Article

Not peer-reviewed version

ERNet: A Rapid Road Cracks Detection Method from Low-Altitude UAV Remote Sensing Image

Zexian Duan , Jiahang Liu * , Xinpeng Ling , Jinlong Zhang , Zhiheng Liu

Posted Date: 2 April 2024

doi: 10.20944/preprints202404.0217.v1

Keywords: UAV remote sensing; Road Cracks; Semantic segmentation; Crack quantification; Edge detection

Article

ERNet: A Rapid Road Cracks Detection Method from Low-Altitude UAV Remote Sensing Image

Zexian Duan, Jiahang Liu *, Xinpeng Ling, Jinlong Zhang and Zhiheng Liu

Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
* Correspondence: jhliu@nuaa.edu.cn

Abstract: Rapid and accurate detection of road cracks is of great significance for road health monitoring, but currently this work is mainly completed through manual site surveys. Low-altitude UAV remote sensing can obtain images with centimeter or even subcentimeter ground resolution, which provides a new efficient and economical approach for rapid crack detection. Nevertheless, crack detection networks face challenges such as edge blurring and misidentification due to the heterogeneity of road cracks and the complexity of the background. To address these issues, we propose a real-time edge reconstruction crack detection network (ERNet) which adopts multi-level information aggregation to reconstruct crack edges and improve the accuracy of segmentation between the target and background. To capture global dependencies across spatial and channel levels, we propose an efficient bilateral decomposed convolutional attention module (BDAM) that combines depth separable convolution and dilated convolution to capture global dependencies across spatial and channel levels. To enhance the accuracy of crack detection, we use a coordinate-based fusion module that integrates spatial, semantic, and edge reconstruction information. In addition, we propose an automatic measurement of crack information for extracting the crack trunk and its corresponding length and width. Experimental results demonstrate that our network achieves the best balance between accuracy and inference speed compared to six established models.

Keywords: UAV remote sensing; road cracks; semantic segmentation; crack quantification; edge detection

1. Introduction

Pavement cracking is a common type of damage that significantly reduces the service life of roads and poses a safety risk to road users [1–3]. Visual interpretation is the main approach of crack detection but it is inefficient and prone to subjective errors. In the past decade, various automatic or semi-automatic methods have been proposed, including the use of sensors such as line scan cameras, RGB-D sensors, and laser scanners[4–6]. However, these sensor-equipped vehicles are costly and easy to cause traffic disruption and road type restrictions.

Nowadays, unmanned aerial vehicles (UAVs) have emerged as efficient and versatile tools for structural inspection [7,8]. UAV-based road crack detection offers significant advantages, including efficient, cost-effective, safe, and flexible image data acquisition [9]. In recent decades, digital image processing has been utilized for crack segmentation [10]. These approaches often require manual feature extraction, which can overlook the interdependence between cracks and lead to unsatisfactory results in practice [11]. UAV remote sensing has successfully solved the data source for crack detection, how to quickly and accurately detect and measure cracks has become the main problem at present.

Machine learning-based detection methods have been rapidly developed in recent years and are becoming the mainstream approach for crack detection [12–14]. These algorithms require different preprocessing of the image to be detected, which can be time-consuming for large images. With the rapid development of deep learning, new ideas are introduced into various computer vision tasks

doi:10.20944/preprints202404.0217.v1

[15–17]. Semantic segmentation is often preferred for crack detection as it provides more accurate and effective road health information, such as crack distribution, width, length, and shape. Liu et al. combined FCN and Deep supervision network (DSN) to propose DeepCrack [18], a multi-scene crack detection algorithm based on the idea of deep supervision. Ren et al. [19] proposed an improved CrackSegNet for pixel-level crack segmentation of tunnel surfaces, which improves accuracy and generalization by spatial pyramid pooling and skip connection modules. Liu et al. [20] proposed the use of U-Net for automatic crack detection, but it may generate redundant recognition due to background interference. Wang et al. [21] combining CNN with transformer, an efficient feedforward network is constructed for global feature extraction. These models often suffer from computational delay and inefficiency due to their large number of parameters and computational redundancy. It becomes particularly acute when dealing with large amounts of data, resulting in a high overhead of computational power.

To reduce computational costs, researchers have proposed some lightweight networks. Many initial lightweight models prioritize speed over spatial detail. These methods may lead to loss of spatial detail and precision, particularly in the boundary regions. Yu et al. introduced the Bilateral Segmentation Network (BiSeNet) in their groundbreaking research [22], processing semantic and detailed information separately. Lightweight models often lack the ability to effectively extract edge information due to the characteristics of narrow cracks, irregular edges, and the potential for confusion with the road background. This can significantly impact detection accuracy. Short-Term Dense Cascade (STDC) module was proposed as a solution to this issue [23]. STDC Segmentation network (STDC-Seg) uses the STDC module to extract multi-scale information, which solves the existing backbone network problems of BiSeNet. Overall, STDC-Seg is a suitable segmentation structure for road crack detection.

Quantitative extraction of physical information from cracks is a downstream task in crack detection. To acquire these information, researchers combine crack detection algorithm with crack quantization algorithm to provide a safe and effective solution for road crack detection. Liao et al. [24] used a spatially constrained strategy on lightweight CNNS to ensure fracture continuity. Yang et al. [25] attempted to quantitative analysis of detected cracks at the pixel level. However, the quantitative results did not meet expectations. Li et al. [26] proposed a pixel-level segmentation of crack method, which fuses the SegNet and the dense condition random field and calculated the width and area of one-way and grid cracks. In general, the density, width and length of cracks can provide important reference for road health evaluation, and accurate crack detection results provide an important foundation for the extraction of these elements.

Due to the narrow shape of most cracks, the crack edge information is very important for the accurate location and segmentation of cracks. In addition, in most crack detection tasks, irregular cracks, rough roads, light, shadows and other factors will affect the location and segmentation of cracks. Accurate detection of crack edges is crucial for semantic segmentation networks to address these challenges and extract quantitative information such as crack length and width. Tao et al. [27] designed a Boundary Awareness Module in their proposed approach, but their label-based learning is prone to misjudge background noise. Pang et al. [28] introduced a two-branch lightweight network into crack detection, but the lightweight design limited the network's ability to extract global information, so the network was easy to miss small cracks. Holistic nested edge detection (HED)[29] and side-output residual network (SRN)[30] are two edge detection networks that build on the idea of deep supervision. Tsai et al. [31] fused the edge detection results of different sizes extracted by the Sobel edge detector on the semantic branch. However, it is often difficult for existing methods to address the problems of weak perception of crack edge details and uneven crack distribution, which makes quantitative information extraction a challenge.

To overcome these limitations, we propose a rapid road crack detection method for UAV remote sensing images. Specifically, we propose a real-time edge reconstruction crack detection network (ERNet), which integrates edge aggregation and enhancement into semantic segmentation. Inspirited by infrared small target detection [32], we develop an edge input module utilizing a soft gating mechanism for edge reconstruction. The proposed method achieves the best trade-off between inference speed and accuracy in the models participating in the comparison experiment. The mIoU score on crack500 dataset is 82.48%, and the F1 score is 79.67%. The mIoU score on DeepCrack dataset is 86.6%, and the F1 score is 84.86%. The mIoU score of the generalization experiment on the self-made UAV dataset is 80.25% and the F1 score is 76.21%. Comparative analysis demonstrates the feasibility and superiority of this method. Our main contributions are as follows:

(1) We propose a novel ERNet to achieve high-precision and fast crack edge segmentation through edge reconstruction and realizes the quantification of crack length and width information on this basis. It provides a whole solution from detection to extraction.

(2) We design a key model called BDAM that effectively improves attention at both spatial and channel levels, selectively represents features in the channel and spatial domains, and captures global contextual information.

The rest of the article is organized as follows. Section Ⅱ describes the architecture of ERNet and its components in detail. Section Ⅲ verifies the effectiveness of our method in improving the comprehensive performance of crack detection with experimental results. Section IV is the conclusion.

2. Methodology

The difficulty of the crack detection task comes from the fuzzy boundary transition of the crack, the chaotic background and the foreground interference, etc. The accurate location of the crack edge is the key to deal with these challenges. By reconstructing the edge details, our proposed network improves the location accuracy of the crack edge, improves the coherence of the detection results, and provides accurate detection results for the quantitative extraction of cracks. The overall structure of the network is shown in Figure 1.

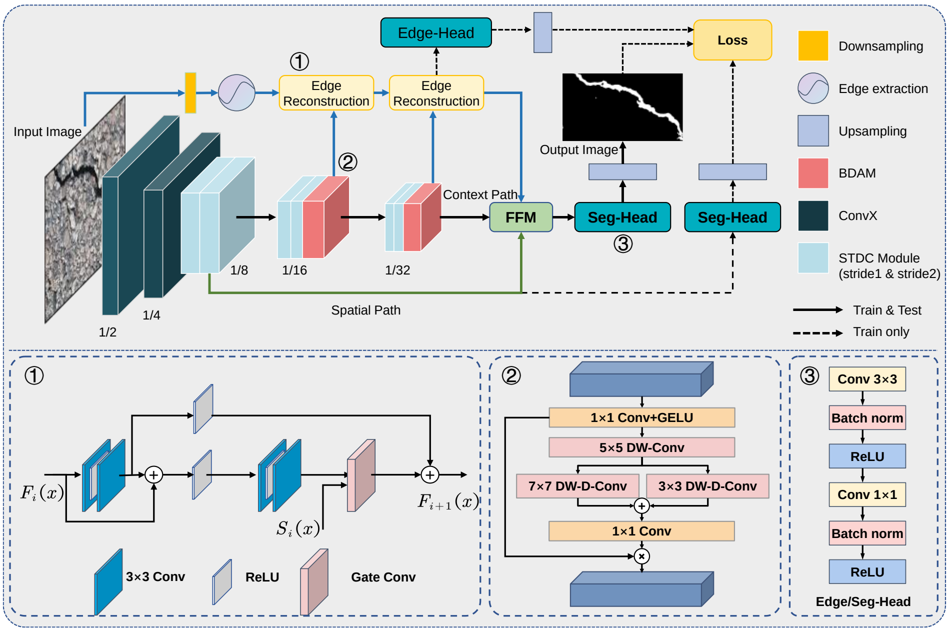


Figure 1. The overview structure of our proposed network for crack detection.

The network uses a three-branch structure to encode features at different levels, including edge path for extracting and preserving high-frequency features, spatial path for preserving detailed information, and semantic path for extracting deep semantic features. In semantic path, we use STDC module for local feature extraction and BDAM for global feature extraction; In the edge path, we input the high-frequency information and semantic information into the edge reconstruction module to encode the edge features, and use the key damage boundary information. In the spatial path, we implement shallow and wide convolution layers to achieve fast downsampling and preserve spatial details.

In this section, we first introduce the backbone network we used, then introduce the bilateral decomposed convolutional feature attention module, and finally describe in detail the side input branch for edge detection and feature fusion module of the model.

2.1. Backbone

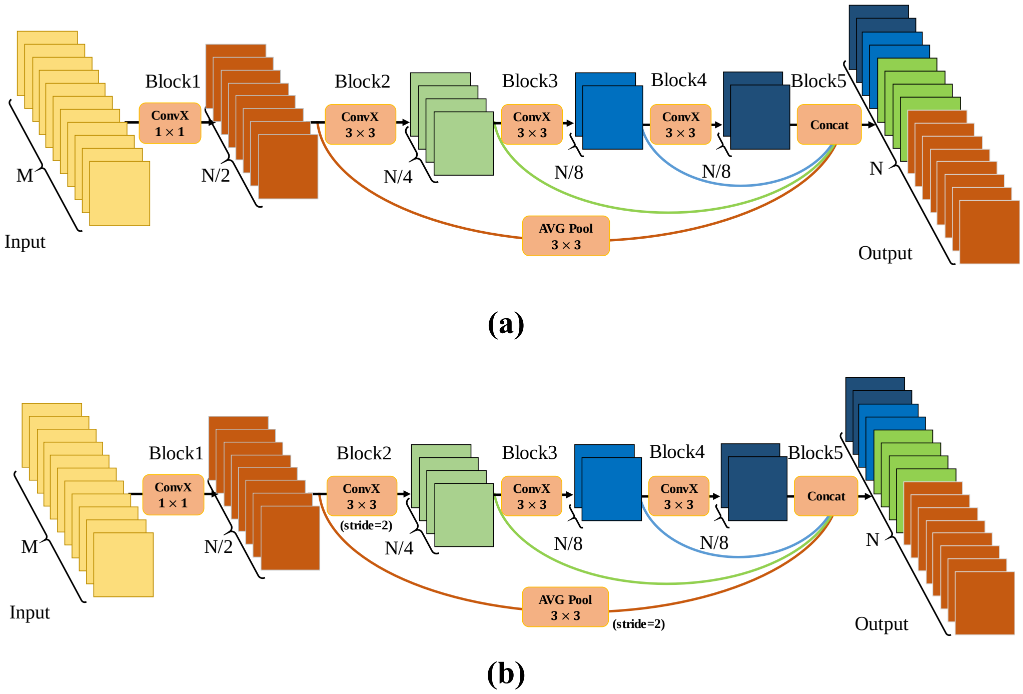
Our proposed model uses the STDC module as a feature extractor and retains the spatial branch. We use the STDC-Seg network backbone as the ERNet backbone. The operation of ConvX includes a convolution layer, a batch normalization layer, and a ReLU activation layer. We used feature maps of 1/8 size instead of 1/4 size as input to spatial branching because it reduces the amount of computation and preserves enough spatial detail. The STDC module is the core component of the backbone network, shown in Figure 2.

Two types of STDC modules, Stride=1 and Stride=2, are used for different tasks. STDC module with a stride of 2 is used to downsampling feature maps, and then the STDC module with a stride of 1 is used for further feature extraction. The number of filters in the i-th convolution layer of the block is N/2i, where N is the output of the STDC module. The number of filters in the last two convolution layers is set to be the same. The STDC module is divided into several blocks. The feature mapping of the i-th block is calculated as equation (1):

$$\boldsymbol{X}_{out} = F(x_1, x_2, ...x_n) \tag{1}$$

Where $\boldsymbol{X}_{out}$ represents the module output, F represents the concat fusion operation, and $x_1, x_2, ...x_n$ are the feature maps of all blocks.

The output of the STDC module integrates the multi-scale information of all blocks. As the number of blocks increases, the receptive field also increases, and the scalable receptive field and information are retained through fusion operations.



(a)

(b)

Figure 2. Illustration the STDC module. (a) STDC module with a stride of 1. (b) STDC module with a stride of 2.

2.2. BDAM

The significance of global context in segmentation tasks has been confirmed by numerous previous studies[33–37]. Convolution-based methods accomplish this by enlarging the receptive field through increased kernel size or stride, whereas transformer-based methods [38,39] usually consider spatial dimension adaptability and ignore channel dimension adaptability, which is important for visual tasks.

To capture distant relationships, we introduce decomposed convolution blocks and design the efficient bilateral decomposed convolutional attention module (BDAM). As illustrated in Figure 3, The large kernel convolution is divided into three parts by BDAM: depth convolution for capturing multi-scale context, multi-branch depth convolution, and 1×1 convolution for establishing relationships between distinct channels. During the decomposition process, we break down the K×K convolution into depth convolution, multi-scale depth expansion convolution, and 1×1 convolution. The BDAM is described as follows:

$$\boldsymbol{Att} = Conv_{1\times1}(DW - DConv_{branch1}(\boldsymbol{X_{in}}) + DW - DConv_{branch2}(\boldsymbol{X_{in}})) \qquad (2)$$

$$\boldsymbol{Out} = \boldsymbol{Att} \otimes \boldsymbol{X_{in}} \qquad (3)$$

Where the $\boldsymbol{X_{in}}$ denotes input features, corresponding to the multiplication operation of element matrices. $\boldsymbol{Att}$ and $\boldsymbol{Out}$ represent attention maps and outputs, respectively.

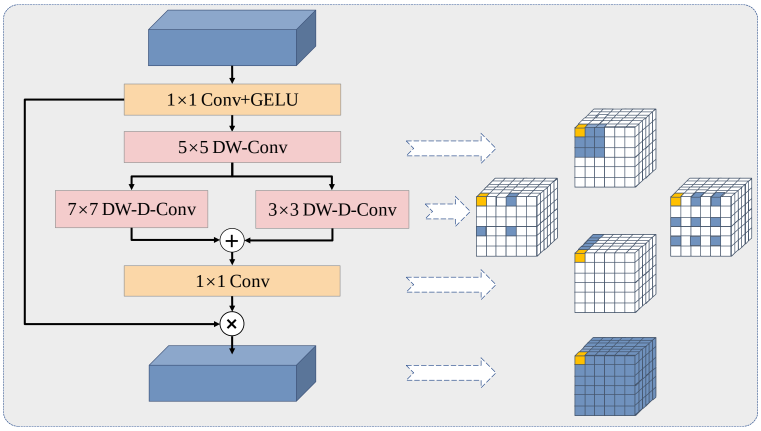


Figure 3. Illustration of the bilateral decomposed convolutional attention module. The BDAM is used to refine the corresponding combined features of the decoding stage. Among them, depth-wise separable convolution and dilated convolution are used to capture the global content, and attention vector is used for guidance.

In this network, depth dilated convolutions in each branch have kernel sizes of 3 and 7, respectively. This configuration aligns with standard convolutions having kernel sizes of 7 and 19, and enables capturing remote relationships across different scales using depth dilated convolutions with varying kernel sizes in a dual-branch structure. The output of the 1×1 convolution serves as the attention weight for input features, providing both spatial and channel adaptability.

2.3. Edge Reconstruction Module

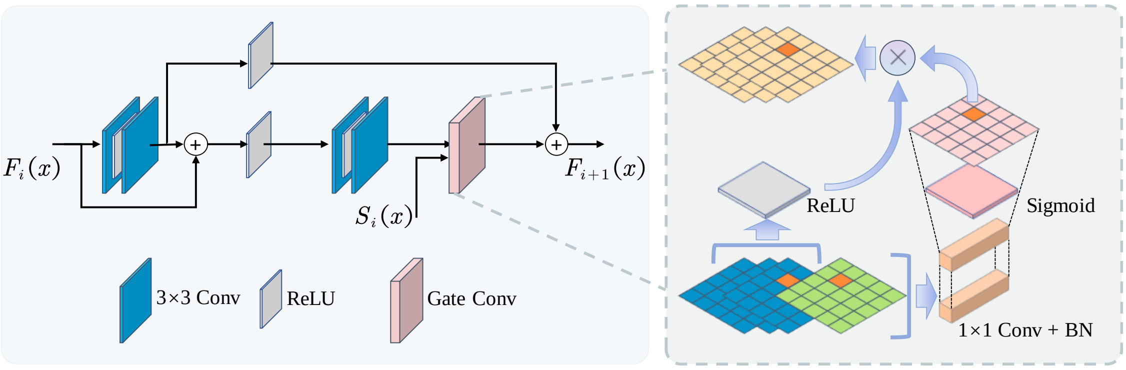
In detection tasks, the small and narrow cracks are often lost in multiple downsampling processes. The feature information of cracks is closely related to their edge information, which includes fine details of the target. To address this issue, the Laplacian operator is adopted as an edge extraction operator to filter the image and further refine the coarse edge information extracted from it. However, the Laplacian operator's use at each stage increases computational complexity, and setting the threshold for judging the boundary too high or too low can result in ineffective edge detection. In addition, it is difficult for the Laplacian operator to extract edge information from convolutional encoded features. After several experiments, we use 1/8 size images as input and choose a threshold of 40 for the Laplacian operator.

Inspired by small target detection, we use the edge reconstruction module (ERM) based on the second-order Taylor finite difference equation. to process rough edge features [40]. The structure of the ERM performs a nonlinear transformation of the shallow edge feature map through two residual blocks to obtain features with less noise and clutter. Then, the soft gate mechanism is employed to perform directed learning on the rough edge results obtained by the Laplacian operator, which better suppresses background noise and focuses on the edge information of the target using the semantic features extracted by the backbone, which is shown in Figure 4. Where $\boldsymbol{F_i}(\boldsymbol{x})$ denotes rough edge features, $\boldsymbol{F_{i+1}}(\boldsymbol{x})$ denotes Refined edge features, and $\boldsymbol{S_i}(\boldsymbol{x})$ denotes high-level semantic features.

The gate convolution learns soft mask automatically from data. Guided by the soft mask, the edge reconstruction module extracts the accurate crack boundary information from the chaotic rough edge features. It is formulated as equation (4):

$$\boldsymbol{Gate_{out}} = \phi(\boldsymbol{Feature_i}(\boldsymbol{x})) \odot \sigma(W_f(\boldsymbol{S_i}(\boldsymbol{x}), \boldsymbol{Feature_i}(\boldsymbol{x}))) \qquad (4)$$

Where σ is sigmoid thus the output gating values are between zeros and ones. ϕ is ReLU. W_f is a sequence of convolutional filters.



(a) Edge reconstruction module

(b) Gate convolution

Figure 4. Illustration of the edge reconstruction module.

In road surface crack detection, the number of crack pixels is significantly lower than that of non-crack pixels, resulting in a class imbalance problem. Weighted cross-entropy, as mentioned in Ref.[41], often leads to rough results. To address this issue, we jointly optimize edge learning using binary cross-entropy and Edge loss[42]. Edge loss is a general Dice-based edge-aware loss module that includes a dice edge loss function for overall contour fitting. The required edge prediction results are defined as follows:

$$\hat{e}_{ij} = squash(g_{ij}) = \frac{|g_{ij}|}{|g_{ij}| + \alpha} \qquad (5)$$

$$\Theta = argmaxDice(\boldsymbol{p_d}, \boldsymbol{g_d}) = argmax\frac{2\sum_{i=j}^{H}\sum_{j}^{W} e_j \hat{e}_{ij} \hat{e}_{ij}}{\sum_{i}^{H}\sum_{j}^{W} e_{ij}^{\;2} + \sum_{i}^{H}\sum_{j}^{W} e_{ij}^{\;2}} \qquad (6)$$

Where $\hat{e}_{ij}$ and g_{ij} respectively represent the edge prediction results and gradient information vectors at (*i*, *j*), and e_{ij} is the edge true value directly obtained from the detail ground-truth, and α is a hyperparameter that controls the model's sensitivity to object contours. In our experiments, we found that setting α to 1 achieves an optimum balance between intra-class unification and inter-class discrimination. The boundary refinement is represented by the dice coefficient maximization problem, as defined in the above equation (6). Where, $\boldsymbol{g_d} \epsilon R^{H\times W}$ is the true segmentation map and $\boldsymbol{p_d} \epsilon R^{H\times W}$ is predicted segmentation map, and Θ represents the parameter of the segmentation network. To implement SGD in the training process, the final edge loss is constructed as equation (7):

$$L_{edge}(\boldsymbol{p_d}, \boldsymbol{g_d}, \Theta) = 1 - Dice(\boldsymbol{p_d}, \boldsymbol{g_d}, \Theta) \quad (7)$$

2.4. Feature Fusion Module

The proposed network's feature fusion module (FFM) extracts multiple feature responses and fuses information from different level feature maps to achieve multi-element and multi-scale information encoding. As shown in Figure 5, the edge features are first concatenated with spatial and semantic features, and then the feature map size is divided into *C*×*H*×1 and *C*×1×*W* along the *X* and *Y* coordinates respectively using average pooling. The resulting feature maps are then divided into two separate tensors along the spatial dimension, and an attention vector is generated by sigmoid to guide the feature response of the spatial branch. This encoding of multi-element and multi-scale information integrates low-level feature maps with spatial information, edge reconstruction feature maps with edge information, and high-level feature maps with large receptive fields.

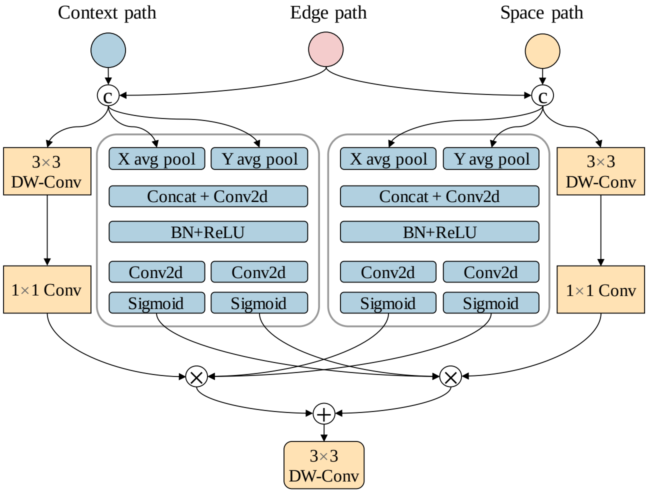


Figure 5. Illustration of the feature fusion module based on coordinate.

2.5. Crack information quantification

Crack length plays a crucial role in road safety prediction, as longer cracks indicate more severe road damage. The segmentation network generates a crack prediction map by predicting cracks at the pixel level. We extract correct crack skeletons by eliminating a large number of erroneous branches identified by Zhang & Suen et al. [43] algorithm (shown in Figure6 (b)) based on connected domain analysis. The result of crack skeleton extraction is shown in Figure 6 (c). The crack trunk can be extracted effectively by debranching algorithm based on connected domain. Finally, the number of adjacent pixels in the crack skeleton and the distance between adjacent cracks are calculated pixel by pixel, and the maximum length value represents the crack length.

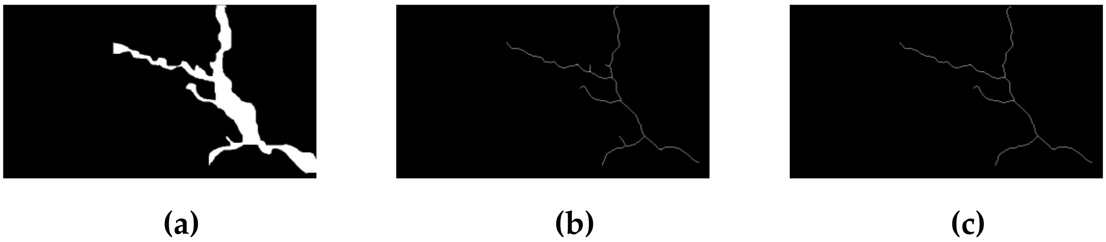


Figure 6. Diagram of crack skeleton backbone extraction. (a) Crack predicted map. (b) Zhang-Suen thinning algorithm. (c) Crack backbone extraction.

The width of the crack is equally important for road damage detection. Based on distance transform method (DTM), the distance between the crack skeleton and the crack edge was calculated, so as to obtain the maximum width. As shown in Figure 7 (a), the wider the crack area, the greater the gray value. Figure 7 (b) shows the results of the crack skeleton weighted with DTM values, so as to obtain the maximum width.

Figure 7. Result of the crack based on distance transformation method. (a) DTM values of crack prediction map. (b) Result of weighting the crack skeleton with DTM values.

3. Experiment and Results

This section details the dataset used for the experiments, training details of the proposed algorithm, evaluation criteria and experimental results.

3.1. Dataset

Because it is difficult to obtain the road crack data based on UAV, considering that the camera resolution of UAV is high enough and the angle of view of UAV is similar to that of mobile phone, it can be considered that the images collected by both have similar definition and imaging angle. We use the public road crack data sets Crack500 and DeepCrack collected by mobile phones, as the training set. In addition, we used the DJI UAV to take some road crack images, and made a small data set for the test of network generalization ability. The specific data set is described as follows.

The UAV dataset is collected for generalization ability test. The images captured using the DJI M300RTK drone equipped with the ZENMUSE H20 camera. The pixel resolution of the images is 5184×3888. Since the size of a single image is too large, we use LabelMe [44] for semantic annotation, and then crop the image to 512×512 size. By using data enhancement operations such as image flipping, we make a generalization data set containing 4692 images of UAV aerial road crack. The data was collected within the campus of Nanjing University of Aeronautics and Astronautics in Nanjing, Jiangsu Province, China. The annotated dataset includes both cement and asphalt road surfaces, with various types of cracks such as net-shaped cracks, longitudinal cracks, and transverse cracks.

The CRACK500 dataset [45] consists of 500 road crack images. In this experiment, each original image was divided into 16 non-overlapping images, each with a scale of 640×352. Images containing more than 1000 pixels of crack area are kept and further divided. The training set comprises 1896 images, the validation set comprises 348 images, and the test set comprises 1124 images.

The DeepCrack dataset [18] consists of 537 road crack images with a size of 544×384, each with a pixel-level binary label image. In our experiments, the dataset was divided into 300 images for the training dataset and 237 images for the validation dataset.

3.2. Implementation Details

All models in the experiments were implemented with the PyTorch framework on a single NVIDIA GTX 3090 GPU. We used SGD[46] to train our ERNet with batch size 8, and training epoch is set to 100, we apply the "poly" learning rate strategy in which the initial rate is multiplied by equation (8):

$$lr = initial\_lr \times \left(1 - \frac{iter}{max\_iter}\right)^{\mathrm{power}} \quad (8)$$

where the *iter* is the number of iterations, *max_iter* is the maximum number of iterations, and power controls the shape of the curve. The initial learning rate was set to 0.01, and the power was set to 0.9.

3.3. Comparative Experiment

We compared our ERNet with three lightweight semantic segmentation networks (BiSeNet[22], STDC2-seg[23], PIDNet[47]) and three crack detection networks (DeepCrackNet[18], CT-CrackSeg[27], LinkCrack[24]) based on the same implementation details and platform.

The accuracy evaluation standard used in this experiment is Intersection over Union (IoU), Precision (Pr), Recall (Re), F1 score (F1) and accuracy (Acc). We also calculated the average frames per second (FPS) of the network reasoning in the validation set while calculating the IOU. The measurements are shown in equations (9)-(13):

$$IOU = \frac{N_{TP}}{N_{FP} + N_{TP} + N_{FN}} \tag{9}$$

$$Acc = \frac{N_{TP} + N_{TN}}{N_{TP} + N_{TN} + N_{FP} + N_{FN}} \tag{10}$$

$$Pr = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{11}$$

$$Re = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{12}$$

$$F1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re} \tag{13}$$

where, N_{TP} is the number of positive samples classified as positive, N_{TN} is the number of negative samples classified as negative, N_{FP} is the number of negative samples classified as positive, and N_{FN} is the number of positive samples classified as negative.

Precision and recall evaluate the detection ability of the method from different perspectives, respectively. The F1 score combines the above two metrics. The IoU can give a better response to the local details of the detection results, and mean IoU(mIoU) is the average of the IoU of road and crack. Acc represents the proportion of correctly classified data (N_{TP}+N_{TN}) relative to the total data These indicators can be used to evaluate the detection performance of the network more objectively. The values of these indicators range between 0 and 1. Higher values, closer to 1, indicate better segmentation ability for crack areas. Validation data is used to select the optimal training iteration.

Table 1 presents the comprehensive indicators for the Crack500 validation dataset, with the best performing values highlighted in bold. Our model has achieved the highest results on IoU, Re, Acc, mIoU and F1 score. And the speed is also the fastest of the four crack detection networks, although our model has a 0.5 FPS lower frame rate than STDC2-seg, it achieves a 2.86% higher IoU for cracks compared to STDC2-seg and a 3.98% higher F1, making this trade-off acceptable. Despite having a lower precision compared to PIDNet, our model's higher F1 score demonstrates its superior ability to distinguish between background and cracks.

Table 1. Comparison of the experimental results of different semantic segmentation networks on the CRACK500 dataset.

	IoU(%)	**mIoU(%)**	**Pr (%)**	**Re (%)**	**F1(%)**	**Acc(%)**	**FPS**
BiSeNet	62.83	79.99	73.31	78.68	75.90	97.87	11.86
STDC2-seg	63.35	80.39	74.71	76.69	75.69	98.37	**22.1**
PIDNet	64.28	80.88	**79.87**	76.71	78.26	97.59	16.35
DeepCrackNet	55.67	76.69	66.75	77.02	71.52	96.54	3.35
CT-CrackSeg	62.54	80.04	60.50	78.00	73.30	97.02	3.42
LinkCrack	57.45	77.92	72.97	72.98	72.98	96.95	11.54
ERNet(ours)	**66.21**	**82.48**	79.21	**80.14**	**79.67**	**98.51**	21.6

Figure 8 presents the segmentation results for several image examples. The first row demonstrates our model’s recognition result has fewer breakpoints and is closer to the original image than other networks, which is not uncommon in the experiments of two datasets. The inference results for the second row of background-mottled crack images reveal that our model exhibits fewer missed detections, clearer edge details, and smoother boundaries compared to other networks. In the third and fourth rows of the image, all other networks have false checks in the shaded area. The fifth row demonstrates our model 's ability to recognize detailed information within the cracks, which is not achieved by other networks.

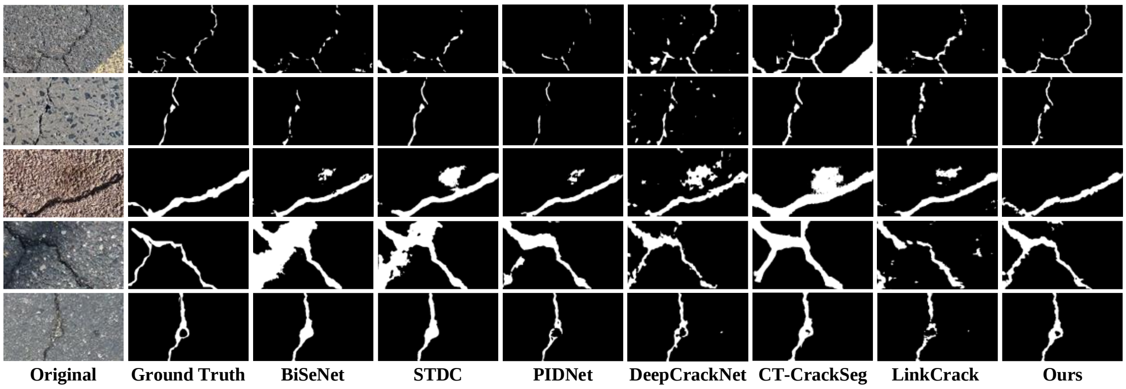


Figure 8. The visualization of different semantic segmentation detection results of compared methods on CRACK500.

Upon testing our model on the DeepCrack dataset, it achieved superior IoU, recall, and F1 scores. Although ERNet is 0.73% lower than PIDNet on precision, it is 3.28% higher on recall and thus 1.46% higher than PIDNet on F1. On the other hand, although our model is 1.15% lower than LinkCrack in recall, it is 7.05% ahead in precision and thus 2.87% higher than LinkCrack in F1. In terms of speed, our model is only lower than STDC2-seg. Although our model is slightly lower STDC2-seg on Acc by 0.16%, it leads 7.31% in IoU and 5.07% in F1, demonstrating its robust stability across various datasets. The detailed comparison results are presented in Table 2. Figure 9 displays the segmentation results for several image examples with increased interference.

Table 2. Comparison of the experimental results of different semantic segmentation networks on the DeepCrack dataset.

	IoU(%)	**mIoU(%)**	**Pr (%)**	**Re (%)**	**F1(%)**	**Acc(%)**	**FPS**
BiSeNet	69.10	83.76	81.29	79.19	80.23	98.97	13.1
STDC2-seg	66.40	82.33	84.52	75.56	79.79	**99.40**	**30.2**
PIDNet	71.54	85.06	**88.52**	78.85	83.40	98.64	25.47
DeepCrackNet	67.90	83.53	81.21	80.56	80.88	98.35	3.40
CT-CrackSeg	64.69	81.79	76.55	80.68	78.56	98.09	3.49
LinkCrack	69.48	84.29	80.74	**83.28**	81.99	98.42	14.07
ERNet(ours)	**73.71**	**86.60**	87.79	82.13	**84.86**	99.24	26.2

For the blurry image in the first and second rows, the small cracks extracted by our model are more coherent and retain more details. The images in the third and fourth rows demonstrate complex interference conditions due to shadows, light, and widespread net-shaped cracks, leading to decreased segmentation results for all network models, with BiSeNet, STDC2-seg, and CT-CrackSeg all showing extensive false detection, and our model has a smaller false detection range compared to other networks. The identification results of the fifth row show that the detection results of our model can preserve the edge details of the crack well, while reducing the number of breakpoints of the zigzag crack.

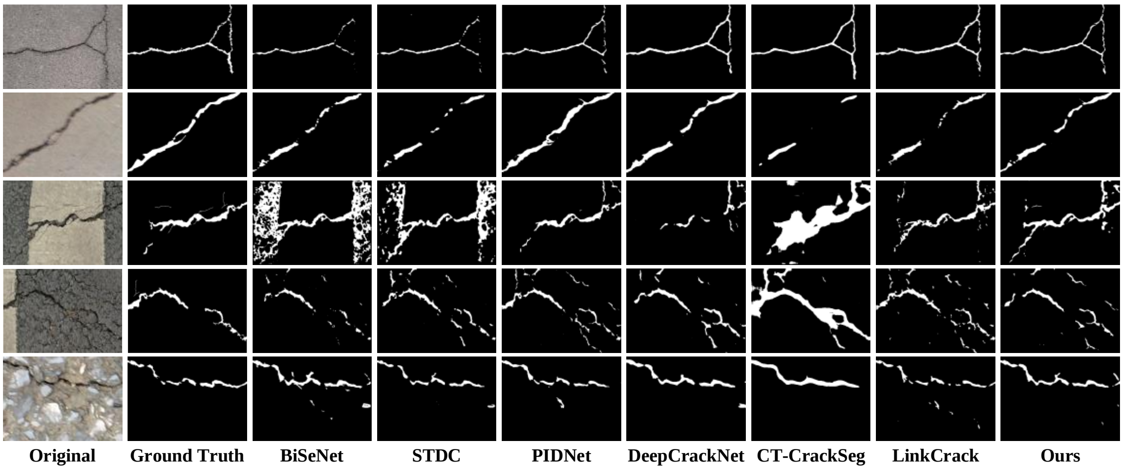


Figure 9. The visualization of different semantic segmentation detection results of compared methods on DeepCrack.

3.4. Generalization Ability Experiment

In this section, we use the best weight of each network obtained on the Crack500 training set to predict on the UAV crack data set, so as to detect the generalization ability of each network. The generalization experiment can truly reflect the ability of each network in the actual detection task. The experimental results are presented in Table 3. Our inference results outperform other models in IoU, mIoU, Precision, F1, and Acc metrics. Meanwhile, other all network performance indicators of a substantial decline. The mIoU of our inference results is only 2.23% lower than that of the Crack500 dataset, the results demonstrate that our proposed model possesses good generalization ability.

Table 3. Comparison of the experimental results of different semantic segmentation networks on UAV remote sensing dataset.

	IoU(%)	mIoU(%)	Pr (%)	Re (%)	F1(%)	Acc(%)
BiSeNet	41.87	69.73	58.71	59.36	59.03	97.62
STDC2-seg	41.71	69.66	59.40	58.34	58.87	97.65
PIDNet	35.15	66.14	51.12	52.94	52.01	97.18
DeepCrackNet	21.25	57.55	23.95	65.32	35.05	93.02
CT-CrackSeg	48.13	73.45	62.09	68.16	64.98	97.88
LinkCrack	26.91	63.21	65.26	31.42	42.41	97.54
ERNet(ours)	**61.56**	**80.25**	**69.91**	**83.75**	**76.21**	**98.36**

Figure 10 illustrates the segmentation results of different networks on UAV remote sensing images. The detection results of DeepCrackNet and LinkCrack are very scattered and lack of clear boundaries. CT-CrackSeg performs best except ERNet, but it still mistakenly detects the pavement markings as cracks, and the missing detection of small cracks is also quite significant. Specifically, the second row compares the segmentation of shallow cracks, our model can identify the main body of shallow cracks and correctly handle pavement markings. The third row contains both longitudinal cracks and transverse cracks. Other networks miss transverse cracks, while our model successfully segments both types of cracks relatively completely. The fourth and fifth rows depict the same area from different angles, demonstrating that our model not only approximates the ground truth but also exhibits high consistency in the same area in the two images. In conclusion, the experiments demonstrate that our proposed model possesses good generalization ability and delivers excellent performance in detecting road images from remote sensing. The fourth and fifth rows depict the same area from different angles, the result demonstrates that our model not only approximates the ground truth but also exhibits high consistency in the same area in the two images. DeepCrackNet has many false detections in the background area far from the crack, which indicates that its ability to extract global semantic information still has room for improvement. In conclusion, the experiments demonstrate that our proposed model possesses good generalization ability and delivers excellent performance in detecting road images from remote sensing.

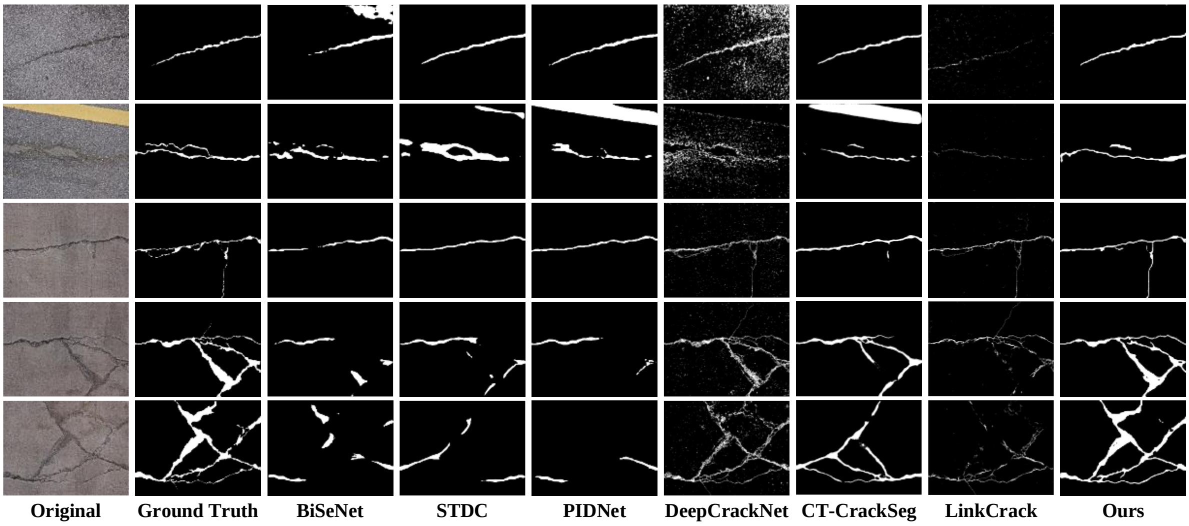


Figure 10. The visualization of different semantic segmentation detection results of compared methods on UAV remote sensing dataset.

3.5. Ablation Experiment Results

In this section, we performed six experiments on the Crack500 dataset using the STDC backbone, incorporating BDAM, ERM, and FFM sequentially. Table 4 presents the contributions of each module and their combinations.

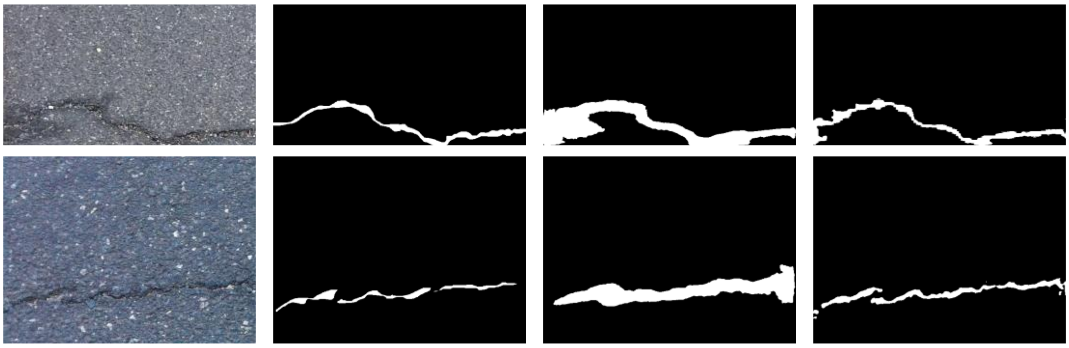
Table 4. The impact of BDAM, ERM, and FFM on network performance.

	IoU(%)	mIoU(%)	Re(%)	Acc(%)
Original	62.71	80.02	76.03	97.48
Original+BDAM	63.42	80.41	76.36	97.51
Original+ERM	63.72	80.54	78.56	97.47
Original+BDAM+ERM	65.08	81.28	78.98	97.53
Original+ERM+FFM	63.79	80.57	79.06	97.47
Original+BDAM+ERM+FFM	66.21	82.48	80.14	98.51

BDAM: In the second experiment, the BDAM enhanced multiscale receptive field information, resulting in a 0.71% improvement in IoU compared to the original. This demonstrates the ability of BDAM to capture global context information.

ERM: In the third experiment, the side input module's enhanced edge positioning led to a 1.66% improvement in crack IoU and a 2.62% increase in recall. This demonstrates that the side input module not only increases the accuracy of edge pixel segmentation but also enhances the overall segmentation accuracy for the category. Figure 11 shows the segmentation results before and after ERM is added, which demonstrates that ERM can effectively improve the segmentation accuracy of crack boundaries.

FFM: In the fourth experiment, FFM was employed to replace the default feature concatenation operation, resulting in a 1.13% improvement in IoU. This suggests that the FFM-based coordinate feature guidance efficiently encodes multi-element and multiscale information.



Original Ground Truth ERNet without ERM ERNet

Figure 11. The visualization of edge reconstruction results.

3.6. Crack Information Quantification Experiment Results

We simulated the actual detection process, input the image of Crack500 dataset into ERNet to get the prediction map, and obtain the calculated value of crack length and width of the prediction map through the proposed crack information quantization algorithm. Two experiments were designed to verify the accuracy of the quantization results and the accuracy of the crack prediction results.

We select 10 sets of calculated prediction values to compare with the calculated label values, in order to verify the effectiveness of the crack information quantization algorithm we proposed above. And select another 10 sets of calculated prediction values to compare with the measured values of the original RGB images, in order to verify the accuracy of the crack prediction results. To assess the universality of the results, we introduced the average absolute error and average relative error. The detailed results are presented in Table 5 and Table 6.

Table 5. Comparison table of ground-truth and prediction crack calculated parameters.

	crack length and error(pixel)				**crack width and error(pixel)**			
number	calculated label value	calculated prediction value	absolute error	relative error/%	calculated label value	calculated prediction value	absolute error	relative error/%
1	640	691	51	7.97	38	42	4	10.53
2	409	441	32	7.82	34	37	3	8.82
3	638	651	13	2.04	64	66	2	3.13
4	339	368	29	8.55	24	26	2	8.33
5	238	272	34	14.29	80	80	0	0.00
6	150	151	1	0.67	37	31	6	16.22
7	286	324	38	13.29	46	49	3	6.52
8	158	175	17	10.76	26	25	1	3.85
9	247	264	17	6.88	44	42	2	4.55
10	355	365	10	2.82	40	42	2	5.00
average			**24.2**	**7.51**			**2.5**	**6.69**

The crack length in the comparison ranges from 150 to 697 pixels, and the width ranges from 24 to 80 pixels. The experimental results show that the average relative errors of the calculated values of crack length and width relative to the labeled cracks are 7.51% and 6.69% respectively, and the average relative errors of the calculated values relative to the original image are 5.85% and 3.92% respectively. Since the crack information measurement based on the original image is manually measured, there will inevitably be errors due to human factors and instrument influences.

Table 6. Comparison table of measured and calculated crack parameters.

	crack length and error(pixel)				**crack width and error(pixel)**			
number	measured value	calculated prediction value	absolute error	relative error/%	measured value	calculated prediction value	absolute error	relative error/%
1	341	323	18	5.28	27	26	1	3.70
2	697	691	6	0.86	44	39	5	11.36
3	405	441	36	8.89	24	24	0	0.00
4	638	651	13	2.04	27	26	1	3.70
5	623	721	98	15.73	30	31	1	3.33
6	166	151	15	9.04	80	80	0	0.00
7	522	500	22	4.21	60	56	4	6.67
8	215	208	7	3.26	50	51	1	2.00
9	375	351	24	6.40	26	25	1	3.85
10	355	365	10	2.82	44	42	2	4.55
average			**24.9**	**5.85**			**1.6**	**3.92**

4. Discussion

Our research aims to automatically detect cracks from road images, a novel method for efficient detection and extraction of road crack is proposed. A segmentation network based on edge reconstruction is utilized to achieve real-time detection of road cracks. To improve the accuracy of edge reconstruction and to guide global detection, we introduce a soft-gate control mechanism to fuse high-level gradient semantic information. In addition, we propose a depth-decomposed convolutional attention module that utilizes deep and dilated convolution techniques to process global contextual information of images. Crack detection results are automatically quantized to extract the length and width of the crack backbone. The experimental results show that our method outperforms other comparative methods. From the experimental results in Section 3.3, it can be seen that CT-CrackSeg is better at detecting fine cracks because they retain shallow information, but it is less effective at detecting road cracks in the presence of complex background disturbances. LinkCrack and PIDNet have better results with complex backgrounds, but they miss the detection of fine cracks and have poorer coherence in the presence of fine cracks. Noting the phenomenon that edge information favors detail information and deep information favors semantic information, our method employs the ERM module for selective enhancement of edge information, and the quantitative and visualization results show that the method has good performance for edge localization of cracks. From Table 1 and Table 2, we can find that our method is faster than some networks but slower than STDC-Seg network, which needs further improvement in the future.

Author Contributions: J.L. supervised the study, designed the architecture, and revised the manuscript; Z.D. wrote the manuscript and designed the comparative experiments. X.L made suggestions to the manuscript and assisted Z.D. in conducting the experiments. J.Z. and Z.L. made suggestions for the experiments and assisted in revising the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Innovative talent program of Jiangsu under Grant JSSCR2021501, and by the High-level talent plan of NUAA, China.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the editors and the reviewers for their valuable suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zheng, M.; Lei, Z.; Zhang, K. Intelligent Detection of Building Cracks Based on Deep Learning. Image and Vision Computing 2020, 103, 103987, doi:10.1016/j.imavis.2020.103987.
2. Wu, C.; Sun, K.; Xu, Y.; Zhang, S.; Huang, X.; Zeng, S. Concrete Crack Detection Method Based on Optical Fiber Sensing Network and Microbending Principle. Safety Science 2019, 117, 299–304, doi:10.1016/j.ssci.2019.04.020.
3. Kim, B.; Yuvaraj, N.; Sri Preethaa, K.R.; Arun Pandian, R. Surface Crack Detection Using Deep Learning with Shallow CNN Architecture for Enhanced Computation. Neural Comput & Applic 2021, 33, 9289–9305, doi:10.1007/s00521-021-05690-8.
4. Gavilán, M.; Balcones, D.; Marcos, O.; Llorca, D.F.; Sotelo, M.A.; Parra, I.; Ocaña, M.; Aliseda, P.; Yarza, P.; Amírola, A. Adaptive Road Crack Detection System by Pavement Classification. Sensors 2011, 11, 9628–9657, doi:10.3390/s111009628.
5. Jahanshahi, M.R.; Jazizadeh, F.; Masri, S.F.; Becerik-Gerber, B. Unsupervised Approach for Autonomous Pavement-Defect Detection and Quantification Using an Inexpensive Depth Sensor. J. Comput. Civ. Eng. 2013, 27, 743–754, doi:10.1061/(ASCE)CP.1943-5487.0000245.
6. Zhang, D.; Zou, Q.; Lin, H.; Xu, X.; He, L.; Gui, R.; Li, Q. Automatic Pavement Defect Detection Using 3D Laser Profiling Technology. Automation in Construction 2018, 96, 350–365, doi:10.1016/j.autcon.2018.09.019.
7. Zhong, X.; Peng, X.; Yan, S.; Shen, M.; Zhai, Y. Assessment of the Feasibility of Detecting Concrete Cracks in Images Acquired by Unmanned Aerial Vehicles. Automation in Construction 2018, 89, 49–57, doi:10.1016/j.autcon.2018.01.005.
8. Peng, X.; Zhong, X.; Zhao, C.; Chen, A.; Zhang, T. A UAV-Based Machine Vision Method for Bridge Crack Recognition and Width Quantification through Hybrid Feature Learning. Construction and Building Materials 2021, 299, 123896, doi:10.1016/j.conbuildmat.2021.123896.
9. Peng, X.; Zhong, X.; Zhao, C.; Chen, Y.F.; Zhang, T. The Feasibility Assessment Study of Bridge Crack Width Recognition in Images Based on Special Inspection UAV. Advances in Civil Engineering 2020, 2020, 1–17, doi:10.1155/2020/8811649.
10. Mazzini, D.; Napoletano, P.; Piccoli, F.; Schettini, R. A Novel Approach to Data Augmentation for Pavement Distress Segmentation. Computers in Industry 2020, 121, 103225, doi:10.1016/j.compind.2020.103225.
11. Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; Ling, H. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. IEEE Trans. Intell. Transport. Syst. 2020, 21, 1525–1535, doi:10.1109/TITS.2019.2910595.
12. Nguyen, T.S.; Begot, S.; Duculty, F.; Avila, M. Free-Form Anisotropy: A New Method for Crack Detection on Pavement Surface Images. In Proceedings of the 2011 18th IEEE International Conference on Image Processing; IEEE: Brussels, Belgium, September 2011; pp. 1069–1072.
13. Liang, X. Image - based Post - disaster Inspection of Reinforced Concrete Bridge Systems Using Deep Learning with Bayesian Optimization. Computer - Aided Civil and Infrastructure Engineering 2019, 34, 415–430, doi:10.1111/mice.12425.
14. Du, P.; Bai, X.; Tan, K.; Xue, Z.; Samat, A.; Xia, J.; Li, E.; Su, H.; Liu, W. Advances of Four Machine Learning Methods for Spatial Data Handling: A Review. J geovis spat anal 2020, 4, 13, doi:10.1007/s41651-020-00048-5.
15. Abou-Chacra, D.; Zelek, J. Effects of Spatial Transformer Location on Segmentation Performance of a Dense Transformer Network. J. Comp. Vis. Imag. Sys. 2017, 3, doi:10.15353/vsnl.v3i1.169.
16. Islam, M.M.M.; Kim, J.-M. Vision-Based Autonomous Crack Detection of Concrete Structures Using a Fully Convolutional Encoder–Decoder Network. Sensors 2019, 19, 4251, doi:10.3390/s19194251.
17. König, J.; Jenkins, M.D.; Mannion, M.; Barrie, P.; Morison, G. Optimized Deep Encoder-Decoder Methods for Crack Segmentation. Digital Signal Processing 2021, 108, 102907, doi:10.1016/j.dsp.2020.102907.
18. Liu, Y.; Yao, J.; Lu, X.; Xie, R.; Li, L. DeepCrack: A Deep Hierarchical Feature Learning Architecture for Crack Segmentation. Neurocomputing 2019, 338, 139–153, doi:10.1016/j.neucom.2019.01.036.

19. Ren, Y.; Huang, J.; Hong, Z.; Lu, W.; Yin, J.; Zou, L.; Shen, X. Image-Based Concrete Crack Detection in Tunnels Using Deep Fully Convolutional Networks. Construction and Building Materials 2020, 234, 117367, doi:10.1016/j.conbuildmat.2019.117367.
20. Liu, Z.; Cao, Y.; Wang, Y.; Wang, W. Computer Vision-Based Concrete Crack Detection Using U-Net Fully Convolutional Networks. Automation in Construction 2019, 104, 129–139, doi:10.1016/j.autcon.2019.04.005.
21. Wang, J.; Zeng, Z.; Sharma, P.K.; Alfarraj, O.; Tolba, A.; Zhang, J.; Wang, L. Dual-Path Network Combining CNN and Transformer for Pavement Crack Segmentation. Automation in Construction 2024, 158, 105217, doi:10.1016/j.autcon.2023.105217.
22. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. In Computer Vision – ECCV 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2018; Vol. 11217, pp. 334–349 ISBN 978-3-030-01260-1.
23. Fan, M.; Lai, S.; Huang, J.; Wei, X.; Chai, Z.; Luo, J.; Wei, X. Rethinking BiSeNet For Real-Time Semantic Segmentation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Nashville, TN, USA, June 2021; pp. 9711–9720.
24. Liao, J.; Yue, Y.; Zhang, D.; Tu, W.; Cao, R.; Zou, Q.; Li, Q. Automatic Tunnel Crack Inspection Using an Efficient Mobile Imaging Module and a Lightweight CNN. IEEE Trans. Intell. Transport. Syst. 2022, 23, 15190–15203, doi:10.1109/TITS.2021.3138428.
25. Yang, X.; Li, H.; Yu, Y.; Luo, X.; Huang, T.; Yang, X. Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network: Pixel-Level Crack Detection and Measurement Using FCN. Computer-Aided Civil and Infrastructure Engineering 2018, 33, 1090–1109, doi:10.1111/mice.12412.
26. Li, G.; Liu, Q.; Ren, W.; Qiao, W.; Ma, B.; Wan, J. Automatic Recognition and Analysis System of Asphalt Pavement Cracks Using Interleaved Low-Rank Group Convolution Hybrid Deep Network and SegNet Fusing Dense Condition Random Field. Measurement 2021, 170, 108693, doi:10.1016/j.measurement.2020.108693.
27. Tao, H.; Liu, B.; Cui, J.; Zhang, H. A Convolutional-Transformer Network for Crack Segmentation with Boundary Awareness. 2023 IEEE International Conference on Image Processing (ICIP) 2023, 86–90, doi:10.1109/ICIP49359.2023.10222276.
28. Pang, J.; Zhang, H.; Zhao, H.; Li, L. DcsNet: A Real-Time Deep Network for Crack Segmentation. SIViP 2022, 16, 911–919, doi:10.1007/s11760-021-02034-w.
29. Xie, S.; Tu, Z. Holistically-Nested Edge Detection. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV); IEEE: Santiago, Chile, December 2015; pp. 1395–1403.
30. Ke, W.; Chen, J.; Jiao, J.; Zhao, G.; Ye, Q. SRN: Side-Output Residual Network for Object Reflection Symmetry Detection and Beyond. IEEE Trans. Neural Netw. Learning Syst. 2021, 32, 1881–1895, doi:10.1109/TNNLS.2020.2994325.
31. Tsai, T.-H.; Tseng, Y.-W. BiSeNet V3: Bilateral Segmentation Network with Coordinate Attention for Real-Time Semantic Segmentation. Neurocomputing 2023, 532, 33–42, doi:10.1016/j.neucom.2023.02.025.
32. Zhang, M.; Zhang, R.; Yang, Y.; Bai, H.; Zhang, J.; Guo, J. ISNet: Shape Matters for Infrared Small Target Detection. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: New Orleans, LA, USA, June 2022; pp. 867–876.
33. Hung, W.-C.; Tsai, Y.-H.; Shen, X.; Lin, Z.; Sunkavalli, K.; Lu, X.; Yang, M.-H. Scene Parsing with Global Context Embedding. 2017 IEEE International Conference on Computer Vision (ICCV) 2017, 2650–2658, doi:10.1109/ICCV.2017.287.
34. Liu, H.; Peng, C.; Yu, C.; Wang, J.; Liu, X.; Yu, G.; Jiang, W. An End-To-End Network for Panoptic Segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2019, 6165–6174, doi:10.1109/CVPR.2019.00633.
35. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, 2818–2826, doi:10.1109/CVPR.2016.308.
36. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation 2017.
37. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Computer Vision – ECCV 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2018; Vol. 11211, pp. 833–851 ISBN 978-3-030-01233-5.
38. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. In Proceedings of the 2021 IEEE/CVF

International Conference on Computer Vision (ICCV); IEEE: Montreal, QC, Canada, October 2021; pp. 548–558.

39. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); IEEE: Montreal, QC, Canada, October 2021; pp. 9992–10002.
40. He, X.; Mo, Z.; Wang, P.; Liu, Y.; Yang, M.; Cheng, J. ODE-Inspired Network Design for Single Image Super-Resolution. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Long Beach, CA, USA, June 2019; pp. 1732–1741.
41. Hu, P.; Caba, F.; Wang, O.; Lin, Z.; Sclaroff, S.; Perazzi, F. Temporally Distributed Networks for Fast Video Semantic Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Seattle, WA, USA, June 2020; pp. 8815–8824.
42. Zheng, X.; Huan, L.; Xia, G.-S.; Gong, J. Parsing Very High Resolution Urban Scene Images by Learning Deep ConvNets with Edge-Aware Loss. ISPRS Journal of Photogrammetry and Remote Sensing 2020, 170, 15–28, doi:10.1016/j.isprsjprs.2020.09.019.
43. Zhang, T.Y.; Suen, C.Y. A Fast Parallel Algorithm for Thinning Digital Patterns. Commun. ACM 1984, 27, 236–239, doi:10.1145/357994.358023.
44. Russell, B.C.; Torralba, A.; Murphy, K.P.; Freeman, W.T. LabelMe: A Database and Web-Based Tool for Image Annotation. Int J Comput Vis 2008, 77, 157–173, doi:10.1007/s11263-007-0090-8.
45. Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; Ling, H. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. IEEE Trans. Intell. Transport. Syst. 2020, 21, 1525–1535, doi:10.1109/TITS.2019.2910595.
46. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. Commun. ACM 2017, 60, 84–90, doi:10.1145/3065386.
47. Xu, J.; Xiong, Z.; Bhattacharyya, S.P. PIDNet: A Real-Time Semantic Segmentation Network Inspired by PID Controllers. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); IEEE: Vancouver, BC, Canada, June 2023; pp. 19529–19539.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.