DSCformer: A Dual-Branch Network Integrating Enhanced Dynamic Snake Convolution and SegFormer for Crack Segmentation

Kaiwei Yu[1], I-Ming Chen[2] and Jing Wu[1]

[1]Department of Mechanical and Energy Engineering, Southern University of Science and Technology, China
[2]School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore

12332394@mail.sustech.edu.cn, michen@ntu.edu.sg, wujing099@gmail.com

Abstract -

In construction quality monitoring, accurately detecting and segmenting cracks in concrete structures is paramount for safety and maintenance. Current convolutional neural networks (CNNs) have demonstrated strong performance in crack segmentation tasks, yet they often struggle with complex backgrounds and fail to fully capture fine-grained tubular structures. In contrast, Transformers excel at capturing global context, but lack precision in detailed feature extraction. We introduce DSCformer, a novel hybrid model that integrates an enhanced Dynamic Snake Convolution (DSConv) with a Transformer architecture for crack segmentation to address these challenges. Our key contributions include the enhanced DSConv through a pyramid kernel for adaptive offset computation and a simultaneous bidirectional learnable offset iteration, significantly improving the model's performance to capture intricate crack patterns. Additionally, we propose a Weighted Channel Attention Module (WCAM), which refines channel attention, allowing for more precise and adaptive feature attention. We evaluate DSCformer on the Crack3238 and FIND datasets, achieving IoUs of 59.22% and 87.24%, respectively. The experimental results suggest that our DSCformer outperforms state-of-the-art methods across different datasets.

Keywords -

Dynamic Snake Convolution; Crack segmentation;

1 Introduction

Monitoring the quality of concrete structures is crucial in industrial settings, as aging concrete buildings, such as pavements and bridges, often develop cracks that pose significant safety risks [1]. To ensure structural health, regular crack inspections are indispensable. While traditional inspection methods have been widely used, they often fall short in accuracy and efficiency. In contrast, deep learning models have emerged as a promising solution, demonstrating superior performance in crack segmentation tasks due to their accuracy and robustness.

While Convolutional Neural Networks (CNNs) excel at capturing local details due to their strong translational invariance and edge-detection capabilities, their limited receptive field can hinder performance in complex backgrounds [2, 3]. Conversely, Transformers leverage embeddings and self-attention mechanisms to model global dependencies effectively, which is advantageous for addressing complex backgrounds, but they often struggle to capture fine details [4]. Combining these two methods offers the potential to leverage both local detail extraction and global context modeling, addressing the respective limitations of each approach.

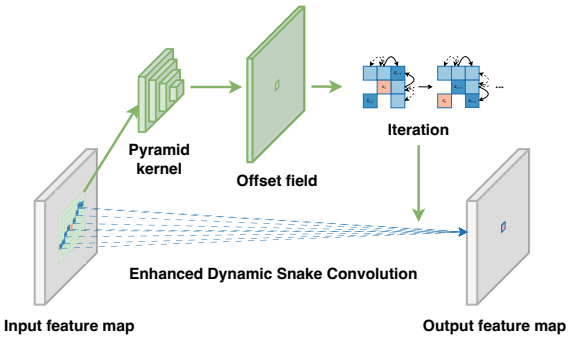

Figure 1. Overview of Enhanced Dynamic Snake Convolution.

The hybrid approach of combining CNNs and Transformers shows promise by leveraging both local detail extraction and global context modeling [5, 6, 7]. However, these methods often fail to account for the unique structure of crack segmentation, leading to suboptimal fusion. Additionally, combining CNNs and Transformers can introduce training fluctuations, making it challenging to fine-tune the model for both local and global features. In contrast, Dynamic Snake Convolution (DSConv) [8] is well-suited for crack segmentation due to its ability to focus on fine, curved local features, improving geometric perception. Despite its strengths, DSConv has limitations, such as fixed kernel sizes and limited adaptability in complex structures, suggesting room for improvement.

In this paper, we introduce DSCformer, a novel crack segmentation model that combines proposed enhanced DSConv with a Transformer-based architecture. Our approach introduces two key improvements to the original DSConv: a) a pyramid kernel for offset computation, consisting of convolutional kernels of varying sizes, and b) a method for learnable offset iteration that simultaneously operates in vertical and horizontal direction. Figure 1 shows the process for enhanced DSConv. These enhancements allow the model to capture fine crack structures more accurately. Additionally, we propose the Weighted Channel Attention Module (WCAM), which refines channel attention by utilizing separate Multi-Layer Perceptrons for average and max pooling. This leads to more adaptive and precise channel attention. The spatial attention module remains unaltered, contributing to spatial focus. By integrating these modifications into the DSConv encoder branch, which focuses on capturing fine local crack details, and combining it with a pre-trained SegFormer[9] encoder that specializes in capturing long-range dependencies, we fuse the extracted features and progressively upsample them to generate the final crack segmentation predictions. The principal contributions of this study are summarized as follows:

- We propose an enhanced DSConv with a pyramid kernel and bidirectional offset iteration, improving crack structure capture.
- We introduce WCAM, refining channel attention for greater adaptability and precision.
- We propose DSCformer, a hybrid model combining DSConv and SegFormer, outperforming state-of-the-art crack segmentation methods across two public datasets.

2 Related Works

2.1 Convolutional Networks for Crack Segmentation

Early works utilized CNN architectures like Fully Convolutional Networks (FCNs) [10] and U-Net [11] for segmentation. DeepCrack [2] follows an encoder-decoder structure. Among methods that alter convolution, Dilated Convolutions [12] expand the receptive field without sacrificing resolution, while Deformable Convolutions [13] improve feature adaptability by introducing learnable offsets to the kernels, enhancing object perception. Cracks, like blood vessels and roads, exhibit fine tubular structures, and enhancing their perception can significantly boost segmentation accuracy. Dynamic Snake Convolution (DSConv) [8] enhances geometric sensing for thin, curved structures by restricting deformable convolution sampling points, yielding improved performance. However, DSConv has limitations, such as a fixed offset kernel size and a one-directional learnable offset iteration, which may hinder adaptive topology perception.

2.2 Hybrid CNN-Transformer Models

Vision Transformers (ViT) [14] capture global image context through patch and position embeddings along with self-attention mechanisms. Swin Transformer [15] enhances computational efficiency with hierarchical shifted windows. Pyramid Vision Transformers (PVT) [16] combine multi-scale feature extraction with Transformer benefits, and SegFormer [9] introduces a hierarchical encoder without position embedding. However, pure Transformers struggle with fine detail capture.

Recent efforts combine CNNs and Transformers for crack segmentation. Crackformer [6] adds a convolution layer atop a Transformer encoder, while UTNet [5] and UCTransNet [17] integrate self-attention into U-Net structures. Crackmer [7] and DTrC-Net [18] employ parallel CNN-Transformer encoders. Despite these advances, more sophisticated hybrid models are needed for effective crack segmentation.

2.3 Attention Mechanisms for Feature Fusion

The Squeeze-and-Excitation (SE) network [19] recalibrates channel feature responses through average pooling, pioneering the concept of channel attention and enhancing the representational capacity of CNNs. Subsequently, the Convolutional Block Attention Module (CBAM) [20] extends the attention mechanism by sequentially applying a Channel Attention Module (CAM) and a Spatial Attention Module (SAM), utilizing both average and max pooling information to enable the network to more accurately focus on relevant features and their locations. In the context of crack segmentation, crackmer introduces a Complementary Fusion Module to more effectively aggregate multi-scale features, leveraging complementary information from different feature levels to improve segmentation accuracy. Nevertheless, the fusion of max and average poolings information is not sufficient.

3 DSCformer for Crack Segmentation

Figure 2 shows the DSCformer framework. Current hybrid crack segmentation methods combining CNN and Transformer mainly focus on feature fusion or sophisticated decoders. However, we believe the limitation lies in feature extraction, not fusion, as conventional convolutions struggle to capture crack features effectively. Dynamic Snake Convolution (DSConv) attempted to improve this but faced challenges with its offset calculation and iteration strategy. Inspired by this, we propose DSCformer, a dual-branch model combining enhanced DSConv and Transformer. The convolutional branch stacks DSC blocks with enhanced DSConv and Weighted Channel Attention

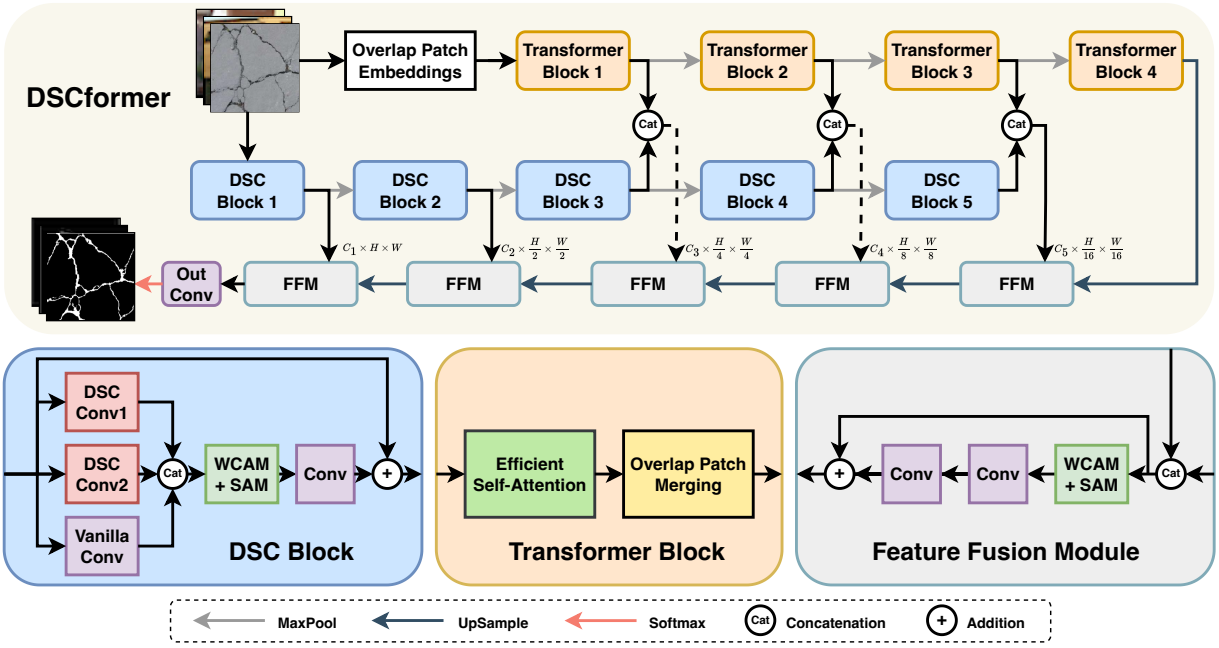

Figure 2. The overview architecture of the proposed DSCformer.

(WCAM), while the Transformer branch uses SegFormer. Crack segmentation masks are generated through progressive feature fusion and upsampling.

3.1 Enhanced DSConv encoder branch

The DSConv encoder branch is employed for local fine-grained structure feature extraction. We refine the computation strategy of DSConv, and propose a DSC block and WCAM. The DSC block incorporates WCAM to enhance feature fusion between vanilla convolution and enhanced DSConv. By stacking DSC blocks and adding max-pooling layers between DSC blocks, we perform progressive downsampling to obtain multi-scale feature maps at resolutions of 1/1, 1/2, 1/4, 1/8, and 1/16.

3.1.1 Enhanced Dynamic Snake Convolution

In this section, we introduce the computation process of Dynamic Snake Convolution (DSConv) and present our improvements. Given a standard $3 * 3$ convolution kernel with a dilation of 1 as K, the kernel samples information from nine points: the central point $K_i = (x_i, y_i)$ and its surrounding eight points, which is expressed as:

$$K = (x-1, y-1), (x-1, y), ..., (x+1, y+1) \quad (1)$$

To enable the convolution kernel to adapt to the tubular topological features around the current location, the goal is to allow the sampling points of the kernel to shift. Inspired by the concept of deformable convolutions, Qi et al. introduced deformation offsets Δ [8]. The offset convolution outputs offsets of 9 sampling points in the vertical and horizontal directions for a total of 18 channels. However, allowing the offsets to be completely learned freely can cause the receptive field to drift away from the target, which is unsuitable for capturing the characteristics of slender tubular structures. To address this, an iterative strategy is adopted to constrain the offsets, ensuring continuity in perception.

Unlike previous method that utilizes a $3 * 3$ convolution kernel to compute the sampling point offsets, we propose a pyramid-structured convolution kernel, which contains kernel sizes of $3 * 3$, $5 * 5$, $7 * 7$, $9 * 9$. Specifically, the $3 * 3$ kernel in our design outputs offsets for points near the center of the dynamic snake convolution, while the larger sizes of kernel outputs offsets for points further away, up to the $9 * 9$ kernel. This approach gradually perceives information from the outer layers, matching the receptive field for offset acquisition with the actual sampling points, thereby avoiding limited perception issues.

In the iterative offset strategy of DSConv, we consider a $9 * 9$ square, which is also its maximum receptive field. Each grid position in the kernel K is represented as $K_{t\pm c} = (x_{t\pm c}, y_{t\pm c})$, where $c = 0, 1, 2, 3, 4$. Unlike previous method, which fixes the offset as an integer in one direction while leaving the offset in the another direction unchanged, the enhanced DSConv do not fix the offset and iterate it in bi-direction simultaneously. So $t \pm c$ represents the iterative order of the convolution sampling points. Starting from the center point K_t,

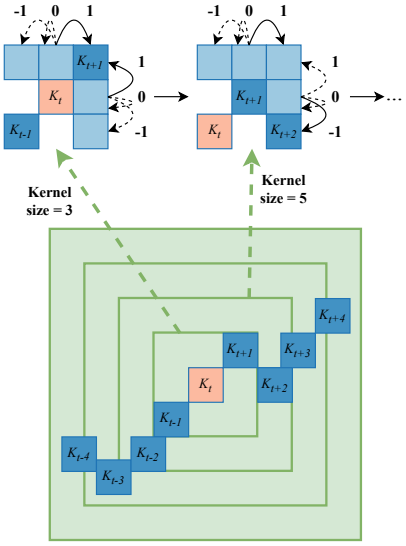

Figure 3. Offset iteration process. Iterating from the central point, each layer of the pyramid kernel represents its maximum receptive field.

$$K_{t+c} = (x_{t+c}, y_{t+c}) = (x_t + \sum_{t}^{t+c} \Delta x, y_t + \sum_{t}^{t+c} \Delta y) \quad (2)$$

$$K_{t-c} = (x_{t-c}, y_{t-c}) = (x_t - \sum_{t-c}^{t} \Delta x, y_t - \sum_{t-c}^{t} \Delta y) \quad (3)$$

After obtaining the coordinate of each sampling point after the offset iteration, the actual sampling point value is obtained by applying bilinear interpolation:

$$K = \sum_{K'} G(K', K) \cdot K' \quad (4)$$

Where K represents the position of the coordinates, K' represents the grid point near K, and G(·, ·) is the bilinear interpolation kernel. Note that G is two dimensional. It is separated into two one dimensional kernels as:

$$G(K, K') = g(K_x, K'_x) \cdot g(K_y, K'_y) \quad (5)$$

As shown in Figure 3, the enhanced DSConv improves the crack sensing ability of the snake chain away from the center point, and more flexibly captures the crack characteristics in the $9 * 9$ receiving field.

3.1.2 Weighted Channel Attention Module

In order to better focus on more relevant channel information, we propose WCAM, which is based on CAM, and makes better utilization of average pooling and maximum pooling information by introducing a few parameters. Reduces the deviation of attention due to accidental maximum values.

Mathematically, similarly, spatial information is first aggregated using max pooling and average pooling operations to obtain two distinct spatial identifiers: F^c_{avg}, F^c_{max}, each containing channel information. These two identifiers are then passed through independent MLPs, denoted as $M_{avg}(\cdot) = W^1_{avg}(W^0_{avg}(\cdot))$ and $M_{max}(\cdot) = W^1_{max}(W^0_{max}(\cdot))$ which, compared to a shared MLP, introduce only a small number of additional parameters with minimal impact on computational complexity. After applying the sigmoid activation function, the two output feature vectors are weighted, with the weights being learnable parameters. In summary, WCAM can be computed as follows:

$$M_c(F) = \sigma(w_{avg} M_{avg}(F^c_{avg}) + w_{max} M_{max}(F^c_{max})) \quad (6)$$

Here, σ denotes the sigmoid function, F contains F^c_{avg} and F^c_{max}, $W^0_{avg}, W^0_{max} \in \mathbb{R}^{C/r \times C}$, $W^1_{avg}, W^1_{max} \in \mathbb{R}^{C \times C/r}$, and $w_{avg}, w_{max} \in \mathbb{R}^C$. The ReLU activation function is followed by W^0_{avg}, W^0_{max}.

3.1.3 DSC block

In the DSConv branch, we propose the DSC block to extract features at different scales. Initially, the feature map is fed into three parallel convolutions: two enhanced DSConvs for extracting fine structural features especially and one standard convolution for capturing local features. The purpose of using two enhanced DSConvs is to better compare the original module (snake convolutions in both x- and y-directions), and the segmentation accuracy is virtually the same as when using only one enhanced DSConv. The outputs of the three convolutions are concatenated along the channel dimension and then processed through WCAM and SAM to adjust the feature map values, focusing on important channel information and spatial positions. This is followed by a standard convolution layer for feature fusion. Finally, a residual connection is introduced to mitigate the issue of gradient vanishing.

3.2 SegFormer encoder branch and decoder

To capture global information, often challenging for convolutional models, we integrate a Transformer branch using the SegFormer-b0 encoder, known for its efficient self-attention and Non-positional encodings, ideal for semantic segmentation. With only three million parameters, this encoder offers a lightweight yet robust solution. Fine-tuned on the ADE20k dataset [21], the pre-trained SegFormer-b0 enhances the DSConv branch by providing background context, improving crack segmentation. The encoder outputs multi-scale feature maps at 1/4, 1/8, 1/16, and 1/32 of the original resolution.

Table 1. Performance Comparison on Crack3238 and FIND Datasets with three test results.

Dataset	Methods	P. (M)	IoU (%) ↑	Pre. (%) ↑	Re. (%) ↑	F1 (%) ↑	HD (mm) ↓
Crack 3238	Unet	17.3	$53.43_{\pm 0.13}$	$65.74_{\pm 0.57}$	$67.67_{\pm 3.42}$	$66.30_{\pm 0.15}$	-
	DeepCrack	30.9	$54.22_{\pm 0.77}$	$65.86_{\pm 1.29}$	$72.06_{\pm 0.87}$	$67.25_{\pm 0.89}$	$56.28_{\pm 5.82}$
	DSCNet	2.1	$53.99_{\pm 0.16}$	$67.15_{\pm 1.69}$	$70.45_{\pm 1.98}$	$66.89_{\pm 0.19}$	$41.15_{\pm 0.90}$
	UCTransNet	66.5	$55.40_{\pm 0.31}$	$67.61_{\pm 0.83}$	$73.26_{\pm 0.38}$	$68.52_{\pm 0.43}$	$44.23_{\pm 1.82}$
	DcsNet	15.2	$56.85_{\pm 0.09}$	$69.83_{\pm 1.48}$	$73.73_{\pm 1.37}$	$70.05_{\pm 0.08}$	$37.17_{\pm 0.73}$
	DTrC-Net	41.8	$56.90_{\pm 0.33}$	$70.17_{\pm 1.42}$	$73.11_{\pm 1.51}$	$70.06_{\pm 0.39}$	-
	DSCformer(ours)	14.8	$\mathbf{58.74}_{\pm 0.47}$	$\mathbf{72.38}_{\pm 2.22}$	$\mathbf{73.99}_{\pm 1.33}$	$\mathbf{71.74}_{\pm 0.48}$	$\mathbf{33.35}_{\pm 0.97}$
FIND	Unet	17.3	$85.57_{\pm 0.18}$	$\mathbf{92.97}_{\pm 0.61}$	$91.65_{\pm 0.82}$	$92.00_{\pm 0.11}$	$13.21_{\pm 0.73}$
	DeepCrack	30.9	$84.25_{\pm 0.16}$	$91.40_{\pm 0.84}$	$91.65_{\pm 0.88}$	$91.21_{\pm 0.09}$	$14.12_{\pm 0.96}$
	DSCNet	2.1	$85.75_{\pm 0.07}$	$92.35_{\pm 0.70}$	$92.40_{\pm 0.67}$	$92.12_{\pm 0.04}$	$12.23_{\pm 0.20}$
	UCTransNet	66.5	$86.05_{\pm 0.26}$	$91.15_{\pm 0.29}$	$93.98_{\pm 0.48}$	$92.30_{\pm 0.16}$	$11.90_{\pm 0.61}$
	DcsNet	15.2	$85.02_{\pm 0.09}$	$91.97_{\pm 1.00}$	$91.98_{\pm 1.09}$	$91.64_{\pm 0.06}$	$14.67_{\pm 0.86}$
	DTrC-Net	41.8	$74.28_{\pm 0.07}$	$83.50_{\pm 0.54}$	$84.94_{\pm 1.59}$	$84.21_{\pm 0.02}$	$14.76_{\pm 0.09}$
	DSCformer(ours)	14.8	$\mathbf{87.31}_{\pm 0.12}$	$92.52_{\pm 0.35}$	$\mathbf{94.01}_{\pm 0.45}$	$\mathbf{93.04}_{\pm 0.09}$	$\mathbf{11.14}_{\pm 0.23}$

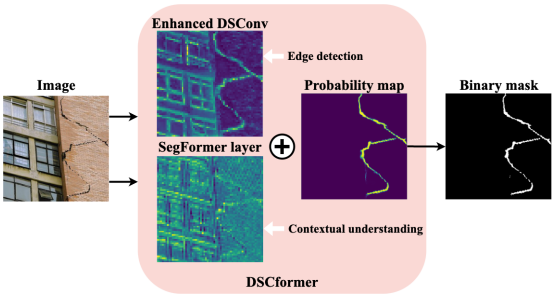

Figure 4. The difference between dual-branch encoder.

For the decoder, we fuse feature maps from the DSConv and SegFormer branches with upsampled lower-resolution features. These fused maps undergo WCAM and SAM processing to refine channel and spatial information. The refined features pass through two convolutional layers with residual connections to smooth the output and address vanishing gradients. As shown in Figure 4, enhanced DSConv excels at extracting edges and fine structures, while SegFormer is effective at global modeling. The final crack mask is obtained by fusing these features.

4 Experiments

4.1 Datasets

We use Crack3238 dataset [18], Fused Image dataset for convolutional neural Network-based crack Detection(FIND) dataset [22] to evaluate our method. Crack3238 dataset contains a total of 3,238 images, which consist of RGB images of cracks from various sources, featuring complex backgrounds, diverse crack patterns, and defective surfaces, significantly increasing the complexity of crack detection scenarios. The FIND dataset comprises 2,500 images, where fused image data is obtained by combining depth information with grayscale images. Both datasets encompass data from different sources and highly complex scenes, making them extremely challenging for crack segmentation tasks. We split the datasets, using 80% for training and 20% for testing.

4.2 Implementation Details

4.2.1 Training details.

Our model, implemented in PyTorch, was trained on a RTX 3090Ti GPU. To reduce overfitting, we applied data augmentation, including flips, rotations, and affine transformations. The SegFormer encoder uses pre-trained weights, while the DSConv encoder and decoder are randomly initialized. We used Adam (lr=0.0001, weight decay=10^{-4}) with cross-entropy and dice loss. Training ran for 100 epochs with a batch size of 8.

4.2.2 Performance metrics.

We evaluated segmentation performance using IoU, Recall, Precision, F1-score, and Hausdorff Distance. The formulations are as follows: IoU is given by $IoU = \frac{TP}{TP+FP+FN}$, Recall is given by $Recall = \frac{TP}{TP+FN}$, Precision is given by $Precision = \frac{TP}{TP+FP}$, and F1-score is given by $F1 = \frac{2\times Precision\times Recall}{Precision+Recall}$. Additionally, we employed the Hausdorff Distance, calculated as $d_H(A,B) = \max\left(\max_{a\in A}\min_{b\in B}|a-b|, \max_{b\in B}\min_{a\in A}|a-b|\right)$, to measure the maximum distance between the segmentation mask and the ground truth. All metrics for each image are calculated and averaged.

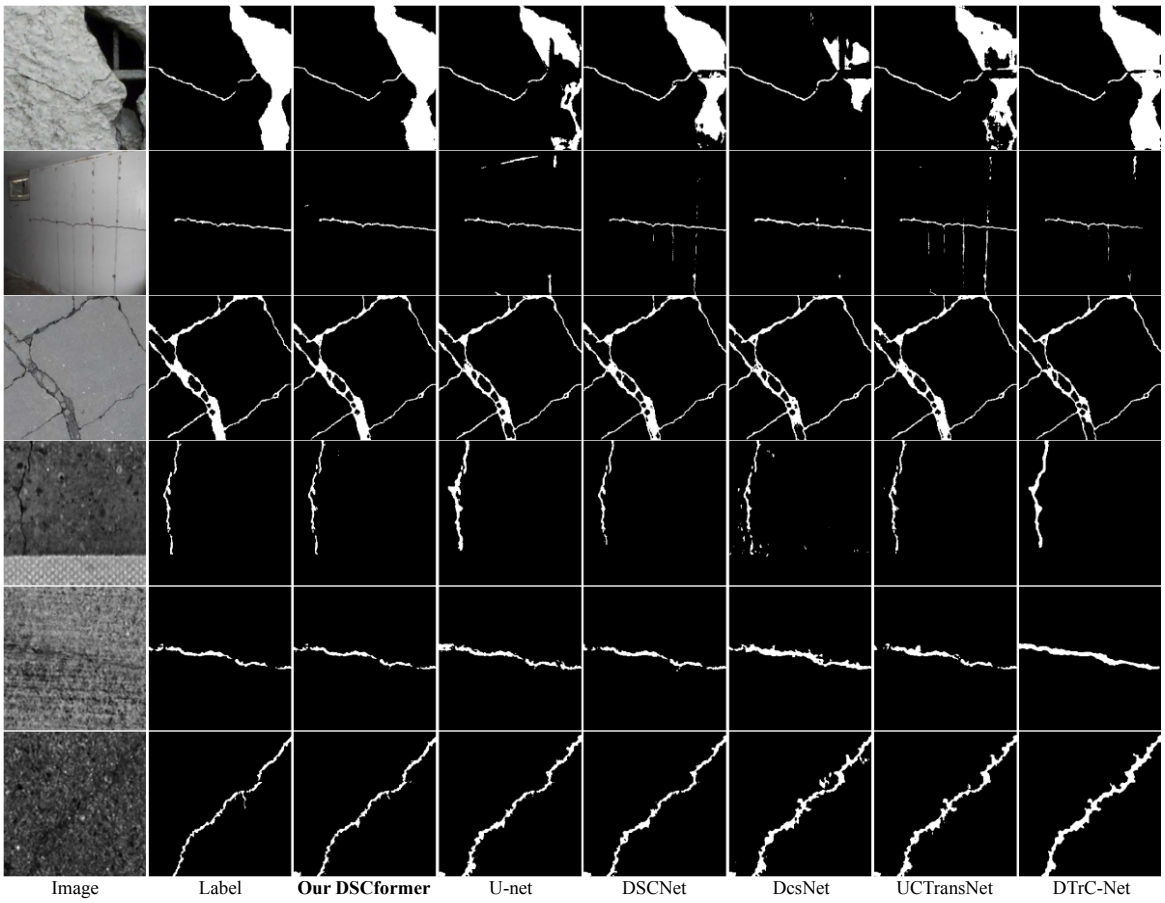

Figure 5. The qualitative comparison on the Crack3238 and FIND datasets.

4.3 Comparison with State-of-the-art Methods

We compare the proposed DSCformer with several state-of-the-art models, including U-Net, DeepCrack, DcsNet, UCTransNet, DTrCNet, and DSCNet. For fairness, we use their original code and settings, testing each model three times. Results are shown in Table 1, with metrics presented as mean ± standard deviation, highlighting the best results in bold. DSCformer outperforms all models, achieving an IoU of 58.74%, Precision of 72.38%, Recall of 73.99%, and F1-score of 71.74% on the Crack3238, and an IoU of 87.31% and F1-score of 93.04% on the FIND. It also balances Precision and Recall effectively, with Precision achieving second-best performance.

Additionally, a visual comparison in Figure 5 illustrates DSCformer's superior performance, particularly in handling noise and accurately capturing large defect areas. DSCformer also excels at delineating crack edges, avoiding the overestimation of crack width seen in other models, and ensuring more precise segmentation, especially at boundaries. This level of detail highlights DSCformer's strong generalization across varying defect types and sizes.

Table 2. Generalization performance on Crack3238 datasets.

Augmentation	DSCformer	DTrC-Net	DSCNet
None	**58.74%**	56.90%	53.99%
Geometric	**53.52%**	50.50%	49.94%
Appearance	**49.36%**	46.38%	48.04%
Combined	**42.40%**	37.50%	39.69%

We also conducted a generalization experiment to evaluate the performance of three models—DSCformer, DTrC-Net, and DSCNet—under various data augmentation conditions, including no augmentation, geometric augmentation, appearance augmentation, and combined augmentation. The results in Table 2 show that DSCformer consistently outperforms the other models across all augmentation types, though its performance declines with increased augmentation complexity, especially under combined augmentation (42.40%). In comparison, DTrC-Net and DSCNet exhibit more significant performance drops, particularly under appearance and combined augmentations. Al-

though affected by the complexity of data augmentation, DSCformer still shows strong generalization ability.

Additionally, in terms of computational efficiency, DSCformer achieves an inference speed of 30 FPS, which is comparable to the hybrid model UCTransNet, while other models range between 60-100 FPS. This shows that DSCformer balances accuracy and computational efficiency well without excessive resource consumption.

Overall, DSCformer's ability to manage noise, precisely segment large defects, and accurately delineate crack edges makes it a highly effective method for crack segmentation tasks, outperforming existing models both quantitatively and visually. Additionally, it demonstrates strong generalization ability, maintaining superior performance across different data augmentation conditions.

4.4 Ablation Studies

We conducted ablation experiments on the Crack3238 dataset to assess the impact of the proposed improvements on the performance of our model. As summarized in Table 3, the dual-branch architecture of DSCformer exhibits a significant advantage over the single-branch models, DSCNet and SegFormer. Specifically, DSCformer achieved an IoU improvement of 4.75% over DSCNet and 4.07% over SegFormer, highlighting the performance of integrating both branches for crack segmentation.

Table 3. Ablation experiments on Crack3238 datasets. 'CA' denotes the Channel Attention.

Methods	Conv	CA	IoU
DSCNet	DSConv	w/o	53.99%
SegFormer	w/o	w/o	54.67%
DSCformer	Vanilla Conv	WCAM	57.46%
	DSConv	WCAM	57.74%
	Enhanced DSConv	w/o	58.16%
	Enhanced DSConv	CAM	58.28%
	Enhanced DSConv	WCAM	58.74%

The enhanced DSConv also shows substantial benefits. By incorporating the enhanced DSConv into the model, we observed a notable 1.00% increase in IoU compared to the original DSConv, indicating that the improvements to the DSConv effectively contribute to more accurate crack boundary detection and segmentation. Additionally, the introduction of WCAM provided further gains in segmentation performance. When WCAM was applied within DSCformer, it outperformed the standard CAM, resulting in a 0.46% increase in IoU. This improvement reflects the value of our modifications to the attention mechanism. Overall, these results emphasize the significance of the hybrid model architecture and the specific enhancements to both the dynamic snake convolution and channel attention mechanisms. Each modification contributes to the overall effectiveness of DSCformer, leading to superior crack segmentation performance.

5 Conclusion

In this work, we introduced DSCformer, a hybrid model combining enhanced Dynamic Snake Convolution (DSConv) and Transformer architectures for crack segmentation. DSCformer addresses the limitations of CNNs and Transformers by capturing both fine local details and global context. The enhanced DSConv improves crack feature detection with a pyramid kernel and bidirectional offset iteration, while the Weighted Channel Attention Module (WCAM) refines channel attention for better feature fusion. Experiments show that DSCformer outperforms state-of-the-art methods on benchmark datasets, demonstrating its robustness and potential for broader applications in analyzing fine topological structures.

6 Acknowledgment

This material is based upon work supported by the National Science Foundation of China(NSFC)62203205 and U1913603. The views, opinions, findings and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the NFSC.

References

[1] Jay Lee and Hanqi Su. A unified industrial large knowledge model framework in industry 4.0 and smart manufacturing. *International Journal of AI for Materials and Design*, page 3681, 2024.

[2] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

[3] Wentao Zhang, Haoran Kang, Wenhui Wang, Yangtao Ge, Haiou Liao, Rui-Jun Yan, I-Ming Chen, Zhenzhong Jia, and Jing Wu. A high-accuracy crack defect detection based on fully convolutional network applied to building quality inspection robot. In *2022 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 63–68. IEEE, 2022.

[4] Elyas Asadi Shamsabadi, Chang Xu, Aravinda S Rao, Tuan Nguyen, Tuan Ngo, and Daniel Dias-da Costa. Vision transformer-based autonomous crack detection on asphalt and concrete surfaces. *Automation in Construction*, 140:104316, 2022.

[5] Yunhe Gao, Mu Zhou, and Dimitris N Metaxas. Utnet: a hybrid transformer architecture for medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24*, pages 61–71. Springer, 2021.

[6] Huajun Liu, Xiangyu Miao, Christoph Mertz, Chengzhong Xu, and Hui Kong. Crackformer: Transformer network for fine-grained crack detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3783–3792, 2021.

[7] Jin Wang, Zhigao Zeng, Pradip Kumar Sharma, Osama Alfarraj, Amr Tolba, Jianming Zhang, and Lei Wang. Dual-path network combining cnn and transformer for pavement crack segmentation. *Automation in Construction*, 158:105217, 2024.

[8] Yaolei Qi, Yuting He, Xiaoming Qi, Yuan Zhang, and Guanyu Yang. Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6070–6079, 2023.

[9] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL `https://arxiv.org/abs/1505.04597`.

[12] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[13] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[16] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021.

[17] Haonan Wang, Peng Cao, Jiaqi Wang, and Osmar R Zaiane. Uctransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2441–2449, 2022.

[18] Chao Xiang, Jingjing Guo, Ran Cao, and Lu Deng. A crack-segmentation algorithm fusing transformers and convolutional neural networks for complex detection scenarios. *Automation in Construction*, 152: 104894, 2023.

[19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[20] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[21] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[22] Shanglian Zhou, Carlos Canchila, and Wei Song. Fused image dataset for convolutional neural network-based crack detection (find), 2022. URL `https://doi.org/10.5281/zenodo.6383044`.