

ARTICLE

UAV-Deep Learning-Based Approach in Civil Structural Diagnosis

Wael A. Altabey*

Department of Mechanical Engineering, Faculty of Engineering, Alexandria University, Alexandria, Egypt

*Corresponding Author: Wael A. Altabey. Email: wael.altabey@gmail.com

Received: 22 January 2026; Accepted: 06 May 2026; Published: 24 August 2026

ABSTRACT: The goal of this paper is to improve the monitoring of civil structures when we pair unmanned aerial vehicles (UAVs) technology with the current proposed algorithm, particularly to identify cracks in concrete structures. Typically, the current UAV methods are more about creating state maps of these structures, but they struggle with the impact of the drone's movement on crack detection accuracy. This presents challenges for using intelligent systems for concrete crack detection. The current approach combines advanced technologies with a network of high-definition cameras mounted on inspection UAV systems and distributed in different parts of the structure's surface, and smart processing methods to keep tabs on structure health accurately and in real-time. By merging a novel Convolutional Neural Network (CNN) with a specialized technique called the Crack Contour Network (CCN) to extract features, we boost the accuracy and reliability of our assessments for target identification, the type, location, and dimensions of cracks, including their borders. After collecting those images using a high-definition camera-based UAV system, they preprocessed the images by cropping and segmenting the crack images, standardizing their size, adjusting the contrast, enhancing the image data to boost the sample count, and creating a database with these labeled samples. Once that's done, use this database for training the CNN-CCN, which was built and tuned to the parameters of the network. Finally, the network tested that's been trained, and the algorithm will output the detection results for the cracks. An algorithm CNN-CCN uses metrics like Intersection over Union (IoU), accuracy rates, regression rates, and F1-score to evaluate its performance in crack identification and determining the crack's contour area. The results indicate that the proposed CNN-CCN network performs well, achieving high IoU for crack detection, accuracy rates of 96.43%, regression rates of 93.77%, and an F1-score of 91.65%, alongside a training time of 75 s and a model size of 199 KB. A deep analysis of the confusion matrix and a side-by-side look at predicted and actual conditions highlight how well the model can tell apart different traditional algorithms in the literature. This technique allows for automatic high-dimensional extraction and complex features from infrastructures, leading to reduced interference and errors in detection, which enhances the clarity and overall effectiveness of civil structure assessments.

KEYWORDS: Deep learning; civil structural diagnosis; UAVs; convolutional neural network (CNN); crack identification; crack contour network (CCN)

1 Introduction

Structural Health Monitoring (SHM) focuses on monitoring the weak spots in structures by gathering suitable data before any damage happens, which ultimately boosts the resilience of civil structures. In deep learning (DL) studies, as shown in Fig. 1, we can sort approaches into two main categories: parametric and non-parametric methods. For parametric approaches, monitoring relies on structural vibrational characteristics like mode shapes, natural frequency, and curvature. On the other hand, non-parametric methods use sensor data or images of structures to identify damage [1]. That said, many of these methods don't tackle the

noise created by environmental factors and operational conditions that can affect the features when used in supervised learning. This creates a bit of a challenge because while supervised learning needs datasets from both intact and damaged structures, in most cases, we only have data from undamaged structures available. So, we often lean on unsupervised learning instead [2].

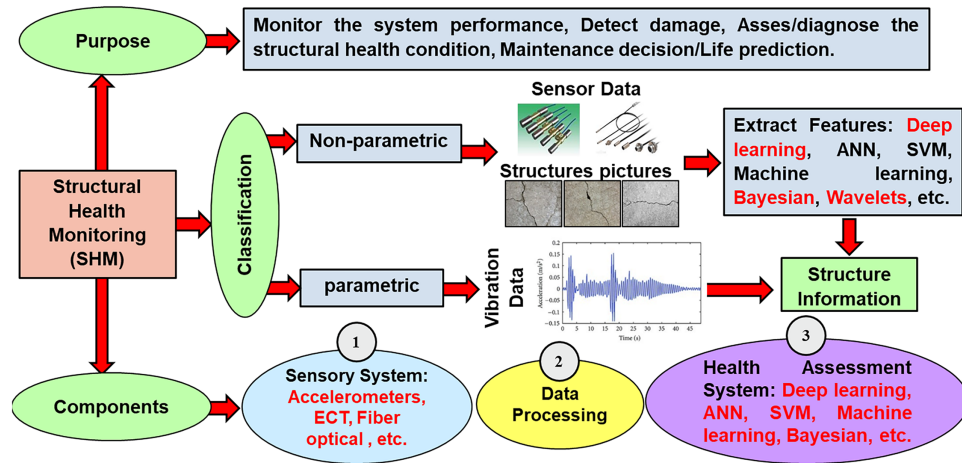


Figure 1: Structural health monitoring (SHM).

1.1 Related Work

Civil infrastructure is fundamental to sustaining long-term national economic stability. Ensuring structural integrity and preventing catastrophic failure necessitates systematic inspection and maintenance protocols. Conventionally, condition assessment during the operational phase has relied upon manual visual inspection [3]; however, this approach is inherently subjective, labor-intensive, and poses significant occupational hazards. Driven by technological integration, contemporary research has pivoted toward intelligent diagnostic methodologies, with automated crack detection emerging as a critical focal point in SHM [4]. Most infrastructure components are susceptible to cracking over their service life due to intrinsic material limitations [5]. In certain cases, environmental stressors and escalating structural loads accelerate crack initiation and propagation, thereby compromising serviceability and elevating the risk of structural failure [6]. Given the morphological diversity and varied etiologies of these defects, traditional identification methods remain inefficient and cost-prohibitive. Consequently, a primary challenge for infrastructure management is the high-fidelity identification of critical cracks that jeopardize structural safety amidst a high density of non-critical surface anomalies [7].

That's why civil engineering researchers have been working together across different industries and fields. They've been combining their unique insights to create a theoretical system for artificial intelligence that fits this sector. Plus, they've integrated deep learning algorithms with cutting-edge tools like advanced sensors, high-definition imaging systems, and lightweight robots. This approach is paving the way for intelligent crack detection, which really helps in making informed decisions about maintaining in infrastructure [8,9].

1.1.1 Unmanned Aerial Vehicles (UAV)-Based Crack Monitoring

Nowadays, Unmanned Aerial Vehicles (UAVs) are commonly used for assessing damage in infrastructure. These drones are equipped with High Definition Cameras (HDC), flying around the external walls of various buildings and sending back real-time video to a monitor on the ground. In traditional inspections,

an inspector has to sift through images of damaged structures to figure out what's wrong, which can be quite time-consuming and labor-intensive, often leading to results that aren't very accurate. Computer vision offers a more efficient way to analyze and monitor these images automatically. However, preparing input images for the training networks can be quite the hassle and isn't always efficient. Still, deep learning algorithms have made great leaps in recent years, excelling at various computer vision tasks, including classification and object detection [10].

When it comes to using UAVs for visual inspections, the way to identify and analyze cracks really hinges on the quality of the images we capture. So, making sure to get high-quality images is super important for getting trustworthy inspection results. But, a lot can mess with image quality—think about changing lighting, background distractions, or even motion blur from the drone not being stable [11,12]. To tackle these challenges, various image enhancement techniques have been put forward. For example, Dan and Dan [13] came up with a two-dimensional Amplitude and Phase Estimation (2D-APES) algorithm that boosts crack visibility by cutting down on low-frequency noise. Meanwhile, Chen et al. [14] developed an automated crack detection model that mixes self-organizing map optimization, fuzzy logic control, and a hyper-rectangle composite neural network to improve accuracy, even in tricky, noisy environments.

Image-based 3D reconstruction allows us to extract geometric details from 2D images, which helps in mapping out the scene in three dimensions. For instance, Liu et al. [15] applied these techniques to create mesh models of bridge piers during their crack detection work. They took crack features identified in UAV images and mapped them onto these models to fix any perspective issues, making it easier to locate them accurately. Ayele et al. [16] took this a step further by integrating 3D reconstruction with the Mask R-CNN semantic segmentation framework [17], which led to better visualization of bridge surfaces and enhanced efficiency in crack detection and measurements. More recently, Chen et al. [18] developed the Resampling-Aided Laplacian-Based Contraction (RALBC) algorithm, combining 3D reconstruction with semantic segmentation for precise crack skeleton extraction and better quantification. Even with these improvements, most methods still treat 3D reconstruction as just a geometric reference; the reconstructed models aren't often used directly for crack detection due to their lack of visual detail. In this study, we're looking to make the most of the structural accuracy offered by image-based 3D reconstruction to improve crack parameter quantification in UAV inspections.

1.1.2 Deep Learning-Based Crack Identification

Recently, the crack detection technologies via computer vision based have gained traction, with many researchers utilizing feature-based machine learning and traditional image processing techniques to analyze structural cracks [19]. However, since cracks often appear in complex and dynamic environments, the data gathered through visual methods frequently carries noise, which can undermine the reliability of the detection results [20]. This has led to a shared goal within the scientific and engineering communities to develop high-accuracy and methods of durable crack detection [21].

Deep learning, a key area of artificial intelligence, mimics human-decision-making processes and has demonstrated advantages over traditional techniques in fields like speech recognition, text translation, and image processing [22]. Particularly in medical image analysis, deep learning has helped doctors extract valuable pathological insights from extensive medical imaging, leading to significant advancements in diagnosis and treatment planning [23–26]. Inspired by these successes, some researchers have begun to apply deep learning to visual crack detection, aiming to enhance precision in identifying cracks within civil infrastructure [27]. Moreover, the rapid advancements in high-performance graphics processing units have further propelled the development and implementation of deep learning-based crack detection algorithms in engineering applications [28].

Zhang et al. [29] came up with a method using convolutional neural networks (CNNs) to spot road cracks, and their results were way better than what you'd get from manual feature extraction. Cha et al. [30] implemented a CNN with a sliding window approach for detecting concrete cracks, leading to significantly better performance compared to the usual edge-detection techniques. They also used a quicker region-based CNN for detecting multiple types of damage [31]. Later on, Li et al. [32] introduced a crack detection method that focused on pixel-level segmentation using a fully convolutional network. Hsieh and Tsai [33] pointed out that pixel-level crack segmentation has become a popular trend in deep learning for crack detection. Furthermore, applying a 3D mapping method alongside the pixel-level segmentation can help locate cracks in three-dimensional space coordinates [34–36]. For instance, Deng et al. [37] developed a method for mapping cracks in 3D to pinpoint them on the surface model of a structure and managed to get precise crack characterization. Meanwhile, Chaiyasarn et al. [38] were able to map crack pixels from a telephoto image to a wide-angle image, which helped them both characterize and localize the cracks at the same time.

As discussed earlier, many crack detection techniques that use CNN networks have shown to be much more effective than older methods, especially when it comes to civil structures like roads, bridges, tunnels, and concrete buildings with simpler backgrounds. But there's not a lot of literature out there on structure crack location and size of the affected areas, including their boundaries. This is a pretty complex issue that ties into a lot of challenges we face in this area.

This work discusses how to identify and extract the shape of surface cracks in civil structures like bridges using CNNs integrated with CCN techniques. The cracks images provided from UAVs systems distributed in different parts of the structure surface and after gathering those images, we preprocess them by cropping and segmenting the crack images, standardizing their size, adjusting the contrast, and enhancing the image data to boost the sample count. Next, we use specific software to mark the images based on the typical cracks we're detecting and create a database with these labeled samples. Once that's done, we feed into CNN built, the samples of training and tweak the parameters of the network. Finally, the network is tested that's been trained. The proposed technique calculates the location and size of the affected areas, including their boundaries. To evaluate the proposed method, we look at four key metrics: IoU, accuracy rates, regression rates, and the F1-score index, which demonstrate how effective our approach is. A deep analysis into the confusion matrix and a side-by-side look at predicted and actual conditions highlight how well the model can tell apart different traditional algorithms in literature.

2 Methodology

Implementing deep learning for crack detection is instrumental in mitigating operational risks, optimizing maintenance expenditures, and advancing the digital transformation of the civil engineering industry. The primary framework for developing such diagnostic models encompasses three core pillars: algorithmic architecture, curated datasets, and robust evaluation metrics. In this study, we integrated a deep learning-based detection model into an UAV platform to facilitate autonomous crack identification within civil infrastructure.

Fig. 2 shows how to identify cracks in structures like bridges. Right now, the inspection UAVs are used including high definition cameras to quickly gather high-quality images of the structure surfaces without making contact. After collecting images, we preprocess them. This includes cropping and segmenting images of any damage, standardizing their sizes, adjusting contrast, and enhancing the image data so we can increase our sample size. Then, we categorize the typical cracks found in structure images and use special software to label them, creating a database of samples. After that, use this database for training a CNN-CCN network, adjusting the parameters and running deep learning on the whole network before testing the trained model.

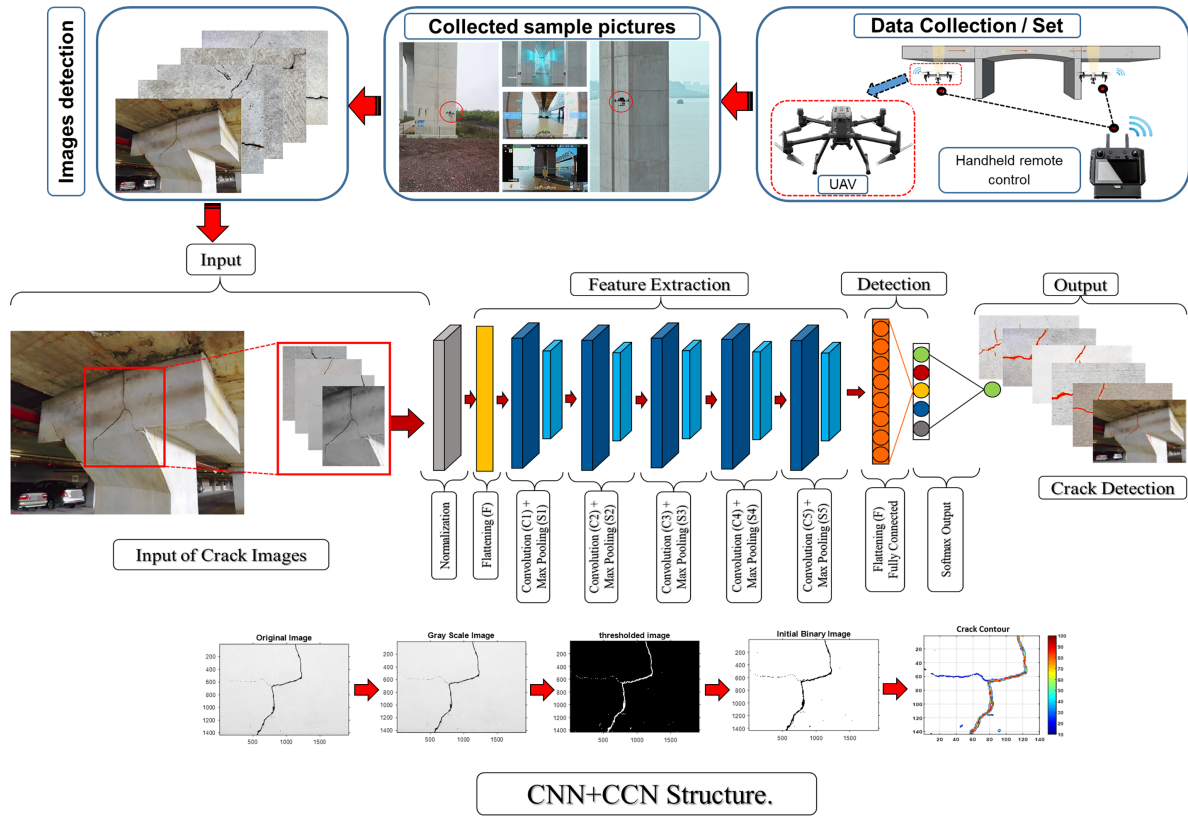


Figure 2: Illustration of the current approach.

Regarding the damage like cracks, the algorithm outputs a surface contour directly and specifies the contour area of the cracks along with a confidence level for their classification. For both horizontal and vertical cracks, once we have the surface contour, we can extract the crack shape and determine the crack area, which gives us a lot more detailed and accurate information.

2.1 Camera Calibration

Camera calibration is a fundamental procedure in diverse applications, including robotic navigation, computer-assisted surgery, and intelligent transportation systems. It serves as a critical prerequisite for the transformation of 2D image data into 3D coordinates. Technically, this process entails the determination of the camera's intrinsic lens parameters and extrinsic sensor orientations. A camera essentially performs a projective transformation, mapping a 3D scene onto a 2D image plane [39]. This projection can be mathematically modeled by the following equation:

$$a = P.A \quad (1)$$

In this context, a represents a 2D point in an image, P is the camera matrix, and A refers to a point in the 3D world.

$$\text{Where } a = \begin{bmatrix} x \\ y \\ x \end{bmatrix}, P = \begin{bmatrix} p1 & p2 & p3 & p4 \\ p5 & p6 & p7 & p8 \\ p9 & p10 & p11 & p12 \end{bmatrix}, A = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \text{ and the matrix Eq. (1) can be:}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} p1 & p2 & p3 & p4 \\ p5 & p6 & p7 & p8 \\ p9 & p10 & p11 & p12 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2)$$

Camera calibration is essential for extracting metric information from real-world scenes. A camera model is defined by two primary sets of parameters: intrinsic and extrinsic. Intrinsic parameters characterize the camera's internal geometry, including focal length, principal point offset, axis skew, and lens distortion. Conversely, extrinsic parameters define the camera's spatial position and orientation relative to a global coordinate system, typically expressed through rotation and translation matrices [40]. Fundamentally, calibration establishes the mathematical relationship between a 3D world coordinate and its corresponding 2D projection on the image plane. The internal characteristics, specifically the focal length, axis skew, and radial distortion, are illustrated in Fig. 3.

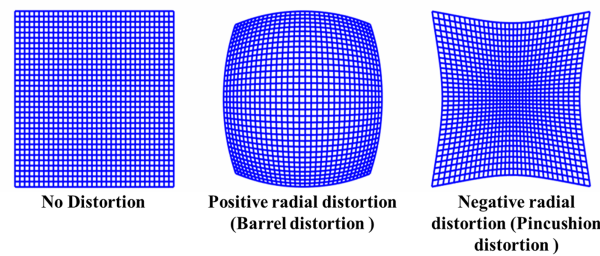


Figure 3: How barrel and pincushion distortion can affect a square.

2.2 Pinhole Camera Model

When we snap a photo with a camera, we turn that 3D space into a flat 2D image. The pinhole camera model explains how this process works mathematically, and it does so without using any lenses, as illustrated in Fig. 4. Basically, light rays come through the camera aperture and create an inverted image on the other side of the camera [41].

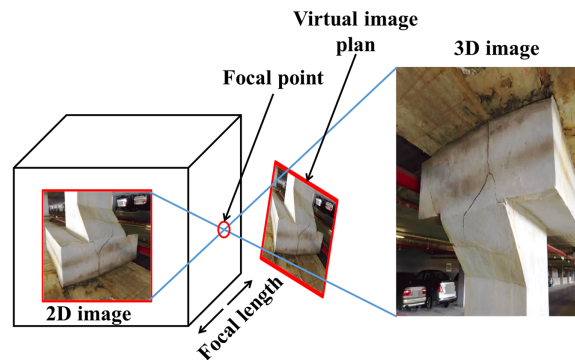


Figure 4: Pinhole camera model example.

We can represent the specs of this pinhole camera model with 3×4 matrices. If we use the intrinsic and extrinsic parameters, we can arrive at the following equation:

The geometric properties of the pinhole camera model are represented by a 3×4 projection matrix. By synthesizing the intrinsic and extrinsic parameters, the mapping between 3D world coordinates and 2D

image coordinates is defined by the following expression:

$$w \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} P \quad (3)$$

In this formulation, w denotes the scalar magnification factor, $x \ y$ represents the coordinates of the projected image point, and $X \ Y \ Z$ denotes the corresponding coordinates of the point in the world reference frame:

$$P = \begin{bmatrix} R \\ t \end{bmatrix} K \quad (4)$$

In this expression, P denotes the projection matrix, while R and t represent the rotation and translation components, respectively, and K signifies the intrinsic calibration matrix. A primary factor influencing measurement precision is lens-induced distortion, which encompasses three principal categories: radial, decentering, and thin prism distortion. Among these, radial distortion—comprising barrel and pincushion effects—is the most prevalent. Such aberrations cause rectilinear features in the physical world to appear curvilinear in the captured image, thereby deviating from the ideal pinhole camera model. For implementation, the OpenCV library utilizes this pinhole framework to characterize image formation by mapping 3D spatial points onto a 2D image plane via perspective transformation.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \quad (5)$$

$$x' = \frac{x}{z} \quad (6)$$

$$y' = \frac{y}{z} \quad (7)$$

$$u = f_x * x' + c_x \quad (8)$$

$$v = f_y * y' + c_y \quad (9)$$

In this context, (X, Y, Z) denotes the coordinates of a point in the 3D world coordinate system, whereas (u, v) represents the 2D pixel coordinates of its corresponding projection. The parameters (c_x, c_y) define the principal point, typically situated near the image center, and f_x and f_y signify the focal lengths expressed in pixel units.

The transformation is defined by the extrinsic matrix, which synthesizes the rotation and translation components. This matrix characterizes the camera's pose—its position and orientation—relative to a static scene, facilitating the transformation of 3D world coordinates into the camera-centered coordinate system. Given that the analysis is performed on 2D image data, projecting 3D spatial coordinates onto the image plane is a prerequisite. Furthermore, as physical optical systems deviate from the idealized pinhole model, it is imperative to incorporate distortion coefficients to account for lens aberrations.

2.3 Camera Descriptions

For this work, we really need a camera that can shoot in high resolution and has slow-motion capabilities. You can check out [Table 1](#) for a detailed look at the camera specs we used in this research.

Table 1: Camera descriptions.

Parameter	Value
Model	TH-ISM-H1 type
Physical quantity	Vertical and lateral displacement
Maximum number of measuring points	30
Maximum sampling rate	50 Hz (ÿ single 5 targets)
Image Sensor Resolution	20 Hz, 3840 * 2160 (not less than 800 W pixels)
Resolution	1/100,000 FOV (field of view)
Maximum test distance	400 m (beyond the range can be customized)
Measurement accuracy	0~50 US: ± 0.2 mm, 0–200 m: ± 1 mm, 200–400 m: ± 1.5 mm
Calibration method	The AI algorithm automatically corrects the influence of the angle of rotation and distance, without the need for distance measurement and leveling
Power supply	DC 24 V
Peak power	35 W
Average power	25 W
Communication Interface	Ethernet
Degree of protection	IP65 (requires optional protective cover)
Temperature drift control	Imaging system built-in constant temperature control device
Imaging system temperature measurement accuracy	$\pm 0.1^{\circ}\text{C}$
Imaging system temperature control accuracy	$\pm 0.2^{\circ}\text{C}$
Operating temperature	$-20^{\circ}\text{C} \sim +70^{\circ}\text{C}$
Physical dimension	310 mm (length)*108 mm (width)*154 mm (height)

3 The Deep Learning Algorithm

Deep learning has significantly accelerated the advancement of industrial intelligence, gaining widespread adoption among researchers and organizations across various domains [42]. Its applications span intelligent transportation, signal processing, natural language processing, robotics, bioinformatics, healthcare, and digital entertainment, as well as information retrieval, computational advertising, and financial modeling. Consequently, deep learning frameworks offer distinct performance advantages and capabilities that surpass the limitations of conventional methodologies [43–46].

In this part, we'll take a look at how deep learning has evolved, especially highlighting the fundamental structure of CNN and why they hold such critical advantages over traditional techniques in computer vision.

3.1 Convolutional Neural Network (CNN)

A CNN plays a crucial role in deep learning, especially in the realm of computer vision. Its sliding operation with the convolution kernel is what really gives it an edge, allowing it to pick up on patterns and structures in graphics effectively. This strength makes it excel at processing image data, often surpassing more traditional methods. Right now, it's the preferred choice for tasks like identifying cracks in images [47–49].

As you can see in Fig. 5, a typical CNN comprises several layers including the convolutional layer, pooling layer, activation function layer, and fully connected layer. The flatten layer usually positioned between the convolutional and pooling layers, acting as a connector. In simpler terms, the convolutional and pooling layers are all about extracting features, whereas the fully connected layers take care of classification or regression tasks.

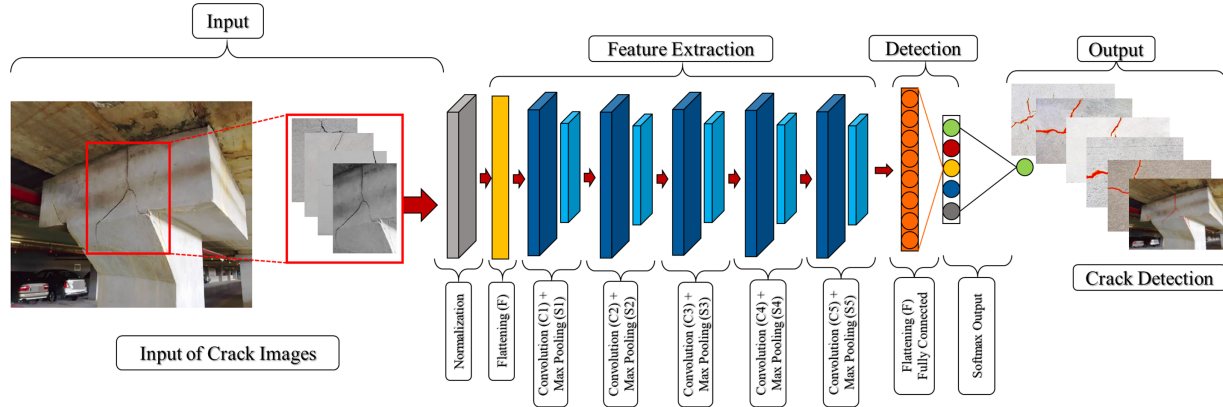


Figure 5: The architecture of a standard CNN.

Table 2 details the architectural specifications for the convolutional layers. For instance, the initial convolutional layer, C1, comprises six feature maps, where each neuron is connected to a 16×16 input receptive field. The resulting feature maps are 196×196 matrices. Utilizing four filters, each characterized by a 16×16 weight matrix and a single bias term, the layer contains $(16 \times 16 + 1) \times 6 = 1542$ trainable parameters. The connectivity between the input and C1, governed by a single kernel, results in $1542 \times (196 \times 196) = 59,237,472$ total connections. Similarly, the sub-sampling layer S1 involves $(1 + 1) \times 2 = 4$ trainable parameters and $(8 \times 8 + 1) \times 2 \times (98 \times 98) = 1,248,520$ connections. Subsequent layers C2 C5 and S2 S5 adhere to a comparable structural logic, culminating in a flatten layer (F) with a connection volume of 112×1 .

Table 2: Detailed settings of the convolution layers.

Layer	Kernel Size	Number of Parameters	Number of Connection
C1	16×16	196	59,237,472
S1	8×16	8	1542
C2	16×16	//	//
S2	8×8	//	//
C3	16×16	//	//
S3	8×8	//	//
C4	16×16	//	//
S4	8×8	//	//
C5	16×16	//	//
S5	8×8	//	//
F	112×1	//	//

In a fully connected (FC) layer, each neuron maintains comprehensive connectivity with all activations from the preceding layer. During the training phase of a CNN, active neurons generate class scores to facilitate the categorical classification of input data. The training loss is computed via a predefined objective function, after which the model weights are iteratively refined through backpropagation and gradient descent. Cross-entropy is frequently employed as the loss function, while Stochastic Gradient Descent (SGD) is a common optimization preference. The architecture of the fully connected layers within this CNN framework is illustrated in Fig. 6. Ultimately, the learned feature maps are flattened into a high-dimensional vector and passed through the output layer to produce the final predictions.

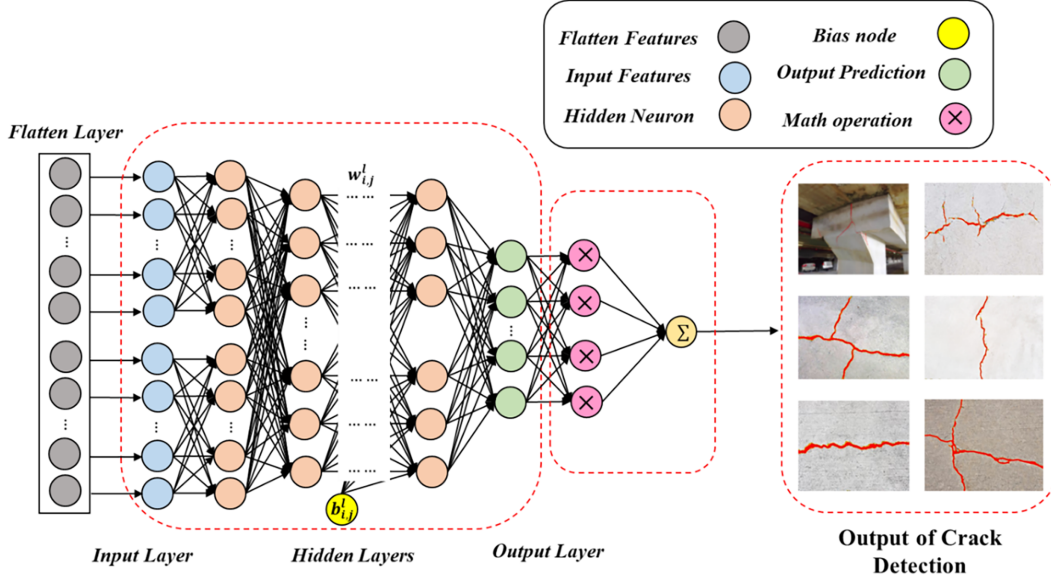


Figure 6: The fully-connected layers-type architecture of the proposed CNN.

The fully connected layer located here used for higher-level features extraction via the process of the training. Using weights matrix w_{ij}^l and biases matrix b_j^l , the layers is trained for identifying more useful nonlinear information, which can be shown like this:

$$x_j^l = f \left(\sum_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l \right) \quad (10)$$

where M_j refers to a group of input feature maps, and we also add a bias term, b , to each output feature map. The input feature maps get convolved with different kernels specific to each output feature map.

3.2 Training, Validation and Test Sets

This work proposes a deep learning-based framework aimed at detecting cracks in structural surfaces. As illustrated in Fig. 7, the main contribution here is the creation of a deep learning network (DLN). To build the DLN model, we use the Feature CNN to generate feature maps. These fused feature maps are then processed through the CNN layers to enable crack detection and segmentation—essentially outlining the cracks. The overall framework outlined in the CNN algorithm consists of two main phases: first, extracting the cracks and producing a binary map; second, measuring and estimating the crack width. In Phase 1, we start with a CNN model that operates at various image feature scales to pinpoint the approximate crack locations, followed by a step for binarization mapping. Fig. 7 illustrates the procedural workflow

of the proposed two-phase methodology, while Fig. 8 presents the schematic architecture for the CNN initialization, training, and validation phases. The critical hyperparameters utilized during the training process are detailed in Table 3. Furthermore, the performance metrics for the supervised training phase are depicted in Fig. 9. Finally, the specific implementation of the CNN training and evaluation protocols, executed within the MATLAB environment, is delineated in Algorithm 1.

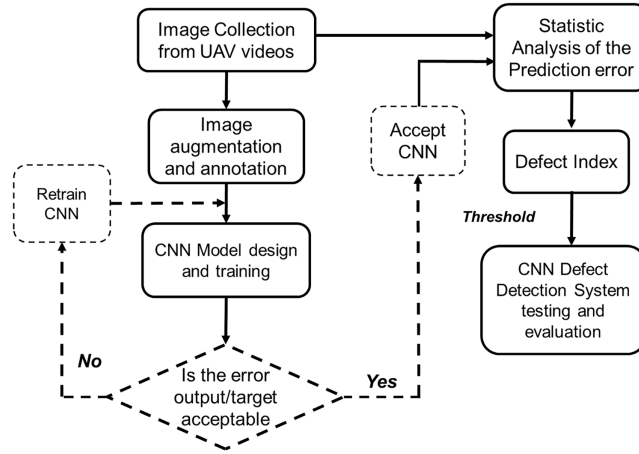


Figure 7: Overall workflow of the proposed CNN model for concrete crack detection.

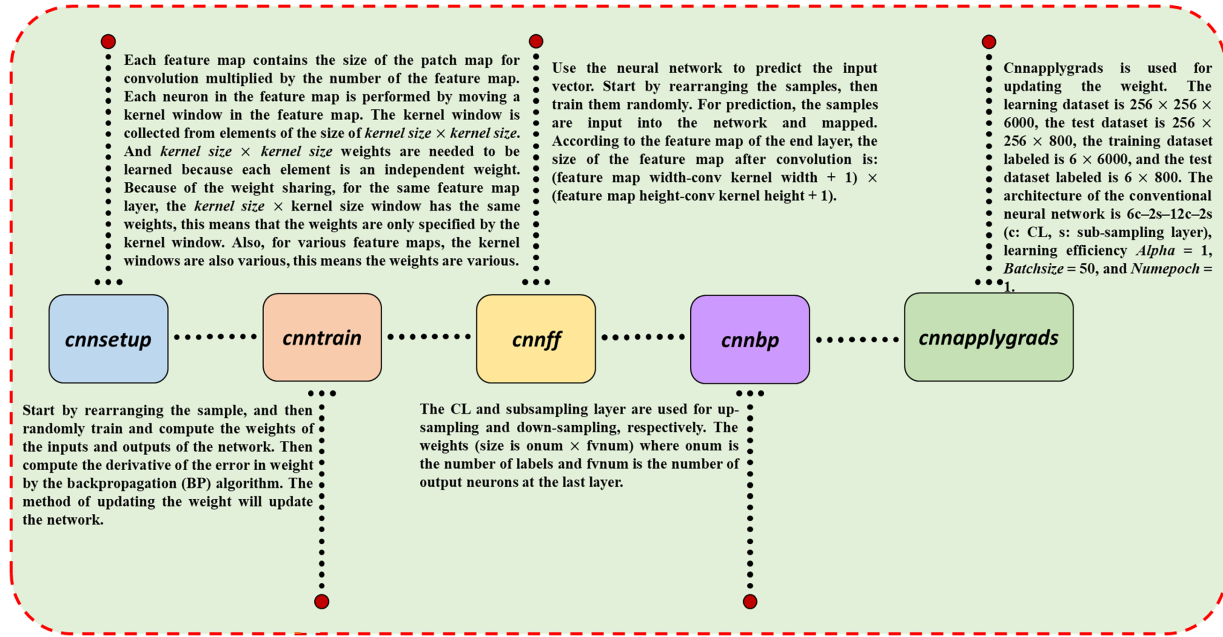


Figure 8: The CNN setup, training, and testing model functions.

Table 3: CNN key parameters.

Training Time	Gauge	Training Rate	Attenuation Factor
75 s	32	10^{-4}	10^{-6}

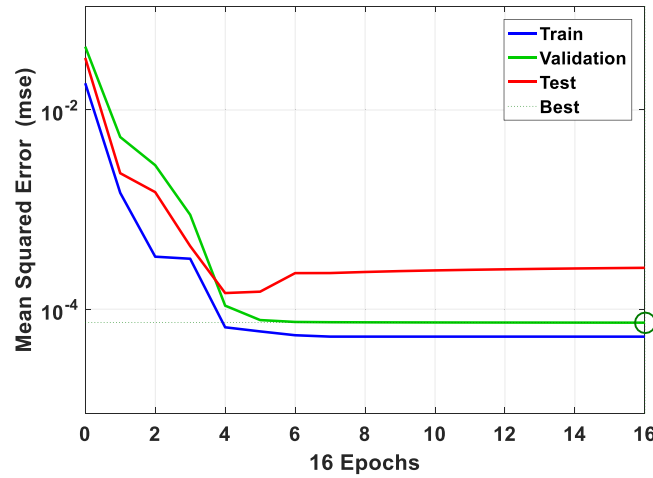


Figure 9: CNN training performance.

Algorithm 1: Training, and testing of proposed CNN

```

1: input:  $W$ : Network parameter matrix weight  $w_{ij}$  and bias  $b_j$ 
2: output: score of CNN trained model on the structure images for crack detection
3: let  $f$  be the feature set 3d matrix
4: for  $i$  in the dataset do
5:   let  $f_i$  be the feature set matrix of sample  $i$ 
6:   for  $j$  in  $i$  do
7:      $V_i \leftarrow \text{vectorize}_{(j,w)}$ 
8:     append  $V_i$  to  $f_i$ 
9:     append  $f_i$  to  $f$ 
10:   $f_{train}, f_{test}, l_{train}, l_{test} \leftarrow$  the split feature set and prediction into train subset and test subset
11:   $M \leftarrow \text{CNN}(f_{train}, l_{train})$ 
12:  score  $\leftarrow$  evaluation( $l, l_{test}, M$ )
13: return score
14: end for
15: end for

```

3.3 Crack Tip Detection Method

After we spotted the crack using CNN, the next thing was to accurately find the crack tip. Just like in Fig. 10, the method we proposed involves four main steps. First off, we used an image threshold segmentation technique to separate the crack from the background. Then, we applied some morphological operations to link up the fragmented crack. After that, we cleaned up the image to get rid of any small noise. Finally, we could pinpoint the crack tip by searching through the cleaned-up image.

3.4 Crack Contour Network (CCN) Approach

Previously, a CNN feature maps are used for training, validation, and testing on both intact and damaged structure systems. The structure images need to be processed using two-value image techniques. We found that background noise is tough to eliminate, and its impact can't be ignored. Because of this, traditional image processing methods often struggle to detect cracks. A series of operations focused on shape helps pull out the

image components that effectively define the target shape, mainly identifying key characteristics like contours and areas of the objects. The common CCN method is employed for processing these structure images, which is a good standard in digital image processing. We restore the crack shape based on the minimum gray value within the bounding box area. You can see the process illustrated in Fig. 11.

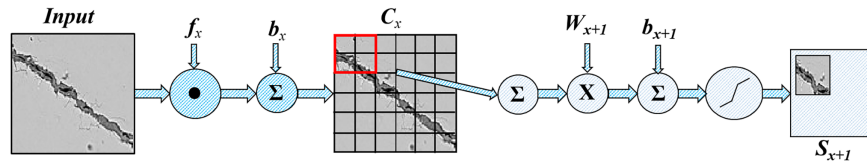


Figure 10: The crack tips detection method.

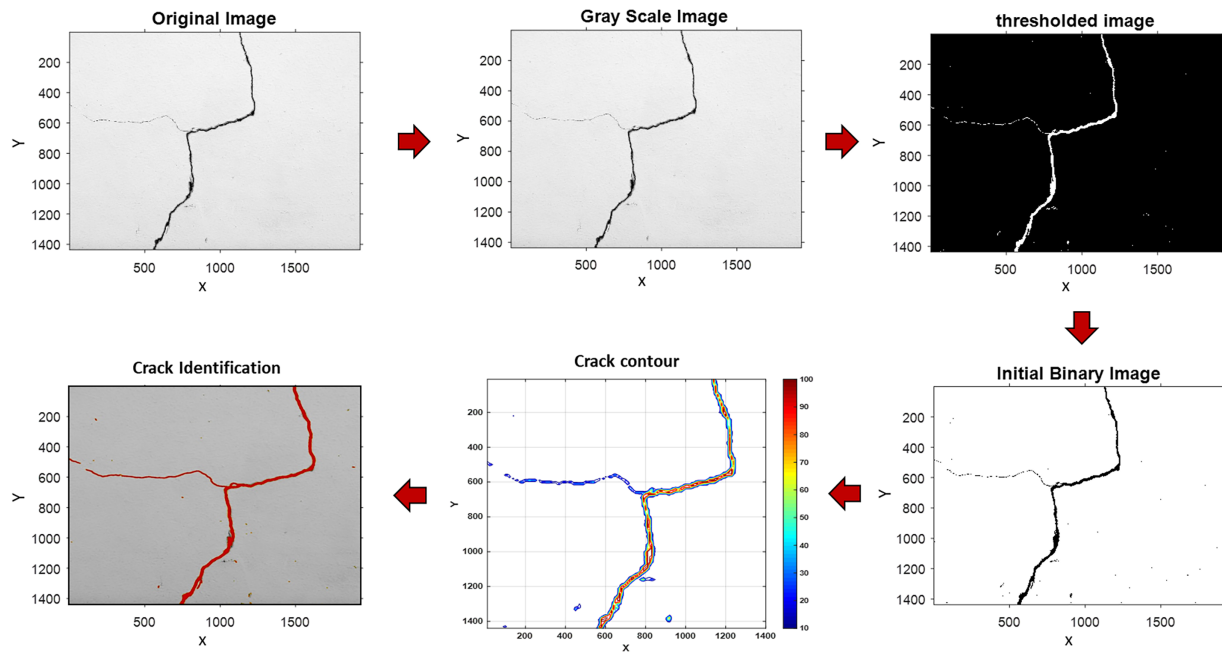


Figure 11: Structure crack extraction flowchart.

The main steps of CCN approach are as follows:

1. When it comes to cracks, one noticeable feature is the difference in gray values (ranging from 0 to 255, with black and white) on the surface of the structure background. Typically, the gray value of a crack is lower than that of the surrounding area, meaning the darkest pixel usually indicates a crack. So, we extract the crack area using CNN processing.
2. The dotted background on the structure can lead to some noise during extraction in the first step, like random black spots. To solve this, the paper focuses on filtering out these pixel areas, removing that local black noise.
3. After reducing noise in the second step, we need to gather the structure pixels again. Fig. 12 lays out the detailed steps for this third stage: 3.1 We apply Gaussian blurring (with a 9×9 convolution kernel) to create new pixels; 3.2 Then, we use low-threshold binarization (threshold set to 0) to make those new pixels more visible, allowing the cracks to extend along the structure pixels; 3.3 Next, we refine the cracks using a larger template erosion (7×7); 3.4 A smaller template (3×3) is then used to smoothen the cracks; 3.5 Finally, we apply a small template (3) once more.

4. To improve the crack shape, we calculate the average skeleton gray level based on those core skeleton values from the first step. Then we go through the pixels of the crack shape extracted earlier in the third step and compare them to the skeleton gray level. If the average value is too far off and the gray value difference exceeds a certain threshold, we remove those pixels. This helps eliminate background pixels that might have been mistakenly identified as cracks during the morphological processing.
5. By using contour detection, we can find the edges of the cracks and localize their locations in the image, as illustrated in Fig. 13. There are many interesting steps in our method for detecting and recognizing structure cracks.

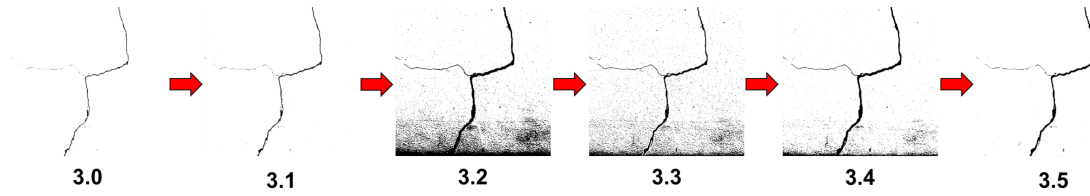


Figure 12: The structure pixels collection steps.

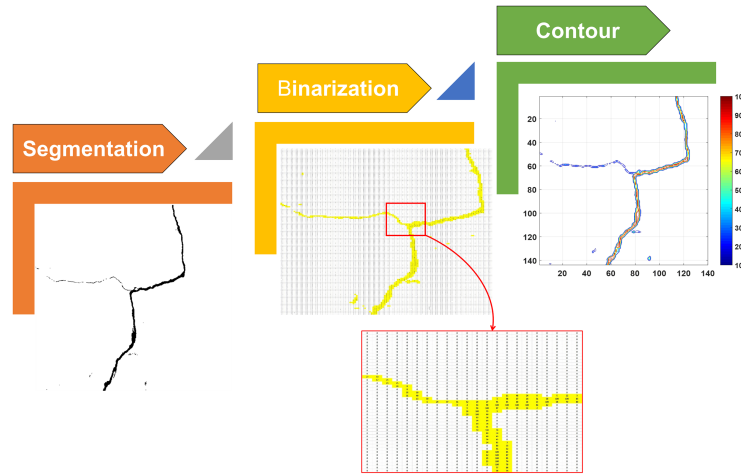


Figure 13: The crack contour in the structure analysis.

4 Results and Analysis

4.1 Concrete Crack Measurement

To calibrate our camera, we use a checkerboard pattern (check out Fig. 14). First, we need to capture several images from different angles. For us, that means taking 20 pictures while the checkerboard stays put on a table.

Next, we find the 2D coordinates that match our checkerboard's real-world coordinates. Once we have those 2D to 3D correspondences, we can get the intrinsic and extrinsic parameters of our camera, which helps us determine the actual size of the crack we've detected. As shown in Fig. 15, We gather data on the 9 point and pixel central points and then determine how they correspond to real-world locations (check out Table 4 for details).

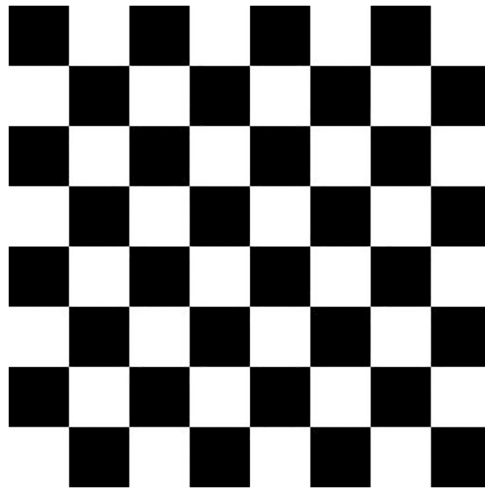


Figure 14: Checkerboard picture.

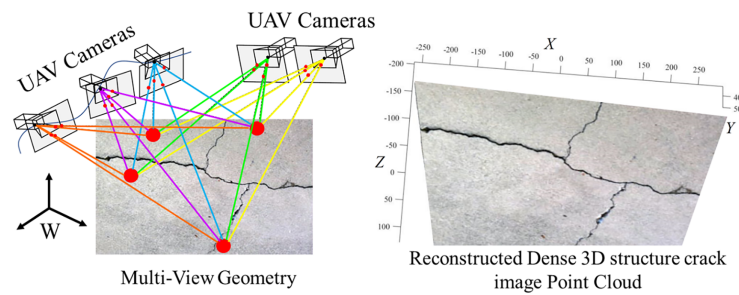


Figure 15: Collect 10 points and their pixel coordinates, then determine how they correspond to actual locations in the real world.

Table 4: Mean and standard deviation for mapping 2D and 3D coordinates.

Real Worlds Coordinates (X, Y, Z)	Pixel Coordinates (X, Y)	Mean	Standard Deviation	Error
[843, 500, 1467]	[673, 390]	173.1898	117.3097	44.1199
[404.59, 430.5, 1467]	[323, 336]	173.1898	116.1582	42.9684
[735.28, 395.90625, 1467]	[587, 309]	173.1898	128.2186	55.0288
[662.63, 392.0625, 1467]	[529, 306]	173.1898	131.7645	58.5747
[571.18, 394.625, 1467]	[456, 308]	173.1898	131.3656	58.1758
[437.16, 386.9375, 1467]	[349, 302]	173.1898	119.2854	46.0956
[600.0, 370.28125, 1467]	[479, 289]	173.1898	130.7964	57.60657
[519.83, 371.5625, 1467]	[415, 290]	173.1898	127.1486	53.95874
[487.26, 383.09375, 1467]	[389, 299]	173.1898	125.0015	51.81165
[31.31, 366.4375, 1467]	[25, 286]	173.1898	190.369	417.1792

After we give a few world points and their pixel matches, we can figure out how our camera's image plane relates to the world image plane (check out [Fig. 14](#)). This lets us calculate the image size in centimeters.

4.2 Intersection over Union (IoU)

In crack detection, Intersection over Union (IoU) metrics (also known as the Jaccard index) are used to quantify the spatial accuracy of a model's predictions—whether as bounding boxes (detection) or pixel-level masks (segmentation).

As shown in the Fig. 16, to find the IoU, by dividing the area where the predicted and actual boxes overlap (intersection area) by the total area covered by both (union area). If the IoU is high, it means a high accurate prediction; but if IoU is low, it shows that the predicted boxes don't match up well with the actual, as following:

$$\text{IoU} = \frac{\text{intersection area}}{\text{union area}} \quad (11)$$

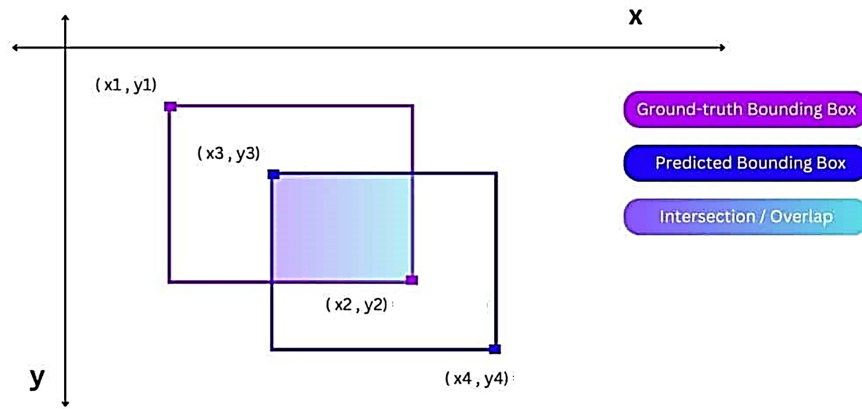


Figure 16: IoU calculation for object detection.

The intersection area is where the actual bounding box overlaps with the predicted one. To find the area where they overlap, you need to determine the coordinates of the top-left and bottom-right corners, as following:

$$\text{intersection area} = |A \cap B| = (x_2 - x_3) \cdot (y_2 - y_3) \quad (12)$$

The union area is essentially the total space taken up by both the actual bounding box and the predicted one. To find the union area, just add the areas of the two boxes together and then take away the area where they overlap as:

$$\text{union area} = |A \cup B| = |A| + |B| - |A \cap B| = (x_2 - x_1) \cdot (y_2 - y_1) + (x_4 - x_3) \cdot (y_4 - y_3) - |A \cap B| \quad (13)$$

$$|A \cup B| = [(x_2 - x_1) \cdot (y_2 - y_1)] + [(x_4 - x_3) \cdot (y_4 - y_3)] - |A \cap B| \quad (14)$$

4.3 The Monitoring Results Based UAV-Deep Learning

Previously, the CNN-CCN feature maps are trained, validated, and tested for civil structures both before and after any cracks appear is presented. In this section, the bridge crack library with 7805 crack images and 3195 non-crack images are established, which includes a pixel-level annotation dataset of concrete. The images are preprocessing by cropping and segmenting, standardizing their size, adjusting the contrast, enhancing the image data to boost the sample count, and create a database with these labeled samples. The convolution operation is analogous to a biological neuron with fixed synaptic weights. Specifically,

the convolutional layer extracts local spatial features from the preceding layer, while the subsequent sub-sampling (pooling) layer aggregates semantically similar features to enhance spatial invariance. The CNN-CCN classifier implements a bivariate detection protocol by systematically traversing the 1024×1024 image space. This architecture enables autonomous detection within each input frame, demonstrating the capacity to identify subtle surface anomalies, such as obscured potholes. Representative detection results are illustrated in Fig. 17.

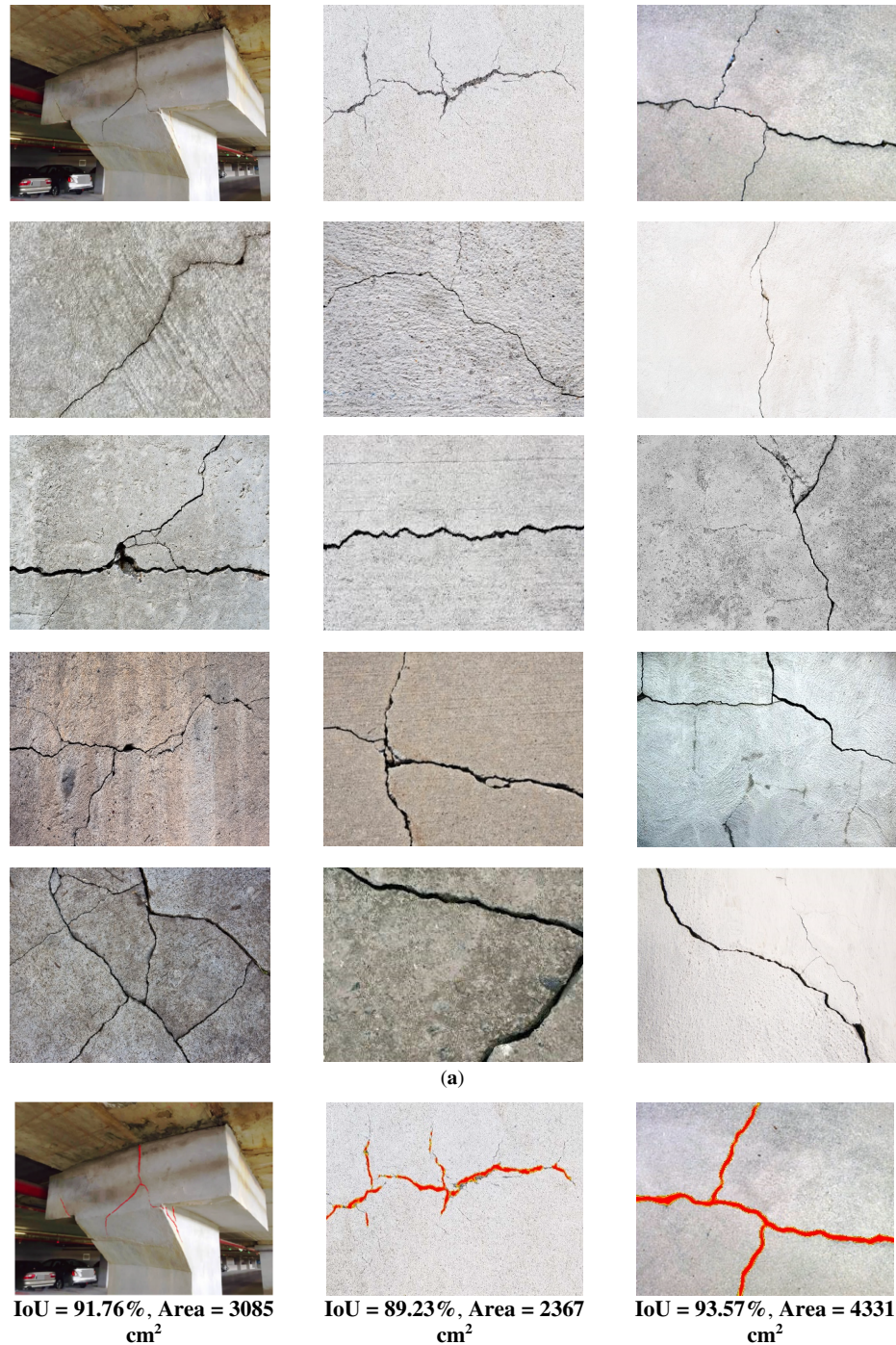


Figure 17: (Continued)

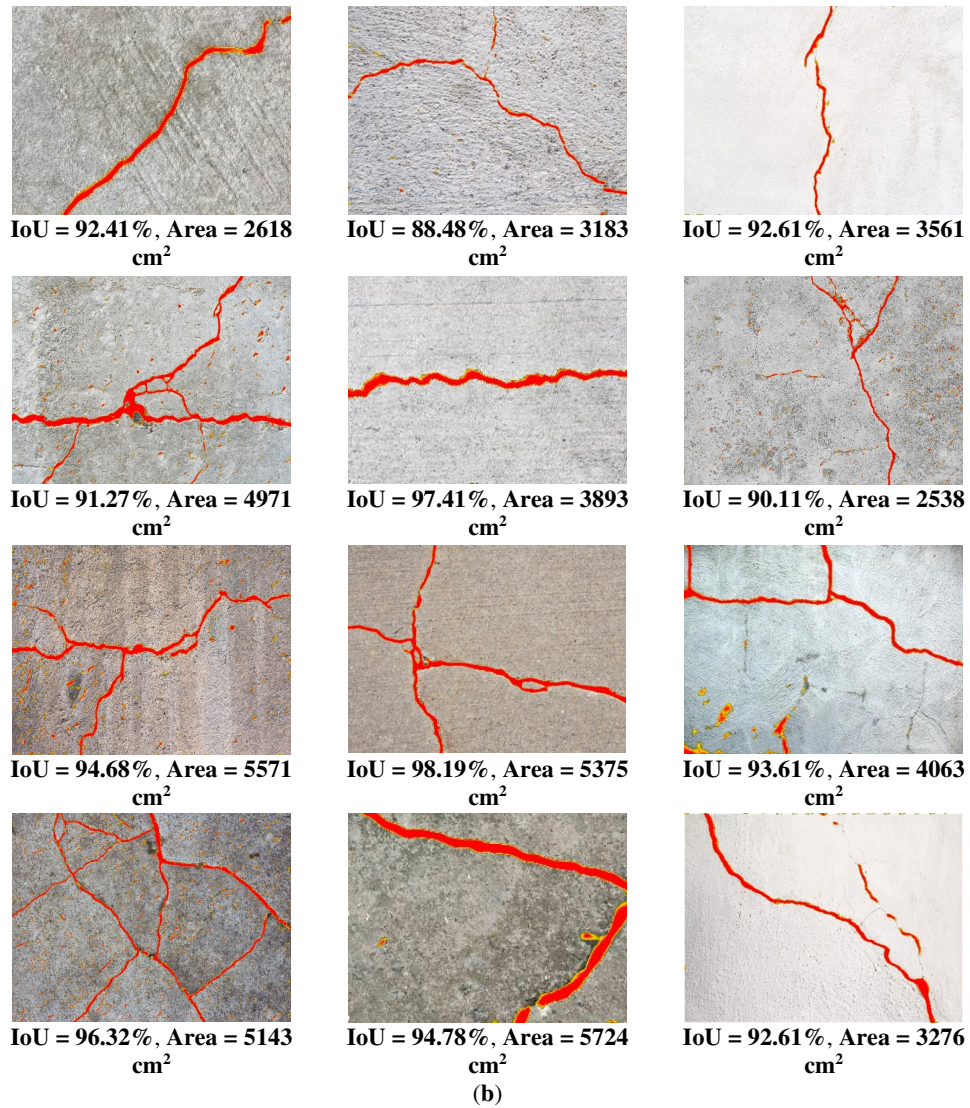


Figure 17: The sample of CNN-CCN output (a) UAV original concrete images, (b) concrete crack detection using the proposed platform.

4.4 Current Algorithm Evaluation

To evaluate how well the proposed method performs on both training and testing sets, we look at four key metrics: the true-positive rate (TPR), true-negative rate (TNR), false-positive rate (FPR), and false-negative rate (FNR). From these, we can derive three important indicators of accuracy, which include the overall accuracy, regression metrics, and the F1-score:

$$\text{accuracy rates} = \frac{N_{TPR}}{N_{TPR} + N_{FPR}} \quad (15)$$

$$\text{Regression rates} = \frac{N_{TPR}}{N_{TPR} + N_{FNR}} \quad (16)$$

$$\text{F1-score} = \frac{2N_{TPR}}{2N_{TPR} + N_{FNR} + N_{FPR}} \quad (17)$$

In this study, we prepared images and divided them into 10 scenarios to evaluate the results of mentoring on the structure like bridge, tracking outputs $f_1, f_2, \dots$, and f_{10} . We ran 1000 iterations (Epochs) to check how effective the CNN-CCN is. Fig. 18 illustrates the accuracy, regression, and F1-score values throughout the 1000 Epochs. Overall, as shown in Fig. 18a we saw accuracy rates of 96.43%, Fig. 18b for regression rates at 93.77%, and Fig. 18c for F1-score of 91.65%. These results show that our proposed method effectively recognizes mentoring in the bridge with solid performance.

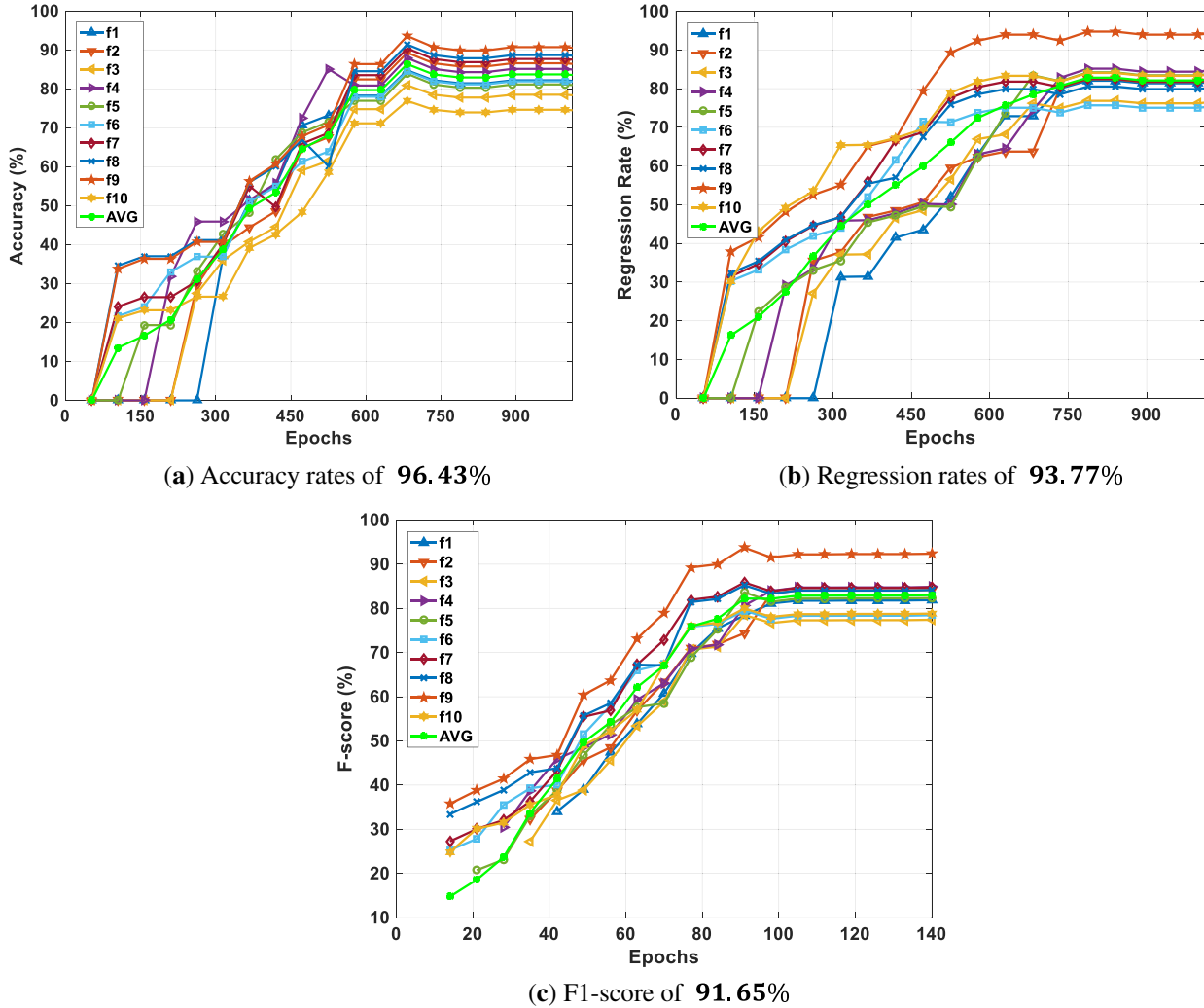


Figure 18: The training proceeded with a comparison based on the monitoring in the bridge.

In Tables 5 and 6, we can see how the CNN-CCN algorithm stacks up against three other methods for detecting concrete damage: Liu et al. [50] used using U-net fully convolutional networks, Gwon et al. [51] used artificial neural networks (ANN) and Ye et al. [52] used support vector machine (SVM).

From Table 6, it's evident that ANN tends to have lower average accuracy across all metrics, including accuracy rates, regression rates, F1-score, when compared to U-net and SVM algorithms. On the flip side, U-net approach outperforms both the SVM and ANN in these areas, while our CNN-CCN model sits at the top performance. Besides, to measure the applicability of the model, training time and model size are considered. CNN-CCN model has the smallest model size and relatively appropriate training time. However,

ANN has the largest model size and the long training time, and both U-net and SVM are lies between them. Overall, it seems pretty clear that the CNN-CCN is the standout option, consistently doing better than all the U-net, SVM and ANN across the board. This is in comparison to the U-net algorithm closest to CNN-CCN in performance, that because that the U-Net algorithm (and its variants) remain the *de facto* standard in medical imaging, and often computationally expensive and require massive amounts of data to outperform CNNs if used in concrete cracks (due to lacking inductive biases like locality).

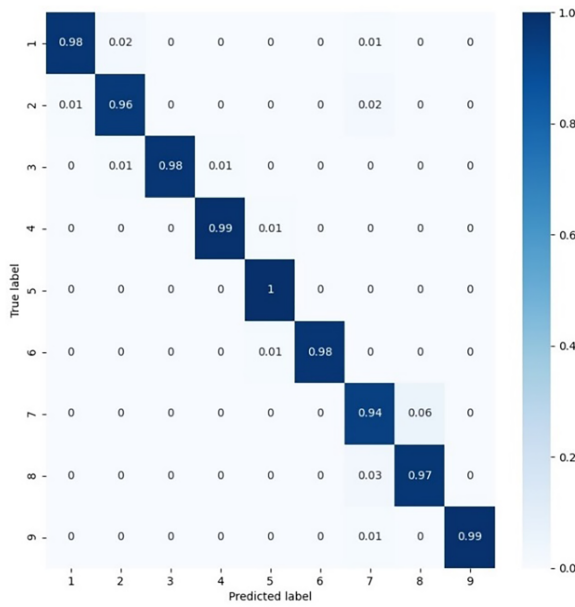
Table 5: Identification performance results for CNN-CCN and other algorithms used in literature.

Indexes	ANN	SVM	U-Net	CNN-CCN
TPR	21.34%	35.67%	46.71%	50.4%
TNR	26.78%	36.36	46.23%	50.34%
FPR	2.22%	2.64%	2.58%	2.66%
FNR	6.25%	3.75%	2.83%	2.5%

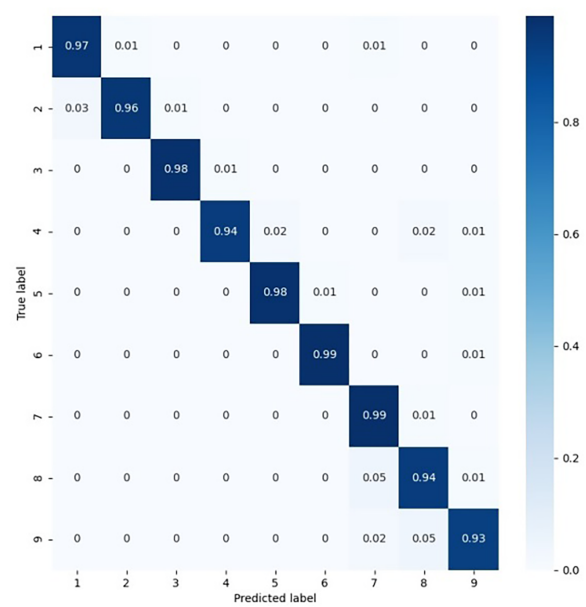
Table 6: Comparison of the test results of the CNN and other algorithms used in literature.

Performance	ANN	SVM	U-Net	CNN-CCN
Accuracy rates	90.58%	93.11%	94.96%	96.43%
Regression rates	89.66%	90.74%	91.54%	93.77%
F1-score	86.35%	88.21%	91.23%	91.65%
Training Time (sec)	112	206	83	75
Model Size (KB)	594	262	217	199

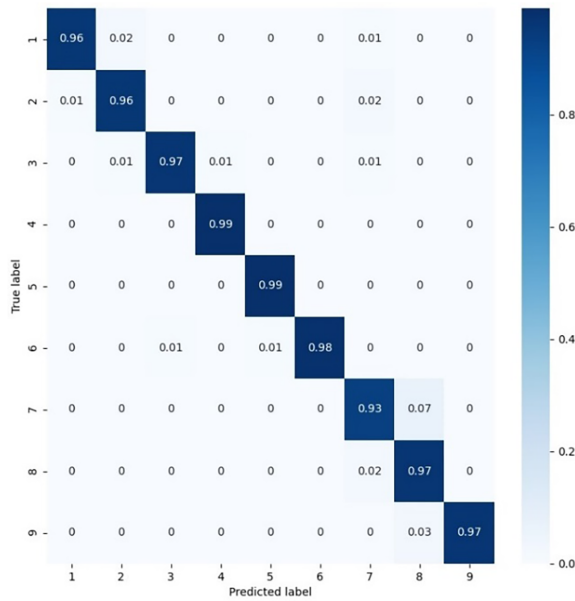
The confusion matrix comparison is presented in Fig. 19 between current method in Fig. 19a and three other methods for detecting concrete damage U-net, ANN, and SVM in Fig. 19b–d, respectively. It outlines how effectively each model identifies various scenarios of prepared images for the CNN-CCN. Based on Table 6, the confusion matrix is plotted for algorithms that have better F-score values. Just so we know, any values below 1% are considered zero in these matrices. Fig. 19 indicates that the CNN-CCN architecture is performing quite well, showing higher average accuracy and appearing to have the advantage in almost all 10 scenarios. Average accuracies are 90.58%, 93.11%, and 94.96% for the ANN, SVM and U-net, respectively.



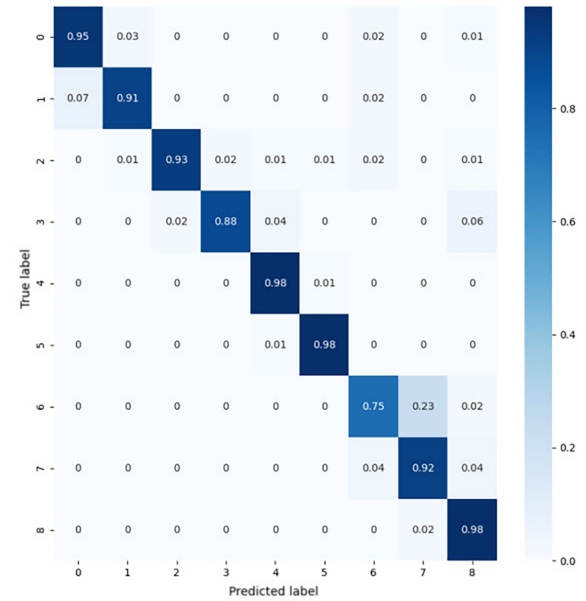
(a) CNN-CCN, Accuracy rates = 96.43%.



(b) U-net, Accuracy rates = 94.96%



(c) ANN, Accuracy rates = 90.58%



(d) SVM, Accuracy rates = 93.11%

Figure 19: The confusion matrices for of the CNN and other algorithms used in literature.

5 Conclusions

In this paper, we explored using deep learning techniques for detecting cracks in civil structures like bridges using CNNs integrated with CCN techniques through aerial images taken by UAVs. We focused on calibrating the UAV cameras to accurately match the coordinates of the UAV, the camera, and the crack images. The proposed CNN-CCN network was introduced for processing images captured by UAVs. From our analysis, we found that our learning approach is effective, even when working with a smaller dataset. The results were promising, with the proposed CNN-CCN achieving high IoU for crack detection, accuracy rates,

regression rates, and an F1-score of 96.43%, 93.77%, and 91.65%, respectively, along with a training time of 75 s and a model size of 199 KB. This demonstrates the potential of combining UAV technology with deep learning for the automated monitoring of civil structures. It stands out in the confusion matrix analysis, plus a visual comparison of crack location predicted, and actual conditions against three other traditional algorithms in literature for detecting concrete cracks, such as ANN, SVM, and U-net, highlights the distinction of the model between the different levels of damage.

6 Future Work

When we check out the research from the last twenty years, it's pretty obvious that there's been a ton of work done on different UAV technologies. This research has really pushed the envelope in how we capture and analyze concrete images. However, we still have some hurdles to overcome, particularly with image acquisition, processing, and feature extraction. Issues like environmental noise and low resolution from limited visibility angles persist as challenges in this domain. Because of these factors, collecting images often ends up being less accurate or constrained by the site conditions. Moving forward, the goal is to tackle the challenges of interpreting sparse concrete images by integrating them with visual data, and also to create more adaptable, user-friendly solutions for monitoring infrastructure. On top of that, it'd be beneficial to set some best practices for choosing scanning locations and distances to ensure we collect comprehensive data and make the scanning process smoother.

Acknowledgement: Not applicable.

Funding Statement: The author received no specific funding for this study.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Altabay WA. The advanced structural health monitoring by non-destructive self-powered wireless lightweight sensor. *Struct Durab Health Monit.* 2025;19(6):1529–45. doi:10.32604/sdhm.2025.069003.
2. Altabay WA. An artificial intelligence-based scheme for structural health monitoring in CFRE laminated composite plates under spectrum fatigue loading. *Struct Durab Health Monit.* 2025;19(5):1145–65. doi:10.32604/sdhm.2025.068922.
3. Agyemang IO, Zhang X, Adjei-Mensah I, Acheampong D, Fiasam LD, Sey C, et al. Automated vision-based structural health inspection and assessment for post-construction civil infrastructure. *Autom Constr.* 2023;156:105153. doi:10.1016/j.autcon.2023.105153.
4. Han X, Zhao Z. Structural surface crack detection method based on computer vision technology. *Build Struct.* 2018;39(S1):418–27. doi:10.2991/iceti-16.2016.1.
5. Zhang X, Long L, Broere W, Bao X. Smart sensing of concrete crack using distributed fiber optics sensors: current advances and perspectives. *Case Stud Constr Mater.* 2025;22:e04493. doi:10.1016/j.cscm.2025.e04493.
6. Fernandez I, Berrocal CG, Rempling R. Two-dimensional strain field analysis of reinforced concrete D-regions based on distributed optical fibre sensors. *Eng Struct.* 2023;278:115562. doi:10.1016/j.engstruct.2022.115562.
7. Xu Y, Bao Y, Chen J, Zuo W, Li H. Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images. *Struct Health Monit.* 2019;18(3):653–74. doi:10.1177/1475921718764873.

8. Altabay WA. A novel framework to identify delamination location/size in BFRP pipe based on convolutional neural network (CNN) algorithm hybrid with capacitive sensors. *Int J Light Mater Manuf.* 2025;8(3):393–401. doi:10.1016/j.ijlmm.2024.12.002.
9. Altabay WA. The fatigue damage monitoring of composite pipeline based on frequency domain analysis of electrical capacitance sensor system measurements. *Int J Light Mater Manuf.* 2025;8(6):779–92. doi:10.1016/j.ijlmm.2025.06.002.
10. Altabay WA. An intelligent system for pavement health monitoring using perception sensors aided deep learning algorithms. *Struct Durab Health Monit.* 2026;20(2):1–10. doi:10.32604/sdhm.2025.073949.
11. Chen Y, Li H, Zhu H, Ren T, Cao Z. Concrete bridge crack detection using unmanned aerial vehicles and image segmentation. *Infrastructures.* 2025;10:161. doi:10.3390/infrastructures10070161.
12. Yang Y, Lin Z, Liu F. Stable imaging and accuracy issues of low-altitude unmanned aerial vehicle photogrammetry systems. *Remote Sens.* 2016;8(4):316. doi:10.3390/rs8040316.
13. Dan D, Dan Q. Automatic recognition of surface cracks in bridges based on 2D-APES and mobile machine vision. *Measurement.* 2021;168:108429. doi:10.1016/j.measurement.2020.108429.
14. Chen JH, Su MC, Cao R, Hsu SC, Lu JC. A self organizing map optimization based image recognition and processing model for bridge crack inspection. *Autom Constr.* 2017;73:58–66. doi:10.1016/j.autcon.2016.08.033.
15. Liu YF, Nie X, Fan JS, Liu XG. Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction. *Comput Aided Civ Infrastruct Eng.* 2020;35(5):511–29. doi:10.1111/mice.12501.
16. Ayele YZ, Aliyari M, Griffiths D, Droguett EL. Automatic crack segmentation for UAV-assisted bridge inspection. *Energies.* 2020;13(23):6250. doi:10.3390/en13236250.
17. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); 2017 Oct 22–29; Venice, Italy.* p. 2980–8. doi:10.1109/ICCV.2017.322.
18. Chen S, Fan G, Li J, Hao H. Automatic complex concrete crack detection and quantification based on point clouds and deep learning. *Eng Struct.* 2025;327:119635. doi:10.1016/j.engstruct.2025.119635.
19. Na Y-H, Kim D-K. Deep learning strategy for UAV-based multi-class damage detection on railway bridges using U-Net with different loss functions. *Appl Sci.* 2025;15:8719. doi:10.3390/app15158719.
20. Li Q, Peng X, Zhong X, Xiao X, Wang H, Zhao C, et al. Quantitative identification of debonding defects in building façades based on UAV-thermography using a two-stage network integrating dual attention mechanism. *Infrared Phys Technol.* 2024;138:105241. doi:10.1016/j.infrared.2024.105241.
21. Shu J, Li J, Ma H. Crack detection method based on feature pyramid network for super large-scale images. *Civ Environ Eng.* 2021;43(6):1–9. doi:10.11835/j.issn.2096-6717.2021.147.
22. Jeong E, Seo J, Wacker JP. UAV-aided bridge inspection protocol through machine learning with improved visibility images. *Expert Syst Appl.* 2022;197:116791. doi:10.1016/j.eswa.2022.116791.
23. Żarski M, Wójcik B, Miszczak JA, Blachowski B, Ostrowski M. Computer vision based inspection on post-earthquake with UAV synthetic dataset. *IEEE Access.* 2022;10:108134–44. doi:10.1109/ACCESS.2022.3212918.
24. Cha YJ, Ali R, Lewis J, Büyüköztürk O. Deep learning-based structural health monitoring. *Autom Constr.* 2024;161:105328. doi:10.1016/j.autcon.2024.105328.
25. Hamishebahar Y, Guan H, So S, Jo J. A comprehensive review of deep learning-based crack detection approaches. *Appl Sci.* 2022;12(3):1374. doi:10.3390/app12031374.
26. Liu Y, Yeoh JKW, Chua DKH. Deep learning-based enhancement of motion blurred UAV concrete crack images. *J Comput Civ Eng.* 2020;34(5):04020028. doi:10.1061/(asce)cp.1943-5487.0000907.
27. Zhang Q, Ro SH, Wan Z, Babanajad S, Braley J, Barri K, et al. Automated unmanned aerial vehicle-based bridge deck delamination detection and quantification. *Transp Res Rec J Transp Res Board.* 2023;2677(8):24–36. doi:10.1177/03611981231155423.
28. Flah M, Nunez I, Ben Chaabene W, Nehdi ML. Machine learning algorithms in civil structural health monitoring: a systematic review. *Arch Comput Meth Eng.* 2021;28(4):2621–43. doi:10.1007/s11831-020-09471-9.

29. Zhang L, Yang F, Daniel Zhang Y, Zhu YJ. Road crack detection using deep convolutional neural network. In: Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP); 2016 Sep 25–28; Phoenix, AZ, USA. p. 3708–12. doi:10.1109/ICIP.2016.7533052.
30. Cha YJ, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. *Comput Aided Civ Infrastruct Eng*. 2017;32(5):361–78. doi:10.1111/mice.12263.
31. Cha YJ, Choi W, Suh G, Mahmoudkhani S, Büyüköztürk O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Comput Aided Civ Infrastruct Eng*. 2018;33(9):731–47. doi:10.1111/mice.12334.
32. Li S, Zhao X, Zhou G. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Comput Aided Civ Infrastruct Eng*. 2019;34(7):616–34. doi:10.1111/mice.12433.
33. Hsieh YA, Tsai YJ. Machine learning for crack detection: review and model performance comparison. *J Comput Civ Eng*. 2020;34(5):04020038. doi:10.1061/(asce)cp.1943-5487.0000918.
34. Jiang W, Zhou Y, Ding L, Zhou C, Ning X. UAV-based 3D reconstruction for hoist site mapping and layout planning in petrochemical construction. *Autom Constr*. 2020;113:103137. doi:10.1016/j.autcon.2020.103137.
35. Zhou Y, Wang L, Love PED, Ding L, Zhou C. Three-dimensional (3D) reconstruction of structures and landscapes: a new point-and-line fusion method. *Adv Eng Inform*. 2019;42:100961. doi:10.1016/j.aei.2019.100961.
36. Zhao S, Kang F, Li J. Concrete dam damage detection and localisation based on YOLOv5s-HSC and photogram-metric 3D reconstruction. *Autom Constr*. 2022;143:104555. doi:10.1016/j.autcon.2022.104555.
37. Deng L, Sun T, Yang L, Cao R. Binocular video-based 3D reconstruction and length quantification of cracks in concrete structures. *Autom Constr*. 2023;148:104743. doi:10.1016/j.autcon.2023.104743.
38. Chaiyasarn K, Buatik A, Likitlersuang S. Concrete crack detection and 3D mapping by integrated convolutional neural networks architecture. *Adv Struct Eng*. 2021;24(7):1480–94. doi:10.1177/1369433220975574.
39. Zhang Z. A flexible new technique for camera calibration. *IEEE Trans Pattern Anal Machine Intell*. 2000;22(11):1330–4. doi:10.1109/34.888718.
40. Davies ER. Chapter 21—image transformations and camera calibration. In: *Machine vision*. 3rd ed. Amsterdam, The Netherlands: Elsevier; 2005. p. 595–623, ISBN 9780122060939. doi:10.1016/B978-012206093-9/50024-1.
41. Sturm P. Pinhole camera model. In: Ikeuchi K, editor. *Computer vision*. Boston, MA, USA: Springer; 2014. doi:10.1007/978-0-387-31439-6_472.
42. Heikkila J, Silven O. A four-step camera calibration procedure with implicit image correction. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 1997 Jun 17–19; San Juan, PR, USA. p. 1106–12. doi:10.1109/CVPR.1997.609468.
43. Yoon J, Shin H, Kim K, Lee S. CNN- and UAV-based automatic 3D modeling methods for building exterior inspection. *Buildings*. 2023;14(1):5. doi:10.3390/buildings14010005.
44. Lyu C, Lin S, Lynch A, Zou Y, Liarokapis M. UAV-based deep learning applications for automated inspection of civil infrastructure. *Autom Constr*. 2025;177:106285. doi:10.1016/j.autcon.2025.106285.
45. Yoon S, Spencer BF Jr, Lee S, Jung HJ, Kim IH. A novel approach to assess the seismic performance of deteriorated bridge structures by employing UAV-based damage detection. *Struct Control Health Monit*. 2022;29(7):e2964. doi:10.1002/stc.2964.
46. Saida T, Rashid M, Nemoto Y, Tsukamoto S, Asai T, Nishio M. CNN-based segmentation frameworks for structural component and earthquake damage determinations using UAV images. *Earthq Eng Eng Vib*. 2023;22(2):359–69. doi:10.1007/s11803-023-2174-z.
47. Sharma P, Saurav S, Singh S. Object detection in power line infrastructure: a review of the challenges and solutions. *Eng Appl Artif Intell*. 2024;130:107781. doi:10.1016/j.engappai.2023.107781.
48. Kulkarni NN, Raisi K, Valente NA, Benoit J, Yu T, Sabato A. Deep learning augmented infrared thermography for unmanned aerial vehicles structural health monitoring of roadways. *Autom Constr*. 2023;148:104784. doi:10.1016/j.autcon.2023.104784.
49. Liu K, Chen BM. Industrial UAV-based unsupervised domain adaptive crack recognitions: from database towards real-site infrastructural inspections. *IEEE Trans Ind Electron*. 2023;70(9):9410–20. doi:10.1109/TIE.2022.3204953.

50. Liu Z, Cao Y, Wang Y, Wang W. Computer vision-based concrete crack detection using U-Net fully convolutional networks. *Autom Constr.* 2019;104:129–39. doi:10.1016/j.autcon.2019.04.005.
51. Gwon GH, Lee JH, Kim IH, Baek SC, Jung HJ. Image-to-image translation-based structural damage data augmentation for infrastructure inspection using unmanned aerial vehicle. *Drones.* 2023;7(11):666. doi:10.3390/drones7110666.
52. Ye XW, Ma SY, Liu ZX, Ding Y, Li ZX, Jin T. Post-earthquake damage recognition and condition assessment of bridges using UAV integrated with deep learning approach. *Struct Control Health Monit.* 2022;29(12):e3128. doi:10.1002/stc.3128.