A Lightweight CNN–Mamba Hybrid Architecture for Efficient Crack Segmentation

Masaya Shimasaki [1], Mitsuteru Sakamoto [1], Toshiaki Satoh [1]

[1] PASCO Corporation, Tokyo, Japan {miaksa4598, moittu9191, tuoost7017}@pasco.co.jp

Keywords: Crack Segmentation, Lightweight Neural Network Model, Mobile Mapping System, Selective State Space Model

Abstract

Pavement crack segmentation is important for road infrastructure inspection, but practical deployment remains challenging because many high-performance deep learning models require substantial computational resources. This issue is particularly critical in large-scale Mobile Mapping System (MMS)-based workflows, where large volumes of road surface images must be processed efficiently. To address this problem, this study proposes a lightweight CNN-Mamba hybrid architecture for crack segmentation as a deployment-oriented redesign of CT-CrackSeg. The proposed method replaces the original MobileViT-based global modelling modules with EfficientViM-inspired blocks based on hidden-state mixer-based state space duality (HSM-SSD), while preserving the overall encoder-decoder structure and refining the boundary enhancement branch with DCNv2-based deformable convolution. Experiments on the publicly available GAPs384 and CamCrack789 datasets show that the proposed model maintains competitive topology-aware segmentation performance while substantially improving computational efficiency. Compared with CT-CrackSeg, the proposed model increases inference speed from 1.49 to 4.44 FPS on GAPs384 and from 1.32 to 3.92 FPS on CamCrack789, while reducing peak memory consumption from 2827 MB to 355 MB on both datasets. At the same time, the clDice score remains comparable, changing from 0.760 to 0.758 on GAPs384 and from 0.921 to 0.922 on CamCrack789. These results indicate that the proposed architecture provides a favourable balance between crack segmentation quality and deployment efficiency, making it a practical option for large-scale pavement inspection and photogrammetric infrastructure monitoring.

1. Introduction

The aging of road infrastructure, particularly that developed during Japan’s period of rapid economic growth, has become a pressing social concern, with further deterioration expected in the coming years. At the same time, a declining and aging population has led to a shortage of skilled engineers, making it increasingly difficult to sustain conventional, labour-intensive inspection practices. These challenges highlight the urgent need for efficient and scalable technologies that enable reliable infrastructure assessment with limited human resources (Ministry of Land, Infrastructure, Transport and Tourism, 2025).

In the domain of road infrastructure, routine inspections play a critical role in evaluating pavement conditions and supporting timely maintenance planning. Traditionally, such evaluations have relied on images acquired by specialized pavement inspection vehicles, where defects such as cracks and rutting are assessed. However, crack detection from road surface images still requires substantial manual review and visual confirmation in many operational workflows, resulting in significant time and labour costs and limiting scalability in large-scale monitoring workflows.

Recent advances in Mobile Mapping Systems (MMS) and photogrammetric technologies have enabled the large-scale and high-density acquisition of road surface data through the integration of high-resolution cameras and high-density LiDAR sensors. While these developments improve the detectability of fine-scale defects such as pavement cracks, they also substantially increase the volume of data and the associated computational burden for downstream analysis.

In parallel, deep learning-based image recognition techniques have shown remarkable success in automating road damage detection. In particular, semantic segmentation methods, which perform pixel-level classification, have demonstrated strong capability in accurately extracting cracks with complex geometries (Gong et al., 2024). Despite their high accuracy, many existing models require substantial computational resources, leading to high processing costs and limited applicability in real-world scenarios, especially in resource-constrained environments such as vehicle-mounted or edge computing platforms (Jing et al., 2025).

Therefore, beyond segmentation accuracy, computational efficiency has become a key requirement for practical deployment (Zhang et al., 2024). Efficient models not only enable scalable processing of large datasets but also facilitate real-time or near-real-time analysis within operational mapping pipelines. Furthermore, improved efficiency in crack detection can support more timely maintenance planning and help reduce inspection-related operational burdens, which is essential for sustainable infrastructure management in modern urban environments.

Recently, Mamba-based architectures have been recognized as an efficient means of modelling long-range dependencies in vision tasks (Gu and Dao, 2024). Considering the need for practical and scalable crack inspection, this study aims to improve the efficiency-accuracy trade-off in pavement crack segmentation by proposing a lightweight CNN-Mamba hybrid architecture that reduces memory consumption and inference time while maintaining competitive segmentation performance.

The main contributions of this study are as follows. First, we present a lightweight redesign of a hybrid crack segmentation architecture with a focus on deployment efficiency. Second, we investigate whether computational cost can be reduced while maintaining topology-aware segmentation quality. Third, we demonstrate through experiments on publicly available datasets that the proposed design achieves a more favourable balance between segmentation performance and computational efficiency.

2. Related Works

2.1 Crack Segmentation Methods

The evolution of crack segmentation methods has largely been driven by the need to balance fine local feature extraction and global contextual modelling. Early studies mainly relied on CNN-based methods, which became the dominant approach because they effectively capture local spatial patterns and are therefore well suited for thin and fine crack structures. Representative models such as DeepCrack (Zou et al., 2018) exploit hierarchical convolutional features to enhance crack representation across multiple scales, and subsequent studies (e.g., Choi et al., 2019; Liu et al., 2020) further demonstrated the effectiveness of CNNs for crack feature extraction. However, because CNNs operate with local receptive fields, they are limited in their ability to model long-range dependencies and global contextual relationships, which are important for preserving the continuity and connectivity of elongated or fragmented cracks.

To overcome this limitation, Transformer-based methods were introduced into crack segmentation (e.g., Liu et al., 2021; Guo et al., 2023). By leveraging self-attention, these methods can capture long-range dependencies and global context over the entire image, which is advantageous for representing elongated and spatially correlated crack structures. Crackformer (Liu et al., 2023) is a representative example. Nevertheless, the high computational and memory cost of self-attention often limits the practicality of Transformer-based models, especially in high-resolution or resource-constrained scenarios.

To combine the strengths of both paradigms, recent studies proposed hybrid CNN-Transformer architectures that integrate local feature extraction with global context modelling. For example, CT-CrackSeg (Tao et al., 2023) combines residual convolutional blocks with MobileViT (Mehta and Rastegari, 2021) in an encoder-decoder framework. Similar hybrid designs have also been reported in recent crack segmentation studies, as summarized in Table 1 (e.g., Zhang et al., 2024; Zim et al., 2025; Goo et al., 2025). Although these methods often improve segmentation performance, they may still retain the computational burden associated with Transformer components.

More recently, State Space Models (SSMs) have emerged as an alternative to Transformers for modelling long-range dependencies in vision tasks (Gu and Dao, 2024). Among them, Mamba has attracted increasing attention because its linear computational complexity makes it potentially more scalable for high-resolution inputs. This property is particularly appealing for crack segmentation, where both fine-detail preservation and long-range structural modelling are essential. Recent studies have therefore begun to explore both Mamba-based and CNN-Mamba hybrid methods. For example, SCSegamba (Liu et al., 2025) demonstrated the potential of Vision Mamba for lightweight and accurate crack extraction, while DCCM-Net (Zhang et al., 2025) combined convolution and Mamba to improve edge-aware feature extraction and contextual modelling. Similar designs have also been reported in recent studies (e.g., Zuo et al., 2024; Zhang et al., 2025; Zhao et al., 2026).

These studies suggest that Mamba is a promising direction for crack segmentation. At the same time, recent studies indicate that the practical efficiency of vision Mamba models does not always fully reflect their theoretical complexity advantage, as realized throughput can depend strongly on scan design, memory access patterns, and implementation details.

Table 1 Summary of Related Models

Methods	Backbone	Published
DeepCrack	CNNs	2018
Crackformer- II	Transformer	2023
CT-CrackSeg	CNNs+Transformer	2023
DECS-Net	CNNs+Transformer	2024
EfficientCrackNet	CNNs+Transformer	2025
Hybrid-Segmenter	CNNs+Transformer	2025
SCSegamba	Mamba	2025
DCCM-Net	CNNs+Mamba	2025

2.2 Long-range Dependencies Modelling Methods

Long-range dependency modelling is a key architectural issue in dense prediction tasks such as crack segmentation, because local feature extractors alone often struggle to preserve the continuity of thin and spatially extended structures. Self-attention (Vaswani et al., 2017) has been widely used for this purpose because it explicitly models global token-to-token interactions. However, its quadratic computational and memory complexity can become a practical limitation for high-resolution inputs. Table 2 summarizes representative long-range dependency modelling strategies and efficiency-oriented variants relevant to this study.

Existing studies have explored two main directions for improving global context modelling efficiency. The first direction seeks to make attention itself more efficient. For example, FlashAttention (Dao et al., 2022) improves the practical efficiency of exact self-attention through an IO-aware implementation, while Linear Attention (Katharopoulos et al., 2020) reduces computational complexity by reformulating attention into a linear-time approximation. These studies suggest that practical efficiency depends not only on asymptotic complexity but also on whether the dominant operations are well aligned with hardware-efficient execution.

The second direction replaces explicit pairwise token interaction with state-based sequence modelling. State Space Models (SSMs) provide a linear-complexity framework in which long-range information is propagated through latent state transitions rather than direct token-to-token affinity computation. This line of research was further developed by Structured State-Space Duality (SSD) (Dao and Gu, 2024), which clarified the relationship between SSMs and attention-style sequence transformations and provided a useful foundation for efficient long-range modelling beyond conventional self-attention.

Nevertheless, theoretical linear complexity does not always translate directly into proportional runtime gains in vision tasks. In practice, the realized efficiency of visual Mamba-type modules can still be strongly affected by scan strategy, recurrent computation patterns, memory access behaviour, token length, and implementation details. Several recent studies have shown that overhead from data rearrangement, input-space projections, and limited hardware utilization can weaken the expected throughput advantage of state-space-based vision modules, especially in deployment-constrained settings (Han et al., 2024; Lee et al., 2025; Yoshimura et al., 2026). Therefore, practical efficiency should be discussed not only in terms of computational complexity, but also in terms of how dominant operations are organized in actual implementations.

From this perspective, HSM-SSD, introduced in EfficientViM (Lee et al., 2025), is particularly relevant. Instead of performing

major channel-mixing and gating operations directly in the input token space, HSM-SSD moves them into a compressed hidden-state space. This reduces the amount of data involved in projection, mixing, and gating, thereby alleviating memory-bound overhead and improving realized execution efficiency. In this sense, HSM-SSD can be regarded as a hardware-conscious refinement of SSD that focuses on practical inference efficiency rather than theoretical FLOPs reduction alone.

For crack segmentation, this issue is especially important because the target structures are thin, elongated, and topologically sensitive, requiring both local detail preservation and sufficiently wide contextual modelling. Although recent CNN–Transformer and CNN–Mamba models have demonstrated the benefit of global context, many of them still rely on relatively costly global modules, and the practical efficiency of Mamba-based designs often remains sensitive to architectural choices. In this study, we adopt CT-CrackSeg as the baseline architecture because its hybrid encoder–decoder design is relatively clear and modular, making it well suited for selectively replacing the original global modelling blocks while preserving the overall network structure. Based on this property, we examine whether its original MobileViT-based global modelling blocks can be replaced with EfficientViM-inspired HSM-SSD blocks to obtain a more favourable balance between segmentation quality and computational efficiency.

Table 2 Comparison of representative long-range dependency modelling strategies and efficiency-oriented variants

Method	Backbone	Complexity	Main practical characteristic
Self-Attention	Transformer	Quadratic	Strong global modelling but costly in memory and runtime for high-resolution inputs
FlashAttention	Transformer	Quadratic	Improves runtime and memory efficiency without changing the attention formulation
Linear Attention	Transformer	Linear	Reduces complexity, but effectiveness depends on the approximation and implementation
SSM	Mamba	Linear	Efficient sequence modelling in principle, but realized speed depends on implementation and scan design
SSD	Mamba	Linear	Improves the practicality of state-space-based sequence modelling
HSM-SSD	Mamba	Linear	Reduces memory-bound overhead by shifting dominant operations into hidden-state space

3. Methodology

3.1 Overall Design

We redesign CT-CrackSeg from a deployment-oriented perspective by replacing its MobileViT-based global modelling blocks with EfficientViM-inspired HSM-SSD blocks, while preserving the overall encoder-decoder framework. The motivation is not to exactly reproduce the original self-attention-style computation, but to provide a more efficient alternative for the long-range dependency modelling role that such global blocks play in hybrid crack segmentation networks. In pavement crack segmentation, this role is important because crack structures are thin, elongated, and often spatially disconnected, making both local detail preservation and wide contextual aggregation necessary. However, explicit attention-style global modelling can still impose substantial computational and memory costs in practical dense prediction settings. Therefore, our redesign focuses on replacing the heavy global modelling component with a lighter sequence-mixing mechanism that can still support long-range contextual interaction.

As shown in Figure 1, the proposed model retains the U-shaped architecture of CT-CrackSeg, consisting of an initial shallow residual block with dilated convolutions, three encoder stages, a bridge module, and four decoder stages with skip connections. The initial block extracts low-level features from the RGB input to produce a 32-channel feature map. The encoder progressively reduces the spatial resolution while increasing the channel dimension from 32 to 64, 128, and 256, and the bridge further downsamples the deepest encoder feature map and projects it to 512 channels. The decoder then gradually restores the spatial resolution through four up-sampling stages, where each upsampled feature map is concatenated with the corresponding encoder feature map and refined by a residual block. Thus, the proposed model should be regarded as a targeted redesign of the encoder internals rather than a fundamental modification of the original CT-CrackSeg architecture.

Importantly, the proposed network does not adopt the original EfficientViM backbone directly. Instead, it selectively incorporates the core EfficientViM block design—namely HSM-SSD-based hidden-state sequence mixing with lightweight local filtering—into the encoder of CT-CrackSeg, while omitting classification-oriented components such as the stem, stage-wise down-sampling hierarchy, and multi-stage hidden-state fusion head.

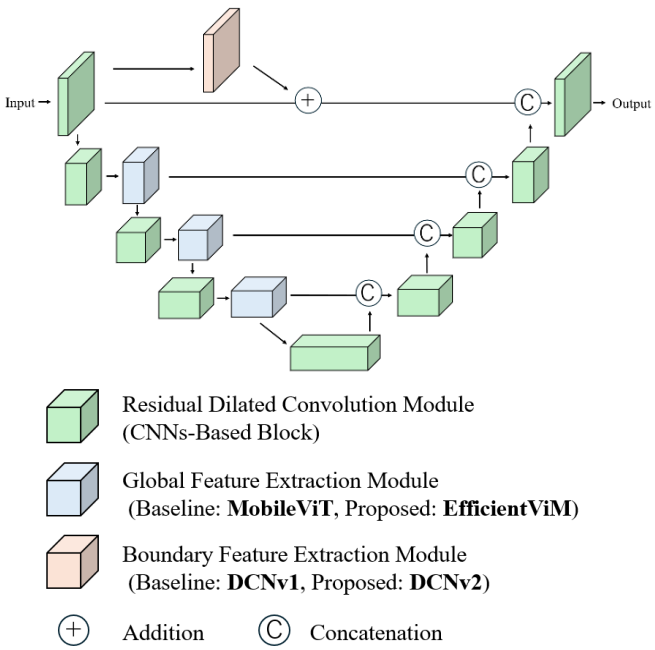

Figure 1 Overall architecture of the proposed model

3.2 EfficientViM-inspired Block

The main architectural modification is introduced in the three encoder down-sampling blocks. In the original CT-CrackSeg, these blocks rely on MobileViT-based global modelling, whereas, as shown in Figure 2, they are reformulated in the proposed network as EfficientViM-inspired blocks built around HSM-SSD. This redesign is based on the view that, for our purpose, the key requirement of the global block is not the explicit computation of pairwise token affinities itself, but the effective aggregation of long-range contextual information. From this perspective, HSM-SSD serves as an efficient alternative to attention-style global modelling by capturing long-range dependencies through hidden-state sequence mixing rather than explicit all-pair interactions.

Specifically, the input feature first undergoes a depthwise convolution for lightweight local spatial filtering. The resulting feature map is then flattened into a 1D sequence, normalized, and processed by an HSM-SSD mixer. Unlike self-attention, which forms global context through explicit token-to-token affinity computation, HSM-SSD propagates long-range information through compressed hidden states. In addition, the dominant channel-mixing and gating operations are performed in the hidden-state space rather than in the full input token space, which reduces projection cost and memory traffic in practice. After sequence mixing, the features are reshaped back into a 2D feature map and further refined by a second depthwise convolution and a lightweight feed-forward network.

In this way, the proposed encoder block retains the hybrid design principle of combining local filtering and global context modelling, while reducing the practical overhead of the global module. In our implementation, the hidden-state dimensions of the three encoder blocks are set to 64, 32, and 16, respectively, with reference to the stage-wise hidden-state configuration used in EfficientViM.

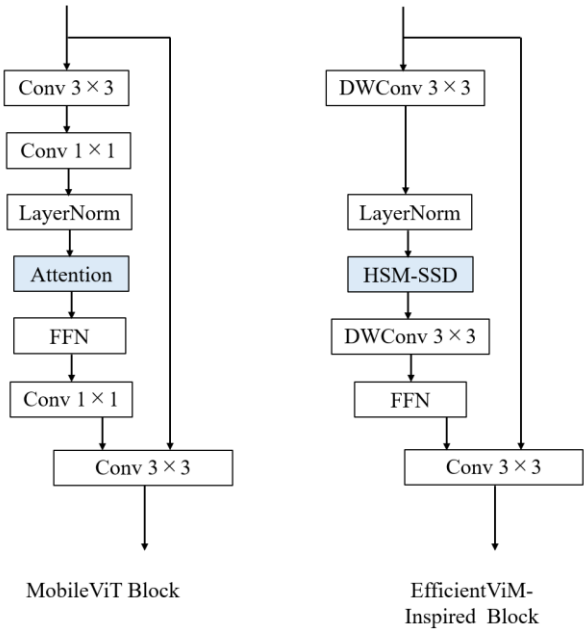

Figure 2 Comparison of the MobileViT block and the proposed EfficientViM-inspired block

3.3 Boundary Enhancement Branch

The boundary enhancement branch of CT-CrackSeg is retained and refined by replacing its deformable convolution module with a DCNv2-style layer, upgrading the original DCNv1 formulation (Dai et al., 2017) to DCNv2 (Zhu et al., 2019). Attached to the first encoder feature map, this branch produces a single-channel boundary response and injects it back into the corresponding low-level feature by channel-wise broadcasting, thereby sharpening thin crack boundaries.

4. Experiments

4.1 Datasets

Experiments were conducted on two publicly available pavement crack datasets, GAPs384 (Eisenbach et al., 2017) and CamCrack789 (Zhu et al., 2024), with training and evaluation performed separately on each dataset. GAPs384 contains 509 high-resolution pavement images acquired by the S.T.I.E.R. mobile mapping system for large-scale road inspection in Germany. The images have a ground sampling distance of 1.2 mm, and the dataset contains images of two sizes, 440 × 540 pixels and 540 × 640 pixels, in roughly equal proportions. As the dataset originates from a mobile mapping workflow used for practical road inspection, it is closely aligned with mobile mapping, photogrammetric infrastructure monitoring, and geospatial image interpretation. The dataset includes diverse crack patterns and surface conditions, providing a benchmark for evaluating segmentation accuracy and robustness. It is particularly suitable for assessing the detection of thin, low-contrast, and discontinuous crack structures.

CamCrack789 consists of 789 images captured using a Microsoft HD camera mounted on a mobile robot under various environmental and lighting conditions. The dataset includes significant background disturbances such as shadows, stains, debris, and motion blur. All images have a resolution of 640 × 480 pixels and are pixel-wise annotated. While it is not a conventional mobile mapping dataset, it represents a practically related mobile sensing scenario in which pavement images are acquired under dynamic and environmentally variable conditions. In this respect, it serves as a useful complementary benchmark for assessing the robustness of segmentation models in deployment-oriented inspection settings.

4.2 Experimental Settings

To verify the effectiveness of the proposed method, eight representative deep learning architectures, including CT-CrackSeg, were selected for comparison (Table 1). The compared models were selected to cover representative CNN-based, Transformer-based, CNN-Transformer hybrid, Mamba-based, and CNN-Mamba hybrid crack segmentation architectures that are recent, publicly available, and relevant to the efficiency-accuracy trade-off addressed in this study. All compared models were implemented in PyTorch and evaluated under a unified Windows-based environment. Whenever possible, we used the official author-released GitHub implementations with only minimal modifications required for training and evaluation on our datasets.

Some Mamba-based baselines were originally implemented with optimized Mamba-SSM kernels that were not directly usable in our Windows-based environment. In order to benchmark all methods under a unified and practically reproducible condition, we replaced these modules with pure PyTorch implementations (Alexandre, 2024) while preserving their architectural principles as closely as possible. As a result, the measured throughput of these baselines reflects not only model design but also implementation accessibility in a realistic engineering environment. We emphasize that the purpose of this study is not to compare the theoretical upper-bound speed of different architectures under highly optimized platform-specific kernels, but to evaluate practical deployability under conditions that are more likely to be reproducible in operational workflows.

Detailed hyperparameter settings, including the optimizer, learning rate, batch size, and training epochs, are summarized in Table 3. To maintain a consistent evaluation setting, optimization parameters such as learning rate were individually adjusted within a limited range only to achieve stable training for each model, while the overall training protocol, datasets, input settings, and evaluation procedure were kept common across methods. With appropriately selected learning rates, all compared models converged stably in our experiments. In addition, the batch size

for all Mamba-based baselines was fixed due to GPU memory constraints (Table 4). To reduce the influence of randomness during training and evaluation, each experiment was repeated three times using different random seeds, and the average performance over the three runs was reported.

Table 3 Common experimental settings for all models

GPU(OS)	NVIDIA RTX A6000 (Windows)
Dataset Split (GAPs384)	Train:370, Val:100 Test:39
Dataset Split (CamCrack789)	Train:426, Val:120 Test:243
Optimizer	Adam
Epochs	100
Data Augmentation	Random flip and rotation, Motion blur, Color jittering, Random resized cropping
Loss function	Focal tversky loss (α=0.3, β=0.7, Γ=4/3)

Table 4 Experimental settings specific to each model

Methods	Batch Size	Learning Rate
DeepCrack	16	0.0002
Crackformer II	16	0.0002
CT-CrackSeg	16	0.0002
DECS-Net	16	0.0002
EfficientCrackNet	16	0.0005
Hybrid-Segmenter	16	0.0005
SCSegamba	4	0.0002
DCCM-Net	4	0.0001
Ours	16	0.0002

Multi-scale Inference: Inference was conducted using a two-branch multi-scale strategy. In the first branch, the original image was decomposed into overlapping local patches, which were individually fed into the network. Their predicted probability maps were restored to the original image coordinates and averaged in overlapping regions to form a local prediction map. In the second branch, the original image was resized to the network input size and processed as a whole to generate a global prediction map, which was then upsampled to the original image resolution. The final output was obtained by equally averaging the local and global prediction maps.

The purpose of this strategy is to balance detail preservation and contextual consistency at inference time. Because pavement cracks are often thin, elongated, and locally discontinuous, predictions based only on globally resized inputs may miss subtle crack responses, while predictions based only on local crops may lack sufficient contextual information to maintain structural coherence. This multi-scale inference strategy addresses this trade-off by combining a detail-sensitive local view with a context-aware global view.

Loss Function: The Focal Tversky loss was used as the training objective. This loss function is particularly suitable for crack segmentation because the number of crack pixels is significantly smaller than that of background pixels, leading to severe class imbalance. By explicitly controlling the balance between false positives and false negatives, the Focal Tversky loss helps the models focus more effectively on sparse crack regions.

The Tversky index (TI) is defined as:

$$TI = \frac{TP}{TP + \alpha FP + \beta FN} \quad (1)$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively, and α and β control the penalties for false positives and false negatives.
Based on this, the Focal Tversky loss (FTL) is formulated as:

$$FTL = (1 - TI)^{\gamma} \quad (2)$$

where γ is a focusing parameter that emphasizes hard-to-segment examples. By adjusting α, β, and γ, the loss function can effectively address severe class imbalance and improve segmentation performance on thin crack regions. Following the recommended settings in prior studies (Nguyen et al., 2023), the parameters were set to α=0.3, β=0.7, and γ=4/3.

Evaluation: For quantitative evaluation, segmentation accuracy was assessed using clDice (Shit et al., 2021), while computational efficiency was evaluated in terms of frames per second (FPS) and peak memory consumption. In the early stage of this study, the F-value was also considered as an evaluation metric. However, detailed inspection of the GAPs384 annotations revealed slight variations in the thickness of the ground-truth crack labels across samples. Because the F-value is highly sensitive to pixel-level overlap, its reliability was considered limited for this dataset.

In addition, crack assessment in practical applications is generally performed based on line-like extracted structures rather than pixel-wise regions. For this reason, clDice was adopted as the primary metric. Originally developed for thin and elongated structures such as roads and vessels, clDice incorporates the accuracy of centerline extraction and enables a more topology-aware evaluation of predicted crack structures than conventional pixel-based metrics.

Let V_p and V_L denote the sets of predicted and ground truth pixels, respectively, and S_p and S_L their corresponding skeletonized representations. Then, topological precision (Tprec) and topological sensitivity (Tsens) are defined as:

$$Tprec(S_P, V_L) = \frac{|S_P \cap V_L|}{|S_P|} \quad (3)$$

$$Tsens(S_L, V_P) = \frac{|S_L \cap V_P|}{|S_L|} \quad (4)$$

The clDice score is then computed as:

$$clDice(V_P, V_L) = 2 \times \frac{Tprec(S_P, V_L) \times Tsens(S_L, V_P)}{Tprec(S_P, V_L) + Tsens(S_L, V_P)} \quad (5)$$

In addition to segmentation accuracy, computational efficiency was evaluated using frames per second (FPS) and peak memory consumption. These metrics provide complementary indicators of inference speed and deployment feasibility, which are both critical in practical large-scale MMS-based inspection workflows. In operational settings, batch inference is commonly performed under a fixed hardware memory budget. From this perspective, the ratio of FPS to memory usage may be interpreted as a rough indicator of the throughput achievable within a given memory constraint. For inference-time efficiency evaluation, all models were benchmarked under identical hardware conditions with a batch size of 1.

5. Results and Discussion

5.1 Quantitative Evaluation

The quantitative results on the GAPs384 and CamCrack789 datasets are summarized in Figures 3 and 4 and in Tables 5 and 6, respectively. Overall, the proposed model demonstrates a consistently strong efficiency–accuracy trade-off across both datasets, achieving topology-aware segmentation performance comparable to strong existing methods while delivering better deployment efficiency.

On the GAPs384 dataset, the proposed model achieves a clDice score of 0.758, which is comparable to CT-CrackSeg (0.760) and slightly higher than DECS-Net (0.754), indicating competitive crack topology preservation. In addition, it requires only 355 MB of memory, the lowest among all compared methods, and runs at 4.44 FPS, which is substantially faster than CT-CrackSeg (1.49 FPS) and DECS-Net (1.55 FPS). Although DeepCrack attains the highest FPS, its lower clDice score suggests weaker preservation of fine crack structures. These results indicate that, on GAPs384, the proposed method provides a more efficient alternative to heavier high-performing models while maintaining comparable segmentation performance.

A similar tendency is observed on CamCrack789. On this dataset, the proposed model achieves a clDice score of 0.922, slightly higher than CT-CrackSeg (0.921) and close to the best-performing DECS-Net (0.925). In terms of efficiency, the proposed model again shows clear advantages, requiring less memory than DECS-Net (355 MB vs. 390 MB) and achieving a much higher inference speed (3.92 FPS vs. 1.31 FPS), while also improving over CT-CrackSeg in both speed and memory usage.

Compared with the other benchmark models, DeepCrack provides the highest inference speed but lower clDice on both datasets, whereas DCCM-Net and SCSegamba achieve reasonable segmentation performance at the cost of substantially higher memory use and lower throughput. DECS-Net remains a strong baseline in terms of segmentation accuracy, but its inference speed is clearly lower than that of the proposed model. Taken together, these results indicate that the proposed CNN-Mamba hybrid design offers a practically superior efficiency profile to the main baselines, while maintaining topology-aware crack segmentation quality at a competitive level. This makes the proposed method particularly attractive for deployment-oriented pavement inspection workflows using large-scale mobile mapping data.

Table 5 Experimental results (GAPs384)

Model	clDice ↑	Memory Usage(MB) ↓	FPS ↑
Hybrid-Segmenter	0.713	1168	2.16
EfficinetCrackNet	0.727	759	2.06
Deepcrack	0.742	458	**6.08**
Crackformer- II	0.731	585	1.36
SCSegamba	0.729	1845	0.62
DCCMNet	0.750	6379	0.30
DECS-Net	0.754	390	1.55
CT-CrackSeg	**0.760**	2827	1.49
Proposed	0.758	**355**	4.44

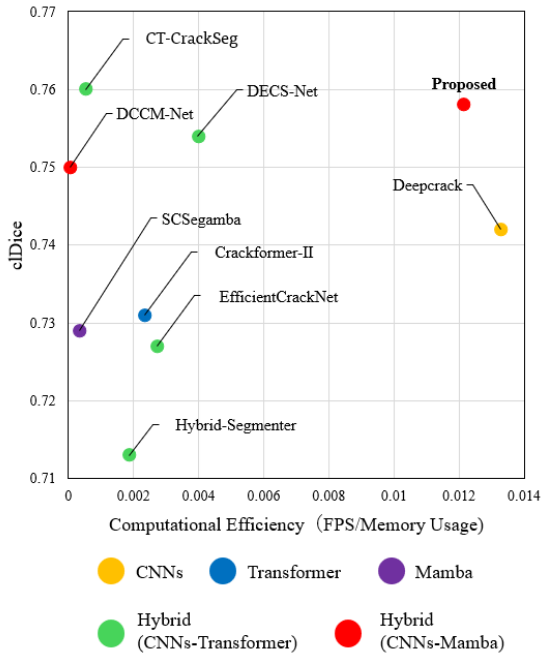

Figure 3 Accuracy–efficiency trade-off on GAPs384

Table 6 Experimental results (CamCrack789)

Model	clDice ↑	Memory Usage(MB) ↓	FPS ↑
DeepCrack	0.904	458	**5.83**
Crackformer-II	0.902	585	1.28
SCSegamba	0.909	1845	0.55
DCCM-Net	0.917	6379	0.26
DECS-Net	**0.925**	390	1.52
CT-CrackSeg	0.921	2827	1.31
Proposed	0.922	**355**	3.92

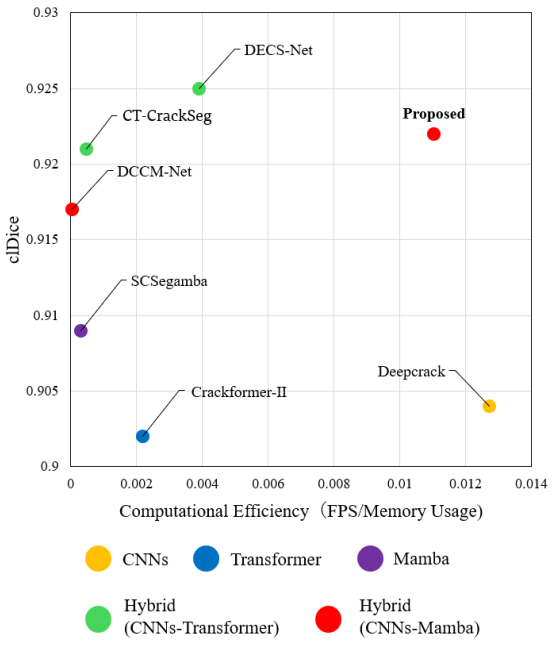

Figure 4 Accuracy–efficiency trade-off on CamCrack789

5.2 Ablation Study

To further analyse the effects of the two main architectural modifications, we conducted an ablation study on GAPs384 and CamCrack789 using three model variants: (1) the original CT-CrackSeg baseline, (2) CT-CrackSeg with the refined DCNv2-based boundary enhancement branch, and (3) the final model combining the DCNv2 refinement with the EfficientViM-inspired encoder redesign. These settings correspond directly to the two proposed modifications: the DCNv2-based refinement of the boundary enhancement branch and the replacement of the original MobileViT-based encoder blocks with EfficientViM-inspired HSM-SSD blocks. The results are summarized in Tables 7 and 8. As in the main evaluation, the ablation study uses topology-aware segmentation metrics, namely topological precision (Tpre), topological sensitivity (Tsens), and clDice, together with peak memory consumption and inference speed (FPS). The reported ± values denote standard deviations over three independent runs with different random seeds.

On GAPs384 (Table 7), the effects on segmentation accuracy are modest. Adding DCNv2 increases Tsens and slightly improves clDice, but decreases Tpre. When the EfficientViM-inspired encoder blocks are further introduced, peak memory consumption is reduced from 2827 MB to 355 MB and inference speed increases from 1.97 FPS to 4.24 FPS, while the changes in Tpre, Tsens, and clDice remain small. Given that many of these differences are comparable to the standard deviations over three runs, the GAPs384 results mainly indicate improved efficiency rather than a clear accuracy gain from each individual modification.

On CamCrack789 (Table 8), adding DCNv2 again slightly decreases Tpre while improving Tsens and clDice, suggesting that the refined boundary enhancement branch mainly enhances sensitivity to thin and elongated crack structures. When the EfficientViM-inspired encoder blocks are further introduced, the model achieves the major efficiency improvement, reducing peak memory consumption from 2827 MB to 355 MB and increasing FPS from 1.74 to 3.92 while maintaining a similar level of topology-aware accuracy.

Taken together, these results support a consistent interpretation across the two datasets. The proposed architecture does not derive its main value from a large accuracy gain produced by any single added component. Rather, the DCNv2-based refinement mainly contributes to sensitivity-oriented structural improvement, whereas the EfficientViM-inspired encoder redesign provides the dominant gain in computational efficiency. The main contribution of the proposed design therefore lies in achieving a more favourable balance between structural crack extraction quality and deployment efficiency while maintaining topology-aware performance at a level comparable to that of the original CT-CrackSeg baseline.

Table 7 Ablation study on GAPs384

Ablation	Tpre ↑	Tsens ↑	clDice ↑	Memory Usage (MB) ↓	FPS ↑
CT-Crackseg (Baseline)	**0.776**± 0.0106	0.744± 0.0177	0.759± 0.0050	2827	1.49
+DCNv2	0.758± 0.0104	**0.766**± 0.0194	**0.762**± 0.0115	2827	1.97
+EfficientViM inspired block	0.767± 0.0211	0.750± 0.0228	0.758± 0.0079	**355**	**4.24**

Table 8 Ablation study on CamCrack789

Ablation	Tpre ↑	Tsens ↑	clDice ↑	Memory Usage (MB) ↓	FPS ↑
CT-Crackseg (Baseline)	**0.922**± 0.0031	0.920± 0.0099	0.921± 0.0036	2827	1.31
+DCNv2	0.916± 0.0093	0.931± 0.0070	**0.923**± 0.0016	2827	1.74
+EfficientViM inspired block	0.909± 0.0053	**0.935**± 0.0059	0.922± 0.0004	**355**	**3.92**

5.3 Qualitative Evaluation and Discussion

As shown in the qualitative results in Figure 5, all evaluated models were generally able to detect crack regions with reasonable visual accuracy, although each model exhibited characteristic tendencies. DeepCrack occasionally produced fragmented predictions, particularly for thin or low-contrast cracks, resulting in reduced crack continuity. In contrast, the Transformer-based and Mamba-based models sometimes generated false positives by misclassifying non-crack structures or background noise as cracks, as highlighted by the red boxes in Figure 5. Meanwhile, the CNN–Transformer hybrid and CNN–Mamba hybrid models showed relatively stable behaviour, with improved crack continuity and fewer obvious false detections. These observations suggest that combining convolution-based local feature extraction with global context modelling is effective for crack segmentation.

From the viewpoint of structural segmentation behaviour, the results further indicate that the proposed model preserves the integrity of crack patterns well despite a substantial reduction in computational cost. The small difference in clDice relative to CT-CrackSeg suggests that the topological consistency of the predicted crack network is largely maintained, which is particularly important for thin and elongated structures such as pavement cracks. In other words, the proposed lightweight redesign not only reduces computational cost but also retains the ability to reconstruct connected crack patterns in a structurally consistent manner.

This balance is particularly meaningful from an operational perspective. In large-scale MMS-based pavement inspection, practical deployment is constrained not only by segmentation quality but also by hardware requirements, memory usage, and processing time. In this context, the reduction in peak memory consumption and the improvement in inference speed indicate that the proposed model is better suited to resource-constrained and large-scale processing scenarios than the original CT-CrackSeg baseline.

At the same time, the practical efficiency of Mamba-based models is influenced not only by their theoretical linear complexity but also by how efficiently their sequence modules can be implemented in a given deployment environment. In our experiments, all methods were benchmarked in a unified Windows-based PyTorch setting, where some Mamba-based baselines required pure PyTorch replacements for their original optimized kernels. Accordingly, the reported FPS values should not be interpreted as architecture-only upper-bound performance. Rather, they should be regarded as empirical throughput under a realistic and reproducible deployment condition. This distinction is important in pavement inspection workflows, because actual deployability depends not only on model design but also on implementation accessibility, environment compatibility, and memory constraints.

Nevertheless, because all methods were evaluated under the same practical execution setting, the comparison still provides meaningful evidence regarding relative deployability. From this perspective, the proposed HSM-SSD-based redesign demonstrates a favourable efficiency-accuracy trade-off without relying solely on highly optimized platform-specific runtime support. More broadly, the experiments suggest that, among recent high-performing crack segmentation models, differences in segmentation quality on standard benchmarks are often relatively small. Under such conditions, practical applicability becomes increasingly dependent on computational efficiency rather than on marginal gains in accuracy alone. This is particularly relevant in large-scale inspection scenarios using MMS or related photogrammetric platforms, where throughput and memory consumption directly affect deployability.

Overall, the proposed model demonstrates that computational efficiency can be substantially improved without sacrificing the practical usefulness of crack segmentation results. By redesigning CT-CrackSeg with EfficientViM-based global modelling and a refined boundary enhancement module, the proposed architecture achieves a favourable balance between computational cost and structural segmentation quality, making it a practical option for large-scale photogrammetric and infrastructure-monitoring applications.

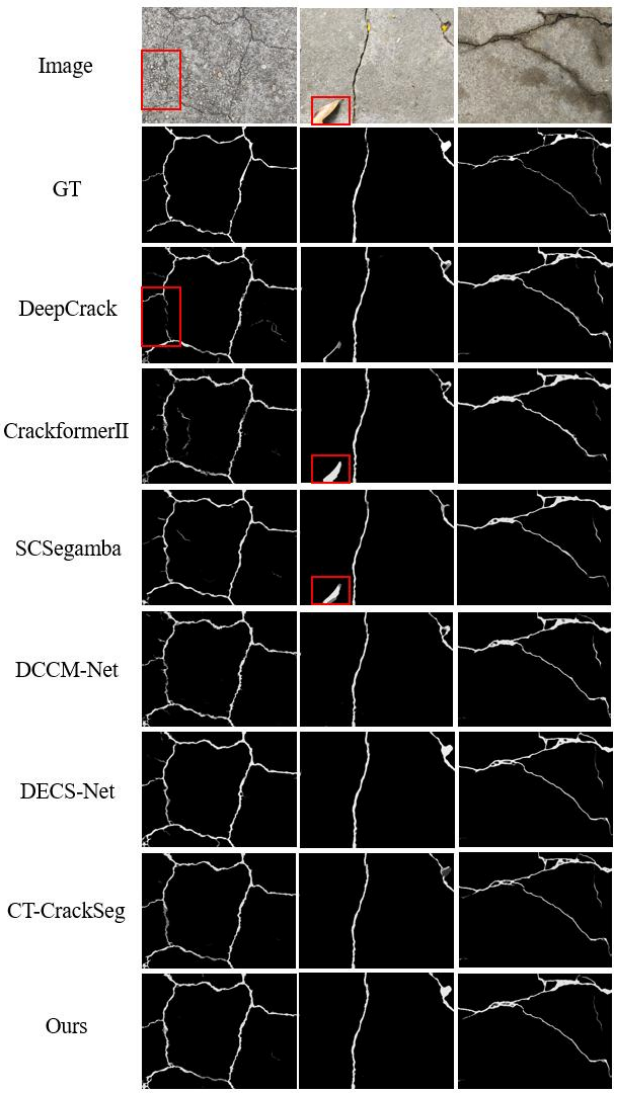

Figure 5 Qualitative segmentation results of all models

6. Conclusions

This study presented a lightweight CNN-Mamba hybrid architecture for pavement crack segmentation as a deployment-oriented redesign of CT-CrackSeg. By replacing the original MobileViT-based global modelling modules with EfficientViM-inspired HSM-SSD blocks and refining the boundary enhancement branch with DCNv2-based deformable convolution, the proposed method improves computational efficiency while preserving the overall encoder-decoder design.

Experiments on the GAPs384 and CamCrack789 datasets showed that the proposed model maintains competitive topology-aware segmentation quality, achieving clDice scores comparable to CT-CrackSeg, while substantially reducing peak memory consumption and increasing inference speed. Peak memory usage was reduced from 2827 MB to 355 MB, and inference speed improved from 1.49 to 4.44 FPS on GAPs384 and from 1.32 to 3.92 FPS on CamCrack789. These results indicate that the proposed redesign achieves a better balance between segmentation quality and deployment efficiency than baseline.

The significance of this study is not the introduction of a fundamentally new crack segmentation paradigm, but the demonstration that the global modelling component of a high-performing hybrid architecture can be redesigned for greater deployment efficiency without materially degrading topology-aware crack extraction quality. This is especially relevant for large-scale MMS-based pavement inspection and related photogrammetric infrastructure monitoring workflows, where memory usage, throughput, and implementation practicality directly affect deployability.

Future work includes validation on larger and more diverse datasets, evaluation under additional hardware and runtime environments, and further optimization for real-time or edge-assisted inspection scenarios.

References

Alexandre, T. L., 2024. mamba.py: A simple, hackable and efficient Mamba implementation in pure PyTorch and MLX. https://github.com/alxndrTL/mamba.py, (accessed 2026.3.30).

Choi, W., Cha, Y. J., 2019. SDDNet: Real-time crack segmentation. *IEEE Transactions on Industrial Electronics*, 67(9), 8016-8025.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y., 2017. Deformable convolutional networks. *Proc. of IEEE international conference on computer vision*, 764-773.

Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C., 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35, 16344-16359.

Dao, T., Gu, A., 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv:2405.21060*.

Du Nguyen, Q. and Thai, H. T., 2023. Crack segmentation of imbalanced data: The role of loss functions. *Engineering Structures*, 297, 116988.

Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Gross, H. M., 2017. How to get pavement

distress detection ready for deep learning? A systematic approach. *International joint conference on neural networks*, 2039-2047.

Gong, H., Liu, L., Liang, H., Zhou, Y. and Cong, L., 2024. A state-of-the-art survey of deep learning models for automated pavement crack segmentation. *International Journal of Transportation Science and Technology*, 13, 44-57.

Goo, J. M., Milidonis, X., Artusi, A., Boehm, J. and Ciliberto, C., 2025. Hybrid-Segmentor: Hybrid approach for automated fine-grained crack segmentation in civil infrastructure. *Automation in Construction*, 170, 105960.

Gu, A., and Dao, T., 2024. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*.

Guo, F., Qian, Y., Liu, J., Yu, H., 2023. Pavement crack detection based on transformer network. *Automation in Construction*, 145, 104646.

Han, D., Wang, Z., Xia, Z., Han, Y., Pu, Y., Ge, C., Song, J., Song, S., Zheng, B., Huang, G., 2024. Demystify mamba in vision: A linear attention perspective. *Advances in neural information processing systems*, 37, 127181-127203.

Jing, J., Yang, X., Cheng, H., Feng, X., Zheng, H., Brilakis, I., Liu, J., 2025. A critical review on pavement distress detection using images and point clouds from visual features to geometric modeling. *Journal of Road Engineering*. 5(4), 583-606.

Katharopoulos, A., Vyas, A., Pappas, N., Fleuret, F., 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. *International conference on machine learning*, 5156-5165.

Lee, S., Choi, J., Kim, H. J., 2025. Efficientvim: Efficient vision mamba with hidden state mixer based state space duality. *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14923-14933.

Liu, H., Jia, C., Shi, F., Cheng, X., Chen, S., 2025. SCSegamba: Lightweight structure-aware vision mamba for crack segmentation in structures. *Proc. of the Computer Vision and Pattern Recognition Conference*, 29406-29416.

Liu, H., Miao, X., Mertz, C., Xu, C., Kong, H., 2021. Crackformer: Transformer network for fine-grained crack detection. *Proc. of the IEEE/CVF international conference on computer vision*, 3783-3792.

Liu, H., Yang, J., Miao, X., Mertz, C. and Kong, H., 2023. CrackFormer network for pavement crack segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 24(9), 9240-9252.

Liu, J., Yang, X., Lau, S., Wang, X., Luo, S., Lee, V. C. S., Ding, L., 2020. Automated pavement crack detection and segmentation based on two-step convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, 35(11), 1291-1305.

Mehta, S., Rastegari, M., 2021. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer, *arXiv:2110.02178*.

Ministry of Land, Infrastructure, Transport and Tourism., 2025. White Paper on Land, Infrastructure, Transport and Tourism.

Shit, S., Paetzold, J. C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylka, A., Pluim, J. P. W., Baur, U., and Menze, B. H., 2021. clDice-a novel topology-preserving loss function for tubular structure segmentation, *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16560-16569.

Tao, H., Liu, B., Cui, J., Zhang, H., 2023. A convolutional-transformer network for crack segmentation with boundary awareness, *Proc. of IEEE international conference on image processing*, pp.86-90.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yoshimura, M., Hayashi, T., Hoshino, Y., Wang, W. Y., Ohashi, T., 2026. SF-Mamba: Rethinking State Space Model for Vision. *arXiv:2603.16423*.

Zhang, A. A., Shang, J., Li, B., Hui, B., Gong, H., Li, L., Zhan, Y., Ai, C., Niu, H., Chu, X., Nie, Z., Dong, Z., He, A., Zhang, H., Wang, D., Peng, Y., Wei, Y., Cheng, H., 2024. Intelligent pavement condition survey: Overview of current researches and practices. *Journal of road engineering*, 4(3), 257-281.

Zhang, J., Li, D., Zeng, Z., Zhang, R., Wang, J. 2025. Dual-branch crack segmentation network with multi-shape kernel based on convolutional neural network and Mamba. *Engineering Applications of Artificial Intelligence*, 150, 110536.

Zhang, J., Zhang, S., Li, D., Wang, J., Wang, J., 2025. Crack segmentation network via difference convolution-based encoder and hybrid CNN-Mamba multi-scale attention. *Pattern Recognition*, 167, 111723.

Zhang, J., Zeng, Z., Sharma, P. K., Alfarraj, O., Tolba, A. Wang, J., 2024. A dual encoder crack segmentation network with Haar wavelet-based high–low frequency attention. *Expert Systems with Applications*, 256, 124950.

Zhao, Z., Ding, Z., Niu, P., Sun, W., Guo, F., 2026. MixerCSeg: An Efficient Mixer Architecture for Crack Segmentation via Decoupled Mamba Attention. *arXiv:2603.01361.*

Zhu, G., Liu, J., Fan, Z., Yuan, D., Ma, P., Wang, M., Wang, K. C., 2024. A lightweight encoder–decoder network for automatic pavement crack detection. *Computer-Aided Civil and Infrastructure Engineering*, 39(12), 1743-1765.

Zhu, X., Hu, H., Lin, S., Dai, J., 2019. Deformable convnets v2: More deformable, better results. *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 9308-9316.

Zim, A. H., Iqbal, A., Al-Huda, Z., Malik, A. and Kuribayashi, M., 2025. EfficientCrackNet: A Lightweight Model for Crack Segmentation. *Proc. of IEEE/CVF Winter Conference on Applications of Computer Vision*, 6279-6289.

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., Wang, S., 2018. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3), 1498–1512.

Zuo, X., Sheng, Y., Shen, J., Shan, Y., 2024. Topology-aware mamba for crack segmentation in structures. *Automation in Construction*, 168, 105845.