frontiers | Frontiers in Built Environment

TYPE Original Research
PUBLISHED 15 November 2024
DOI 10.3389/fbuil.2024.1485774

OPEN ACCESS

EDITED BY
Baki Ozturk,
Hacettepe University, Türkiye

REVIEWED BY
Hailu Yang,
University of Science and Technology Beijing, China
Zidong Xu,
Southeast University, China
Xiaohua Li,
Chongqing University, China

*CORRESPONDENCE
Mehdi Ravanshadnia,
ravanshadnia@srbiau.ac.ir

RECEIVED 24 August 2024
ACCEPTED 30 October 2024
PUBLISHED 15 November 2024

CITATION
Sarhadi A, Ravanshadnia M, Monirabbasi A and Ghanbari M (2024) Using an improved U-Net++ with a T-Max-Avg-Pooling layer as a rapid approach for concrete crack detection. *Front. Built Environ.* 10:1485774. doi: 10.3389/fbuil.2024.1485774

Using an improved U-Net++ with a T-Max-Avg-Pooling layer as a rapid approach for concrete crack detection

Ali Sarhadi[1], Mehdi Ravanshadnia[1]*, Armin Monirabbasi[2] and Milad Ghanbari[3]

[1]Department of Civil Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran, [2]Department of Civil Engineering, Payame Noor University, Tehran, Iran, [3]Department of Civil Engineering, East Tehran Branch, Islamic Azad University, Tehran, Iran

The monitoring of concrete structures has advanced remarkably with the aid of deep learning technologies. Since concrete is multi-purpose and low-cost, it is extensively used for construction purposes. Concrete is very enduring. Nevertheless, it tends to crack which endangers the integrity of the structure and results in complications. The current study offers a new image segmentation approach for detecting cracks in concrete by making use of an optimized U-Net++ architecture. The proposed model gives the features of the T-Max-Avg Pooling layer which effectively combines the advantages of traditional max and average pooling using a learnable parameter to balance feature extraction dynamically. This innovation both improves the output accuracy and processing speed and captures the fine details. In addition, it mitigates noise and transcends the limitations of conventional pooling methods. Moreover, using learnable pruning and shortening skip connections in U-Net++ reduce redundant computations, making the model faster without compromising accuracy. In comparison with other models like Mask R-CNN and VGG-U-Net, the proposed model had considerably faster inference times (21.01 ms per image) and fewer computational requirements (40G FLOPs), making it very suitable for real-time monitoring applications. The DeepCrack and Concrete Pavement Crack datasets were employed to assess the model thoroughly which yielded an MIoU score of 82.1%, an F1 score of 90.12%, a Dice loss score of 93.7%, and an overall accuracy of 97.65%. According to the results, the enhanced U-Net++ with T-Max-Avg Pooling provided a balanced trade-off between segmentation accuracy and computational efficiency. This indicates its considerable potential for automated real-time crack detection in concrete structures by employing resource-constrained environments including drones and mobile platforms.

KEYWORDS

concrete crack detection, deep learning, image segmentation, skip connection pruning, U-Net++ architecture

1 Introduction

Concrete crack detection is vital for the maintenance and inspection of concrete structures because the existing cracks are the chief indicators of the durability and potential damage of the structure. Regular crack measurement systems are part of the construction inspection programs of many countries and are usually conducted visually by technicians (Sjolander et al., 2023; Yan et al., 2019). The conventional devices for crack detection, such as the Schmidt Hammer and ultrasonic wave generators, are effective. Nonetheless, they are time-consuming, labor-intensive, and prone to human error (Yu et al., 2022; Nie et al., 2018). Furthermore, these devices lack the precision needed for a reliable and consistent crack detection in difficult conditions. By providing superior efficiency, robustness, and accuracy, recent improvements in deep learning, especially convolutional neural networks (CNNs), have led to a revolution in crack detection (Alam et al., 2020). Deep learning-based methods transcend the limitations of conventional approaches by automatically learning complex features from the image data. In this way, the need for expert intervention and manual feature extraction is eliminated. In spite of their theoretical advantages, these methods are challenging to use in practical applications such as real-time monitoring in resource-constrained environments (Sanjerehei and Rundel, 2020). Deep learning offers significant advantages over traditional image processing and machine learning techniques including independence from expert-guided thresholds, superior precision, and robustness with respect to various images. CNNs are the driving force behind advancements in computer vision (LeCun et al., 2015). Image segmentation, a crucial aspect of visual systems, involves dividing an image into its constituent components. This process is vital in applications such as medical image analysis, self-driving vehicles, and augmented reality. Recent deep learning-based segmentation models have significantly outperformed traditional methods, leading to a paradigm shift in image segmentation. U-Net is a convolutional neural network architecture designed for biomedical image segmentation. It was introduced by Ronneberger et al., in 2015 and has since gained popularity due to its simple yet effective design (Ronneberger et al., 2015). The U-Net architecture consists of a contracting path to capture context and a symmetric expanding path for precise localization, making it particularly effective for segmenting images with limited annotated data. U-Net++ is an extension of the original U-Net architecture proposed to improve segmentation by redesigning the skip connections. The main innovation in U-Net++ is the use of nested and dense skip pathways which aim to reduce the semantic gap between the encoder and decoder subnetworks. This results in a more accurate segmentation especially in medical imaging tasks (Qian et al., 2024). EfficientDet is a family of object detection models that leverage the EfficientNet backbone for feature extraction. Presented in 2020, this model is well-known for its balance between efficiency and accuracy. EfficientDet introduces a weighted bi-directional feature pyramid network (BiFPN) as well as a compound scaling method that uniformly scales the width, depth, and resolution of the backbone, box/class prediction network, and feature network, providing a state-of-the-art performance with fewer computations and parameters (Nawaz et al., 2022). In addition to U-Net++ enhancements, EfficientDet, a state-of-the-art object detection model, has shown a remarkable performance in various detection tasks due to its efficient architecture and compound scaling (Tan et al., 2020). EfficientDet balances accuracy and computational efficiency by scaling the network width, depth, and resolution uniformly. While EfficientDet has been primarily designed for object detection, its principles can be applied to image segmentation tasks, providing insights into developing more efficient and accurate models for crack detection. Given the advancements in image segmentation, its application for detecting cracked zones has increased (Lyu et al., 2023). Recent advancements have demonstrated that physics-informed neural networks (PINNs) have become more and more popular in various engineering fields. Physical laws, represented by partial differential equations, are directly integrated by PINNs into the learning process of the neural network. This enhances the reliability and interpretability of the model, particularly for engineering applications that need a solid basis in physical principles. For example, the application of hierarchical deep learning and physics-informed finite element analysis in engineering has shown their significant potential in improving the accuracy of prediction and reducing the computational requirements, as emphasized by recent studies (Antony et al., 2023; Rodriguez-Torrado et al., 2022; Asadzadeh et al., 2023). Moreover, research on structural health monitoring and failure analysis has also incorporated PINNs, showing their considerable ability in modeling complex systems with embedded physical constraints, as observed in recent works. These contributions are especially relevant in the field of concrete crack detection in which the integration of physics into data-driven models can improve their generalization and robustness, resulting in more dependable real-time monitoring systems. By highlighting these developments, the importance of incorporating physics-informed strategies besides deep learning techniques becomes evident. In fact, they provide new opportunities for enhancing the reliability and accuracy of anomaly detection in civil infrastructure (Wang et al., 2023). The aim of this study is to bridge the gap between the theoretical capabilities of deep learning and its practical deployment for infrastructure monitoring. By reducing the computational costs and having a high accuracy, the proposed enhanced U-Net++ architecture offers a balanced trade-off that makes it viable for real-world large-scale crack detection applications, ultimately improving the safety of the infrastructure and the efficiency of its maintenance.

2 Materials and methods

2.1 Data collection

The dataset utilized for this study comprised high-resolution images of concrete surfaces including both cracked and non-cracked areas. These images were sourced from publicly available datasets and augmented with additional images collected from real-world inspections to ensure a comprehensive representation of various crack types and conditions. The images were meticulously annotated to label the crack regions accurately. The images had various sizes. Therefore, they had to be preprocessed before running the algorithm.

2.1.1 The dataset details

The DeepCrack dataset consisted of 20,000 images, while the Concrete-Pavement Crack dataset included 10,000 images. The images had a high resolution and ensured a wide range of cracks. Some of the data used in this study are shown in Figure 2.

2.1.2 Dataset features

The crack regions in all images were manually annotated by civil engineering experts to ensure high-quality ground-truth data. The images included variations in surface types, lighting conditions (daylight and shadow), and environmental factors (wet and dry conditions).

2.1.3 Dataset preparation and usage

The images were resized to a standard resolution of 256 × 256 pixels and normalized to values between 0 and 1 to align with the input requirements of the network. The datasets was split into 70% for training, 15% for validation, and 15% for testing to ensure a robust evaluation.

2.1.4 Analysis and evaluation

The datasets had a significant imbalance. There were more intact regions than cracks in the majority of the images. To address this problem and to increase the accuracy of the model, data augmentation techniques were used.

2.1.1.1 The data augmentation techniques and their effects

In this study, data augmentation techniques were employed to enhance the robustness of the model by increasing the diversity of the training dataset. The data augmentation techniques included random rotations (between −30 to +30°), translations (up to 10% of the image size along the X and Y-axes), scaling (between 0.8 and 1.2), horizontal and vertical flips (with a probability of up to 50%), brightness adjustment (between 0.5 and 1.5), contrast adjustment (between 0.5 and 1.5), and Gaussian noise addition. These techniques were selected to mimic various real-world conditions under which concrete crack images could be captured, thus helping the model generalize better to different scenarios. These augmentation techniques were selected based on their ability to cover a vast range of conditions that are likely to be met in real-life scenarios such as changes in image quality, perspective, or lighting. Applying the data augmentation techniques improved the performance of the model significantly. In particular, by providing diverse input data, it mitigated overfitting. This allowed the model to learn more generalized features. The changes randomly applied to the images as well as their descriptions are listed in Table 1.

2.2 Model architecture

The model used in this study was an improved version of the U-Net++ architecture augmented with a novel T-Max-Avg-Pooling layer for enhanced feature extraction and output refinement.

2.2.1 U-Net++ architecture

U-Net++ is an advanced variant of the traditional U-Net architecture designed specifically for image segmentation tasks. It aims to improve performance by introducing dense skip connections and nested dense convolutional blocks (Zhou et al., 2018). Additionally, the use of the T-Max-Avg Pooling layer enhances its ability to capture diverse features. Figure 1 shows the U-Net++ architecture.

2.2.2 The overview of the U-Net++ architecture and its comparison with the traditional U-Net architecture

The U-Net++ architecture is a state-of-the-art version of the traditional U-Net. It was designed to overcome the limitations in segmentation accuracy and feature propagation. U-Net++ presents two principal innovations: nested convolutional blocks and dense skip connections. These dense skip connections bridge the gap between the encoder and decoder characteristics more efficiently than the original U-Net. This increases feature propagation and decreases the semantic disparity between the encoder and decoder pathways. The skip connections in the original U-Net connect each encoder layer to its corresponding decoder layer. This contributes to the preservation of spatial information. Nonetheless, these connections can result in a considerable semantic gap between high-level and low-level feature maps. U-Net++ mitigates this gap by incorporating convolutional operations and intermediate dense skip pathways between the encoder and decoder stages. This both refines the feature maps and improves the quality of the segmentation output (Zhang et al., 2023). The new T-Max-Avg Pooling layer further improves the performance of U-Net++ by combining average pooling, max pooling, and a trainable pooling operation. This pooling strategy permits the model to capture the prominent features adaptively and reduce the noise. This offers a more thorough feature extraction in comparison with the standard average or max pooling layers utilized in the traditional U-Net.

2.2.2.1 Traditional Pooling Methods

- Max Pooling: In this method, the maximum value from each patch of the feature map is selected. Although it selects the most notable feature, it may discard other important information. Besides, it is sensitive to noise (Liu Z. et al., 2019).
- Average Pooling: In this method, the average value of the features in a patch is calculated. Though it reduces noise, it may obscure the valuable features. In addition, unlike Max Pooling, it lacks the ability to select the sharp details or edges (Su and Wang, 2020).
- Mixed Pooling: This method is a combination of max pooling and average pooling (in the conventional sense of the term) (Li et al., 2023).

2.2.3 T-Max-Avg Pooling

T-Max-Avg Pooling is a cutting-edge method which combines the advantages of both Max and Average Pooling. Specifically:

- It chooses the K highest pixel values from the feature map and employs parameter T to control the outputs of the average and maximum values of these pixels.
- This flexibility lets the model retain both prominent features (like max pooling) and reduce noise (like average pooling). This makes the model more adaptive to different types of data.

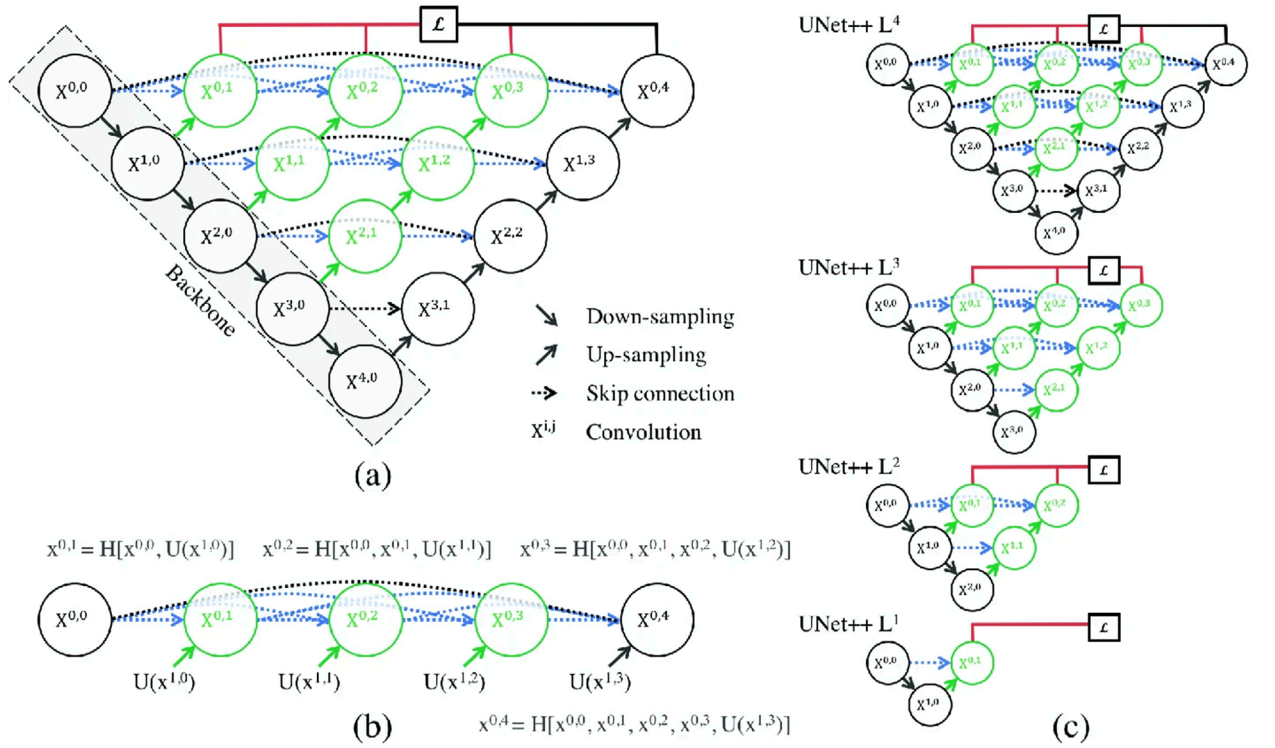

FIGURE 1
(A) U-Net++ consists of an encoder and a decoder that are connected through a series of nested dense convolutional blocks. The main idea behind U-Net++ is to bridge the semantic gap between the feature maps of the encoder and decoder prior to fusion. For example, the semantic gap between X0, 0 and X1, 3 is bridged using a dense convolutional block with three convolutional layers. **(B)** The detailed analysis of the first skip pathway of U-Net++. **(C)** U-Net++ can be pruned at inference time if trained with deep supervision (Zhou et al., 2018).

FIGURE 2
Examples of the used images.

- By learning adaptive pooling operations, T-Max-Avg improves feature extraction and assures that the model extracts both fine details and extensive contextual information.

2.2.3.1 Mathematical Formula

Let X represent the input feature map of a pooling window and let K indicate the number of top values selected from this window. The T-Max-Avg Pooling operation is expressed in the following way:

$$F(T-Max-Avg(X)) = T \cdot \mathrm{Max}(X_k) + (1-T) \cdot \mathrm{Avg}(X_k) \quad (1)$$

The detailed derivations for Equations 1–11 can be found in this study. Where:

TABLE 1 The random changes to the images and their descriptions.

Description	Change type
Rotating the image by a random angle within a specified range (−30 to +30°)	Random Rotation
Shifting the image randomly along the X and Y-axes within a specified range (up to 10% of the image size)	Random Translation
Scaling the image by a random factor within a specified range (0.8–1.2)	Random Scaling
Flipping the image horizontally with a certain probability (up to 50%)	Random Horizontal Flip
Flipping the image vertically with a certain probability (up to 50%)	Random Vertical Flip
Adjusting the brightness of the image by a random factor within a specified range (0.5–1.5)	Random Brightness Adjustment
Adjusting the contrast of the image by a random factor within a specified range (0.5–1.5)	Random Contrast Adjustment
Adding Gaussian noise to the image with a specified mean and standard deviation	Gaussian Noise Addition

TABLE 2 Comparing the performance of the models.

Pooling method	Dice (%)	mIoU (%)	F1 (%)	Precision (%)	Recall (%)	Inference time (ms)
Max Pooling	93.1	76.5	85.4	88.7	81.2	30.05
Average Pooling	92.8	75.9	84.9	87.9	80.8	28.5
Mixed Pooling	93.5	77	86.2	89	82.4	29.5
T-Max-Avg Pooling	93.7	82.1	90.12	87.6	94.5	21.01

TABLE 3 The runtime, memory usage, and FLOPs for the proposed model and the existing benchmarks.

Models	Inference time (ms/image)	Memory (GB)	FLOPs (G)
Deep Crack (Liu et al., 2019b)	98.23	6	45
CNN (Fan et al., 2018)	90.57	4	25
VGG-U-Net (Shi et al., 2021)	86.71	11	120
SegNet (Nguyen et al., 2022)	95.45	9	70
U-Net (Liu et al., 2019c)	30.05	9	60
U-Net++ (Zhou et al., 2020)	28.65	8	50
EfficientDet (Sohaib et al., 2024)	32.15	5	45
Mask R-CNN (He et al., 2018)	85.36	13	160
The Proposed Method	21.01	8	40

- X indicates the set of pixel values or feature values in the pooling window.
- K represents the number of highest values taken from the window.
- X_k stands for the top K highest values from the pooling window.
- Max(X_k) indicates the maximum of these K values.
- Avg(X_k) is the average of the K highest values.
- T is the tunable parameter that determines the contribution of the Max and Average Pooling.
  - When T = 1, the pooling operation behaves like Max Pooling and entirely focuses on the largest feature in the window.
  - When T = 0, the operation behaves like Average Pooling and smooths the features by averaging all of them.

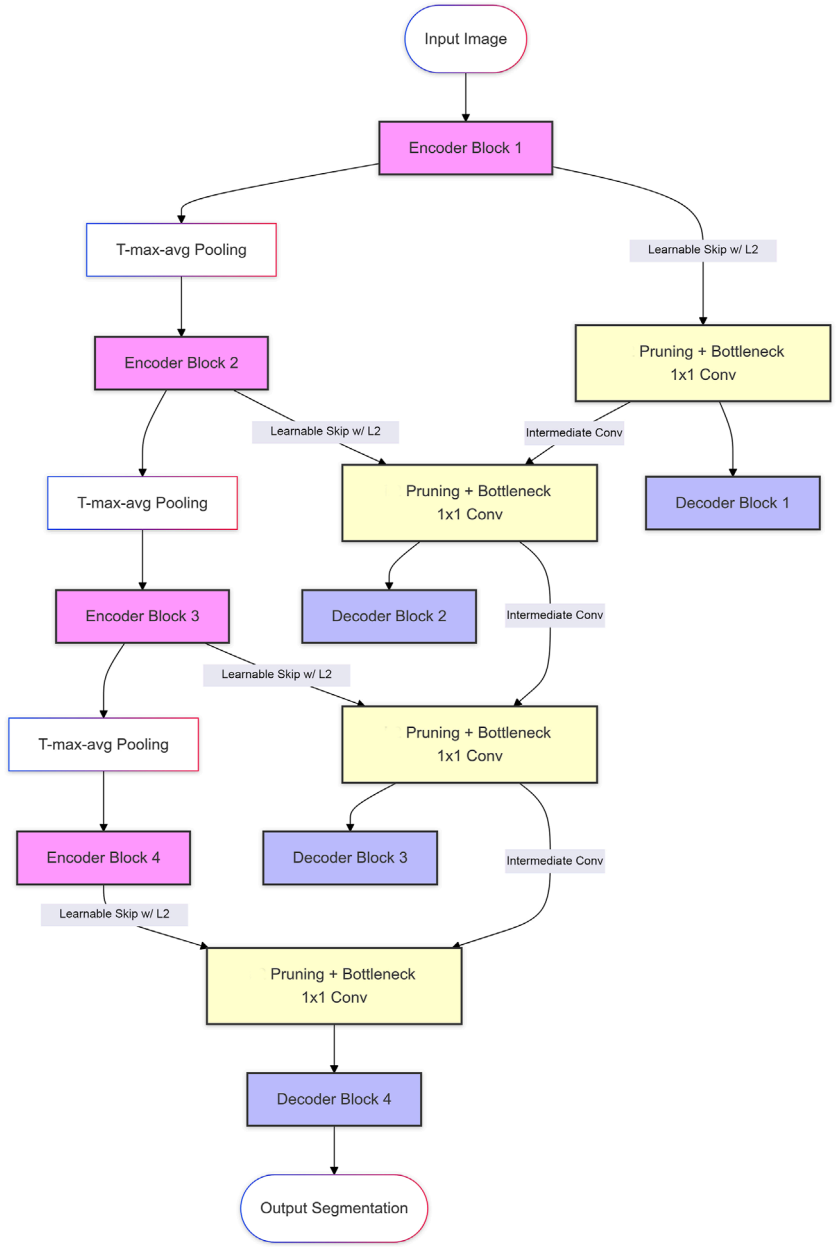

FIGURE 3
The architecture of the proposed method.

- When $0 < T < 1$, the operation mixes both pooling strategies which allows the model to adapt dynamically according to the features in the input window (Zhao and Zhang, 2024).

2.2.4 The contributions of T-Max-Avg Pooling and dense skip connections

The dense skip connections improve gradient flow as well as feature reuse across layers. Liu et al. (2019a) and Oluwaseun (2023) provide further details regarding similar pooling techniques and their applications in image segmentation. This considerably enhances the performance of the model with respect to both convergence speed and accuracy. Furthermore, learnable skip connection pruning, which gives weights to each skip connection, is used. In this way, the model is enabled to remove the redundant pathways during training without a considerable loss in performance. This both accelerates the model and mitigates computational complexity, making U-Net++ appropriate for resource-constrained environments. A new feature of the proposed architecture, the T-Max-Avg Pooling layer improves the ability of the network to capture both broader contextual information and fine details. Traditional max pooling often concentrates on the most

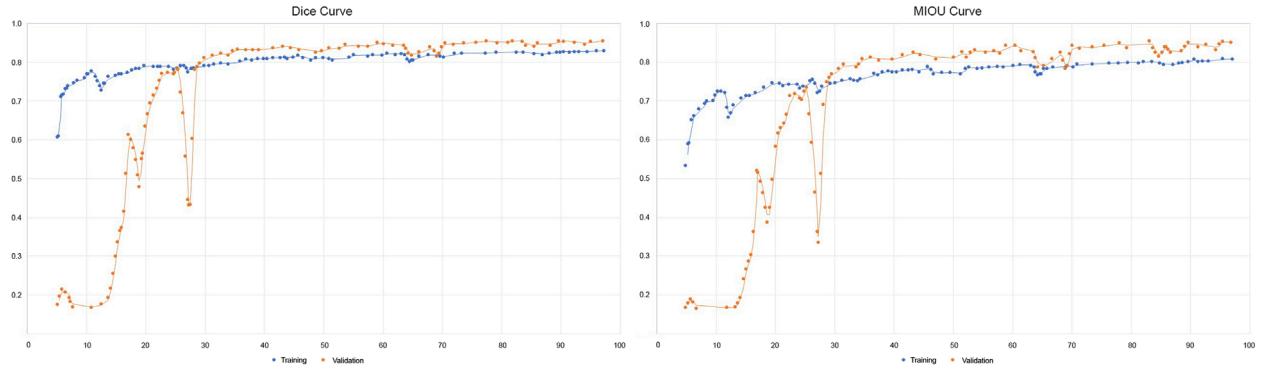

FIGURE 4
The Dice coefficient and MIoU of the network.

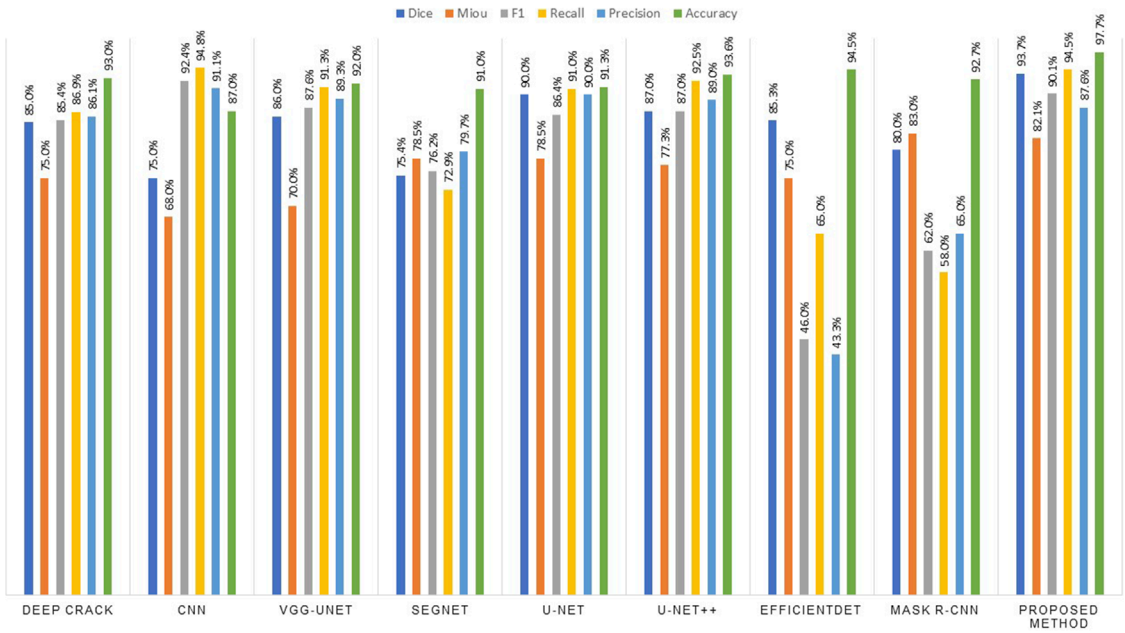

FIGURE 5
Comparing the validations of the methods.

prominent features, while average pooling mitigates the noise but may blur the vital details. T-Max-Avg Pooling dynamically balances the maximum and average values by combining these approaches with a trainable parameter (T). This enables the model to capture the important features adaptively which improves the accuracy of segmentation.

2.2.5 The effect of pruning on the performance and training time of the model

Pruning skip connections in U-Net++ presents a practical way to accelerate the model by reducing the redundant calculations and maintaining the quality of its performance. The pruning process involves the gradual removal of the less important skip connections detected by giving weights to each connection and adjusting them by employing the L2 norm during training. The connections with a minimal contribution to the final output are pruned. This permits the model to concentrate on critical information pathways. The effect of pruning on performance is considerable in two principal aspects, computational efficiency and speed (Chatzikonstantinou et al., 2021). When the less essential skip connections are pruned, the complexity of the model is reduced which leads to faster training and inference times without a marked decrease in accuracy. Particularly, using learnable skip connections assures that only redundant pathways are eliminated and the ability of the model to combine low- and high-level feature maps effectively is preserved. This approach both accelerates the training process and improves the adaptability of the model by preventing overfitting to unnecessary connections. Moreover, bottleneck compression is utilized to further reduce the computational load through 1 × 1 convolutional layers to decrease the dimensionality of the

feature maps in the skip connections. This decreases the number of parameters and FLOPs (floating-point operations) which leads to faster inference times and lower memory consumption. In addition, the critical features are preserved for an accurate segmentation. In general, the pruning strategy provides a balanced compromise between computational efficiency and accuracy, making U-Net++ more appropriate for real-time applications, especially in resource-constrained environments. The results of the experiments showed that proper pruning, combined with bottleneck compression, leads to a noticeable reduction in training time and a notable increase in processing speed without compromising the performance of the model. Doing experiments with trade-offs between accuracy and speed assures that U-Net++ becomes more effective and efficient as mentioned in Table 3.

2.2.6 The benefits of the T-Max-Avg Pooling layer over traditional pooling methods

The T-Max-Avg Pooling layer was designed to transcend several inherent limitations in traditional pooling methods such as average pooling and max pooling in order to enhance both the feature extraction and performance of the model. This part mentions the theoretical advantages that make T-Max-Avg Pooling an exceptional choice for deep learning tasks, especially in regard to crack detection in concrete structures.

2.2.6.1 Adaptive feature retention

One of the main benefits of the T-Max-Avg Pooling layer is its flexibility. By combining average pooling, max pooling, and a tunable parameter, the T-Max-Avg approach permits the model to retain the prominent features (as in max pooling) and to reduce noise at the same time (as in average pooling). Parameter T finds a balance between the average and maximum values and makes the model flexible to adapt dynamically to various kinds of features within the pooling window. This flexibility is especially helpful in image segmentation tasks in which both prominent features and nuanced details are vital.

2.2.6.2 Enhancing robustness and reducing noise

Max pooling is useful for capturing the most remarkable features. However, it is susceptible to noise. In contrast, average pooling often obscures the sharp features by averaging. T-Max-Avg Pooling overcomes these limitations by taking advantage of both approaches: it chooses the highest pixel values and calculates a weighted combination of the average and maximum values. This hybrid approach assures that the most significant features are preserved and the effects of noise are mitigated. This leads to more robust feature maps.

2.2.6.3 Improved generalization

The learnable parameter permits the model to generalize better across different datasets by dynamically adapting the pooling behavior based on the specific features of the input data. This means that the T-Max-Avg Pooling can productively handle data variability which is essential for real-world applications like crack detection in which the characteristics of the input images can differ remarkably owing to varied environmental conditions.

2.2.7 Comparing the T-Max-Avg-Pooling method with other state-of-the-art pooling methods

The proposed T-Max-Avg-Pooling layer provides a unique combination of average pooling, max pooling, and an additional learnable parameter (T) which permits the model to adapt the pooling operation dynamically between the dominant features and the feature maps. This flexibility gives T-Max-Avg-Pooling the ability to make a balance between preserving the fine details and reducing noise. This makes it highly effective in different scenarios like crack detection in concrete structures. To ensure the effectiveness and novelty of the proposed method, it is compared below with other advanced pooling methods such as spatial pyramid pooling (SPP) and stochastic pooling. Unlike conventional pooling methods that depend on max or average values, stochastic pooling selects the activation functions according to a multinomial distribution formed over each pooling region. This stochasticity increases the robustness of the model by preventing overfitting, especially in scenarios with small datasets. Nevertheless, the randomness introduced by stochastic pooling can result in variability in the output which might not always be proper for applications such as crack detection in which maintaining a consistently high accuracy is vital (Ju et al., 2023). In contrast, the T-Max-Avg-Pooling layer presents a deterministic yet adaptive mechanism that assures both consistency and robustness by learning how much emphasis should be placed on the average and maximum values within a region. SPP improves feature extraction by permitting the network to pool from different spatial scales which is especially helpful for capturing multi-scale contextual information. It is useful in scenarios in which the input images vary remarkably in size since it provides the generation of fixed-length representations regardless of the input dimensions. Nonetheless, SPP increases computational complexity because of its multi-level pooling. This makes it less appropriate for deployment in resource-constrained environments or real-time applications. In contrast, T-Max-Avg-Pooling finds a balance between feature robustness and computational efficiency, needs fewer resources, and productively captures the features crucial for an accurate segmentation.

2.3 Architecture structure

1. Encoder: The encoder part of U-Net++ consists of multiple stages each of which contains nested dense convolutional blocks followed by down-sampling operations. The T-Max-Avg pooling layer is used to reduce the spatial dimensions of the feature maps and to increase the depth.
2. Nested Dense Convolutional Blocks: Within each stage, multiple convolutional layers are densely connected. The output of each convolutional layer is concatenated with the inputs of subsequent layers, forming dense connections.
3. Decoder: The decoder part of U-Net++ consists of up-sampling stages that gradually restore the spatial dimensions of the feature maps. Each up-sampling stage includes a combination of transposed convolutions concatenated with the corresponding encoder feature maps (skip connections) and nested dense convolutional blocks.

4. Output Layer: The final output layer consists of a 1 × 1 convolutional layer to produce the segmentation map with the desired number of classes.

By integrating these advanced components, U-Net++ achieves superior performance on various image segmentation tasks, making it a powerful tool for medical imaging, remote sensing, and other applications. Figure 3 shows the proposed architecture.

2.4 Training procedure

The model was trained using the following approach:

- Loss Function: Due to its effectiveness in handling imbalanced data scenarios typical in crack detection tasks, the Dice loss function was employed to optimize the model.
- Optimizer: The Adam optimizer was selected due to its adaptive learning rate capabilities and efficiency.
- Learning Rate: An initial learning rate of 0.001 was set. A scheduler was used to adjust the rate dynamically during training.
- Batch Size: A batch size of 16 was chosen to balance the memory constraints and an effective training.

2.5 Evaluation metrics

2.5.1 Accuracy

Accuracy is a measure of how often the classifier is correct. It is the ratio of the number of correct predictions to the total number of predictions.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \tag{2}$$

In terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), accuracy can be expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

2.5.2 Precision

Precision is a metric employed to measure the accuracy of positive predictions made by a model. In the context of classification or segmentation tasks like crack detection, precision measures how often the predicted positive results of the model are correct, focusing on the ability of the model to avoid false positives (incorrectly labeling non-crack areas as cracks).

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

2.5.3 F1 Score

The F1 score is a measure of a test's accuracy. The F1 score, which makes a balance between recall and precision, is selected to evaluate how well the model detects the true positive crack pixels and minimizes the false positives. This metric is vital in crack detection to prevent overestimating the crack regions which could result in needless and expensive maintenance tasks. Moreover, the F1 score gives a balanced view of both false negatives and false positives, which is essential for assuring the reliability of crack detection. It considers both the precision (P) and the recall (R) of the test to compute the score. The F1 score is the harmonic mean of precision and recall.

$$F1\ Score = \frac{2 \times (P \times R)}{P + R} \tag{5}$$

where:

- P is the ratio of the true positive predictions to the total number of positive predictions:

$$P = \frac{TP}{TP + FP} \tag{6}$$

- R is the ratio of the true positive predictions to the total number of actual positives:

$$R = \frac{TP}{TP + FN} \tag{7}$$

2.5.4 Mean Intersection-over-Union (MIoU)

MIoU is a standard metric used for assessing the overall performance of the segmentation models. In crack detection tasks, MIoU presents a thorough measure of how well the predicted crack regions are matched with the actual cracks by considering both over-segmentation and under-segmentation. It provides a general perspective on the performance of the model across the entire image, assuring that the model adequately segments the cracks regardless of their position and size. It calculates the average Intersection-over-Union (IoU) for all classes.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{8}$$

where:A is the set of predicted pixels and B is the set of ground truth pixels.

The MIoU is the mean of the IoU scores for all classes.

$$MIoU = \frac{1}{C} \times \sum \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \tag{9}$$

where:C is the number of classes, A_i is the set of predicted pixels for class i, and B_i is the set of ground truth pixels for class i.

2.5.5 Dice Loss

Dice loss is very suitable for segmentation tasks which use imbalanced datasets. Using Dice loss is common for crack detection in which the crack regions are usually much smaller than the background. By maximizing the overlap between the ground truth and predicted regions, Dice loss assures that the model pays especial attention to these small but crucial features, thereby improving the accuracy of segmentation. Dice loss is a loss function commonly used for image segmentation tasks. It is derived from the Dice coefficient which measures the overlap between two samples. Dice loss is particularly useful for imbalanced datasets.

The Dice coefficient (also known as the Sorensen–Dice index) is given by:

$$Dice\ Coefficient = \frac{2|A \cap B|}{|A| + |B|} \tag{10}$$

where:A is the set of predicted pixels and B is the set of ground truth pixels.

The Dice loss is then defined as:

$$Dice\ Loss = 1 - Dice\ Coefficient \tag{11}$$

2.6 Experimental setup

The experiments were conducted using a high-performance computing system equipped with an NVIDIA GPU to facilitate efficient training and inference. The following details outline the setup:

- Hardware: NVIDIA Tesla T4
- Software: TensorFlow and Keras libraries for implementing and training the deep learning model.
- Splitting the Training and Validation Datasets: The dataset was split into 80% training and 20% validation subsets to evaluate the generalization ability of the model.

3 Results

After tuning the network and its parameters, the network was trained on Google Colab with a T4 VGA, 16 Gigabyte of RAM, and a Ryzen 7 CPU. One hundred epochs were set for training.

In Figure 4 the right chart, the MIoU is shown, whereas in the left chart, the Dice loss is plotted. The blue and orange charts illustrate error in the training and validation phases, respectively.

3.1 The impact of T-Max-Avg Pooling: The results of the ablation study

To further elucidate the impact of the T-Max-Avg Pooling layer on the performance of the model, an ablation study was conducted to compare the proposed pooling method with traditional pooling methods. The study involved four configurations: max pooling, average pooling, mixed pooling, and T-Max-Avg Pooling. Each of them was tested under identical conditions to evaluate its effect on key metrics such as accuracy, Dice loss, MIoU, and inference time. The results, summarized in Table 2, demonstrated a significant improvement when the T-Max-Avg Pooling layer was used. Specifically, the proposed pooling method achieved an MIoU score of 82.1%, an F1 score of 90.12%, and a Dice coefficient of 93.7% which were consistently higher than those achieved by the other pooling methods. The adaptability of the T-Max-Avg Pooling allows it to make a balance between retaining the prominent features (similar to max pooling) and reducing noise (similar to average pooling). This flexibility contributes to its superior performance in capturing intricate crack details which is crucial for an accurate segmentation. In terms of computational efficiency, the T-Max-Avg Pooling also exhibited the shortest inference time (21.01 ms per image) compared to max pooling (30.05 ms), average pooling (28.5 ms), and mixed pooling (29.5 ms). This reduced inference time, combined with enhanced accuracy metrics, positions T-Max-Avg Pooling as an optimal choice for real-time crack detection applications where both precision and speed are critical. Overall, the ablation study underscored the advantages of the T-Max-Avg Pooling layer in achieving a well-balanced trade-off between model accuracy and computational efficiency, making it suitable for practical deployment in real-time infrastructure monitoring scenarios.

3.2 Computational efficiency

The recommended U-Net++ model with T-Max-Avg Pooling showed high computational accuracy and efficiency, making it appropriate for environments with limited resources or real-time applications. The feasibility of the model is analyzed below according to key performance metrics:

3.2.1 Inference time

The inference time of 21.01 ms per image in the proposed method is highly promising for real-time monitoring systems. This speed enables the model to be deployed in applications where timely feedback is critical such as continuous crack monitoring using drones or mobile cameras. Real-time crack detection allows for proactive maintenance, reducing the risk of structural failures. The efficiency of the model also makes it suitable for deployment on various platforms. For real-time applications, the hardware requirements are flexible, ranging from high-performance GPUs to edge devices with moderate computational capabilities. The low memory consumption (7.6 GB) and the relatively low computational requirements (40 GFLOPs) suggest that the model can run effectively on modern GPUs, high-performance edge devices, and even resource-constrained environments with optimized settings. For example, with additional optimization techniques such as model quantization, the model could be deployed on mobile devices or embedded systems, enabling widespread adoption for infrastructure health monitoring. Thus, the inference time and computational efficiency of the proposed method make it very useful for practical scenarios that demand real-time performance as in drone-based infrastructure inspections, where processing speed and accuracy are crucial.

3.2.2 Comparison with other methods

The proposed method consistently outperforms the other models not only in terms of computational efficiency and inference time but also with respect to segmentation performance metrics (e.g., high Dice and precision scores). Models such as EfficientDet and SegNet provide a moderate computational efficiency. However, they are not very accurate in crack detection tasks. This makes the recommended model an optimal choice for environments which require both high accuracy and real-time performance. The recommended U-Net++ model with T-Max-Avg Pooling presents a highly efficient solution for real-time crack detection, balancing high accuracy with low inference time and moderate memory consumption. This balance makes it appropriate for deployment in real-world resource-constrained environments (e.g., embedded systems, drones, and mobile devices) without compromising crack detection performance in concrete structures. All parameters are presented in Table 3. As observed in Figure 5, on identical datasets, the proposed method had a higher accuracy and a lower error

rate compared to the other methods. To evaluate the speed of the proposed method, eight relevant algorithms were run and their results were compared with those of the proposed method.

4 Discussion

4.1 The limitations of the proposed model

Despite its superior performance over other existing models, the proposed U-Net++ architecture with the T-Max-Avg-Pooling layer has its own limitations. One noticeable limitation is its potential difficulty in generalizing to different real-world environments. The datasets employed for training and validation mostly compromise controlled images with well-defined crack features. In uncontrolled environments like those involving various textures, lighting conditions, or obstructions like debris or dirt, the robustness of the model could be challenged, reducing its accuracy. Another limitation of the model is its high computational requirements. Even though the proposed model is more efficient than other benchmarks in terms of memory usage and inference speed, deploying it in environments with very limited memory or processing power like older constrained embedded systems or mobile devices may still present some challenges. In these cases, the computational efficiency of the model might not totally balance the hardware limitations. In addition, the reliance of the model on high-quality input images can affect performance in scenarios in which image quality is compromised owing to suboptimal camera conditions or sensor limitations. In such cases, identifying fine cracks might be more challenging, potentially resulting in false negatives or decreased detection accuracy. Finally, though the T-Max-Avg-Pooling layer enhances feature robustness and extraction, it is also susceptible to parameter tuning. Tuning the pooling parameter (T) improperly may result in suboptimal pooling results, affecting the overall performance of the model. Further automated tuning strategies and optimization could decrease this sensitivity.

4.2 Potential scenarios with suboptimal performance

The proposed method may not have an optimal performance in scenarios with large environmental variations such as outdoor inspections where surface and lighting conditions vary extensively. Moreover, in environments with excessive occlusions or noise, the model may struggle to differentiate accurately between actual cracks and irrelevant features. Such challenges highlight the need for further pre-processing steps or supplementary training data that consider diverse real-world conditions to further improve the robustness of the model.

4.3 The analysis of the impact of environmental factors on the performance of the model

To evaluate the robustness of the model under various real-world environmental conditions, some experiments were conducted to understand how different factors affect the accuracy of crack detection. The performance of the model was tested against variations in noise levels, lighting conditions, and surface debris, which are typical challenges encountered in field applications.

1. Lighting Variations: The model was tested under diverse lighting conditions, ranging from bright sunlight to low-light environments. The results demonstrated that under well-lit conditions, the Dice coefficient was 93.7%, whereas under dim lighting, it dropped to 89.2%. The decrease in accuracy was primarily due to shadow effects and insufficient visibility which hindered precise crack segmentation.
2. Noise Levels: A Gaussian noise was added to simulate real-world scenarios such as interference from dust or sensor noise. By increasing the noise level, the precision of crack detection decreased. For instance, at moderate noise levels, the F1 score dropped from 90.12% to 85.4%. However, using the T-Max-Avg Pooling layer mitigated some of these effects by maintaining a balanced representation of the features, resulting in a better trade-off between precision and noise compared to traditional pooling techniques.
3. Surface Debris and Occlusions: There is often debris such as leaves, dirt, or other materials on concrete surfaces that partially hide the cracks. The recall score of the model decreased from 94.5% to 88.3% when tested with synthetic occlusions. The use of dense skip connections in the U-Net++ architecture was helpful to some extent in distinguishing the cracks from the background noise. However, the performance was still affected by the occlusion of critical features.

These analyses demonstrated that while the proposed U-Net++ model with T-Max-Avg Pooling performed well under controlled conditions, its robustness in real-world environments can be further improved.

5 Conclusion

In this study, an improved U-Net++ architecture featuring the novel T-Max-Avg Pooling layer was introduced, aiming at enhancing the efficiency and accuracy of concrete crack detection. The proposed model demonstrated superior results compared to existing methods, achieving a balanced trade-off between computational speed, memory efficiency, and detection accuracy. Specifically, the model achieved an inference time of 21.01 ms per image, outperforming other well-known models like Mask R-CNN and Deep Crack. This makes it highly suitable for real-time applications such as infrastructure monitoring using drones or mobile platforms. The integration of dense skip connections, learnable pruning techniques, and the T-Max-Avg Pooling layer improved feature extraction, robustness, and computational efficiency. The experiments, which included the Concrete Pavement Crack and DeepCrack datasets, yielded promising results with an MIoU score of 82.1%, a precision score of 87.6%, an F1 score of 90.12%, a Dice loss score of 93.7%, a recall score of 94.5%, and an overall accuracy of 97.65%. These metrics indicated that the proposed architecture not only enhanced segmentation performance but also made it more suitable for deployment in resource-constrained environments.

The proposed U-Net++ architecture, combined with the T-Max-Avg Pooling layer, showed a great potential for automating the process of crack detection in concrete structures, thereby improving maintenance efficiency, reducing manual labor, and minimizing human error.

5.1 Future works

Future studies can focus on optimizing several specific aspects of the current U-Net++ architecture to improve its efficiency and performance. One of the principal goals of future works can be applying quantization techniques to reduce the size and computational requirements of the model, making it more suitable for deployment on edge devices. By changing the parameters of the model from floating-point level to a lower level of precision, it is expected that accuracy is maintained while the inference speed is significantly improved, especially for resource-constrained environments. Furthermore, the potential of transfer learning in enhancing the robustness of the crack detection model can be explored. By pre-training the model on large-scale and diverse datasets beyond those related to concrete, the ability of the model to generalize to different types of materials and surface conditions can be improved. This will involve fine-tuning the model to enable it to detect cracks in various construction materials, such as asphalt, brick, and even metallic surfaces, expanding its applicability beyond concrete structures. In addition, the architecture of the model can be optimized by integrating advanced attention mechanisms. Particularly, the incorporation of self-attention modules can improve the focus of the model on the most important areas of the images, further enhancing its detection performance in complex scenarios. This improves the ability of the model in detecting fine details and subtle anomalies, which is vital for accurate structural evaluations. Finally, the proposed enhancements, including attention mechanisms, transfer learning, and model quantization, aim to make the model more efficient and versatile, enabling real-time crack detection across a wide range of infrastructure conditions and materials.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

AS: Conceptualization, Investigation, Methodology, Software, Writing–original draft. MR: Methodology, Project administration, Writing–original draft. AM: Data curation, Resources, Writing–review and editing. MG: Supervision, Writing–review and editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article. The authors declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Alam, M., Samad, D. M., Vidyaratne, L., Glandon, A., and Iftekharuddin, K. M. (2020). Survey on deep neural networks in speech and vision systems. *Neurocomputing* 417, 302–321. doi:10.1016/j.neucom.2020.07.053

Antony, A. N. M., Narisetti, N., and Gladilin, E. (2023). Fdm data driven u-net as a 2d Laplace pinn solver. *Sci. Rep.* 13, 9116. doi:10.1038/s41598-023-35531-8

Asadzadeh, M. Z., Roppert, K., and Raninger, P. (2023). Material data identification in an induction hardening test rig with physics-informed neural networks. *Materials* 16, 5013. doi:10.3390/ma16145013

Chatzikonstantinou, C., Konstantinidis, D., Dimitropoulos, K., and Daras, P. (2021). Recurrent neural network pruning using dynamical systems and iterative fine-tuning. *Neural Netw.* 143, 475–488. doi:10.1016/j.neunet.2021.07.001

Fan, Z., Wu, Y., Lu, J., and Li, W. (2018). *Automatic pavement crack detection based on structured prediction with the convolutional neural network.*

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2018). "Mask R-CNN," in 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22-29 October 2017 (IEEE).

Ju, L., Zou, X., Zhang, X., Xiong, X., Liu, X., and Zhou, L. (2023). An infusion containers detection method based on yolov4 with enhanced image feature fusion. *Entropy* 25, 275. doi:10.3390/e25020275

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature* 521, 436–444. doi:10.1038/nature14539

Li, G., Fang, Z., Mohammed, A. M., Liu, T., and Deng, Z. (2023). Automated bridge crack detection based on improving encoder–decoder network and strip pooling. *J. Infrastructure Syst.* 29, 04023004. doi:10.1061/JITSE4.ISENG-2218

Liu, Y., Yao, J., Lu, X., Xie, R., and Li, L. (2019a). Deepcrack: a deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing (Amst).* 338, 139–153. doi:10.1016/j.neucom.2019.01.036

Liu, Y., Yao, J., Lu, X., Xie, R., and Li, L. (2019b). Deepcrack: a deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing* 338, 139–153. doi:10.1016/j.neucom.2019.01.036

Liu, Z., Cao, Y., Wang, Y., and Wang, W. (2019c). Computer vision-based concrete crack detection using u-net fully convolutional networks. *Automation Constr.* 104, 129–139. doi:10.1016/j.autcon.2019.04.005

Lyu, C., Fan, X., Qiu, Z., Chen, J., Lin, J., and Dong, C. (2023). "Efficientdet based visial perception for autonomous driving," in 2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA) (IEEE), 443–447. doi:10.1109/ICCCBDA56900.2023.10154715

Nawaz, M., Nazir, T., Javed, A., Tariq, U., Yong, H.-S., Khan, M. A., et al. (2022). An efficient deep learning approach to automatic glaucoma detection using optic disc and optic cup localization. *Sensors* 22, 434. doi:10.3390/s22020434

Nguyen, T.-N., Tran, V.-T., Woo, S.-W., and Park, S.-S. (2022). "Image segmentation of concrete cracks using segnet," in *Intelligence of Things: Technologies and applications*. Editors N.-T. Nguyen, N.-N. Dao, Q.-D. Pham, and H. A. Le (Cham: Springer International Publishing), 348–355.

Nie, E., Yu, J., Dutta, D., and Zhu, Y. (2018). *In silico* simulation of liver crack detection using ultrasonic shear wave imaging. *BMC Med. Imaging* 18, 15–11. doi:10.1186/s12880-018-0249-5

Oluwaseun, O. (2023). *Concrete and pavement crack dataset*. doi:10.34740/KAGGLE/DSV/5130126

Qian, L., Wen, C., Li, Y., Hu, Z., Zhou, X., Xia, X., et al. (2024). Multi-scale context unet-like network with redesigned skip connections for medical image segmentation. *Comput. Methods Programs Biomed.* 243, 107885. doi:10.1016/j.cmpb.2023.107885

Rodriguez-Torrado, R., Ruiz, P., Cueto-Felgueroso, L., Green, M. C., Friesen, T., Matringe, S., et al. (2022). Physics-informed attention-based neural network for hyperbolic partial differential equations: application to the buckley-leverett problem. *Sci. Rep.* 12, 7557. doi:10.1038/s41598-022-11058-2

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: convolutional networks for biomedical image segmentation," in Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015 (Springer), 234–241. doi:10.1007/978-3-319-24574-4-28

Sanjerehei, M. M., and Rundel, P. W. (2020). A comparison of methods for detecting association between plant species. *Ecol. Inf.* 55, 101034. doi:10.1016/j.ecoinf.2019.101034

Shi, J., Dang, J., Cui, M., Zuo, R., Shimizu, K., Tsunoda, A., et al. (2021). Improvement of damage segmentation based on pixel-level data balance using vgg-unet. *Appl. Sci.* 11, 518. doi:10.3390/app11020518

Sjolander, A., Belloni, V., Peterson, V., and Ledin, J. (2023). Experimental dataset to assess the structural performance of cracked reinforced concrete using digital image correlation techniques with fixed and moving cameras. *Data Brief* 51, 109703. doi:10.1016/j.dib.2023.109703

Sohaib, M., Hasan, M. J., Shah, M. A., and Zheng, Z. (2024). A robust self-supervised approach for fine-grained crack detection in concrete structures. *Sci. Rep.* 14, 12646. doi:10.1038/s41598-024-63575-x

Su, C., and Wang, W. (2020). Concrete cracks detection using convolutional neuralnetwork based on transfer learning. *Math. Problems Eng.* 2020, 1–10. doi:10.1155/2020/7240129

Tan, M., Pang, R., and Le, Q. V. (2020). "Efficientdet: Scalable and efficient object detection," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, 13-19 June 2020 (IEEE), 10781–10790. doi:10.1109/CVPR42600.2020.01079

Wang, L., Wang, H., Liang, L., Li, J., Zeng, Z., and Liu, Y. (2023). Physics-informed neural networks for transcranial ultrasound wave propagation. *Ultrasonics* 132, 107026. doi:10.1016/j.ultras.2023.107026

Yan, J., Downey, A., Cancelli, A., Laflamme, S., Chen, A., Li, J., et al. (2019). Concrete crack detection and monitoring using a capacitive dense sensor array. *Sensors* 19, 1843. doi:10.3390/s19081843

Yu, Y., Rashidi, M., Samali, B., Mohammadi, M., Nguyen, T. N., and Zhou, X. (2022). Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm. *Struct. Health Monit.* 21, 2244–2263. doi:10.1177/14759217211053546

Zhang, J., Zhang, Y., Jin, Y., Xu, J., and Xu, X. (2023). Mdu-net: multi-scale densely connected u-net for biomedical image segmentation. *Health Inf. Sci. Syst.* 11, 13. doi:10.1007/s13755-022-00204-9

Zhao, L., and Zhang, Z. (2024). A improved pooling method for convolutional neural networks. *Sci. Rep.* 14, 1589. doi:10.1038/s41598-024-51258-6

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., and Liang, J. (2018). "Unet++: a nested u-net architecture for medical image segmentation," in Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018 (Springer), 3–11. doi:10.1007/978-3-030-00889-5-1

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., and Liang, J. (2020). Unet++: redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* 39, 1856–1867. doi:10.1109/tmi.2019.2959609