Ruan *et al*. *Intell. Robot*. **2025**, *5*(1), 105-18
DOI: 10.20517/ir.2025.07

Intelligence & Robotics

Research Article

Open Access

SKPNet: snake KAN perceive bridge cracks through semantic segmentation

Yudi Ruan[1], Di Wang[1], Yijing Yuan[2], Shixin Jiang[1], Xianyi Yang[3]

[1]School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China.
[2]College of Letters & Science, University of Wisconsin-Madison, Madison, WI 53706, USA.
[3]State Key Laboratory of Mountain Bridge and Tunnel Engineering, Chongqing Jiaotong University, Chongqing 400074, China.

Correspondence to: Prof. Di Wang, School of Information Science and Engineering, Chongqing Jiaotong University, No. 66 Xuefu Avenue, Nan'an District, Chongqing 400074, China. E-mail: diwang@cqjtu.edu.cn; ORCID: 0000-0001-9679-7592

How to cite this article: Ruan, Y,; Wang, D.; Yuan, Y.; Jiang, S.; Yang, X. SKPNet: snake KAN perceive bridge cracks through semantic segmentation. *Intell. Robot.* **2025**, *5*(1), 105-18. http://dx.doi.org/10.20517/ir.2025.07

Received: 1 Nov 2024 **First Decision:** 16 Dec 2024 **Revised:** 3 Jan 2025 **Accepted:** 8 Jan 2025 **Published:** 5 Feb 2025

Academic Editor: Zengshun Chen **Copy Editor:** Pei-Yun Wang **Production Editor:** Pei-Yun Wang

Abstract

As the demands for ensuring bridge safety continue to rise, crack detection technology has become more crucial than ever. In this context, deep learning methods have been widely applied in the field of intelligent crack detection for bridges. However, existing methods are often constrained by complex backgrounds and computational limitations, struggling with issues such as weak crack continuity and insufficient detail representation. Inspired by biological mechanisms, a dynamic snake convolution (DSC) with tubular offsets is incorporated to tackle these challenges effectively. Additionally, a channel-wise self-attention (CWSA) mechanism is introduced to efficiently fuse multi-scale features in U-Net, significantly enhancing the ability of the model to capture fine details. In the classification head, the traditional linear layer is replaced with a Kolmogorov-Arnold network (KAN) structure, which strengthens the robustness and generalization capacity of the model. Experimental results demonstrate that the proposed model improves detection accuracy, achieving a mean intersection over union (mIoU) of 0.877, while maintaining almost the same number of parameters, showcasing exceptional performance and practical applicability. Our project is released at https://github.com/ruanyudi/KanSeg-Bi.

Keywords: Crack detection, dynamic snake convolution, KAN, attention, U-Net, biomimetic

1. INTRODUCTION

As the service life of bridges extends, the materials thereof are progressively subjected to the aging effects of environmental factors. Constant exposure to ultraviolet radiation, weathering, and chemical corrosion[1] leads to alterations in the physical and chemical properties of the bridge materials. This decline in strength and ductility makes the materials more susceptible to the formation of cracks.

Traditional object detection methods primarily rely on techniques such as edge detection, texture and color analysis, and sliding window and template matching. A commonly used edge detection method is the canny edge detector, which identifies edges by calculating image gradients. For instance, Abdel-Qader *et al.* utilized Fourier and Hough transforms in conjunction with the canny operator to extract crack edges[2]. Salman *et al.* proposed an automatic crack classification approach using Gabor filters, while Zhou *et al.* applied frequency domain filtering and contour analysis on 3D laser range data for crack detection[3,4]. Vivekananthan *et al.* employed a combination of grayscale discrimination and the Otsu method to detect cracks in diverse images[5]. Additionally, Zhu *et al.* developed a crack detection framework based on 2D digital image correlation, using a displacement-based method to evaluate fracture performance in concrete structures[6]. Although traditional methods are computationally efficient and perform well in simpler scenarios, they lack robustness and their performance degrades significantly when handling complex backgrounds, varying shapes, or changes in scale and rotation.

In 1998, LeCun *et al.* introduced the modern convolutional neural network (CNN)[7], which has since become a cornerstone of computer vision tasks, excelling in feature extraction. Building on this, Shelhamer *et al.* pioneered semantic segmentation with the fully convolutional network (FCN)[8]. In 2014, Adhikari *et al.* proposed a change detection model[9], providing a digital representation of crack images, which enabled easy comparison of temporal defects. Two years later, Zhang *et al.* developed a deep CNN for road crack detection, effectively addressing challenges in complex backgrounds and low-contrast conditions[10]. However, this method showed limited accuracy in detecting small cracks and struggled with adaptability to varying lighting and weather conditions. Also in 2016, Mokhtari *et al.* evaluated four classification techniques - artificial neural networks (ANN), decision trees, k-nearest neighbors, and adaptive neuro-fuzzy inference systems (ANFIS) - in a computer vision-based pavement crack detection system[11]. In 2019, Xu *et al.* proposed an automatic bridge crack detection method using CNNs and high-resolution imagery, achieving efficient and accurate results[12]. Hoskere *et al.* introduced a deep neural network that identifies material types and structural damages through multi-objective optimization[13]. By 2023, Iraniparast *et al.* advanced the field by leveraging transfer learning and multi-resolution image processing techniques for concrete crack detection and segmentation[14]. Their use of pre-trained deep learning models reduced training time and improved the model's generalization capabilities. Ding *et al.* proposed Sw-YoloX, which leverages advanced training strategies, including simple optimal transport assignment (SimOTA) and multi-model integration, alongside convolutional block attention (CBAM) and atrous spatial pyramid pooling (ASPP) modules, to enhance object detection performance on blurred sea surface images[15]. Xu *et al.* introduced a task-significance-aware meta-learning paradigm for multi-type structural damage segmentation[16]. Ye *et al.* improved the YOLOv7 network with custom modules to enhance crack detection, achieving robust performance on images with noise and effectively identifying cracks of different sizes[17]. Some methods had significantly advanced lightweight network design; for instance, Wu *et al.* proposed a DeepLabV3+ architecture with MobileNetV2 as the backbone, replacing standard convolutions with depthwise separable convolutions to reduce parameters[18].

The aforementioned methods are often limited by complex backgrounds and computational constraints, facing challenges such as weak crack continuity and insufficient detail representation. To address these issues, we propose a novel network (SKPNet) that integrates snake convolution, Kolmogorov-Arnold network (KAN), and pyramid channel-wise self-attention (CWSA). Our approach features a multi-scale framework combined with dynamic snake convolution (DSC) to improve feature extraction. Additionally, we introduce a CWSA

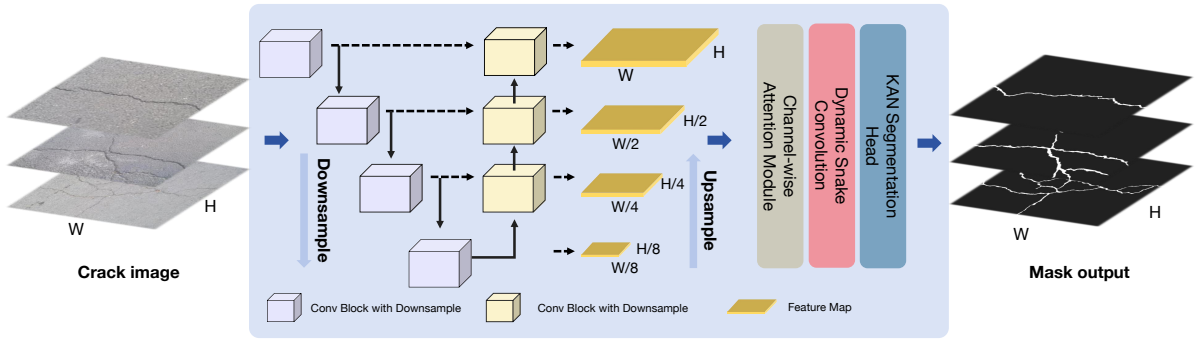

Figure 1. Flowchart of SKPNet.

mechanism to effectively fuse the multi-scale features. Finally, we replace the traditional multi-layer perceptron (MLP) with a KAN to enhance the model's expressive power.

2. METHODS

2.1. Preliminary

The main architecture of our method is illustrated in Figure 1. The input first passes through a U-Net backbone to extract multi-scale features, which are then fused using a CWSA module . Subsequently, information is further extracted via DSC with tubular offsets. Finally, the segmentation output is generated through a classification head composed of KAN layers.

Firstly, The U-Net with residual connections is adopted as the backbone of our model, where three stages of 2× resampling facilitate the extraction of multi-scale information from the input. During the upsampling process, we adapt resampling parameters corresponding to those used in the downsampling stages, ensuring that the scale at each upsampling stage matches its corresponding downsampling stage. In each upsampling operation, residual connections are utilized to fuse information from the corresponding downsampling layers, which helps mitigate the loss of detailed information caused by downsampling. The specific process is given in

$$\begin{cases} x_{i+1} & = Down(Conv(x_i)), \quad for\ i = 1, 2, 3 \\ y_4 & = x_4, \\ y_i & = Up(Conv(y_{i+1})) + Res(x_i), \quad for\ i = 3, 2, 1, \end{cases} \tag{1}$$

where $x_1 \in \mathbb{R}^{3\times H\times W}$ represents the input image to be detected, and $Down(\cdot)$ and $Up(\cdot)$ represent the downsampling and upsampling operations, respectively. $Conv(\cdot)$ and $Res(\cdot)$ denote the convolution operation and residual connection, respectively. Subsequently, we design a CWSA module that can process multi-scale features and capture internal relationships using an attention mechanism. It accepts the results produced by each upsampling operation and utilizes self-attention to extract the correlations between each channel. This operation can integrate large-scale visual detail information with small-scale semantic information. This process can be given in

$$Feat = CWSA(Concat([y_1, y_2, y_3, y_4])), \tag{2}$$

where $CWSA(\cdot)$ represents the operation of CWSA, and $Concat(\cdot)$ represents concatenation operations. Additionally, DSC is integrated to adaptively extract tubular cracks. It is equipped with an offset specially designed for tubular structures, enabling the convolutional kernel to dynamically deform according to the input, which helps preserve the integrity of the information. This will be discussed further in Section "DSC". DSC can be expressed as

$$Feat = DSC(Feat), \tag{3}$$

where $DSC(\cdot)$ represents the operation of the DSC module. Finally, a classification head composed of KAN layers is used to classify each pixel, outputting the final detection results. The final outputs can be given in

$$Mask = KANs(Feat), \tag{4}$$

where $KANs$ represents operations in the classification head.

2.2. CWSA

Given an input feature map $X \in \mathbb{R}^{B\times C\times H\times W}$, where B indicates the batch size, C stands for the number of channels, H represents the height, and W denotes the width; the channel-wise attention is computed using self-attention as follows. First, the input tensor is reshaped into a sequence of channel descriptors by applying global average pooling as

$$X_c = AvgPool(X) \in \mathbb{R}^{B\times C}. \tag{5}$$

Next, the query, key, and value matrices for self-attention are computed as

$$Q = W_q X_c, \quad K = W_k X_c, \quad V = W_v X_c, \tag{6}$$

where $W_q, W_k, W_v \in \mathbb{R}^{C\times C}$ are learned weight matrices. Then, the attention scores are computed by calculating the similarity between the query and key matrices using dot-product attention as

$$A = softmax\left(\frac{QK^T}{\sqrt{C}}\right), \tag{7}$$

where $A \in \mathbb{R}^{B\times C\times C}$ represents the attention map between channels. Finally, the output is computed by multiplying the attention map with the value matrix as

$$X' = AV, \tag{8}$$

where $X' \in \mathbb{R}^{B\times C}$ is the reweighted channel descriptor. The attention weights are then applied back to the original feature map by

$$\tilde{X} = X' \cdot X, \tag{9}$$

where $\tilde{X} \in \mathbb{R}^{B\times C\times H\times W}$ is the final channel-attended feature map. CWSA is incorporated to effectively leverage the multi-scale information output from the U-Net model. This integration allows the network to dynamically reweight channel features, enhancing ability of the model to focus on critical information while suppressing noise. By capturing long-range dependencies and emphasizing relevant features across different scales, the CWSA mechanism significantly improves the accuracy and robustness of crack detection in complex images.

2.3. DSC

In this study, DSC[19] is introduced to improve boundary delineation and feature extraction. Given an input tensor $\mathbf{X} \in \mathbb{R}^{B\times C\times H\times W}$, where B is the batch size, C is the number of channels, and $H\times W$ represents the spatial dimensions. DSC dynamically adjusts the offsets of convolutional kernels based on the geometric properties of the cracks as shown in Figure 2. The left panel illustrates standard convolution, where fixed kernel positions are applied, represented by green dots. The right panel demonstrates DSC, where the white dots represent the original kernel positions, and the green dots indicate the adjusted positions after applying offsets. This tubular offset mechanism allows DSC to better capture the detailed information of tubular structures.

This dynamic adjustment allows the convolution operation to align more effectively with irregular shapes, such as the edges and contours of cracks in concrete surfaces. Unlike standard convolutions, which apply fixed filters across spatial dimensions, DSC modifies the receptive field of kernels by learning deformation offsets during training. This leads to more precise crack boundary detection by capturing subtle variations in crack shapes

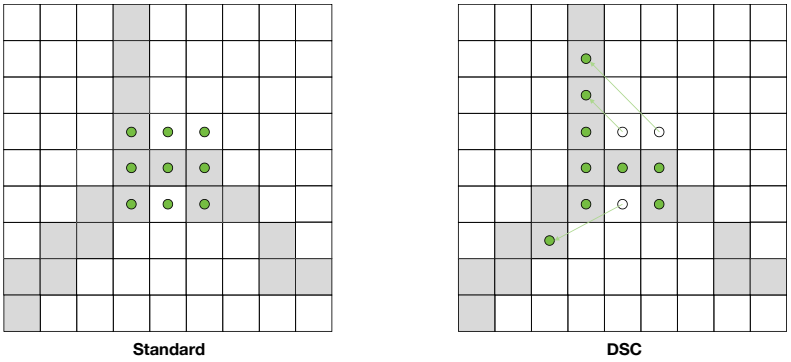

Figure 2. Comparison between standard convolution and DSC. DSC: Dynamic snake convolution.

and orientations. Mathematically, for each convolutional operation at spatial location (i, j), the kernel $\mathbf{K}$ is adjusted with learned offsets $\Delta(i, j)$, such that the convolution result at (i, j) becomes

$$y(i,j) = \sum_{m=-k}^{k} \sum_{n=-k}^{k} \mathbf{K}(m,n) \cdot \mathbf{X}(i+\Delta m, j+\Delta n) \tag{10}$$

where $\Delta m, \Delta n$ represent the dynamically learned offsets that adjust the position of the kernel. By incorporating these dynamic offsets, DSC enhances the ability of the model to detect complex crack structures, ensuring more accurate segmentation and classification results.

2.4. KAN: Kolmogorov-Arnold network

In the final stage of our model, the traditional MLP is replaced with a KAN[20] to leverage its superior function approximation capabilities. The MLP is commonly structured as a series of matrix multiplications and nonlinear activation functions, defined as

$$MLP(\mathbf{Z}) = (W_{K-1} \circ \sigma \circ W_{K-2} \circ \sigma \circ \ldots \circ W_1 \circ \sigma \circ W_0)\mathbf{Z} \tag{11}$$

where $\mathbf{Z}$ is the input, W_k represents the weight matrices, and σ denotes a nonlinear activation function applied between each layer. The symbol $\circ$ represents the composition of matrix multiplication and activation functions. While effective, this architecture may have limitations in capturing highly complex dependencies due to its fixed linear transformations between layers. In contrast, the KAN leverages the Kolmogorov-Arnold representation theorem, which asserts that any multivariate continuous function can be represented as a finite composition of continuous univariate functions. The KAN is formulated as

$$KAN(\mathbf{Z}) = (\Phi_{K-1} \circ \Phi_{K-2} \circ \ldots \circ \Phi_1 \circ \Phi_0)\,\mathbf{Z}, \tag{12}$$

where each Φ_k is a set of univariate continuous functions defined as

$$\Phi = \{\phi_{q,p}\}, \quad p = 1, 2, \ldots, n_{in}, \quad q = 1, 2, \ldots, n_{out}, \tag{13}$$

where $\phi_{q,p}$ represents the learned univariate mappings between input and output dimensions. These univariate mappings are implemented through splined activation functions.

$$\phi(x) = w_b b(x) + w_s \text{spline}(x), \tag{14}$$

where

$$b(x) = \text{silu}(x) = \frac{x}{1+e^{-x}}, \tag{15}$$

and

$$\text{spline}(x) = \sum_i c_i B_i(x), \tag{16}$$

Table 1. Experimental setup configuration

Type	Statement
Operating system	Ubuntu 22.04
RAM	64 G
CPU	Intel i9-13900k
GPU	NVIDIA GeForce RTX 4090
CUDA version	12.4
Pytorch version	2.4.1
Python version	3.9.19

where, w_b, w_s, and c_i are all trainable parameters. During the initialization phase, w_s is set to 1 and spline$(x) \approx 0$, while w_b is initialized according to the Xavier initialization method. This structure enables KAN to capture more complex and nonlinear dependencies between input features compared to the fixed transformations of an MLP. By replacing the MLP with KAN, we introduce a more flexible framework capable of approximating complex functions, which is particularly beneficial for tasks such as crack segmentation. The enhanced ability to capture subtle relationships in the data results in improved segmentation accuracy and robustness, especially in handling irregular and intricate patterns present in crack structures.

2.5. Experimental configuration

2.5.1. Dataset

The dataset employed for this experiment is derived from the SDNET2018 [21] collection. SDNET2018 offers over 56,000 images depicting concrete structures, including bridge decks, walls, and pavements, intended to facilitate the development, validation, and testing of algorithms for detecting concrete cracks. To evaluate the performance of our optimized model, 500 images were selected from the dataset, specifically targeting bridge crack surfaces. These images have a resolution of 512 × 512 pixels and contain three color channels. The pixel-level annotations were carefully generated using the LabelMe tool, with white pixels indicating crack areas and black pixels marking the background. For data augmentation, horizontal flipping and rotations were applied. After flipping, the images were rotated by 90 and 180 degrees, ensuring the corresponding labels underwent the same transformations to preserve alignment. As a result of data augmentation, the dataset size was expanded to 1,600 images. The dataset was then divided into two portions, with 70% used for training and 30% reserved for testing. As shown in Figure 3, this displays some samples from the dataset. The sample images in Figure 3 show the range of crack widths, surface conditions, and other environmental factors, such as color shifts, coarse cracks, and fine cracks. Combined with data augmentation, these images effectively reflect the variety of crack images encountered in real-world environments.

In addition to SDNET2018, we utilized the newly developed CrackVision dataset [22], which combines 12,000 images derived from 13 publicly available crack datasets. This unified dataset addresses the limitations of small-scale datasets and inconsistent annotation standards by employing consistent image processing techniques to produce standardized masks. Furthermore, to mitigate the issue of class imbalance commonly encountered in crack datasets, images with fewer than 5,000 crack pixels were excluded. Data augmentation techniques, including Gaussian noise addition and random rotations, were applied to further diversify the dataset.

2.5.2. Training settings

The network model described in this paper was implemented using PyTorch. Table 1 provides a detailed summary of the experimental setup. During training, the Adam optimizer is employed to fine-tune the weights of network. The batch size was set to 8, and the learning rate was initialized at 1e-4. We recorded the loss changes during training, as shown in Figure 4. After 150 epochs, the model stabilized, and performance on the test set no longer improved. Therefore, we selected 150 epochs as the performance recording point.

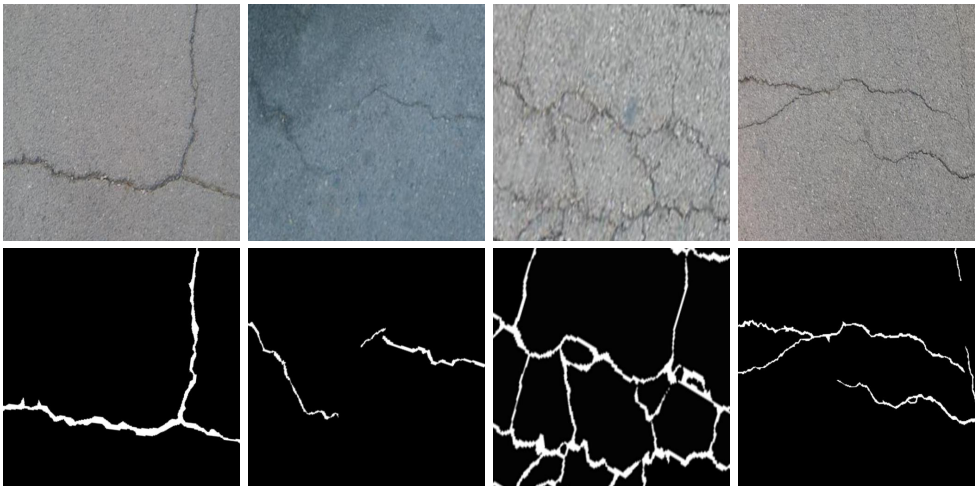

Figure 3. Display of samples from the dataset.

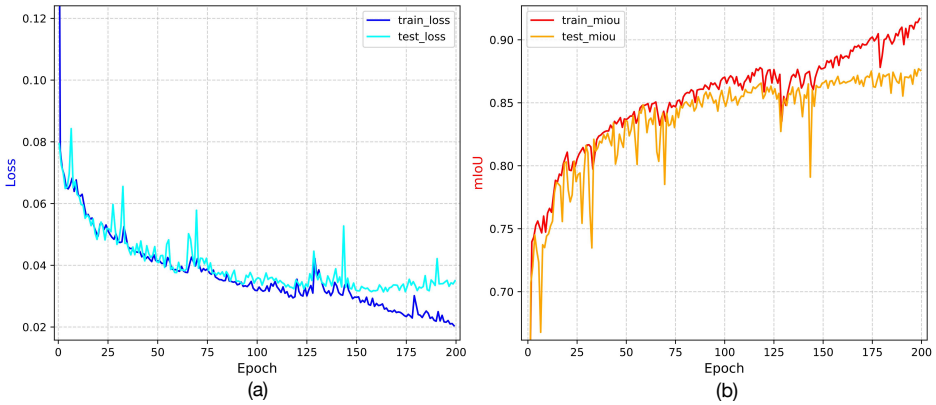

Figure 4. Visualization of training process. (A) Losses during training; (B) mIoU during training. mIoU: Mean intersection over union.

2.5.3. Evaluation metrics

In evaluating our crack segmentation task, several key metrics were applied, including mean intersection over union (mIoU), F1 Score, Precision, and Recall. mIoU is particularly relevant in crack segmentation, as cracks are often continuous and tubular. If the model struggles to capture the continuity of cracks, the mIoU score will be lower, thus providing indirect insight into the model's ability to detect continuous crack structures.

mIoU is the average ratio of the intersection between the predicted and ground truth areas to their union, across all classes, which can be expressed as

$$mIoU = \frac{1}{C}\sum_{i=1}^{C}\frac{|P_i \cap G_i|}{|P_i \cup G_i|} \tag{17}$$

where C is the number of classes, P_i is the predicted area for class i, and G_i is the ground truth area for class i. The numerator represents the intersection of the predicted and ground truth areas, while the denominator represents their union. Precision measures the proportion of correctly predicted positive cases out of all

Table 2. Results of ablation study

	CWSA	DSC	KANs	mIoU
Baseline	×	×	×	0.833
A1	✓	×	×	0.857
A2	×	✓	×	0.869
A3	×	×	✓	0.846
A4	✓	✓	✓	0.877

CWSA: Channel-wise self-attention; DSC: dynamic snake convolution; KANs: Kolmogorov-Arnold networks; mIoU: mean intersection over union.

predicted positives, as given by

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

where TP represents true positives, indicating the number of correctly predicted positive pixels, and FP represents false positives, indicating the number of incorrectly predicted positive pixels. Recall, also known as Sensitivity, measures the proportion of correctly predicted positive cases out of all actual positives, which is defined by

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

where FN represents false negatives, indicating the number of actual positive pixels incorrectly predicted as negative. $F1\ Score$ is the harmonic mean of Precision and Recall, as given in

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{20}$$

3. RESULTS

3.1. Ablation study

The effectiveness of different modules in our proposed model was investigated through ablative experiments. The steps were as follows: (1) Experimenting with the model using only the CWSA mechanism, denoted as A1; (2) Incorporating the DSC with tubular offsets, denoted as A2; (3) Adding the KAN structure in the classification head, denoted as A3; (4) The final configuration includes all three components: CWSA, DSC, and KANs, denoted as A4. The results of the ablative experiment evaluation metrics are presented in Table 2. It provides a detailed comparison of performance of the model with different combinations of the proposed modules. The inclusion of the CWSA mechanism alone (A1) resulted in an mIoU of 0.857. The CWSA module, integrated with the U-Net architecture, helps to fuse multi-scale information, enabling the model to capture both fine and coarse details of the cracks more effectively. Adding the DSC module (A2) improved the mIoU to 0.869. DSC is particularly effective for capturing tubular crack structures, as it adapts well to the deformable, continuous nature of cracks, enhancing the model's ability to detect crack continuity and handle variations in crack shapes. Incorporating the KAN structure (A3) yielded an mIoU of 0.846. KAN boosts the model's expressive power, allowing it to better handle complex crack patterns and improve generalization, particularly in challenging conditions. The final configuration, which includes all three components (A4), achieved the highest mIoU of 0.877. Compared to the baseline value of 0.833, the addition of each component has progressively improved the model's performance, demonstrating the complementary strengths of the CWSA, DSC, and KAN modules in enhancing crack detection accuracy.

Additionally, Figure 5 illustrates the visual results of our ablation study, specifically comparing the performance with and without the DSC module. The top row shows the crack detection results without the DSC module, where the cracks appear less continuous and more fragmented. In contrast, the bottom row demonstrates the

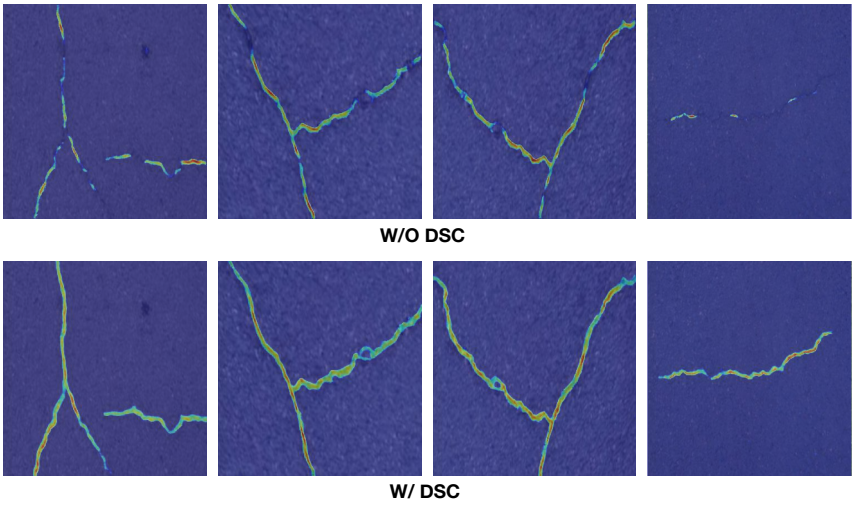

Figure 5. Comparison of ablation experiments for the DSC module. DSC: Dynamic snake convolution.

results with the DSC module, where the cracks are detected with greater continuity and accuracy. These visual results corroborate the quantitative improvements shown in Table 2, highlighting the significant impact of the DSC module on enhancing crack detection performance.

3.2. Comparative performance analysis of different models

In the field of crack detection, we conducted a comparative analysis of our proposed method against several classical approaches. The models compared include: (1) U-Net[23], a symmetrical network that employs stacking operations; (2) FCN[24], a semantic segmentation network that can handle input images of any size; (3) PSPNet[25], an image segmentation network based on spatial pyramid pooling and spatial pyramid attention mechanisms; (4) AFSM-Net[26], which synergistically integrates Transformer and attention mechanism modules, building upon the DeepLabV3+ network model; and (5) Adaptive[27], a traditional algorithm for crack detection in digital images. During the training of these networks, the fluctuations of mIoU for each network are documented. As depicted in Figure 6, our proposed SKPNet model demonstrates a distinct advantage over U-Net, FCN, PSPNet, and the traditional adaptive threshold method. Our model exhibited superior crack segmentation accuracy and excellent continuity in segmenting consecutive lines compared to the other methods. U-Net, while effective, lacks precision in segmenting fine details. FCN can approximate the contours of cracks but suffers from poor segmentation continuity. PSPNet, despite extracting crack information from multiple scales, is affected by background noise, leading to compromised segmentation accuracy. The traditional adaptive threshold method is the least effective, as it struggles to capture the semantic information of images and faces challenges in integrating higher-level semantic understanding.

Table 3 provides a detailed comparison of experimental metrics for different methods, including mIoU, F1-score, Precision, Recall, and the number of parameters. Each result in the table represents the average performance obtained from multiple experiments. Among the deep learning network methods, FCN exhibits the poorest performance with an mIoU of 0.507. There is a significant performance gap between traditional methods and deep learning approaches. From Table 3, it can be observed that our modified SKPNet algorithm outperforms other compared algorithms in terms of accuracy-related metrics, achieving an mIoU of 0.877, F1-score of 0.865, Precision of 0.843, and Recall of 0.889, despite having a moderate parameter count of 31.15M. Furthermore, SKPNet demonstrates efficient inference performance, with an average processing time of 4.2 ms for a 512×512 image on a GPU. Moreover, SKPNet achieves 52.424 GFLOPS, representing a modest 1.28% increase compared to U-Net (51.762 GFLOPS), while demonstrating a substantial improvement in segmentation performance, with an mIoU of 0.877 compared to 0.832. This indicates that the improved SKPNet

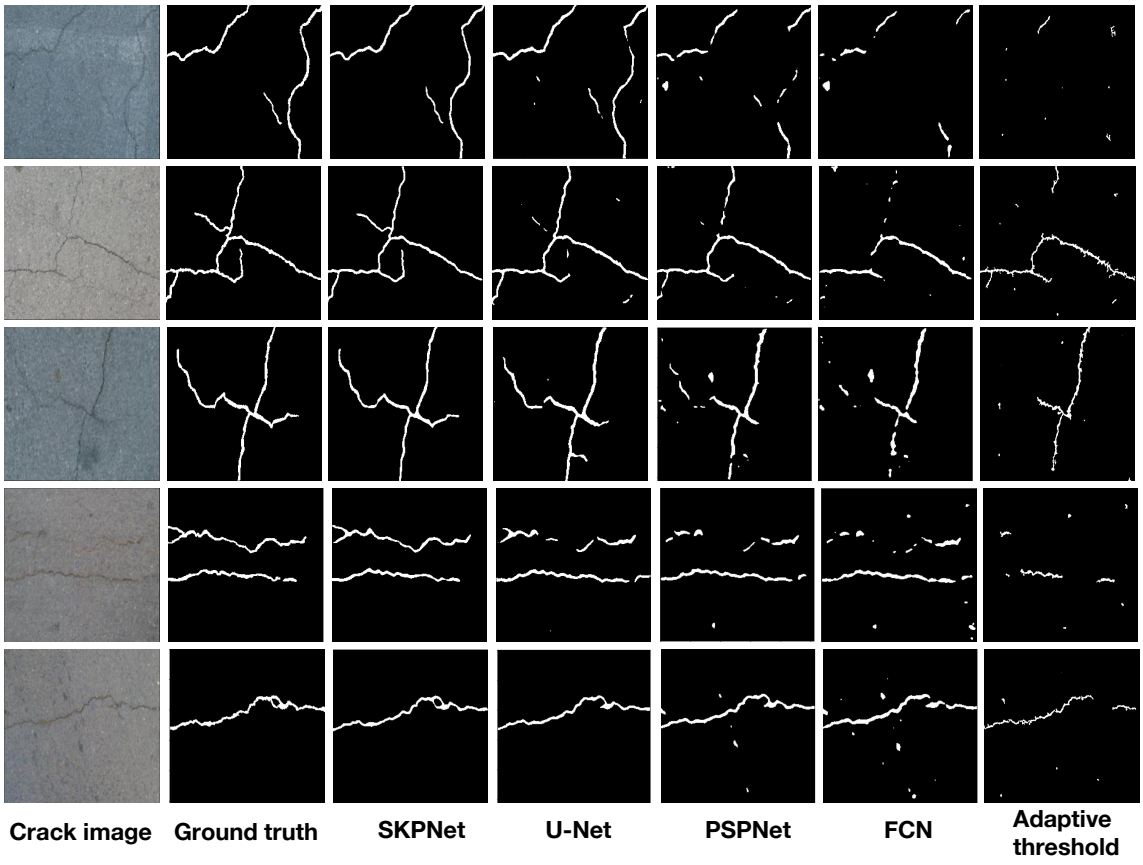

Figure 6. Results of different methods in the test dataset.

Table 3. Comparison of experimental metrics for different methods on SDNET2018

Methods	mIoU	F1-score	Precision	Recall	Parameters
SKPNet	0.877	0.865	0.843	0.889	31.15M
AFSM-Net	0.857	0.843	0.824	0.874	30.23M
U-Net	0.832	0.81	0.789	0.833	24.89M
FCN	0.507	0.749	0.771	0.727	134.26M
PSPNet	0.708	0.615	0.762	0.516	2.38M
Adaptive threshold	0.187	0.326	0.657	0.216	-

The underline represents the highest metric among all meth-ods. mIoU: Mean intersection over union; AFSM: Atrous fu-sion model; FCN: fully convolutional network; PSPNet: pyra-mid scene parsing network.

algorithm enhances crack detection accuracy within an acceptable parameter range, enabling the network to better fulfill the task of bridge crack detection. The results of experiments on the CrackVision test dataset are presented in Table 4. Among the compared methods, SKPNet demonstrates superior performance, achieving the highest IoU (0.660) and F1-score (0.774). This indicates that SKPNet consistently outperforms baseline methods, including FCN, U-Net, DeepCrack[28], SegFormer[29], HrSegNet[30], Hybrid-Segmentor[22].

Figure 7A illustrates the performance comparison of various segmentation models, including SKPNet, FCN, holistically-nested edge detection (HED)[31], CCS-Net[32], and U-Net, across multiple epochs. SKPNet, the proposed method, consistently demonstrates superior performance throughout training. At early epochs, such

Table 4. Comparison of experimental metrics for different methods on CrackVision

Metrics	FCN	U-Net	DeepCrack	SegFormer	HrSegNet	Hybrid-Segmentor	SKPNet
IoU	0.598	0.603	0.592	0.580	0.612	0.630	0.660
F1-score	0.746	0.750	0.741	0.730	0.757	0.770	0.774

The underline represents the highest metric among all methods. FCN: Fully con-volutional network.

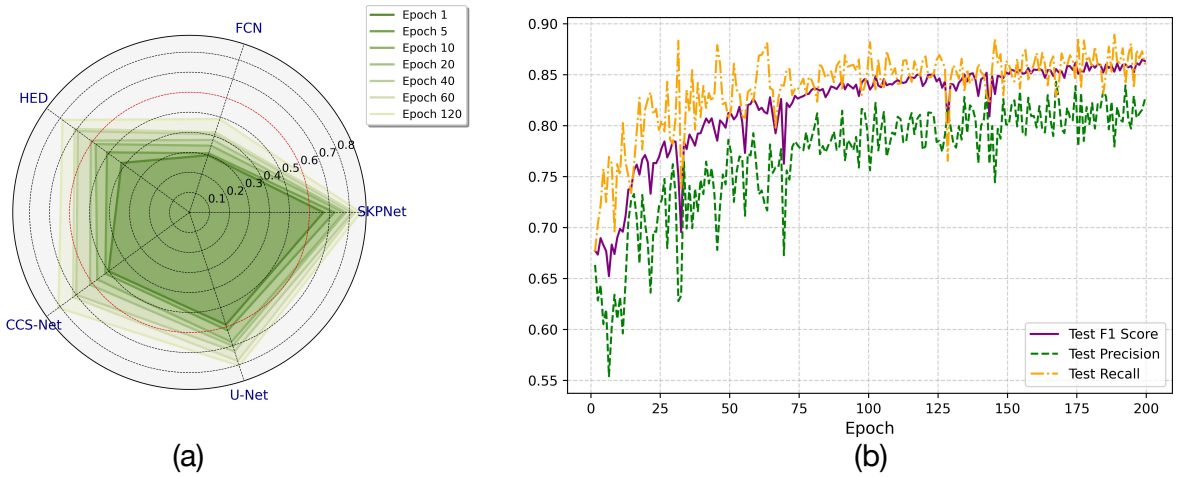

Figure 7. Model performance on the test dataset at different epochs. (A) mIoU performance of different methods at different epochs; (B) Performance of SKPNet on the test dataset at different epochs. mIoU: Mean intersection over union.

as epoch 20, SKPNet achieves an mIoU of 0.780, significantly outperforming the baseline models. As the training progresses, SKPNet further improves, reaching an mIoU of 0.856 at epoch 120, the highest among all compared methods. While U-Net and CCS-Net show competitive results, with mIoUs of 0.805 and 0.810 respectively at epoch 120, they still lag behind SKPNet. On the other hand, HED and FCN, despite showing gradual improvements, exhibit consistently lower performance. These results clearly demonstrate the effectiveness and robustness of SKPNet in segmentation tasks, particularly in capturing detailed structural features. Figure 7B illustrates the performance of SKPNet on the test set in terms of F1-score, Precision, and Recall over 200 epochs. The results demonstrate a consistent improvement during the initial training phase, with all three metrics gradually stabilizing as training progresses. Notably, the F1 Score exhibits steady growth, reaching a plateau after approximately 150 epochs, which indicates a balance between Precision and Recall. This suggests that SKPNet maintains robust generalization capabilities on unseen data while achieving high segmentation accuracy.

Additionally, we evaluated the performance of our model (SKPNet) and the baseline model (U-Net) on more challenging test cases[33], as shown in Figure 8. Panels (a) and (b) depict scenarios with complex textured backgrounds and strong background interference, while (c), (d), and (e) illustrate examples involving fine cracks and severe illumination or shadow interference. From the output results of U-Net, it is evident that under strong background interference, its detection results are prone to noise, leading to false positives. For instance, in (a), U-Net misclassifies the shadowed region at the top as part of the crack, and in (d), it fails to detect cracks located in shadowed areas. In contrast, our model, SKPNet, demonstrates a higher accuracy in extracting crack regions and performs more robustly under challenging conditions with background interference and shadow effects. These experimental results further validate the robustness and superiority of SKPNet in handling complex environments.

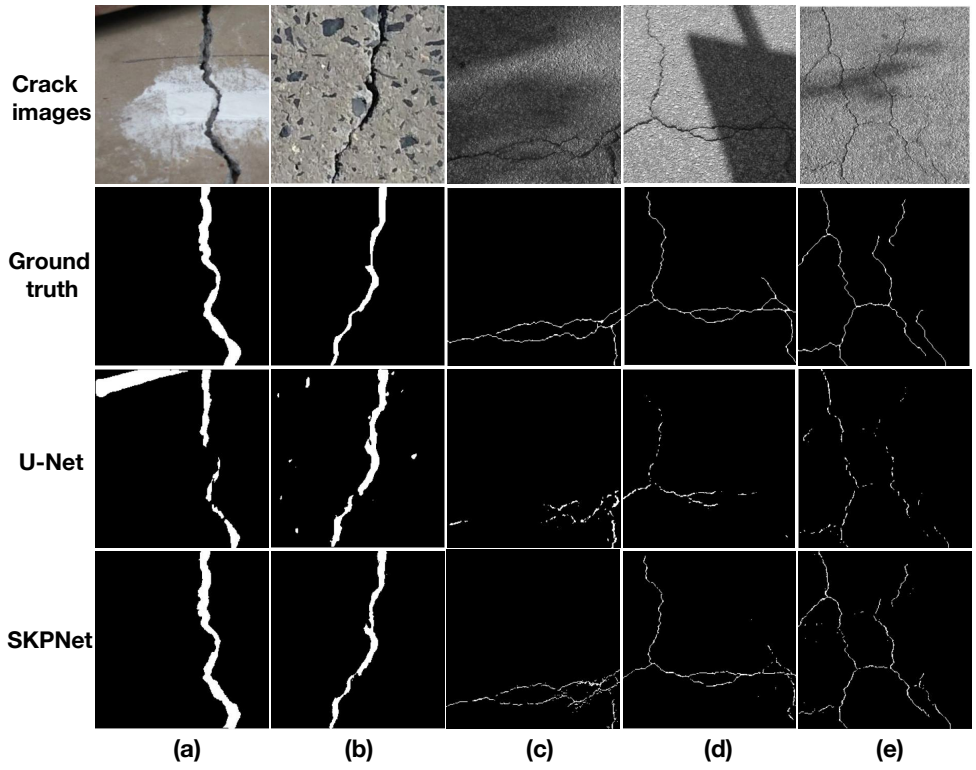

Figure 8. Challenges in handling adverse cases for crack segmentation.

4. CONCLUSIONS

The proposed method effectively tackles the challenges of crack detection in complex backgrounds by integrating three innovative modules. DSC with tubular offsets enhances ability of the model to capture continuous crack structures, while the CWSA mechanism improves multi-scale feature fusion, enabling better detection of fine cracks. Additionally, the KAN layer in the classification head further boosts robustness and generalization of the model. Together, these components contribute to an mIoU of 0.877, demonstrating exceptional performance while maintaining a fast inference speed and nearly identical parameter count compared to baseline models. Future work will focus on reducing inference time and testing on more diverse datasets to validate adaptability to various bridge types and damage forms. However, a key limitation of the current method lies in its sensitivity to domain differences. For instance, the model's performance decreases when testing on images with backgrounds or crack types significantly different from those in the training data, such as cracks on brick surfaces instead of concrete. Addressing this issue will require incorporating domain adaptation techniques and expanding the diversity of training data.

DECLARATIONS

Authors' contributions

Methodology and writing - original draft preparation: Ruan, Y.
Methodology and reviewed the manuscript: Wang, D.
Data processing and figure plotting: Yuan, Y.
Commentary and critical review: Yang, X.; Jiang, S.

Availability of data and materials

The datasets supporting the findings of this study are submitted as Supplementary Materials along with the manuscript.

Financial support and sponsorship

This research was funded by the National Natural Science Foundation of China (Grant No. 62103068, Grant No. 52478141, Grant No. 62003063, Grant No. 62205039), Science and Technology Research Program of Chongqing Municipal Education Commission of China (Grant No. KJQN202100745, Grant No. KJZD-K202400709), and the Natural Science Foundation of Chongqing, China (Grant No. CSTB2022NSCQ-MSX1599).

Conflicts of interest

Wang, D. is Junior Editorial Board Member and Yang, X. is Editor in Chief of the journal *Intelligence & Robotics*. They were not involved in any steps of editorial processing, notably including reviewer selection, manuscript handling, or decision-making. The other authors declare that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Vu, K. A. T.; Stewart, M. G. Structural reliability of concrete bridges including improved chloride-induced corrosion models. *Struct. Saf.* **2000**, *22*, 313–33. DOI
2. Abdel-Qader, I.; Pashaie-Rad, S.; Abudayyeh, O.; Yehia, S. PCA-based algorithm for unsupervised bridge crack detection. *Adv. Eng. Softw.* **2006**, *37*, 771–78. DOI
3. Salman, M.; Mathavan, S.; Kamal, K.; Rahman, M. Pavement crack detection using the Gabor filter. In *16th international IEEE conference on intelligent transportation systems (ITSC 2013)*, The Hague, Netherlands, Oct 06-09, 2013; IEEE, 2013; pp 2039–44. DOI
4. Zhou, S.; Song, W. Robust image-based surface crack detection using range data. *J. Comput. Civ. Eng.* **2020**, *34*, 0000873. DOI
5. Vivekananthan, V.; Vignesh, R.; Vasanthaseelan, S.; Joel, E.; Kumar, K. S. Concrete bridge crack detection by image processing technique by using the improved OTSU method. *Mater. Today. Proc.* **2023**, *74*, 1002–7. DOI
6. Zhu, Z.; Al-Qadi, I. L. Crack detection of asphalt concrete using combined fracture mechanics and digital image correlation. *J. Transp. Eng. Part. B. Pavements.* **2023**, *149*, 04023012. DOI
7. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE.* **1998**, *86*, 2278–324. DOI
8. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE. Trans. Pattern. Anal. Mach. Intell.* **2017**, *39*, 640–51. DOI
9. Adhikari, R. S.; Moselhi, O.; Bagchi, A. Image-based retrieval of concrete crack properties for bridge inspection. *Autom. Constr.* **2014**, *39*, 180–94. DOI
10. Zhang, L.; Yang, F.; Daniel Zhang, Y.; Zhu, Y. J. Road crack detection using deep convolutional neural network. In *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, USA, Sep 25-28, 2016; IEEE, 2016; pp 3708-12. DOI
11. Mokhtari, S.; Wu, L.; Yun, H. B. Comparison of supervised classification techniques for vision-based pavement crack detection. *Transp. Res. Rec.* **2016**, *2595*, 119–27. DOI
12. Xu, H.; Su, X.; Wang, Y.; Cai, H.; Cui, K.; Chen, X. Automatic bridge crack detection using a convolutional neural network. *Appl. Sci.* **2019**, *9*, 2867. DOI
13. Hoskere, V.; Narazaki, Y.; Hoang, T. A.; Spencer, B. F. Jr. MaDnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure. *J. Civil. Struct. Health. Monit.* **2020**, *10*, 757–73. DOI
14. Iraniparast, M.; Ranjbar, S.; Rahai, M.; Moghadas Nejad, F. Surface concrete cracks detection and segmentation using transfer learning and multi-resolution image processing. *Structures* **2023**, *54*, 386–98. DOI
15. Ding, J.; Li, W.; Pei, L.; Yang, M.; Ye, C.; Yuan, B. Sw-YoloX: an anchor-free detector based transformer for sea surface object detection. *Expert. Syst. Appl.* **2023**, *217*, 119560. DOI

16. Xu, Y.; Fan, Y.; Bao, Y.; Li, H. Task-aware meta-learning paradigm for universal structural damage segmentation using limited images. *Eng. Struct.* **2023**, *284*, 115917. DOI
17. Ye, G.; Qu, J.; Tao, J.; Dai, W.; Mao, Y.; Jin, Q. Autonomous surface crack identification of concrete structures based on the YOLOv7 algorithm. *J. Build. Eng.* **2023**, *73*, 106688. DOI
18. Wu, Z.; Tang, Y.; Hong, B.; Liang, B.; Liu, Y. Enhanced precision in dam crack width measurement: leveraging advanced lightweight network identification for pixel-level accuracy. *Int. J. Intell. Syst.* **2023**, *2023*, 9940881. DOI
19. Qi, Y.; He, Y.; Qi, X.; Zhang, Y.; Yang, G. Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. *arXiv* **2023**, arXiv:2307.08388. Available online: https://doi.org/10.48550/arXiv.2307.08388 (accessed 16 Jan 2025).
20. Liu, Z.; Wang, Y.; Vaidya, S.; et al. KAN: Kolmogorov-Arnold networks. *arXiv* **2024**, arXiv:2404.19756. Available online: https://doi.org/10.48550/arXiv.2404.19756 (accessed 16 Jan 2025).
21. Dorafshan, S.; Thomas, R. J.; Maguire, M. SDNET2018: an annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data. Brief.* **2018**, *21*, 1664–8. DOI
22. Goo, J. M.; Milidonis, X.; Artusi, A.; Boehm, J.; Ciliberto, C. Hybrid-segmentor: a hybrid approach to automated fine-grained crack segmentation in civil infrastructure. *arXiv* **2024**, arXiv:2409.02866. Available online: https://doi.org/10.48550/arXiv.2409.02866 (accessed 16 Jan 2025).
23. Liu, F.; Wang, L. UNet-based model for crack detection integrating visual explanations. *Constr. Build. Mater.* **2022**, *322*, 126265. DOI
24. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *arXiv* **2014**, arXiv:1411.4038. Available online: https://doi.org/10.48550/arXiv.1411.4038 (accessed 16 Jan 2025).
25. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. *arXiv* **2016**, arXiv:1612.01105. Available online: https://doi.org/10.48550/arXiv.1612.01105 (accessed 16 Jan 2025).
26. Song, F.; Wang, D.; Dai, L.; Yang, X. Concrete bridge crack semantic segmentation method based on improved DeepLabV3+. In *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS)*, Kaifeng, China, May 17-19, 2024; IEEE, 2024; pp 1293-8. DOI
27. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62-6. DOI
28. Zou, Q.; Zhang, Z.; Li, Q.; Qi, X.; Wang, Q.; Wang, S. DeepCrack: learning hierarchical convolutional features for crack detection. *IEEE Trans. Image. Proc.* **2019**, *28*, 1498–512. DOI
29. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; Luo, P. SegFormer: simple and efficient design for semantic segmentation with transformers. *arXiv* **2021**, arXiv:2105.15203. Available online: https://doi.org/10.48550/arXiv.2105.15203 (accessed 16 Jan 2025).
30. Li, Y.; Ma, R.; Liu, H.; Cheng, G. Real-time high-resolution neural network with semantic guidance for crack segmentation. *Autom, Const.* **2023**, *156*, 105112. DOI
31. Xie, S.; Tu, Z. Holistically-nested edge detection. *arXiv* **2015**, arXiv:1504.06375. Available online: https://doi.org/10.48550/arXiv.1504.06375 (accessed 16 Jan 2025).
32. Yang, J.; Li, H.; Zou, J.; Jiang, S.; Li, R.; Liu, X. Concrete crack segmentation based on UAV-enabled edge computing. *Neurocomputing* **2022**, *485*, 233–41. DOI
33. Goo, J. M.; Milidonis, X.; Artusi, A.; Boehm, J.; Ciliberto, C. Hybrid-segmentor: a hybrid approach to automated fine-grained crack segmentation in civil infrastructure. *arXiv* **2024**, arXiv:2409.02866. Available online: https://doi.org/10.48550/arXiv.2409.02866 (accessed 16 Jan 2025).