Self-Supervised Multi-Scale Transformer with Attention-Guided Fusion for Efficient Crack Detection

Abstract

This study presents *Crack-Segmenter*, a fully self-supervised segmentation framework for crack detection. Unlike existing supervised methods requiring expensive pixel-level annotations, our approach eliminates this dependency completely through three integrated modules: Scale-Adaptive Embedder (SAE), Directional Attention Transformer (DAT), and Attention-Guided Fusion (AGF). These modules work together seamlessly, with SAE capturing multi-scale crack features, DAT enhancing linear crack continuity through directional attention, and AGF adaptively integrating these representations into unified segmentation outputs. Extensive experiments on ten public crack datasets demonstrated that *Crack-Segmenter* consistently outperformed 13 state-of-the-art supervised methods across all evaluation metrics, including mean Intersection over Union (mIoU), Dice score, XOR, and Hamming Distance (HM). These results demonstrate that annotation-free crack segmentation can achieve superior performance while enabling scalable infrastructure monitoring and automated maintenance decision-making, advancing the state of self-supervised learning in infrastructure applications.

Keywords: Self-supervised segmentation, Crack detection, Multi-scale transformer, Directional attention, Infrastructure monitoring, Annotation-free learning, Road maintenance automation

1. Introduction

Transportation infrastructure, particularly road networks, is critical for public safety and economic development. Roads connect communities, facilitate commerce, and ensure efficient movement of people and goods. However, the constant exposure of roads to traffic loads and weather conditions gradually weakens their structural integrity, often resulting in surface cracks. Even tiny cracks that develop on these roads can quickly grow into severe defects such as potholes or large pavement failures if they are not detected and repaired early. This makes preventive maintenance very vital in prolonging the lifespan of pavements. For instance, research indicates that preventive maintenance on small pavement cracks can reduce future repair costs by approximately 50–70%, highlighting the financial benefit of timely intervention [1, 2]. Accurate, pixel-level crack detection maps are thus essential for enabling early and low-cost pavement maintenance. These detailed maps can guide agencies in prioritizing repairs, allocating budgets efficiently, and reducing overall maintenance expenses. To generate such maps automatically, researchers have developed supervised learning approaches that have achieved strong performance in diverse pavement crack segmentation tasks.

Several fully-supervised segmentation models for pavement crack segmentation have been proposed. For instance, Lau et al. [3] replace the U-Net encoder with a pretrained ResNet-34 and report F1 scores of 96 % on CFD and 73 % on Crack500, proving that full-mask supervision can yield strong accuracy. [4] proposed Context-CrackNet, a supervised model using global and local attention modules to accurately segment tiny and large pavement cracks. [5] also developed PDSNet, a supervised deep learning framework for segmenting multiple asphalt pavement distresses using 2D and 3D images, achieving a high mean Intersection over Union (MIoU) of 83.7%. All these approaches required extensive annotations during the model training process. Especially, [5] had to manually annotate 5000 images of pavement cracks by drawing pixel-level labels which is labor-intensive. Given the practical challenges and costs of extensive pixel-level annotations in fully supervised approaches, there is growing interest in alternative learning methods that reduce annotation demands while preserving segmentation performance.

Weakly supervised and semi supervised segmentation methods have emerged as promising alternatives, significantly reducing annotation efforts while aiming to retain segmentation accuracy. For example, Xiang et al. [6] introduced UWSCS, a crack segmentation framework that uses limited coarse labels alongside superpixel and shrink-based correction modules to train a dual encoder network. Similarly, recent studies have integrated the Segment Anything Model (SAM) with interactive segmentation using bounding box prompts and deep transfer learning to enable semi supervised crack detection [7, 8]. However, these approaches still rely on manual bounding box annotations, which remain costly and time consuming, particularly for large scale pavement crack datasets. To further reduce annotation dependency, some recent frameworks have incorporated advanced learning strategies such as adversarial training [9, 10, 11], student teacher learning [12, 13, 14], and graph based modeling using graph convolutional networks [15, 16]. These techniques are often used within weakly or semi supervised architectures to enhance learning from limited annotations, but they still depend on some form of manual supervision during training. Although several recent studies describe their approaches as fully self supervised, many of them still rely on ground truth annotations at some point in the pipeline, such as during pre-training, pseudo label generation, or model calibration [17, 18]. As a result, they do not fully meet the criteria of annotation-free segmentation. These continued limitations highlight the need for a robust, efficient, and scalable self supervised framework that can accurately segment pavement cracks without relying on any annotated data during training and still achieve superior performance compared to current supervised and semi-supervised methods.

To address the limitations discussed above, we propose an efficient, fully self-supervised segmentation framework designed specifically for pavement crack segmentation. Unlike previous approaches, our method requires no manual annotations or ground truth pixel labels, thus significantly reducing the cost and time associated with labeling. Our proposed architecture consists of three main modules: the Scale-Adaptive Embedder, the Directed Multi-Branch Transformer, and the Attention-Guided Fusion module. The Scale-Adaptive Embedder transforms input images into token representations at three distinct spatial scales, capturing features ranging from tiny hairline cracks to wider pavement defects. Building upon these multi-scale embeddings, our Directed Multi-Branch Transformer applies directional efficient attention mechanisms, effectively preserving linear crack structures and capturing essential spatial relationships without extensive computational costs. To seamlessly merge these multi-scale representations, our Attention-Guided Fusion module adaptively weights features from different scales, ensuring that both fine-grained details and broader contextual information are combined effectively. The introduction of cross-scale consistency losses [19] further enhances the model's ability to learn robust representations without manual labels. The main contributions of our work have been summarized below:

- We introduce *Crack-Segmenter*, an end-to-end self-supervised framework for pavement-crack segmentation that requires no pixel-level or weak annotations, thereby reducing annotation costs significantly.
- We design three modules: the *Scale-Adaptive Embedder* (SAE) for

multi-resolution feature extraction, the *Directional Attention Transformer* (DAT) to preserve elongated crack geometry, and the *Attention-Guided Fusion* (AGF) to adaptively merge scale-specific representations.

- We develop inter-scale and intra-scale consistency losses to enhance coherent feature representations, substantially improving model learning without manual supervision.

- We evaluated *Crack-Segmenter* on ten public crack datasets against 13 state-of-the-art fully supervised models. It outperformed every baseline with statistically significant gains.

2. Related Works

2.1. Fully-Supervised Crack Segmentation Methods

Fully supervised methods have utilized rich pixel-level annotations to develop specialized models for accurate pavement crack segmentation. For example, Dung et al. [20] adopted a Fully Convolutional Network (FCN) based on a VGG16 encoder for automated concrete crack detection, where high-level features were upsampled via deconvolution to produce crack masks. While this approach proved the feasibility of end-to-end crack segmentation, the basic FCN architecture struggled with limited context and spatial consistency, often missing fine crack details. Researchers subsequently improved crack segmentation by integrating multi-scale feature fusion and boosting techniques. Yang et al. [21] introduced a Feature Pyramid and Hierarchical Boosting Network (FPHBN) that fuses semantic information from deep and shallow layers and utilizes a feature pyramid to enhance detection of cracks at different scales. This method improved the generalization to various crack widths and backgrounds by considering multi-level features simultaneously. Similarly, Liu et al. [22] proposed a deeply supervised encoder–decoder network (DeepCrack) which aggregates multi-scale features from multiple network layers, capturing both fine and coarse crack patterns for more robust segmentation. These multi-scale approaches demonstrated higher accuracy than plain FCNs, but the heavy pooling in their backbones could still cause some loss of fine spatial information. To better preserve crack locality, later studies adopted encoder–decoder architectures like U-Net. Jenkins et al. [23] showed that a vanilla U-Net can effectively segment road cracks at the pixel level, and further improvements added attention gating to suppress irrelevant background features. Building on this idea, Pan et al. [24] developed an attention-enhanced U-Net variant called SCHNet, which incorporates parallel spatial, channel, and feature-pyramid attention modules into a VGG19-based network. Transformer-based architectures have also been explored: Guo et al. [25] proposed a Crack Transformer (CT) model using a Swin Transformer encoder and all-MLP decoder, showing robust performance in detecting long, complex cracks even under noisy conditions. Similarly, [26] embedded a Transformer encoder within a U-shaped CNN in their CrackFormer network, substantially improving segmentation continuity and accuracy for thin cracks. However, these supervised approaches are heavily dependent on pixel-level annotations, which are time-consuming, costly, and practically difficult to obtain at a large scale [27].

2.2. Weakly-Supervised and Semi-Supervised Crack Segmentation

To alleviate the burden of dense labeling, researchers have developed weakly supervised frameworks that learn from coarse annotations (e.g. image-level labels). Al-Huda et al. [28] proposed a two-stage weakly supervised method based on class activation mapping and iterative refinement: a crack classification network first produces initial pixel indications of cracks at multiple scales, then a U-Net with an attention mechanism is trained on these noisy masks and incrementally fine-tunes the predictions. He et al. [29] similarly employed image-level labels to drive crack segmentation by using a generative adversarial localization strategy. They trained a U-GAT-IT model to generate class activation maps of cracks, iteratively erased detected regions to discover new crack areas, and then converted these refined maps into pseudo-labels to train a segmentation network. Semi-supervised methods have emerged to exploit unlabeled roadway images alongside limited labeled data, further reducing the need for annotations. Shim et al. [30] pioneered an adversarial learning-based semi-supervised segmentation approach for concrete crack detection. Their model employed multiscale feature extractors and a generative adversarial network to train on labeled and unlabeled data simultaneously. Building on consistency-driven learning, Shi et al. [31] developed a crack segmentation model that enforces mutual consistency constraints between dual network predictions and incorporates a boundary-aware loss. Another innovative strategy is the two-stage CrackDiffusion framework proposed by Han et al. [32], which combines an unsupervised anomaly detection stage with a supervised refinement stage. In the first stage, a diffusionbased inpainting model removes cracks from images to generate crack-free counterparts and uses the differences (via structural similarity measures) to localize cracks without labels. In the second stage, those initial crack maps inform a U-Net segmentation model that learns to produce precise crack masks. Despite these advancements, weakly and semi-supervised segmentation methods still depend on pseudo-labels or sparse annotations. This requirement makes it impractical to annotate large-scale, high-resolution pavement crack segmentation datasets which has completely no pixel-level labels. Because of this limitation, researchers have turned their attention toward fully self-supervised segmentation methods. These fully self-supervised approaches aim to eliminate the need for ground-truth pixel-level annotations entirely during the training process. However, research in the area of fully self-supervised pavement distress segmentation remains limited and underexplored. Moreover, the few existing pavement crack segmentation techniques that claim to be fully self-supervised still suffer from significant limitations and drawbacks.

2.3. Limitations

Self-supervised learning (SSL) is a representation learning paradigm in which the supervisory signal is derived automatically from the structure of the unlabelled data itself, rather than from externally provided human annotations [33]. In the context of semantic segmentation, self-supervised segmentation extends this idea from learning image-level representations to learning pixel-level assignments without any human-drawn masks. The model first designs an intrinsic task such as grouping pixels with similar colour statistics, enforcing consistency between differently augmented views, and then treats the resulting pseudo-labels as ground truth for training [34, 35]. Because these labels arise automatically from the data, the pipeline needs no external annotations at any stage. SSL has been applied to pavement and other civil-infrastructure images through tasks such as self-training for cracks [36, 37], pavement-surface anomaly detection [38, 39], sidewalk-quality classification [40]. Some studies have attempted to apply SSL in segmentation of pavement cracks. For example, [17] propose SS-YOLO, a YOLOv8-based crack segmentation model that fuses CBAM and Gaussian multi-head self-attention with curriculum learning–driven pseudo-labeling. However, SS-YOLO still starts from a fully supervised YOLOv8 backbone trained on real crack masks before it ever “self-labels” unannotated data, so it isn’t end-to-end self-supervised. Similarly, Zhang et al. [18] proposed a dual cycle-GAN that learns to translate crack image patches into GT-like structure patterns (and back) using an unpaired "structure library" of binary skeletons. Conversely, because that library is built from pixel-precise, human-annotated curves (e.g., VOC object-boundary masks, Berkeley contour annotations, and public crack GTs), the method still depends on existing pixel-level labels and isn't truly end-to-end self-supervised. Song et al. [41] also proposed a two-stage pavement-crack framework: an improved U-Net (U-Net augmented with residual blocks and attention gates) is first contrastively pre-trained on unlabeled crack (background) patches, then fully fine-tuned with pixel-level ground-truth masks during the second stage. Because the network does not directly learn the image-to-mask mapping until this second stage where every gradient is computed against human-annotated labels, the approach is merely semi-supervised, not an end-to-end fully self-supervised segmentation method. Ma et al. [42] introduced UP-CrackNet, pavement-crack detector that trains a conditional GAN to inpaint randomly masked regions of crack-free road images. At inference, cracks are segmented by thresholding the pixel-wise reconstruction residuals, allowing annotation-free training. One problem with UP-CrackNet is that since it never sees real crack patterns during optimization, it must treat every unfamiliar texture as a defect. This makes the network yield a high error map and treats labels artifacts present in the crack image (e.g. leaf, tyre mark) as a crack, inflating false positives that a crack-aware model could reject.

3. Methodology

3.1. Problem Structure and Overview

Pavement binary crack segmentation involves accurately classifying every pixel in an image as either crack or non-crack. Currently, this task relies heavily on extensive ground truth pixel-level annotations (masks). That is, given a batch of input pavement images $I \in \mathbb{R}^{B\times H\times W\times C}$ where B, H, W, and C denote the batch size, height, width, and number of channels respectively, the goal of this task is to predict their corresponding binary segmentation maps $S \in \{0,1\}^{B\times H\times W\times 1}$, where each pixel is assigned a value of 1 if it belongs to a crack and 0 otherwise. This supervised learning approach has achieved significant success in binary crack segmentation. However, obtaining the precise ground truth annotations for supervised learning is both costly and impractical for large-scale crack datasets.

To address this limitation, we propose a fully self-supervised framework that completely eliminates the need for ground truth labels or masks by learning robust feature representations directly from the input images. Our framework utilizes multi-scale feature extraction, directional attention mechanisms, and adaptive scale fusion to identify and segment pavement cracks accurately. The challenge here is formulating a learning paradigm that can effectively differentiate crack pixels from non-crack pixels without explicit supervision, thus making it scalable and cost-efficient.

3.2. Overall Framework

The proposed self-supervised segmentation framework comprises of three primary modules designed to collectively address the aforementioned challenge: the **Scale Adaptive Embedder** (Φ_{SAE}), **Directional Attention Transformer** (f_{DAT}), and **Attention-Guided Fusion** (f_{AGF}). The overall framework for our proposed architecture has been shown in Figure 1.

The proposed framework begins with the Scale Adaptive Embedder module, which processes a batch of input images I simultaneously at multiple scales: fine, small, and large. Specifically, given a batch of input image I, the module produces three distinct feature embeddings with embedding dimension D:

$$\Phi_{\text{SAE}}(I) = \{F_f, F_s, F_l\}, \tag{1}$$

where $F_f, F_s \in \mathbb{R}^{B\times D\times H\times W}$ and $F_l \in \mathbb{R}^{B\times D\times \frac{H}{2}\times \frac{W}{2}}$. F_f, F_s, F_l correspond to the fine, small, and large scale feature embeddings respectively. This multi-scale representation ensures comprehensive capture of cracks of varying widths and complexities, enhancing sensitivity to fine details and broad spatial context simultaneously.

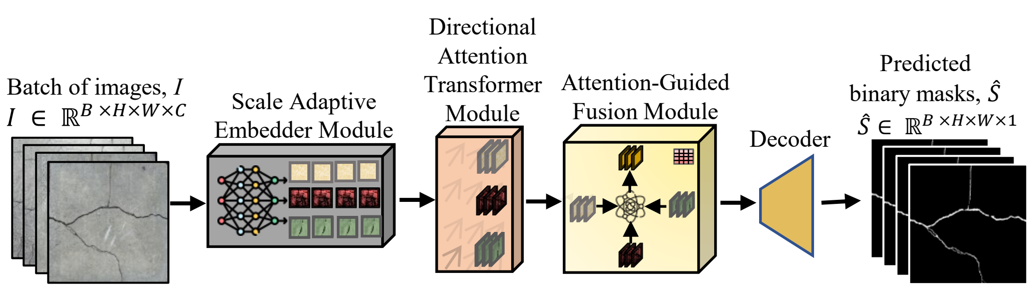

Figure 1: Overall framework for our proposed architecture.

Next, each scale-specific feature embedding from the Scale Adaptive Embedder undergoes further refinement through the Directional Attention Transformer module. This transformer applies efficient attention with directed convolutions to maintain and emphasize linear crack structures within feature maps, essential for accurate pavement crack identification. Formally, this transformation is represented as:

$$F'_f = f_{\mathrm{DAT}}(F_f) \tag{2}$$

$$F'_s = f_{\mathrm{DAT}}(F_s) \tag{3}$$

$$F'_l = f_{\mathrm{DAT}}(F_l) \tag{4}$$

where $F'_f, F'_s \in \mathbb{R}^{B\times D\times H\times W}$ and $F'_l \in \mathbb{R}^{B\times D\times \frac{H}{2}\times \frac{W}{2}}$. F'_f, F'_s, F'_l represent the refined feature embeddings for the different scales after passing through the Directional Attention Transformer module. This module enhances local contextual consistency within each scale, preserving crucial spatial relationships pertinent to cracks.

Finally, the Attention-Guided Fusion module integrates these refined scale-specific features from the Directional Attention Transformer module using attention-based adaptive weighting to form a unified, robust representation. Specifically, the large-scale feature map F'_l is upsampled and projected to match the spatial dimensions of the other scales. Subsequently, the fusion module computes attention weights and merges the scales:

$$F_{\text{fused}} = f_{\mathrm{AGF}}\big(F'_f,\ F'_s,\ \text{upsample}(F'_l)\big), \tag{5}$$

where $F_{\text{fused}} \in \mathbb{R}^{D\times H\times W}$ and *upsample* represents the upsampling operation. This adaptive fusion ensures optimal integration of both detailed crack structures and broader contextual information.

Finally, the fused feature representation F_{fused} is processed through a linear decoding layer to produce the final segmentation map prediction $\hat{S}$:

$$\hat{S} = f_{\text{decode}}(F_{\text{fused}}), \tag{6}$$

where f_{decode} is the decoding layer and $\hat{S} \in \mathbb{R}^{B\times H\times W\times 1}$ represents the predicted segmentation maps.

Through cross-scale consistency losses, specifically inter-scale and intra-scale self-supervised losses, our model learns to produce consistent and accurate segmentation predictions without any ground truth annotations (masks).

Thus, the proposed framework efficiently addresses pavement crack segmentation challenges, significantly reducing annotation costs while maintaining high segmentation performance. The next section goes into details on the modules used.

3.3. Scale Adaptive Embedder

The **Scale Adaptive Embedder (SAE)** module is designed to effectively capture multi-scale spatial information from pavement images, crucial for accurately identifying cracks of varying sizes and complexities. This module embeds the input image into feature representations at fine, small, and large spatial scales, enhancing the model's ability to detect both detailed and broad crack structures simultaneously. Figure 2 shows the architecture of the Scale Adaptive Embedder.

Given a batch of input pavement images $I \in \mathbb{R}^{B\times C\times H\times W}$, the SAE module first applies convolutional projection operations at three distinct scales to produce feature maps tailored for fine, small, and large-scale analyses.

For the fine-scale embedding, the convolution operation is mathematically defined as:

$$F_f = \sigma\big(W_f * I + b_f\big), \quad F_f \in \mathbb{R}^{B\times D\times H\times W}, \tag{7}$$

where $W_f \in \mathbb{R}^{D\times C\times 1\times 1}$ represents the convolutional kernel weights, $b_f \in \mathbb{R}^D$ are biases, $*$ denotes the convolution operation, and σ is a nonlinear activation function. This fine-scale operation ensures the capture of detailed and fine-grained crack patterns.

For the small-scale embedding, we similarly define the convolution operation as:

$$F_s = \sigma\big(W_s * I + b_s\big), \quad F_s \in \mathbb{R}^{B\times D\times H\times W}, \tag{8}$$

where $W_s \in \mathbb{R}^{D\times C\times 3\times 3}$ with appropriate padding and stride set to 1 to maintain spatial dimensions. This operation captures crack structures at intermediate spatial resolutions, preserving local contextual relationships.

The large-scale embedding convolution operation is expressed as:

$$F_l = \sigma\big(W_l * I + b_l\big), \quad F_l \in \mathbb{R}^{B\times D\times \frac{H}{2}\times \frac{W}{2}}, \tag{9}$$

where $W_l \in \mathbb{R}^{D\times C\times 3\times 3}$ with stride set to 2 and appropriate padding to reduce spatial dimensions. This large-scale embedding helps in capturing broader spatial contexts and large-scale pavement defects.

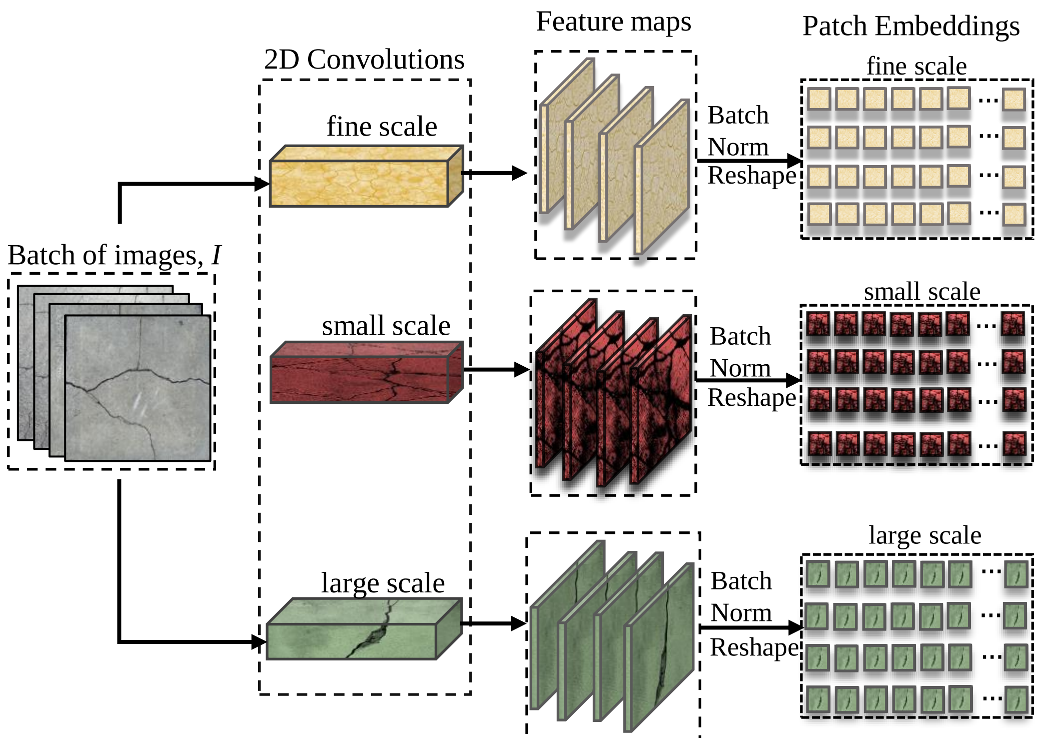

Figure 2: Scale Adaptive Embedder Module.

After convolutional projections, each feature map undergoes batch normalization to stabilize learning and enhance convergence:

$$F'_f = \mathrm{BatchNorm}(F_f) \tag{10}$$

$$F'_s = \mathrm{BatchNorm}(F_s) \tag{11}$$

$$F'_l = \mathrm{BatchNorm}(F_l) \tag{12}$$

where the Batch Normalization operation $\mathrm{BatchNorm}(\cdot)$ standardizes feature maps across each batch, thus improving training stability.

Finally, these normalized feature maps are reshaped and transposed to form sequences of patch embeddings compatible as input to the Directional Attention Transformer. Mathematically, the reshaping and transposition operation is defined as:

$$F''_{\text{scale}} \in \mathbb{R}^{B\times(H_{\text{scale}}W_{\text{scale}})\times D}, \tag{13}$$

where scale denotes the specific scale (fine, small, or large), and $H_{\text{scale}}, W_{\text{scale}}$ represent the corresponding spatial dimensions of each scale-specific embedding.

This multi-scale adaptive embedding strategy addresses the segmentation challenge of capturing both narrow, hairline cracks and wider crack formations within pavement images. By employing distinct yet complementary scale-specific convolutions, the SAE module ensures robust feature extraction at multiple resolutions, effectively supporting downstream modules for precise segmentation without manual annotations. In the next section, we will talk about the Directional Attention Module.

3.4. Directional Attention Transformer

The **Directional Attention Transformer (DAT)** module is designed to explicitly model directional spatial relationships, significantly enhancing the detection and segmentation of elongated, linear crack structures in pavement images. This module integrates multi-scale embeddings from the Scale Adaptive Embedder (SAE). It refines feature representations using spatially-directed attention mechanisms that effectively distinguish critical crack features from background noise. The architecture for the DAT module has been shown in Figure 3

The DAT module takes as input the multi-scale embeddings from the SAE module, denoted as

$$F''_{\text{scale}} \in \mathbb{R}^{B\times(H_{\text{scale}}W_{\text{scale}})\times D},$$

where scale $\in \{f, s, l\}$ corresponds to fine, small, and large scales respectively, B is the batch size, and D is the embedding dimension. These embeddings are first normalized via Layer Normalization to stabilize and improve training:

$$\hat{F}_{\text{scale}} = \text{LayerNorm}\big(F''_{\text{scale}}\big), \quad \hat{F}_{\text{scale}} \in \mathbb{R}^{B\times(H_{\text{scale}}W_{\text{scale}})\times D}. \tag{14}$$

Subsequently, the normalized embeddings are reshaped back to their spatial dimensions:

$$\hat{F}^{\text{spatial}}_{\text{scale}} \in \mathbb{R}^{B\times D\times H_{\text{scale}}\times W_{\text{scale}}}. \tag{15}$$

Directional convolutions are then applied to capture elongated structural patterns essential for crack segmentation. We define these convolutions as a function g, parameterized by kernels W_k and biases b_k. Specifically, for each direction k (e.g., horizontal $(1, 3)$, vertical $(3, 1)$), we compute:

$$Q_k = g\big(\hat{F}^{\text{spatial}}_{\text{scale}}; W_{Q_k}, b_{Q_k}\big), \quad Q_k \in \mathbb{R}^{B\times D\times H_{\text{scale}}\times W_{\text{scale}}}, \tag{16}$$

$$K_k = g\big(\hat{F}^{\text{spatial}}_{\text{scale}}; W_{K_k}, b_{K_k}\big), \quad K_k \in \mathbb{R}^{B\times D\times H_{\text{scale}}\times W_{\text{scale}}}, \tag{17}$$

where

$$g(X; W, b) = W * X + b, \tag{18}$$

with $*$ denoting convolution. Here $W_{Q_k}, W_{K_k} \in \mathbb{R}^{D\times D\times k_h\times k_w}$ are directional kernels and b_{Q_k}, b_{K_k} are biases.

Values are obtained via point-wise convolution:

$$V = W_V * \hat{F}^{\text{spatial}}_{\text{scale}} + b_V, \quad V \in \mathbb{R}^{B\times D\times H_{\text{scale}}\times W_{\text{scale}}}. \tag{19}$$

Attention maps are computed by softmax-normalizing the element-wise similarity of queries and keys:

$$A_k = \text{softmax}\Big(\tfrac{Q_k \odot K_k}{\sqrt{D}}\Big), \quad A_k \in \mathbb{R}^{B\times D\times H_{\text{scale}}\times W_{\text{scale}}}, \tag{20}$$

where $\odot$ is element-wise multiplication. Directional context features follow:

$$C_k = A_k \odot V, \quad C_k \in \mathbb{R}^{B\times D\times H_{\text{scale}}\times W_{\text{scale}}}. \tag{21}$$

All C_k are concatenated and reprojected:

$$F^{\text{attn}}_{\text{scale}} = W_O\big(C_1 \oplus C_2 \oplus \cdots \oplus C_K\big) + b_O, \quad F^{\text{attn}}_{\text{scale}} \in \mathbb{R}^{B\times D\times H_{\text{scale}}\times W_{\text{scale}}}, \tag{22}$$

where $\oplus$ denotes channel-wise concatenation.

Finally, these features undergo Layer Normalization and a depth-wise convolutional feed-forward network:

$$\tilde{F}_{\text{scale}} = \text{LayerNorm}\big(F^{\text{attn}}_{\text{scale}}\big), \tag{23}$$

$$F^{\text{FFN}}_{\text{scale}} = \text{FFN}_{dw}\big(\tilde{F}_{\text{scale}}\big) + F^{\text{attn}}_{\text{scale}}, \quad F^{\text{FFN}}_{\text{scale}} \in \mathbb{R}^{B\times D\times H_{\text{scale}}\times W_{\text{scale}}}. \tag{24}$$

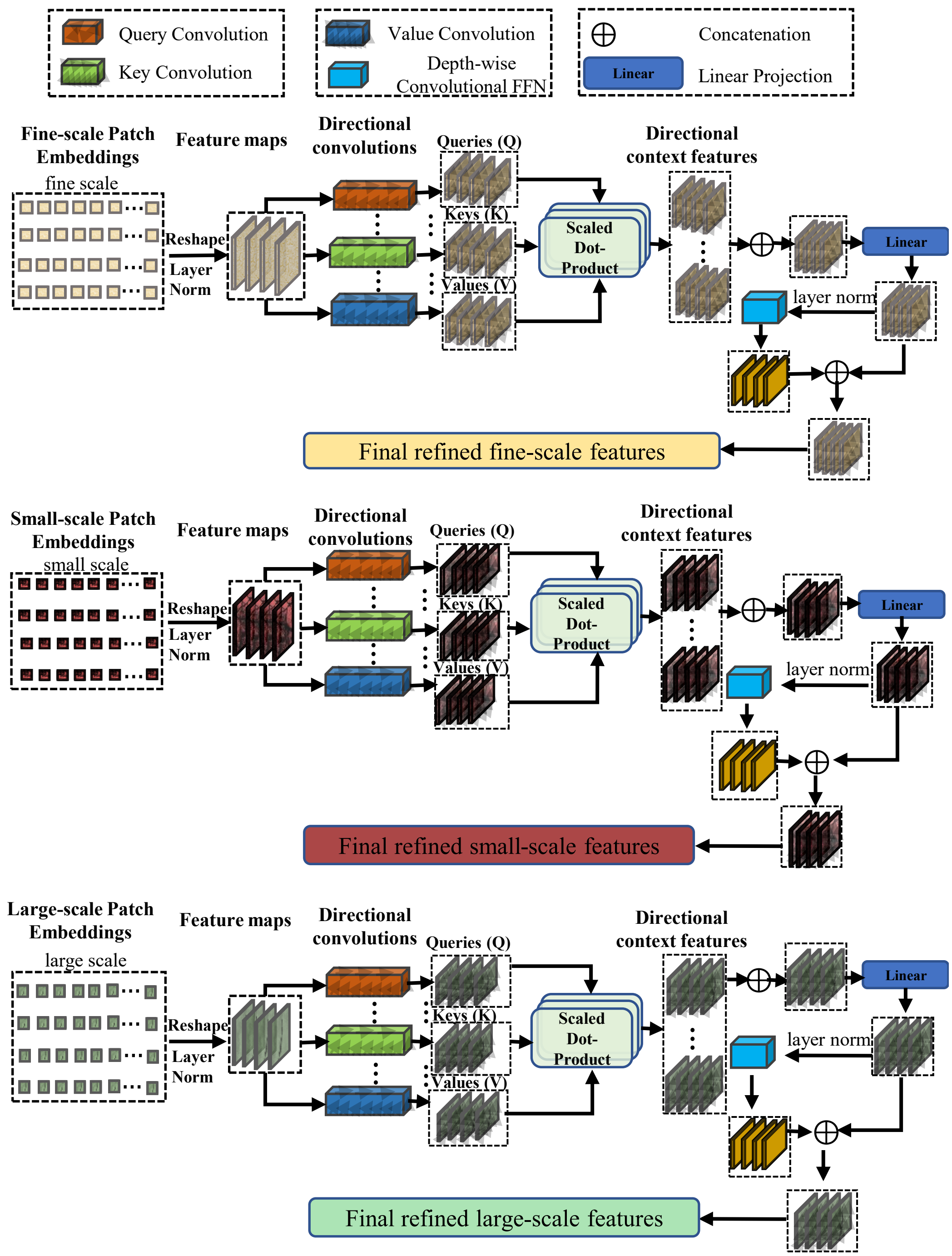

Figure 3: Directional Attention Transformer Module.

The output $F_{\text{scale}}^{\text{FFN}}$ thus encodes directional spatial relationships, enhancing multi-scale crack segmentation. These refined features feed into the Attention-Guided Fusion module.

3.5. Attention-Guided Fusion Module

The **Attention-Guided Fusion (AGF)** module intelligently integrates multi-scale feature representations into a unified, robust feature map. Following the Scale Adaptive Embedder and Directional Attention Transformer modules, this module uses adaptive attention mechanisms to determine the optimal combination of features from the fine, small, and large scales. The AGF module applies dynamic weights to each scale-specific feature according to its contextual relevance. This ensures that the final representation captures essential pavement crack details across multiple resolutions. Figure 4 shows the full architecture for the AGF module.

Let $F_f^{\text{FFN}} \in \mathbb{R}^{B\times D\times H\times W}$, $F_s^{\text{FFN}} \in \mathbb{R}^{B\times D\times H\times W}$, and $F_l^{\text{FFN}} \in \mathbb{R}^{B\times D\times \frac{H}{2}\times \frac{W}{2}}$ denote refined features from the fine, small, and large scales, respectively, obtained from the Directional Attention Transformer. To align spatial dimensions across scales, the large-scale feature map F_l^{FFN} is first upsampled and projected:

$$F_l^{\text{proj}} = \text{Conv}_{1\times 1}\big(\text{Upsample}(F_l^{\text{FFN}})\big), \quad F_l^{\text{proj}} \in \mathbb{R}^{B\times D\times H\times W}, \tag{25}$$

where $\text{Conv}_{1\times 1}$ is a 1×1 convolution to reduce dimensionality and match spatial dimensions, and Upsample denotes bilinear interpolation.

Next, these scale-specific feature maps are concatenated along the channel dimension:

$$F_{\text{cat}} = \big[F_l^{\text{proj}};\, F_s^{\text{FFN}};\, F_f^{\text{FFN}}\big], \quad F_{\text{cat}} \in \mathbb{R}^{B\times 3D\times H\times W}. \tag{26}$$

The composite map F_{cat} undergoes an attention-based weighting mechanism:

$$A = \sigma\big(W_A * F_{\text{cat}} + b_A\big), \quad A \in \mathbb{R}^{B\times 3\times H\times W}, \tag{27}$$

where $W_A \in \mathbb{R}^{3\times 3D\times 1\times 1}$, $b_A \in \mathbb{R}^3$, and σ is the sigmoid activation. A provides scale-specific attention weights.

The concatenated feature map (F_{cat}) is then split back into its scale-specific components (F_l^{split}, F_s^{split}, F_f^{split}):

$$F_l^{\text{split}} \in \mathbb{R}^{B\times D\times H\times W}, \tag{28}$$

$$F_s^{\text{split}} \in \mathbb{R}^{B\times D\times H\times W}, \tag{29}$$

$$F_f^{\text{split}} \in \mathbb{R}^{B\times D\times H\times W}. \tag{30}$$

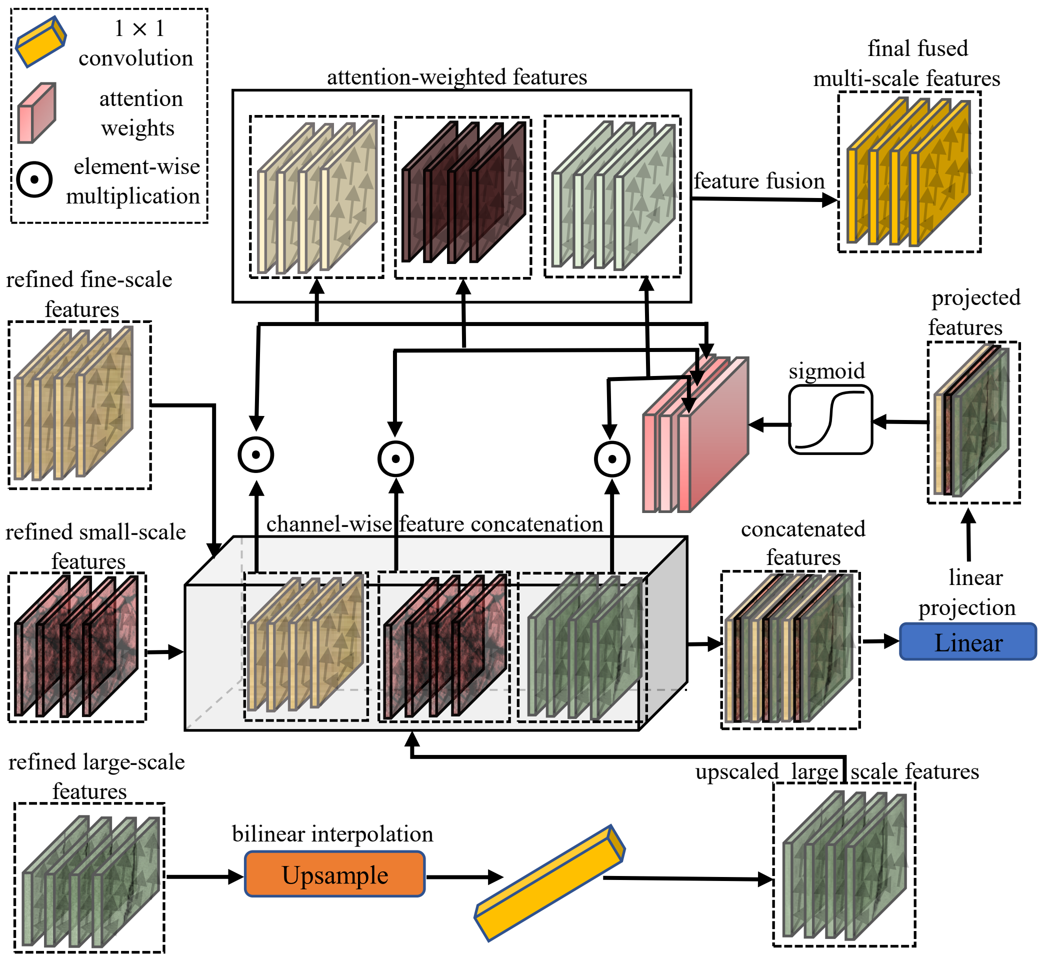

Figure 4: Attention-Guided Fusion Module.

Each component is weighted by its corresponding attention slice:

$$F_l^{\text{weighted}} = F_l^{\text{split}} \odot A[:, 0:1, :, :], \tag{31}$$
$$F_s^{\text{weighted}} = F_s^{\text{split}} \odot A[:, 1:2, :, :], \tag{32}$$
$$F_f^{\text{weighted}} = F_f^{\text{split}} \odot A[:, 2:3, :, :], \tag{33}$$

where $\odot$ denotes element-wise multiplication.

Finally, the attention-weighted features are summed:

$$F_{\text{fused}} = F_l^{\text{weighted}} + F_s^{\text{weighted}} + F_f^{\text{weighted}}, \quad F_{\text{fused}} \in \mathbb{R}^{B \times D \times H \times W}. \tag{34}$$

This adaptive fusion strategy ensures multi-scale information is optimally integrated for accurate pavement crack segmentation.

3.6. Cross-scale Consistency Loss

To effectively address the challenge of self-supervised pavement crack segmentation, our framework incorporates a cross-scale consistency loss [43], composed of two complementary components: Inter-scale Consistency Loss and Intra-scale Consistency Loss. These losses are specifically designed to promote consistency and coherence in the learned feature representations across and within scales, significantly enhancing the robustness and accuracy of the model without reliance on manual annotations.

3.6.1. Inter-scale Consistency Loss

The Inter-scale Consistency Loss enforces similarity by minimizing the cosine distance between feature representations at different scales (fine, small, large) , encouraging the model to learn scale-invariant crack features. Let $\mathbf{G}^f, \mathbf{G}^s, \mathbf{G}^l \in \mathbb{R}^d$ be the contextual feature vectors from fine, small, and large scales, respectively. The cosine similarity between two vectors is defined as:

$$\cos\big(\mathbf{G}^x, \mathbf{G}^y\big) = \frac{\mathbf{G}^x \cdot \mathbf{G}^y}{\|\mathbf{G}^x\|\,\|\mathbf{G}^y\|}, \qquad x, y \in \{f, s, l\}. \tag{35}$$

The Inter-scale Consistency Loss $\mathcal{L}_{\text{inter}}(\mathbf{G}^x, \mathbf{G}^y)$ minimizes the cosine dissimilarity:

$$\mathcal{L}_{\text{inter}}(\mathbf{G}^x, \mathbf{G}^y) = 1 \;-\; \cos\big(\mathbf{G}^x, \mathbf{G}^y\big). \tag{36}$$

In practice, this loss is computed pairwise between fine–small and small–large scales, with a weighting factor:

$$\mathcal{L}_{\text{inter}} = \lambda_1 \Big[\mathcal{L}_{\text{inter}}(\mathbf{G}^f, \mathbf{G}^s) + \mathcal{L}_{\text{inter}}(\mathbf{G}^s, \mathbf{G}^l)\Big], \tag{37}$$

where λ_1 is the weighting factor for the inter-scale consistency loss.

3.6.2. Intra-scale Consistency Loss

The Intra-scale Consistency Loss improves internal consistency within each scale-specific feature representation. Given an attention map $\mathbf{A} \in \mathbb{R}^{L\times L}$ and the identity matrix $\mathbf{I} \in \mathbb{R}^{L\times L}$, we enforce $\mathbf{A}$ to resemble $\mathbf{I}$ via an L_1 loss:

$$\mathcal{L}_{\text{intra}}(\mathbf{A}) = \frac{1}{L^2} \sum_{i=1}^{L} \sum_{j=1}^{L} \big|A_{ij} - I_{ij}\big|. \tag{38}$$

We weight this loss as:

$$\mathcal{L}_{\text{intra}} = \lambda_2 \, \mathcal{L}_{\text{intra}}(\mathbf{A}), \tag{39}$$

where λ_2 is the weighting factor for the intra-scale consistency loss.

3.6.3. Cross-Entropy Loss

To further support self-supervision, a pseudo-target was derived from the model's output itself $\mathbf{T} = [T_n] \in \{0,1\}^N$ by taking the highest predicted crack probability for each pixel:

$$T_n = \begin{cases} 1, & \text{if } O_n \geq 0.5, \\ 0, & \text{otherwise,} \end{cases} \qquad n = 1, \dots, N, \tag{40}$$

where $\mathbf{O} = [O_n] \in [0,1]^N$ is the model's output probability map and N is the total number of pixels.

The binary Cross-Entropy Loss between $\mathbf{O}$ and $\mathbf{T}$ is then defined as

$$\mathcal{L}_{\text{CE}}(\mathbf{O}, \mathbf{T}) = -\frac{1}{N} \sum_{n=1}^{N} \Big[T_n \, \log\big(O_n\big) + (1 - T_n) \, \log\big(1 - O_n\big) \Big], \tag{41}$$

where $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss, O_n is the predicted probability that pixel n belongs to the crack class, and T_n is the corresponding pseudo-target label.

3.6.4. Total Consistency Loss

Finally, the overall cross-scale consistency loss integrates all components:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}}(\mathbf{O}, \mathbf{T}) + \frac{1}{B} \sum_{b=1}^{B} \Big(\mathcal{L}_{\text{inter}}^{(b)} + \mathcal{L}_{\text{intra}}^{(b)} \Big), \tag{42}$$

where B is the batch size.

These self-supervisory losses ensure multi-scale representations remain coherent, internally consistent, and aligned with the model's predictions, significantly enhancing segmentation accuracy without ground truth annotations (masks).

4. Experimental details

In this section, the ten public datasets and the preprocessing steps used to prepare data for training and testing are described. Each dataset includes unique crack patterns and imaging conditions, which enables testing the model across varied scenarios. The routine adopted for training the proposed model including the training settings and evaluation metrics is then explained. Our study is complemented with ablation studies that helps isolate the contribution of each module in our proposed model.

4.1. Datasets and Preprocessing

Experiments were conducted on ten publicly available datasets to assess the effectiveness of the proposed self supervised segmentation approach. Each dataset contains images capturing different crack patterns across various surfaces, lighting conditions, and environmental scenarios. Utilizing these diverse datasets allows us to effectively assess the generalization capability and robustness of our segmentation model under different real-world conditions. Specifically, we employed the CFD dataset [44], Crack500 [45], CrackTree200 [46], DeepCrack [22], Eugen Miller [47], Forest [48], GAPs384 [49], Rissbilder [50], Sylvie [51], and Volker [50]. Each dataset is characterized by distinct types of cracks, surface materials, and varying illumination conditions, making them well-suited for training and validating our self-supervised model's performance. Table 1 provides a concise overview of each dataset's key characteristics, highlighting the variations in crack patterns and imaging environments.

Furthermore, each dataset was split into training and testing subsets, maintaining an 80:20 split to ensure consistency in our evaluation protocol. This division provides a fair comparison of our model's predictive performance on previously unseen data. To enhance the generalization and robustness of our model, several data augmentation strategies were applied during training. These augmentations included random horizontal and vertical flips, rotation transformations, and scaling variations. These augmentation techniques simulate the variability encountered in practical pavement inspection scenarios, enabling the model to better generalize and accurately segment cracks under diverse conditions.

Dataset	Crack Types	Surface Material	Lighting Conditions
CFD	Thin linear cracks	Asphalt pavement	Outdoor daylight, shadows, oil stains
Crack500	Hairline, wide cracks	Asphalt road surfaces	Mixed outdoor, varied weather
CrackTree200	Linear, alligator cracks	Asphalt pavement	Low contrast, uneven lighting
DeepCrack	Pavement, stone cracks	Asphalt concrete; stone	Daylight, some laser-lit
Eugen Miller	Random cracks	Tunnel concrete	Tunnel lighting
Forest	Thin linear cracks	Asphalt pavement	Outdoor daylight, shadows
GAPs384	Longitudinal, transverse, block	Asphalt roads	Dry daylight
Rissbilder	Architectural cracks	Concrete, masonry	Varied lighting
Sylvie	Linear, network cracks	Asphalt pavement	Outdoor varied lighting
Volker	Structural cracks	Concrete facades	Field conditions, well-lit

Table 1: Comparison of crack detection datasets

4.2. Implementation details

Our proposed self-supervised segmentation model was developed using the PyTorch deep learning library. The AdamW optimizer was used with a weight decay of 1×10^{-5} to mitigate potential overfitting. The initial learning rate was established at 1×10^{-4}. To enhance training efficiency, a learning rate scheduler that reduced the learning rate by a factor of 0.5 whenever validation performance stagnated for five consecutive epochs was used.

Only the original images from the datasets were used during training, explicitly excluding any segmentation masks to ensure a genuinely self supervised learning scenario.. For all experiments, the datasets were randomly divided into training and validation subsets, ensuring consistency and fairness across evaluations. For model evaluation and performance benchmarking, predicted segmentation masks were directly compared against the ground truth masks available within the validation datasets. Training was performed with a batch size of 8 for up to 500 epochs, incorporating early stopping with a patience of 100 epochs to prevent unnecessary computations and overfitting.

The performance of our model was compared with 14 state-of-the-art fully supervised segmentation models, including FCN [52], U-Net [53], U-Net++ [54], PSPNet [55], PAN [56], MAnet [57], LinkNet [58], FPN [59], DeepLabV3 [60], DeepLabV3+ [61], UPerNet [62], Segformer [63], and CrackFormer [26]. In total, all 14 segmentation models (the 13 fully-supervised baselines and our proposed self-supervised model) were each trained on all the 10 datasets, yielding a total of 140 experimental runs. Each baseline model was trained using the same experimental settings as our proposed model for consistency.

All computations were performed using the hardware and software configurations detailed in Table 2.

Component	**Details**
GPU	NVIDIA A40 (48 GB)
Framework	PyTorch 2.7
Programming Language	Python 3.9.12
CUDA Version	11.8
Optimizer	AdamW
Learning Rate Scheduler	Adaptive learning rate annealing

Table 2: Summary of the hardware and software environment used in experiments.

4.3. Evaluation metrics

Evaluation of the models were conducted using several standard metrics, including mean Intersection over Union (mIoU) and Dice coefficient. While these metrics assess overall overlap and similarity, additional metrics such as the XOR metric and Hammoud Distance (HM) were incorporated to further capture spatial disagreement and misalignment, providing deeper insights into the segmentation quality.

Mean Intersection over Union (mIoU). The mIoU metric calculates the average overlap between predicted masks and ground truth masks across different classes. For a particular class, IoU is computed by dividing the intersection of the prediction and ground truth by their union:

$$\mathrm{IoU}_c = \frac{|P_c \cap G_c|}{|P_c \cup G_c|},$$

where P_c and G_c represent the predicted and ground truth pixel sets for class c. The mIoU then averages the IoU scores across all C classes:

$$\mathrm{mIoU} = \frac{1}{C} \sum_{c=1}^{C} \mathrm{IoU}_c.$$

Dice Coefficient. The Dice coefficient measures how closely the predicted segmentation aligns with the ground truth. It emphasizes regions with smaller or finer details, making it particularly useful for pavement cracks. Dice is computed as:

$$\mathrm{Dice} = \frac{2 \times |P \cap G|}{|P| + |G|},$$

where P and G denote the predicted and actual sets of crack pixels, respectively.

XOR. The XOR metric quantifies discrepancies between the predicted and ground truth masks. It highlights areas exclusively classified as crack or non-crack in one mask but not in the other, thereby capturing mismatches effectively:

$$\mathrm{XOR} \;=\; \frac{1}{H\,W} \sum_{i=1}^{H} \sum_{j=1}^{W} \bigl(P_{ij} \oplus G_{ij}\bigr),$$

where P_{ij} and G_{ij} are the predicted and ground truth binary labels at pixel (i, j), and $\oplus$ denotes the logical XOR operation. A lower XOR value indicates better segmentation performance.

Hammoud Distance (HM). The Hammoud distance evaluates the spatial dissimilarity between predicted and ground truth masks. It quantifies the extent of spatial misalignment or inconsistency, defined mathematically as:

$$\mathrm{HM}(P, G) \;=\; \max\left\{ \max_{p \in P} \min_{g \in G} \bigl\|p - g\bigr\|_2,\;\; \max_{g \in G} \min_{p \in P} \bigl\|g - p\bigr\|_2 \right\},$$

where P and G are the sets of crack-pixel coordinates in the predicted and ground truth masks, respectively, and $\|\cdot\|_2$ denotes the Euclidean distance.

Lower HM values indicate higher segmentation accuracy, emphasizing better spatial alignment between prediction and ground truth.

5. Results and Discussion

5.1. Quantitative results

Table 3 provides a comprehensive performance comparison between our proposed self-supervised segmentation model and various state-of-the-art fully supervised methods across ten pavement crack datasets. The assessment utilized four metrics: mean mIoU, Dice score, XOR, and Hammoud distance.

Our proposed model achieved significantly superior performance compared to existing methods across nearly all datasets. On the CFD dataset, our model recorded a remarkable mIoU of 0.8875 and a Dice score of 0.9340, substantially surpassing U-Net++ (the best competitor among supervised methods) with an mIoU of 0.5257 and Dice of 0.6869. Moreover, our method exhibited notably lower XOR (0.6138) and HM (0.6138) values, demonstrating minimal spatial discrepancies and better alignment of predicted cracks.

Similarly, on the CRACK500 dataset, our self-supervised model achieved outstanding results, with an mIoU of 0.9332 and Dice of 0.9647, greatly exceeding the best-performing supervised method, Linknet, which attained mIoU and Dice scores of 0.6449 and 0.7838, respectively. The XOR and HM values of 0.1957 and 0.1629 further confirmed our model's exceptional spatial accuracy and lower false-positive rates.

The DeepCrack and Forest datasets also highlighted our model's robustness, achieving mIoU values of 0.8217 and 0.8167, respectively. These values substantially outperformed the next best methods, U-Net++ (0.7166 mIoU on DeepCrack) and U-Net (0.5392 mIoU on Forest). The high Dice scores of 0.8952 (DeepCrack) and 0.8896 (Forest) reinforced the accuracy improvements enabled by our framework's modules.

The CrackTree200 dataset, known for challenging crack patterns, demonstrated a significant performance leap with our model attaining an mIoU of 0.8670 and a Dice score of 0.9223, far superior to the second-best, U-Net, with 0.4861 mIoU and 0.6498 Dice scores. Similarly, our approach excelled on the GAPs dataset, delivering an mIoU of 0.8096 and Dice of 0.8854, surpassing Segformer's mIoU of 0.4016 and Dice of 0.5714 by a large margin.

Performance on Eugen Miller was also notably superior, with our model achieving an mIoU of 0.8451 and Dice of 0.9071, outperforming U-Net++

Table 3: Validation Results of *o*ur proposed and other models across all datasets.

Model	CFD				CRACK500				DeepCrack				Forest			
	mIoU↑	Dice↑	XOR↓	HM↓	mIoU↑	Dice↑	XOR↓	HM↓	mIoU↑	Dice↑	XOR↓	HM↓	mIoU↑	Dice↑	XOR↓	HM↓
FCN	0.3509	0.5155	1.7027	0.6491	0.5750	0.7088	0.8035	0.4250	0.6123	0.7448	0.7623	0.3877	0.3796	0.5460	1.5054	0.6204
Segformer	0.4734	0.6398	1.6703	0.6421	0.6381	0.7776	0.6747	0.4249	0.6556	0.7910	0.6538	0.3694	0.4743	0.6406	1.4148	0.6110
PSPNet	0.3552	0.5191	1.2426	0.6695	0.6207	0.7653	0.6716	0.4197	0.6576	0.7925	1.0336	0.4658	0.3840	0.5487	1.2346	0.6523
Linknet	0.4648	0.6232	1.7669	0.6835	0.6449	0.7838	0.6987	0.4263	0.7047	0.8243	1.3199	0.5122	0.4883	0.6412	1.5864	0.6554
FPN	0.4212	0.5833	1.3548	0.6373	0.6316	0.7740	0.6503	0.4348	0.6783	0.8078	0.9757	0.4544	0.4343	0.5976	1.2351	0.6181
Unet	0.5123	0.6743	2.2378	0.7018	0.6398	0.7796	0.7171	0.4174	0.7059	0.8255	1.2395	0.4970	0.5392	0.6974	1.5924	0.6441
PAN	0.3654	0.5166	78.3467	0.9853	0.6231	0.7674	0.6937	0.4240	0.6535	0.7891	0.8343	0.4317	0.4580	0.6261	1.2560	0.6160
DeepLabV3	0.3167	0.4744	1.9940	0.7353	0.6420	0.7816	0.6699	0.4132	0.6807	0.8092	1.0298	0.4622	0.4821	0.6477	1.3812	0.6352
DeepLabV3Plus	0.3863	0.5526	1.3633	0.6674	0.6376	0.7784	0.7092	0.4166	0.6641	0.7961	0.9354	0.4437	0.4610	0.6273	1.2988	0.6079
CrackFormer	0.4494	0.6178	1.4242	0.6186	0.6306	0.7702	0.8168	0.4435	0.6179	0.7592	0.6708	0.3755	0.4450	0.6139	1.3694	0.6201
UPerNet	0.4674	0.6352	1.5864	0.6286	0.6432	0.7813	0.7292	0.4255	0.6581	0.7929	0.6514	0.3644	0.4724	0.6381	1.4423	0.6179
MAnet	0.5166	0.6761	1.6948	0.6589	0.6341	0.7754	0.7453	0.4284	0.7004	0.8216	1.3568	0.5158	0.5174	0.6752	1.6298	0.6495
UnetPlusPlus	0.5257	0.6869	1.7524	0.6635	0.6443	0.7834	0.6703	0.4202	0.7166	0.8344	1.2447	0.4983	0.5457	0.7044	1.5658	0.6388
Crack-Segmenter (ours)	**0.8875**	**0.9340**	**0.6138**	**0.6138**	**0.9332**	**0.9647**	**0.1957**	**0.1629**	**0.8217**	**0.8952**	**0.3685**	**0.2405**	**0.8167**	**0.8896**	**0.5836**	**0.4003**
Crack-Segmenter-v0 (ours)	0.5435	0.7025	0.8597	0.4565	0.8572	0.9210	0.1782	0.1428	0.6955	0.7898	0.5003	0.3045	0.5723	0.7257	0.7695	0.4277
Crack-Segmenter-v1 (ours)	0.5432	0.7023	0.8603	0.4568	0.5360	0.6965	0.8708	0.4640	0.5698	0.7026	0.7064	0.4303	0.5554	0.7102	0.7972	0.4446
Crack-Segmenter-v2 (ours)	0.5360	0.6965	0.8708	0.4640	0.5711	0.7247	0.7699	0.4289	0.6836	0.8013	0.5388	0.3164	0.5711	0.7247	0.7699	0.4289
Crack-Segmenter-v3 (ours)	0.5779	0.7302	0.7300	0.4221	0.7457	0.8407	0.3033	0.2543	0.6797	0.7996	0.5590	0.3203	0.5153	0.6617	0.7999	0.4847

Model	CrackTree200				GAPs				Eugen Miller			
	mIoU↑	Dice↑	XOR↓	HM↓	mIoU↑	Dice↑	XOR↓	HM↓	mIoU↑	Dice↑	XOR↓	HM↓
FCN	0.065	0.1215	13.8985	0.9353	0.3648	0.5128	1.4705	0.6352	0.6215	0.7660	0.5168	0.3785
Segformer	0.2916	0.4509	12.3826	0.9299	0.4016	0.5714	1.3561	0.6268	0.5973	0.7402	0.6956	0.4543
PSPNet	0.2930	0.4477	6.7798	0.9178	0.2907	0.4432	1.2396	0.7015	0.6062	0.7479	0.5292	0.3905
Linknet	0.4804	0.6411	12.2496	0.9355	0.3996	0.5659	1.7396	0.6894	0.6021	0.7414	0.5740	0.4178
FPN	0.2962	0.4469	7.6561	0.9197	0.3068	0.4624	1.3438	0.7369	0.4682	0.6161	0.6412	0.4997
Unet	0.4861	0.6498	11.4862	0.9301	0.3832	0.5508	1.5567	0.6753	0.6315	0.7655	0.4998	0.3771
PAN	0.2755	0.4284	6.7856	0.9185	0.2761	0.4259	1.2340	0.7191	0.5796	0.7265	0.5651	0.4189
DeepLabV3	0.3178	0.4708	8.4792	0.9221	0.3685	0.5343	1.5615	0.6844	0.6291	0.7633	0.5928	0.4039
DeepLabV3Plus	0.2788	0.4243	9.0466	0.9205	0.3168	0.4705	1.5988	0.7019	0.6047	0.7457	0.5501	0.3999
CrackFormer	0.2656	0.4189	12.3523	0.9348	0.3470	0.5112	1.6896	0.6595	0.4756	0.6352	0.8786	0.6001
UPerNet	0.2569	0.4049	12.9757	0.9328	0.4029	0.5724	1.3435	0.6121	0.5835	0.7295	0.6135	0.4350
MAnet	0.4198	0.5773	11.9556	0.9330	0.3793	0.5414	1.6618	0.6844	0.5892	0.7322	0.5547	0.4110
UnetPlusPlus	0.4255	0.5927	8.9129	0.9183	0.3926	0.5561	1.6479	0.6858	0.7072	0.8214	0.4798	0.3505
Crack-Segmenter (ours)	**0.8670**	**0.9223**	**5.0154**	**0.8409**	**0.8096**	**0.8854**	**0.6542**	**0.4497**	0.8451	0.9071	0.3952	0.3669
Crack-Segmenter-v0 (ours)	0.1572	0.2716	5.1871	0.8428	0.6353	0.7711	0.6264	0.3647	**0.8597**	**0.9245**	**0.1635**	**0.1403**
Crack-Segmenter-v1 (ours)	0.1474	0.2568	5.8197	0.8526	0.6312	0.7682	0.6293	0.3688	0.8016	0.8860	0.2237	0.1984
Crack-Segmenter-v2 (ours)	0.1406	0.2453	5.1946	0.8594	0.6353	0.7711	0.6264	0.3647	0.6860	0.8080	0.3556	0.3140
Crack-Segmenter-v3 (ours)	0.1385	0.2430	5.0391	0.8615	0.4953	0.6450	0.7485	0.5047	0.7230	0.8369	0.3078	0.2770

Model	Rissbilder				Sylvie				Volker			
	mIoU↑	Dice↑	XOR↓	HM↓	mIoU↑	Dice↑	XOR↓	HM↓	mIoU↑	Dice↑	XOR↓	HM↓
FCN	0.5098	0.6707	0.8074	0.4902	0.7498	0.8439	0.3016	0.2502	0.6658	0.7981	0.4685	0.3342
Segformer	0.5169	0.6810	0.7938	0.4912	0.6600	0.7899	0.3960	0.2939	0.6632	0.7973	0.4732	0.3407
PSPNet	0.5042	0.6685	0.7935	0.5339	0.5548	0.6998	0.4102	0.3592	0.6865	0.8132	0.4625	0.3401
Linknet	0.6084	0.7497	0.8332	0.4941	0.6430	0.7750	0.5250	0.3740	0.7235	0.8344	0.4871	0.3439
FPN	0.5822	0.7348	0.7652	0.4932	0.5993	0.7456	0.4893	0.3898	0.7026	0.8242	0.4505	0.3341
Unet	0.6449	0.7836	0.8240	0.4836	0.6558	0.7891	0.4501	0.3269	0.7454	0.8532	0.4788	0.3349
PAN	0.5415	0.7015	0.7529	0.5039	0.6073	0.7504	0.4030	0.3438	0.6790	0.8076	0.4574	0.3405
DeepLabV3	0.5970	0.7474	0.7754	0.4857	0.6639	0.7968	0.3633	0.2972	0.7213	0.8378	0.4448	0.3280
DeepLabV3Plus	0.5650	0.7210	0.7781	0.4965	0.6508	0.7826	0.3683	0.3038	0.6871	0.8139	0.4748	0.3467
CrackFormer	0.4638	0.6313	0.8590	0.5126	0.4829	0.6402	0.6936	0.5363	0.6182	0.7631	0.5042	0.3614
UPerNet	0.5150	0.6794	0.8045	0.4872	0.6663	0.7946	0.3450	0.2680	0.6682	0.8009	0.4628	0.3331
MAnet	0.6370	0.7771	0.8190	0.4811	0.6254	0.7642	0.4140	0.3246	0.7027	0.8245	0.5236	0.3598
UnetPlusPlus	0.6564	0.7920	0.7964	0.4744	0.6719	0.8021	0.5109	0.3890	0.7641	0.8659	0.4655	0.3274
Crack-Segmenter (ours)	0.7998	0.8785	0.5047	0.4344	0.8891	0.9352	0.3341	0.3122	0.6707	0.7955	0.6434	0.6240
Crack-Segmenter-v0 (ours)	0.8075	0.8932	0.2398	0.1925	0.9063	0.9503	0.1060	0.0937	0.8746	0.9330	0.1437	0.1254
Crack-Segmenter-v1 (ours)	**0.8079**	**0.8934**	**0.2394**	**0.1921**	**0.9065**	**0.9504**	**0.1057**	**0.0935**	**0.8761**	**0.9338**	**0.1421**	**0.1239**
Crack-Segmenter-v2 (ours)	0.7894	0.8810	0.2611	0.2106	0.7873	0.8745	0.2326	0.2127	0.8637	0.9265	0.1560	0.1363
Crack-Segmenter-v3 (ours)	0.6740	0.7979	0.3898	0.3260	0.8802	0.9354	0.1344	0.1198	0.5804	0.7303	0.4637	0.4196

with 0.7072 mIoU and 0.8214 Dice. These results indicate our model's effectiveness even in diverse structural conditions.

However, the performance on the Volker dataset was slightly lower, with an mIoU of 0.6707 and Dice of 0.7955, compared to U-Net++, which achieved an mIoU of 0.7641 and Dice of 0.8659. This suggests room for further refinement of our multi-scale attention mechanisms to handle highly variable crack structures effectively. Figure 5 shows mIoU and Dice scores of Crack-Segmenter and all the baseline models across the different datasets summarised in a radar plot.

The consistently superior quantitative results clearly demonstrate the effectiveness of the integrated modules within our self-supervised segmentation framework. Specifically, the Scale-Adaptive Embedder efficiently captures comprehensive multi-scale feature details; the Directional Attention Transformer emphasizes linear crack structures, improving the detection accuracy; and the Attention-Guided Fusion optimally merges multi-scale features, enhancing overall segmentation performance. Collectively, these modules facilitate accurate, annotation-free crack segmentation, significantly advancing the state-of-the-art in pavement distress assessment.

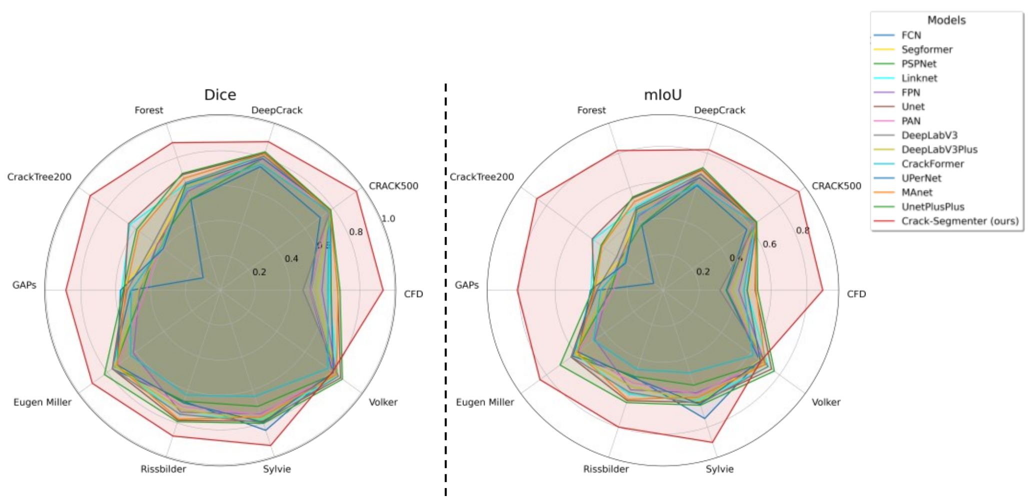

Figure 5: Radar plots for validation Dice and mIoU scores of all the baseline models and Crack-Segmenter across all the 10 datasets.

5.2. *Ablation studies*

To evaluate the individual contributions of each proposed module within our self-supervised segmentation framework, we conducted comprehensive ablation experiments on the DeepCrack dataset. This systematic analysis isolates the effectiveness of each component: the Scale-Adaptive Embedder (SAE), Directional Attention Transformer (DAT), and Attention-Guided Fusion (AGF), and examines their interactions when combined in various configurations.

5.2.1. *Experimental Setup*

We evaluated six distinct architectural variants using mIoU and Dice score as primary metrics. Each variant represents a specific combination of the proposed modules, allowing us to quantify both individual contributions to segmentation performance. The following variants were systematically evaluated:

Baseline: Standard U-Net architecture with ResNet-18 encoder, representing conventional supervised segmentation without self-supervised components.

Crack-Segmenter: Complete architecture incorporating all three proposed modules.

Crack-Segmenter-v0: Integration of SAE module only, focusing on multi-scale representation learning capabilities.

Crack-Segmenter-v1: Combination of SAE and DAT modules, emphasizing directional attention alongside multi-scale features.

Crack-Segmenter-v2: Integration of SAE and AGF modules, combining multi-scale embeddings with attention-guided fusion.

Crack-Segmenter-v3: Combination of DAT and AGF modules without multi-scale embeddings.

5.2.2. *Analysis*

To assess the individual contributions of each proposed modules, we conducted systematic ablation experiments on the DeepCrack dataset. These studies are crucial for understanding the role each component plays in the overall performance of our self-supervised segmentation framework. All ablation variants were evaluated using the four key metrics: mIoU, Dice score, XOR Score, and HM Score.

We begin by defining the **Baseline** model, which consists of a standard U-Net architecture with a ResNet-18 encoder and none of the proposed self-supervised modules. This configuration serves as a conventional supervised benchmark. As shown in Table 4, it achieved an mIoU of 0.5866 and a Dice score of 0.7258. These results establish a foundational point of comparison for the proposed architectural variations.

When only the SAE module was included in the architecture (**Crack-Segmenter-v0**), we observed a substantial improvement in both metrics, reaching an mIoU of 0.6955 and a Dice score of 0.7898. This confirms the strong impact of multi-scale representations in capturing crack structures of varying widths. SAE provides the model with diverse spatial resolutions, enabling it to better recognize both fine-grained and coarse crack patterns, which are typically missed in single-scale encoders.

In the **Crack-Segmenter-v1** variant, we added the DAT module on top of SAE while leaving out AGF. Interestingly, this combination led to a drop in performance (mIoU of 0.5698 and Dice of 0.7026), falling even below the baseline. This suggests that applying directional attention without a proper fusion mechanism may not be sufficient for effective feature integration. DAT focuses on enhancing linear crack continuity through spatially aware convolutions, but without adaptive fusion to resolve scale-level redundancies, it may introduce conflicting or misaligned representations.

The combination of SAE and AGF in **Crack-Segmenter-v2** demonstrated stronger performance, with 0.6836 mIoU and 0.8013 Dice. These results reinforce the importance of adaptive feature fusion when dealing with multi-scale representations. AGF dynamically learns to assign relevance weights across different spatial resolutions, ensuring optimal use of both local and contextual information during segmentation. The effective integration of scale-specific details allows the model to better adapt to irregular crack geometries and surrounding textures.

We then evaluated **Crack-Segmenter-v3**, which includes only DAT and AGF, excluding SAE. This variant resulted in an mIoU of 0.6797 and Dice score of 0.7996. Although this is an improvement over the baseline, the absence of SAE resulted in weaker feature diversity, reducing the ability of the fusion and attention mechanisms to operate on rich, scale-aware representations. This again highlights the importance of SAE as a foundation for multi-scale learning.

Finally, the full model (**Crack-Segmenter**) that integrates all three modules: SAE, DAT, and AGF achieved the highest performance across all metrics: 0.8217 mIoU, 0.8952 Dice, 0.3685 XOR, and 0.2405 HM. These results confirm that the complete architecture benefits from the combined strengths of its components. SAE enables comprehensive multi-scale feature extraction, DAT enhances directional continuity, and AGF fuses these features adaptively. Together, this integration result in a robust and coherent segmentation map that generalizes well across complex crack structures and challenging visual conditions.

Table 4: Ablation study results on the DeepCrack dataset. Each variant systematically evaluates different module combinations to assess individual contributions.

Model Variant	**Metric**				**Module Components**		
	mIoU↑	**Dice↑**	**XOR↓**	**HM↓**	**SAE**	**DAT**	**AGF**
Baseline	0.5866	0.7258	0.7675	0.4134	✗	✗	✗
Crack-Segmenter	**0.8217**	**0.8952**	**0.3685**	**0.2405**	✓	✓	✓
Crack-Segmenter-v0	0.6955	0.7898	0.5003	0.3405	✓	✗	✗
Crack-Segmenter-v1	0.5698	0.7026	0.7064	0.4303	✓	✓	✗
Crack-Segmenter-v2	0.6836	0.8013	0.5388	0.3164	✓	✗	✓
Crack-Segmenter-v3	0.6797	0.7996	0.5590	0.3203	✗	✓	✓

These ablation results validate the design of our self-supervised architecture and demonstrate how each module contributes to enhancing segmentation quality. The clear performance gains of the full model emphasize the necessity of combining multi-scale embeddings, directional attention, and guided fusion in an integrated manner for accurate pavement crack segmentation.

5.3. Model Explainability

Interpreting model behavior is essential for validating its reliability, especially in safety-critical applications such as pavement crack detection. To assess whether the proposed *Crack-Segmenter* attends to relevant spatial structures, attention map visualizations were employed to analyze its focus during inference.

Attention maps were extracted from the final transformer block at each of the three spatial scales (fine, small, and large) in the architecture. These layers capture the model's most refined representations across different resolutions. For blocks with multiple attention heads, attention weights were averaged across heads to produce a single map per scale, ensuring clarity while preserving dominant spatial patterns. All attention maps were then normalized to the range [0, 1] and overlaid as heatmaps on the original DeepCrack images, where color intensity reflects attention strength. Regions receiving strong attention appear warmer (e.g., red or yellow), while areas with low attention are cooler (e.g., blue), enabling clear visual identification of the model's focus during segmentation.

Figure 6 presents example attention visualizations for sample images in the DeepCrack dataset. The model consistently concentrates attention along actual crack regions, while suppressing distractors such as shadows, surface texture, and background noise. This focused behavior highlights the effectiveness of the Directional Attention Transformer and Attention-Guided Fusion modules in guiding the network toward semantically meaningful features. These visualizations confirm that the model captures relevant structural cues without explicit supervision, reinforcing both the design rationale and the effectiveness of the proposed framework.

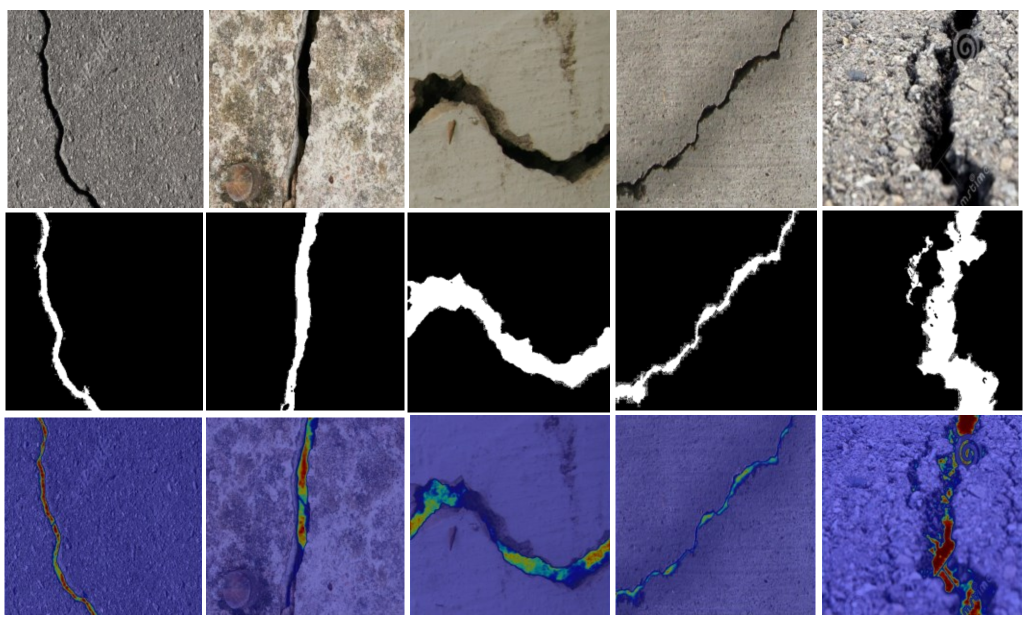

Figure 6: Attention map visualization of *Crack-Segmenter* on samples images from DeepCrack dataset. Top row: input images. Middle row: predicted masks. Bottom row: attention map overlays highlighting regions the model focused on during crack segmentation.

5.4. Statistical Analysis

To rigorously assess the statistical significance of our proposed *Crack-Segmenter*, we conducted detailed statistical analysis comparing its performance against state-of-the-art segmentation methods. These analysis focused on evaluating consistency and superiority across multiple segmentation metrics, including mIoU, Dice score, XOR, and HM. Table 5 summarizes the mean performance and standard deviations of each method averaged over all datasets.

From Table 5, it is evident that the proposed *Crack-Segmenter* achieved consistently superior results across all evaluation metrics. Specifically, our model attained the highest mIoU (0.8340 ± 0.0714) and Dice score (0.9008 ± 0.0458), indicating significantly more accurate segmentation performance compared to other baseline models. Additionally, it showed the lowest XOR (0.9309 ± 1.4432) and HM (0.4446 ± 0.2013) values, reflecting fewer misclassifications and better spatial alignment with ground-truth crack structures.

Table 5: Mean Performance and Standard Deviation by Model accross all datasets

Model	**mIoU**†	**Dice**†	**XOR**‡	**HM**‡
DeepLabV3	0.5419 ± 0.1569	0.6863 ± 0.1434	1.7292 ± 2.4292	0.5367 ± 0.2004
DeepLabV3Plus	0.5252 ± 0.1525	0.6712 ± 0.1434	1.7123 ± 2.6091	0.5305 ± 0.1915
FCN	0.4894 ± 0.2017	0.6228 ± 0.2123	2.2237 ± 4.1299	0.5106 ± 0.2018
FPN	0.5121 ± 0.1475	0.6593 ± 0.1378	1.5562 ± 2.1700	0.5518 ± 0.1777
Linknet	0.5760 ± 0.1104	0.7180 ± 0.0930	2.1780 ± 3.5753	0.5532 ± 0.1849
MAnet	0.5722 ± 0.1111	0.7165 ± 0.0976	2.1355 ± 3.4871	0.5447 ± 0.1866
PAN	0.5059 ± 0.1529	0.6540 ± 0.1469	9.1329 ± 24.3942	0.5702 ± 0.2333
PSPNet	0.4953 ± 0.1525	0.6446 ± 0.1428	1.4397 ± 1.9048	0.5450 ± 0.1865
Segformer	0.5372 ± 0.1273	0.6880 ± 0.1146	2.0511 ± 3.6555	0.5184 ± 0.1894
UPerNet	0.5334 ± 0.1366	0.6829 ± 0.1266	2.0954 ± 3.8467	0.5105 ± 0.1941
Unet	0.5944 ± 0.1111	0.7369 ± 0.0923	2.1082 ± 3.3477	0.5388 ± 0.1948
UnetPlusPlus	0.6050 ± 0.1265	0.7439 ± 0.1050	1.8047 ± 2.5484	0.5366 ± 0.1867
CrackFormer	0.4796 ± 0.1187	0.6361 ± 0.1120	2.1258 ± 3.6134	0.5662 ± 0.1662
Crack-Segmenter (Ours)	**0.8340 ± 0.0714**	**0.9008 ± 0.0458**	**0.9309 ± 1.4432**	**0.4446 ± 0.2013**

†Higher values indicate better performance (mIoU, Dice). ‡Lower values indicate better performance (XOR, HM). **Bold** values indicate best performance for each metric.

To determine whether these performance improvements were statistically meaningful, paired t-tests were conducted between our model and each baseline method, as shown in Table 6. Our model exhibited statistically significant improvements in both mIoU and Dice score compared to all baseline methods, as indicated by positive mean differences and very low p-values ($p < 0.01$ and mostly $p < 0.001$). For instance, when compared with commonly used segmentation models such as U-Net and DeepLabV3, the proposed method demonstrated substantial mean improvements in mIoU (+0.2396 and +0.2921, respectively) and Dice score (+0.1639 and +0.2144, respectively), all statistically significant at $p < 0.01$.

Furthermore, the highest statistical significance was noted against the CrackFormer model, with a mean difference of +0.3544 in mIoU and +0.2647 in Dice score (both $p < 0.001$). This strong statistical evidence confirms the effectiveness of integrating multi-scale embeddings, directional attention mechanisms, and adaptive fusion within our architecture. These improvements likely stem from the enhanced capability of our model to accurately capture diverse crack patterns and textures, as demonstrated consistently across datasets. Figure 7 shows distribution of the dice and mIoU scores of Crack-Segmenter and the baseline models.

Table 6: Statistical significance tests comparing Crack-Segmenter against baseline methods. Mean differences, t-statistics, p-values, and significance levels are reported for each evaluation metric. Positive mean differences for mIoU and Dice indicate superior performance.

		Models					
Metric	**Statistic**	**FCN**	**SegFormer**	**PSPNet**	**LinkNet**	**FPN**	**U-Net**
mIoU	mean diff.	+0.3446	+0.2968	+0.3387	+0.2581	+0.3220	+0.2396
	t-statistic	4.829	6.091	5.840	5.560	5.657	5.089
	p-value	0.000935	0.000181	0.000247	0.000352	0.000311	0.000655
	significance	***	***	***	***	***	***
Dice	mean Diff.	+0.2779	+0.2128	+0.2562	+0.1828	+0.2415	+0.1639
	t-statistic	3.886	5.204	5.084	5.125	4.927	4.586
	p-value	0.003696	0.000561	0.000659	0.000624	0.000816	0.001317
	significance	**	***	***	***	***	**

Metric	**Statistic**	**PAN**	**DeepLabV3**	**DeepLabV3+**	**CrackFormer**	**UPerNet**	**MANet**	**U-Net++**
mIoU	mean diff.	+0.3281	+0.2921	+0.3088	+0.3544	+0.3006	+0.2618	+0.2290
	t-statistic	5.682	4.852	5.394	7.523	5.837	5.649	4.449
	p-value	0.000301	0.000906	0.000437	0.000036	0.000248	0.000314	0.001603
	significance	***	***	***	***	***	***	**
Dice	mean Diff.	+0.2468	+0.2144	+0.2295	+0.2647	+0.2178	+0.1843	+0.1568
	t-statistic	4.799	4.180	4.573	6.516	4.887	4.993	3.968
	p-value	0.000975	0.002377	0.001341	0.000109	0.000863	0.000746	0.003264
	significance	***	**	**	***	***	***	**

Note: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$. Paired t-tests assume the same dataset was used across all models.

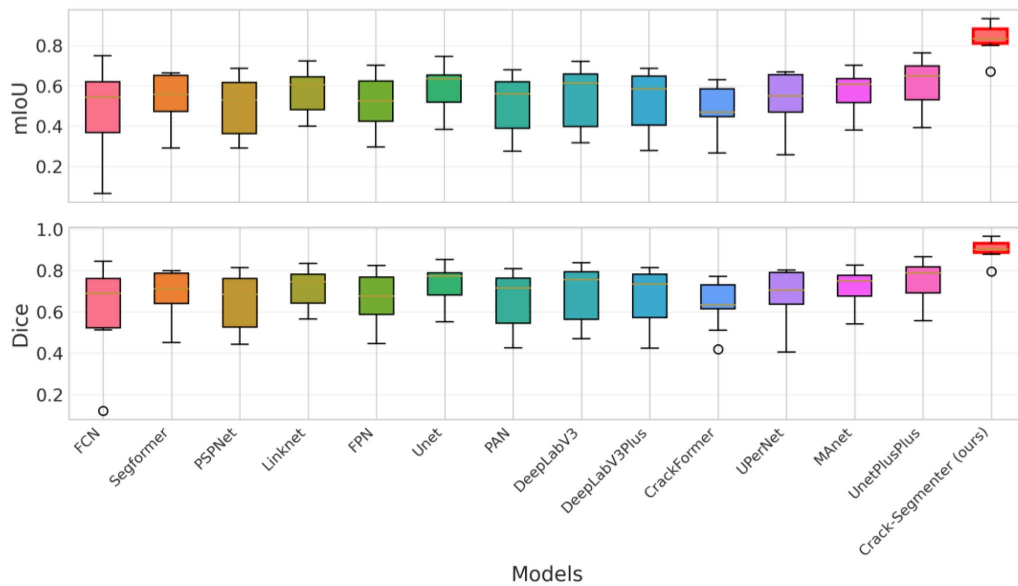

Figure 7: Box plot showing the mIoU and Dice Scores of Crack-Segmenter against other model baselines.

5.5. Qualitative Results

Figure 8 presents visual comparisons between our self-supervised method and supervised baselines across nine datasets. Each row displays a different dataset, enabling comprehensive assessment of segmentation performance across diverse crack patterns and imaging conditions.

On the CFD dataset with thin linear cracks, our method produces clean, continuous segmentations closely matching ground truth. While baseline methods capture general crack structure, they show inconsistent crack width. Our multi-scale embeddings effectively capture fine details at appropriate resolutions, resulting in well-defined boundaries without excessive dilation.

For CRACK500's vertical cracks with varying widths, our approach maintains consistent segmentation quality throughout, accurately preserving narrow and wider sections. The Directional Attention Transformer emphasizes vertical continuity, leading to better boundary precision and endpoint detection compared to baselines.

The DeepCrack dataset contains horizontal cracks with irregular edges. All methods perform competently, but ours shows superior noise suppression while maintaining sharp boundaries. The Attention-Guided Fusion module balances detail preservation with contextual consistency, reducing background artifacts present in baseline predictions. On Forest's multiple branching cracks, our method excels at maintaining connectivity at junctions, preserving complete network structure.

CrackTree200 presents dense interconnected crack networks. Our approach captures intricate patterns while maintaining clear boundaries between closely spaced cracks. Multi-scale processing enables simultaneous detection of major branches and fine subsidiary cracks, whereas baselines either over-segment (merging adjacent cracks) or under-segment (missing finer branches).

For GAPs' diagonal cracks with variable contrast, our method demonstrates robust performance even in low-contrast regions, maintaining consistent quality along entire crack lengths. The self-supervised learning identifies patterns based on structural characteristics rather than intensity contrasts alone.

The Eugen Miller dataset contains curved tunnel cracks under varying lighting. Our approach provides smoother curve representations with fewer discretization artifacts than baselines. The directional attention adapts to changing crack orientations effectively.

On Rissbilder's intersecting cracks, our method accurately segments both primary cracks and intersections without creating artificial connections or gaps. This demonstrates effective handling of complex spatial relationships through attention-guided fusion.

For Sylvie's prominent horizontal cracks, all methods achieve satisfactory results, but ours produces the most consistent width representation along the entire length. This consistency stems from balanced feature extraction across scales.

These qualitative results confirm that our self-supervised approach achieves segmentation quality matching or exceeding supervised methods while eliminating manual annotation requirements. The visual evidence demonstrates superior crack continuity preservation, consistent boundary maintenance, and adaptation to diverse crack morphologies across different pavement conditions.

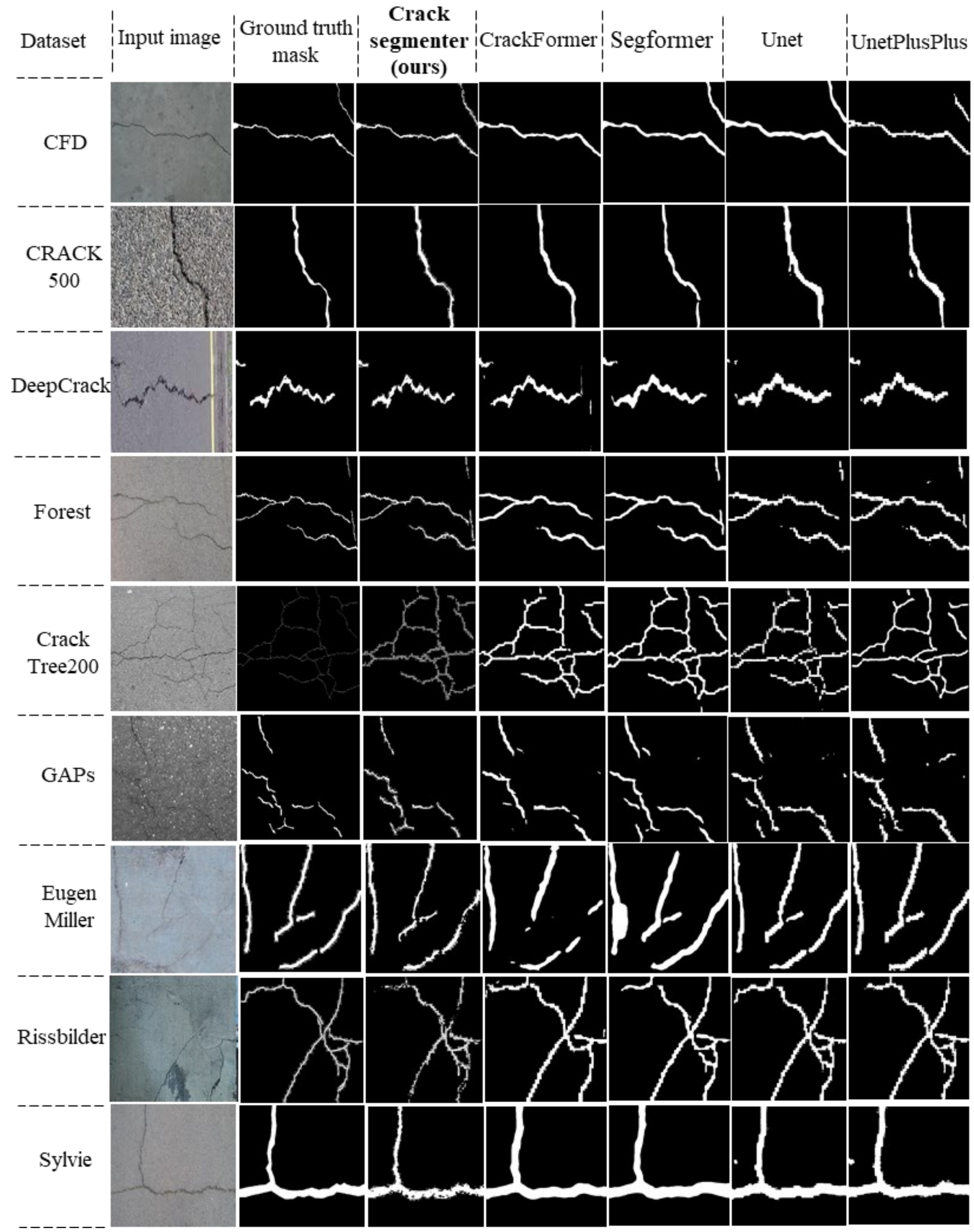

Figure 8: Qualitative comparison of predicted segmentation masks across multiple datasets. Each row shows the input image, ground truth, and predictions from the chosen baseline models (CrackFormer, Segformer, Unet, UnetPlusPlus) and the proposed *Crack-Segmenter.*

6. Practical Applications

The proposed *Crack-Segmenter* framework offers several practical benefits in construction and infrastructure management. Efficient pavement crack detection is vital for ensuring road safety, enhancing infrastructure durability, and optimizing asset management. Eliminating manual annotation costs enables transportation agencies to conduct regular, comprehensive inspections more affordably.

A primary application is preventive pavement maintenance planning. The accurate crack segmentation maps from our model support early identification and timely repair of pavement defects, enabling efficient budget allocation and reducing long-term maintenance expenses.

Additionally, our method seamlessly integrates with automated inspection technologies, such as unmanned aerial vehicles (UAVs) and vehicle-mounted systems, enabling rapid, continuous monitoring of large road networks with minimal human involvement. This automation increases inspection efficiency and consistency across expansive areas.

Construction and engineering firms can apply this approach for proactive monitoring of new or rehabilitated pavement surfaces, swiftly identifying potential structural weaknesses. This early detection allows for timely corrective measures, extending pavement lifespan and enhancing public safety.

Finally, the annotation-free nature of our approach facilitates adoption in resource-constrained regions, democratizing access to advanced pavement assessment tools and supporting equitable infrastructure management practices globally.

7. Conclusion

This study presented *Crack-Segmenter*, a fully self-supervised segmentation framework developed specifically for pavement crack detection. Our approach successfully eliminates the dependence on costly and labor-intensive pixel-level annotations, addressing a significant limitation of existing segmentation methods. The proposed model integrates three innovative modules: the Scale-Adaptive Embedder (SAE), Directional Attention Transformer (DAT), and Attention-Guided Fusion (AGF). Each module targets specific challenges in crack segmentation, collectively enhancing the model's capability to accurately detect diverse crack patterns.

Experimental evaluations conducted on ten publicly available datasets demonstrated the effectiveness of the proposed framework. *Crack-Segmenter* significantly outperformed state-of-the-art fully supervised methods across multiple metrics, including mIoU, Dice score, XOR, and Hammoud Distance (HM). Comprehensive statistical analysis confirmed the statistical significance of these improvements, with notably strong performance gains observed over prominent baseline methods.

Ablation studies further underscored the importance of each proposed module. Specifically, SAE facilitated effective multi-scale feature extraction, capturing cracks across varied widths and complexities. DAT enhanced spatial coherence and continuity, crucial for linear crack structures. Finally, AGF intelligently fused these features, emphasizing contextually relevant information at each scale. Collectively, these modules delivered superior segmentation accuracy, confirming their individual and combined effectiveness.

Attention map visualizations provided interpretability, confirming that *Crack-Segmenter* correctly focused on meaningful crack regions, reducing background distractions. Such explainability strengthens confidence in deploying this method in practical, real-world pavement monitoring tasks.

In future work, extending this framework to handle other pavement distress types and further optimizing computational efficiency could expand its utility. Additionally, exploring methods to integrate temporal information from sequential pavement inspections may improve detection robustness over time, enhancing preventive maintenance practices.

References

[1] M. Barman, J. Munch, U. M. Arepalli, Cost/benefit analysis of the effectiveness of crack sealing techniques, Research Report MN/RC 2019-26, University of Minnesota, Duluth. Department of Civil Engineering, prepared for the Local Road Research Board (June 2019).
URL `https://rosap.ntl.bts.gov/view/dot/61830`

[2] B. A. Kyem, E. K. O. Denteh, J. K. Asamoah, A. Aboah, Pavecap: The first multimodal framework for comprehensive pavement condition assessment with dense captioning and pci estimation, ArXiv abs/2408.04110 (2024).
URL `https://api.semanticscholar.org/CorpusID:271768959`

[3] S. L. H. Lau, E. Chong, X. Yang, X. Wang, Automated pavement crack segmentation using u-net-based convolutional neural network, IEEE Access 8 (2020) 114892–114899. doi:10.1109/ACCESS.2020.3003638.

[4] B. Agyei Kyem, J. K. Asamoah, A. Aboah, Context-cracknet: A context-aware framework for precise segmentation of tiny cracks in pavement images, Construction and Building Materials 484 (2025) 141583. doi:https://doi.org/10.1016/j.conbuildmat.2025.141583.
URL https://www.sciencedirect.com/science/article/pii/S0950061825017337

[5] T. Wen, S. Ding, H. Lang, J. J. Lu, Y. Yuan, Y. Peng, J. Chen, A. Wang, Automated pavement distress segmentation on asphalt surfaces using a deep learning network, International Journal of Pavement Engineering 24 (2) (2023) 2027414. arXiv:https://doi.org/10.1080/10298436.2022.2027414, doi:10.1080/10298436.2022.2027414.
URL https://doi.org/10.1080/10298436.2022.2027414

[6] C. Xiang, V. J. Gan, L. Deng, J. Guo, S. Xu, Unified weakly and semi-supervised crack segmentation framework using limited coarse labels, Engineering Applications of Artificial Intelligence 133 (2024) 108497. doi:https://doi.org/10.1016/j.engappai.2024.108497.
URL https://www.sciencedirect.com/science/article/pii/S0952197624006559

[7] J. Li, C. Yuan, X. Wang, G. Chen, G. Ma, Semi-supervised crack detection using segment anything model and deep transfer learning, Automation in Construction 170 (2025) 105899. doi:https://doi.org/10.1016/j.autcon.2024.105899.
URL https://www.sciencedirect.com/science/article/pii/S0926580524006356

[8] A. A. Neema Jakisa Owor, Yaw Adu-Gyamfi, M. Amo-Boateng, Pavesam – segment anything for pavement distress, Road Materials and Pavement Design 0 (0) (2024) 1–25. arXiv:https://doi.org/10.1080/14680629.2024.2374863, doi:10.1080/14680629.2024.2374863.
URL https://doi.org/10.1080/14680629.2024.2374863

[9] G. Li, J. Wan, S. He, Q. Liu, B. Ma, Semi-supervised semantic segmentation using adversarial learning for pavement crack detection, IEEE Access 8 (2020) 51446–51459. doi:10.1109/ACCESS.2020.2980086.

[10] S. Shim, J. Kim, G.-C. Cho, S.-W. Lee, Multiscale and adversarial learning-based semi-supervised semantic segmentation approach for

crack detection in concrete structures, IEEE Access 8 (2020) 170939–170950. doi:10.1109/ACCESS.2020.3022786.

[11] B. A. Kyem, J. K. Asamoah, Y. Huang, A. Aboah, Weather-adaptive synthetic data generation for enhanced power line inspection using stargan, IEEE Access 12 (2024) 193882–193901. doi:10.1109/ACCESS.2024.3520120.

[12] W. Wang, C. Su, Semi-supervised semantic segmentation network for surface crack detection, Automation in Construction (2021). doi:10.1016/J.AUTCON.2021.103786.

[13] M. Mohammed, Z. Han, Y. Li, Z. Al-Huda, W.-D. Wang, Enhanced pavement crack segmentation with minimal labeled data: a triplet attention teacher-student framework, International Journal of Pavement Engineering 2024, VOL. 25 (2024) 2400562. doi:10.1080/10298436.2024.2400562.

[14] Z. Jian, J. Liu, Cross teacher pseudo supervision: Enhancing semi-supervised crack segmentation with consistency learning, Adv. Eng. Inform. 59 (C) (Jan. 2024). doi:10.1016/j.aei.2023.102279. URL https://doi.org/10.1016/j.aei.2023.102279

[15] H. Feng, W. Li, Z. Luo, Y. Chen, S. Fatholahi, M. Cheng, C. Wang, J. M. Junior, J. Li, Gcn-based pavement crack detection using mobile lidar point clouds, IEEE Transactions on Intelligent Transportation Systems 23 (2021) 11052–11061. doi:10.1109/tits.2021.3099023.

[16] H. Feng, L. Ma, Y. Yu, Y. Chen, J. Li, Scl-gcn: Stratified contrastive learning graph convolution network for pavement crack detection from mobile lidar point clouds, International Journal of Applied Earth Observation and Geoinformation 118 (2023) 103248. doi:https://doi.org/10.1016/j.jag.2023.103248. URL https://www.sciencedirect.com/science/article/pii/S1569843223000705

[17] M. Sohaib, M. J. Hasan, M. Shah, Z. Zheng, A robust self-supervised approach for fine-grained crack detection in concrete structures, Scientific Reports 14 (06 2024). doi:10.1038/s41598-024-63575-x.

[18] K. Zhang, Y. Zhang, H. Cheng, Self-supervised structure learning for crack detection based on cycle-consistent generative adversarial networks, Journal of Computing in Civil Engineering 34 (2020) 04020004. doi:10.1061/(ASCE)CP.1943-5487.0000883.

[19] S. Karimijafarbigloo, R. Azad, A. Kazerouni, D. Merhof, Ms-former: Multi-scale self-guided transformer for medical image segmentation, in: I. Oguz, J. Noble, X. Li, M. Styner, C. Baumgartner, M. Rusu, T. Heinmann, D. Kontos, B. Landman, B. Dawant (Eds.), Medical Imaging with Deep Learning, Vol. 227 of Proceedings of Machine Learning Research, PMLR, 2024, pp. 680–694.
URL https://proceedings.mlr.press/v227/karimijafarbigloo24a.html

[20] C. V. Dung, L. D. Anh, Autonomous concrete crack detection using deep fully convolutional neural network, Automation in Construction 99 (2019) 52–58. doi:https://doi.org/10.1016/j.autcon.2018.11.028.
URL https://www.sciencedirect.com/science/article/pii/S0926580518306745

[21] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, H. Ling, Feature pyramid and hierarchical boosting network for pavement crack detection, IEEE Transactions on Intelligent Transportation Systems 21 (4) (2020) 1525–1535. doi:10.1109/TITS.2019.2910595.

[22] Y. Liu, J. Yao, X. Lu, R. Xie, L. Li, Deepcrack: A deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139–153. doi:https://doi.org/10.1016/j.neucom.2019.01.036.
URL https://www.sciencedirect.com/science/article/pii/S0925231219300566

[23] M. D. Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy, G. Morison, A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks, in: 2018 26th European signal processing conference (EUSIPCO), IEEE, 2018, pp. 2120–2124.

[24] Y. Pan, G. Zhang, L. Zhang, A spatial-channel hierarchical deep learning network for pixel-level automated crack detection, Automation in Construction 119 (2020) 103357. doi:https://doi.org/10.1016/j.autcon.2020.103357.
URL https://www.sciencedirect.com/science/article/pii/S0926580520309377

[25] F. Guo, Y. Qian, J. Liu, H. Yu, Pavement crack detection based on transformer network, Automation in Construction 145 (2023) 104646. doi:https://doi.org/10.1016/j.autcon.2022.104646.
URL https://www.sciencedirect.com/science/article/pii/S0926580522005167

[26] H. Liu, J. Yang, X. Miao, C. Mertz, H. Kong, Crackformer network for pavement crack segmentation, IEEE Transactions on Intelligent Transportation Systems 24 (9) (2023) 9240–9252. doi:10.1109/TITS.2023.3266776.

[27] B. A. Kyem, E. K. O. Denteh, J. K. Asamoah, K. A. Tutu, A. Aboah, Advancing pavement distress detection in developing countries: A novel deep learning approach with locally-collected datasets, ArXiv abs/2408.05649 (2024).
URL https://api.semanticscholar.org/CorpusID:271854773

[28] Z. Al-Huda, B. Peng, R. N. A. Algburi, S. Alfasly, T. Li, Weakly supervised pavement crack semantic segmentation based on multi-scale object localization and incremental annotation refinement, Applied Intelligence 53 (11) (2023) 14527–14546. doi:10.1007/s10489-022-04212-w.
URL https://doi.org/10.1007/s10489-022-04212-w

[29] T. He, H. Li, Z. Qian, C. Niu, R. Huang, Research on weakly supervised pavement crack segmentation based on defect location by generative adversarial network and target reoptimization, Construction and Building Materials 411 (2024) 134668. doi:https://doi.org/10.1016/j.conbuildmat.2023.134668.
URL https://www.sciencedirect.com/science/article/pii/S0950061823043891

[30] S. Shim, J. Kim, G.-C. Cho, S.-W. Lee, Multiscale and adversarial learning-based semi-supervised semantic segmentation approach for crack detection in concrete structures, IEEE Access 8 (2020) 170939–170950. doi:10.1109/ACCESS.2020.3022786.

[31] T. Shi, Y. Wang, Y. Fang, Y. Zhang, Semi-supervised segmentation model for crack detection based on mutual consistency constraint and boundary loss, Engineering Applications of Artificial Intelligence 139 (2025) 109683. doi:https://doi.org/10.1016/j.engappai.2024.109683.
URL https://www.sciencedirect.com/science/article/pii/S0952197624018414

[32] C. Han, H. Yang, T. Ma, S. Wang, C. Zhao, Y. Yang, Crackdiffusion: A two-stage semantic segmentation framework for pavement crack combining unsupervised and supervised processes, Automation in Construction 160 (2024) 105332. doi:https://doi.org/10.1016/j.autcon.2024.105332.
URL https://www.sciencedirect.com/science/article/pii/S0926580524000682

[33] T. Uelwer, J. Robine, S. S. Wagner, M. Höftmann, E. Upschulte, S. Konietzny, M. Behrendt, S. Harmeling, A survey on self-supervised representation learning (2023). arXiv:2308.11455.
URL https://arxiv.org/abs/2308.11455

[34] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features (2019). arXiv:1807.05520.
URL https://arxiv.org/abs/1807.05520

[35] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, M. Goldblum, A cookbook of self-supervised learning (2023). arXiv:2304.12210.
URL https://arxiv.org/abs/2304.12210

[36] D. A. Beyene, D. Q. Tran, M. B. Maru, T. Kim, S. Park, S. Park, Unsupervised domain adaptation-based crack segmentation using transformer network, Journal of Building Engineering 80 (2023) 107889. doi:https://doi.org/10.1016/j.jobe.2023.107889.
URL https://www.sciencedirect.com/science/article/pii/S2352710223020697

[37] Q. D. Nguyen, H.-T. Thai, S. D. Nguyen, Self-training method for structural crack detection using image blending-based domain mixing and mutual learning, Automation in Construction 170 (2025) 105892. doi:https://doi.org/10.1016/j.autcon.2024.105892.
URL https://www.sciencedirect.com/science/article/pii/S0926580524006289

[38] Z. Lin, H. Wang, S. Li, Pavement anomaly detection based on transformer and self-supervised learning, Automation in Construction 143 (2022) 104544. doi:https://doi.org/10.1016/j.autcon.2022.104544.
URL https://www.sciencedirect.com/science/article/pii/S0926580522004150

[39] A Self-Supervised Learning Technique for Road Defects Detection Based on Monocular Three-Dimensional Reconstruction, Vol. Volume 3: 21st International Conference on Advanced Vehicle Technologies; 16th International Conference on Design Education of International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. arXiv:https://asmedigitalcollection.asme.org/IDETC-CIE/proceedings-pdf/IDETC-CIE2019/59216/V003T01A021/6453236/v003t01a021-detc2019-98135.pdf, doi:10.1115/DETC2019-98135.
URL https://doi.org/10.1115/DETC2019-98135

[40] H.-Y. Yoon, J.-H. Kim, J.-W. Jeong, Classification of the sidewalk condition using self-supervised transfer learning for wheelchair safety driving, Sensors 22 (1) (2022). doi:10.3390/s22010380.
URL https://www.mdpi.com/1424-8220/22/1/380

[41] Q. Song, W. Yao, H. Tian, Y. Guo, R. C. Muniyandi, Y. An, Two-stage framework with improved u-net based on self-supervised contrastive learning for pavement crack segmentation, Expert Syst. Appl. 238 (Part F) (2024) 122406.
URL https://doi.org/10.1016/j.eswa.2023.122406

[42] N. Ma, R. Fan, L. Xie, Up-cracknet: Unsupervised pixel-wise road crack detection via adversarial image restoration, IEEE Transactions on Intelligent Transportation Systems 25 (2024) 13926–13936.
URL https://api.semanticscholar.org/CorpusID:267311537

[43] S. Karimijafarbigloo, R. Azad, A. Kazerouni, D. Merhof, Ms-former: Multi-scale self-guided transformer for medical image segmentation, in: I. Oguz, J. Noble, X. Li, M. Styner, C. Baumgartner, M. Rusu, T. Heinmann, D. Kontos, B. Landman, B. Dawant (Eds.), Medical Imaging with Deep Learning, Vol. 227 of Proceedings of Machine Learning Research, PMLR, 2024, pp. 680–694.
URL https://proceedings.mlr.press/v227/karimijafarbigloo24a.html

[44] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Transactions on Intelligent Transportation Systems 17 (12) (2016) 3434–3445.

[45] L. Zhang, F. Yang, Y. D. Zhang, Y. J. Zhu, Road crack detection using deep convolutional neural network, in: Image Processing (ICIP), 2016 IEEE International Conference on, IEEE, 2016, pp. 3708–3712.

[46] Q. Zou, Y. Cao, Q. Li, Q. Mao, S. Wang, Cracktree: Automatic crack detection from pavement images, Pattern Recognition Letters 33 (3) (2012) 227–238.

[47] S. Ham, S. Bae, H. Kim, I. Lee, G.-P. Lee, D. Kim, Training a semantic segmentation model for cracks in the concrete lining of tunnel, Journal of Korean Tunnelling and Underground Space Association (2021) 549–558doi:10.9711/KTAJ.2021.23.6.549.

[48] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Transactions on Intelligent Transportation Systems 17 (12) (2016) 3434–3445.

[49] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, H.-M. Gross, How to get pavement distress detection ready for deep learning? a systematic approach., in: International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2039–2047.

[50] M. Pak, S. Kim, Crack detection using fully convolutional network in wall-climbing robot, in: J. J. Park, S. J. Fong, Y. Pan, Y. Sung (Eds.), Advances in Computer Science and Ubiquitous Computing, Springer Singapore, Singapore, 2021, pp. 267–272.

[51] R. Amhaz, S. Chambon, J. Idier, V. Baltazart, Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection, IEEE Transactions on Intelligent Transportation Systems 17 (10) (2016) 2718–2729. doi:10.1109/TITS.2015.2477675.

[52] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014) 3431–3440.
URL https://api.semanticscholar.org/CorpusID:1629541

[53] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M.

Wells, A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[54] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, J. Liang, Unet++: Redesigning skip connections to exploit multiscale features in image segmentation, IEEE Transactions on Medical Imaging 39 (6) (2020) 1856–1867. doi:10.1109/TMI.2019.2959609.

[55] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6230–6239. doi:10.1109/CVPR.2017.660.

[56] H. Li, P. Xiong, J. An, L. Wang, Pyramid attention network for semantic segmentation, in: Proceedings of the British Machine Vision Conference (BMVC), 2018, p. 285. doi:10.48550/arXiv.1805.10180.

[57] R. LI, S. Zheng, C. Zhang, C. Duan, J. Su, L. Wang, P. Atkinson, Multiattention network for semantic segmentation of fine-resolution remote sensing images, IEEE Transactions on Geoscience and Remote Sensing PP (2021) 1–13. doi:10.1109/TGRS.2021.3093977.

[58] A. Chaurasia, E. Culurciello, Linknet: Exploiting encoder representations for efficient semantic segmentation, in: 2017 IEEE Visual Communications and Image Processing (VCIP), 2017, pp. 1–4. doi:10.1109/VCIP.2017.8305148.

[59] A. Kirillov, R. Girshick, K. He, P. Dollar, Panoptic feature pyramid networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6392–6401. doi:10.1109/CVPR.2019.00656.

[60] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, ArXiv abs/1706.05587 (06 2017). doi:10.48550/arXiv.1706.05587.
URL `https://api.semanticscholar.org/CorpusID:22655199`

[61] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Com-

puter Vision – ECCV 2018, Springer International Publishing, Cham, 2018, pp. 833–851.

[62] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, J. Sun, Unified perceptual parsing for scene understanding, in: European Conference on Computer Vision, 2018, pp. 432–448.
URL `https://api.semanticscholar.org/CorpusID:50781105`

[63] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Álvarez, P. Luo, Segformer: Simple and efficient design for semantic segmentation with transformers, in: Neural Information Processing Systems, 2021, pp. 12077–12090.
URL `https://api.semanticscholar.org/CorpusID:235254713`