IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 May 2024, accepted 13 May 2024, date of publication 20 May 2024, date of current version 7 June 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3403389*

RESEARCH ARTICLE

# Optimizing Concrete Crack Detection: An Attention-Based SWIN U-Net Approach

**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 14778-93855, Iran

Corresponding author: Mehdi Ravanshadnia (ravanshadnia@srbiau.ac.ir)

**ABSTRACT** Utilizing convolutional neural network (CNN) models, computer vision technology has become a reliable and powerful tool for detecting potential damage in concrete structures at the pixel level. In this study, an advanced SWIN U-Net architecture was introduced to detect concrete cracks. The model integrated attention-based convolutional neural networks to enhance the speed and accuracy of crack detection significantly. The distinctive features of the SWIN Transformer made the application of the model to images of varying sizes possible while the computational resources were used efficiently. To train the model, a dataset consisting of crack images, each accompanied by a corresponding mask that highlighted the relevant regions within the image, was used. The training data were augmented using Flip, Rotate, Random Contrast, Random Gamma, Random Brightness, Elastic Transformation, Grid Distortion, and Optical Distortion to counter potential overfitting. Additionally, L2 and spatial dropout regularization techniques were applied to the proposed model. The model was fine-tuned using stochastic gradient descent with the Adam optimizer, employing the binary cross-entropy loss function, and a learning rate of 0.001. The model was trained over 100 epochs with an adjustment scheduler. The performance of the model was evaluated using various metrics. Then, it was compared with three benchmarks and impressive results were achieved. Notably, the Dice loss and IOU values were 93% and 79%, respectively. The trained model had the exceptional performance score of 0.99 in accuracy, precision, recall, F1, and sensitivity.

**INDEX TERMS** Attention-based mechanism, concrete crack detection, generalizable model, SWIN U-Net, transfer learning.

## I. INTRODUCTION

Cracks in concrete structures and buildings are highly important as they can directly affect the quality, safety, and lifespan of structures. Therefore, the timely detection of cracks is crucial for mitigating potential risks and making appropriate repairs. In fact, many countries have implemented systematic crack assessment systems as part of their inspection programs. Over the past few decades, various methods have been introduced for crack detection in concrete including manual inspection, non-destructive techniques such as ultrasonic waves, and image processing. With advancements in machine learning and convolutional neural networks, various methods have been employed for automatic crack detection in concrete including the utilization of architectures such as U-Net, SegNet, and PSPNet. However, these methods still have certain limitations [1]. 1- They require a large training dataset: Training deep learning models necessitates large and diverse training datasets which can be time-consuming and costly to collect and annotate; 2- They have generalization limitations: Some deep learning models may face challenges in detecting cracks in images of different sizes. This means that they may struggle to detect cracks in non-standard or larger-sized images; 3- They do not model non-local relationships: Previous models often relied on convolution-based methods that heavily focused on local relationships. This limitation prevented the models from capturing the non-local relationships in concrete images effectively, restricting their

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang.

crack detection capabilities; 4- They have difficulty detecting small and rough cracks: Certain previous methods sometimes had difficulty detecting small and rough cracks accurately. This is particularly important in concrete images as small and rough cracks can indicate serious structural issues.

In this regard, transformers, initially developed for natural language processing tasks, have been integrated into convolutional neural network architectures. These architectures, which operate based on attention mechanisms between input and output, have a significant strength in modeling non-local relationships and capturing certain image features that may not be captured by convolution-based methods [2]. In this architecture, the input image is first transformed into feature vectors or feature maps. For each pixel in the feature map, a feature vector or key point is generated. These feature vectors, using multi-head attention layers and spatial layers, combine and extract important visual information by modeling local and non-local relationships in the image. This approach allows models to improve the relationships between different image features and identify the salient features. It enhances the interpretability of the image features and improves the accuracy of crack detection. Due to the non-local nature of cracks in concrete images, the proposed architecture is capable of modeling the relationships among distant parts in an image. Furthermore, the bidirectional interaction between the features is facilitated in this architecture. This means that each feature in the image can interact with other features. This transfers information across the entire image, improves the detection of crack-related points in it, and increases the detection accuracy.

In summary, using the transformer architecture for crack detection in concrete images improves the interpretability of image features, detection accuracy, and processing speed. This approach enhances the feature interactions and accelerates the modeling of non-local relationships in images. These architectures perform well in various image processing applications including object detection, face recognition, and satellite image analysis. Therefore, it is expected that combining these architectures with convolutional neural networks such as U-Net will further improve the accuracy and speed of crack detection in concrete.

In this paper, the SWIN U-Net architecture is proposed for crack detection in concrete. This architecture enhances the accuracy and speed of crack detection by combining attention-based neural networks and U-Net. The advantages of using this architecture include increased accuracy, higher generalizability (effective crack detection in images of different sizes), improved interaction between the extracted features in each network layer, transfer of learning capabilities, and detection of smaller cracks.

In the following sections, we delve into the details of the proposed architecture, its implementation, and training. We then compare its performance with those of previous methods using a dataset comprising the images of intact and cracked concrete. Finally, we discuss the future prospects and applications of this architecture in the construction industry.

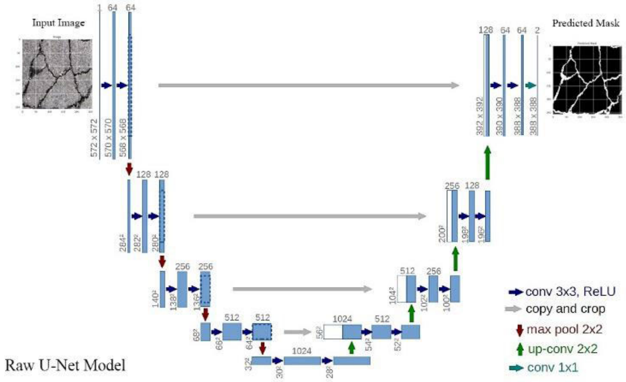

**FIGURE 1.** U-Net network.

## II. U-Net ARCHITECTURE

Ronnenberg et al. [7] were the first to train the U-Net network for the segmentation of biological microscopic images using data augmentation. This network consists of an encoder network that extracts the background image as well as a symmetric decoder network that expands the image. The encoder network, or the contracting path, utilizes a convolutional network for feature extraction. On the other hand, the decoder network, or the expansive path, employs a deconvolutional network to increase the dimensionality of the low-resolution feature maps [7].

In the above figure, the contracting and expansive paths are depicted in detail. In the contracting path, 3*3 convolutional layers which are responsible for implementing filters on the image are used. After each convolutional layer, there is an activation function layer, commonly referred to as the linear activation function. Finally, a max-pooling layer with 2*2 dimensions and moving with a stride of 2 over the extracted features from the filters is included [6].

Dice loss: The Dice similarity coefficient, also known as the Dice coefficient, was originally used for Boolean data in the past but is now widely employed for evaluating image segmentation. This metric defines the similarity coefficient for each voxel (the smallest element of a three-dimensional image) as true positive, false positive, and false negative [6].

$$\boldsymbol{DSC = \frac{2TP}{2TP + FP + FN}} \tag{1}$$

In Eq. 1, TP represents true positives, FP represents false positives, and FN represents false negatives.

This metric is used to measure the overlap between two regions. The higher the overlap, the higher the value of this metric which is calculated using Equation 2.

$$\boldsymbol{IOU = \frac{Area\ of\ intersection\ of\ two\ boxes}{Area\ of\ union\ of\ two\ boxes}} \tag{2}$$

In this equation, the numerator represents the intersection area between the two boxes, whereas the denominator represents the union area of the two boxes under consideration [7].

## III. RELATED WORKS

In general, crack detection methods can be divided into image classification models for distinguishing images with and without cracks and models for detecting the presence or absence of cracks. Image classification methods commonly utilize convolutional neural networks (CNNs) for automatic crack detection in concrete images [1], [2], [4], [5]. These methods typically employ standard CNN architectures which combine convolutional layers, activation functions, pooling layers (in some cases) [1], normalization layers, and fully connected layers. Yang et al. [1] used a CNN on a dataset of 40,000 images with the dimensions of 256*256. To enhance the generalizability of the proposed method, a sliding window technique was employed to detect cracks in larger images exceeding the standard network size. The training results on this dataset demonstrated that the proposed method outperformed older methods (such as edge detection) in terms of accuracy. Moreover, this method had the ability to detect cracks under various conditions such as strong lighting, shadows, and thin cracks.

For crack detection in images, Sallem et al. [2] utilized a combination of two models, a convolutional neural network (CNN) and a region-based CNN with masking. Their proposed method in their article involved using image cropping and Mask-RCNN to downsize the high-resolution images into smaller ones in order to detect smaller cracks and reduce the computational load. In this approach, the high-resolution images were transformed into 512*512 images. The presence or absence of cracks in these images was determined using a crack detection block. Finally, these images were connected to form the original image. This change in image processing and transformation enhanced the accuracy in detecting smaller cracks. However, it prolonged the detection process.

In addition to classification methods, image segmentation methods are also employed in crack detection [3], [6], [9], [11], [13]. The most widely used algorithm in recent years is the U-Net architecture. As mentioned earlier, this architecture consists of an encoder and a decoder. The encoder layers extract the image features using convolutional layers, whereas the decoder layers reconstruct the original image using the extracted features. Lu et al. [3] used a standardized architecture for concrete crack detection in images. A controllable aerial vehicle was used to collect the data. Real images under realistic conditions were used to train the network. Other papers, such as that of Chu et al. [6], focused on detecting small cracks. In this paper, one of the obstacles causing difficulties in detecting small cracks was the class imbalance issue, whereby images containing small cracks formed a smaller class. Consequently, the features of the images containing small cracks were limited. To address this problem, a novel method called ‘Tiny-Crack-Net’ was proposed. This new model included a multi-scale feature fusion network with an attention mechanism which was utilized for extracting the features of small cracks. Additionally, a dual-attention network was incorporated into the main network to better distinguish the small cracks from the background. The task of the multi-scale network was to preserve the details of the crack edges. The proposed mechanism in the article achieved an accuracy of over 91.44% for cracks larger than 0.05 millimeters.

One approach that researchers, such as Jing et al. [9], have used is the combination of classification and segmentation methods. In this article, they proposed a model called CrackDetector, which was a convolutional neural network for detecting the presence or absence of cracks in an image, and a tool called CrackSegmentor which was a standard U-Net architecture that identified the crack regions within the image. The first part of the model was designed to reduce the computational load in the second part. However, it still imposed a computational burden on the overall system.

In a similar vein, other researchers, like Li et al. [11], have employed ensemble learning methods to improve the efficiency of concrete crack detection. In this model, multiple algorithms were used for classification. The philosophy behind using multiple learning algorithms for a single task was to create a strong model from a set of weak models working together. The architectures used in this article included U-Net, DeepLabV3, DeepLabV3+, DANNet, and FCN-8s. Previous research has shown that each of these models has weaknesses that can be covered by the strengths of the others, leading to a higher accuracy compared to using each algorithm individually.

Kim et al. [8] also utilized a similar idea to enhance the accuracy of image segmentation. They used a combination of super-resolution networks and generative adversarial networks (GANs). Generally, the networks developed for segmentation tasks employed an identical encoder-decoder model. In these architectures, the encoder layer extracted the features from the image, while the decoder layer reconstructed the spatial information. The performance of these models was highly dependent on the presence of image noise, blurriness, and even image jitter. However, they might not perform well for small-sized cracks due to the insufficient number of pixels. To address this issue, a two-stage structure was proposed.

These approaches demonstrate the utilization of combined classification and segmentation methods as well as ensemble learning and advanced network architectures to improve the accuracy and efficiency of crack detection and segmentation tasks.

In the first stage, a super-resolution network was used to increase the number of pixels in small regions. This network, called an elliptical approximation network, was applied to the input image to identify and reduce the blurriness of small cracks in low-resolution images. This method enhanced the visibility of the cracks within the image. However, it might not accurately distinguish between the cracks and the background since the boundaries of the cracks, particularly in blurry images, were often irregular. To improve the image quality, the SRGAN network, which predicted the values between two points, was utilized. In this case, the boundary between the cracks and the background was more accurately determined.

In addition to semantic segmentation methods, researchers such as Inam et al. [13] have used sample-based segmentation models such as YOLO for the automated detection of cracks and damages on the surfaces and infrastructures of bridges. In their article, they employed the YOLOv5 model which had a high detection speed. In the YOLO algorithm, initially, the entire image passes through convolutional layers and all objects within the image are identified. The image is divided into N*N grids and if the center of an object falls within a grid, that grid is responsible for detecting the object. This operation leads to the high speed of the YOLO algorithm. Its 98% accuracy demonstrates its effectiveness in crack detection.

Rong et al. [4] developed an improved neural network called CrackSegNet for the pixel-level segmentation of concrete cracks. This network employed convolutional layers for feature extraction and dimension reduction. However, the primary objective of using CNN was to preserve the spatial relationship between the pixels in the images. Lower layers in the network usually extracted the structural details and spatial information from an image, while higher layers extracted the semantic features. The activation functions used in this network were the ReLU functions which enabled the network to capture the nonlinear features. The proposed network had a higher accuracy and lower detection time than segmentation methods like U-Net.

Li et al. [5] improved the U-Net algorithm to enable it to detect fine cracks. The proposed method in their article used a full attention mechanism in the U-Net architecture. This mechanism worked by inserting an attention layer after each convolutional layer. The attention layer connected with the skip connection in the decoder layer. This strategy improved the training process by providing more information to the decoder layer. This additional information acted like a double-edged sword since it contained both useful and noisy information. To prevent the addition of noise to the network, an attention gate was used in each skip connection. In essence, the attention gate was an attention mechanism in which the activation functions were sigmoid functions and its input and output had the same size. The results of the mIoU evaluation metric showed an 85.88% improvement compared to the regular U-Net network.

With the continuous development of deep learning algorithms, newer methods have been developed for crack detection. For example, in Klick and König's article [7], a U-Net-like architecture composed of encoder and decoder networks was proposed. Unlike the U-Net network, the proposed network utilized specific connections between these encoder and decoder blocks. Additionally, an attempt was made to improve the U-Net architecture using the squeeze-and-excitation architecture. The activation function in this network was of the sigmoid type. Alongside this approach, another solution proposed for vertebral models was the use of transfer learning. By employing a pre-trained model trained on a large amount of data, a significant reduction in the F1 error rate could be achieved. The output results demonstrated an F1 error rate of 88.56%.

Other models have also been developed by other researchers such as Chlorney et al. [8] who developed the model for image processing in the pipeline. The model proposed for the pipeline prepared the images in four stages. In the first stage, using a YOLO-v5 model, the cracks inside the image were identified and the rectangles were drawn to indicate the boundaries of the cracks. In the second part, each of the rectangles obtained in the previous stage was cropped to a size of 400*400. In the next stage, all the cropped crack images underwent unsupervised training using the DINO model to extract their features. The final stage of this model involved combining the feature images based on FPN to perform a binary classification for each pixel. Finally, the noises were removed based on the segmentation mask and resized to fit the bounding box. In terms of the MIoU error, the proposed method outperformed other methods with a value of 0.7712.

As evident from the conducted studies, the use of speed-focused models is expanding. U-Net-based networks with attention layers show a higher accuracy compared to the other models and reduce the dependence of the model on a large amount of data. However, in these models, only the skip connections toward the decoder layer have attention mechanisms, while the network layers maintain their traditional structure. In the proposed model, attention mechanisms were utilized in both the encoder and decoder layers. This model, known as Swin U-Net, employs Swin transformer blocks (explained in detail below) instead of convolutional networks.

## IV. THE PROPOSED METHOD

### A. CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional Neural Networks are one of the most important components of deep learning and one of the most widely used methods in the field of computer vision. CNNs consist of three layers.

The convolutional layer: In this layer, a convolutional kernel is used to map the image features. This mapping extracts various features from the image. Weight sharing is applied, reducing the number of network parameters. It evaluates the local dependencies in the image and, ultimately, maintains spatial invariance if an object is present in the image.

The above figure illustrates a convolutional process on an image. As evident, the convolution filter consists of numerical values that are randomly selected. The final value for each pixel is determined by point-wise multiplication with the original image.

The pooling layer: This layer is placed after the convolutional layers. The purpose of using this layer is to reduce the feature mapping. Considering the spatial proximity of the image pixels and disregarding the spatial variations of the objects in the image, the most common pooling

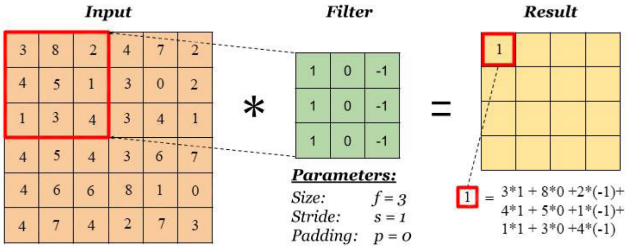

**FIGURE 2.** The process of convolutional filters.

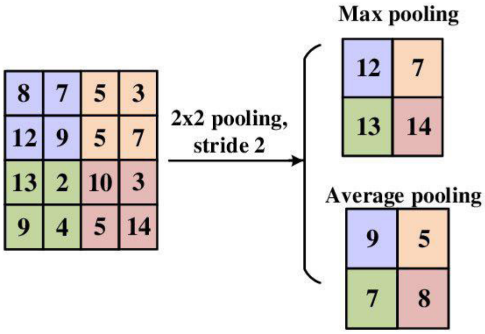

**FIGURE 3.** The processes of the pooling layers.

operations used in this layer are maximum pooling or average pooling [4].

The above figure represents the processes of maximum pooling and average pooling. In the maximum pooling process, a filter with specific dimensions slides over the pixels and selects the maximum number on them. As evident, in the 2*2 window, the maximum value of 112 is selected, whereas in the next window, the value of 37 is chosen. This process continues across all dimensions. Similarly, the average pooling process calculates the average of values within a 2*2 window and replaces the output value with the average value [5].

The fully connected layer: This layer is placed at the end of the network after the last pooling layer. It converts the 2D feature vector from the previous layer into a 1D vector and uses it for training. It functions like a traditional neural network and is used for feature vector classification. Additionally, it has a large number of parameters for training which increases the computational complexity of the network [6].

### B. THE DATABASE

The database used in this research was a combination of 12 concrete crack databases used for image segmentation. This database consisted of 150,000 images with a size of 448*448 pixels. A portion of the analyzed data is shown in the following figure which includes the original image along with its mask. The image mask is displayed in black-and-white format.

Due to the limited number of images available for training the network, data augmentation techniques were employed to enhance its training. Data augmentation is a technique that artificially expands the previous dataset by applying modifications to the existing images without the need to create entirely new images. With this technique, the network can be trained more effectively.

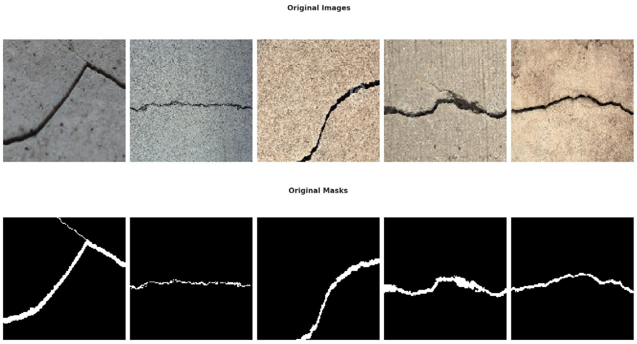

**FIGURE 4.** An example of the data used.

The objective of data augmentation is to generate new images with novel features that provide a better understanding of the network besides the original images. These modifications to the images can include shifts, rotations, zooming, adding noises, etc. Convolutional neural networks have the ability to learn the features of an image regardless of their locations within the image. Therefore, data augmentation techniques can contribute to this learning process and ultimately improve the training of the network.

Typically, data augmentation techniques are applied to the training data. The following table demonstrates the random modifications applied to the images along with their descriptions.

## V. THE IMPLEMENTATION METHOD

The structure of the proposed model consists of four modules that are similar to the U-Net network. The main building blocks of this model are Swin transformers.

The transformer was first introduced in 2021 [32] and its main aim was to address the drawbacks of convolutional layers. The Swin transformer network utilizes the self-attention mechanism and divides the input image into overlapping patches which are then processed by these blocks. Another advantage of this model is its hierarchical design which enables the extraction of both local and global features.

Swin transformer blocks have several advantages over traditional convolutional networks. Their most important feature is their reduced computational complexity despite the increasing size of the input images. Unlike convolutional networks in which the number of trainable parameters linearly increases with the image size, Swin transformer networks apply self-attention in local windows which improves scalability. These networks also outperform standard transformer models which typically use a global attention mechanism. In such cases, the relationships between one extracted token and all tokens in the network increase computational complexity, leading to a slow performance when the network deals with high-resolution images [32].

**TABLE 1. The random changes to the images with their descriptions.**

| Description | Change type |
|---|---|
| Mirror flipping with a 0.7 limitation | Flip |
| Rotating around the vertical or horizontal axes with a 0.7 limitation | Rotate |
| Modifying the image contrast at random | Random contrast |
| Increasing the luminance of the image | Random gamma |
| Changing the image brightness to emulate day and night | Random brightness |
| Image elongation and foreshortening (making waves) | Elastic transform |
| Image reduction, magnification, and shadow creation | Grid distortion |
| Image formation by convex and concave lenses | Optical distortion |

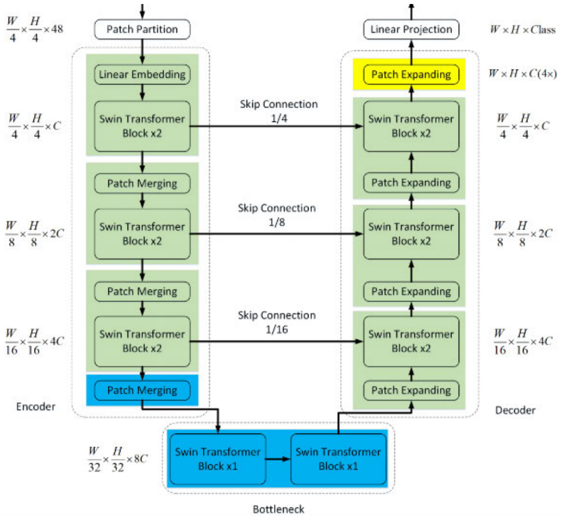
$\frac{W}{4}\times\frac{H}{4}\times48$
Patch Partition
Linear Projection
$W\times H\times Class$
Linear Embedding
Patch Expanding
$W\times H\times C(4\times)$
$\frac{W}{4}\times\frac{H}{4}\times C$
Swin Transformer Block x2
Skip Connection 1/4
Swin Transformer Block x2
$\frac{W}{4}\times\frac{H}{4}\times C$
Patch Merging
Patch Expanding
$\frac{W}{8}\times\frac{H}{8}\times 2C$
Swin Transformer Block x2
Skip Connection 1/8
Swin Transformer Block x2
$\frac{W}{8}\times\frac{H}{8}\times 2C$
Patch Merging
Patch Expanding
$\frac{W}{16}\times\frac{H}{16}\times 4C$
Swin Transformer Block x2
Skip Connection 1/16
Swin Transformer Block x2
$\frac{W}{16}\times\frac{H}{16}\times 4C$
Patch Merging
Patch Expanding
Encoder
Decoder
$\frac{W}{32}\times\frac{H}{32}\times 8C$
Swin Transformer Block x1
Swin Transformer Block x1
Bottleneck

**FIGURE 5. The proposed architecture.**

To address this issue, the Swin transformer network utilizes local window computations. The image is divided into non-overlapping windows with M*M sizes. The computational complexity of a self-attention module in an image with h*w dimensions is calculated as follows [32]:

$$\Omega\ (\mathrm{MSA})=\ 4\mathrm{hw}C^2+2(\mathrm{hw})^2C \tag{3}$$

$$\Omega\ (W-\mathrm{MSA})=\ 4\mathrm{hw}C^2+2M^2hwC \tag{4}$$

In equations 3 and 4, h and w represent the height and width of the image. In equation 4, M denotes the size of the transformer window. In the window-based attention mechanism, there is no direct interaction between the windows which reduces the modeling power. To address this issue, a shifted window partitioning strategy is proposed and used in Swin transformers. This method employs a regular window partitioning strategy that starts from the top left corner and uniformly divides an 8*8 feature map into 4*4 windows. The next module then utilizes the settings of the previous layer and shifts the windows |M/2|*|M/2| to obtain regular windows. This approach allows consecutive blocks to have mutual interactions with the windows [32].

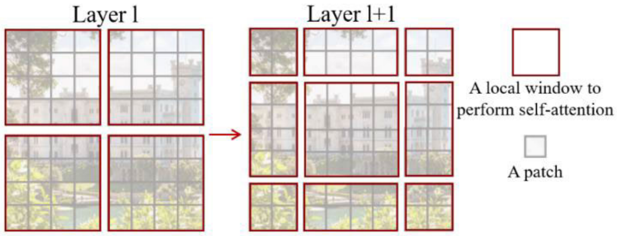

**FIGURE 6. The comparison of the windows in the shifted and non-shifted window approaches. (a) non-shifted 4*4 windows; (b) shifted windows.**

In the above figure, an approach using the shifted window for self-attention calculation in the proposed architecture is shown. (a) (left) illustrates a regular window partitioning method; (b) (right) demonstrates the shifted partitioning [32].

## A. THE PROPOSED ARCHITECTURE

In Figure 5, the proposed architecture for image segmentation is depicted. To process the input images, they are initially divided into non-overlapping windows with 4*4 sizes. The feature dimension of each window is 48 (4*4*3). Additionally, a linear embedding layer is added to transform the feature dimensions to the desired size denoted by C. After passing through various layers, hierarchical features are generated. The next layer is the path aggregation which is responsible for downsampling and increasing the dimensions. In the up path, there is an expansion path layer designed for upsampling. This layer doubles the dimensions of the feature map. This process continues after each layer until reaching the original size of the image. Finally, another linear embedding layer is applied to the image to predict the class of each pixel. The specific tasks of each layer will be further examined below.

### 1) ENCODER

In this block, the inputs, which are passed through an embedding layer with dimensions C and resolution H/4 and W/4, are sent to two consecutive Swin transformer blocks for learning, while the feature dimensions and resolution remain unchanged. Meanwhile, the path aggregation layer reduces the number of tokens (2x downsampling) and doubles the feature dimensions. This operation is performed three times in the encoder [32].

### 2) PATH AGGREGATION LAYER

The input patches are divided into four parts and combined together by the path aggregation layer. Through this process, the feature resolution is halved and the feature dimensions are quadrupled. A linear layer is applied to the combined features to transform the feature dimensions back to twice the initial size [32].

### 3) BOTTLENECK

Since transformers are deep and converge with difficulty, only two consecutive Swin transformer blocks are used to create a bottleneck for deep feature learning. In the bottleneck output, the feature dimensions and resolution remain unchanged [32].

#### 4) DECODER

Corresponding to each encoder, a symmetric decoder based on the Swin transformer is constructed. For this purpose, the path expansion layer is used in the decoder, in contrast to the path aggregation layer used in the encoder, to increase the resolution of the deep extracted features. The path expansion layer plays the role of reshaping the neighboring features into a larger feature map with a higher resolution (2x upsampling) and, at the same time, reduces the feature dimensions to half of the initial size [32].

#### 5) PATH EXPANSION LAYER

For example, in the first path expansion layer, before performing upsampling, a linear layer is applied to the input features ($w/32\times h/32 \times 8c$) to increase the feature dimensions to twice the initial size ($w/32\times h/32 \times 16c$). Then, using the shuffle operation, the feature resolution is doubled and the feature dimensions are reduced to one-fourth of the input size ($w/32\times h/32 \times 8c \rightarrow w/32\times h/32 \times 16c$) [32].

#### 6) THE SKIP CONNECTIONS

Similar to U-Net, skip connections are used in Swin transformers to combine the multi-scale features from the encoder with the upsampled features. The low-level and deep-level features are connected to prevent the loss of spatial information caused by downsampling. After a linear layer, the concatenated features remain the same size as the upsampled features [32].

#### 7) THE ANALYSIS RESULTS

After configuring the network and its related parameters, the next step is to train the network. The system used for training consisted of an NVIDIA graphics card (RTX 3070 Ti), 16 GB of RAM, and a Ryzen 7 CPU. The network was trained for 100 epochs. In the left-side graph, the accuracy of the network during training is displayed. The blue curve represents the accuracy during the training phase, while the orange curve represents the accuracy during the validation phase. The right-side graph shows the error rate of the network. The blue curve represents the error rate during training, while the orange curve represents the error rate during validation.

In Figure 8, the left-side diagram (a) represents the IOU error rate, while the right-side diagram (b) represents the Dice error rate. The y-axis represents the error rate, whereas the x-axis represents the different stages or iterations of the network.

The visual results of the proposed network (after training) are shown in the images below.

As evident from the results and the output images generated by the proposed model, the model exhibits a high level of accuracy. The middle image represents the predicted output mask, while the image on the right shows the ground truth mask based on which the network made its predictions.

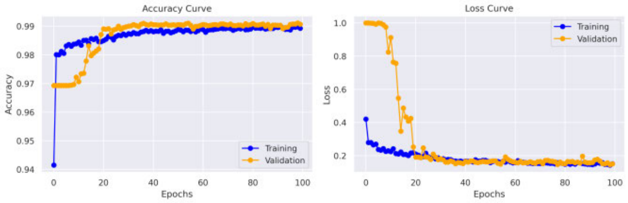

**FIGURE 7. Network error and accuracy: (a) represents the network accuracy; (b) illustrates the network error.**

As observed from the first and second images, the proposed network successfully detected all the cracks within the image including small and subtle ones. The detection process is facilitated by the utilization of sliding windows in each block which locally examine the features of each image. Additionally, training the image features separately increases accuracy when dealing with fine cracks.

### *B. DISCUSSION AND CONCLUSION*

In this paper, the Swin U-Net network was used for crack detection. It was a combination of the U-Net architecture and the Swin transformer network and was capable of detecting cracks in images accurately. The network was trained on a large dataset consisting of images with various types of cracks. The experimental results demonstrated that the Swin U-Net network outperformed other models in crack detection. The accuracy, IOU error, and Dice error in crack detection were 99%, 71%, and 93%, respectively. Additionally, the execution time of the network was significantly shorter than those of the other models, indicating its practical efficiency.

The analysis of the accuracy of the network revealed that the Swin U-Net was capable of detecting both small and large cracks accurately. Moreover, in cases where cracks overlapped with the entire image, the network was able to segment the cracks properly and extract accurate information.

Given that the Swin U-Net is a powerful model for crack detection, it can be used in various applications and cases, including the rehabilitation and maintenance of infrastructures, geohazard detection, and quality control. Furthermore, there is potential for further development and improvement of the network in the future. The proposed model has been compared with previous models below.

## VI. COMPARING THE PROPOSED MODEL WITH PREVIOUS ONES

This section compares the results of the proposed method with those of the other three methods based on the mentioned criteria. The Swin U-Net differs from CNN-based methods in its approach to feature representation and learning. The Swin U-Net capitalizes on the Swin Transformer block as the fundamental unit for feature representation and long-range semantic information interactive learning. This is in contrast to CNN-based methods which rely on convolutional neural networks for feature extraction and learning. The use of the Swin Transformer block by the Swin U-Net allows for a different approach to capturing long-range dependencies

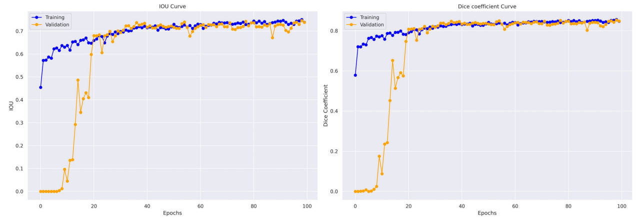

**FIGURE 8.** **The network error rate: (a) displays the IOU error rate; (b) shows the Dice error rate.**

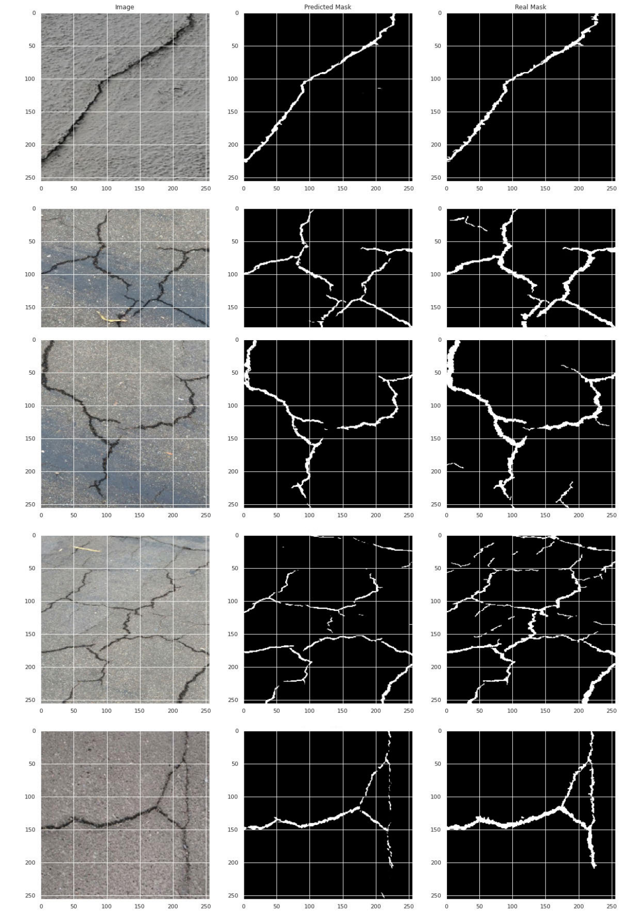

**FIGURE 9.** **The output images of the network: a) the original image (left); b) the predicted mask (middle); c) the ground truth mask (right).**

and semantic information, which can result in different performance and efficiency characteristics compared to traditional CNN-based methods [33]. The Swin U-Net model combines the strengths of the Swin Transformer for global context understanding and the U-Net for local feature extraction, offering a hybrid approach for semantic segmentation tasks like tunnel lining crack detection. On the other hand, CNN-based methods primarily rely on convolutional operations for feature extraction and segmentation, focusing more on local features with limited consideration for long-range dependencies [34].

*4.1.1. Deep Crack:*

The proposed network was a CNN in which the feature maps were extracted in every convolutional layer. A filter was applied to the feature maps to predict the outputs.

The architecture of the proposed method is shown in Figure 10.

*4.1.2. CNN Architecture:*

Similarly, the second method was a CNN operating at the pixel level which decided whether a pixel belonged to the 'crack' category or the 'no-crack' category [30].

The network designed in this study is illustrated in Figure 11.

*4.1.3. The VGG Architecture:*

The third network was a U-Net based on a VGG located in the encoding layer.

The network designed in this study is illustrated in Figure 12. All the methods are compared and shown in Table 3 based on the mentioned metrics.

As observed in Table 3, the proposed method had a higher accuracy and a lower error rate on an identical dataset than the other methods.

As observed in the above table, the proposed method outperformed the other methods and had a higher accuracy and a lower error rate on the same dataset. In a previous study [29], due to increased computational complexity, the training time and inference of the network increased. In contrast, the complexity of the proposed model was significantly reduced by utilizing its self-attention mechanism. In another study [31], the use of a pre-trained model reduced the computational overhead. However, the extracted features in the encoding layer were not suitable for the task of crack detection. Additionally, all three papers ( [29], [30], and [31]) required a large amount of training data due to their usage of convolutional layers.

## VII. CONCLUSION

By using its self-attention mechanism, the proposed method eliminates the need for a large amount of data in crack detection. Moreover, by incorporating the local features of each image during training, it can easily detect small cracks as well. Considering the inherent complexity of the dataset and the extensive image labeling effort required, the proposed model exhibited highly satisfactory results. The accurate detection and localization of concrete cracks provide a great potential for diverse applications in structural health monitoring and concrete infrastructure maintenance.

### A. CONSTRAINTS

- Dataset: Using an appropriate and comprehensive dataset for training and evaluating the model is crucial. Access to properly annotated image datasets can be challenging. In fact, many researchers have collected their own datasets using drones or other means.
- Parameter tuning: The Swin U-Net model, like other deep learning networks, has multiple parameters that need to be properly tuned. Improper parameter settings can have an impact on the performance and results of the model.
- Computational constraints: Due to its utilization of encoders and decoders, the Swin U-Net architecture requires significant computational resources.

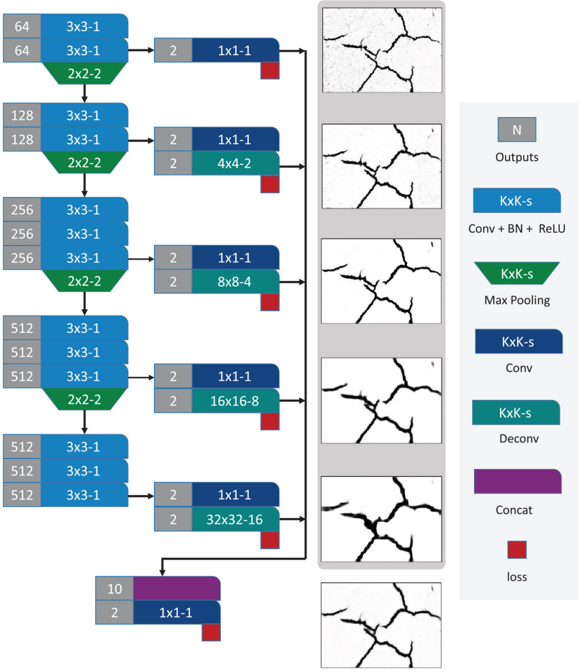

**FIGURE 10.** The architecture of the deep crack.

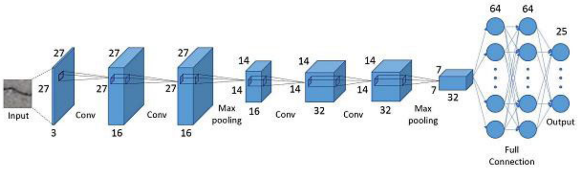

**FIGURE 11.** The architecture of the CNN.

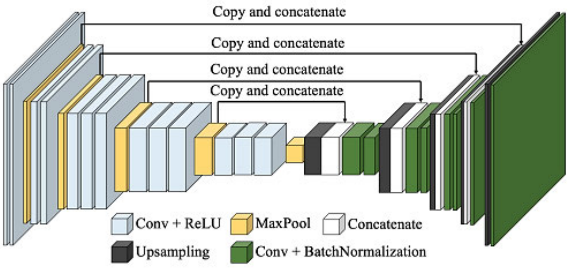

**FIGURE 12.** VGG architecture.

**TABLE 2.** Units for magnetic properties.

| Metrics | DICE | IOU | Accuracy |
|---|---|---|---|
| Deep Crack | -- | 70% | 98.6% |
| CNN | -- | -- | 90% |
| VGG | 90% | 78% | -- |
| Proposed method | 93% | 79% | 99% |

Computational limitations and hardware constraints can hinder the fast and efficient execution of the algorithm.

- Examples of failure cases:

1. Misclassification of ambiguous regions: The SWIN U-Net may misclassify regions in the input image that are ambiguous or contain complex structures, leading to segmentation errors.

2. Segmentation artifacts: In some cases, the SWIN U-Net may produce segmentation masks with artifacts such as disconnected regions, over-segmentation, or under-segmentation.

3. Failure on out-of-distribution data: When presented with data that significantly deviate from the training distribution, the SWIN U-Net may produce inaccurate segmentation results or fail entirely.

4. Inability to handle occlusions: The SWIN U-Net may struggle to segment accurately objects that are partially occluded or overlap with other objects in the scene.

While the SWIN U-Net model has shown a promising performance in various computer vision tasks, it still has some limitations and can encounter failure cases like any other model. Here are some of its limitations:

1. Complexity and computational cost: The SWIN U-Net is a deep neural network with a large number of parameters, making it computationally expensive to train and deploy. This complexity can lead to longer training times and higher resource requirements.
2. Overfitting: Like other deep learning models, the SWIN U-Net is susceptible to overfitting, especially when trained on limited data. Overfitting can result in poor generalization performance whereby the model performs well on the training data but fails to generalize to unseen data.
3. Limited generalization to diverse data: The SWIN U-Net may struggle to generalize to data that significantly differ from the distribution of the training data. For example, if the model is trained on a specific type of medical images, it may not perform well when applied to images from a different imaging modality or medical condition.
4. Sensitivity to hyperparameters: The performance of the SWIN U-Net can be sensitive to hyperparameters such as the learning rate, batch size, and network architecture. Suboptimal hyperparameter choices can lead to a slower convergence or an inferior performance.
5. Handling of small objects or details: The SWIN U-Net may face difficulty in segmenting or detecting small objects or fine details in the input images accurately. This limitation can arise due to the downsampling operations in the U-Net architecture which may cause the loss of spatial information.
6. Limited interpretability: While the SWIN U-Net can produce accurate segmentation results, it may not provide insights into the reasoning behind its predictions. Understanding why the model makes certain decisions can be challenging, especially in critical applications like medical imaging.

### B. SUGGESTIONS

- Carefully curate and annotate a diverse and representative dataset that adequately covers various types of cracks.
- Conduct thorough parameter tuning experiments to find the optimal settings for the Swin U-Net model.
- Utilize powerful hardware resources or consider distributed computing to overcome computational limitations and expedite the training and inference processes.

## VIII. FUTURE WORKS

To improve the algorithm presented in this paper and make advancements in crack detection using the Swin U-Net, one can consider the following:

1) Using a more extensive dataset: Utilizing larger and more diverse datasets for training and evaluation is crucial as it enhances the generalizability of the model. Incorporating larger datasets can improve the performance of the model.
2) Parameter optimization: Conducting more experiments to tune the parameters and determine the optimal combination can significantly improve the performance of the model. Consider selecting an appropriate optimization algorithm to find the best parameters. For example, using the parameter 'search processes' can help discover the optimal parameter values.
3) Evaluating generalizability: In this paper, it was claimed that the Swin U-Net model had a high generalizability. One can evaluate this claim by using the proposed model on different datasets to assess its performance in various scenarios.
4) Damage assessment: In the aftermath of natural disasters or accidents, the SWIN U-Net can aid in the rapid damage assessment of concrete structures. By processing images acquired from UAVs or ground-based sensors, the model can identify areas of structural damage and prioritize response efforts.

5. Environmental monitoring: The SWIN U-Net can be deployed for environmental monitoring applications related to concrete structures such as assessing the impacts of weathering, pollution, or chemical exposure. By analyzing the images collected over time, the model can track changes in the surface conditions of concrete and predict maintenance needs.

By addressing these points, one can enhance the effectiveness and robustness of the Swin U-Net model for crack detection.

## REFERENCES

[1] Y. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, May 2017, doi: 10.1111/mice.12263.

[2] M. R. Saleem, J. W. Park, J. H. Lee, H. J. Jung, and M. Z. Sarwar, "Instant bridge visual inspection using an unmanned aerial vehicle by image capturing and geo-tagging system and deep convolutional neural network," *Struct. Health Monit.*, vol. 20, no. 4, pp. 1760–1777, 2020, doi: 10.1177/1475921720932384.

[3] K. Liu, X. Han, and B. M. Chen, "Deep learning based automatic crack detection and segmentation for unmanned aerial vehicle inspections," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dali, China, Dec. 2019, pp. 381–387, doi: 10.1109/ROBIO49542.2019.8961534.

[4] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, and X. Shen, "Image-based concrete crack detection in tunnels using deep fully convolutional networks," *Construct. Building Mater.*, vol. 234, Feb. 2020, Art. no. 117367, doi: 10.1016/j.conbuildmat.2019.117367.

[5] F. Lin, J. Yang, J. Shu, and R. J. Scherer, "Crack semantic segmentation using the U-Net with full attention strategy," 2021, *arXiv:2104.14586*.

[6] H. Chu, W. Wang, and L. Deng "Tiny-Crack-Net: A multiscale feature fusion network with attention mechanisms for segmentation of tiny cracks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 37, no. 14, pp. 1914–1931, 2022, doi: 10.1111/mice.12881.

[7] 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, pp. 1–10, Dec. 2021, doi: 10.1109/JPHOT.2021.3118924.

[8] F. Çelik and M. König, "A sigmoid-optimized encoder–decoder network for crack segmentation with copy-edit-paste transfer learning," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 37, no. 14, pp. 1875–1890, Nov. 2022, doi: 10.1111/mice.12844.

[9] C. Xiang, W. Wang, L. Deng, P. Shi, and X. Kong, "Crack detection algorithm for concrete structures based on super-resolution reconstruction and segmentation network," *Autom. Construct.*, vol. 140, Aug. 2022, Art. no. 104346, doi: 10.1016/j.autcon.2022.104346.

[10] Q. An, X. Chen, H. Wang, H. Yang, Y. Yang, W. Huang, and L. Wang, "Segmentation of concrete cracks by using fractal dimension and UHK-net," *Fractal Fractional*, vol. 6, no. 2, p. 95, Feb. 2022, doi: 10.3390/fractalfract6020095.

[11] T. Lee, J.-H. Kim, S.-J. Lee, S.-K. Ryu, and B.-C. Joo, "Improvement of concrete crack segmentation performance using stacking ensemble learning," *Appl. Sci.*, vol. 13, no. 4, p. 2367, Feb. 2023, doi: 10.3390/app13042367.

[12] 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," *Construct. Building Mater.*, vol. 329, Apr. 2022, Art. no. 127157, doi: 10.1016/j.conbuildmat.2022.127157.

[13] H. Inam, N. U. Islam, M. U. Akram, and F. Ullah, "Smart and automated infrastructure management: A deep learning approach for crack detection in bridge images," *Sustainability*, vol. 15, no. 3, p. 1866, Jan. 2023, doi: 10.3390/su15031866.

[14] H. Kim, E. Ahn, M. Shin, and S.-H. Sim, "Crack and noncrack classification from concrete surface images using machine learning," *Struct. Health Monitor.*, vol. 18, no. 3, pp. 725–738, May 2019, doi: 10.1177/1475921718768747.

[15] Z. Zhou, L. Yan, J. Zhang, Y. Zheng, C. Gong, H. Yang, and E. Deng, "Automatic segmentation of tunnel lining defects based on multiscale attention and context information enhancement," *Construct. Building Mater.*, vol. 387, Jul. 2023, Art. no. 131621, doi: 10.1016/j.conbuildmat.2023.131621.

[16] H. Karacan and M. Sevri, "A novel data augmentation technique and deep learning model for web application security," *IEEE Access*, vol. 9, pp. 150781–150797, 2021, doi: 10.1109/ACCESS.2021.3125785.

[17] Y. Yang, Z. Niu, L. Su, W. Xu, and Y. Wang, "Multi-scale feature fusion for pavement crack detection based on transformer," *Math. Biosciences Eng.*, vol. 20, no. 8, pp. 14920–14937, 2023, doi: 10.3934/mbe.2023668.

[18] Y. Choi, H. W. Park, Y. Mi, and S. Song, "Crack detection and analysis of concrete structures based on neural network and clustering," *Sensors*, vol. 24, no. 6, p. 1725, Mar. 2024, doi: 10.3390/s24061725.

[19] F.-C. Chen and M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2018.

[20] S. Egodawela, A. Khodadadian Gostar, H. A. D. S. Buddika, A. J. Dammika, N. Harischandra, S. Navaratnam, and M. Mahmoodian, "A deep learning approach for surface crack classification and segmentation in unmanned aerial vehicle assisted infrastructure inspections," *Sensors*, vol. 24, no. 6, p. 1936, Mar. 2024, doi: 10.3390/s24061936.

[21] C. M. Yeum and S. J. Dyke, "Vision-Based automated crack detection for bridge inspection," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 30, no. 10, pp. 759–770, Oct. 2015, doi: 10.1111/mice.12141.

[22] L. Ying and E. Salari, "Beamlet transform-based technique for pavement crack detection and classification," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 25, no. 8, pp. 572–580, Nov. 2010, doi: 10.1111/j.1467-8667.2010.00674.x.

[23] E. Zalama, J. Gómez-García-Bermejo, R. Medina, and J. Llamas, "Road crack detection using visual features extracted by Gabor filters," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 29, no. 5, pp. 342–358, May 2014, doi: 10.1111/mice.12042.

[24] A. Zhang, K. C. P. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J. Q. Li, and C. Chen, "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 10, pp. 805–819, Oct. 2017.

[25] D. Ziou and S. Tabbone, "Edge detection techniques—An overview," *Pattern Recognit. Image Anal., Adv. Math. Theory Appl.*, vol. 8, no. 4, pp. 537–559., 1998.

[26] 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.

[27] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "ResUNet-A: A deep learning framework for semantic segmentation of remotely sensed data," 2019, *arXiv:1904.00592*.

[28] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, Apr. 2019, doi: 10.1016/j.neucom.2019.01.036.

[29] A. Ji, X. Xue, Y. Wang, X. Luo, and W. Xue, "An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement," *Autom. Construct.*, vol. 114, Jun. 2020, Art. no. 103176, doi: 10.1016/j.autcon.2020.103176.

[30] Z. Fan, Y. Wu, J. Lu, and W. Li, "Automatic pavement crack detection based on structured prediction with the convolutional neural network," 2018, *arXiv:1802.02208*.

[31] J. Shi, J. Dang, M. Cui, R. Zuo, K. Shimizu, A. Tsunoda, and Y. Suzuki, "Improvement of damage segmentation based on pixel-level data balance using VGG-UNet," *Appl. Sci.*, vol. 11, no. 2, p. 518, Jan. 2021, doi: 10.3390/app11020518.

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021, *arXiv:2103.14030*.

[33] H. Zhang, A. A. Zhang, Z. Dong, A. He, Y. Liu, Y. Zhan, and K. C. P. Wang, "Robust semantic segmentation for automatic crack detection within pavement images using multi-mixing of global context and local image features," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 13, 2024, doi: 10.1109/TITS.2024.3360263.

[34] Z. Zhou, J. Zhang, and C. Gong, "Hybrid semantic segmentation for tunnel lining cracks based on Swin transformer and convolutional neural network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 38, no. 17, pp. 2491–2510, Nov. 2023, doi: 10.1111/mice.13003.

**ALI SARHADI** is currently a Lecturer with Islamic Azad University and Payame Noor University. Some of his articles have been published in several international journals. 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. He is a member of the Managerial Board of Tehran Construction Engineering Organization, projects, consulting freeway, and railway. He is also active in the stock exchange, investment companies, and financial markets. His books *Construction Contracts and Legal Affairs* and *Green Building Information*. His research interests include construction law, contracts, value engineering, strategic project management, bidding, sustainable buildings, and the portfolios of construction corporations. He is a member of the board of Tehran Construction Engineering Organization (the most significant engineering organization in Iran).

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

**MILAD GHANBARI** received the Ph.D. degree in construction engineering and management from Islamic Azad University, Science and Research Branch. He is currently an Associate Professor with the Faculty of Civil Engineering, East Tehran Branch, Islamic Azad University. He is an experienced Project Control Specialist with a demonstrated history of working in the civil engineering industry. He was skilled in system dynamics, supply chain optimization, contractors, construction, waste management, and concrete. He has a strong program and project management professional with the Ph.D. degree.