

Article

Multi-Structure Crack Detection Based on a Dense U-Net++

Dong-Bum Kim, Jin-Chul Heo, Sitara Afzal and Jong-Ha Lee *

Department of Biomedical Engineering, College of Engineering, Keimyung University, Daegu 42601, Republic of Korea; 1006120@stu.kmu.ac.kr (D.-B.K.); washingbuffer@gmail.com (J.-C.H.); dr.afzalsitara@gmail.com (S.A.)

* Correspondence: segeberg@kmu.ac.kr; Tel.: +82-10-8968-8769

Abstract

Cracks indicate the deterioration of civil engineering structures, and early detection through regular inspection is crucial for structural safety. However, traditional manual inspection is time- and labor-intensive and depends heavily on the inspector's expertise. Although deep learning-based crack detection has been actively studied, most prior studies focus on a single environment such as asphalt or concrete, and research integrating asphalt, concrete, and tunnel environments remains scarce. In this study, we propose a Dense U-Net++-based model trained on a unified dataset of drone-captured crack images across all three environments. The model combines dense connections with U-Net++'s nested skip pathways to mitigate the semantic gap in the encoder–decoder structure, enabling effective feature fusion and precise restoration of fine crack boundaries. Focal loss addresses the severe class imbalance between background and crack pixels, and area-based postprocessing suppresses spurious detections. The proposed method achieved a precision of 95.47%, a recall of 92.35%, an F1-score of 93.91%, and an IoU of 88.58%, outperforming both the baseline Dense U-Net++ and Mask R-CNN. Qualitative evaluation confirmed reliable detection across rough asphalt textures, striped concrete patterns, and low illumination in tunnels. These results indicate that the proposed framework achieves promising crack segmentation performance for automated structural inspection.

Keywords: crack detection; feature fusion; U-Net++; dense connection; deep learning

1. Introduction

Cracks are among the most commonly observed defects in civil engineering structures [1], and they are an important indicator of structural deterioration [2]. If they are left unaddressed, crack propagation can ultimately lead to severe structural damage or catastrophic collapse [3]. Therefore, early crack detection through regular inspection and timely maintenance is essential for predicting deterioration trends and extending the service life of structures [2,4]. Traditionally, manual inspection has been employed for periodic structural assessments [2]. However, these inspections require trained experts and specialized equipment, making the process time-consuming, labor-intensive, and costly [5]. Moreover, the reliability of inspection outcomes is often influenced by the inspector's expertise and experience [3,5]. To overcome these limitations, automated crack detection methods have attracted increasing attention, including image processing techniques [6] and machine learning-based approaches [7,8]. However, their performance is often affected by changes in image illumination conditions, and they may have difficulty extracting complex and discriminative features from crack images [6,7,9].



Academic Editor: Guanqiu Qi

Received: 29 June 2026

Revised: 27 July 2026

Accepted: 28 July 2026

Published: 1 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

Recently, Convolutional Neural Network (CNN)-based deep learning methods have been increasingly applied to crack detection. Bui et al. reported that deep learning-based methods, including CNN and recurrent neural networks, achieved superior object recognition performance compared with conventional machine learning-based methods such as random forest, support vector machine and SP-HPM [10]. Liu et al. adopted a U-Net-based approach [11] for concrete crack detection, demonstrating high efficiency, strong robustness and higher accuracy than conventional fully convolutional networks (FCNs) [12]. In particular, U-Net effectively preserves detailed spatial information through its encoder–decoder structure and skip connections, and it has been widely used in the field of biomedical image segmentation, where it shows excellent performance. Thus, CNN-based methods have shown excellent performance in pixel-level image segmentation tasks such as crack detection. Khan et al. utilized a model based on Nested U-Net (U-Net++) [13], an extended variant of U-Net, to improve fine crack detection performance in images captured by drones and handheld cameras. In particular, by incorporating an efficient encoder, residual blocks [14], and a convolutional block attention module (CBAM) [15] into an Efficient Nested Res-U-Net structure, they achieved both high crack segmentation performance and computational efficiency [2]. However, existing crack detection studies have mainly focused on asphalt [16] or concrete structures [5,11,17,18], using either a single dataset or a combined dataset of these two materials [19]. Unlike asphalt and concrete structures, tunnel environments [20] present challenging conditions, such as low illumination and non-uniform lighting, and have therefore been treated as separate research. For these reasons, studies that integrate asphalt, concrete, and tunnel environments for unified training and analysis remain limited.

In addition, drone-based approaches for efficiently capturing large structural areas and detecting cracks have recently attracted increasing research attention [2,18]. Drone-based imaging enables the safe and efficient acquisition of crack images from structures that are difficult to access; however, crack segmentation performance can be degraded by variations in imaging distance, illumination, shadows, and complex backgrounds [18]. Therefore, in this study, we train the proposed model using an integrated drone-captured crack dataset consisting of asphalt, concrete, and tunnel environments, and propose a Dense U-Net++-based model that achieves stable crack segmentation performance across diverse environments and imaging conditions.

The main contributions of this study are summarized as follows:

- We constructed a comprehensive dataset by integrating crack images captured from asphalt, concrete, and tunnel environments.
- We propose a Dense U-Net++-based crack segmentation network that integrates nested skip connections with dense blocks to enhance feature representation.
- The proposed model effectively fuses encoder and decoder features, enabling accurate restoration of crack shapes and boundary details.
- The proposed model achieves stable crack segmentation performance across various structural surfaces, complex backgrounds, illumination changes, and imaging conditions.

The remainder of this paper is organized as follows. Section 2 describes the proposed Dense U-Net++-based crack segmentation framework and the integrated dataset used in this study. Section 3 presents the experimental setup, evaluation metrics, and comparative results. Section 4 discusses the experimental findings and provides further analysis. Finally, Section 5 concludes the paper and discusses potential directions for future research.

2. Proposed Methodology

In this study, we introduce a crack detection model for drone-captured images acquired from asphalt, concrete, and tunnel environments. The main objective of the proposed

method is to achieve stable crack segmentation across various structural surfaces, complex backgrounds, illumination variations, and imaging conditions. Figure 1 illustrates the overall workflow of the proposed method, including drone-based image acquisition, crack image collection from different environments, model-based crack segmentation, and crack mask generation.

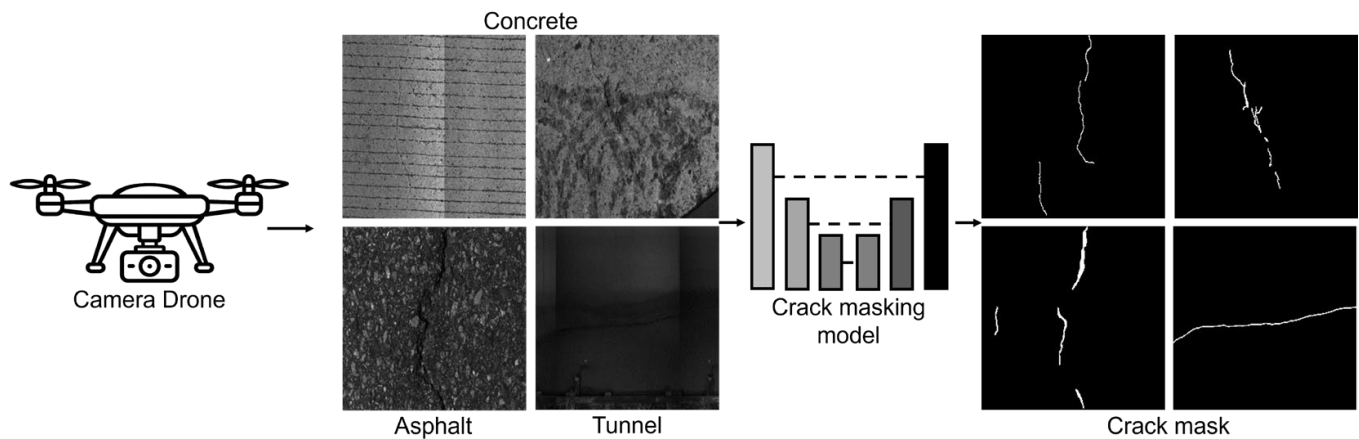


Figure 1. Overall workflow of the proposed crack segmentation approach.

2.1. Preprocessing

Image preprocessing is a crucial step in applying deep learning models to image analysis tasks [21]. To improve the efficiency and performance of model training, we performed a three-stage preprocessing technique. Drone-acquired images are typically high-resolution to capture detailed crack morphology. Therefore, without appropriate preprocessing, the computational load of the model can increase substantially. To address this issue, (1) we first cropped the original images into patches of 1024×1024 pixels. (2) The extracted patches were then resized to 256×256 pixels to match the model input size and improve computational efficiency. Directly cropping the original drone images to 256×256 would generate an excessive number of patches per image, increasing the computational burden, and would fragment individual cracks across multiple patches, hindering the learning of their continuous morphology. The resized patches preserved the overall crack patterns, allowing them to be classified into crack-containing and non-crack patches. Many of the generated patches contain only background without visible cracks. As a result, using all generated patches for training may lead to a class imbalance problem. (3) Therefore, crack-containing patches were selected to construct the final training dataset.

2.2. Focal Loss

Crack segmentation aims to identify crack regions within an image at the pixel level. In general, crack pixels occupy only a small proportion of the entire image. Therefore, when cross-entropy loss is used, the model tends to be biased toward the dominant background class, resulting in a severe class imbalance problem between background and crack pixels. To alleviate this issue, focal loss was adopted as the loss function. As shown in Equation (1), focal loss extends cross-entropy loss by assigning greater importance to hard-to-classify samples, which is particularly effective for detecting thin and boundary-ambiguous cracks.

$$\text{Focal Loss} = -(\alpha(1-p)^{\gamma}\log(p) + (1-\alpha)(p^{\gamma})\log(1-p)) \quad (1)$$

Here, p is the probability predicted by the model for the crack class, and γ is the focusing parameter that reduces the loss contribution of easily classified background pixels. As the predicted crack probability increases, the term $(1-p)^{\gamma}$ becomes small, thereby

reducing the loss contribution of easily classified pixels. In contrast, when the predicted crack probability is low, this term remains large, allowing the model to focus more on difficult crack pixels. In this way, the model reduces the influence of easily classified pixels and focuses more on crack regions that are prone to misclassification.

In addition, α is a weighting factor used to address class imbalance. A larger weight is assigned to the crack class, which contains fewer pixels, whereas a smaller weight is assigned to the background class. This balances the loss contribution between the two classes. As a result, focal loss accounts for both classification difficulty and class imbalance, rather than treating all pixels equally. This improves the detection of crack regions, which occupy only a small portion of the image.

2.3. Dense U-Net++ Architecture

The proposed model is a crack segmentation network that integrates the nested skip-pathways design of U-Net++ with dense convolutional blocks inspired by DenseNet [22]. This architecture connects the encoder and decoder through nested skip pathways to enhance feature propagation. Unlike conventional skip connections that directly transfer encoder features to the decoder, U-Net++ uses nested and dense skip pathways to reduce the semantic gap between encoder and decoder features and to improve multi-scale feature learning. From the perspective of crack segmentation, cracks appear at varying scales and often exhibit low contrast against the background, where both multi-scale feature fusion and a reduced semantic gap are effective for accurate detection. Building upon this U-Net++ backbone, we introduce dense connections into the convolutional blocks within the skip pathways, allowing features from previous stages to be reused. This strengthens feature propagation, improves representation capability, and helps preserve fine crack details. Figure 2 illustrates the architecture of the proposed Dense U-Net++ model, including the nested encoder–decoder structure, dense skip pathways, and dense convolutional blocks used for feature refinement.

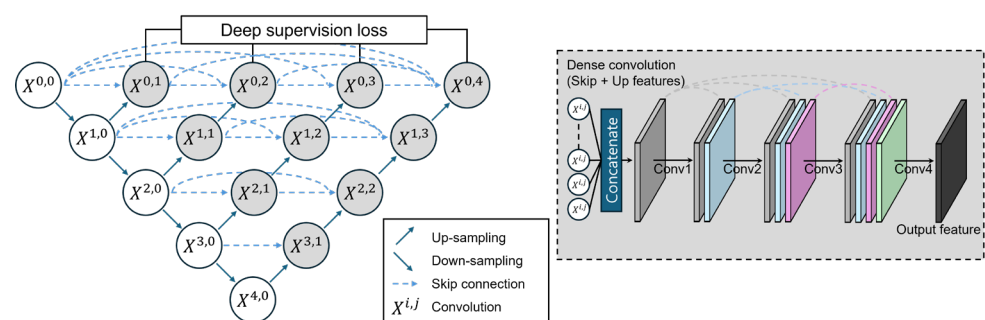


Figure 2. Architecture of the proposed Dense U-Net++ crack segmentation model. White circles denote standard convolutional layers, and gray circles denote the dense layers incorporated into the skip pathways.

2.3.1. U-Net++-Based Crack Detection Algorithm

This dataset consists of crack images collected from different surface materials and imaging environments. Surface texture, image contrast, and crack size and morphology vary across different environments. In particular, drone-captured images often include variations in acquisition distance and viewing angle, resulting in cracks appearing at different scales. To address variations in environments and scales, U-Net++ adopts nested skip connections instead of the single skip connections used in the conventional U-Net. Each node receives encoder features from the same depth and up-sampled decoder features from lower levels. This nested structure progressively fuses multi-depth features between the encoder and decoder. Compared with single-scale feature representations, this design

captures crack information across different environments and spatial scales more effectively. By learning multi-scale features during training, the model achieves more stable crack segmentation under challenging conditions, including rough asphalt textures and low-contrast cracks in concrete and tunnel environments.

2.3.2. Dense Blocks in the Skip Pathways

Dense blocks were originally introduced in DenseNet. They are configured such that each layer receives and concatenates the feature maps from all preceding layers [22]. By accumulating and reusing features from preceding layers, dense blocks improve parameter efficiency and alleviate the vanishing-gradient problem in deep networks. This helps maintain stable gradient propagation during training. In this study, dense connections were incorporated into the skip pathways of U-Net++. The original skip pathways, which consist of standard convolutional layers, are replaced with dense convolutional blocks. These dense skip pathways help reduce the semantic gap between encoder and decoder features. Encoder features preserve spatial details but contain limited semantic information. Decoder features, in contrast, are semantically rich but often lose fine details because of their low spatial resolution. The dense convolutional blocks progressively integrate skip features with up-sampled decoder features instead of relying only on simple concatenation. Repeated feature fusion and refinement enhance crack boundary reconstruction. As a result, fine crack boundaries can be reconstructed more accurately, enabling crack regions to be distinguished even in noisy or low-contrast environments.

2.4. Postprocessing with Connected Components

During crack segmentation, the model may detect not only actual crack regions but also surrounding noise caused by the imaging environment. Such noise can result from illumination non-uniformity, surface contamination, and shadows, and it appears in the detection results as small point-shaped objects unrelated to actual cracks, becoming a major cause of false positives (FP). To alleviate this, we performed a postprocessing technique. Connected component analysis was performed on the output mask of the model to compute the area of each object. Objects with an area of 10 pixels or less were judged to be noise and removed. The threshold was determined experimentally and set conservatively so that point-shaped noise could be suppressed without reducing the recall of the model. Because cracks have a thin, long, continuous shape with an area above a certain level, whereas noise appears as independent point shapes with a very small area, area-based filtering can effectively distinguish cracks from noise. Through this postprocessing technique, FP were reduced and detection performance was improved.

2.5. Evaluation Metrics

To evaluate the proposed method, the segmentation results were quantitatively assessed using several metrics. After all models were trained on the same dataset and computing environment, the performance of Mask R-CNN, the proposed Dense U-Net++, and Dense U-Net++ with postprocessing was compared.

The confusion matrix is a tool for evaluating the performance of segmentation models and shows the relationship between predicted values and actual observed values. It provides valuable insights into model performance across various categories and helps identify potential types of errors. In this evaluation, precision, recall, F1-score, and IoU were used to compare the results of the models. Each metric was calculated for the crack and background classes separately, as defined in Equations (2)–(5), and then averaged over the two classes to obtain the macro-averaged value. The metrics were computed for each of the 200 test images and averaged over all test images. Therefore, the values reported correspond to the macro-averaged metrics across the two classes and the 200 test

images. Precision represents the proportion of samples that are actually positive among all cases predicted as positive, as shown in Equation (2). A higher value indicates that the model more effectively suppressed FP, thereby reducing the misclassification of background regions and noise as cracks.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall quantifies the proportion of actual positive samples that the model correctly identified, as shown in Equation (3). A higher value indicates that the model detected more actual crack regions with fewer missed detections.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

The F1-score is the harmonic mean of precision and recall, and it was used to consider the performance of both metrics simultaneously, as shown in Equation (4). A high value indicates that the model achieved a balanced performance between suppressing false positives and detecting crack regions.

$$F1\text{-score} = 2TP / (2TP + FP + FN) \quad (4)$$

The IoU measures the overlap between the predicted crack region and the ground-truth region relative to their union, as shown in Equation (5). A high value indicates that the predicted mask closely matches the ground truth in both extent and position, since IoU simultaneously penalizes false positives and false negatives.

$$IoU = \frac{TP}{TP + FP + FN} \quad (5)$$

For all evaluation metrics, 1 denotes the highest score, and values closer to 0 denote lower performance. Here, true positives (TP) denote samples correctly predicted as positive, FP denote negative samples incorrectly predicted as positive, and false negatives (FN) denote positive samples incorrectly predicted as negative [18,23,24].

3. Experimentation and Results

To evaluate the effectiveness of the proposed method, its performance was compared with Mask R-CNN [25], Dense U-Net++ without postprocessing, and Dense U-Net++ with postprocessing. The comparison was conducted using the same dataset split and training conditions to ensure a fair evaluation. The proposed model was assessed in terms of crack segmentation accuracy, performance consistency across different structural environments, and its ability to reduce false-positive regions. The dataset configuration, implementation details, evaluation metrics, and experimental results are described in the following subsections.

3.1. Dataset

In this study, we used a self-constructed crack dataset. The original images were drone-captured images collected from various parts of bridge structures, including decks and piers, as well as a tunnel lining in the Republic of Korea, covering three structural environments: concrete, asphalt, and tunnel. The image collection procedure followed the inspection guidelines of the Korea Expressway Corporation. The images were acquired using a DJI Matrice 300 RTK unmanned aerial vehicle equipped with a DJI Zenmuse H20 camera (both from DJI, Seoul, Republic of Korea). Images were captured within 5 m of the target surface, from multiple viewing directions depending on the structural component. The dataset includes variations in surface texture, crack morphology, illumination

conditions, background complexity, and imaging distance, making it suitable for evaluating crack segmentation performance under practical inspection conditions. The original images were preprocessed before being used as network inputs for training. As described in Section 2.1, the preprocessing procedure included patch extraction and image resizing to match the required input size of the proposed model. We manually labeled the crack masks using the computer vision annotation tool (CVAT). Ambiguous regions such as shadows, surface stains, and structural joints were not labeled as cracks unless a continuous crack morphology was clearly visible. After preprocessing, the dataset consisted of 5000 crack-containing patches of size 256×256 pixels. Among these, 200 patches were held out as the test set, evenly drawn from four surface conditions (asphalt, rough-textured concrete, striped-pattern concrete, and tunnel linings) with 50 patches each. The remaining 4800 patches, comprising the three structural environments (concrete, asphalt, and tunnel) in similar proportions, were divided into training and validation sets at a ratio of 8:2, resulting in 3840 training and 960 validation patches. The test set was separated at the original-image level before patch extraction, so that patches from the same original image were not shared across subsets, and it was not used during training or validation. The principal characteristics of the self-constructed crack dataset are summarized in Table 1.

Table 1. Principal characteristics of the self-constructed crack dataset.

Item	Description
Image acquisition	Drone-captured images
Structural environments	Concrete, asphalt, tunnel surfaces
Total number of patches	5000
Original image format	RGB, high-resolution
Preprocessing	(1) Patch extraction (1024×1024), (2) Resizing to 256×256 , (3) Manual selection of crack-containing patches
Postprocessing	Area-based noise removal (≤ 10 px)

3.2. Implementation Details

The proposed model was implemented using the PyTorch framework (version 1.13.1) and trained in an environment equipped with an AMD Ryzen 5 5600X CPU (AMD, Santa Clara, CA, USA) and a NVIDIA RTX 3080Ti GPU (NVIDIA, Santa Clara, CA, USA). All networks were trained from randomly initialized weights. To ensure a fair comparison, all compared models were trained using the same training, validation, and test splits. In all experiments, the Adam optimizer [26] was used with an input image size of 256×256 pixels and a batch size of 20. The initial learning rate was set to 0.001 and decayed by a factor of 0.9 every 300 epochs. Training was performed for 3000 epochs. The proposed model was trained with focal loss, whereas Mask R-CNN was trained using its original multitask loss. The focal loss was applied with $\alpha = 0.25$ and $\gamma = 2.0$, and the predicted probability maps were binarized using a threshold of 0.5. A fixed random seed was not used, and repeated experiments were not conducted. Only input normalization was applied, and no other data augmentation was performed. The model with the highest validation performance was selected and used for comparison. The final performance was evaluated on the independent test set using quantitative segmentation metrics and visual comparison of the predicted crack masks. The detailed hyperparameters and experimental settings are summarized in Table 2.

Table 2. Hyperparameters and experimental settings. All models trained using identical settings.

Item	Setting
Input image size	256×256
Batch size	20
Total epochs	3000
Initial learning rate	0.001
Learning rate decay	$\times 0.9$ every 300 epochs
Optimizer	Adam
Loss function	Focal loss ($\alpha = 0.25, \gamma = 2.0$)
Binarization threshold	0.5

3.3. Quantitative Results

To evaluate the performance of the model and postprocessing step in detail, predictions were generated for 200 test images with a size of 256×256 pixels. All test samples were new data that were not used during training or validation and had not been seen before testing. Before visually analyzing the test images, precision, recall, F1-score and IoU were calculated by comparing the predicted masks with the corresponding ground-truth masks. Figure 3 presents box plots showing the distributions of precision, recall, F1-score, and IoU for each method across the 200 test images.

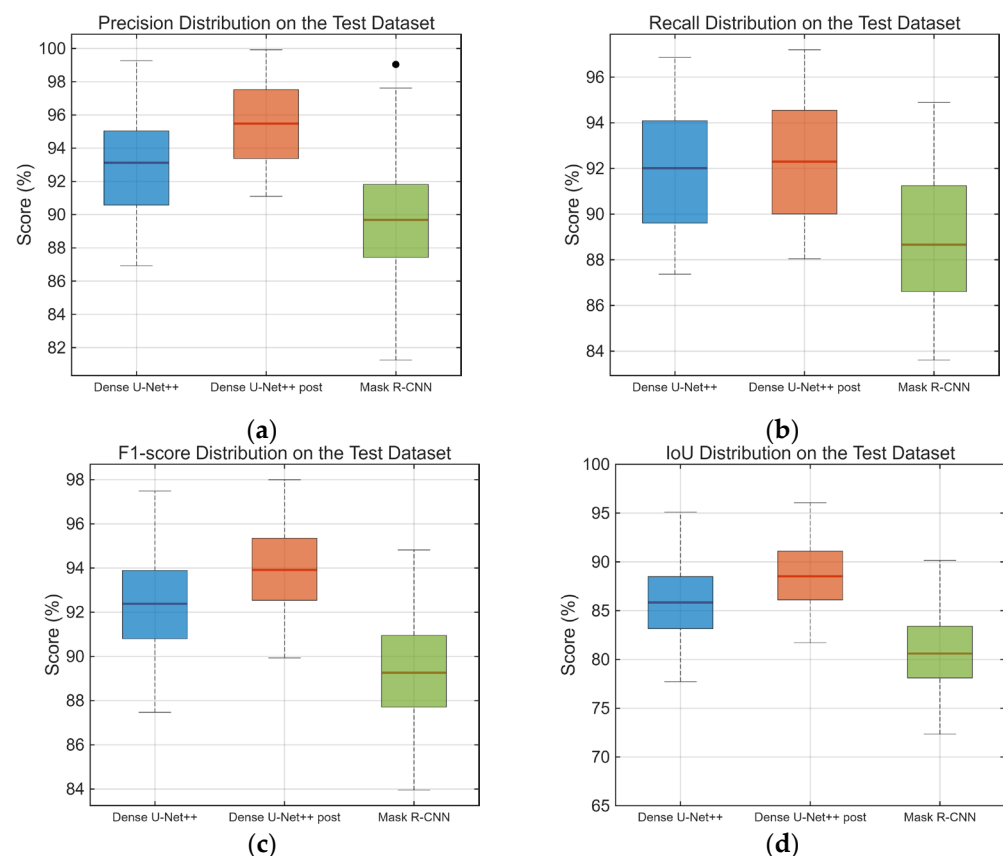


Figure 3. Performance distribution analysis of the three methods on test dataset. All metrics are macro-averaged over the crack and background classes. For all metrics, Dense U-Net++ with postprocessing showed statistically significant improvements over both Dense U-Net++ and Mask R-CNN ($p < 0.001$). (a) Precision distribution. The dot represents an outlier in the distribution for the Dense U-Net++ model, (b) recall distribution, (c) F1-score distribution, (d) IoU distribution.

The proposed method (Dense U-Net++ with postprocessing) exhibits the highest median and the narrowest distribution, whereas Mask R-CNN shows the lowest median and the widest distribution (Figure 3a). In terms of recall, Dense U-Net++ and Dense U-Net++ with postprocessing show similar distributions, both outperforming Mask R-CNN (Figure 3b). A similar trend is observed for the F1-score. Dense U-Net++ with postprocessing achieves the highest median with the smallest variation, whereas Mask R-CNN exhibits the lowest performance and the widest distribution among the three methods (Figure 3c). The IoU distribution follows the same tendency, with Dense U-Net++ with postprocessing showing the highest median and Mask R-CNN the lowest (Figure 3d). The 95% confidence intervals of the three methods did not overlap for any metric, and the improvements of the proposed method over both Dense U-Net++ and Mask R-CNN were statistically significant (Wilcoxon signed-rank test, $p < 0.001$). No correction for multiple comparisons was applied, as all p -values were far below the significance threshold.

The average performance corresponding to the box plot results is shown in Table 3. Among the three methods, the combination of the proposed model and postprocessing showed the highest performance across all metrics, followed by Dense U-Net++ (single) and then Mask R-CNN.

Table 3. Average precision, recall, F1-score, and IoU of the three methods on the test dataset. All metrics are macro-averaged over the crack and background classes and reported as mean $\pm$ standard deviation across the 200 test images, with 95% confidence intervals in parentheses estimated using a normal approximation.

Methods	Precision	Recall	F1-Score	IoU
Mask R-CNN	89.80 $\pm$ 3.26 (89.35–90.25)	88.91 $\pm$ 2.83 (88.52–89.31)	89.35 $\pm$ 2.15 (89.06–89.66)	80.83 $\pm$ 3.52 (80.34–81.32)
Dense U-Net++	92.91 $\pm$ 2.92 (92.51–93.32)	91.94 $\pm$ 2.73 (91.56–92.32)	92.42 $\pm$ 2.08 (92.14–92.72)	85.99 $\pm$ 3.60 (85.49–86.49)
Dense U-Net++ with postprocessing	95.47 $\pm$ 2.45 (95.13–95.81)	92.35 $\pm$ 2.71 (91.97–92.73)	93.91 $\pm$ 1.86 (93.65–94.17)	88.58 $\pm$ 3.31 (88.12–89.04)

3.4. Qualitative Results

To confirm the qualitative performance of the proposed method, the prediction results across the three environments (asphalt, concrete, tunnel) were visually compared. Each figure shows, in order, the original image, the ground-truth mask, and the prediction mask of the proposed method.

Figure 4 shows the prediction results in the asphalt environment. The asphalt surface has clear contrast and small grains. However, the background has dark characteristics similar to cracks. Cracks that resemble the background can be a factor that hinders detection, but because sufficient contrast is secured, the proposed model distinguishes cracks and background relatively clearly. In some images, cracks were detected in additional branching regions, but the overall shape and path show good agreement with the ground-truth mask.

Figure 5 shows the prediction results in a concrete environment with a rough texture. Rough-textured concrete has bright contrast, and its grains appear coarse. The coarse grains have a color tone similar to cracks and are therefore prone to being misidentified. Nevertheless, the proposed method effectively distinguished small grains from actual cracks and accurately detected cracks in a shape similar to the ground-truth mask. This demonstrates that the model learns not only simple contrast but also the continuous and linear morphological features of cracks, enabling it to stably cope with noise arising on rough surfaces in these test images.

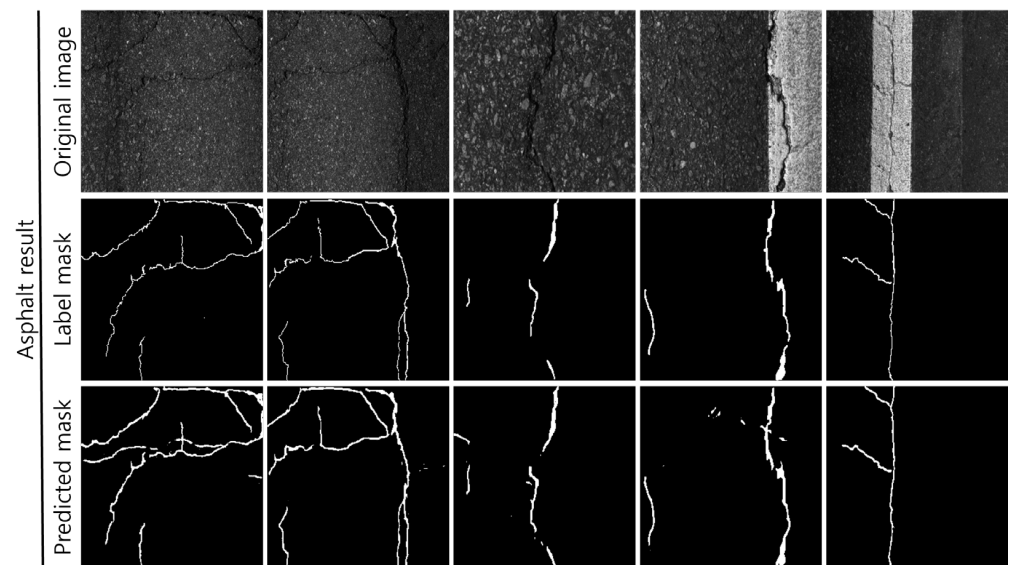


Figure 4. Prediction results of the proposed method on asphalt test images.

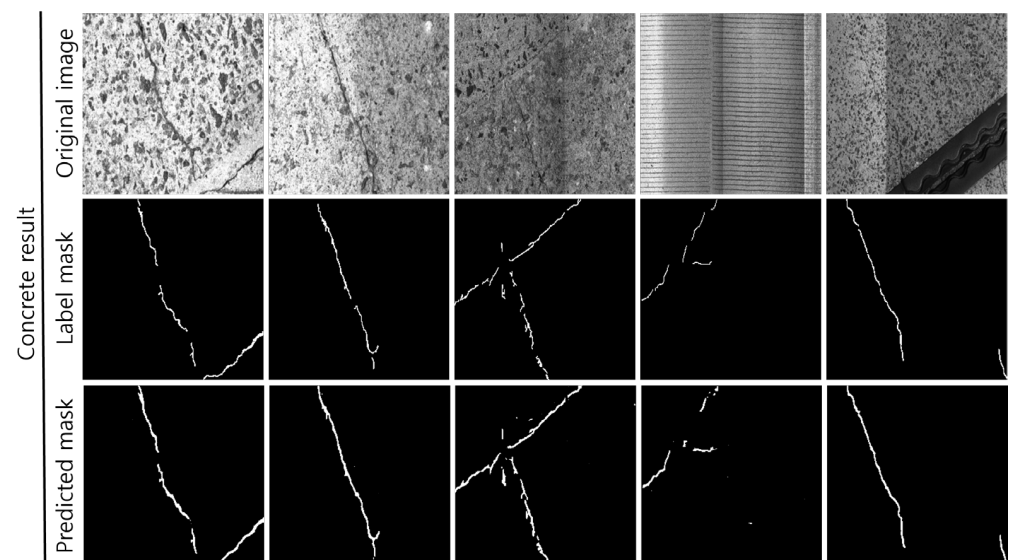


Figure 5. Prediction results of the proposed method on rough concrete test images.

Figure 6 shows the prediction results in a concrete environment with a striped pattern. Concrete with a striped pattern has bright contrast, and a regular striped pattern appears distinctly. Such a repetitive linear pattern is similar to the linear shape of cracks and can act as a factor that easily confuses the model. Nevertheless, the proposed method effectively distinguished the striped pattern from actual cracks, detecting long, continuous diagonal cracks and various branching cracks in a manner similar to the ground-truth mask. This suggests that, for these test images, the model was largely robust to the repetitive pattern noise of the surface and captured the irregular and continuous morphological features of cracks.

Figure 7 shows the prediction results in the tunnel environment. Although the interior of a tunnel is dark, with low contrast between cracks and background, the proposed method stably detected cracks of various shapes and numbers in a manner similar to the ground-truth mask. In particular, in some images, the curved (circular) shape of the tunnel structure made the crack path appear visually broken, or the curves of the structure appeared crack-like. However, even amid such structural distortion and potential confusion, the proposed

model did not lose the continuity of the cracks but naturally connected the broken parts in its predictions, and it effectively distinguished the curves of the structure from actual cracks. This demonstrates that, even amid the visual discontinuity and confusion caused by the dark illumination environment and curved structure, the model captured the overall shape and path of cracks in these test images.

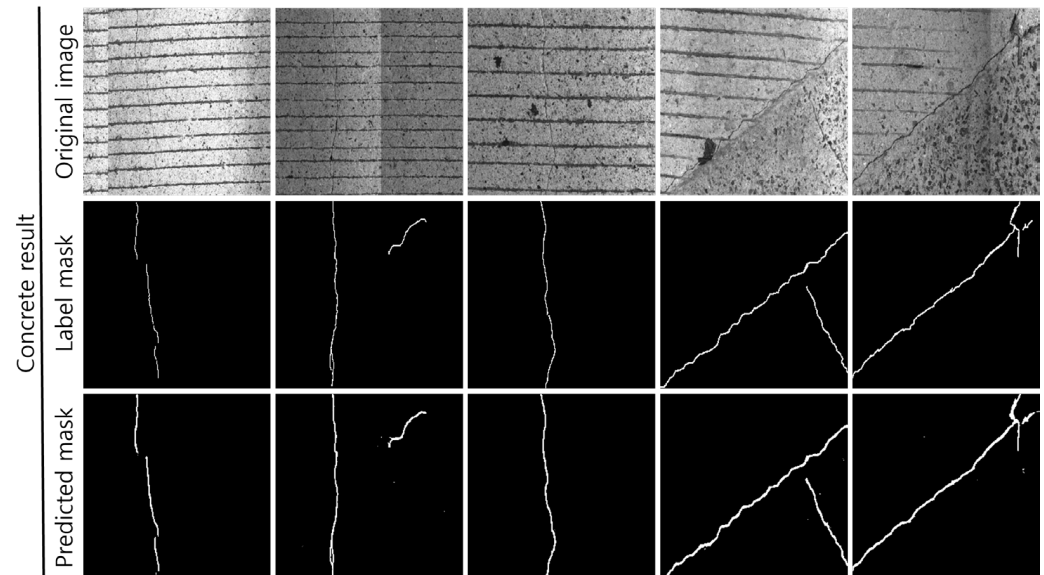


Figure 6. Prediction results of the proposed method on striped concrete test images.

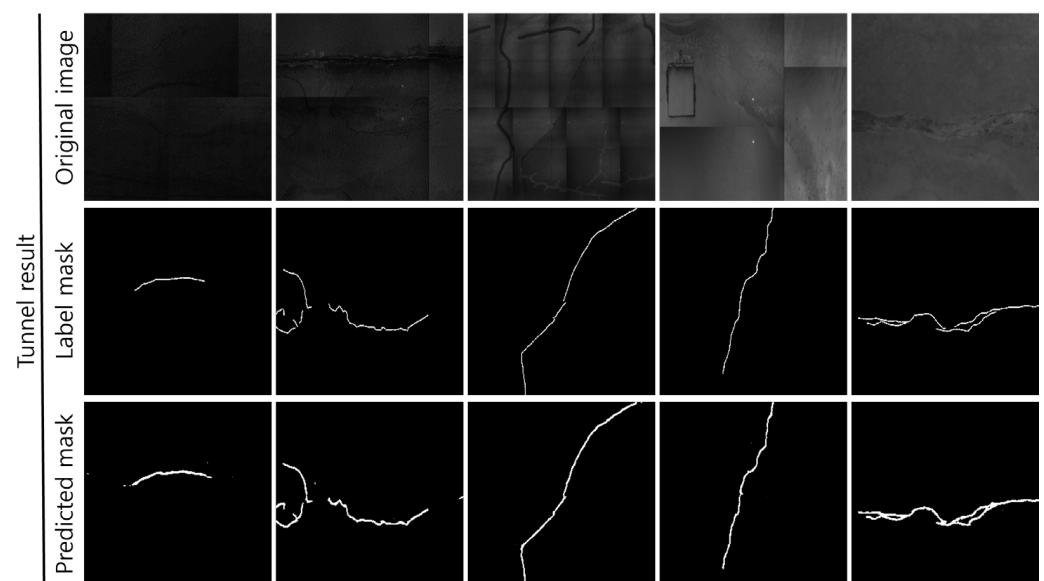


Figure 7. Prediction results of the proposed method on tunnel test images.

4. Discussion

To analyze the performance of the proposed method in detail, the box plot distributions in Figure 3 and the quantitative results in Table 3 were compared.

4.1. Performance Analysis of Individual Methods

As shown in Figure 3a, the proposed method showed the highest precision and the narrowest distribution. These results indicate that the method with postprocessing consistently maintained low false positives across most test samples. These results further

suggest that area-based filtering can stably remove noise regardless of variability between samples. As shown in Figure 3b, the recall distributions of Dense U-Net++ and Dense U-Net++ with postprocessing appeared similar. This finding suggests that removing objects with an area of 10 pixels or less during postprocessing suppressed background false positives while preserving actual crack regions, so that the macro-averaged recall was maintained. In other words, postprocessing achieved an efficient trade-off that increases precision without sacrificing recall. As shown in Figure 3c, the highest F1-score and the narrowest distribution confirm that precision and recall improved in a balanced manner. This indicates that, on the evaluated test images, the proposed method is not biased toward any particular metric and achieved the highest overall detection performance among the compared methods. As shown in Figure 3d, the proposed method also achieved the highest IoU, indicating that the predicted crack masks overlap more closely with the ground truth in both extent and position. Since IoU penalizes false positives and false negatives simultaneously, this result confirms that the improvement is not limited to a single aspect of the prediction but reflects a more accurate reconstruction of the overall crack regions. In contrast, Mask R-CNN showed the lowest performance and the widest distribution across all four metrics. This observation suggests that the instance segmentation-based Mask R-CNN is relatively weak at precisely capturing, at the pixel level, the boundaries of cracks that have thin and irregular shapes. This limitation is considered to arise from the fact that Mask R-CNN is optimized for instance-level bounding box-based detection and is structurally unsuited to handling thin, continuous linear structures.

4.2. Comparison Results Among Different Methods

The fact that Dense U-Net++ showed higher performance than Mask R-CNN demonstrates that the U-Net++-based structure, which effectively fuses multi-scale features through nested skip pathways and dense connections, is more suitable for precisely segmenting thin and irregularly shaped cracks at the pixel level than Mask R-CNN, which detects objects on an instance basis. In addition, the fact that the model with postprocessing applied showed the highest performance means that area-based filtering effectively removed noise-induced FP in the background class while preserving the correctly detected crack regions, thereby improving the macro-averaged performance. As a result, the proposed method also recorded the highest F1-score, which considers precision and recall simultaneously, confirming that it possesses a balanced ability to reduce misclassification and to detect cracks without omission.

4.3. Limitations and Future Work

The postprocessing technique used in this study adopts a fixed threshold approach that uniformly removes objects with an area of 10 pixels or less. Such a fixed threshold can effectively remove most point-shaped noise, but there is a risk that fine cracks at an early stage or some regions of actual cracks that start very thin may be removed together if they have an area similar to noise. This can act as a limitation that degrades detection accuracy at the early stage before a crack has sufficiently progressed. To address this, future research will introduce an adaptive postprocessing technique that, instead of an area-based fixed threshold, also considers the morphological features of cracks (e.g., length, curvature, continuity), or a learning-based noise classification module, to more precisely distinguish noise from actual early-stage cracks. In addition, this study relies on a single self-constructed dataset. As a result, there is a limitation that the generalization performance of the model on external data has not been verified. In future research, cross-dataset validation using publicly available crack datasets or external datasets collected by other institutions will be performed to secure the robustness of the model against various data distributions.

In addition, the scope of the experimental evaluation was limited. The primary objective of this study was to construct a crack detection system that operates consistently on a unified dataset spanning asphalt, concrete, and tunnel environments, rather than to identify the optimal segmentation architecture. Accordingly, performance comparison with various models and exploration of the optimal model configuration were beyond the scope of this study. The model was also evaluated only on crack-containing patches under nominal imaging conditions, so its false-alarm behavior on crack-free surfaces and its resilience to perturbations such as motion blur and exposure changes remain to be verified. These aspects will be addressed in future work.

5. Conclusions

In this study, we proposed a Dense U-Net++-based model trained on an integrated crack dataset containing asphalt, concrete, and tunnel images. By combining dense connections with the nested skip connections of U-Net++, the model reduced the semantic gap between the encoder and decoder and improved multi-scale feature fusion. Focal loss was used to address class imbalance, and area-based postprocessing was applied to remove detection noise.

Experimental results showed that the proposed method achieved a precision of 95.47%, a recall of 92.35%, an F1-score of 93.91%, and an IoU of 88.58%, outperforming the evaluated comparison models across all evaluation metrics. Compared with Mask R-CNN, the F1-score and IoU improved by 4.56% and 7.75%, respectively. Postprocessing substantially improved precision with little effect on recall, indicating effective suppression of false positives. Visual evaluation on the test images also showed reliable crack detection under the evaluated conditions, including rough asphalt textures, striped concrete surfaces, and low-illumination tunnel images. A limitation of this study is that the postprocessing step relies on a fixed area threshold, which may remove small early-stage cracks. In addition, the scope of the experimental evaluation was limited. The primary objective of this work was to construct a crack detection system that operates consistently on a unified dataset spanning asphalt, concrete, and tunnel environments, which have conventionally been studied separately, rather than to identify the optimal segmentation architecture for this task. Accordingly, performance comparison with various models, exploration of the optimal model configuration, and validation on publicly available crack datasets were beyond the scope of the present study. The model was also evaluated only on crack-containing patches under nominal imaging conditions, so its false-alarm behavior on crack-free surfaces and its resilience to perturbations such as motion blur, exposure changes, and altitude variations remain to be verified. Reliable detection of early-stage cracks and generalization across diverse structural environments therefore remain key challenges.

Future work will address these issues through adaptive postprocessing that incorporates crack morphology, comparative evaluation against recent architectures with systematic ablation studies, cross-dataset validation using publicly available crack datasets, assessment on crack-free surfaces, and controlled robustness analyses under synthetic perturbations.

Overall, the proposed Dense U-Net++ model achieved promising crack segmentation performance across different structural surfaces and imaging conditions, demonstrating its potential for automated structural inspection and maintenance applications.

Author Contributions: Conceptualization, J.-C.H. and J.-H.L.; methodology, J.-C.H.; software, D.-B.K.; validation, J.-C.H. and J.-H.L.; formal analysis, D.-B.K. and S.A.; investigation, D.-B.K. and J.-C.H.; resources, J.-C.H. and J.-H.L.; data curation, J.-C.H.; writing—original draft preparation, D.-B.K.; writing—review and editing, S.A. and J.-H.L.; visualization, D.-B.K.; supervision, J.-H.L.; project administration, J.-H.L.; funding acquisition, J.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (Grant No. RS-2025-12345678); the Collabo R&D Program between Industry, University, and Research Institute funded by the Ministry of SMEs and Startups (MSS), Republic of Korea (Grant No. RS-2026-25533076); the Regional Innovation System & Education (RISE) Program through the Daegu RISE Center, funded by the Ministry of Education (MOE) and Daegu Metropolitan City, Republic of Korea (Grant No. 2025-RISE-03-002); and the Commercialization Promotion Agency for R&D Outcomes (COMP), funded by the Ministry of Science and ICT (MSIT), Republic of Korea (Grant No. RS-2025-02413013, IP Enhancement and Commercialization to Promote the Market Adoption of Vision Intelligence-Based Multi-Biosignal Measurement Technology).

Data Availability Statement: The datasets presented in this article are not readily available because they are part of an ongoing study. Requests to access the datasets should be directed to the corresponding author.

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

References

1. Yuan, Q.; Shi, Y.; Li, M. A Review of Computer Vision-Based Crack Detection Methods in Civil Infrastructure: Progress and Challenges. *Remote Sens.* **2024**, *16*, 2910. [\[CrossRef\]](#)
2. Khan, S.; Jan, A.; Seo, K. Accurate Structural Crack Detection Using NestedUNet from Drone and Handheld Camera Images. *KSCE J. Civ. Eng.* **2025**, *29*, 100204. [\[CrossRef\]](#)
3. Golding, V.P.; Gharineiat, Z.; Munawar, H.S.; Ullah, F. Crack Detection in Concrete Structures Using Deep Learning. *Sustainability* **2022**, *14*, 8117. [\[CrossRef\]](#)
4. Shahin, M.; Chen, F.F.; Maghanaki, M.; Hosseinzadeh, A.; Zand, N.; Koodiani, H.K. Improving the Concrete Crack Detection Process via a Hybrid Visual Transformer Algorithm. *Sensors* **2024**, *24*, 3247. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Kruachottikul, P.; Cooharajanane, N.; Phanomchoeng, G.; Chavarnakul, T.; Kovitangoon, K.; Trakulwaranont, D. Deep Learning-Based Visual Defect-Inspection System for Reinforced Concrete Bridge Substructure: A Case of Thailand's Department of Highways. *J. Civ. Struct. Health Monit.* **2021**, *11*, 949–965. [\[CrossRef\]](#)
6. Mohan, A.; Poobal, S. Crack Detection Using Image Processing: A Critical Review and Analysis. *Alex. Eng. J.* **2018**, *57*, 787–798. [\[CrossRef\]](#)
7. Hsieh, Y.-A.; Tsai, Y.J. Machine Learning for Crack Detection: Review and Model Performance Comparison. *J. Comput. Civ. Eng.* **2020**, *34*, 04020038. [\[CrossRef\]](#)
8. Khan, S.I.; Qadir, Z.; Munawar, H.S.; Nayak, S.R.; Budati, A.K.; Verma, K.D.; Prakash, D. UAVs Path Planning Architecture for Effective Medical Emergency Response in Future Networks. *Phys. Commun.* **2021**, *47*, 101337. [\[CrossRef\]](#)
9. Tahir, A.; Munawar, H.S.; Akram, J.; Adil, M.; Ali, S.; Kouzani, A.Z.; Mahmud, M.A.P. Automatic Target Detection from Satellite Imagery Using Machine Learning. *Sensors* **2022**, *22*, 1147. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Bui, H.M.; Lech, M.; Cheng, E.; Neville, K.; Burnett, I.S. Using Grayscale Images for Object Recognition with Convolutional-Recursive Neural Network. In Proceedings of the 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE), Ha-Long, Vietnam, 27–29 July 2016; pp. 321–325.
11. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
12. Liu, Z.; Cao, Y.; Wang, Y.; Wang, W. Computer Vision-Based Concrete Crack Detection Using U-Net Fully Convolutional Networks. *Autom. Constr.* **2019**, *104*, 129–139. [\[CrossRef\]](#)
13. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In *Proceedings of the Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Stoyanov, D., Taylor, Z., Carneiro, G., Syeda-Mahmood, T., Martel, A., Maier-Hein, L., Tavares, J.M.R.S., Bradley, A., Papa, J.P., Belagiannis, V., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–11.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
15. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In *Proceedings of the Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–19.

16. Zhang, D.; Zou, Q.; Lin, H.; Xu, X.; He, L.; Gui, R.; Li, Q. Automatic Pavement Defect Detection Using 3D Laser Profiling Technology. *Autom. Constr.* **2018**, *96*, 350–365. [[CrossRef](#)]
17. Chen, W.; Peng, Z.; Chen, X.; Zhao, L.; Xu, T.; Li, Q.; Huang, X.; Kannangara, K.K.P.M. Concrete Crack Detection and Classification Methods Based on Machine Vision and Deep Learning. *Sensors* **2026**, *26*, 2381. [[CrossRef](#)] [[PubMed](#)]
18. Dorafshan, S.; Thomas, R.J.; Maguire, M. Benchmarking Image Processing Algorithms for Unmanned Aerial System-Assisted Crack Detection in Concrete Structures. *Infrastructures* **2019**, *4*, 19. [[CrossRef](#)]
19. Zhang, H.; Liu, J.; Hu, G. FCN Attention Enhancing Asphalt Pavement Crack Detection through Attention Mechanisms and Fully Convolutional Networks. *Sci. Rep.* **2025**, *15*, 25257. [[CrossRef](#)] [[PubMed](#)]
20. Ren, Y.; Huang, J.; Hong, Z.; Lu, W.; Yin, J.; Zou, L.; Shen, X. Image-Based Concrete Crack Detection in Tunnels Using Deep Fully Convolutional Networks. *Constr. Build. Mater.* **2020**, *234*, 117367. [[CrossRef](#)]
21. Huang, W.-C.; Luo, Y.-S.; Liu, W.-C.; Liu, H.-M. Deep Learning-Based Crack Detection on Cultural Heritage Surfaces. *Appl. Sci.* **2025**, *15*, 7898. [[CrossRef](#)]
22. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
23. Kim, D.-B.; Lee, J.-H. Digital Staining Algorithm for Multi-Domain Transformation of Unstained Images. *IEEE Access* **2025**, *13*, 96758–96766. [[CrossRef](#)]
24. Metlek, S. CellSegUNet: An Improved Deep Segmentation Model for the Cell Segmentation Based on UNet++ and Residual UNet Models. *Neural Comput. Appl.* **2024**, *36*, 5799–5825. [[CrossRef](#)]
25. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017*; IEEE: New York, NY, USA, 2017; pp. 2961–2969.
26. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.