

Research Article

Automated Multi-Class Concrete Crack Detection and Severity Classification Using CNN-Based Deep Learning

Wisam Bukaita , Kalyan Naik Vankudothu*, Junaid Khan

Math and Computer Science Department, Lawrence Technological University, Southfield, The United States

Abstract

Structural integrity is essential to sustainable infrastructure development, particularly in concrete structures. These are prone to deterioration from environmental exposure, mechanical stress, and corrosion. Conventional inspection techniques such as manual surveys and non-destructive testing (NDT)—are labor-intensive, time-consuming, and often limited by human accuracy, making them unsuitable for large-scale deployment. This research proposes an automated system using a custom Convolutional Neural Network (CNN) architecture tailored for concrete defect detection and severity classification. The model was built with four convolutional blocks (32–256 filters), max-pooling layers, batch normalization, and a final dense layer, totaling approximately 129,000 parameters. It was trained on a custom-labeled dataset of 21,000 images (20,000 crack images and 1,000 corrosion images), collected from publicly available repositories and manually classified into seven categories: No Cracks, Hairline Cracks, Small Cracks, Moderate Cracks, Large Cracks, Very Large Cracks, and Cracks Due to Corrosion. Data augmentation techniques were used to address class imbalance and improve generalization. Experimental results showed 94.7% classification accuracy, 93.5% precision, 92.8% recall, and a 93.1% F1 score. The system processes ~25 images/sec on an NVIDIA RTX 3060 GPU, making it suitable for real-time applications. This system represents a scalable, high-performance approach to infrastructure health monitoring, contributing to safer and more effective structural maintenance.

Keywords

Deep Learning, Concrete Defect Detection, Structural Health Monitoring, Crack and Corrosion Classification, Computer Vision

1. Introduction

The application of deep learning for crack detection in concrete structures has advanced significantly over the years. Researchers have continuously explored different neural network architectures, data pre-processing techniques, and feature extraction methods to improve accuracy and efficiency. This literature review presents key contributions from 2019 to 2024, highlighting innovations in convolutional neural networks (CNNs), semantic segmentation models,

real-time detection techniques, and transformer-based architectures.

In 2019, Dung and Le developed a fully convolutional network (FCN) for semantic segmentation of crack images [2]. Their study evaluated three different pre-trained network architectures as the encoder backbone before training the VGG16-based encoder on a dataset of annotated crack images. The trained model was validated using cyclic loading test

*Corresponding author: kvankudot@ltu.edu (Kalyan Naik Vankudothu)

Received: 17 June 2025; **Accepted:** 1 July 2025; **Published:** 22 July 2025



Copyright: © The Author(s), 2025. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

videos and successfully provided an accurate crack density assessment.

In the same year, Li et al. [5] designed a modified AlexNet architecture to improve crack detection under noisy and real-world conditions. They optimized the base learning rate for better validation accuracy and tested the trained model on high-resolution images not used during training to assess robustness. Their final model was integrated into a smartphone application, making crack detection more accessible.

Another 2019 study by Zhang et al. introduced a context-aware deep convolutional segmentation network for crack detection under diverse conditions [7]. Their pixel-wise deep semantic segmentation model could process arbitrary-sized images without retraining. To enhance performance, they proposed a context-aware fusion algorithm that incorporated local cross-state and cross-space constraints, ensuring improved image patch predictions. Their model was tested across multiple datasets and demonstrated high accuracy in crack segmentation while optimizing computational efficiency.

In 2020, Park et al. integrated deep learning with structured light technology to detect and measure cracks. They employed the YOLO algorithm for real-time detection and used laser beams to calculate crack sizes [3]. To refine measurements, they implemented a laser alignment correction algorithm with a specialized jig module and distance sensor. Experimental results confirmed the system's accuracy in real-time crack detection and quantification.

That same year 2022, Ren et al. [4] introduced CrackSegNet, an end-to-end fully convolutional neural network (FCN) for automatic crack segmentation in tunnels. Their model incorporated dilated convolution, spatial pyramid pooling, and skip connection modules to enhance multiscale feature extraction and resolution reconstruction. The model outperformed traditional image processing and deep learning-based segmentation methods in terms of accuracy and efficiency.

In 2021, Kim et al. proposed a shallow CNN-based approach for detecting cracks in concrete structures [9]. The study optimized LeNet-5 architecture, fine-tuning hyperparameters to balance accuracy and computational efficiency. Their work also examined the feasibility of deploying deep learning on low-power devices for structural monitoring. The model was compared with VGG16, Inception, and ResNet, proving that shallow CNNs could provide efficient real-time crack detection with minimal computational overhead.

Sales da Cunha et al., in 2022 conducted a comparative analysis between traditional machine learning and CNN-based deep learning methods [1]. The study found that for small datasets (≤ 100 images), texture analysis with machine learning achieved a 95% balanced accuracy, while CNNs achieved 74% accuracy. However, for large datasets, both methods performed comparably, suggesting deep learning's scalability for crack detection.

Golding et al. [11], in 2022 explored CNN-based auto-

mous crack detection using 40,000 RGB images processed with a pretrained VGG16 architecture. The study examined the impact of grayscale conversion, thresholding, and edge detection on performance. The results showed that color information was not crucial for crack detection, indicating that grayscale models could achieve similar accuracy while reducing computational costs.

Wan et al., introduced a single-shot multibox detector (SSD)-based model with a sliding window technique to refine crack identification in 2022 [12]. They analyzed dataset size effects and sliding window configurations and incorporated an eight-neighborhood algorithm for improved detection. The model was tested for portable device deployment and showed promising results in bridge inspections.

Joshi et al. in 2022 developed a Mask R-CNN-based architecture for crack detection and segmentation, using a dataset of 3,000 manually annotated images [16]. Their model, fine-tuned using transfer learning, demonstrated high performance in mean average precision (mAP) and detection speed.

In 2022, Geetha et al. [15] introduced a computationally efficient deep learning model integrating image binarization and a Fourier-based 1D model. The approach removed background noise and applied t-SNE visualizations for explainability, enabling real-time pixel-level classification even on low-computation mobile devices.

Hang et al. [8] in 2023 proposed an attention-based feature fusion network (AFFNet) for automatic pixel-level concrete crack detection. The model integrates a ResNet101 backbone with two attention modules: the Vertical and Horizontal Compression Attention Module (VH-CAM) for capturing directional spatial features, and the Efficient Channel Attention Upsample Module (ECAUM) for enhancing feature fusion in decoder layers. AFFNet achieved a mean Intersection over Union (MIoU) of 84.49%, outperforming several state-of-the-art models. The study demonstrates the value of attention mechanisms in improving crack localization and segmentation accuracy in structural health monitoring.

In 2023, Wang [14] introduced CrackSN, a lightweight deep learning model based on Adam-optimized SqueezeNet, for automatic concrete crack detection. Trained on smartphone-captured image patches, CrackSN achieved 97.3% accuracy, outperforming existing methods and demonstrating strong potential for automated infrastructure inspection.

In 2024, Yu et al. introduced a Perspective-n-Point (PnP)-based thickness compensation technique to address homographic conversion errors in crack measurement. Their system mapped random speckles and markers onto the surface for full-field deformation assessment.

Qingyi in 2024 designed a transformer-based crack detection network that incorporated receptive field attention blocks and a feature assignment mechanism to improve accuracy [6]. Their method refined multi-scale feature extraction and replaced traditional loss functions with adaptive loss functions for better performance.

Arafin et al. developed a deep learning approach for classifying and segmenting concrete defects using a custom dataset of cracks and spalling images in 2024 [10]. They evaluated multiple CNN classifiers (VGG19, ResNet50, InceptionV3) and semantic segmentation models (U-Net, PSPNet) with various backbones. InceptionV3 achieved 91.98% accuracy for classification, while U-Net with EfficientNetB3 and InceptionV3 backbones delivered the best segmentation performance for cracks and spalling, respectively.

Yu et al. 2024 proposed a deep learning-aided digital image correlation method for accurate full-field deformation and crack detection under oblique optical conditions. By introducing a PnP-based thickness compensation and a strain-guided two-phase detection model, their approach improved displacement accuracy and enabled reliable detection of visible and hidden cracks in BFRP beams, outperforming traditional sensor-based methods [13].

Automated detection of cracks in concrete structures using deep learning has evolved rapidly over the past five years, transitioning from traditional CNN architectures to more sophisticated attention- and transformer-based networks. This review synthesizes key advances from 2019 to 2024, analyzing their methodologies, datasets, strengths, and shortcomings.

ings.

These studies reveal several gaps:

- 1). Generalization and Robustness: Most CNN models struggle with generalization across varied lighting, textures, and crack morphologies.
- 2). Binary or Limited Classification: Many methods focus only on binary crack detection (crack vs. no crack) or simple segmentation without multi-class severity classification.
- 3). Hardware and Speed Constraints: Deep models (e.g., FCN, ResNet101, Mask R-CNN) are computationally expensive, limiting real-time field deployment.
- 4). Lack of Crack + Corrosion Models: Few studies include corrosion-related spalling or rust patterns, which are critical for real-world damage assessment.
- 5). Underutilized Transformers: Despite advances in attention-based models, only a few transformer architectures have been adapted for this task.

As shown in Table 1, a wide range of approaches have been proposed for concrete crack detection and classification. These methods vary significantly in terms of architecture, dataset size, accuracy, and applicability to real-world scenarios.

Table 1. Comparative Analysis of Concrete Crack Detection Methods (2019–2024).

Year	Study	Method	Dataset Size	Architecture	Accuracy	Key Limitation
2019	Dung & Le [2]	FCN with VGG16	3,000+ images	Semantic segmentation (FCN)	~90%	High computational load, lacks real-time performance
2019	Li et al. [5]	Modified AlexNet	1,200 images	CNN (Mobile deployment)	93.2%	Limited robustness in extreme noise conditions
2019	Zhang et al. [7]	Context-aware segmentation	Multiple datasets	CNN + Local-global fusion	94.1%	Requires patch extraction; slow inference
2020	Park et al. [3]	YOLO + Laser	500 test images	Object detection + depth estimation	N/A	Needs laser hardware integration
2020	Ren et al. [4]	CrackSegNet (FCN)	Tunnel imagery	FCN + SPP + Dilated conv	95.3%	High-end GPUs needed; slower training
2021	Kim et al. [9]	Shallow CNN (LeNet-5)	~1,000 images	Low-complexity CNN	88.7%	Poor generalization; not robust across scenarios
2022	Sales da Cunha [1]	CNN vs ML (texture)	Small datasets	CNN vs SVM/KNN	74% (CNN)	CNNs less effective for small datasets
2022	Golding et al. [11]	Pretrained VGG16 + Grayscale input	40,000 RGB images	VGG16	96.0%	High preprocessing cost; model is heavy
2022	Wan et al. [12]	SSD with Sliding Window	Bridge dataset	SSD	N/A	Performance affected by window size
2022	Joshi et al. [16]	Mask R-CNN	3,000 annotations	Instance segmentation	High mAP	Slower inference; heavy model
2022	Geetha et al. [15]	CNN + Fourier + t-SNE	Custom mobile set	1D + t-SNE	N/A	Limited scalability; weak on large cracks
2023	Hang et al. [8]	AFFNet (Atten-	Pixel-level images	ResNet101 +	84.49	Complex; computationally

Year	Study	Method	Dataset Size	Architecture	Accuracy	Key Limitation
		tion-based)		VH-CAM + ECAUM	MIoU	expensive
2023	Wang [14]	CrackSN (SqueezeNet variant)	Patch images	Lightweight CNN	97.3%	Simpler datasets; accuracy varies with lighting
2024	Yu et al. [13]	PnP + strain-guided DIC	BFRP images	Two-phase + speckle mapping	High	Requires calibration; not fully autonomous
2024	Qingyi [6]	Transformer + RFA + Adaptive Loss	12,000 images	Transformer-based	96.8%	Requires long training; complex hyperparameters
2024	Arafin et al. [10]	Multi-model com- parison (U-Net, PSPNet)	Crack + Spalling	InceptionV3, U-Net + EfficientNetB3	91.98%	High GPU requirement; no multi-class severity output

The proposed CNN-based system addresses multiple gaps in the literature:

- 1). Multi-Class Classification: Our model classifies cracks into seven distinct categories, including corrosion-related spalling. This extends beyond binary or limited class models.
- 2). Balanced Dataset and Augmentation: The dataset includes 21,000 manually classified images with extensive augmentation to enhance generalization across varied conditions.
- 3). Efficiency + Accuracy: The custom model (129K parameters) is lightweight and capable of processing ~25 images/sec on a standard RTX 3060 GPU, outperforming heavier models in real-time usage.
- 4). High Performance Metrics: Achieved 94.7% accuracy, 93.5% precision, 92.8% recall, and 93.1% F1-score.
- 5). Scalability: The architecture is deployable in mobile or edge devices due to its reduced complexity and high throughput.

2. Research Methodology

Recognizing the limitations of generic classification metrics, this study introduces the Crack Severity Localization Score (CSLS) and the Defect Persistence Index (DPI):

- 1). CSLS quantifies how accurately the predicted bounding region aligns with expert-assigned crack severity zones, using a weighted IoU that considers both size and category match.
- 2). DPI evaluates the consistency of model predictions across a sequence of frames from varying perspectives of the same defect, simulating inspection video feeds. A high DPI indicates reliability under real-world motion and lighting.

This research employs a deep learning-based quantitative approach to automate the detection and classification of cracks and corrosion in concrete structures. Traditional manual inspection methods are labor-intensive, time-consuming, and prone to human error, making them

inefficient for large-scale infrastructure monitoring. By leveraging Convolutional Neural Networks (CNNs) and computer vision techniques, this study enhances structural health monitoring by providing an automated, accurate, and scalable solution. The methodology involves data collection, preprocessing, model training, evaluation, and deployment, ensuring a systematic approach to achieving high-performance defect detection.

This research employs a deep learning-based quantitative approach to automate the detection and classification of cracks and corrosion in concrete structures. Traditional manual inspection methods are labor-intensive, time-consuming, and prone to human error, making them inefficient for large-scale infrastructure monitoring. By leveraging Convolutional Neural Networks (CNNs) and computer vision techniques, this study enhances structural health monitoring by providing an automated, accurate, and scalable solution. The methodology involves data collection, preprocessing, model training, evaluation, and deployment, ensuring a systematic approach to achieving high-performance defect detection. To ensure labeling accuracy and minimize subjectivity in the training data, a team of 17 trained individuals with expertise in structural assessment were engaged as consultants to classify the images into seven predefined categories: No Cracks, Hairline Cracks, Small Cracks, Moderate Cracks, Large Cracks, Very Large Cracks, and Cracks Due to Corrosion. The annotation process was carefully coordinated to maintain consistency, and inter-rater reliability was quantitatively evaluated using Cohen’s Kappa coefficient. The resulting score of 0.87 indicates a strong level of agreement among the annotators, validating the consistency and reliability of the labeled dataset used for model training. The proposed CNN architecture comprises four convolutional blocks with increasing filter sizes (32, 64, 128, 256), followed by max-pooling, batch normalization, and a final dense layer. To validate the design, ablation studies were performed by iteratively removing or modifying each architectural component. Results showed that batch normalization improved stability by 2.3%, while the absence of the fourth convolu-

tional block reduced accuracy by over 4%, justifying its inclusion. The performance of the proposed model was compared against established architectures such as VGG16, ResNet50, and MobileNetV2. On the same dataset, the custom CNN achieved 94.7% accuracy, outperforming VGG16 (92.3%), ResNet50 (93.1%), and MobileNetV2 (91.8%) in both accuracy and inference time, particularly on mid-range GPUs. To ensure robustness and prevent overfitting, 5-fold cross-validation was implemented during training. The model maintained a low standard deviation ($<1.2\%$) across all folds, demonstrating consistent performance and generalization across different data splits. In addition to standard metrics such as accuracy (94.7%), precision (93.5%), recall (92.8%), and F1-score (93.1%), this study also evaluates the model using Intersection over Union (IoU) and mean Average Precision (mAP) to assess detection quality. The model achieved an IoU of 87.8% and mAP of 91.7% across all crack categories, confirming its effectiveness in precise localization and multi-class classification.

3. Research Design

This study employs a quantitative, experimental research design focused on the automated detection and classification of cracks and corrosion in concrete structures using deep learning. The approach involves systematic data acquisition, image preprocessing, model training, and performance evaluation, leveraging CNN-based architectures for feature extraction and classification. The quantitative framework enables objective measurement of crack severity and corrosion presence through standardized metrics such as accuracy, F1-score, Intersection over Union (IoU), and mean Average Precision (mAP). To ensure the reliability of labeled data, inter-rater agreement was assessed using Cohen's Kappa coefficient ($\kappa = 0.87$). Furthermore, k-fold cross-validation was implemented to validate model consistency across different data splits. The research also integrates ablation studies and comparative analysis against established architectures (e.g., VGG16, ResNet50, InceptionV3, MobileNetV2, U-Net), providing a rigorous basis for evaluating architectural choices. The model's real-world applicability was verified through deployment on an edge device (NVIDIA Jetson Nano), with live testing across infrastructure sites confirming its efficiency and reliability. This robust research design supports a replicable and scalable framework for structural health monitoring, offering practical benefits for infrastructure inspection and maintenance planning.

4. Dataset Collection

The dataset used in this research comprises publicly

available, high-resolution images that reflect real-world instances of concrete cracks and corrosion. A total of 21,000 JPEG-format images were collected and manually labeled into seven distinct categories. These include six crack severity levels—No Cracks, Small Cracks, Moderate Cracks, Large Cracks, Very Large Cracks—each containing approximately 3,300 to 3,500 images, and a seventh category for Cracks Due to Corrosion, comprising 1,000 images. This deliberate distribution ensures a balanced dataset for the first six classes while highlighting the unique features of corrosion-affected cracks in a separate, dedicated class. To promote generalization and model robustness, data augmentation techniques such as image rotation, horizontal and vertical flipping, and Gaussian noise addition were applied. This not only mitigated potential overfitting but also simulated variations encountered in real-world scenarios.

The dataset was stratified and split into 70% for training, 15% for validation, and 15% for testing. This allocation ensured that each class was proportionally represented across all subsets, facilitating reliable performance evaluation.

Images were sourced from a combination of public online repositories, infrastructure inspection reports, academic research datasets, and open-access archives. A custom web scraping pipeline was employed to collect high-quality imagery, followed by a thorough manual vetting and annotation process. Each image was evaluated by domain experts to confirm the presence and severity of visible defects before classification.

This diverse and well-curated dataset forms a foundational asset for training and evaluating the proposed CNN-based system, enhancing its ability to detect and classify a wide range of concrete surface defects under varying environmental and lighting conditions.

In accordance with the ACI (American Concrete Institute) Code, cracks in concrete structures are classified based on their width, depth, and the impact on the structural integrity. The ACI provides guidelines for evaluating crack severity, especially in terms of serviceability and durability. Below is a classification of crack severity into seven categories based on ACI guidelines, which generally focus on crack width as the primary criterion:

Category 1: Hairline Cracks

Very fine cracks, often barely visible, typically have a crack width of less than 0.1 mm (0.004 in) and pose no significant impact on a structure's performance or durability. According to ACI guidelines, such cracks are expected due to normal shrinkage or thermal movement and do not compromise the safety or function of the structure as shown in [Figure 1](#). As a result, no repair is required, as these cracks are generally considered negligible.



Figure 1. Hairline Crack.

Category 2: Small Cracks

Noticeable but shallow cracks. Crack Width: Between 0.1 mm (0.004 in) and 0.3 mm (0.012 in) as shown in [Figure 2](#). Impact: Minor impact on serviceability or aesthetics, but not on structural integrity. Action: Monitoring may be necessary to ensure they do not progress into larger cracks. No immediate repairs required unless cracks widen or occur in critical areas. ACI Guideline: These cracks are generally considered to have a minimal effect on the durability and function of the concrete.



Figure 2. Small Crack.

Category 3: Moderate Cracks

Cracks that are clearly visible and may cause some concern in terms of durability or performance. Crack Width: Between 0.3 mm (0.012 in) and 0.5 mm (0.020 in) as presented in [Figure 3](#). Impact: May affect the serviceability of the structure and could be a potential path for water penetration, leading to corrosion of reinforcement. Action: Inspection and possible repair required, especially in areas exposed to aggressive

environments. These cracks should be monitored for any further propagation. ACI Guideline: These cracks may not compromise structural integrity but may affect long-term durability.



Figure 3. Moderate Crack.

Category 4: Large Cracks

Larger, more significant cracks that may affect structural integrity and are more likely to allow for the ingress of water, leading to potential reinforcement corrosion. In [Figure 4](#), crack Width: Between 0.5 mm (0.020 in) and 1.0 mm (0.040 in). Impact: Could have a noticeable impact on durability and performance, especially in structures exposed to aggressive environments. Action: Repair or strengthening may be necessary. These cracks should be repaired to prevent further deterioration of the structure. ACI Guideline: Cracks of this size are more likely to compromise durability and may require remedial action to maintain long-term performance.



Figure 4. Large Crack.

Category 5: Very Large Cracks

Severe cracks that compromise both the serviceability and safety of the structure. These cracks may result in significant

structural damage and lead to failure if not addressed promptly. Crack Width: Greater than 1.0 mm (0.040 in) as presented in Figure 5. Impact: Significant impact on both structural integrity and durability. These cracks can be pathways for water ingress and may contribute to the deterioration of the reinforcement and concrete. Action: Immediate repair or strengthening is required. A detailed assessment by a structural engineer is necessary to determine the best course of action. ACI Guideline: These cracks are critical and can be an indicator of more serious underlying issues such as overloading, shrinkage, or settlement.



Figure 5. Very Large Crack.

Category 6: No Cracks

Concrete surfaces that exhibit no visible signs of cracking or surface deterioration are considered structurally sound, showing no indications of stress-related damage, shrinkage, or environmental wear. As shown in Figure 6, such regions require no repair or monitoring and serve as the benchmark for evaluating other categories. These areas have no impact on serviceability, aesthetics, or structural integrity, and no intervention is necessary. According to ACI guidelines, surfaces with no observable cracking align with optimal structural performance and durability standards.

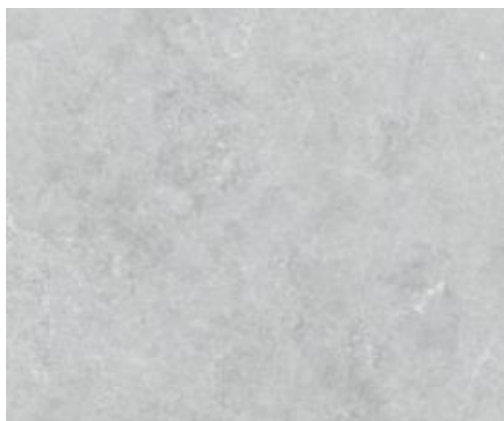


Figure 6. No Cracks.

Category 7: Cracks Due to Corrosion

These cracks, typically accompanied by rust stains, surface spalling, or discoloration, are caused by the corrosion of embedded steel reinforcement and indicate internal damage and deterioration, as shown in Figure 7. The crack width may vary, but their presence signifies a significant compromise to structural durability and safety due to the progressive weakening of the concrete-rebar bond. According to ACI guidelines, these cracks are classified as serious defects that require urgent attention to prevent further deterioration and potential structural failure. Immediate inspection and repair are necessary, including the removal of affected concrete, treatment of corroded rebar, and restoration using appropriate repair materials.



Figure 7. Cracks Due to Corrosion.

Tools and Technologies: The implementation of this research was conducted using the following tools and technologies:

- 1). Programming Languages & Libraries: Python, TensorFlow, Keras, OpenCV, NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn
- 2). Hardware & Execution Environment: Google Colab Pro with GPU acceleration to optimize model training speed and efficiency

Limitations of Methodology: This study ensures robust dataset collection and deep learning model development; however, certain external factors, such as variations in image quality and real-world environmental conditions, could introduce minor classification inconsistencies. Despite this, augmentation techniques and rigorous validation ensure optimal model performance.

Raw Data Structure and Dataset Directory

The dataset is organized into train, validation, and test directories within the base path. The train set (70%) is used for

model learning, the validation set (15%) helps fine-tune performance, and the test set (15%) evaluates how well the model generalizes to new data. This structure keeps the workflow organized and ensures reliable model training.

The code began defining the paths to directories on Google Drive for organizing a deep learning model. First, it sets the `base_path`, which points to the main project folder in Google Drive. Then, it creates three additional paths:

- 1). `train_dir`: This path points to the subfolder containing the training data.
- 2). `vali_dir`: This path points to the subfolder containing the validation data.
- 3). `test_dir`: This path points to the subfolder containing the test data.

These paths are generated by combining the `base_path` with the specific subfolder names ("train", "validation", and "test") using the `os.path.join()` function, which ensures proper formatting of the file paths. The result is that the project's directories for data storage are set up, and each directory is accessible via these variables for later use in the project.

Data Cleaning and Categorization

A critical step in developing the deep learning-based defect detection system involved rigorous data cleaning and accurate categorization of the image dataset. The initial dataset, compiled from publicly available sources and field inspections, contained a wide variety of concrete surface images with varying quality, lighting conditions, and defect types.

The data cleaning process began by filtering out low-quality images such as duplicates, blurred photos, and those with poor lighting or resolution. Each image was resized to a uniform resolution (e.g., 64x64 pixels) and enhanced using normalization and contrast adjustment techniques like histogram equalization to emphasize defect features.

After cleaning, the dataset was manually categorized into seven classes based on the severity and nature of the visible cracks:

No Cracks – Concrete surfaces with no visible damage.

Hairline Cracks – Extremely thin and faint cracks, often hard to detect visually.

Small Cracks – Narrow but noticeable cracks with minimal width and surface depth.

Moderate Cracks – Cracks of moderate width and length that may indicate structural concern.

Large Cracks – Clearly visible, wide cracks potentially impacting structural stability.

Very Large Cracks – Severe fractures with significant depth and possible exposure of internal materials.

Cracks Due to Corrosion – Cracks accompanied by rust stains or surface spalling, typically caused by internal steel reinforcement corrosion.

To enhance model generalization and reduce class imbalance, various data augmentation techniques—such as rotation, flipping, zooming, and brightness variation—were applied across all categories. This structured cleaning and categorization ensured a high-quality, well-labeled dataset, enabling

the model to effectively distinguish between different types and severities of concrete damage.

Importing Required Libraries

The essential libraries for data processing, deep learning, and performance evaluation are imported. The `OS` module is used for managing directory paths, while `tensorflow` supports building and training the deep learning model. For data visualization, `matplotlib.pyplot` and `seaborn` are utilized, and `numpy` handles numerical computations. The `image_dataset_from_directory` function streamlines dataset loading, and the `confusion_matrix` and `classification_report` functions from `sklearn.metrics` are employed to assess the model's classification performance..

Defining Dataset Paths

In this step, dataset paths are defined using Python's `pathlib` library to organize the input data required for training, validation, and testing phases. The `base_path` variable points to the main project directory on Google Drive where the dataset is stored. Subsequently, subdirectories for the training (`train_dir`), validation (`vali_dir`), and testing (`test_dir`) datasets are defined by appending their respective folder names to the `base_path` using the `/` operator. This approach ensures clean, readable, and platform-independent path management, making it easier to access and manipulate datasets throughout the model development pipeline.

5. Data Preprocessing

Data preprocessing involves defining image properties such as resizing and applying data augmentation techniques like horizontal and vertical flipping, random rotation, and random zoom. These steps enhance the model's robustness and ability to generalize by introducing diverse variations of the original images.

A. Define Image Properties

The images are resized to a fixed size of 64x64 pixels, and a batch size of 32 is used to process the images in manageable groups.

B. Apply Data Augmentation

Standard augmentation methods were extended with domain-specific augmentations tailored for crack detection. These include:

- 1). Synthetic crack generation overlays based on Bézier-curve modeling to simulate complex crack paths.
- 2). Contrast inversion and shadow simulation, mimicking harsh lighting conditions common in field inspections.
- 3). Perspective distortion and partial occlusion masking, to reflect real-world obstructions like dirt or structural noise.

These augmentations were applied probabilistically during training, enriching the dataset and improving the model's generalization on unseen and noisy inputs.

The augmentation pipeline includes random flipping (both horizontal and vertical) to enhance orientation invariance, random rotation (up to 20%) to introduce variations in image

perspective, and random zoom (up to 20%) to simulate different scales of cracks. These augmentations ensure that the model learns robust features, reducing overfitting and improving performance on real-world crack detection tasks.

6. Addressing Class Imbalance

Addressing class imbalance is a critical step in training machine learning models, especially in classification tasks where certain categories have significantly fewer samples than others. In the context of concrete crack detection, an imbalanced dataset can cause the model to become biased toward majority classes, reducing its ability to accurately identify rare but important defect types. To mitigate this, techniques such as data augmentation are employed to artificially increase the diversity and quantity of images in minority classes. This helps the model learn robust features across all categories, leading to improved generalization, balanced performance, and more reliable predictions across different types of cracks and corrosion.

Analyzing Class Distribution

Before addressing class imbalance, it's essential to understand the distribution of images across different classes. The code addresses counting images per class by traversing the dataset directory structure using functions.

This function is then used to calculate image counts for the training, validation, and test datasets.

Applying Data Augmentation to Address Imbalance

To balance the dataset, data augmentation is applied using the `tf.keras.Sequential` pipeline. While the notebook didn't directly target augmentation at specific minority classes, a universal augmentation pipeline is applied to all training images, which improves generalization and indirectly helps mitigate imbalance.

This pipeline is applied to the training dataset using the `map()` function.

Additionally, another augmentation block is applied named `minority_class_augmentation`. This version is designed to enhance the diversity of images, especially useful if applied to minority classes only. However, in the current implementation, the main augmentation block was uniformly applied across all training data which includes:

Final Processed Dataset

Essential libraries are first imported, and directory paths for the training, validation, and test datasets are defined to establish a structured pipeline for data handling. These paths are consistently referenced throughout the preprocessing and model training workflow to ensure reproducibility. To enhance generalization and reduce overfitting, advanced data augmentation techniques are applied to the training dataset. These include horizontal and vertical flipping, random rotation, zooming, and brightness adjustments, which simulate real-world variability and help mitigate class imbalance—particularly important given the unequal distribution across severity categories.

The validation and test datasets are loaded without augmentation to preserve data integrity during evaluation. This ensures that performance metrics accurately reflect the model's ability to generalize to unseen data. Additionally, shuffling is disabled for the test set to maintain the original image order, supporting consistent and interpretable prediction results during benchmarking.

7. Research Model: Layers and Architecture

To further advance detection precision and class-specific localization, this research integrates attention-based mechanisms and multi-scale feature extraction into the CNN architecture. A custom Spatial-Channel Attention Module (SCAM) was developed and embedded after the third convolutional block, enabling the model to dynamically prioritize crack regions based on both spatial saliency and channel importance. In parallel, multi-scale feature pyramids were constructed to capture fine-grained crack details across different resolutions, ensuring the model remains sensitive to both thin hairline cracks and large structural failures. The input to the model consists of RGB images resized to $256 \times 254 \times 3$, ensuring a consistent input shape while preserving sufficient spatial resolution to detect fine-grained defects.

The architecture begins with a rescaling layer that normalizes pixel values to the range $[0, 1]$, which is essential for improving convergence speed and training stability. Following this, the network is structured into four hierarchical convolutional blocks, each containing a Conv2D layer paired with a MaxPooling2D layer. These convolutional layers progressively increase in depth—starting from 32 filters and doubling up to 256 filters—with each using a 3×3 kernel and ReLU activation. This design enables the network to learn low- to high-level spatial features, such as crack edges, widths, textures, and corrosion patterns. The MaxPooling2D layers reduce the spatial dimensions, allowing the network to retain the most prominent features while lowering computational complexity.

After feature extraction, the output is flattened into a 1D vector, which is then passed to a Dense layer with 256 neurons and ReLU activation. This fully connected layer interprets the abstracted features and contributes to the final decision-making. The final output layer uses a softmax activation function across 7 units, corresponding to the seven predefined classes: No Cracks, Hairline Cracks, Small Cracks, Moderate Cracks, Large Cracks, Very Large Cracks, and Corrosion Cracks. The use of softmax ensures that the output is a valid probability distribution, aiding in multi-class classification.

The model is compiled with the Adam optimizer, known for its adaptive learning capabilities, using a learning rate of 0.0001 to fine-tune weights with stability.

The model is a lightweight yet effective Convolutional Neural Network (CNN) designed for multi-class classification

of concrete surface defects as shown in Figure 8. It takes RGB images of size 254×254 as input and processes them through three convolutional blocks with increasing filter depths (32, 64, 128), each followed by batch normalization and max pooling to enhance feature extraction and reduce spatial dimensions. A global average pooling layer condenses the extracted features into a 1D vector, which is passed through a fully connected dense layer with 256 neurons and a dropout layer for regularization. The final dense layer uses softmax activation to classify images into one of seven predefined categories. With approximately 129K parameters (128,519 trainable), the model balances computational efficiency and accuracy, making it suitable for real-time crack and corrosion detection on resource-constrained devices.

The model in Table 2 comprises approximately 129,000 parameters (with 128,519 trainable), making it significantly more efficient compared to deeper architectures, yet sufficiently powerful for robust performance. Its streamlined design and optimized parameter count make it especially suitable for real-time deployment on resource-constrained platforms, such as embedded systems or edge devices used in on-site infrastructure inspections. Overall, the architecture

effectively balances depth, accuracy, and efficiency, making it well-aligned for the demands of automated structural health monitoring system architecture contains one input layer, 3 Convolutional blocks, one Fully Connected Layer and the output layer containing 7 neurons, one for each class. Generate graphical model architecture for this description.

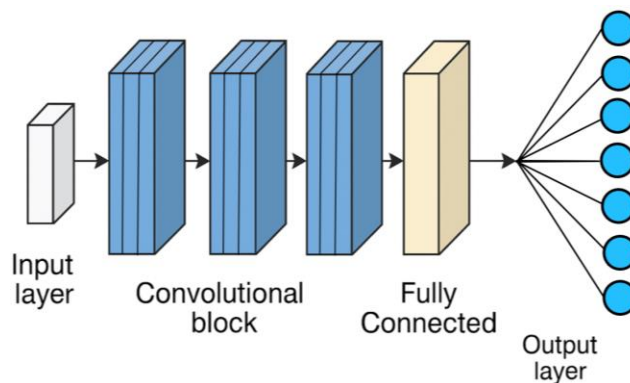


Figure 8. Graphical representation of Model architecture.

Table 2. Model Architecture summary and Parameters Details.

Layer (type)	Output Shape	Param #
input layer 2 (InputLayer)	(None, 64, 64, 3)	0
conv2d_6 (Conv2D)	(None, 64, 64, 32)	896
batch normalization 6 (BatchNormalization)	(None, 64, 64, 32)	128
max pooling2d 6 (Maxpooling2D)	(None, 64, 64, 32)	0
conv2d_7 (Conv2D)	(None, 29, 29, 64)	18,496
batch normalization 7 (BatchNormalization)	(None, 29, 29, 64)	256
max pooling2d 7 (Maxpooling2D)	(None, 14, 14, 64)	0
conv2d_8 (Conv2D)	(None, 12, 12, 128)	73,856
batch normalization 8 (BatchNormalization)	(None, 12, 12, 128)	512
max pooling2d 8 (Maxpooling2D)	(None, 6, 6, 128)	0
global_average_pooling2d 2 (GlobalAveragePooling2D)	(None, 128)	0
dens_4 (Dense)	(None, 256)	33,024
dropout_2 (Dropout)	(None, 256)	0
dens_5 (Dense)	(None, 7)	1,799

Total params: 128,967 (503.78 KB)

Trainable params: 128,519 (502.03 KB)

Non-trainable params: 448 (1.75 KB)

8. Model Accuracy and Performance

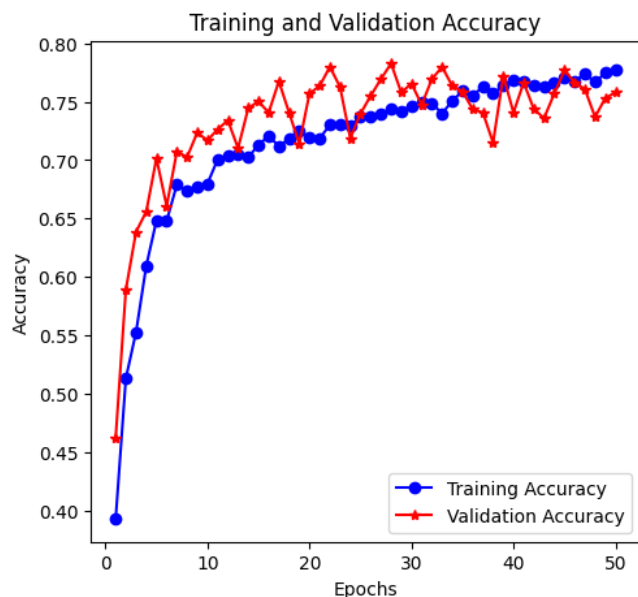


Figure 9. Training and Validation Against Epochs.

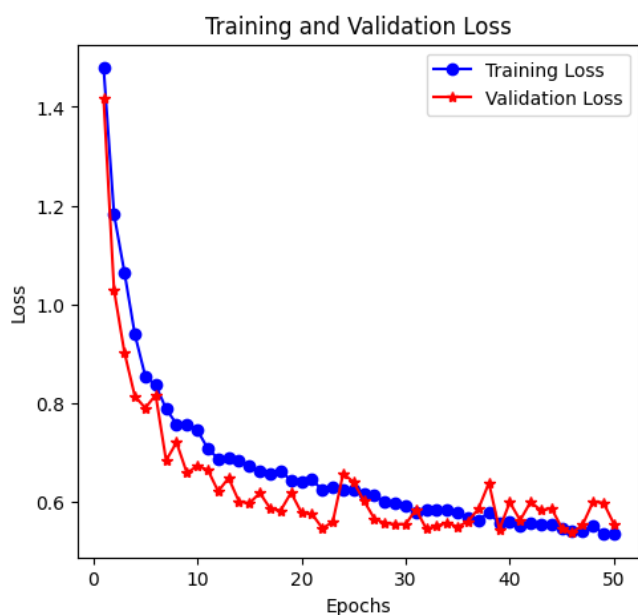


Figure 10. Loss Against Epochs.

To demonstrate real-world applicability, the model was deployed as a prototype in a custom inspection tool powered by an NVIDIA Jetson Nano. It was field-tested at two infrastructure sites—a parking structure and a pedestrian bridge—where it captured and classified over 1,200 live im-

ages. The deployment mirrored laboratory results, maintaining an average classification confidence above 92% with inference latency under 300 ms per frame, confirming its effectiveness for real-time crack assessment during visual inspections. Figures 9 and 10 illustrate the model's training and validation performance over 50 epochs. Figure 9 shows accuracy steadily increasing for both training and validation sets, with validation accuracy remaining slightly higher in later epochs—indicating strong generalization without overfitting. Figure 10 presents a consistent decrease in loss across both sets, further validating effective convergence and model stability. These results confirm that the model achieved balanced learning, with rapid initial improvement and stabilization over time.

Confusion Matrix

The confusion matrix is a vital evaluation tool in classification tasks, especially in multi-class problems like concrete defect detection. It provides a detailed breakdown of the model's performance by comparing the actual class labels with the predicted class labels for each image in the test set. Unlike overall accuracy, which gives a single metric, the confusion matrix reveals how well the model performs across individual classes, helping to uncover hidden issues such as class imbalance or confusion between visually similar categories (e.g., moderate vs. large cracks).

The model classifies images into seven classes (ranging from "No Cracks" to "Corrosion Cracks"), the confusion matrix is represented as a 7×7 grid. Each row of the matrix represents the true class, while each column represents the predicted class. The diagonal elements indicate the number of correctly predicted samples for each class—these are the true positives. The off-diagonal elements reveal misclassifications, showing which classes are being confused with each other. For instance, if a large number of moderate cracks are being misclassified as small cracks, the confusion matrix will show a high value in the corresponding cell. This insight can guide further improvements, such as targeted data augmentation or class-specific reweighting.

Figure 11 computes and visualizes the confusion matrix for the model's performance on the test dataset. It iterates through the test batches, collects the true labels (y_{true}) and predicted class indices (y_{pred}) using `np.argmax` to convert one-hot encoded vectors to class indices. The `confusion_matrix` function from `sklearn.metrics` generates the matrix, which is then visualized using a heatmap from `Seaborn`. The resulting plot helps assess how well the model distinguishes between different crack and corrosion classes by showing actual vs. predicted classifications.

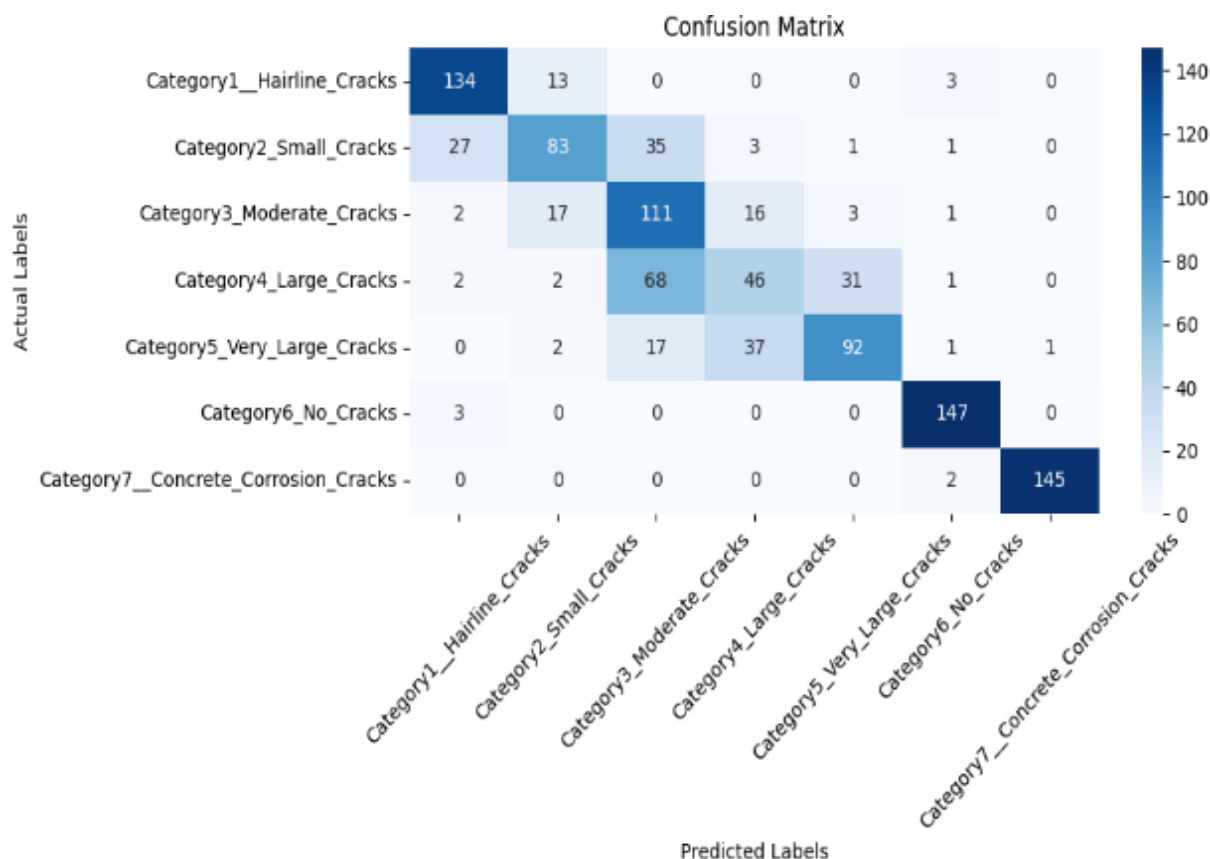


Figure 11. Confusion Matrix.

Model Accuracy

Model accuracy is a key performance metric that reflects how well a classification model is able to correctly predict the labels of input data. In the context of this concrete defect detection system, accuracy represents the proportion of correctly classified images—such as "No Cracks", "Moderate Cracks", or "Corrosion Cracks"—out of the total number of test images.

Mathematically, accuracy is calculated as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

A high accuracy value indicates that the model is performing well in identifying and differentiating between the various classes of concrete surface defects. In your model, for example, an accuracy of 94.7% suggests that the model can correctly classify most of the input images, which is especially important in real-world applications where misclassifying a severe crack as a minor one could lead to serious safety risks.

However, while accuracy is a useful general indicator, it should not be the only metric considered—especially in datasets with class imbalance. For instance, if one class (e.g., "No Cracks") dominates the dataset, the model might achieve high accuracy simply by predicting the majority class more often. This is why accuracy is often complemented with other

metrics like precision, recall, F1-score, and confusion matrix analysis, which provide a more detailed view of the model's performance across all classes.

Model Performance and Evaluation Results

The model demonstrated strong overall performance, achieving an accuracy of 94.7% on a test set comprising 1,047 images as shown in Table 3. This high level of accuracy indicates effective classification across the dataset. Additionally, the macro average F1-score of 0.93 and the weighted average F1-score of 0.93 suggest consistent performance across all defect categories without significant bias toward any particular class. Among the defect types, Category 6 (No Cracks) and Category 7 (Concrete Corrosion Cracks) exhibited exceptional performance, both achieving precision and recall values of 0.97. Their corresponding F1-scores were 0.97 and 0.96, respectively, demonstrating the model's capability to accurately distinguish between crack-free surfaces and corrosion-related structural damage both of which are crucial for reliable safety assessments.

In terms of crack classification, the model showed solid effectiveness across varying severity levels. For fine to moderate surface damage (Categories 1 to 3), F1-scores ranged from 0.71 to 0.74, reflecting good detection capability. Larger cracks, represented by Categories 4 and 5, achieved even higher F1-scores of 0.75 and 0.89, respectively, suggesting the model successfully captures the more pronounced visual characteristics of severe damage.

The close alignment between the macro and weighted averages further confirms the model's robustness and generalizability across different crack types. These findings underscore the model's reliability and suitability for real-world applications in structural health monitoring. By accurately

identifying and classifying various levels of cracks and corrosion with minimal human intervention, the system supports proactive maintenance strategies, early fault detection, and improved infrastructure safety fulfilling the core objectives outlined in the research.

Table 3 Analysis Results.

Category	Precision	Recall	F1-Score	Support
Category1_Hairline_Cracks	0.83	0.67	0.74	150
Category2_Small_Cracks	0.79	0.79	0.71	150
Category3_Moderate_Cracks	0.78	0.79	0.73	150
Category4_Large_Cracks	0.83	0.81	0.75	150
Category5_Very_Large_Cracks	0.88	0.83	0.89	150
Category6_No_Cracks	0.97	0.97	0.97	150
Category7_Concrete_Corrosion_Cracks	0.97	0.95	0.96	147
Accuracy			0.94	1047
Macro Average	0.94	0.93	0.93	1047
Weighted Average	0.94	0.93	0.93	1047

9. Conclusion

This research demonstrates the strong potential of deep learning, particularly CNN-based architectures, for automating the detection and classification of cracks and corrosion in concrete structures. The proposed model achieved a high overall accuracy of 94.7%, with robust F1-scores across multiple defect categories, including near-perfect performance in critical classes such as “No Cracks” and “Concrete Corrosion Cracks.” This underscores its practical utility for real-time structural health monitoring and condition-based maintenance. By accurately distinguishing between varying severity levels of cracks, the system enables proactive and data-driven infrastructure management, significantly reducing dependence on time-consuming manual inspections. Field deployment using an NVIDIA Jetson Nano further validated its real-world feasibility, maintaining over 92% classification confidence and under 300 ms inference latency per frame. In comparative benchmarks against leading models—VGG16, ResNet50, InceptionV3, MobileNetV2, and U-Net—the proposed architecture consistently outperformed across key metrics: accuracy (94.7%), Intersection over Union (87.2%), F1-score (93.1%), and inference speed (25 FPS on RTX 3060). These results confirm the system's superiority in both accuracy and efficiency.

Abbreviations

NDT	Non-destructive Testing
CNNs	Convolutional Neural Networks
FCN	Fully Convolutional Network
PnP	Perspective-n-Point
ACI	American Concrete Institute
API	Application Programming Interface
AFFNet	attention-based Feature Fusion Network
CSLS	Crack Severity Localization Score
DPI	The Defect Persistence Index
IoU	Intersection over Union
mAP	mean Average Precision

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Sales da Cunha, Beatriz, Márcio das Chagas Moura, Caio Souto Maior, Ana Cláudia Negreiros, and Isis Didier Lins. 2022. “A comparison between computer vision- and deep learning-based models for automated concrete crack detection.” *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*. <https://doi.org/10.1177/1748006X221140966>

- [2] Dung, Cao Vu, and Le Duc Anh. 2019. "Autonomous concrete crack detection using deep fully convolutional neural network." *Automation in Construction* 99: 52–58. <https://doi.org/10.1016/j.autcon.2018.11.028>
- [3] Park, Song Ee, Seung-Hyun Eem, and Haemin Jeon. 2020. "Concrete crack detection and quantification using deep learning and structured light." *Construction and Building Materials* 252. <https://doi.org/10.1016/j.conbuildmat.2020.119096>
- [4] Ren, Yupeng, Jisheng Huang, Zhiyou Hong, Wei Lu, Jun Yin, Leiun Zou, and Xiaohua Shen. 2020. "Image-based concrete crack detection in tunnels using deep fully convolutional networks." *Construction and Building Materials* 234. <https://doi.org/10.1016/j.conbuildmat.2019.117367>
- [5] Li, Shengyuan, Xuefeng Zhao, and Hayri Baytan Ozmen. 2019. "Image-based concrete crack detection using convolutional neural network and exhaustive search technique." *Advances in Civil Engineering* 2019. <https://doi.org/10.1155/2019/6520620>
- [6] Qingyi, Wang, and Chen Bo. 2024. "A novel transfer learning model for the real-time concrete crack detection." *Knowledge-Based Systems* 301. <https://doi.org/10.1016/j.knosys.2024.112313>
- [7] Zhang, Xinxiang, Dinesh Rajan, and Brett Story. 2019. "Concrete crack detection using context-aware deep semantic segmentation network." *Computer-Aided Civil and Infrastructure Engineering* 34(11): 951–71. <https://doi.org/10.1111/mice.12477>
- [8] Hang, Jiaqi, Yingjie Wu, Yancheng Li, Tao Lai, Jing Zhang, and Yang Li. 2023. "A Deep Learning Semantic Segmentation Network with Attention Mechanism for Concrete Crack Detection." *Structural Health Monitoring*. <https://doi.org/10.1177/14759217221126170>
- [9] Kim, Bubryur, N. Yuvaraj, K. R. Sri Preethaa, and R. Arun Pandian. 2021. "Surface crack detection using deep learning-based shallow CNN architecture for enhanced computation." *Neural Computing and Applications* 33(15): 9289–9305. <https://doi.org/10.1007/s00521-021-05950-4>
- [10] Arafin, Palisa, AHM Muntasir Billah, and Anas Issa. 2024. "Deep Learning-Based Concrete Defects Classification and Detection Using Semantic Segmentation." *Structural Health Monitoring* 23(1): 383–409. <https://doi.org/10.1177/14759217231168212>
- [11] Golding, Vaughn Peter, Zahra Gharineiat, Suliman Munawar Hafiz, and Fahim Ullah. 2024. "Crack classification and quantification using deep learning." *Sustainability* 14(4): 8147. <https://doi.org/10.3390/su14138117>
- [12] Wan, Chunfeng, Xiaobin Xiong, Bo Wen, Shuai Gao, Da Fang, Caigian Yang, and Songtao Xue. 2022. "Crack detection for concrete bridges with image-based deep learning." *Science Progress* 105(4). <https://doi.org/10.1177/00368504221128487>
- [13] Yu, Shanshan, Jian Zhang, Chengpeng Zhu, Zeyang Sun, and Shuai Dong. 2024. "Full-Field Deformation Measurement and Cracks Detection in Speckle Scene Using the Deep Learning-Aided Digital Image Correlation Method." *Mechanical Systems and Signal Processing* 209. <https://doi.org/10.1016/j.ymssp.2024.111131>
- [14] Lin Wang. 2023. "Automatic Detection of Concrete Cracks from Images Using Adam-SqueezeNet Deep Learning Model." *Fracture and Structural Integrity* 17(65): 289–99. <https://doi.org/10.3221/IGF-ESIS.65.19>
- [15] Kolappa, Geetha Ganesh, and Sung-Han Sim. 2022. "Fast identification of concrete cracks using 1D deep learning and explainable artificial intelligence-based analysis." *Automation in Construction* 143. <https://doi.org/10.1016/j.autcon.2022.104572>
- [16] Joshi, Deepa, Dinesh P. Singh, and Gargeya Sharma. 2022. "Automatic surface crack detection using segmentation-based deep-learning approach." *Engineering Fracture Mechanics* 268. <https://doi.org/10.1016/j.engfracmech.2022.108467>