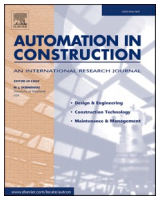
Automation in Construction 166 (2024) 105646

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon



Crack segmentation using discrete cosine transform in shadow environments

Yingchao Zhang , Cheng Liu *

Department of Systems Engineering, City University of Hong Kong, Hong Kong

ARTICLE INFO

Keywords:
Non-destructive testing
Deep learning
Crack segmentation
Shadow removal

ABSTRACT

Accurate pavement crack segmentation is crucial for quantifying the extent of pavement damage. However, shadows from roadside trees or buildings often significantly impact the results of crack segmentation in actual crack detection or segmentation processes. To solve this problem, this paper presents a pavement crack segmentation network called SCSNet based on discrete cosine transform. SCSNet combines a proposed shadow removal module with a loss function based on pixel frequency distribution to further minimize the impact of shadows on accuracy. Additionally, a crack dataset with shadows was introduced. By comparing with the classical semantic segmentation network, the results show that SCSNet with training from scratch, outperforms classic models based on pre-trained weights. The result of the ablation experiments also demonstrate that the proposed tricks are effective. Finally, the actual crack segmentation results further demonstrate the superiority of SCSNet in segmenting cracks in complex environments.

1. Introduction

Nowadays, most countries in the world have built high-density highway networks, which provide excellent convenience for transportation and economic development. As highways are used frequently, cracks of various shapes and sizes develop in the pavement. Overloaded vehicles can accelerate this process, causing these cracks to become larger damage. The presence of pavement damage can significantly diminish driving comfort, accelerate the wear and tear of vehicles, and even result in loss of vehicle control, thereby increasing the likelihood of traffic accidents. Therefore, timely detection of pavement cracks plays a vital role in maintaining highway safety and the healthy development of transportation.

The main detection methods for pavement damage are manual inspection, multi-functional inspection cars, and inspection methods based on computer vision or deep learning. The manual can flexibly adjust the inspection strategy according to the actual situation of the highway. For the unpredictable damage situation, the manual inspection is more adaptable to the changes in the field. Furthermore, it does not rely on specific technology or equipment and is not hindered by equipment or technology failure. However, manual inspection of roads is time-consuming, and inefficient, and requires the road to be closed during the inspection. As a result, multi-functional inspection cars are gradually replacing human labor to detect road damage. These cars are equipped with advanced sensors that can detect road damage with high precision, thus reducing the error that can occur in the manual inspection. Multi-functional inspection cars also have certain limitations, high costs, and complex operation processes that make it difficult to be massively expanded. For some complex defects, manual confirmation is still needed. Therefore, more and more scholars have begun to explore simple but efficient pavement damage detection methods [1,2]. Computer vision technology can solve this problem well.

Many scholars have used convolution neural networks (CNN) for the detection and segmentation of pavement cracks, potholes, and other damage [3–6]. [7] used a Generative Adversarial Network (GAN) to generate pavement texture images to improve the classification accuracy of pavement defects. Zhang et al. [8] used a modified YOLO v3 model to realize the detection of pavement cracks and potholes. Following the introduction of the dataset containing pavement damage from multiple countries [9], the Institute of Electrical and Electronics Engineers (IEEE) organized an international competition focused on the detection of road damage [10]. In this competition, different teams used different strategies to improve the accuracy of detection. Data augmentation [11], ensemble learning [12,13], and attention mechanisms [14,15] can significantly improve the accuracy of the model in road disease detection. Among all the participants, [12] achieved an

* Corresponding author.
E-mail address: cliu647@cityu.edu.hk (C. Liu).

https://doi.org/10.1016/j.autcon.2024.105646
Received 22 May 2024; Received in revised form 21 July 2024; Accepted 22 July 2024
Available online 1 August 2024

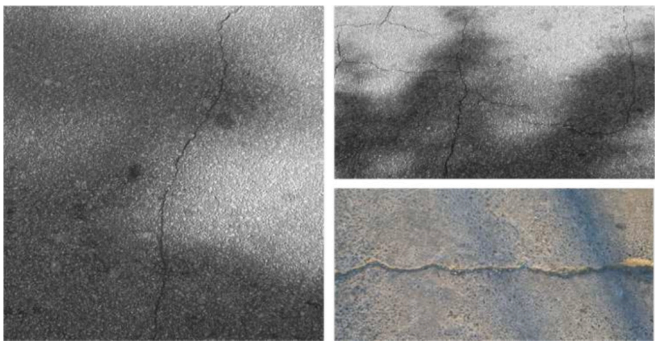


Fig. 1. Crack images with shadows.

accuracy of 0.77 in the dataset of 6 countries and took the first place.

The studies mentioned above only involved detecting the location and category of the defects in the pictures, without categorizing them at the pixel level. Therefore, more people started using semantic segmentation models to segment damage. Choi et al. [16] proposed SDDNet based on standard convolution, modified spatial pyramid pooling module, and densely connected separable convolution, which can realize real-time crack segmentation. Zhang and Liu [17] combined deformable convolution [18] with the U-Net network [19] to achieve precise segmentation of pavement cracks. Guan et al. [20] combined 2D and 3D images and proposed a U-Net segmentation network based on depthwise separable convolution to realize the three-dimensional segmentation of pavement cracks and potholes. YOLO v3 [21] and Faster RCNN [22] were used by Liu et al. [23] to perform a two-stage crack detection and segmentation task, which achieved the state of the art at that time.

However, these models were trained on datasets collected under normal conditions, meaning good lighting and no distractions. During the practical detection activities, it is common to encounter shadows caused by trees, vehicles, and other objects, which can be seen in Fig. 1. These shadows can significantly impact the detection and segmentation accuracy.

Shadow crack images can be classified into six categories since trees or objects, such as buildings, on both sides of the road that create shadows:

1. Images with large areas of block shadows.
2. Images with many strips shadows.
3. Images with many scattered shadows.

These shadows can affect the gradient, brightness, and other features of crack or pothole edges, potentially leading to misidentification. To solve this problem, local outlier factor (LOF) was used by Wang et al. [24] to remove the shadow and enhance the defects information. Ju et al. [25] used an illumination compensation model and *k*-means clustering algorithm to detect pavement cracks under the influence of shadows with a 93.86% F-measure score. [26] combined the wavelet transform with the Retinex algorithm, thus compensating for the information lost in the wavelet transform, and the whole system achieved a crack recognition accuracy of 95.8%. And a crack segmentation method based on the grayscale standard deviation of the local window and the distance standard deviation of the connected regions was proposed by [27]. Pavement cracks are detected with an accuracy of 96%. Although these methods effectively avoid the influence of shadows on crack edge features, the artificial design of filtering algorithms is very complex and less robust. In addition, image quality may degrade, and classification or segmentation accuracy may be affected by direct digital image

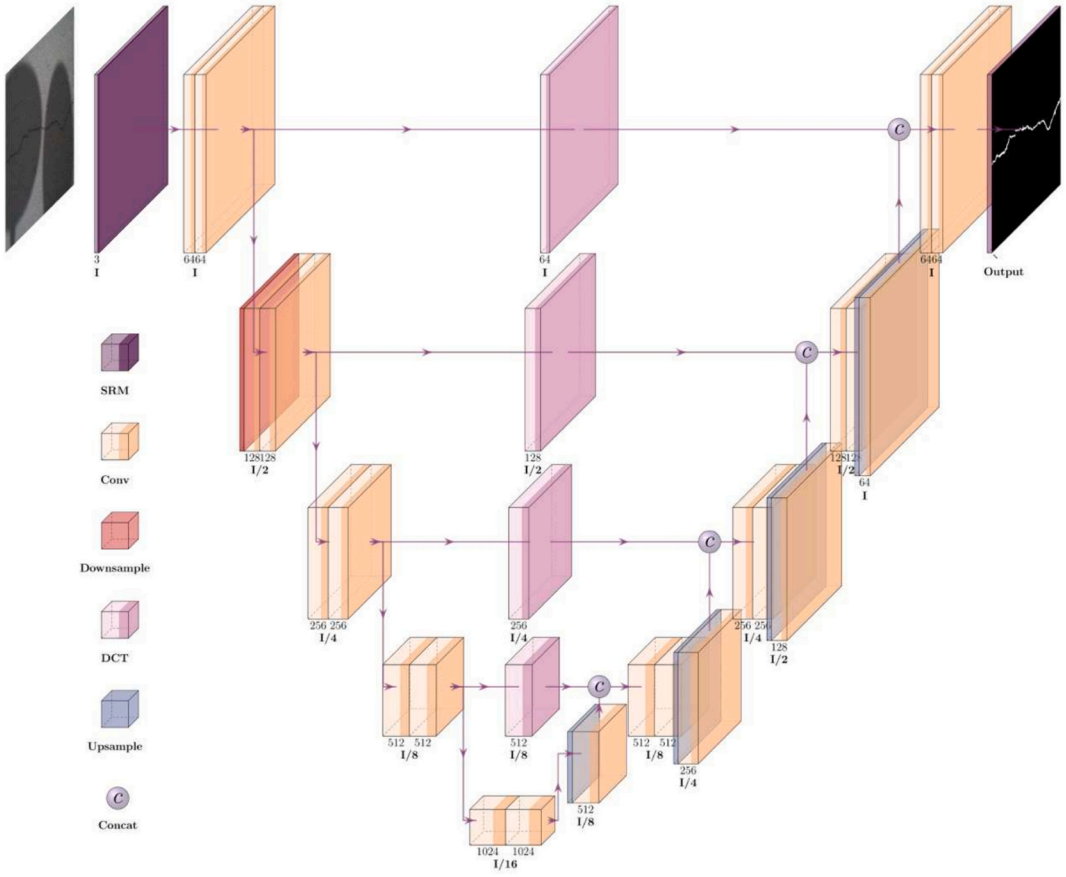


Fig. 2. Network structure of SCSNet.

processing operations on images containing shadows [28].

Deep learning-based shadow removal methods typically require simultaneous training on shadow and shadow-free images [29,30]. This means that a lot of effort needs to be spent on the dataset. To decrease the dataset requirement, Liu et al. [31] used the shadow image along with its mask for training and achieved shadow removal through weakly supervised learning. Le et al. [32] trained generative adversarial networks based on the physical principles of shadow generation to achieve competitive shadow removal results, reaching the state-of-the-art in video shadow removal.

The research methods mentioned above are commonly used in the field of computer vision. The shadows obtained from actual pavement damage detection equipment are often more complicated. At present, few studies directly use deep learning models for damage detection in shadows. The approach of some scholars is to divide the detection of defects into two parts: shadow removal and disease detection. [33] proposed a pavement shadow and crack dataset and a two-stage crack detection network. The network first performed shadow removal and then used a luminance difference algorithm to detect pavement cracks. [34] also performed noise or shadow removal in the first stage and finally detected cracks by CNN model. Zou et al. [35] developed a three-stage algorithm for crack detection. First, cracks were removed using a geodesic shading removal algorithm. Then, tensor voting was used to create a probabilistic map of the cracks. Finally, a graphical model was used to determine the shape of the cracks.

At present, there are few studies on the use of deep learning techniques for detecting cracks in shaded environments. Previous studies have primarily employed a staged detection method, which can result in a significant reduction in inference speed. To solve this problem, we have integrated a shadow processing module into a deep neural network, which allows us to achieve crack segmentation in shaded environments in one stage.

The main contributions of this article are as follows:

1. A novel crack segmentation algorithm used in shadowed environments (SCSNet) was proposed, which mainly involved the discrete cosine transform, shadow removal, and pixel frequency loss components.
2. A dataset containing 4776 pavement cracks in shaded environments was proposed for training and evaluating different models.
3. The discrete cosine transform module (DCT) was proposed to maintain the important crack information while removing the redundant shadow and background information.
4. A dual-branch shadow removal module (SRM) based on depthwise separable convolution was proposed to reduce the impact of shadows on the effectiveness of crack segmentation.
5. Pixel frequency loss was proposed to evaluate the effectiveness of the output of SRM. This loss function is calculated by Kullback-Leibler Divergence to compare the pixel distribution after shadow removal with the pixel distribution of the shadow-free dataset.

The structure of the whole paper can be divided as follows: Section 2 focuses on the structure of the SCSNet, Section 3 includes the introduction of the dataset, Section 4 will present the details of the evaluation metrics and training, Section 5 contains the results and discussion, and Section 6 is the conclusion.

2. Our proposed method

2.1. The total model

The structure of SCSNet is introduced in this section and is shown in Fig. 2. Images of any size can be input into SCSNet and transformed into images of 512 × 512 resolution. The input image is first passed through SRM to minimize the effect of shadows on the cracks and then different features are extracted by convolutional module (Conv). In Fig. 2, I equal to 512.

SCSNet primarily comprises an encoding part and a decoding part. The encoding process extracts and compresses features from the input images, while the decoding process recovers these features and generates high-resolution segmented images, thereby achieving high-precision image segmentation. In the encoding part, the feature maps with resolutions of $I/2$, $I/4$, $I/8$, and $I/16$ are sequentially obtained after Conv and the downsampling module. These feature maps are then processed by the DCT to extract important channel feature information. In the decoding part, the feature map with the resolution of $I/16$ is sequentially upsampled to obtain the feature maps with the resolutions of $I/8$, $I/4$, $I/2$, and I, respectively. The feature maps processed by DCT are concatenated with the feature maps obtained from upsampling in the channel direction. In the end, the detection result map with a resolution of 512 × 512 is obtained.

There are three main loss functions used in the study, including focal loss, dice loss, and pixel frequency loss proposed by us. The formula is shown in (1)–(3). There is a class imbalance because of the large gap between the number of cracked pixels and the number of background pixels. Thus, focal loss is applied to solve this problem. Dice loss is mainly used to evaluate the similarity between predicted and true values.

$$L_{loss} = L_{fl} + L_{dl} + L_{pfl} \tag{1}$$

$$L_{fl} = \begin{cases} -(1-\widehat{p})^{\gamma} log\widehat{p} & \text{if } y = 1 \\ -\widehat{p}^{\gamma} log(1-\widehat{p}) & \text{if } y = 0 \end{cases} \tag{2}$$

$$L_{dl} = 1 - \frac{2^{*}pred \cap true}{pred \cup true} \tag{3}$$

where L_{fl}, L_{dl}, and L_{pfl} denote focal loss [36], dice loss, and pixel frequency loss [37], respectively. γ is equal to 2 in this study. *Pred* means the set of predicted values and *true* indicates the set of true values.

The specific details of SRM, DCT, and L_{kl} are introduced in the following sections.

2.2. Discrete cosine transform module

DCT is a variation technique commonly used in signal and image processing, especially in image compression. It can transform images from the spatial domain to the frequency domain, which helps to remove redundant information while maintaining important features. According to the conclusions of [38,39], the application of global average pooling in channel attention means that only the lowest frequency information is maintained, while all other frequency features are dropped. The DCT can capture the components of multiple frequencies in the channel, thus improving the feature extraction capability of the model. In this study, we use the DCT module in the proposed model to compress the channel information, preserve the information of the important channels, and overlook the information of the channels that are severely affected by shadows.

DCT can be realized by element-by-element multiplication, and it is differentiable, so it can be easily integrated into CNN. In two dimensions, the common DCT is shown (4)–(5).

$$B_{h,w}^{i,j} = cos\left(\frac{\pi h}{H}\left(i+\frac{1}{2}\right)\right) cos\left(\frac{\pi w}{W}\left(j+\frac{1}{2}\right)\right) \tag{4}$$

$$f_{h,w}^{2d} = \sum_{i=0}^{H-1}\sum_{j=0}^{W-1} x_{i,j}^{2d} B_{h,w}^{i,j} \tag{5}$$

where $h, i \in \{0, 1, \ldots, H-1\}, w, j \in \{0, 1, \ldots, W-1\}, x_{i,j}^{2d}$ represents the input feature map ($x_{i,j}^{2d} \in \mathbb{R}^{H \times W}$), H and W are the height and width of $x_{i,j}^{2d}$.

First, the feature map with shape (C, W, H) is divided into n parts, as

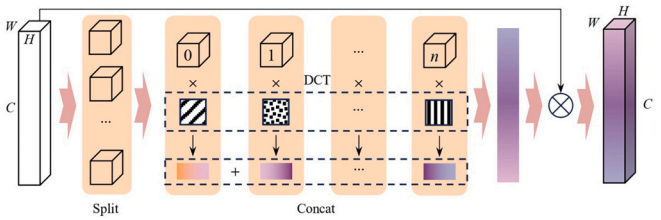


Fig. 3. Network structure of SCSNet.

shown in Fig. 3. Each part can be represented by x^i, and its size is (C', W, H), where $i \in \{0, 1, \ldots, n\}$ and $C' = \frac{C}{n}$. Then, each part of the feature map is assigned a DCT with different frequency components to compress the channel information, and this process can be described by (6).

$$X^i = \mathrm{DCT}^{u_i,v_i}(x^i) = \sum_{h=0}^{H-1}\sum_{w=0}^{W-1} x^i B_{h,w}^{u_i,v_i} s.t.i \in \{0, 1, \ldots, n-1\} \tag{6}$$

where u_i, and v_i are the frequency component 2D indices corresponding to x^i, and the size of X^i is $(C', 1, 1)$. In this study, n was set to 16, and the settings of the frequency components of the DCT are shown in Table.1. The values of frequency components are referenced to [38]. Many DCT components were proposed in that paper and later we would analyze the reasons for adopting that component.

After that, we can merge X^i in the channel direction to get the DCT results for the different channels, as shown in (7). Finally, F was sequentially passed through the Linear, ReLU, Linear, and Sigmoid layers to obtain feature maps of size (C,1,1). These feature maps were multiplied with the original input feature maps (X) to obtain the final results. The initial linear layer primarily serves to transform the feature representations. Subsequently, these features pass through a ReLU activation function, after which a second linear layer adjusts and strengthens them. The Sigmoid function scales the features to a range between 0 and 1, facilitating the generation of weight information for the feature points. In summary, these network layers enhance critical crack information while suppressing noise-related data such as shadows, thus improving model performance. The total process can be shown in (8) and Fig. 3.

$$F = \mathrm{Concat}(X^0, X^1, \ldots, X^{n-1}) \tag{7}$$

$$X_{output} = X \times \mathrm{Sigmoid}(\mathrm{Linear}(\mathrm{ReLU}(\mathrm{Linear}(F)))) \tag{8}$$

2.3. Shadow removal module

The input to the segmentation model is images of pavement cracks with shadows. To minimize the influence of shadows on the accuracy of crack segmentation, we propose a dual-branch shadow removal module (SRM) based on the idea of [32], the structure of which is shown in Fig. 5. We hypothesize that the channels contain varying degrees of shadow information. To prove this hypothesis, we plotted the histogram of the frequency distribution of the RGB channels of the crack image with shadows, as shown in Fig. 4. The observed similarity among the peaks of the RGB channels is noteworthy. The first significant peak consists mainly of the effect of shadows on the pixel frequencies, and we can find that the pixel frequencies around this peak are not the same, so we can prove that our hypothesis is correct. Thus, depthwise separable convolution (DSC) [40] was employed to integrate features across different channels.

In Fig. 5, the feature map with shape (3, H, W) was firstly passed through DSC to get the feature map of (4, H, W), then it was processed by normal convolution for shadow removal and kept the channels unchanged. Finally, DSC decreased the channels to 3 to get f_{sr}. The secondary branch primarily determines the importance of each channel through convolution and pooling, and the sigmoid function was used to get the importance coefficient w. Upon obtaining w and f_{sr}, the input feature map of the segmentation model was calculated by (9) with 3 channels.

$$x_{sr} = x \times (1 - w) + f_{sr} w \tag{9}$$

where x indicates the original input image, and w represents the importance coefficient.

To better explain the function of f_{sr}, w, and x_{sr} in Fig. 5, we plotted a

Table 1
Frequency components of DCT.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
u_i	0	0	6	0	0	1	1	4	5	1	3	0	0	0	3	2
v_i	0	1	0	5	2	0	2	0	0	6	0	4	6	3	5	2

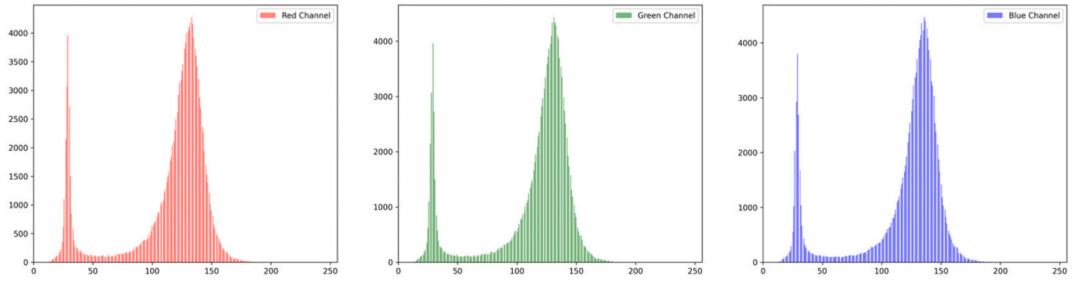


Fig. 4. Histogram of frequency distribution for different channels.

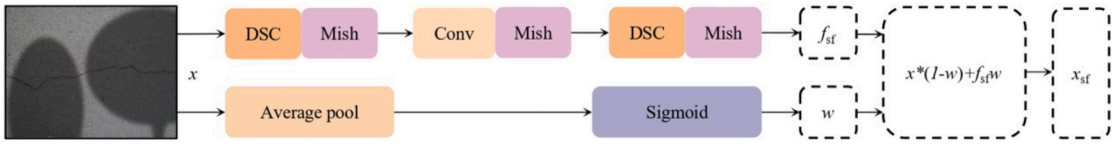


Fig. 5. Shadow removal module.

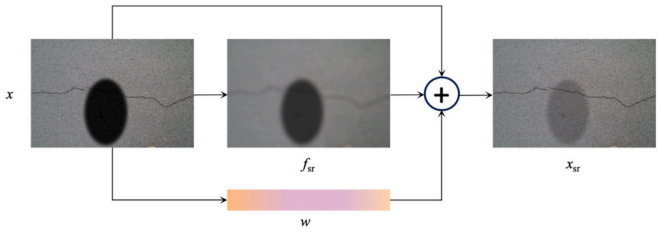


Fig. 6. Illustration of shadow removal process.

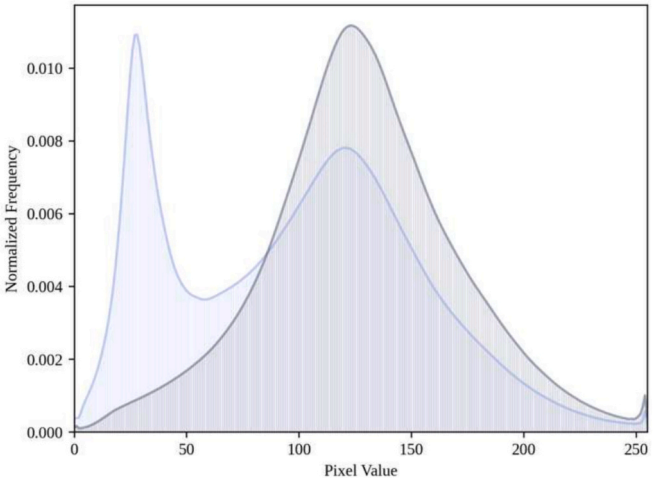


Fig. 7. Pixel frequency distribution histogram.

schematic diagram of the shadow removal process, as shown in Fig. 6. f_{sr} is primarily utilized to extract the feature information of shadows, while w is employed to emphasize the significance of the three channels. By integrating f_{sr}, w, and x_{sr}, shadow-attenuated images can be obtained.

2.4. Pixel frequency loss

In the previous study, we proposed a SRM module for diminishing the effect of shadows on crack segmentation. However, the effectiveness of the output of this module is difficult to evaluate, so we propose a pixel frequency loss (PF loss) for supervision of the effectiveness of the SRM.

We counted the pixel frequency distribution of all images in the crack dataset containing shadows, as shown in Fig. 7. The histogram of the frequency distribution of pixels in the shadow-free pavement crack image shows a single peak. However, the histogram exhibits a double peak distribution in the shadow dataset. This phenomenon is shown in Fig. 7. This is primarily because shaded areas are darker, leading to a lower peak in the histogram's gray value, whereas non-shaded areas, being brighter, exhibit a higher gray value peak. We expect the feature maps after SRM processing to have pixel distributions similar to those without shadows. Therefore, we used the Kullback-Leibler divergence to measure the difference between these two distributions, which can be seen in (10). L_{pfl} represents the pixel frequency loss. A smaller L_{kl} indicates a lower variance between the true value and the predicted value, and conversely, a larger L_{kl} signifies a greater discrepancy. Kullback-Leibler divergence is an asymmetric measure of the difference between two probability distributions. It is non-negative and equals zero if and only if $y_i = t_i$, representing the additional information required to encode data from distribution y_i using distribution t_i. KL divergence is widely used in model selection, feature selection, anomaly detection.

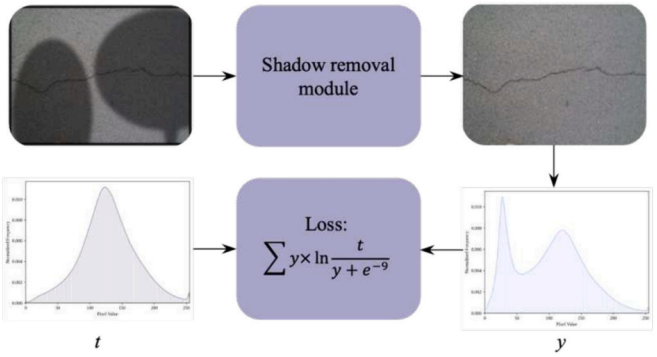


Fig. 8. Pixel frequency loss.

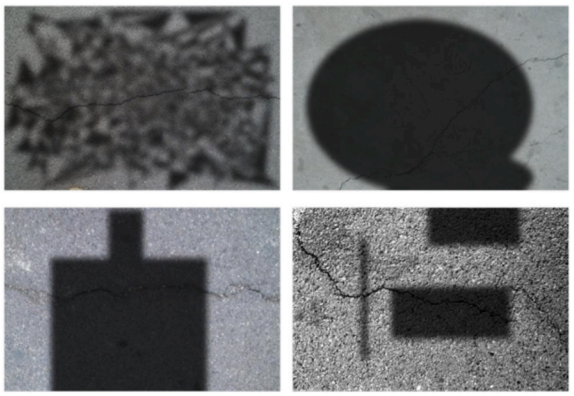


Fig. 9. Samples of shadowed crack dataset.

$$L_{pfl} = \sum_{i}^{255} y_i \times ln \frac{y_i}{t_i + e^{-9}} \tag{10}$$

where t means the output of SRM, and y is the ground truth. This process can be seen from Fig. 8.

3. Shadowed crack dataset

Upon reviewing the literature, it was found that the majority of road crack datasets were collected in well-lighted environments, such as Crack500 [41], Crack Forest Dataset (CFD) [42], and DeepCrack [43]. There are very few images in these datasets that have shadows, which makes it inconvenient to carry out crack segmentation studies under such conditions. Considering the difficulty of constructing a dataset from scratch, we added shadows to the images using digital image processing based on the open-access cracks image dataset.

We mixed Crack500 with the CFD dataset to get a total of 4776 crack images in jpg format. First, we created a 4-channel fully transparent image with the same size as the original image. Then we defined the color and transparency of the shadows in the transparent layer (it was set to 200, where 255 means completely opaque and 0 means completely transparent). Next, polygonal, elliptical, or rectangular shadows were randomly generated and applied to the full-transparent image. A Gaussian blur filter was then applied to process the shadow layer. After all this had been done, the original image was merged with the blurred shadow layer to get the shadow image in PNG format. During the training process, we reconverted the images from PNG format to JPG format. The whole process is shown in the code below. Examples of the

Table 2
Software and hardware versions of training.

Content	Versions	Content	Versions
CPU	Intel i9-13900K	Learning rate	1e-4
GPU	Nvidia RTX 4090	Batch size	8
Operating system	Ubuntu 20.04	Epochs	200
Python	3.8.18	Momentum	0.937
PyTorch	2.1.0	Number worker	4
RAM	128GB	Optimizer	Adam

generated data are shown in Fig. 9. We call this dataset the shadow crack dataset (SCD).

files, the labeling work was performed manually. The final dataset contains 4776 images. We divided the SCD dataset into training sets, validation sets, and test sets according to the ratio of 8:1:1.

4. Model training process

4.1. Software and hardware versions

All models in this study were trained under the following environments as shown in Table 2. In our initial experiments, SGD required significant hyperparameter tuning and exhibited slower convergence compared to Adam. Therefore, Adam optimizer was used by us.

```
Procedure Create_shadows(file)
    if file is jpg format then
        Open images
        Create shadow_layer with the same size as images, fully transparent
        Define shadow color with partial transparency (200)
        Define polygon, ellipse, and rectangle drawing subroutine
        Select a random drawing subroutine and execute it
        Apply Gaussian blur to shadow_layer
        Composite shadow_layer onto images
        Save the resulting image to disk with a new filename
    end if
end procedure
```

Common objects around the road include tree branches, road signs, and neighboring buildings, all of which can cause shadows of different shapes. Therefore, the shapes of the shadows created by us are similar to the common objects around the road, which can be seen in Fig. 9. To reduce the workload of data labeling, we used the matching labeled files in Crack500 and CFD datasets. For the images that lacked annotation

All models were trained for 200 epochs. The loss observed in both training and test sets is shown in Fig. 10, and we can see that after the number of training epochs is larger than 100, the loss value gradually tends to stabilize, which indicates that the model has reached a state of convergence.

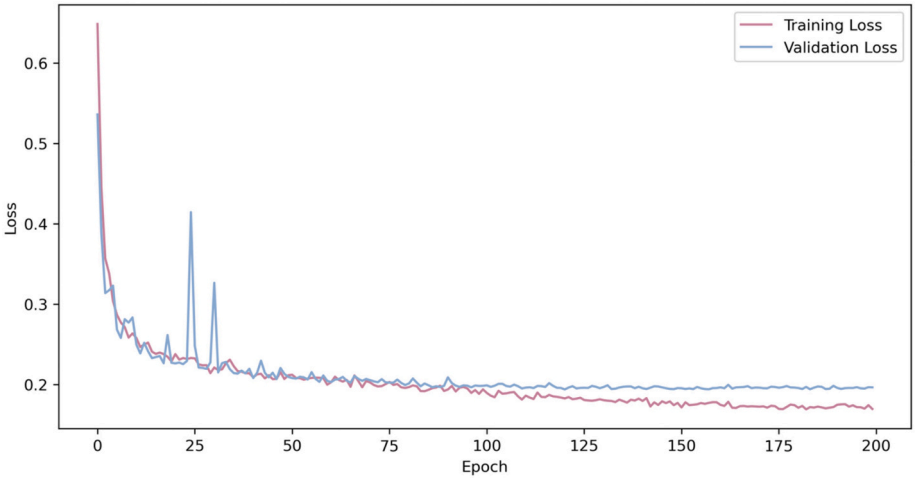


Fig. 10. Train and val loss during training.

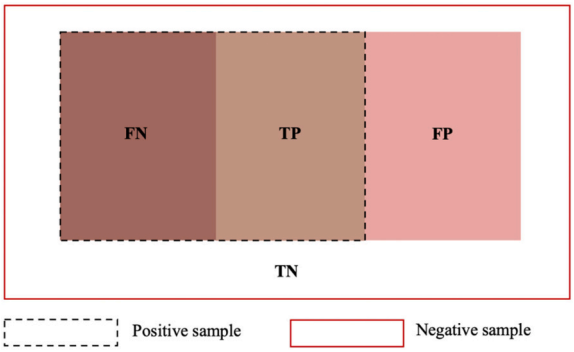


Fig. 11. Confusion matrix for binary classification.

4.2. Model evaluation metrics

In this study, *Accuracy*, mean Intersection over Union (*mIoU*), and mean Pixel Accurazcy (*mPA*) are used as the evaluation metrics. These metrics can be obtained based on true positives (*TP*), false negatives (*FN*), false positives (*FP*), and true negatives (*TN*). We can get the definition of the above variables according to Fig. 11. *TP* indicates correctly predicted positive samples, *TN* represents negative samples predicted correctly, *FP* denotes negative samples incorrectly predicted as positive, and *FN* refers to positive samples incorrectly predicted as negative.

$$P_c = \frac{TP}{TP + FP} \tag{11}$$

$$P_b = \frac{TN}{TN + FN} \tag{12}$$

$$mPA = \frac{1}{2}(P_c + P_b) \tag{13}$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{14}$$

$$IoU_c = \frac{TP}{TP + FN + FP} \tag{15}$$

$$IoU_b = \frac{TN}{TN + FN + FP} \tag{16}$$

$$mIoU = \frac{1}{2}(IoU_c + IoU_b) \tag{17}$$

The mPA can be obtained from (11)–(13), where P_c denotes the Intersection over Union (IoU) of the cracks and IoU_b represents the IoU of the background. Accuracy represents the ratio of the number of pixels correctly classified to the total pixel count, which can be seen from (14). The mIoU can be obtained from (15)–(17), where IoU_c denotes the category pixel accuracy of the cracks and IoU_b represents the category pixel accuracy of the background.

Table 3
Comparison of original dataset with SCD dataset.

Model	Pre-training	Dataset	mIoU	mPA	*Accuracy*
UNet [19]	√	shadow-free	79.58	87.80	97.91
	√	shadow	77.52	85.43	97.69
		shadow	75.24	83.38	97.41
Segformer [44]	√	shadow-free	78.53	84.25	97.73
	√	shadow	76.64	81.97	97.76
		shadow	64.40	68.25	96.60
PSPNet [45]	√	shadow-free	79.50	87.15	97.93
	√	shadow	78.28	85.58	97.81
		shadow	78.22	85.45	97.74
HRNet [46]	√	shadow-free	80.37	87.95	98.03
	√	shadow	78.66	85.79	97.87
		shadow	77.49	84.90	97.66
Deeplab v3+ [47]	√	shadow-free	80.67	89.07	98.02
	√	shadow	78.24	85.71	97.80
		shadow	77.17	85.43	97.56

Table 6
Ablation experiment.

Model	mIoU	mPA	Accuracy
Baseline	75.24	83.38	97.41
+DCT	77.92	86.19	97.69
+DCT + SRM	78.33	86.43	97.81
+DCT + SRM + PF Loss	**79.52**	**87.67**	**97.93**

5. Results and discussion

5.1. Comparison of the original dataset with the SCD dataset

In this section, we focus on comparing the performance of the segmentation model on the original dataset with the SCD dataset.

U-Net [19], Segformer [44], PSPNet [45], HRNet [46], and Deeplab V3+ [47] were used to verify that the existence of shadows affects the overall detection accuracy, and this result can be seen in Table.3. These models were trained on shadow-free dataset as well as shadow dataset, respectively. It is worth noting that all three metrics of detection are considerably lower when shadows are present than when there are no shadows. The presence of shadows typically reduces the mIoU values by about 2%, which is a very significant effect. Thus, the results demonstrate that the presence of shadows can cause misdetection of cracks, thus reducing accuracy. The principal reason for this outcome is that the edge characteristics of shadows are typically more similar to those of cracks, so models recognize the shadow edges as cracks.

In addition, the pre-training weights also have a significant impact on the segmentation accuracy of the cracks. Pre-training weights usually refer to the weight files obtained by training the model on large datasets such as COCO or VOC. These weights can speed up the training process to reach a converged state. The results can be seen from Table.3. For U-Net, PSPNet, HRNet, and Deeplab v3+, which are based on convolutional neural networks, the utilization of un-pretrained weights have been observed to result in a reduction in segmentation accuracy. Among these models, U-Net is the most affected, while PSPNet is the least affected.

For U-Net, PSPNet, HRNet, and Deeplab v3+ which are based on convolutional neural networks, not using pre-trained weights decreases the detection accuracy. Among them, U-Net was the most affected and PSPNet was the least affected. A 10% decrease in accuracy was observed for the Transformer-based Segformer model, which lacked pre-training weights. The primary reason for this is the complexity of the Transformer model and the number of parameters it contains. Training a transformer model from scratch requires significant GPU resources and time. In this study only 200 epochs were trained, making it difficult to train the model to a converged state.

Table 3 demonstrates that both the pre-training weights and the presence of shadows significantly affect the segmentation accuracy of the cracks. Since the SCSNet proposed in this study is the latest network, all parameters are randomly generated using normal distribution in the training phase.

5.2. Ablation study

To demonstrate the validity of the individual modules, ablation experiments were performed and the results are shown in Table.6. When the discrete cosine transform (DCT) is added to the baseline, the mIoU value increases from 75.24 to 77.92, which is a significant improvement for crack segmentation. DCT can transform crack images from the spatial domain to the frequency domain, where the frequency domain can

Table 7
Results of serial ensemble training compared to end-to-end training.

Training mode	Model	mIoU	mPA	Accuracy
Serial ensemble training	Model 1	78.61	86.89	97.83
	Model 2	77.92	86.19	97.69
	Model 1 + 2	79.29	87.33	97.89
End-to-end training		**79.52**	**87.67**	**97.93**

Table 4
Comparison of different models with shadows.

Model	mIoU	mPA	Accuracy	FPS
UNet [19]	77.52	85.43	97.69	67
Segformer [44]	76.64	81.97	97.76	70
PSPNet [45]	78.28	85.58	97.81	**133**
HRNet [46]	78.66	85.79	97.87	36
Deeplab v3+ [47]	78.24	85.71	97.80	69
SCSNet	**79.52**	**87.67**	**97.93**	54

effectively capture edge and texture information in the images. These features are crucial in semantic segmentation tasks. Additionally, DCT can maintain important high-frequency information while removing unnecessary low-frequency information in shadowed environments. Such operations can reduce the interference from noise like shadows, thereby improving the model's robustness and accuracy. DCT can also enhance the contrast and details of the image, enabling better distinguishing the edge information of the cracks. Combining the above factors, DCT can significantly improve the accuracy of crack segmentation.

The SRM module can further improve the accuracy of the model based on the DCT module. Because the SRM module is located between the network input and the backbone. Before the model processes the crack images, the SRM module fades shadows through a dual-branch structure, thereby reducing the impact of shadowed environments on crack segmentation. Compared to DCT, the improvement in mIoU and other metrics by the SRM module was smaller, primarily because the SRM module cannot fully control the effect of fading shadows. In combination with the PF loss function proposed in this study, the mIoU, as well as mPA metrics, were significantly increased. This is because the frequency of the pixel distribution of the crack images in the shadow-free dataset is taken as the ground truth to control the effect of shadow removal by the SRM module. The frequency distribution of crack images with shadows typically exhibits a double peak, whereas those without shadows display a single peak. By designing the frequency distributions of both types of images as a loss function, we can control the effectiveness of the SRM module, leading to significant improvements in various metrics.

The ablation experiment also indicated that all three tricks proposed in this study (DCT module, SRM module, and PF loss) are effective for crack segmentation.

In this study, PF Loss specifically optimizes the SRM module during training. Therefore, the SCSNet network could be considered a serial ensemble learning model that combines the “SRM + PF loss” with “Baseline + DCT”. We performed a comparison of the results of end-to-end training with serial ensemble training. First, the SRM + PF Loss (Model 1) and the DCT module (Model 2) were trained separately by us, and the rest of the parts were kept the same except for the difference of the modules. After obtaining the individual results, we placed Model 1 and Model 2 in parallel within the same framework. The outputs from Model 1 and Model 2 were fed into a simple classifier to verify consistency with the results of end-to-end training, as detailed in Table. 7.

The results demonstrate that end-to-end training surpasses serial ensemble training across all evaluation metrics, including mIoU, mPA, and Accuracy. This is due to the fact that end-to-end training can better coordinate the parameter tuning between the two modules and reduce the accumulation of errors from independent training.

Table 5
Comparison of different models without shadows.

Model	mIoU	mPA	Accuracy
UNet [19]	79.58	87.80	97.91
Segformer [44]	78.53	84.25	97.73
PSPNet [45]	79.50	87.15	97.93
HRNet [46]	80.37	87.95	97.93
Deeplab v3+ [47]	80.67	89.07	98.02
SCSNet	**81.89**	**90.12**	**98.15**

5.3. Comparison of different models

Table 4 shows a comparison of the results of our proposed SCSNet model with common segmentation models under shaded conditions. It is important to note that these compared models were trained based on pre-trained weights, while SCSNet models were trained from scratch. Our proposed SCSNet model achieved the best results in all three metrics. Under shadow conditions, the *mIoU* value of SCSNet reached 79.52, which is higher than the second-place HRNet with a score of 78.66. This means that SCSNet performs better in distinguishing the boundaries between background and cracks. This is because the wrong segmentation of crack pixels typically occurs at the edge of the cracks. The commonly used semantic segmentation models are not good at segmenting the edge of the cracks, although they can segment the approximate location where the cracks are located. Under the interference of shadow edges, segmentation accuracy can further decline. The fact that SCSNet can be optimal in *mIoU* is enough to prove that it can well handle shadow segmentation tasks under shadow interference.

The *mPA* value of SCSNet reaches 87.67, which is 2% higher than other models, which is a significant improvement in the field of crack segmentation. The higher *mPA* value means that SCSNet can better distinguish cracks from shadowed pixels. Even though shadowed pixels can have a significant impact on the segmentation results, SCSNet can reduce the interference of shadows on the segmentation of crack pixels with the help of SRM, DCT, and PF loss. Almost all the background pixels can be correctly predicted, and only those pixels located at the edges of the cracks are susceptible to being mispredicted. Since the number of these pixels is too small compared to the number of background pixels, the Accuracy values are all high. SCSNet achieved an Accuracy value of 97.93, which is the optimal value among all models.

To further validate the effectiveness of the SCSNet model, we tested the accuracy of SCSNet in the shadow-free dataset, as shown in Table.5. Table.5 indicates that the accuracy of all models improved in the absence of shadow interference. SCSNet still outperforms the compared models, demonstrating its performance in crack segmentation. The mIoU value for crack segmentation was improved to 81.89 by SCSNet, which is still superior to the well-performing HRNet and Deeplab v3+ models. HRNet mainly improves the accuracy of the model by fusing feature maps of different scales. However, SCSNet mainly extracts the important channels containing crack feature information through DCT. The results in Tables.4 and 5 indicate that channel information is often more critical in crack segmentation.

In addition, we conducted experiments to evaluate the inference speed of each model, with the results presented in Table. 4. Frames per second (FPS) served as the criterion for assessing model inference speed, representing the number of images the model can process in one second. The results in Table. 4 were obtained using RTX 4090. From the table we find that PSPNet exhibits the highest inference speed, while HRNet demonstrates the slowest, and SCSNet is only better than HRNet. This performance discrepancy is attributed to the inclusion of both the SRM and DCT modules in SCSNet, which increase the model's parameter and computational complexity, thereby reducing its inference speed.

The detection results of SCSNet are shown in Fig. 12. Fig. 12 includes

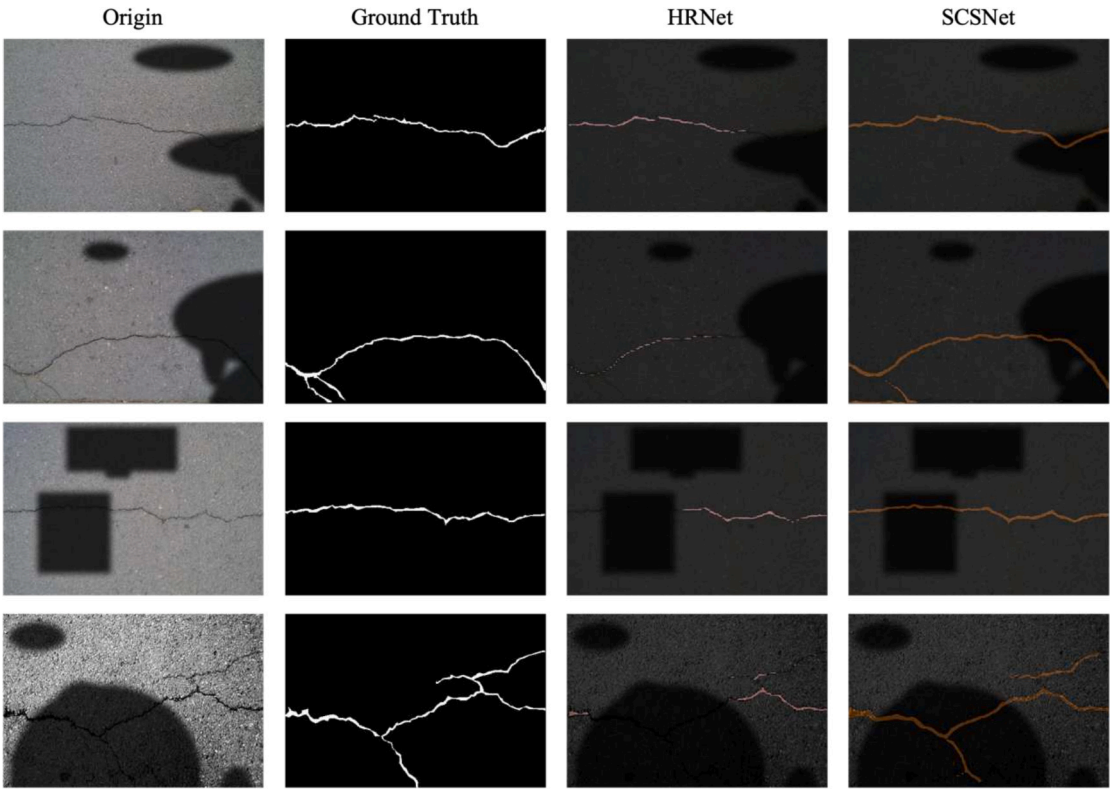


Fig. 12. Segmentation results of SCSNet.

Table 8
Results of different DCT frequency components.

Frequency components	mIoU	mPA	Accuracy
this paper	**77.92**	**86.19**	**97.69**
top32	77.32	85.71	97.45
top8	77.44	85.7	97.47
top4	77.14	85.87	97.4
top2	76.29	84.77	97.51
top1	76.1	84.5	97.49
bot32	75.45	83.66	97.43
bot16	75.54	83.57	97.45
bot8	75.71	83.75	97.47
bot4	75.5	83.57	97.44
bot2	75.64	83.71	97.46
bot1	75.6	83.72	97.45
low1	76.07	84.55	97.48
low2	76.27	84.66	97.51
low4	76.19	84.44	97.51
low8	76.34	84.69	97.52
low16	76.44	84.82	97.53
low32	76.19	84.7	97.5

the original image with shadows, the ground truth, and the detection results of HRNet and SCSNet. The results show that the shadow region has a large impact on the HRNet model, and it is difficult to recognize the cracked pixels within the shadow range. The detection results of SCSNet are close to the ground truth, which further proves the performance of SCSNet.

5.4. DCT frequency components

In the selection of DCT frequency components, 18 DCT components, as specified in [38], were tested. Only the names of the components are listed here, the specific parameters can be found in that paper. From Table. 8, we can see that the parameters we used are the best among all the results.

5.5. On-site experiment

To validate the robustness of the SCSNet model, we collected several pavement crack images in complex environments from Hong Kong. The detection results of SCSNet are shown in Fig. 13. HRNet, which has shown good segmentation performance, was used as a comparison benchmark in our study. There is serious noise in our captured images. From the segmentation results, we can observe that HRNet exhibits a large number of mispredictions, whereas our proposed SCSNet can accurately segment the cracks present in the images. Although the segmentation is not effective in some details, this experiment is sufficient to demonstrate that SCSNet can minimize the effect of surrounding noise on crack segmentation.

6. Conclusions

To address the impact of roadside buildings or trees on pavement crack segmentation, this paper presents a model called SCSNet for segmenting pavement cracks in complex environments such as shadows. The innovations of the model are the discrete cosine transform module (DCT), the shadow removal module (SRM), and the pixel frequency distribution loss function (PF loss). Ablation experiments show that the DCT module mainly transforms crack image information from the spatial domain to the frequency domain, retains important high-frequency information, and removes low-frequency interference. This enhances the accuracy of crack segmentation in shadowed environments. The combined use of the SRM module and the PF loss allows SCSNet to effectively mitigate shadows and further enhance the accuracy of crack segmentation in shadowed environments. Comparison with classical semantic segmentation models shows that our proposed SCSNet model consistently achieves higher crack segmentation accuracy than models based on convolutional neural networks or Transformer architectures, regardless of the presence or absence of shadow interference. In addition, the strong robustness of SCSNet has been proven by

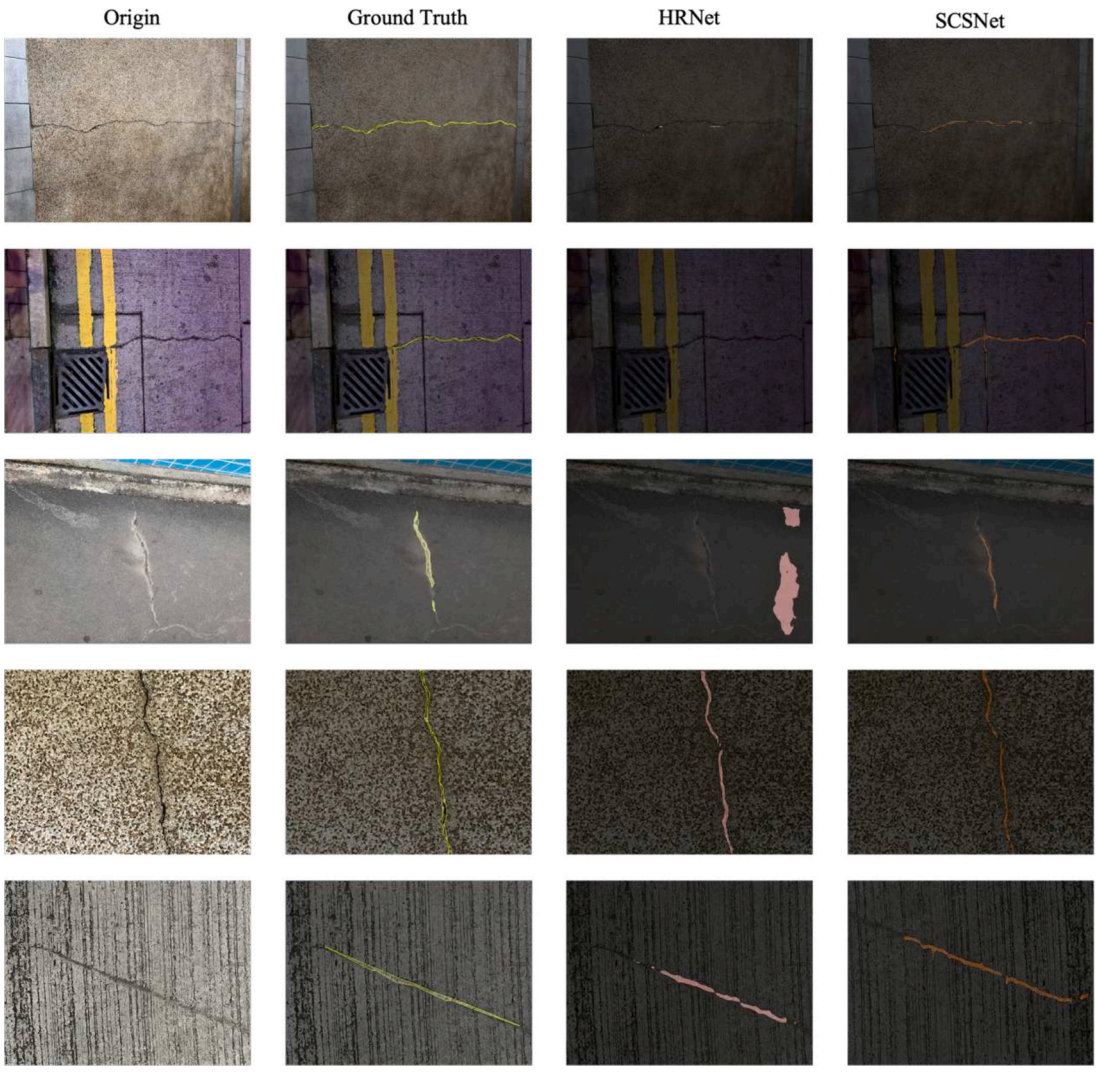


Fig. 13. On-site experiment.

on-site experiments. Therefore, the SCSNet proposed in this study can be applied in real engineering to quantify the geometric characteristic information of cracks.

Although the proposed SCSNet exhibits advanced segmentation performance, its inference speed is relatively slow, making real-time pavement crack segmentation challenging. Future research could focus on developing the lightweight crack segmentation network to achieve real-time segmentation of cracks in complex environments. With the development of large language models, these algorithms can be combined with large language models to enable the generation of text information such as disease damage characteristics and maintenance recommendations. The ultimate goal of this research should be to develop fully automated pavement damage vehicles.

CRediT authorship contribution statement

Yingchao Zhang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Data curation. **Cheng Liu:** Writing – review & editing, Writing – original draft, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The research is partly supported by the New Faculty Start-up Fund from City University of Hong Kong with grant number 9610612, partly supported by the Research Matching Grant Scheme with grant number 9229141, partly supported by the CityU Strategic Interdisciplinary Research Grant with grant number 7020076.

References

[1] X. Weng, Y. Huang, W. Wang, Segment-based pavement crack quantification, Autom. Constr. 105 (2019) 102819, https://doi.org/10.1016/j.autcon.2019.04.014.
[2] W. Wang, C. Su, Automatic concrete crack segmentation model based on transformer, Autom. Constr. 139 (2022) 104275, https://doi.org/10.1016/j.autcon.2022.104275.
[3] Z. He, W. Chen, J. Zhang, Y. Wang, Crack segmentation on steel structures using boundary guidance model, Autom. Constr. 162 (2024) 105354, https://doi.org/10.1016/j.autcon.2024.105354.

[4] K.W. Tse, R. Pi, Y. Sun, C.Y. Wen, Y. Feng, A novel real-time autonomous crack inspection system based on unmanned aerial vehicles, Sensors 23 (7) (2023) 3418, https://doi.org/10.3390/s23073418.
[5] F. Panella, A. Lipani, J. Boehm, Semantic segmentation of cracks: data challenges and architecture, Autom. Constr. 135 (2022) 104110, https://doi.org/10.1016/j.autcon.2021.104110.
[6] Y. Zhang, C. Liu, Real-time pavement damage detection with damage shape adaptation, IEEE Trans. Intell. Transp. Syst. (2024), https://doi.org/10.1109/TITS.2024.3416508.
[7] N. Chen, Z. Xu, Z. Liu, Y. Chen, Y. Miao, Q. Li, Y. Hou, L. Wang, Data augmentation and intelligent recognition in pavement texture using a deep learning, IEEE Trans. Intell. Transp. Syst. 23 (12) (2022) 25427–25436, https://doi.org/10.1109/TITS.2022.3140586.
[8] Y. Zhang, Z. Zuo, X. Xu, J. Wu, J. Zhu, H. Zhang, J. Wang, Y. Tian, Road damage detection using UAV images based on multi-level attention mechanism, Autom. Constr. 144 (2022) 104613, https://doi.org/10.1016/j.autcon.2022.104613.
[9] D. Arya, H. Maeda, S.K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, Y. Sekimoto, Deep learning-based road damage detection and classification for multiple countries, Autom. Constr. 132 (2021) 103935, https://doi.org/10.1016/j.autcon.2021.103935.
[10] D. Arya, H. Maeda, Y. Sekimoto, From global challenges to local solutions: a review of cross-country collaborations and winning strategies in road damage detection, Adv. Eng. Inform. 60 (2024) 102388, https://doi.org/10.1016/j.aei.2024.102388.
[11] F. Kluger, C. Reinders, K. Raetz, P. Schelske, B. Wandt, H. Ackermann, B. Rosenhahn, Region-based cycle-consistent data augmentation for object detection//2018 IEEE international conference on big data (big data), IEEE (2018) 5205–5211, https://doi.org/10.1109/BigData.2018.8622318.
[12] W. Ding, X. Zhao, B. Zhu, Y. Du, G. Zhu, T. Yu, L. Li, J. Wang, An ensemble of one-stage and two-stage detectors approach for road damage detection//2022 IEEE international conference on big data (big data), IEEE (2022) 6395–6400, https://doi.org/10.1109/BigData55660.2022.10021000.
[13] D. Jeong, J. Kim, Road damage detection using yolo with image tiling about multi-source images//2022 IEEE international conference on big data (big data), IEEE (2022) 6401–6406, https://doi.org/10.1109/BigData55660.2022.10020282.
[14] V. Pham, D. Nguyen, C. Donan, Road damage detection and classification with YOLOv7//2022 IEEE international conference on big data (big data), IEEE (2022) 6416–6423, https://doi.org/10.1109/BigData55660.2022.10020856.
[15] S. Wang, Y. Tang, X. Liao, J. He, H. Feng, H. Jiao, X. Su, Q. Yuan, An ensemble learning approach with multi-depth attention mechanism for road damage detection//2022 IEEE international conference on big data (big data), IEEE (2022) 6439–6444, https://doi.org/10.1109/BigData55660.2022.10021018.
[16] W. Choi, Y.J. Cha, SDDNet: real-time crack segmentation, IEEE Trans. Ind. Electron. 67 (9) (2019) 8016–8025, https://doi.org/10.1109/TIE.2019.2945265.
[17] Y. Zhang, C. Liu, Network for robust and high-accuracy pavement crack segmentation, Autom. Constr. 162 (2024) 105375, https://doi.org/10.1016/j.autcon.2024.105375.
[18] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks//Proceedings of the IEEE international conference on computer vision, 2017, pp. 764–773, https://doi.org/10.48550/arXiv.1703.06211.
[19] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation//Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, Springer International Publishing, 2015, pp. 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.
[20] J. Guan, X. Yang, L. Ding, X. Cheng, V.C.S. Lee, C. Jin, Automated pixel-level pavement distress detection based on stereo vision and deep learning, Autom. Constr. 129 (2021) 103788, https://doi.org/10.1016/j.autcon.2021.103788.
[21] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint (2018), https://doi.org/10.48550/arXiv.1804.02767.
[22] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1137–1149, https://doi.org/10.1109/TPAMI.2016.2577031.
[23] J. Liu, X. Yang, S. Lau, X. Wang, S. Luo, V.C.S. Lee, L. Ding, Automated pavement crack detection and segmentation based on two-step convolutional neural network, Comput. Aided Civ. Inf. Eng. 35 (11) (2020) 1291–1305, https://doi.org/10.1111/mice.12622.
[24] W. Wang, H. Li, K. Wang, C. He, M. Bai, Pavement crack detection on geodesic shadow removal with local oriented filter on LOF and improved level set, Constr. Build. Mater. 237 (2020) 117750, https://doi.org/10.1016/j.conbuildmat.2019.117750.
[25] J. Huyan, W. Li, S. Tighe, R. Deng, S. Yan, Illumination compensation model with k-means algorithm for detection of pavement surface cracks with shadow, J. Comput. Civ. Eng. 34 (1) (2020) 04019049, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000869.
[26] S. Liu, Y. Han, L. Xu, Recognition of road cracks based on multi-scale Retinex fused with wavelet transform, Array 15 (2022) 100193, https://doi.org/10.1016/j.array.2022.100193.
[27] W. Wang, L. Li, Y. Han, Crack detection in shadowed images on gray level deviations in a moving window and distance deviations between connected components, Constr. Build. Mater. 271 (2021) 121885, https://doi.org/10.1016/j.conbuildmat.2020.121885.
[28] P. Palevičius, M. Pal, M. Landauskas, U. Orinaite, I. Timofejeva, M. Ragulskis, Automatic detection of cracks on concrete surfaces in the presence of shadows, Sensors 22 (10) (2022) 3662, https://doi.org/10.3390/s22103662.
[29] S.H. Khan, M. Bennamoun, F. Sohel, R. Togneri, Automatic shadow detection and removal from a single image, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2015) 431–446, https://doi.org/10.1109/TPAMI.2015.2462355.
[30] L. Zhang, C. Long, X. Zhang, C. Xiao, Ris-gan: Explore residual and illumination with generative adversarial networks for shadow removal, in: Proceedings of the AAAI Conference on Artificial Intelligence 34, 2020, pp. 12829–12836, https://doi.org/10.1609/aaai.v34i07.6979, 07.
[31] Z. Liu, H. Yin, X. Wu, Z. Wu, Y. Mi, S. Wang, From shadow generation to shadow removal, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 4927–4936, https://doi.org/10.48550/arXiv.2103.12997.
[32] H. Le, D. Samaras, From shadow segmentation to shadow removal//Computer Vision–ECCV 2020, in: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, Springer International Publishing, 2020, pp. 264–281, https://doi.org/10.1007/978-3-030-58621-8_16.
[33] L. Fan, S. Li, Y. Li, B. Li, D. Cao, F. Wang, Pavement cracks coupled with shadows: a new shadow-crack dataset and a shadow-removal-oriented crack detection approach, IEEE/CAA J. Automatica Sinica 10 (7) (2023) 1593–1607, https://doi.org/10.1109/JAS.2023.123447.
[34] N.H.T. Nguyen, S. Perry, D. Bone, H.T. Le, T.T. Nguyen, Two-stage convolutional neural network for road crack detection and segmentation, Expert Syst. Appl. 186 (2021) 115718, https://doi.org/10.1016/j.eswa.2021.115718.
[35] Q. Zou, Y. Cao, Q. Li, Q. Mao, S. Wang, CrackTree: automatic crack detection from pavement images, Pattern Recogn. Lett. 33 (3) (2012) 227–238, https://doi.org/10.1016/j.patrec.2011.11.004.
[36] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988, https://doi.org/10.48550/arXiv.1708.02002.
[37] F. Milletari, N. Navab, S.A. Ahmadi, V-net: fully convolutional neural networks for volumetric medical image segmentation, in: 2016 fourth international conference on 3D vision (3DV), IEEE, 2016, pp. 565–571, https://doi.org/10.1109/3DV.2016.79.
[38] Z. Qin, P. Zhang, F. Wu, X. Li, Fcanet: Frequency channel attention networks, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 783–792, https://doi.org/10.48550/arXiv.2012.11879.
[39] Y. Zhang, Z. Ma, X. Song, J. Wu, S. Liu, X. Chen, X. Guo, Road surface defects detection based on imu sensor, IEEE Sensors J. 22 (2022) 2711–2721, https://doi.org/10.1109/JSEN.2021.3135388.
[40] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258, https://doi.org/10.48550/arXiv.1610.02357.
[41] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, H. Ling, Feature pyramid and hierarchical boosting network for pavement crack detection, IEEE Trans. Intell. Transp. Syst. 21 (4) (2019) 1525–1535, https://doi.org/10.1109/TITS.2019.2910595.
[42] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Trans. Intell. Transp. Syst. 17 (12) (2016) 3434–3445, https://doi.org/10.1109/TITS.2016.2552248.
[43] 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, https://doi.org/10.1016/j.neucom.2019.01.036.
[44] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, P. Luo, SegFormer: simple and efficient design for semantic segmentation with transformers, Adv. Neural Inf. Proces. Syst. 34 (2021) 12077–12090, https://doi.org/10.48550/arXiv.2105.15203.
[45] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881–2890, https://doi.org/10.48550/arXiv.1612.01105.
[46] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, B. Xiao, Deep high-resolution representation learning for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 43 (10) (2020) 3349–3364, https://doi.org/10.1109/TPAMI.2020.2983686.
[47] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801–818, https://doi.org/10.48550/arXiv.1802.02611.