CrackFormer: Transformer Network for Fine-Grained Crack Detection

Huajun Liu[1*], Xiangyu Miao[1], Christoph Mertz[2], Chengzhong Xu[3], Hui Kong[3*]
[1]Nanjing University of Science and Technology,[2]Carnegie Mellon University,[3]University of Macau
{liuhj, miaoxy}@njust.edu.cn, cmertz@andrew.cmu.edu, {czxu, huikong}@um.edu.mo

Abstract

Cracks are irregular line structures that are of interest in many computer vision applications. Crack detection (e.g., from pavement images) is a challenging task due to intensity in-homogeneity, topology complexity, low contrast and noisy background. The overall crack detection accuracy can be significantly affected by the detection performance on fine-grained cracks. In this work, we propose a Crack Transformer network (CrackFormer) for fine-grained crack detection. The CrackFormer is composed of novel attention modules in a SegNet-like encoder-decoder architecture. Specifically, it consists of novel self-attention modules with 1x1 convolutional kernels for efficient contextual information extraction across feature-channels, and efficient positional embedding to capture large receptive field contextual information for long range interactions. It also introduces new scaling-attention modules to combine outputs from the corresponding encoder and decoder blocks to suppress non-semantic features and sharpen semantic ones. The CrackFormer is trained and evaluated on three classical crack datasets. The experimental results show that the CrackFormer achieves the Optimal Dataset Scale (ODS) values of 0.871, 0.877 and 0.881, respectively, on the three datasets and outperforms the state-of-the-art methods.

1. Introduction

Pavement crack detection from images is a challenging issue due to intensity inhomogeneity, topology complexity, low contrast, and noisy texture background [18]. In addition, crack's diversity (thin, grid or thick crack etc.) makes it more difficult.

There are a large number of studies on crack detection [6, 22, 2, 36, 37, 35, 10]. Recent studies have employed convolutional neural networks (CNNs) to boost detection accuracy to a higher level. In this study, we consider the problem of detecting thin cracks from the image of an asphalt surface. In general, it is much easier to detect thick cracks than thin cracks. Thus, crack detection performance is largely affected by how well one method can detect thin cracks.

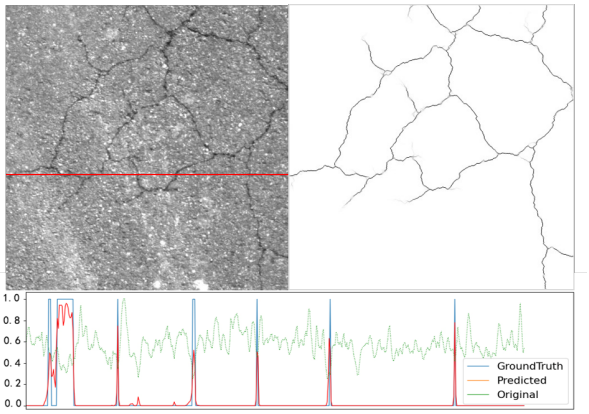

Figure 1. Crack prediction from our CrackFormer model (Best viewed in color). The upper left is a classical crack image. The upper right is the predicted result. The bottom shows a profile slice with normalized grey scale, its ground truth and corresponding crack predicted probabilities.

The state-of-the-art (SOTA) methods heavily rely on Fully Convolutional Networks (FCNs) [9], such as SegNet [31], U-Net [27] and their variants [21]. SegNets and U-Nets use an encoder-decoder architecture, where the encoder extracts high-level semantic representations by using a cascade of convolution and pooling layers, and the decoder leverages memorized pooling indices or skip connections to re-use high-resolution feature maps from the encoder in order to recover lost spatial information from high-level representations. Despite their outstanding performance, these methods suffer from limitation in complex segmentation tasks, e.g. when dealing with thin cracks or when there exists low contrast between crack and background. In general, these models rely on stacked 3×3 convolution and pooling operations, and could not achieve pixel-level segmentation precision in the convolution-pooling pipeline, resulting in blur and coarse crack segmentation. Moreover, suffering from the limited receptive field by using 3×3 convolutional kernels, these

*Corresponding author

methods tend to fail in detecting long cracks, resulting in discontinuous crack detection.

In this work, we propose a Crack Transformer network (CrackFormer) by combining novel self-attention and scaling-attention mechanisms for crack detection. It explores to leverage the merits of Transformer models [30] to capture long-range interactions and simultaneously adopt small convolution kernels for fine-grained attentive perception. CrackFormer keeps the regular layout by using a SegNet-like architecture, but introduces attention mechanisms in two different ways. Fig. 2 shows our network structure. The main contribution of this paper can be summarized as follows,

1. A new self-attention block (Self-AB) is proposed (Fig. 3). The Self-AB can fully extract contextual information across feature-channels by leveraging the 1×1 convolution kernels, and capture large receptive field contextual information across spatial-domain by an efficient position embedding.

2. A new scaling-attention block (Scal-AB) is proposed (Fig. 4), where a set of scaling-attention masks are generated by nonlinearizing the encoder's feature maps, and used to suppress non-semantic features and sharpen the semantic cracks.

3. We propose a Transformer encoder-decoder structure integrating the proposed Self-AB and Scal-AB blocks, where the Self-AB is embedded into different levels of the encoder and decoder modules, and the Scal-AB is introduced between the encoder feature maps and corresponding decoder ones.

A crack prediction result by our method is shown in Fig. 1, where the original image is shown in the upper left, the predicted result shown in the upper right. In the lower row, we can observe from the profile that the cracks are predicted precisely.

2. Related Work

Crack detection via classification - Since CNNs were introduced to pavement crack detection, research in this field has achieved significant progress [6, 22, 2, 36, 37, 35, 10]. Earlier works on crack detection are based on object detection pipeline for damage region-proposal and damage classification. For instance, the Faster R-CNN [6], YOLO [2], SSD Inception and SSD MobileNet [22] etc have ever been used for pavement damage-region extraction. Although these bounding-box based methods can detect crack regions reasonably well, they do not provide as precise information, e.g., crack's width and shape etc, as pixel-wise segmentation methods do.

Crack detection via segmentation - Since Zhang et al. [36] proposed pixel-level asphalt crack detection based on CNN models, some more accurate methods analyze pavement damage using deep neural networks [37, 35, 10]. For example, Liu et al. [21] proposed a pyramid features aggregation network and a Condition Random Fields (CRFs) post-processing scheme for crack segmentation. Zou et al. [37] provided a multi-stage fusion on the SegNet encoder-decoder architecture for crack segmentation. Yang et al. [35] proposed a feature pyramid and hierarchical boosting network for pavement crack detection, which integrates context information to low-level features for crack detection in a feature-pyramid way. Fei et al. [10] proposed the CrackNet-V model, which stacks several 3×3 convolutional layers and a 15×15 convolution kernel for deep abstraction to achieve a high performance for crack segmentation. Although these segmentation-based crack detection methods have obtained promising results, they cannot afford to achieve satisfying performance at the pixel-level segmentation precision and result in blur and coarse segmentation.

Self attention - Very recently, the Transformer [30]'s self-attention mechanism [28, 4, 7, 26, 3] has been adopted or improved on image segmentation task. The DANet [11] proposed a parallel position-attention and channel-attention enhanced FCN, but its computational complexity $O((HW)^2C) + O(HWC^2)$ is high. The CCNet [15] harvests the contextual information in horizontal and vertical directions to enhance pixel-wise representative capability and works more efficient than non-local block [33]. The Axial-attention [32] shows that self-attention layers alone could be stacked to form a fully attentional model by restricting the receptive field of self-attention to a local square region for panoptic segmentation.

Moreover, it has been shown [7] that multi-head self-attention layer with sufficient number of heads is at least as expressive as any convolutional layer. Exploration on replacing 3×3 convolution kernels in popular backbones, such as the ResNet [13] etc, by a self-attention augmented convolution model [4], a stand-alone self-attention model [26] or a Lambda-attention layer [3] has yielded remarkable gains. These self-attention works, with merits of long-distance interaction, local receptive field, computation- and memory-efficiency, have inspired us to explore more efficient and effective self-attention mechanism for crack segmentation task.

Scaling attention - The self attention is effective for global-dependency modeling, and it is likely to be valuable for connected and long-range crack segmentation. However, the self-attention only might not be enough for fine-grained cracks, which can be strongly affected by noisy background. Therefore, we seek the help from scaling-attention mechanism.

Scaling attention focuses on emphasizing semantic features and suppressing non-semantic ones. For example, the

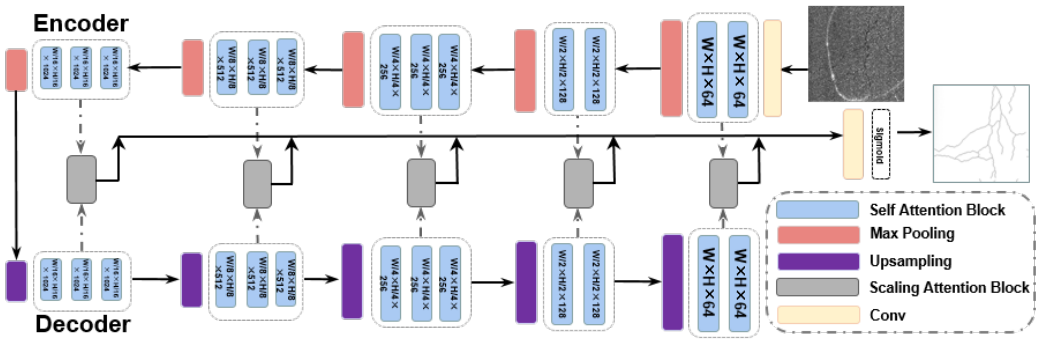

Figure 2. The structure of Crack Transformer network.

attention-gate mechanism [23] identifies salient image regions and prune feature responses to preserve only the activation relevant to the specific task to boost segmentation performance. The squeeze-and-excitation (SE) [14] module uses global average pooling and a linear layer to compute a scaling factor for each channel and then scales the channels accordingly. Convolutional block attention module (CBAM) [24] added global max pooling in addition to global average pooling and an extra spatial attention submodule to compute scaling factors on channel and spatial domain separately. The Spatial attention [12] and multi-scale attention on the U-Net [5] combine local features with their corresponding global dependencies, explicitly modeling the dependencies between channels and different scale spatial information in segmentation task. Oktay et al. [23] propose a soft-attention mechanism to softly weight the encoder and decoder features at each pixel location. These scaling-attention or attention-gate methods operate on a local receptive field is helpful for sharpening semantic feature and suppress non-semantic ones by a soft mask after Sigmoid normalization.

3. Our work

3.1. Overview

The CrackFormer adopts the basic structure of the SegNet [31], shown in Fig.2. To establish long-range interaction between the low-level feature maps, we propose novel self-attention blocks as the basic module. To enhance crack crispness, we introduce a local attention block between encoder and the corresponding decoder features to generate attention masks. Finally, multi-stage side fusion is exploited between feature maps of different stages to fuse coarse to fine cracks to obtain a refined result.

Similar to the SegNet, the CrackFormer's encoder consists of the first 13 convolutional layers of the VGG16 [29] network, and they are deployed in 5 stages according to the layout of $\{2, 2, 3, 3, 3\}$. Meanwhile, the corresponding decoder has a symmetrical layout of $\{3, 3, 3, 2, 2\}$. In contrast, the 3×3 convolution module at each layer of SegNet is replaced by the self-attention block in the CrackFormer (Sec.3.2).

At the end of each stage, a maximum pooling with a 2×2 window and stride of 2 (non-overlapping window) is used to reduce the size of feature maps by one half. The max-pooling indices, i.e, the locations of the maximum feature value in each pooling window, are memorized for each encoder feature map. The appropriate decoder upsamples its input feature map(s) using the memorized max-pooling indices from the corresponding encoder feature map(s). At each stage, the corresponding encoder and decoder features are concatenated to generate an attentive mask by a scaling-attention block to refine each tensor of each stage (Sec.3.3).

The predicted results in all stages are then fused to generate the final result. The predicted results in each stage and fused features are resized to the original dimension of the input image, and the model is supervised by a multi-loss function in training phase.

3.2. Self Attention for Long Range Capture

In the CrackFormer, the self-attention block (Self-AB) in encoder and decoder is a bottleneck module with two CBR (Conv-BatchNorm-ReLU) blocks, composed of 1×1 conv, BatchNorm and ReLU, and a self-attention layer (SAL in short) between them (Fig.3(b)).

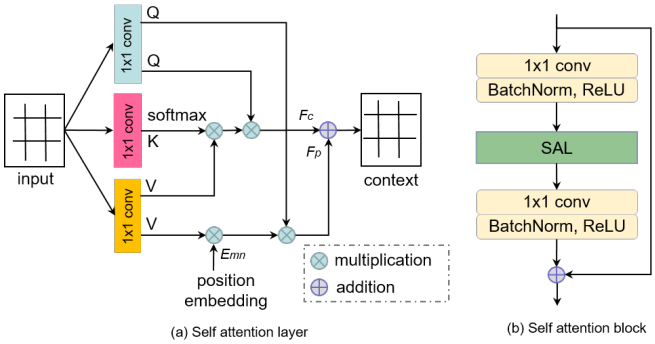

Figure 3. The self-attention block and self-attention layer.

The self-attention layer is a position embedded self-attention block simultaneously with large receptive field and 1x1 convolution kernels, which is shown as Fig. 3(a). Let $X \in \mathbb{R}^{d_{in} \times WH}$ be an input tensor, where W and H represent the width and height of image and d_{in} denotes the channel of input tensor. This layer applies three 1×1 convolutions to generate the keys, queries, and values to generate new features $\boldsymbol{F}^c \in \mathbb{R}^{d_{out} \times WH}$ (d_{out} denotes the channel of output features) using the following content-based multi head self-attention operation,

$$\boldsymbol{F}^c = \boldsymbol{Q} \otimes \left(\sigma \left(\boldsymbol{K}^\top \right) \otimes \boldsymbol{V} \right), \tag{1}$$

where $\otimes$ is a matrix multiplication operation. Let h be the number of head, d_u be intra-depth dimension, r be the receptive field size, d_k and d_v be the dimension of tensor K and V, respectively. Then we have $\boldsymbol{Q} \in \mathbb{R}^{d_k \times h \times WH}$, $\boldsymbol{K} \in \mathbb{R}^{d_k \times d_u \times WH}$, and $\boldsymbol{V} \in \mathbb{R}^{d_v \times d_u \times WH}$. Let σ denote the operation of applying softmax normalization on the tensor. This attention operation can be interpreted as first aggregating the pixel features in $\boldsymbol{V}$ into global context vectors using the weights in $\sigma \left(\boldsymbol{K}^\top \right)$, and then redistributing the global context vectors back to individual pixels using the weights in $\boldsymbol{Q}$. We notice its similarity to the one used in Bello [3], but it does not use batch normalization on queries and values. Softmax normalizing on the keys constrains the output features to be convex combinations of the global context vectors.

The relative position embedding can make the global context vector obtain an effective receptive field in a neighbour region. A relative positional embedded kernel $\boldsymbol{E}^r_{mn} \in \mathbb{R}^{d_k \times d_u \times r \times r}$ is defined as learnable weight parameters, where r indexes the possible relative positions for all (n, m) pairs. The contextual vector $\boldsymbol{E}^r_{nm}$ as a convolutional kernel is embedded to the context vectors according to the following equation,

$$\boldsymbol{F}^p = \boldsymbol{Q} \otimes \left(\otimes_{\boldsymbol{E}^r_{nm}} (\boldsymbol{V}) \right) \tag{2}$$

Finally, the output context vector of the self-attention layer is the element-wise addition of the global content vector and position embedded vector, $\boldsymbol{F}^n = \boldsymbol{F}^c \oplus \boldsymbol{F}^p$, where $\oplus$ is the matrix element-wise addition operator. Note that the computational and memory complexities of this layer are $O(N)$ in the number of pixels.

3.3. Scaling Attention for Sharpening Crack

At each stage, the feature vectors in the encoder and decoder are connected and combined according to the scaling-attention block (Scal-AB) to generate the salient and crisp crack boundary map (Fig. 4). The attention-gate mechanism in Attention U-Net [23] inspires that the feature vectors in a specific decoder block can boost segmentation performance by combining the feature vectors in the corresponding encoder block. In essence, the attention-gate mechanism generates an attentive mask, which is normalized to $\alpha_i \in [0, 1]$ by a Sigmoid activation function, and multiplies element-wise to those features to be refined. In this way, it acts as a filter to activate some features within a region of interest and simultaneously suppress other irrelevant features.

Thus, we propose a scaling-attention block on the encoder and decoder features. Specifically, at each stage of the CrackFormer, we use features in encoder to generate an attentive mask as attention coefficients and multiply them element-wise to the corresponding features in decoder to active crack features and suppress the non-crack ones.

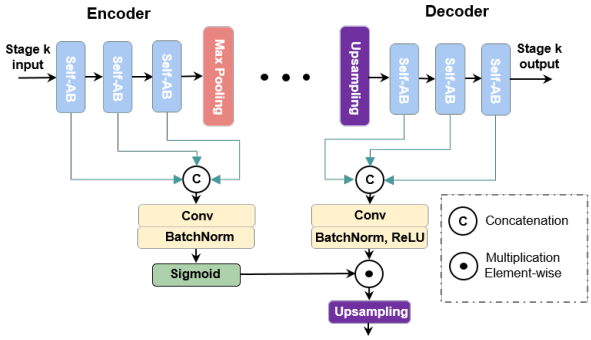

Figure 4. The scaling-attention block.

Let's take the k^{th}-stage fusion as an example, features from the encoder and decoder are $\{X_1^k, X_2^k, X_3^k\}$ and $\{Y_1^k, Y_2^k, Y_3^k\}$, respectively. Based on Fig. 4, the mask is generated according to

$$L^k_{Mask} = \delta \left(BN \left(\otimes_{3\times3} \left(\Gamma \left(X_1^k, X_2^k, X_3^k \right) \right) \right) \right) \tag{3}$$

where $\Gamma(\cdot)$ denotes a tensor concatenation operation, and $\otimes_{3\times3}(\cdot)$ denotes a 3×3 convolution operation and followed by a BatchNorm BN, and $\delta(\cdot)$ is a Sigmoid activation function. Subsequently, side output of the k^{th} stage is predicted by the scaling-attention mechanism as follows,

$$S^k_{side} = L^k_{Mask} \odot BN \left(\otimes_{3\times3} \left(\Gamma \left(Y_1^k, Y_2^k, Y_3^k \right) \right) \right) \tag{4}$$

where $\odot$ denotes an element-wise multiplication operation. The scaling-attention maps at each stage are visualised as Fig. 5, which is an attention coefficient mask from high-level features. Around the semantic crack, there is a stronger response. From the output features of different stages, we can see a coarse to fine semantic crack response, which can be used to refine more crisp boundary.

After the feature in each stage is upsampled in order to make its dimension the same as that of the input image, we get five predicted results $S^k_{side}, k = 1, 2, \cdots, 5$, which are

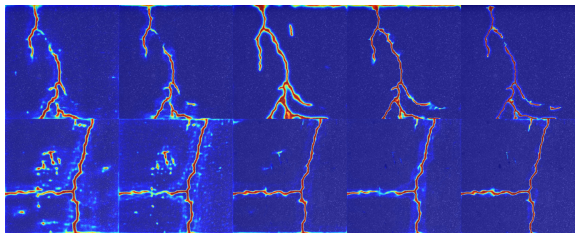

Figure 5. From left to right: the scaling-attention maps from stage 1 to stage 5, respectively.

concatenated and fused to generate the final output S_{fuse} like the HED [34], RCF [19] and DeepCrack [37] etc. Finally, all sides and fused output are fully supervised by the crack ground truth labels.

3.4. Loss Function

The balanced weight cross-entropy loss function ever used in the RCF network [20] is adopted with a small modification for training, where the pixels in the label whose normalized intensities y_i are higher than η are considered as positive samples, and the pixels with probability between 0 and 0.05 as negative samples, and pixels in between as neglected.

$$l\left(X_i; W\right) = \begin{cases} \alpha \log\left(1 - P\left(X_i; W\right)\right) & 0 \leq y_i \leq 0.05 \\ 0 & 0.05 < y_i \leq \eta \\ \beta \log P\left(X_i; W\right) & y_i > \eta \end{cases} \tag{5}$$

where $\alpha = \lambda \frac{|Y^+|}{|Y^+|+|Y^-|}$, and $\beta = \frac{|Y^-|}{|Y^+|+|Y^-|}$, with $|Y^+|$ and $|Y^-|$ representing the number of positive and negative samples, respectively. λ is used to balance the loss ratio of positive and negative samples. Let X_i be the value of pixel i, and y_i be the probability of pixel i in the labeled image, and $P\left(X_i; W\right)$ representing the predicted probability of the pixel being crack, and W the weight of the model.

To reweigh each side output in training process, we weigh the loss on different side outputs, and increase the weights in the last two sides and the fusion side. The total loss function is

$$\mathrm{L(W)} = \sum_{i=1}^{n} \left(\sum_{k=1}^{5} S_{side}^{k} \cdot l\left(X_i^k; W\right) + S_{fuse} \cdot l\left(X_i^{\text{fuse}}; W\right) \right), \tag{6}$$

where $S_{side}^{k}, k \in \{1, 2, 3, 4, 5\}$, represents the loss weight of the k_{th} stage, S_{fuse} the loss weight of the fusion layer, n the total number of pixels in each sample, and k the number of side outputs, respectively.

4. Implementation Details

Data Augmentation - We augment the training set by random clipping, flipping, and rotation operations. We also use Gamma transformation on the training images to reduce the influence of brightness. In the end, we expand each training set by 228 times of the original samples.

Training & Validation Parameters - To improve the robustness of the model, the images in the training set hold its original dimension and have not been resized. The *BatchSize* in the experiment is set to 1, and the *Shuffle* strategy is set True. We choose the *Stochastic Gradient Descent (SGD)* as optimizer and set the MOMENTUM to 0.9.

Due to the data augmentation, the total training epoch is set to 500 and the initial learning rate is set to 1e-3. We adopt the *StepLR* strategy to adjust the learning rate at epoch 20, 50 and 100. At each epoch milestone, the learning rate will decay $1/10$ times of the previous one.

5. Experiments

5.1. Datasets

Our model is trained and evaluated on three public benchmarks, the CrackTree260, CrackLS315 and Stone331.

The CrackTree260 [25] contains 260 road pavement images. These pavement images are captured by an area-array camera under visible-light illumination, and the size of each sample is 800×600. 200 samples are chosen for training, 20 samples for validation and 40 samples for testing.

The CrackLS315 [37] contains 315 images of asphalt pavement captured under laser illumination by a line-array camera. Each image has a size of 512×512. Among them, 265 samples are selected for training, and the remaining 10 samples for validation and 40 samples for testing.

The Stone331 [17] contains 331 images of the stone surface, captured by an area-array camera under visible-light illumination. Original image size is 1024×1024, because of the irregularity of cutting surface, original images are center-cropped to 512×512 clipped samples. 261 images of them are chosen for training, 20 for validation and 50 for testing.

5.2. Performance Metrics

The performance metrics of $Precision$ (abbre. as PR) and $Recall$ (abbre. as RE) are calculated as $PR = \frac{TP}{TP+FP}$ and $RE = \frac{TP}{TP+FN}$ for binary classification tasks.

Specifically, for each image, PR and RE can be calculated by comparing the detected cracks against the human annotated ground truth. Then, the F-measure ($\frac{2 \cdot PR \cdot RE}{PR+RE}$) can be computed as an overall metric for performance evaluation. Specifically, three different F-measure-based metrics are employed in the evaluation, the best F-measure on the data set for a fixed threshold - Optimal Dataset Scale (ODS), the aggregate F-measure on the data set for the best threshold on each image - Optimal Image Scale (OIS), and

the average precision (AP), which is equivalent to the area under the precision-recall curve [8].

5.3. Comparison with the SOTA methods

To evaluate our model's performance, some classical models, such as the SE [8], HED [34], RCF [20], SegNet [31], SRN [16], U-Net [27], FPHBN [35] and DeepCrack [37] are compared with ours on crack detection task. The SE [8] is a classical method based on random decision forest used for edge detection. The HED [34] is a model based on the VGG16, whose feature maps are generated at each stage of the VGG16 and aggregated for multi-stage fusion. The RCF [20] and SRN [16] are similar with the HED, which is an extension of the HED. The SegNet [31] and U-Net [27] are encoder and decoder architecture with symmetrical structures. The DeepCrack [37] is an extension to the SegNet for crack detection.

5.3.1 The Results on the CrackTree260

The CrackTree260 is a thin crack dataset labeled with a single pixel width or extremely tiny edges. On the asphalt surface and under visible-light illumination, the crack exhibits extreme weak contrast between the "crack" and "non-crack" pixels.

Model	ODS↑	OIS↑	AP↑	FLOPs↓	mPara↓
SE [8]	0.662	0.673	0.683	-	-
FPHBN [35]	0.517	0.579	-	-	-
SRN [16]	0.774	0.781	0.779	451.3G	28.5M
HED [34]	0.816	0.820	0.831	146.9G	14.7M
SegNet [1]	0.844	0.851	0.862	311.3G	29.5M
U-Net [27]	0.847	0.832	0.869	400.0G	31.0M
RCF [20]	0.857	0.863	0.861	187.9G	14.8M
DeepCrack [37]	0.852	0.864	0.875	1001.7G	30.9M
CrackFormer	**0.881**	**0.883**	**0.896**	**123.0G**	**7.35M**

Table 1. Performance on the CrackTree260.

From the precision-recall curves in Fig. 9(a) and statistical performance in Tab. 1, it can be seen that the CrackFormer outperforms the compared SOTA methods on the CrackTree260, with 0.881 on ODS, 0.883 on OIS and 0.896 on AP, respectively. We obtain a gain of 2.9% on ODS, 2.3% on OIS and 2.1% on AP, respectively. compared with the DeepCrack. Visualized results in Fig. 6 show that the CrackFormer's results are more continuous and crisp than the compared deep learning models. The crack profile shows that the CrackFormer can achieve high prediction accuracy even for cracks with one-pixel width or tiny edge.

5.3.2 The Results on the CrackLS315

The images of this dataset are captured under laser illumination. The training on this dataset is more difficult than on the other datasets because of the extreme low contrast. The precision-recall curves are shown in Fig. 9(b).

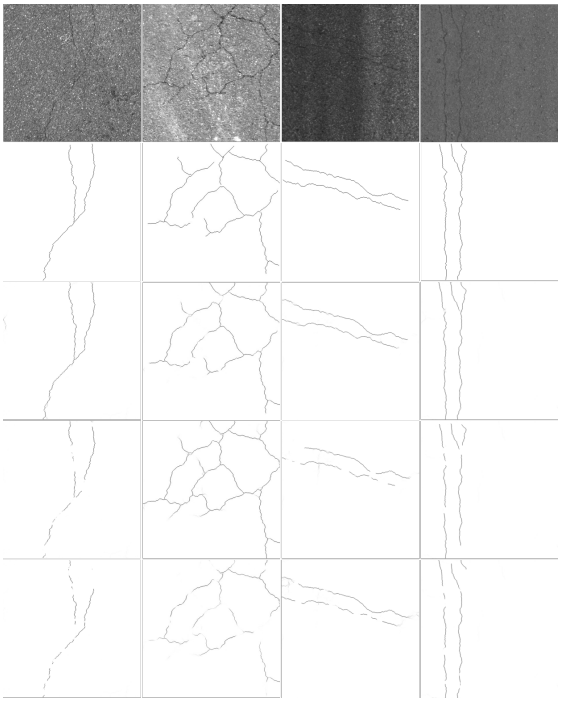

Figure 6. Predicted results on the CrackTree260. From top to bottom row: the original crack images, the ground truth, the results by the proposed CrackFormer, the results by DeepCrack [37], the results by RCF [19], respectively.

Model	ODS↑	OIS↑	AP↑	FLOPs↓	mPara↓
SE [8]	0.459	0.521	0.495	-	-
U-Net [27]	0.672	0.703	0.740	218.6G	31.0M
SRN [16]	0.755	0.789	0.795	246.6G	28.5M
SegNet [1]	0.761	0.780	0.780	170.1G	29.5M
HED [34]	0.763	0.798	0.829	80.3G	14.7M
RCF [20]	0.788	0.816	0.829	102.7G	14.8M
DeepCrack [37]	0.853	0.867	0.877	547.4G	30.9M
CrackFormer	**0.871**	**0.879**	**0.883**	**67.2G**	**7.35M**

Table 2. Performance on the CrackLS315.

It can be seen from Tab. 2 that the CrackFormer achieves the best performance on the CrackLS315. It obtains a gain of 1.8% on ODS, 1.2% on OIS, 0.6% on AP, respectively, compared with the DeepCrack. The ODS of the HED, SRN, SegNet and U-Net, is 10.8%, 11.6%, 11.0% and 19.9% lower than the CrackFormer, respectively. Compared with the method SE, the DeepCrack obtains an improvement of 41.2% in terms of ODS. The HED, SRN, RCF and SegNet show comparable results, while the CrackFormer has better performance than these methods. Visualized results in Fig. 7 (seen as the middle row) show that the CrackFormer can predict more detailed thin crack from low contrast asphalt pavements.

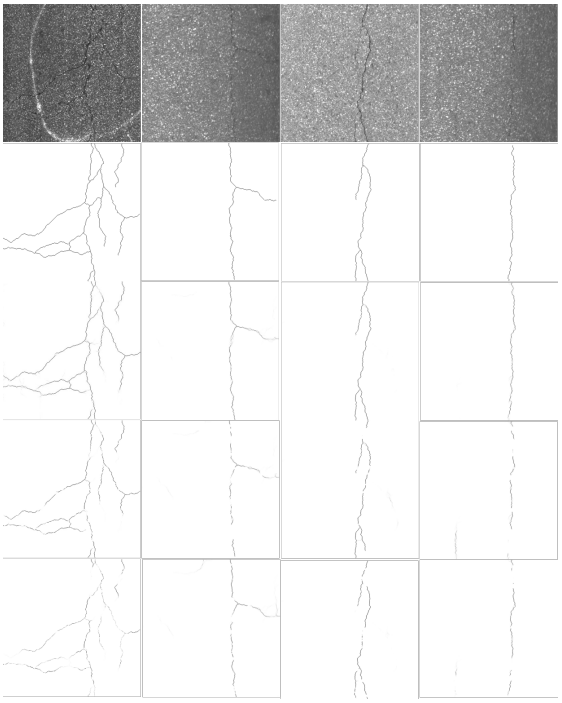

Figure 7. Predicted results on the CrackLS315. From top to bottom row: the original crack images, the ground truth, the results by the CrackFormer, the results by the DeepCrack [37], the results by the RCF [19], respectively.

5.3.3 The Results on the Stone331

This dataset is from stone cutting surface and its smooth surface makes the crack texture too weak to be observed even by human eyes. The visualized results in Fig. 8 (seen as the first row) show that the CrackFormer can predict the most continuous and complete crack detection results. It can be seen from precision-recall curves in Fig. 9(c), the CrackFormer outperforms the other compared methods.

Model	ODS↑	OIS↑	AP↑	FLOPs↓	mPara↓
SE [8]	0.557	0.623	0.605	-	-
HED [34]	0.719	0.763	0.758	80.3G	14.7M
SRN [16]	0.735	0.776	0.741	246.6G	28.5M
U-Net [27]	0.757	0.776	0.809	218.6G	31.0M
RCF [20]	0.789	0.829	0.820	102.7G	14.8M
SegNet [1]	0.794	0.815	0.787	170.1G	29.5M
DeepCrack [37]	0.856	0.875	0.888	547.4G	30.9M
CrackFormer	**0.877**	**0.885**	**0.894**	**67.2G**	**7.35M**

Table 3. Performance on the Stone331.

From statistical performance in Tab. 3, the CrackFormer achieves an ODS of 0.877, 0.885 OIS and 0.894 AP, respectively, on the test dataset. The CrackFormer obtains a gain of 2.1% on ODS, 1.0% on OIS and 0.6% on AP, respectively, compared with the DeepCrack. Compared with the mainstream deep learning models, it outperforms by 8.3%, 8.8%, 12.0% and 14.2% on ODS over the SegNet, RCF, U-Net and SRN, respectively. Compared with traditional method SE, the CrackFormer obtains an improvement of 32% in terms of ODS.

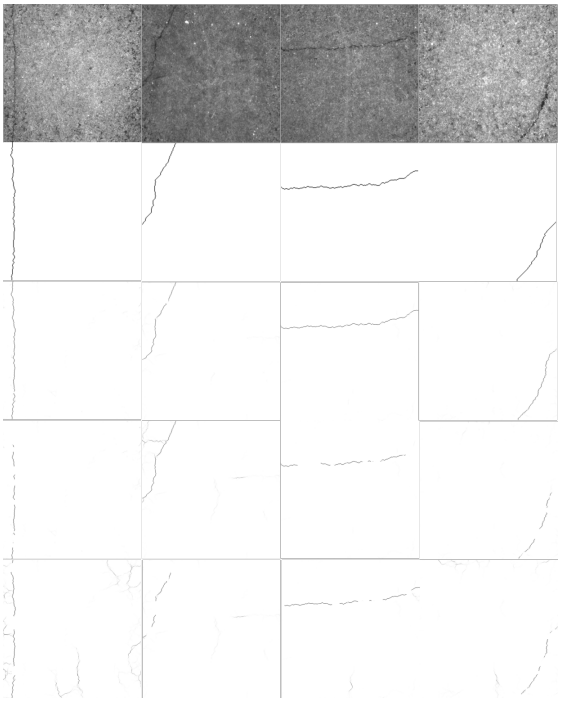

Figure 8. Predicted results on the Stone331. From top to bottom: the original crack images, the ground truth, the results by the CrackFormer, the results by the DeepCrack [37], the results by the RCF [19], respectively.

5.4. Multi-scale Analysis

The multi-scale fusion scheme has proven to be an effective way to enhance crack detection performance [18]. In fact, because crack images exhibit different characteristics at different scales. At a large-scale stage, crack detection is reliable, but its localization is poor and may miss thin cracks. At a small-scale stage, details are preserved, but detection suffers a lot from clutters in background texture. Therefore, we quantitatively analyze output of different-scale stage and scale-wise fusion performance on the three datasets. The statistical results are shown in Tab. 4. Overall, the ODS and OIS values increase step by step from stage S1 to S5, and we obtain a 9.4% ODS gain in average. This means that the output of the CrackFormer from coarse to fine scale (stage) gradually matches the true scale of this kind of thin crack benchmark. From the viewpoint of multi scale fusion, it can be found that the incremental fusion experiments from S1+S2 to S1+S2+S3+S4, or even to all scale fusion (S1+S2+S3+S4+S5) could increase the ODS and OIS values over the output of each single scale. Moreover, the final fused results can further obtain the ODS

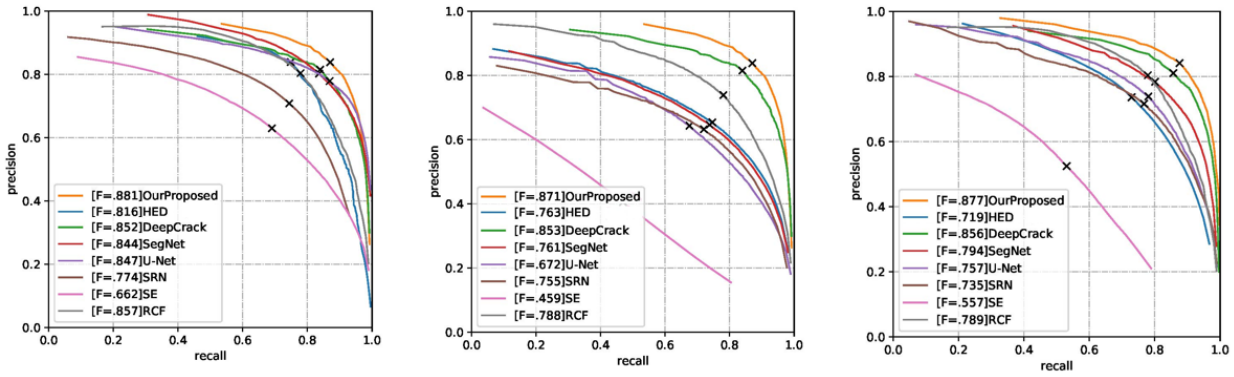

Figure 9. The precision-recall curves on the CrackTree260, CrackLS315 and Stone331, respectively.

2* Scale	CrackTree260		CrackLS315		Stone331	
	ODS↑	OIS↑	ODS ↑	OIS↑	ODS↑	OIS↑
S1	0.680	0.702	0.648	0.671	0.760	0.771
S2	0.709	0.723	0.691	0.632	0.769	0.775
S3	0.740	0.742	0.746	0.652	0.779	0.812
S4	0.756	0.761	0.755	0.661	0.796	0.821
S5	0.799	0.801	0.761	0.665	0.809	0.815
S1+S2	0.768	0.772	0.735	0.742	0.815	0.820
S1+S2+S3	0.802	0.818	0.809	0.821	0.821	0.823
S1+S2+S3+S4	0.854	0.857	0.828	0.835	0.851	0.867
Fused	**0.881**	**0.883**	**0.871**	**0.879**	**0.877**	**0.883**

Table 4. Multi-the scale analysis on the three datasets.

gain over the finest scale (S5) by 8.7% in average.

5.5. Efficiency Analysis

The FLOPs test and parameters calculation on the compared models are shown from Tab. 1 to Tab. 3 with different inference image sizes (600×800, 512×512 and 512×512, respectively). It shows that the CrackFormer is more efficient and requires fewer parameters. Specifically, the CrackFormer achieves higher accuracy than the the DeepCrack [37] with 8.1x fewer FLOPs and 4.2x fewer parameters. Compared to the other classical models, the CrackFormer achieves a higher ODS value, 2x to 3x faster in average and 2x to 3x fewer parameters.

5.6. Ablation Study

To further check the gain of each module of our model, ablation study is done on the CrackLS315. The experimental results are shown in Tab. 5. We first select the SegNet as the baseline. After the conv 3×3 is replaced by the Self-AB in the encoder and decoder, the gain on ODS and OIS is 9.8% and 8.9%, respectively, showing that the self-attention block is effective for fine-grained crack representation. Similarly, the Scal-AB can get a 9.7% ODS gain and an 8.9% OIS gain independently. Furthermore, compared with the DeepCrack, after the Self-AB is applied to the DeepCrack, the gain on ODS and OIS is 1.1% and 0.3%, respectively. In addition, the Scal-AB block can get a gain of 0.9% and 0.5% on ODS and OIS, respectively, indicating that the model works better on multi-scale fusion architecture as well. Finally, the Self-AB and Scal-AB modules further achieve a gain of $0.7\% - 0.9\%$ and $0.7\% - 0.9\%$ on ODS and OIS, respectively, indicating that the two attention mechanisms are compatible in crack detection task.

Model	Self-AB	Scal-AB	MSF	ODS ↑	OIS ↑
SegNet				0.761	0.780
DeepCrack			✓	0.853	0.867
-	✓			0.859	0.869
-		✓		0.858	0.869
-	✓		✓	0.864	0.870
-		✓	✓	0.862	0.872
CrackFormer	✓	✓	✓	0.871	0.879

Table 5. Ablation study on the CrackLS315.

6. Conclusion

The proposed CrackFormer aims at detecting fine-grained cracks. We derive our model from the SegNet basic architecture and novel attention mechanisms. The proposed self-attention modules are embedded in the encoder-decoder blocks, where the 1x1 convolutional kernels are adopted for extracting contextual information across feature-channels, and efficient positional embedding to capture large receptive field spatial contextual information for long range interactions. The proposed scaling-attention modules combine output from the corresponding encoder and decoder, and enable to obtain crisp crack boundary. On three classical crack detection benchmark datasets, the CrackTree, CrackLS315 and Stone331, we can obtain pixel-level crack detection accuracy and achieve SOTA performance.

References

[1] Krizhevsky Alex, Sutskever Ilya, and Hinton Geoffrey E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84–90, 2017.

[2] Abdullah Alfarrarjeh, Dweep Trivedi, Seon Ho Kim, and Cyrus Shahabi. A deep learning approach for road damage detection from smartphone images. In *IEEE International Conference on Big Data (Big Data)*, 2018.

[3] Irwan Bello. Lambdanetworks: Modeling long-range interactions without attention. In *ICLR*, 2021.

[4] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. In *ICCV*, 2019.

[5] Yutong Cai and Yong Wang. Ma-unet: An improved version of unet based on multi-scale and attention mechanism for medical image segmentation, 2020. arXiv:2012.10952.

[6] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361–378, 2017.

[7] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *ICLR*, 2020.

[8] Piotr Dollár and C. Lawrence Zitnick. Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1558–1570, 2015.

[9] Cao Vu Dung and Le Duc Anh. Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, 99:52–58, 2018.

[10] Yue Fei, Kelvin C. P. Wang, Allen Zhang, Cheng Chen, Joshua Q. Li, Yang Liu, Guangwei Yang, and Baoxian Li. Pixel-level cracking detection on 3d asphalt pavement images through deep-learning-based cracknet-v. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):273–284, 2020.

[11] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019.

[12] Changlu Guo, Márton Szemenyei, Yugen Yi, Wenle Wang, Buer Chen, and Changqi Fan. Sa-unet: Spatial attention u-net for retinal vessel segmentation. In *ICPR*, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. In *CVPR*, 2018.

[15] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *CVPR*, 2019.

[16] Wei Ke, Jie Chen, Jianbin Jiao, Guoying Zhao, and Qixiang Ye. Srn: Side-output residual network for object symmetry detection in the wild. In *CVPR*, 2017.

[17] Jacob Koniga, Mark David Jenkinsa, Mike Manniona, Peter Barriea, and Gordon Morisona. Optimized deep encoder-decoder methods for crack segmentation, 2020. arXiv:2008.06266v1.

[18] Haifeng Li, Dezhen Song, Yu Liu, and Binbin Li. Automatic pavement crack detection by multi-scale image fusion. *IEEE Trans. on Intelligent Transportation System*, 20(6):2025–2036, 2019.

[19] Yun Liu and Ming-Ming Cheng. Richer convolutional features for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:1936–1946, 2019.

[20] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *CVPR*, 2017.

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

[22] Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiyama, and Hiroshi Omata. Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, 33(12):1127–1141, 2018.

[23] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas, 2018. arXiv:1804.03999.

[24] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In SoKweon. Bam: bottleneck attention module. In *BMVC*, 2018.

[25] QinZou, YuCao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33:227–238, 2012.

[26] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. In *NIPS*, 2019.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[28] Zhuoran Shen, Mingyuan Zhang, Shuai Yi, Junjie Yan, and Haiyu Zhao. Efficient attention: Self attention with linear complexities. In *arXiv:1812.01243*, 2018.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit abd Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[31] Badrinarayanan Vijay, Kendall Alex, and Cipolla Roberto. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 39(12):2481–2495, 2017.

[32] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *ECCV*, 2020.

[33] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[34] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *CVPR*, 2015.

[35] Fan Yang, Lei Zhang, Sijia Yu, Danil Prokhorov, Xue Mei, and Haibin Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1525–1535, 2019.

[36] Allen Zhang, Kelvin C. P. Wang, Baoxian Li, and Enhui Yang. Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network. *Computer Aided Civil Infrastructure Engineering*, 32(10):805–819, 2017.

[37] Qin Zou and Zheng Zhang. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28:1498–1512, 2019.