IEEE Open Journal of the Computer Society

Received 7 July 2024; revised 8 October 2024; accepted 11 October 2024. Date of publication 16 October 2024; date of current version 28 October 2024. The review of this article was arranged by Associate Editor Wei Liu.

Digital Object Identifier 10.1109/OJCS.2024.3481000

An Innovative Dense ResU-Net Architecture With T-Max-Avg Pooling for Advanced Crack Detection in Concrete Structures

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 14778-93855, Iran
[2]Department of Civil Engineering, Payame Noor University, Tehran 19395-4697, Iran
[3]Department of Civil Engineering, East Tehran Branch, Islamic Azad University, Tehran 15847-43311, Iran

CORRESPONDING AUTHOR: MEHDI RAVANSHADNIA (e-mail: ravanshadnia@srbiau.ac.ir).

ABSTRACT Computer vision which uses Convolutional Neural Network (CNN) models is a robust and accurate tool for precise monitoring and pixel-level detection of potential damage in concrete structures. Using a state-of-the-art Dense ResU-Net model integrated with T-Max-Avg pooling layers, the present study introduces a novel and effective method for crack detection in concrete structures. The major innovation of this research is the introduction of the T-Max-Avg pooling layer within the Dense ResU-Net architecture which synergistically combines the strengths of both max and average pooling to improve feature retention and minimize information loss during crack detection. In addition, the incorporation of Residual and Dense blocks within the U-Net framework significantly enhances feature extraction and network depth, resulting in a more robust anomaly detection. The implementation of extensive data augmentation techniques improves the robustness of the model while the application of spatial dropout and L2 regularization techniques prevents overfitting. The proposed model showed a superior performance, outperforming traditional and state-of-the-art models. It had a Dice Coefficient score of 97.41%, an Intersection-over-Union (IoU) score of 98.63%, and an accuracy of 99.2% using a batch size of 32. These results confirmed the reliability and efficacy of the Dense ResU-Net with T-Max-Avg pooling layer for accurate crack detection, demonstrating its potential for real-world applications in structural health monitoring. By taking advantage of advanced deep learning techniques, the proposed method addressed the limitations of traditional crack detection techniques and offered significant improvements in robustness and accuracy.

INDEX TERMS Crack detection, deep learning, dense ResU-net, image segmentation.

I. INTRODUCTION

Crack detection in concrete structures is a critical and challenging task. The existing cracks in structures are important indicators of the damage and durability in them. Many countries require regular crack detection as part of their construction inspection programs which are typically conducted by a technician visually. The technician records the information including the position, thickness, and other features of the cracks. The periodic inspections of infrastructure systems are crucial for maintaining their service capabilities and ensuring public safety. Structural health monitoring (SHM) is a methodology used to perform structural assessments and involves detecting, locating, identifying, and quantifying damages.

However, these stages often rely on subjective human assessment and can be inaccurate. To address this issue, more quantitative and advanced statistical approaches have been proposed. In terms of deep learning-based techniques, crack detection methods for concrete structures can be divided into 'intelligent' and 'unintelligent' categories [1], [2]. Conventional methods, such as Schmidt Hammer or ultrasonic wave generator devices, are effective in detecting cracks. However, researchers are moving towards more innovative and precise methods. Unintelligent methods include the Schmidt

Hammer method and visual assessment, while intelligent methods include laser image processing, ultrasonic image processing, and digital image processing. Intelligent methods offer some advantages such as not needing direct access to the site or experts and having high precision and speed. However, preparing these systems is also challenging as they require large datasets and robust hardware [3]. Advances in science and technology have allowed humans to create systems (such as artificial intelligence (AI)) that imitate their behavior. AI has many practical applications in our daily lives including image-processing-based crack detection in assessments of structural health and maintenance. AI-based methods use various statistical features of cracks, imaging, and machine learning algorithms to identify, classify, and analyze cracks among other tasks.

The recent integration of deep learning, specifically convolutional neural networks (CNNs), into crack detection methods is a major development in this field. Deep learning offers several advantages over traditional image processing and feature-based machine learning techniques including independence from expert-defined thresholds and engineering features, superior precision, and robust analysis of different images. CNNs are a driving force in the field of computer vision research [4]. Image segmentation, which divides an image into its components, is an important part of visual systems and is used in a wide range of applications including medical image analysis, self-driving vehicles, image monitoring, and virtual reality, among others [4]. In recent years, deep learning-based models have revolutionized image segmentation with significantly improved performance and precision. As a result, the use of these methods for crack detection has increased [5]. Dung investigated cracks in concrete structures and introduced a semantic segmentation approach utilizing CNNs. This research used 40000 images as well as a VGG16 model which resulted in an accuracy of 90%. The proposed model was then tested using experimental footage showing that it was able to determine crack density accurately [6]. Dorafshan et al. employed a hybrid approach combining edge detection and CNNs to detect cracks. The images were preprocessed using various edge detection techniques, such as Laplacian of Gaussian, Sobel, etc., and were processed with an AlexNet-based architecture. The model was trained to detect cracks with a width greater than 0.1 mm and had an accuracy range of 53% to 79%. These researchers also utilized an image classification model with an accuracy of 99% and applied it to a transfer-learning model with an accuracy of 86%. The hybrid model was capable of detecting cracks as small as 0.08 mm, whereas the transfer-learning model was able to detect cracks as small as 0.04 mm. The proposed hybrid model was able to reduce noise effectively [7]. Qing Guo et al. developed a framework for identifying cracks in concrete surfaces using the region-growing and edge detection algorithms. They also utilized an optimization method to improve and speed up the retrieval process. The final algorithm was presented as an integrated approach capable of classifying different cracks based on image topology [8]. Inspired by the ImageNet algorithm, Li et al. conducted a research on transfer learning and created a database by collecting construction images. They implemented a two-step transfer-learning strategy on the organized database and trained the model. Gradient-weighted Class Activation Mapping (Grad-CAM) was used to find the locations of the subsurface cracks. The proposed framework was applied to detect cracks in concrete dam structures and the analysis results showed its excellent performance [9].

The major contribution of this research is the creation of a state-of-the-art Dense ResU-Net architecture by integrating a novel T-Max-Avg pooling layer, particularly devised for detecting cracks in concrete structures. This pooling method combines the advantages of both max and average pooling. Moreover, it enhances the extraction of significant features and reduces information loss which are the main challenges in crack detection tasks. In addition, using Dense and Residual blocks within the U-Net architecture improves feature reuse and network depth. This improves training stability, model performance, and precision in the detection of complex crack patterns. The contributions of this article address the drawbacks of contemporary and traditional deep learning-based methods and set a novel benchmark in the field of automated structural health monitoring.

II. RELATED WORK

Yi et al. explored the use of deep learning to detect cracks in railroads through a rapid crack detection framework. Given the vulnerability of railroads to various environmental factors, fast and accurate crack detection is crucial for their maintenance and repair. The study collected numerous railroad images and developed a deep learning-based network architecture for crack detection. The proposed network, called STCNet 1, reduced the training parameters by 96.03% and 93.28% and the computation time by 49.94% and 73.27%, respectively, compared to the VGG16 and ResNet50 models [10]. Cha et al. conducted a research on the detection of cracks in concrete structures through learning methods. They employed image processing to identify concrete cracks. Lights and shadows were among the most significant challenges that they faced. The method consisted of a CNN architecture trained on a 40000-image database (256 $\times$ 256 pixels) and achieved an accuracy of 98% [11]. Zhu et al. developed a deep learning network for crack detection in concrete bridges which achieved an accuracy of 96% and outperformed other conventional models. The model used a new pooling technique (ASPP) that enabled the network to extract multi-scale context information and reduced the computational costs [12]. Yang et al. proposed an automatic pixel-level crack detection model to address the limitations of existing methods. The deep learning model was trained and evaluated using various crack images. The results showed the quantitative representation of the crack regions whose characteristics such as the crack topology, maximum width, maximum length, and mean crack width were extracted [13]. Hacıefendioğlu and Başağa proposed a Faster-R-CNN architecture-based approach to detect

cracks in concrete structures to extend the lifespan of the structure and address the maintenance issues. The approach was evaluated using 323 images with an aspect ratio of 9:16 and a resolution of 4128 × 2322 under different weather conditions (sunny, foggy, and cloudy). The results showed a similar crack detection performance on sunny and foggy days but a 50% decrease in detection from sunset to moonrise, a 25% decrease from 6:00 PM to 7:00 PM, and an 85% decrease from 7:00 PM to 8:00 PM [14].

Wang and Su studied automatic concrete crack segmentation using transformers. They employed new transformer-based architectures to supervise and inspect concrete structures. The new model, known as SegCrack, was implemented with a hierarchical structure for image segmentation at the pixel level using transformer encoder networks. The encoder layers were utilized to identify the cases better and to improve feature extraction. According to the results of the proposed model, the recall, F1, and MIoU metrics were 96%, 95.46%, and 96.05% on the testing data, respectively [15]. Xu et al. developed an intelligent crack detection method for the maintenance procedure. They compared and adopted the best method using Faster R-CNN and Mask R-CNN. The results indicated that being trained on merely 130 images, these two techniques outperformed YOLO v3 under identical conditions. Additionally, a training method was developed to improve the results [16]. Fan et al. proposed a method to tackle the noise issue using a parallel ResNet. They examined the proposed method on two databases, i.e., CrackTree and CFD. The results of recall, precision, and F1 for CrackTree were 92.55%, 94.27%, and 93.08%, respectively, while for CFD, they were 96.21%, 95.12%, and 95.63%, respectively. In addition, the crack morphology was extracted from the images to measure the crack length, width, and area accurately so that the geometry of the crack could be determined [17]. Lui and Wang proposed a model to detect concrete cracks using the U-Net algorithm to have better visual descriptions. Despite the frequent use of CNNs due to their significant advances, they are hard to interpret and are considered black boxes. Thus, a U-Net-based model was proposed to improve the image descriptions. It exploited simple CNN models, including UNet-inceptionResNetv2, UNet-VGG19, and UNet-EfficientNetb3 (sorted by their output accuracy), for complex computations. However, several limitations, such as rough or dark backgrounds, affected the output results [18]. Barisin et al. presented a model to identify cracks in 3D images. This article used all classical image processing techniques including edge-detection filters and region-growing algorithms. Furthermore, learning methods such as deep learning and random forest (RF) were utilized and evaluated. The evaluation results showed that the learning methods had a better detection, especially with respect to thin cracks with low-contrast images [19]. Rajadurai and Kang presented a method to detect concrete cracks automatically at a high rate using image processing techniques based on machine vision and machine learning methods. Using the AlexNet architecture, this method classified images into two categories, i.e., 'crack' and 'no-crack'. Some modifications were performed to improve the final accuracy, such as tuning the neuron weights, displacing the final layer for binary classification, and augmenting the data with stochastic variations. In addition, this method used stochastic gradient descent (SGD) for optimization and reached an accuracy of 99.9% with a 0.1% error rate [20]. Reghukumar and Anbarasi studied deep learning and image processing methods for concrete crack detection. In this research, to better detect cracks, a method based on Mask R-CNN integrated with an active contour model and a Chan–Vese algorithm was applied on 40000 training data. The output results of these models showed that deep learning methods were good alternatives for visual concrete crack inspection methods [21].

Ji et al. recommended a technique for crack detection using deep CNNs optimized by an enhanced chicken swarm algorithm which had a higher accuracy and robustness than conventional optimization methods. This approach emphasized the potential of combining CNN architectures with evolutionary algorithms to improve performance in demanding structural conditions [22]. Zhou et al. offered a set of deep convolutional neural networks paired with decision-level data fusion for the evaluation of coating and corrosion drawbacks in coal handling and preparation plants (CHPP). This approach permits the combination of multiple CNN models which enhances the detection capability by taking advantage of different model strengths. This improves the robustness of defect detection [23].

Moreover, Wang et al. presented a vision-based concrete crack detection technique which incorporated a hybrid approach that considered the effect of noise on detection accuracy. By integrating various noise reduction methods into the deep learning model, the recommended hybrid framework obtained a significant improvement in detecting cracks under varying real-world conditions. This showed the significance of considering noise effects in practical applications [24].

These recent studies provide insights into the diverse methodologies and innovations being employed in the field of defect detection using deep learning, which include optimization techniques, ensemble models, and noise-handling mechanisms. Incorporating these advancements into the current literature review will provide a more comprehensive overview of the state-of-the-art techniques, showing how the proposed Dense ResU-Net with T-Max-Avg pooling layer aligns with and advances upon recent developments.

The detection of cracks in concrete structures plays a crucial role in ensuring their health as well as public safety. Conventional inspection methods are laborious, inconsistent, and imprecise for detailed analysis. The present study introduces an advanced deep learning technique utilizing a Dense ResU-Net model for an effective crack detection through image segmentation. The motivation for this research arises from the need for more reliable and automated inspection methods that can supplement manual processes. Significant contributions include the design and integration of dense and residual blocks within the U-Net framework with T-Max-Avg

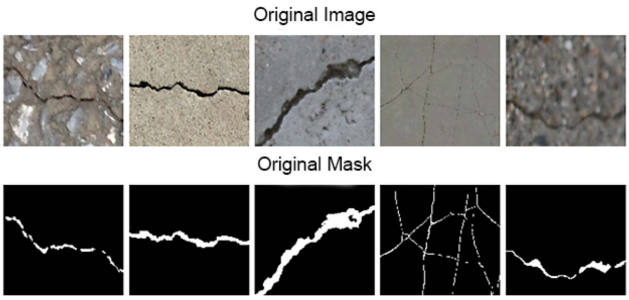

FIGURE 1. A few examples of the used images.

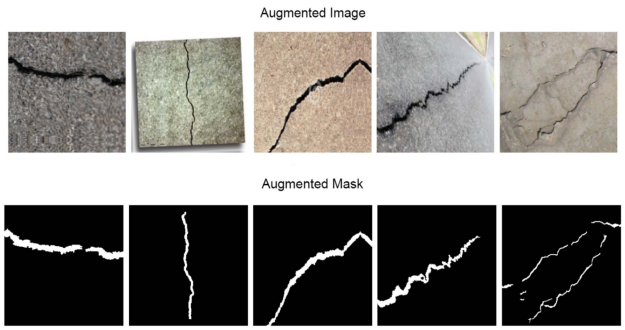

FIGURE 2. Samples of augmented images and masks.

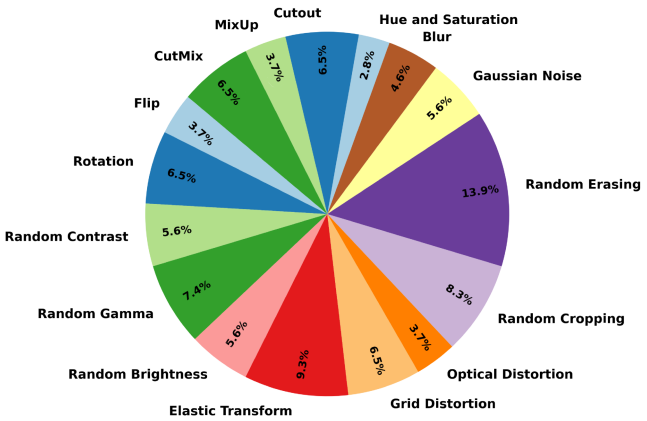

FIGURE 3. Effectiveness of data augmentation techniques based on percentage.

pooling layers, presenting sophisticated data augmentation methods to enhance the training process, and the application of spatial dropout and L2 regularization techniques to improve the model generalization. The proposed method has been rigorously tested and compared with three benchmark models and other related researches, demonstrating its superior performance in terms of accuracy and robustness.

III. MATERIALS AND METHODS

A. DATASETS

This study utilized the four datasets, containing 30000 training images and 25000 testing images with corresponding masks, which indicates the relevant sections of the image. Fig. 1 shows some of the data used in this study on a random basis. These data are original images and their masked versions are shown as grayscale images. The dataset was also

TABLE 1. Random Changes to the Images With Their Descriptions

Change type	Description
Flip	Mirror flipping with a 0.7 limitation
Rotation	Rotating around vertical or horizontal axes with a 0.7 limitation
Random Contrast	Modifying the image contrast at random
Random Gamma	Increasing the image luminance
Random Brightness	Changing the image brightness to emulate day and night
Elastic Transform	Image elongation and foreshortening (making waves)
Grid Distortion	Zooming out the image and creating a shadow
Optical Distortion	Image formation by convex and concave lenses
Random Cropping	Randomly cropping a portion of the image to create variations in the viewpoint and aspect ratio
Random Erasing	Randomly erasing parts of the image and simulating occlusions
Gaussian Noise	Adding a Gaussian noise to the image to simulate real-world conditions with varying noise levels
Blur	Applying different types of blur (Gaussian, Median, and Motion) to simulate various camera or environmental conditions
Hue and Saturation	Adjusting the hue and saturation to simulate different lighting conditions and color variations
Cutout	Randomly cutting out the square sections of the images forcing the model to focus on the remaining parts
MixUp	Combining two images by taking a weighted average of their pixel values and their labels
CutMix	Replacing a random portion of one image with a patch from another image and adjusting the labels accordingly

configured to deal with complex background scenarios such as hands, building accessories, non-focused objects, stones, leaves, etc. which commonly occur. The surface damage was obtained from cylindrical concrete, mortar, beam test, and concrete wall specimens. Since there were few training data, the image data augmentation technique was utilized for training the network. Data augmentation is a technique for adding new simulated data to reinforce and train the network by applying some changes to the existing images without the need to provide new original images. Here, the purpose was to produce new images with new features enabling the network to learn better beyond the original images.

The changes made to the images included shifting, rotating, magnifying, and introducing intentional noise. CNNs can learn the features of an image regardless of their positions in the image. Therefore, the data augmentation technique can facilitate learning and yield a well-trained network. Typically, this technique is also applied to the training data. As mentioned before, the data augmentation technique was used in this study to insert new data into the training process. The changes made to the images at random, along with their descriptions, are listed in Table 1.

As shown in Fig. 2, samples of augmented images and their corresponding masks were used for training. Fig. 3 illustrates the effectiveness of these data augmentation techniques based on percentage improvement.

Validating the augmented data is crucial to ensure that the data augmentation process enhances the performance of

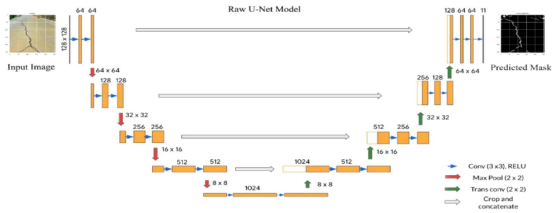

FIGURE 4. Architecture of the U-Net algorithm.

the model without introducing bias or errors. To validate the augmented data effectively, the following methods were used:

IV. VALIDATION OF THE AUGMENTED DATA

A. VISUAL INSPECTION

Random Sampling: A random sample from the augmented images was visually inspected to ensure that the transformations were applied correctly and the images were still realistic and relevant.

Comparison with the Original Image: The augmented images were compared with the original images to check for any anomalies or unrealistic artifacts.

B. STATISTICAL ANALYSIS

Analysis of Distribution: The distributions of pixel values, image features, or other relevant metrics were analyzed before and after augmentation to ensure consistency.

Feature Consistency The augmented images were checked to ensure that the essential features of the original images were retained and that the augmentation did not distort the important aspects of the data.

C. CONSISTENCY CHECKS

Class Consistency: It was ensured that the augmented data did not change the class labels and maintained the integrity of the labeled data.

Augmentation Consistency: Augmentation techniques were applied consistently across the dataset to prevent the introduction of any biases by an uneven application.

V. MODEL ARCHITECTURE

The model utilized in this study was an improved U-Net network designed in 2015 to enhance speed and precision. Since this network had a fully connected layer, it did not require a large number of image segmentation data [25]. As the following image shows, the model was a fully symmetric network in which the numbers of shrinking and expansion layers were equal. Any information removed in the shrinking path was recovered in the expansion path.

In Fig. 4, the shrinking and expansion paths are thoroughly illustrated. As can be seen, convolutional layers in the 3×3 shrinking path were used to apply filters to the image. After every convolutional layer, there was a rectified linear unit (ReLU) activation function layer, also called 'the nonlinear activation function'. A 2×2 Max Pooling layer slid with a stride of 2 over the extracted features. In this research, some colorful images were gathered and integrated as training data to have multiple types of detection.

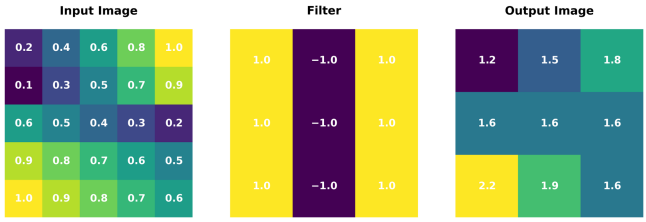

FIGURE 5. The operation of the convolutional layer: - Input image: The initial 5x5 matrix with specific numerical values. - Filter: The 3x3 convolutional filter with predefined values. - Output image: The resulting 3x3 matrix after applying the convolution operation, showing the feature map.

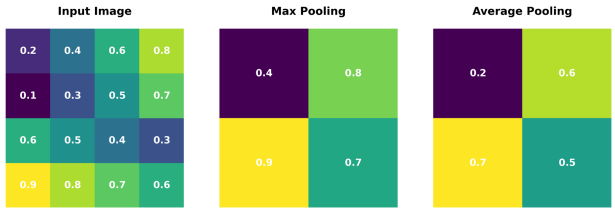

FIGURE 6. The pooling layer operation - Input image: The initial 4x4 matrix with specific numerical values. - Max pooling: The maximum value from each 2x2 region of the input image. - Average pooling: The average value from each 2x2 region of the input image.

Equation (1) represents the characteristic equation of the i-th mapping:

$$Y_i = \sum_{j=0}^{d} F_j = (X_i) + B_i \tag{1}$$

where Y is the characteristic mapping, X is the input tensor, B is the bias vector shared with the output characteristic mapping, and d is the input tensor depth.

The convolutional operation has pre-defined kernels and strides. The images are processed via the application of nonlinear activation functions on convolutional layers. According to their computational performance, the sigmoid, hyperbolic tangent, arc tan, and ReLU functions are successful in deep neural network (DNN). To simplify the calculations, facilitate the training process, and enjoy its simple derivations, the ReLU function was used in this research [26].

The pooling layer reduces the size of the processing data, the main approach for data reduction. There are two types of pooling operation: max and average pooling. Max pooling calculates a single maximum value for a window/kernel by sliding over the input tensor stride by stride, whereas the average pooling calculates a single average value [27]. Fig. 5 demonstrates the convolution operation process, showing how the feature map is generated from the input image. The pooling operation, which reduces the data size using max and average pooling, is depicted in Fig. 6.

A. T-MAX-AVG POOLING LAYER

The pooling layer reduces the size of the processing data. It was used as the main approach for data reduction in the

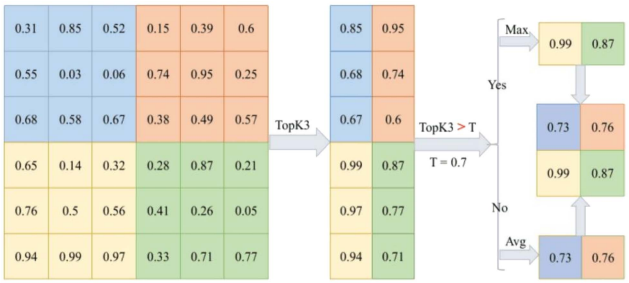

FIGURE 7. 3 × 3 T-Max-Avg pooling [25].

present study. According to a new research, T-Max-Avg pooling layer was used in the present study. The T-Max-Avg pooling method continuously selects the K highest pixel values from the input data and uses parameter T to control the output which is a combination of the average and maximum values of these K highest pixel values. This new pooling approach aims to leverage the strengths of both max and average pooling, providing a more robust feature extraction mechanism [28].

$$F\left(T-\text{Max}-\text{Avg}\left(X\right)\right)=T\times\max\left(Y1,Y2,\ldots,YK\right)+\left(1-T\right)\times\text{avg}\left(Y1,Y2,\ldots,YK\right)\quad(2)$$

Where:

X represents the set of elements with pixels selected based on the pool size values obtained from the convolutional layer data.

Y_i represents the i-th largest item in X.

T is a parameter which ranges from 0 to 1 and balances the contributions of the maximum and average values.

K represents the number of high-value items.

$F(T-\text{Max}-\text{Avg}(X))$ represents the final result.

Fig. 7 illustrates the operations for K=3 in a T-Max-Avg pooling operation with a filter size of 3 × 3 in a 6 × 6 pixel input.

These instances show that the T-Max-Avg method yields values according to both maximum pooling and Avg-TopK. The values acquired from this method can derive the most significant information from the image to a large extent [25].

Also known as transposed convolutional layers, non-convolutional layers are considered as backpropagation, up-sampling, and fractal layers. A non-convolutional layer is a transposed convolutional layer with a specific stride size and layering that transforms the coarse input tensor into a dense output tensor. First, every element in the input tensor is multiplied by the non-convolutional kernel. Then, these intermediate matrices are aggregated with long strides in both horizontal and vertical axes. If the coefficients overlap, they are aggregated to build an expanded matrix from the input. Finally, the matrix is cut to the required size and the bias vector is added. The output size is larger than the input size which leads to an efficient upsampling approach. The utilization of non-convolutional layers is the same as that of standard convolutional layers. However, non-convolutional layers use the typical transposed filter.

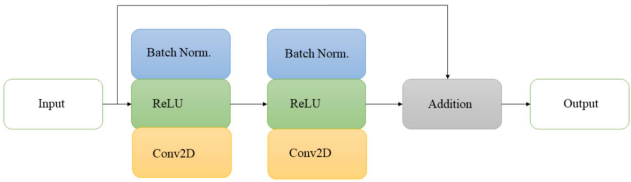

FIGURE 8. A sample residual block.

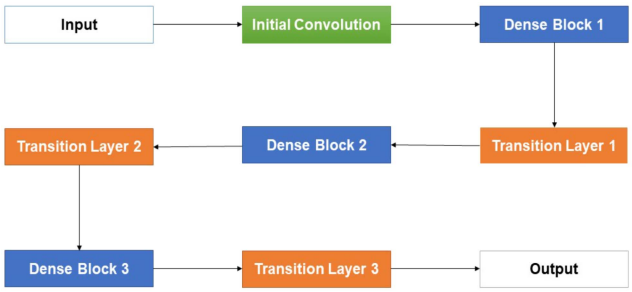

FIGURE 9. A sample Dense block.

B. RESIDUAL BLOCKS

ResU-Net is a variation of the standard U-Net network. It was first used for the semantic segmentation of roads in aerial images. Moreover, due to its high performance, it has also been used in other fields. The aim of designing this network is to increase efficiency with fewer trainable parameters using the advantage of residual blocks along with the U-Net structure. Using residual blocks allows for designing deeper networks with easier training and without worrying about vanishing or exploding gradient problems. Additionally, the skip connections in this architecture help to improve the flow of data between layers. Like the U-Net network, the Res-Unet network has an encoder path, a decoder path, and a bridge that connects these two paths. However, unlike the U-Net network which uses 3 × 3 convolutional layers with ReLU activation function, this architecture uses residual blocks [29].

C. DENSE BLOCK

A dense block is a chief part of the Dense ResU-Net architecture designed to improve feature reuse as well as gradient flow in the network. The dense block was inspired by DenseNet and comprises several layers each of which receives input from all preceding layers and gives its own feature maps to all following layers. This dense connectivity pattern leads to a considerable boost in the flow of information as well as an efficient feature reuse [30]. Fig. 8 presents a sample residual block, while Fig. 9 shows a dense block, highlighting the dense connectivity used in the architecture.

1) KEY CHARACTERISTICS OF A DENSE BLOCK

Dense Connectivity: Each layer is linked to all other layers in a feed-forward manner. This signifies that the input to each layer encompasses the outputs of all previous layers within the same block.

Feature Reuse: By concatenating the feature maps from all preceding layers, each layer can use the features learned by earlier layers. This promotes feature reuse and improves efficiency.

Improved Gradient Flow: Dense connections help reduce the vanishing gradient problem by assuring better gradient propagation in the network.

Reduced Overfitting: The improved feature reuse and flow help reduce the probability of overfitting and make the model more robust [30].

2) STRUCTURE OF A DENSE BLOCK

A common dense block comprises several convolutional layers each of which is followed by batch normalization and activation functions. The inputs of all following layers are concatenated with the output of each layer. This leads to a growth of feature maps as we advance into the block [27].

Encoder

Like the U-Net network, the encoder is responsible for extracting pixel-level features. This architecture employs three encoder blocks which are placed one after the other. The output of each of these blocks is connected to the decoder layer as a skip connection. To decrease the dimensionality of the feature vectors extracted from each layer, the first convolution layer in the blocks employs a stride of 2. This reduces the dimensions from 256 to 128.

Bridge

In this architecture, a residual block with a stride of 2 is utilized.

Decoder

The task of the decoder part is to receive the feature maps from the bridge and the skip connections from different layers and to learn the best semantic representation for illustrating the segmented mask. This part of the architecture is also composed of three decoder blocks. After each block, the dimensions of the feature maps double and the feature channels decrease.

D. REGULARIZATIONS

There are two important aspects to consider regarding the dataset. First, it is relatively small in size. Second, the images in the dataset have a complex foreground which could lead to overfitting during the training and validation phases. To overcome this issue, regularization techniques are employed to enhance the generalization ability of neural networks. In this study, two regularization techniques (namely spatial dropout and L2) were used.

1) SPATIAL DROPOUT

To address overfitting in the image recognition task, the spatial dropout technique was proposed as a means of randomly deactivating a feature, leading to model averaging. However, since adjacent pixels in images are often highly correlated, this technique may not regularize the model effectively. To overcome this issue, the spatial dropout technique was introduced by Tompson et al. [28]. This technique works by deactivating the entire feature maps rather than the individual pixels. Several studies have shown that using the spatial dropout technique in CNN models can improve performance. In this study, the U-Net model uses the spatial dropout technique with a rate of 0.5 which is implemented after every convolution layer, following the approaches in previous studies.

2) L2 REGULARIZATION

L2 regularization owes its name to the L2 norm of the vector w. The two-dimensional norm is as follows:

$$\|w\|_2 = \left(|\omega_1|^2 + |\omega_2|^2 + \ldots + |\omega_N|^2\right)^{\frac{1}{2}} \tag{3}$$

The 2-norm (also known as the L2 norm or the Euclidean norm)

A linear regression model that implements the (squared) L2 norm for regularization is called 'the ridge regression'. To implement the model, note that the linear regression model stays the same:

$$\hat{y} = \omega_1 x_1 + \omega_2 x_2 + \ldots + \omega_N x_N + b \tag{4}$$

However, it is the calculation of the loss function that includes these regularization terms:

$$Loss = Error\,(y, \hat{y}) + \lambda \sum_{i=1}^{N} \omega_i^2 \tag{5}$$

The last equation (ridge regression) is a loss function with the squared L2 norm of the weights (note the remainder of the square root). The regularization terms are constraints to which an optimization algorithm must adhere when minimizing the loss function besides having to minimize the error between the true y and the predicted $\hat{y}$.

E. POST-PROCESSING TECHNIQUES

Though chiefly created for crack detection, the recommended Dense ResU-Net model has the ability to assess crack features using post-processing steps. After the model has detected the crack regions, different image processing and analysis techniques are utilized to obtain the geometric features of the cracks. The following steps are particularly taken:

1) *Crack Length Estimation:* After the crack is detected, the connected components labeling method is employed to trace the path of the crack. The crack length is calculated by adding up the pixel distances along the identified crack path. To carry out real-world measurements, the pixel-based length is scaled according to established reference dimensions from the input image.
2) *Crack Width Calculation:* The crack width is assessed by computing the distance between the crack boundaries at different points along its length. A skeletonization process is employed to reduce the crack to a single-pixel width. Afterward, the perpendicular distance to the crack edges is computed at various locations. Then, the average and maximum widths are obtained from these measurements which can indicate the severity of the cracks.

3) *Crack Orientation Analysis:* The crack orientation is obtained by fitting a line to the identified crack points by employing such techniques as the Principal Component Analysis (PCA) or the Hough Transform. The angle of this fitted line with regard to a reference axis gives the crack orientation which is essential for a better understanding of the potential effect of this method on structural integrity.

By using these post-processing methods in tandem with the crack detection capabilities of the Dense ResU-Net model, a thorough assessment of the crack features is carried out. This makes a more detailed evaluation of structural safety beyond simple crack detection possible, providing quantitative measurements of crack characteristics that are vital for repair and maintenance decisions.

VI. EVALUATION METRICS

There are four commonly used evaluation measures for the classification task including true positive (TP), true negative (TN), false positive (FP), and false negative (FN). In the multi-class segmentation problem, each class is evaluated separately by considering the other classes in the foreground. For instance, when calculating the parameters for cracks, TP indicates that both the ground truth and the prediction are cracks, TN means that both the ground truth and the prediction are non-cracks, FP signifies that the ground truth is a non-crack, while the prediction is a crack, and FN means that the ground truth is a crack, while the prediction is a non-crack [29].

A. ACCURACY

The accuracy metric shows the percentage of correct predictions the model makes which is calculated by (6) [30].

$$Accuracy = (TP + TN) / (Total\ examples) \tag{6}$$

B. DICE LOSS

The Sorensen–Dice coefficient, also known as the Dice similarity coefficient, was used for Boolean data types. Today it is used for the evaluation of image segmentation. This metric is defined by (7) based on the classification of each voxel (the smallest structural component in a 3D image) [34]:

$$DSC = 2TP/(2TP + FP + FN) \tag{7}$$

C. IOU

The IOU metric shows the overlapping extent of two areas. The more extensive the overlap, the larger the IOU. It is typically used in image segmentation systems and is calculated by (8) [35].

$$IOU = \frac{\text{Area of the intersection of two boxes}}{\text{Area of the union of two boxes}} \tag{8}$$

D. ADAM OPTIMIZER

The primary objective of an optimizer is to minimize the errors of a neural network model by performing the required learning steps through a gradient descent algorithm. These learning steps use the backpropagation method to modify the weight and bias in the neural network. The Adam optimizer is a widely used optimizer that has proven to be highly effective for most deep learning tasks. Furthermore, it is regarded as one of the top-performing optimizers based on evaluations [31], [34].

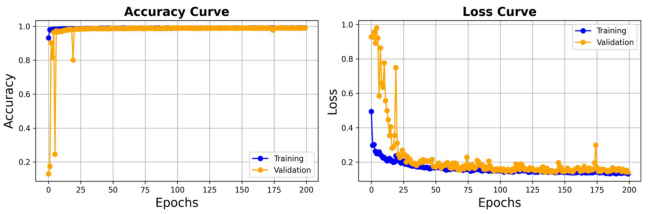

FIGURE 10. The accuracy and error of the network.

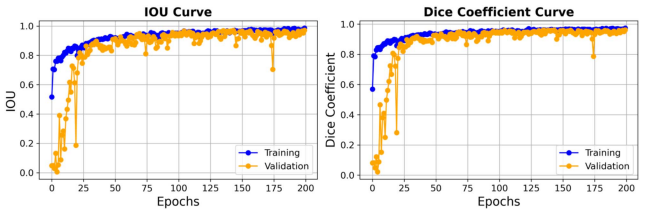

FIGURE 11. The left graph displays the IOU error rate and the right graph illustrates the dice coefficient error rate.

E. HARDWARE AND SOFTWARE CONFIGURATIONS

The network was trained after its parameters were tuned. The network was trained on Google Colab with a 16 GB NVIDIA Tesla T4, and a Ryzen 7 CPU. The number of epochs for training was set to 200. TensorFlow (ver. 2.16.0) and PyTorch (ver. 2.0) were used for the calculations. However, due to its simplicity and efficient computation, TensorFlow was chosen as the software API.

VII. RESULTS AND DISCUSSION

The datasets used in this study contained 30000 training images and 25000 testing images of concrete cracks, each with its corresponding mask. These images were collected from three different databases on Kaggle, cumulated together, and randomly selected for training and testing to ensure a diverse and comprehensive dataset. The images were augmented using 16 different techniques to enhance the robustness of the model.

In Fig. 10, the left chart illustrates the accuracy of the model in the training phase. The blue line shows the accuracy of the model in the training phase, while the orange one indicates its accuracy in the validation phase. The right chart shows the network error rate. Similarly, the blue line shows the training phase error, whereas the orange line indicates the validation phase error.

In Fig. 11, the IOU is shown in the left chart and the Dice loss is plotted in the right chart. The blue and orange lines illustrate the error in the training and validation phases, respectively.

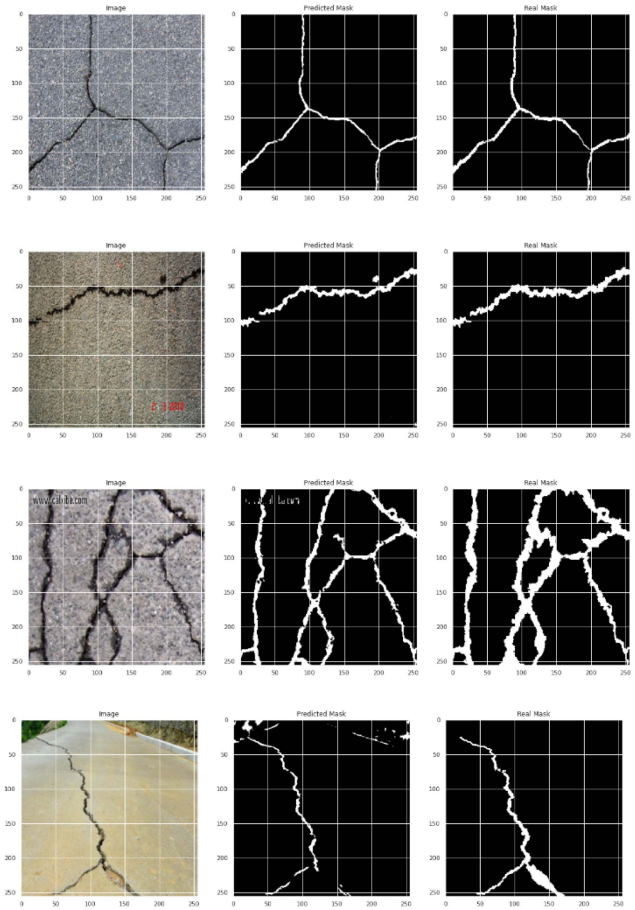

FIGURE 12. Samples of the input image, predicted mask, and real mask.

The evaluation results of the testing data can be seen in Fig. 12. The original image, the masked image, and the predicted image are depicted.

A. COMPARATIVE ANALYSIS OF STATE-OF-THE-ART PAPERS

Dung achieved an accuracy of 90% using a VGG16 model and Zhu et al. reported an accuracy of 96% using a new pooling technique (ASPP), the proposed model in the present study demonstrated a higher accuracy and better performance metrics. Additionally, the proposed method showed significant improvements over the hybrid approaches of Dorafshan et al. as well as the CNN-based methods of Cha et al. and Fan et al. Which can be seen in Fig. 13. Table 2 presents a comparison of the proposed method with existing benchmark models, highlighting improvements in accuracy and robustness.

The proposed model was particularly created to improve robustness against different types of noise that may impact the images of concrete cracks. To do this, multiple strategies were utilized in the development and evaluation phases of the model. First, several data augmentation methods including random transformations, varying brightness, and adding Gaussian noise were applied to the training dataset to simulate real-world conditions in which noise is unavoidable. These augmentations assured that the model was exposed to various noise scenarios to improve its ability to generalize to noisy

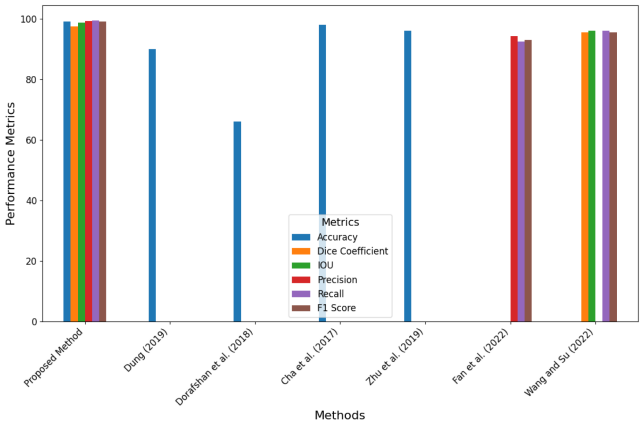

FIGURE 13. Comparison of the errors of the methods.

TABLE 2. Comparison of the Proposed Method With the Benchmarks

Metrics	DICE	IOU	Accuracy
Faster R-CNN [37]	--	70%	98.6%
YOLOv3 [38]	90%	--	97%
RALU-Net [39]	--	64.47%	98%
Proposed method	97.41%	98.63%	99.2%

data. Furthermore, the hybrid pooling mechanism (T-Max-Avg pooling) had an essential role in improving the robustness of the feature extraction process. By taking advantage of the strengths of both max and average pooling, the model was better able to retain the relevant features while mitigating the effect of noisy or unrelated information in the feature maps.

During assessment, some experiments were precisely performed to evaluate the model performance under different levels of noise. According to the results, the recommended model had a high accuracy and a stable performance across different noise levels, showing its robustness. Compared with baseline models, the Dense ResU-Net demonstrated less degradation in performance when it was subjected to noisy inputs. This can be ascribed to both the noise-robust training process and the architecture design.

To sum up, the recommended model showed a strong robustness against noise effects through the use of targeted evaluation, hybrid pooling mechanism, and data augmentation under noisy conditions. These measures assured that the model was dependable for real-world applications where noise is a typical challenge.

VIII. CONCLUSION

In this study, an innovative and efficient method was introduced for detecting cracks in concrete structures using a Dense ResU-Net model with T-Max-Avg pooling layer. This approach addressed the limitations of traditional crack detection methods by leveraging deep learning techniques to achieve high accuracy and robustness in image segmentation tasks. The proposed Dense ResU-Net architecture underwent a systematic and rigorous hyperparameter tuning process

to ensure optimal performance. A combination of grid and random search methods was used for tuning the critical hyperparameters including the learning rate, batch size, dropout rate, and the number of filters in each convolutional layer. The batch size also varied (16, 32, 64) to make a balance between the training speed and stability of the model. The learning rate was adjusted in a range between 0.0001 and 0.01 to determine the most effective value for faster convergence while avoiding overfitting or underfitting.

Cross-validation was utilized to validate the performance of the model across different parameter settings. During the training phase, a learning rate scheduler was employed to adjust the learning rate dynamically based on the validation loss, further improving the convergence and stability of the model. The dropout rate was carefully selected (ranging from 0.2 to 0.5) to prevent overfitting, especially given the complex nature of the dataset used for crack detection. Regularization techniques, such as L2 regularization, were also applied to control the model complexity.

The optimizer also played a critical role in achieving a high accuracy. After experimenting with various optimizers, including Adam, RMSprop, and SGD, the Adam optimizer was chosen owing to its adaptive learning rate capabilities. Adam provided the best results in terms of convergence speed and accuracy.

To assure robustness, different experiments were performed to assess the effect of each hyperparameter on the performance of the model. The final hyperparameter settings were selected according to the combination that consistently gave the highest prediction accuracy on the validation set to assure that the model had a good balance for robustness and generalization.

A. DENSE BLOCKS

Dense blocks are a key component of DenseNet designed to improve feature reuse and gradient flow.

Dense Connectivity

Advantage: This connectivity ensures maximum information flow between layers, leading to better feature reuse and learning efficiency.

Feature Reuse

Advantage: This reduces the number of parameters needed and mitigates the risk of overfitting, especially when working with smaller datasets.

Improved Gradient Flow

Advantage: This alleviates the vanishing gradient problem, making it easier to train deeper networks.

B. RESIDUAL BLOCKS

Residual blocks, introduced in ResNet, address the issue of training deep neural networks by allowing layers to learn residual functions.

Skip Connections

Advantage: These connections allow gradients to flow directly through the network. This stabilizes training and enables the construction of very deep networks.

Learning Residual Functions

Advantage: This simplifies the learning process and improves the convergence of the network.

Prevention of Vanishing/Exploding Gradients

Advantage: This prevents the gradients from vanishing or exploding which is a common issue in deep networks.

C. INTEGRATION WITHIN THE U-NET FRAMEWORK

Combining Dense and Residual blocks within the U-Net architecture leverages the strengths of both approaches, enhancing the overall performance of the network for image segmentation tasks.

Enhanced Feature Extraction

Advantage: This combination results in more accurate and detailed feature maps which are crucial for tasks requiring precise segmentation such as crack detection.

Improved Network Depth and Performance

Advantage: This enhances the ability of the network to capture intricate patterns and details in the images.

Better Generalization

Advantage: This is particularly important in real-world applications where the model must perform reliably with diverse datasets.

The experimental results demonstrated the superior performance of the proposed method, achieving a Dice Coefficient score of 97.41%, an IOU score of 98.63%, and the overall accuracy, precision, recall, F1, and sensitivity scores of at least 0.99. These metrics showed that the proposed model outperformed several existing models in the literature and provided more reliable and accurate crack detection capabilities.

D. REAL-WORLD APPLICATIONS AND IMPLICATIONS

Infrastructure Monitoring

The proposed Dense ResU-Net with T-Max-Avg pooling can be effectively deployed for real-time monitoring of concrete infrastructures such as bridges, tunnels, and dams. By providing accurate crack detection, the model can help in the early identification of structural weaknesses, preventing potential failures and enhancing public safety.

Disaster Mitigation and Prevention

In regions prone to natural disasters such as earthquakes or floods, continuous monitoring using this model can provide critical data on the structural health of buildings and infrastructures. This information is vital for disaster preparedness and mitigation efforts, ensuring timely interventions and reducing the risk of catastrophic failures.

E. CONSTRAINTS

While the proposed method demonstrated significant improvements in the accuracy and robustness of crack detection, several constraints need to be acknowledged:

1) *Size and Diversity of the Dataset:* The size and diversity of the dataset used for training and testing play a crucial role in the performance of the model. Although data augmentation techniques were employed to enhance the

training process, the limited size and scope of the original dataset could affect the generalizability of the model to different types of cracks and concrete structures.

2) *Computational Resources:* Training deep learning models, especially those with complex architectures like the Dense ResU-Net with T-Max-Avg pooling layers, requires substantial computational resources. Access to high-performance GPUs and sufficient memory is essential for their efficient training and real-time implementation which may not be readily available in all practical settings.
3) *Generalization to Different Structures:* The model was primarily trained on specific types of concrete structures and crack patterns. Therefore, its ability to generalize across different types of structures, materials, and damage patterns may require additional training with more diverse datasets.

F. FUTURE RESEARCH

The proposed method significantly improved the accuracy and robustness of crack detection. However, future research should focus on extending the capabilities of the model in several key areas. First, expanding the dataset to include diverse types of concrete structures, environmental conditions, and damage patterns will help improve the generalizability of the model to different real-world scenarios. Additionally, future work could explore integrating multi-modal data, such as thermal or ultrasonic imaging, to enable the model to detect subsurface defects that are invisible to the naked eye.

Another potential area for future research is the development of light versions of the model that can be deployed on edge devices for real-time monitoring of infrastructure. This would enable efficient crack detection in remote or resource-limited environments. Moreover, incorporating state-of-the-art self-supervised learning methods could mitigate the need for extensive labeled datasets, which is usually a limitation in training deep learning models for structural health monitoring.

Finally, applying the recommended model for predicting crack propagation over time using temporal data can give helpful insights into structural damage progression. Such predictive capabilities are vital for proactive maintenance and assuring the long-term safety of the infrastructure.

REFERENCES

[1] G. S. Zhou, Y. Pan, X. Huang, D. Yang, Y. Ding, and R. Duan, "Crack texture feature identification of fiber reinforced concrete based on deep learning," *Materials*, vol. 15, no. 11, Jun. 2022, Art. no. 3940, doi: 10.3390/ma15113940.

[2] Y. Lu, X. Qin, H. Fan, T. Lai, and Z. Li, "WBC-Net: A white blood cell segmentation network based on UNet++ and ResNet," *Appl. Soft Comput.*, vol. 101, Mar. 2021, Art. no. 107006, doi: 10.1016/j.asoc.2020.107006.

[3] H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 155–168, Mar. 2013, doi: 10.1109/tits.2012.2208630.

[4] P. Prasanna et al., "Automated crack detection on concrete bridges," *IEEE Trans. Automat. Sci. Eng.*, vol. 13, no. 2, pp. 591–599, Apr. 2016, doi: 10.1109/tase.2014.2354314.

[5] M. Sohaib, S. Jamil, and J.-M. Kim, "An ensemble approach for robust automated crack detection and segmentation in concrete structures," *Sensors*, vol. 24, no. 1, Jan. 2024, Art. no. 257, doi: 10.3390/s24010257.

[6] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Automat. Construction*, vol. 99, pp. 52–58, Mar. 2019, doi: 10.1016/j.autcon.2018.11.028.

[7] S. Dorafshan, R. J. Thomas, and M. Maguire, "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete," *Construction Building Mater.*, vol. 186, pp. 1031–1045, Oct. 2018, doi: 10.1016/j.conbuildmat.2018.08.011.

[8] T. Qingguo, L. Qijun, B. Ge, and Y. Li, "A methodology framework for retrieval of concrete surface crack′s image properties based on hybrid model," *Optik*, vol. 180, pp. 199–214, Feb. 2019, doi: 10.1016/j.ijleo.2018.11.013.

[9] C. Li et al., "A review of clustering methods in microorganism image analysis," in *Information Technology in Biomedicine. Advances in Intelligent Systems and Computing*, vol. 1186, E. Pietka, P. Badura, J. Kawa, and W. Wieclawek, Eds., Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-49666-1_2.

[10] W. Ye, S. Deng, J. Ren, X. Xu, K. Zhang, and W. Du, "Deep learning-based fast detection of apparent concrete crack in slab tracks with dilated convolution," *Construction Building Mater.*, vol. 329, Apr. 2022, Art. no. 127157, doi: 10.1016/j.conbuildmat.2022.127157.

[11] Y.-J. Cha, J. G. Chen, and O. Büyüköztürk, "Output-only computer vision based damage detection using phase-based optical flow and unscented Kalman filters," *Eng. Structures*, vol. 132, pp. 300–313, Feb. 2017, doi: 10.1016/j.engstruct.2016.11.038.

[12] H. Xu, X. Su, Y. Wang, H. Cai, K. Cui, and X. Chen, "Automatic bridge crack detection using a convolutional neural network," *Appl. Sci.*, vol. 9, no. 14, Jul. 2019, Art. no. 2867, doi: 10.3390/app9142867.

[13] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, "Automatic pixel-level crack detection and measurement using fully convolutional network," *Comput.-Aided Civil Infrastructure Eng.*, vol. 33, no. 12, pp. 1090–1109, Aug. 2018, doi: 10.1111/mice.12412.

[14] K. Hacıefendioğlu and H. B. Başağa, "Concrete road crack detection using deep learning-based faster R-CNN method," *Iranian J. Sci. Technol., Trans. Civil Eng.*, vol. 46, pp. 1621–1633, Jun. 2021, doi: 10.1007/s40996-021-00671-2.

[15] W. Wang and C. Su, "Automatic concrete crack segmentation model based on transformer," *Automat. Construction*, vol. 139, Jul. 2022, Art. no. 104275, doi: 10.1016/j.autcon.2022.104275.

[16] X. Xu et al., "Crack detection and comparison study based on faster R-CNN and mask R-CNN," *Sensors*, vol. 22, no. 3, Feb. 2022, Art. no. 1215, doi: 10.3390/s22031215.

[17] Z. Fan, H. Lin, C. Li, J. Su, S. Bruno, and G. Loprencipe, "Use of parallel ResNet for high-performance pavement crack detection and measurement," *Sustainability*, vol. 14, no. 3, pp. 1825–1825, Feb. 2022, doi: 10.3390/su14031825.

[18] F. Liu and L. Wang, "UNet-based model for crack detection integrating visual explanations," *Construction Building Mater.*, vol. 322, Mar. 2022, Art. no. 126265, doi: 10.1016/j.conbuildmat.2021.126265.

[19] T. Barisin, C. Jung, F. Müsebeck, C. Redenbach, and K. Schladitz, "Methods for segmenting cracks in 3d images of concrete: A comparison based on semi-synthetic images," *Pattern Recognit.*, vol. 129, Sep. 2022, Art. no. 108747, doi: 10.1016/j.patcog.2022.108747.

[20] R.-S. Rajadurai and S.-T. Kang, "Automated vision-based crack detection on concrete surfaces using Deep learning," *Appl. Sci.*, vol. 11, no. 11, Jun. 2021, Art. no. 5229, doi: 10.3390/app11115229.

[21] A. Reghukumar and L. J. Anbarasi, "Crack detection in concrete structures using image processing and deep learning," in *Advances in Electrical and Computer Technologies. ICAECT 2020. Lecture Notes in Electrical Engineering*, vol. 711, T. Sengodan, M. Murugappan, and S. Misra, Eds. Singapore: Springer, 2020, doi: 10.1007/978-981-15-9019-1_19.

[22] X. Ji, L. Zhang, and H. Li, "Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm," *Automat. Construction*, vol. 123, 2021, Art. no. 103502, doi: 10.1177/14759217211053546.

[23] Y. Zhou, W. Feng, and Z. Wang, "Corrosion and coating defect assessment of coal handling and preparation plants (CHPP) using an ensemble of deep convolutional neural networks and decision-level data fusion," *Struct. Health Monit.*, vol. 21, no. 6, pp. 1832–1845, 2022, doi: 10.1007/s00521-023-08699-3.

[24] S. Wang, J. Liu, and R. Zhang, "Vision-based concrete crack detection using a hybrid framework considering noise effect," *J. Civil Struct. Health Monit.*, vol. 13, no. 1, pp. 45–59, 2023, doi: 10.1007/s13349-023-00745-8.

[25] W. Wang and C. Su, "Semi-supervised semantic segmentation network for surface crack detection," *Automat. Construction*, vol. 128, Aug. 2021, Art. no. 103786, doi: 10.1016/j.autcon.2021.103786.

[26] R. Vignesh, B. Narenthiran, S. Manivannan, R. Arul Murugan, and V. RajKumar, "Concrete bridge crack detection using convolutional neural network," in *Materials, Design, and Manufacturing for Sustainable Environment. Lecture Notes in Mechanical Engineering*, S. Mohan, S. Shankar, and G. Rajeshkumar, Eds., Singapore: Springer, 2021, doi: 10.1007/978-981-15-9809-8_58.

[27] K. Izak Cabanilla, R. Z. Mohammad, and J. Ernie, "Neural networks with ReLU powers need less depth," *Neural Netw.*, vol. 172, Dec. 2023, Art. no. 106073, doi: 10.1016/j.neunet.2023.12.027.

[28] L. Zhao and Z. Zhang, "A improved pooling method for convolutional neural networks," *Sci. Rep.s*, vol. 14, no. 1, Jan. 2024, Art. no. 1589, doi: 10.1038/s41598-024-51258-6.

[29] Y. Yang, S. Yue, and H. Quan, "CS-UNet: Cross-scale U-net with semantic-position dependencies for retinal vessel segmentation," *Network*, vol. 35, no. 2, pp. 134–153, Dec. 2023, doi: 10.1080/0954898x.2023.2288858.

[30] G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Weinberger, "Convolutional networks with dense connectivity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8704–8716, Dec. 2022, doi: 10.1109/tpami.2019.2918284.

[31] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proc. 2015 IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 648–656, doi: 10.1109/CVPR.2015.7298664.

[32] A. Zhang et al., "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network," *Comput.-Aided Civil Infrastructure Eng.*, vol. 32, no. 10, pp. 805–819, Aug. 2017, doi: 10.1111/mice.12297.

[33] R. Zhang, Q. Xiao, Y. Du, and X. Zuo, "DSPI filtering evaluation method based on Sobel operator and image entropy," *IEEE Photon. J.*, vol. 13, no. 6, Dec. 2021, Art. no. 7800110, doi: 10.1109/jphot.2021.3118924.

[34] D. Ziou and S. Tabbone, "Edge detection techniques-an overview," *Распознавание образов и анализ изображен /Pattern Recognit. Image Anal.: Adv. Math. Theory Appl.*, vol. 8, no. 4, 1998, Art. no. 537, Accessed: Jun. 29, 2024. [Online]. Available: https://inria.hal.science/inria-00098446

[35] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 227–238, Feb. 2012, doi: 10.1016/j.patrec.2011.11.004.

[36] J. Shang et al., "Automatic pixel-level pavement sealed crack detection using Multi-fusion U-Net network," *Measurement*, vol. 208, Feb. 2023, Art. no. 112475, doi: 10.1016/j.measurement.2023.112475.

[37] A. Chaudhuri, "Smart traffic management of vehicles using faster R-CNN based deep learning method," *Sci. Rep.s*, vol. 14, no. 1, 2024, Art. no. 10357, doi: 10.1038/s41598-024-60596-4.

[38] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018. *arXiv:1804.02767*.

[39] C. Yu, J. Du, M. Li, Y. Li, and W. Li, "An improved U-Net model for concrete crack detection," *Mach. Learn. With Appl.*, vol. 10, 2022, Art. no. 100436, doi: 10.1016/j.mlwa.2022.100436.

ALI SARHADI is currently a Visiting Lecturer with Islamic Azad University, Tehran, Iran, and Payame Noor University, Tehran. His research interests include construction engineering and management, image processing, and structure health monitoring (SHM)

MEHDI RAVANSHADNIA is currently an Associate Professor of construction engineering and management with the Science and Research Branch, Islamic Azad University, Tehran, Iran. He is a Member of the Managerial Board of Tehran Construction Engineering Organization, projects, consulting freeway, and railway.

ARMIN MONIRABBASI is currently an Associate Professor with the Faculty of Civil Engineering, Payame Noor University, Tehran, Iran. His contributions have been featured in high-impact journals. His research focuses on concrete artificial intelligence.

MILAD GHANBARI received the Ph.D. degree in construction engineering and management from Islamic Azad University, Science and Research Branch, Tehran, Iran. He is currently an Associate Professor with the Faculty of Civil Engineering, East Tehran Branch, Islamic Azad University.