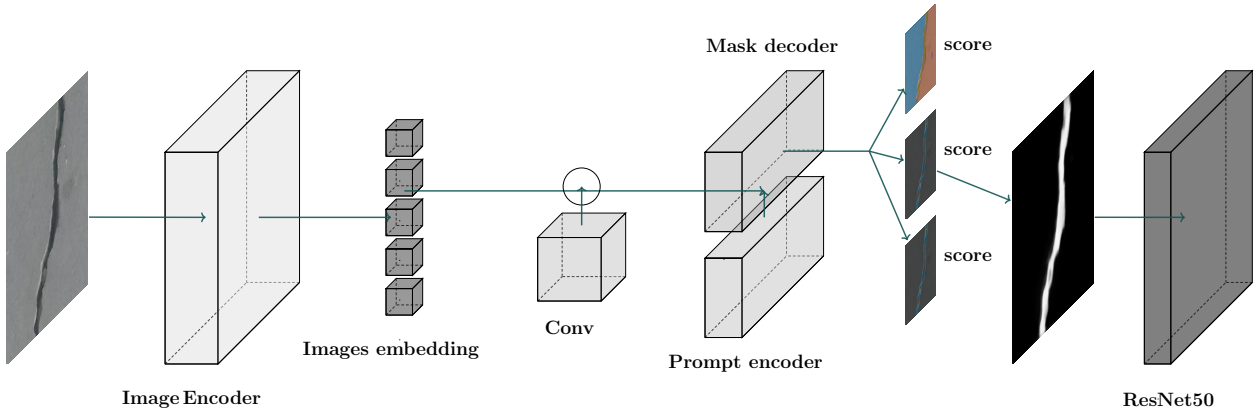


Figure 3: SAM-ResNet classifier model applied to crack detection.

Although SAM is designed to work with various types of prompts – such as points, bounding boxes, and text descriptions – to guide segmentation, in this work, we did not utilize any prompt-based inputs. Instead, we leveraged a version of SAM fine-tuned

https://doi.org/10.21528/lnlm-vol23-no2-art4

for automatic crack segmentation. The model was directly applied to the input images without the use of manual or interactive prompts, allowing it to autonomously identify and segment crack regions based on learned features.

The SAM was configured using the high-capacity Vision Transformer variant with a pretrained checkpoint. An instance of the automatic mask generator was created to enable automatic segmentation of cracks within images. The generator was fine-tuned with parameters such as a 32×32 grid of sampling points and a stability score threshold (0.92) to ensure high-quality and reliable masks. Additionally, settings like minimal region area filtering and one layer of cropping refinement were applied to balance accuracy and performance.

5 Crack characterization

Figure 4 shows the proposed procedure for crack characterization. After obtaining the segmented input image, the algorithm performs the following tasks: numbering the cracks; evaluating their slopes; determining the relative positions of the cracks; and checking whether the cracks are isolated or of the map type. These extracted features serve as the basis for engineers and civil construction professionals to determine the causes and propose measures to prevent the cracks.

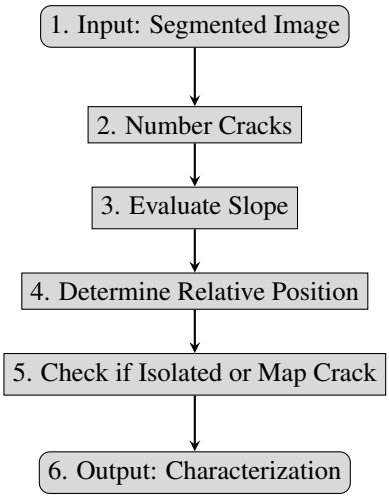


Figure 4: Proposed crack characterization algorithm. The input is the segmented image obtained from the SAM model.

Each step performed by the algorithm (Figure 4) is relevant to the crack characterization process. Step 2, numbering the cracks, is crucial to identify how many cracks are present and to enable assigning individual features to each one, such as slope, relative location, and type.

Step 3, evaluating the slope of the cracks, helps to indicate potential causes. For example, vertical cracks in the tension zone of a beam may suggest reinforcement issues. The slope of a crack is determined based on the line equation that passes through two extreme points (minimum and maximum) of the crack. The line is given by

$$y = m \cdot x + b, \tag{8}$$

where m is the slope and b is the y-intercept. The slope m is computed by

$$m = \frac{y_2 - y_1}{x_2 - x_1}, \tag{9}$$

and the angle of the crack with respect to the x-axis is calculated as

$$\text{angle} = \arctan(m). \tag{10}$$

This angle helps determine the orientation of the crack. Figure 5 illustrates this process by showing the two points used in the calculations.

In Step 4, determining the relative position of the cracks also contributes to diagnosing their causes. For example, diagonal cracks near the corners of wall openings often have different origins compared to those occurring centrally in the wall. Additionally, the spacing between cracks can reveal material behavior and load history [37]. Finally, in Step 5, checking whether the crack propagates in an isolated manner or forms a network (map crack) also supports cause identification. Isolated cracks usually indicate localized stress concentrations, while map cracks suggest systemic structural issues such as widespread degradation, thermal effects, or significant load redistribution [38].

Note that the problem remains binary – identifying whether a crack is present or not. Once a positive crack detection has occurred, the characterization stage is executed to analyze the properties of the detected crack. At this point, the algorithm assesses whether each detected crack is isolated or part of a network (map crack). This distinction is made after segmentation.

https://doi.org/10.21528/lnlm-vol23-no2-art4

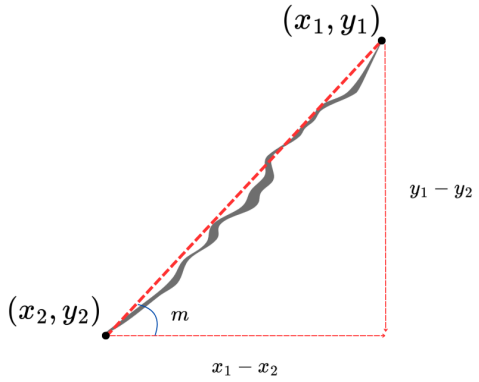


Figure 5: Illustration of crack slope calculation using two extreme points (x_1, y_1) and (x_2, y_2).

6 Results and discussion

In this section, the results of the dataset studies are presented and discussed. The results are divided into two sections: 6.1 and 6.2. Section 6.1 presents the detection and segmentation of cracks, detailing how the model identifies and isolates crack regions within the input images. Section 6.2 focuses on crack characterization, describing the analysis performed on the detected cracks to assess their properties, such as orientation.

6.1 Detection and segmentation

The model training process was accelerated using a NVIDIA GeForce GTX 1660 Ti GPU with 30 GiB of RAM, running on an Ubuntu 22.04.3 LTS x86_64 operating system. TensorFlow 2.13.0 was employed as the primary deep learning framework for building and training neural networks. For image processing and augmentation tasks, OpenCV-Python 4.8.1.78 was used, offering extensive functionality for manipulating and preparing visual data. The model was trained for 30 epochs with a batch size of 32. In terms of computational complexity, ResNet50 can be approximated as $O(L \cdot n \cdot k^2 \cdot c)$, where n is the number of input pixels, $L = 50$ is the network depth, k is the kernel size, and c is the number of channels. For an input resolution of 96×96, the forward pass requires approximately 0.73 GFLOPs per image.

Table 3 presents the metrics of the SAM-ResNet model for crack detection on four different datasets: bricks, concrete, steel, and wood. The metrics considered are accuracy, precision, recall, and F1-score. For both the bricks and concrete datasets, the SAM-ResNet model achieved a score of 1.0000 for all metrics. These results suggest that the features of cracks in bricks and concrete are well captured by the model. In contrast, the performance of the model drops for the steel dataset, with slightly lower recall (0.9047), although precision remains high (1.0000), indicating that some cracks are missed (false negatives). The performance is significantly lower for the wood dataset, with an accuracy of 0.6071. This drop in performance can be attributed to the inherent visual complexity and texture of wood surfaces. Unlike bricks or steel, wood often presents natural grain patterns, knots, and irregularities that can resemble cracks or obscure them. These characteristics introduce noise that challenges the model's ability to generalize, especially in distinguishing real cracks from background texture. The high precision in this case shows that the model is confident when predicting a crack, but the low recall highlights that it misses a large number of actual cracks, leading to many false negatives.

Table 3: Model metrics for crack detection.

Dataset	Model	Accuracy	Precision	Recall	F1-score
Bricks [33]	SAM-ResNet	1.0000	1.0000	1.0000	1.0000
Concrete [26]	SAM-ResNet	1.0000	1.0000	1.0000	1.0000
Steel [34]	SAM-ResNet	0.9047	1.0000	0.9047	0.9500
Wood [35]	SAM-ResNet	0.6071	1.0000	0.6071	0.7555

Figures 6, 7, and 8 depict the segmentation results for bricks, steel, and wood datasets, respectively. The SAM-ResNet model achieves a precision of 1 and an $IoU > 0.9$ on all datasets, which is valid. This suggests that the model is good in confirming that predicted cracks are indeed cracks (no false positives). Nevertheless, the recall varies across datasets, suggesting that the model is sensible to detecting all actual cracks varies with the material. The variance could be due to the different surface characteristics and crack patterns inherent to each material. For applications requiring high confidence in crack detection (i.e., minimizing false alarms), the SAM-ResNet model performs well.

https://doi.org/10.21528/lnlm-vol23-no2-art4

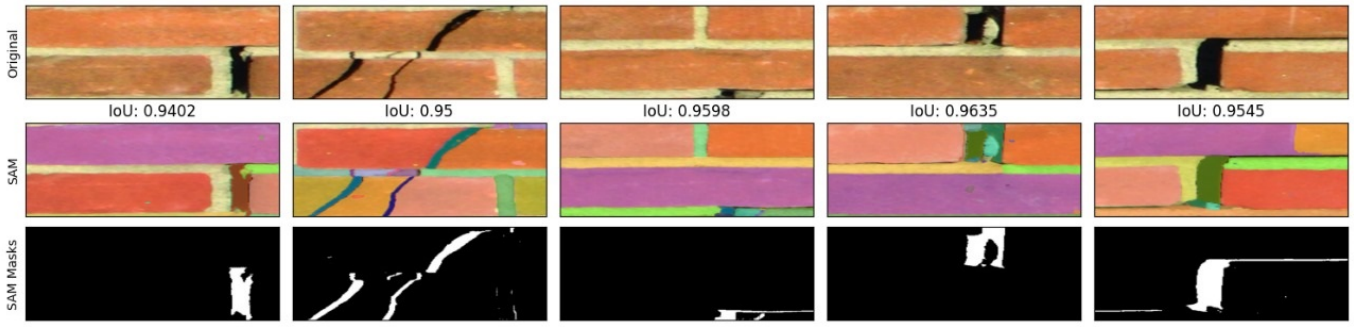


Figure 6: Crack bricks segmentation outcomes utilizing SAM model.

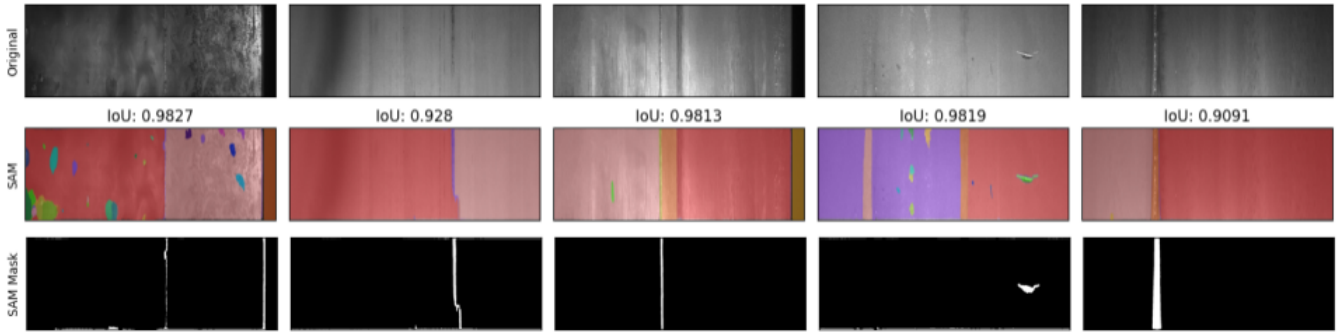


Figure 7: Crack steel segmentation outcomes utilizing SAM model.

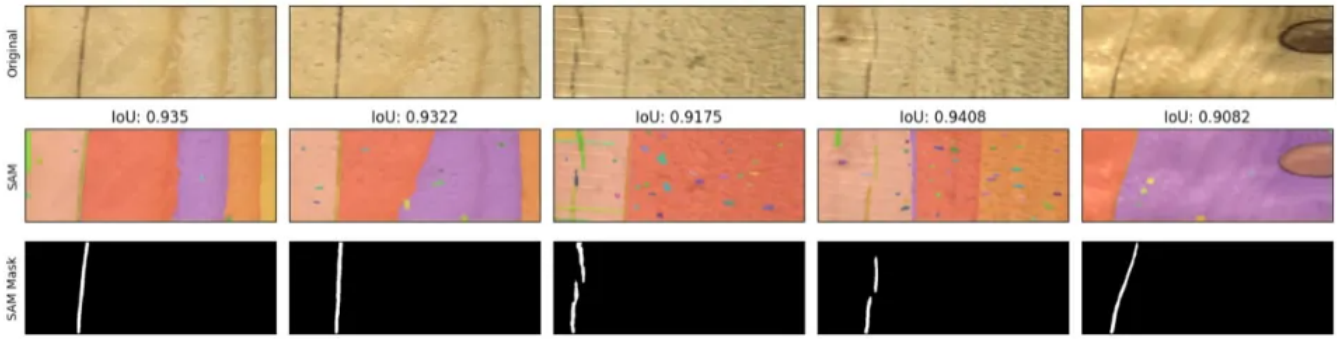


Figure 8: Crack wood segmentation outcomes utilizing SAM model.

6.2 Crack characterization

Table 4 presents model metrics for detecting isolated and map cracks using the ResNet-50 model on the bricks dataset. In this section, we applied only the bricks dataset because it contains instances of both isolated and map cracks. While the model achieves high accuracy (0.9994) and perfect precision (1.0000) for identifying isolated cracks, it exhibits lower recall (0.7500) for map cracks. This imbalance results in a lower F1-score (0.8571) for map cracks compared to isolated ones. These results indicate that the model performs well in correctly identifying isolated cracks but struggles to detect all instances of map cracks, leading to a higher number of false negatives in this category.

Table 4: Model metrics for detecting isolated or map crack.

Dataset	Model	Accuracy	Precision	Recall	F1-score
Bricks [33]	ResNet	0.9994	1.0000	0.7500	0.8571

6.2.1 Isolated cracks

Figures 9 (a)-(d) present examples of isolated cracks. We applied the crack characterization procedure proposed in section 4 to obtain the results in Figures 9 (e)-(h). Figures 9 (a) and (e), respectively original and segmented image, show a crack that was classified as isolated, with a 60.71° slope, located at the top right corner of a wall opening. The type, slope, and relative position

of the crack suggest an issue related to the lintel size of the opening. Figure 9 (b) and Figure 9 (f) show an image of a crack in a street light pole. Despite the craze pattern (several micro map cracks), only an isolated crack fully reached the outer surface and was then detected by the algorithm. The crack is, therefore, isolated, has a slope of $5.70°$, and occurs in the central part of the pole lateral side. These features may indicate shrinkage (due to the crack pattern), combined with issues related to the pole concrete reinforcement or thermal action.

Figure 9 (c) displays a single crack, but due to the angle the picture was taken from, its segmented format in Figure 9 (g) shows 4 distinct lines. However, the algorithm correctly classified that crack as isolated, according to specific criteria that take into account that all the lines have approximately the same slope and are close to each other. The crack is, then, isolated, has a mean slope of $2.38°$, and occurs along the mortar around the bricks. The causes may be settlement or thermal action. The crack in Figure 9 (d) and Figure 9 (h) is isolated, has a slope of $-84.96°$ (clockwise, then negative), and is located in the center of the image, which depicts a specific part of the floor. The most probable cause of the crack is thermal action.

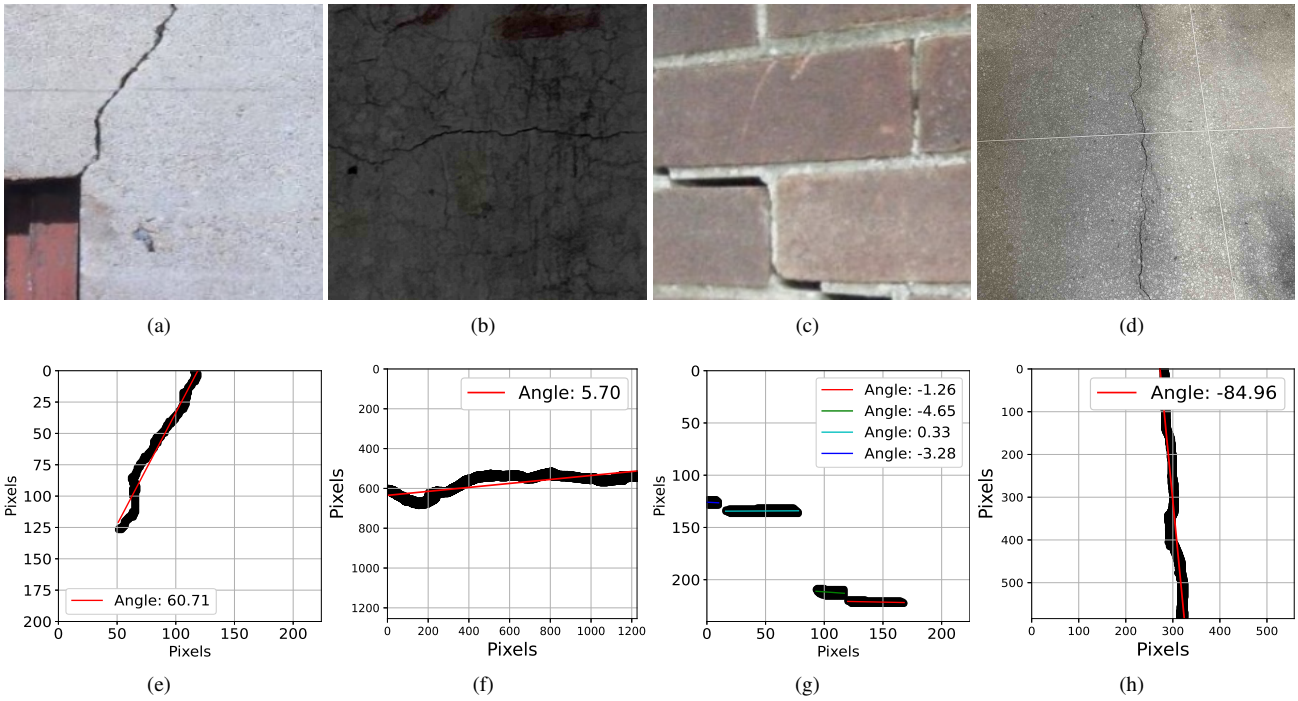


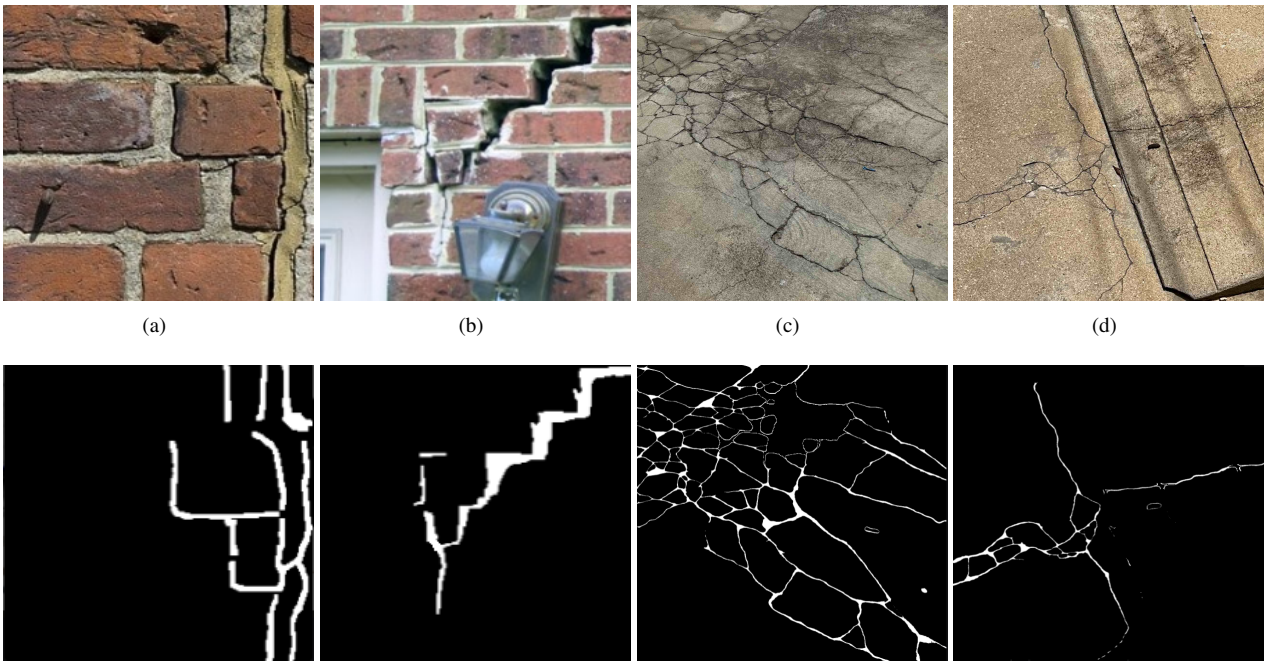
Figure 9: Example of isolated cracks: (a) crack in the corner of the door, (b) street light pole crack, (c) crack in the brick, (d) floor crack, (e) crack in the corner of the door from (a), (f) street light pole crack from (b), (g) crack in the brick wall of a house from (c), and (h) floor crack from Figure (d) .

6.2.2 Map cracks

Figures 10 (a)-(d) depict examples of map cracks. We applied the proposed crack characterization procedure and achieved the results presented in Figures 10 (e)-(h). For map cracks, we will not explicitly consider the slopes, since each single branch crack has its own slope.

In Figure 10 (a) and Figure 10 (e), respectively original and segmented format, we observe map cracks along the mortar joints. These cracks may be caused by differential settlement, thermal action, or moisture movement. The crack in Figure 10 (b) is mostly isolated, but it becomes mapped at its lower portion; this can be seen in more detail in Figure 10 (f). We consider it a map crack, since it has a mapped part; however, for further analysis, the crack may be split into two parts, an isolated and a map one. Furthermore, its relative position is along the mortar joints. Such cracks may occur due to foundation settlement or thermal action. Figure 10 (c) and Figure 10 (g) show map cracks, located in the central part of the picture (the studied area). The cracks were probably caused by concrete shrinkage along with thermal action. Figure 10 (d) and 10 (h) depict map cracks in a concrete floor, located next to a ramp. Some of the cracks are parallel to the ramp edge line, suggesting thermal action; other cracks are perpendicular to the ramp line and also to the walk pavement, suggesting thermal action in another direction. Because there are multiple single hairline cracks, concrete shrinkage may have also caused the map cracks.

https://doi.org/10.21528/lnlm-vol23-no2-art4



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 10: Example of map cracks: (a) horizontal and vertical cracks in the brick, (b) map stair-step cracks in the corner of the door, (c) map of cracks in the floor, (d) floor cracks, (e) segmented image from (a), (f) segmented image from (b), (g) segmented image from (c), and (h) segmented image from (d).

7 Conclusions

We used residual neural networks and image segmentation to detect and characterize cracks on building elements. The model achieved good accuracy, with an IoU score of 0.95, and was able to accurately detect, outline, and classify cracks according to the proposed criteria. Therefore, the model presented real potential for effective crack detection in professional applications. Although the algorithm currently has limited classification features, future versions will include additional ones, such as determining crack width, to provide a basis for identifying the causes and further correcting cracks. Moreover, in applications involving structural safety, minimizing false negatives is typically more critical than reducing false positives, since undetected cracks may compromise structural integrity. While the high precision across all datasets ensures minimal false alarms, the recall – especially in the wood dataset – indicates that the model still misses some cracks. Therefore, approaches to further reduce false negatives, including multi-stage verification strategies, will be considered in future work.

Acknowledgments. To the Federal University of Semi-Arid Region for financial support (EDITAL PROPPG 15/2023), and to the laboratories CILAB and LOADS at UFERSA for all the support.

REFERENCES

[1] N. Karimi, M. Mishra and P. B. Lourenço. "Automated surface crack detection in historical constructions with various materials using deep learning-based YOLO network". *International Journal of Architectural Heritage*, pp. 1–17, 2024.

[2] N.-D. Hoang. "Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding". *Advances in Civil Engineering*, vol. 2018, no. 1, pp. 3924120, 2018.

[3] A. Saberironaghi and J. Ren. "DepthCrackNet: A Deep Learning Model for Automatic Pavement Crack Detection". *Journal of Imaging*, vol. 10, no. 5, pp. 100, 2024.

[4] H. C. Dantas, L. M. Morais, P. H. Bezerra and R. C. Rego. "Concrete Crack Detection Using Embedded Machine Learning". In *2024 8th International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT)*, pp. 1–6. IEEE, 2024.

[5] X. Wu and X. Liu. "Building crack identification and total quality management method based on deep learning". *Pattern Recognition Letters*, vol. 145, pp. 225–231, 2021.

[6] F. Hu, H. Gou, H. Yang, H. Yan, Y.-q. Ni and Y.-w. Wang. "Automatic PAUT Crack Detection and Depth Identification Framework Based on Inspection Robot and Deep Learning Method". *Journal of Infrastructure Intelligence and Resilience*, p. 100113, 2024.

[7] J. Michael, S. H. Smith, S. A. Durham and M. G. Chorzepa. "Crack control in concrete walls through novel mixture design, full-scale testing, and finite element analysis". *Construction and Building Materials*, vol. 166, pp. 301–314, 2018.

[8] P. H. A. Bezerra, H. C. Dantas, L. M. G. Morais and R. C. B. Rego. "A Deep Learning Artificial Intelligence Algorithm to Detect Cracks on Civil Engineering Building Elements". In *XX International Conference on Building Pathology and Constructions Repair*, volume 1. XX International Conference on Building Pathology and Constructions Repair, 2024.

[9] R. C. B. Rego, H. C. Dantas, L. M. G. Morais and P. H. A. Bezerra. "Civil infrastructure crack detection using deep learning and image segmentation-based techniques". In *Proceedings of the XXV Congresso Brasileiro de Automática*, Rio de Janeiro, Brazil, October 2024.

[10] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition". In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015.

[11] Y. Liu, J. Yao, X. Lu, R. Xie and L. Li. "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation". *Neurocomputing*, vol. 338, pp. 139–153, 2019.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich. "Going deeper with convolutions". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[13] K. He, X. Zhang, S. Ren and J. Sun. "Deep residual learning for image recognition". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[14] L. Ali, F. Alnajjar, W. Khan, M. A. Serhani and H. Al Jassmi. "Bibliometric analysis and review of deep learning-based crack detection literature published between 2010 and 2022". *Buildings*, vol. 12, no. 4, pp. 432, 2022.

[15] M. Tan and Q. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

[16] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger. "Densely connected convolutional networks". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[17] Y. Fei, K. C. Wang, A. Zhang, C. Chen, J. Q. Li, Y. Liu, G. Yang and B. Li. "Pixel-level cracking detection on 3D asphalt pavement images through deep-learning-based CrackNet-V". *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 273–284, 2019.

[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally and K. Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size". *arXiv preprint arXiv:1602.07360*, 2016.

[19] Q. Yang, W. Shi, J. Chen and W. Lin. "Deep convolution neural network-based transfer learning method for civil infrastructure crack detection". *Automation in Construction*, vol. 116, pp. 103199, 2020.

[20] D. Tabernik, S. Šela, J. Skvarč and D. Skočaj. "Segmentation-based deep-learning approach for surface-defect detection". *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 759–776, 2020.

[21] H. Bae, K. Jang and Y.-K. An. "Deep super resolution crack network (SrcNet) for improving computer vision–based automated crack detectability in in situ bridges". *Structural Health Monitoring*, vol. 20, no. 4, pp. 1428–1442, 2021.

[22] W. Cao, J. Li, X. Zhang, F. Kang and X. Wu. "Sonar combines deep learning and building information modeling for underwater crack detection of concrete structures". In *Structures*, volume 70, p. 107834. Elsevier, 2024.

[23] J. Wang, T. Ueda, P. Wang, Z. Li and Y. Li. "Building damage inspection method using uav-based data acquisition and deep learning-based crack detection". *Journal of Civil Structural Health Monitoring*, vol. 15, no. 1, pp. 151–171, 2025.

[24] J. Wang, P. Wang, L. Qu, Z. Pei and T. Ueda. "Automatic detection of building surface cracks using UAV and deep learning-combined approach". *Structural Concrete*, vol. 25, no. 4, pp. 2302–2322, 2024.

[25] M. Interlando, M. G. Pacifico, A. Novellino and V. P. Pastore. "Ensembles of deep neural networks for the automatic detection of building facade defects from images". *IEEE Access*, 2024.

[26] Çağlar Fırat Özgenel. "Concrete crack images for classification", 2019.

[27] M. Maguire, S. Dorafshan and R. J. Thomas. "SDNET2018: A concrete crack image dataset for machine learning applications". 2018.

[28] H. Xu, X. Su, Y. Wang, H. Cai, K. Cui and X. Chen. "Automatic bridge crack detection using a convolutional neural network". *Applied Sciences*, vol. 9, no. 14, pp. 2867, 2019.

[29] E. Agustsson and R. Timofte. "Ntire 2017 challenge on single image super-resolution: Dataset and study". In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 126–135, 2017.

[30] Y. Shi, L. Cui, Z. Qi, F. Meng and Z. Chen. "Automatic road crack detection using random structured forests". *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016.

[31] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei and H. Ling. "Feature pyramid and hierarchical boosting network for pavement crack detection". *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1525–1535, 2019.

[32] Q. Zou, Y. Cao, Q. Li, Q. Mao and S. Wang. "CrackTree: Automatic crack detection from pavement images". *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.

[33] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*. "Segment anything". In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

[34] A. Grishin and BorisV. "Severstal: Steel Defect Detection". `https://www.kaggle.com/competitions/severstal-steel-defect-detection`, 2019. Accessed: 2024-12-02.

[35] P. Kodytek, A. Bodzas and P. Bilik. "A large-scale image dataset of wood surface defects for automated vision-based quality control processes". *F1000Research*, vol. 10, 2021.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár. "Focal loss for dense object detection". In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[37] S. Das, S. Dutta, D. Adak and S. Majumdar. "On the crack characterization of reinforced concrete structures: Experimental and data-driven numerical study". In *Structures*, volume 30, pp. 134–145. Elsevier, 2021.

[38] M. Mishra, P. B. Lourenço and G. V. Ramana. "Structural health monitoring of civil engineering structures by using the internet of things: A review". *Journal of Building Engineering*, vol. 48, pp. 103954, 2022.