fractal and fractional

Article

An Algorithm for Crack Detection, Segmentation, and Fractal Dimension Estimation in Low-Light Environments by Fusing FFT and Convolutional Neural Network

Jiajie Cheng, Qiunan Chen * and Xiaocheng Huang

School of Civil Engineering, Hunan University of Science and Technology, Xiangtan 411201, China;
21010201005@mail.hnust.edu.cn (J.C.); xchuang@cqu.edu.cn (X.H.)
* Correspondence: cqndoc@163.com; Tel.: +86-139-7326-1005

Abstract: The segmentation of crack detection and severity assessment in low-light environments presents a formidable challenge. To address this, we propose a novel dual encoder structure, denoted as DSD-Net, which integrates fast Fourier transform with a convolutional neural network. In this framework, we incorporate an information extraction module and an attention feature fusion module to effectively capture contextual global information and extract pertinent local features. Furthermore, we introduce a fractal dimension estimation method into the network, seamlessly integrated as an end-to-end task, augmenting the proficiency of professionals in detecting crack pathology within low-light settings. Subsequently, we curate a specialized dataset comprising instances of crack pathology in low-light conditions to facilitate the training and evaluation of the DSD-Net algorithm. Comparative experimentation attests to the commendable performance of DSD-Net in low-light environments, exhibiting superlative precision (88.5%), recall (85.3%), and F1 score (86.9%) in the detection task. Notably, DSD-Net exhibits a diminutive Model Size (35.3 MB) and elevated Frame Per Second (80.4 f/s), thereby endowing it with the potential to be seamlessly integrated into edge detection devices, thus amplifying its practical utility.

Keywords: crack detection; crack segmentation; deep learning; fractal dimension estimation; low-light environment

check for updates

Citation: Cheng, J.; Chen, Q.; Huang, X. An Algorithm for Crack Detection, Segmentation, and Fractal Dimension Estimation in Low-Light Environments by Fusing FFT and Convolutional Neural Network. *Fractal Fract.* **2023**, *7*, 820. https://doi.org/10.3390/fractalfract7110820

Academic Editor: Zine El Abiddine Fellah

Received: 24 September 2023
Revised: 18 October 2023
Accepted: 22 October 2023
Published: 14 November 2023

1. Introduction

Cracks, as apparent damage to infrastructure construction, can lead to safety hazards, structural damage, and high maintenance costs and can also be considered as an early warning phenomenon before serious damage to the building structure occurs. Timely detection of surface cracks in concrete facilities and determining their severity can help develop a scientific maintenance program to avoid disaster. Traditional manual crack disease detection methods are time-consuming, labor-intensive, subjective, and have limited efficiency and credibility. In recent years, with the development of image processing technology and machine learning technology, scholars have conducted much research on crack detection technology for identifying and quantifying crack disease and have developed many computer vision-based crack detection algorithms.

These algorithms can be divided into two categories: target detection and semantic segmentation. Target detection methods can use bounding boxes to help accurately locate and mark the position of concrete cracks, while semantic segmentation separates cracks from the background and provides information about crack boundaries. In practice, it is important to locate the cracks and obtain their segmentation information [1]. Earlier algorithms based on traditional image processing, such as thresholding [2,3], edge detection [4], morphological methods [5], and texture feature methods [6], locate cracks by pixel-level processing and feature extraction from the image. In the case where the environmental pixels are similar to the crack pixels, they often cannot meet the application requirements.

With the development of artificial intelligence technology in recent years, deep learning-based crack detection algorithms have become a mainstream method due to their potential to improve efficiency and accuracy.

Convolutional neural networks (CNNs) can effectively capture local features due to their local perception property, avoiding the complex process of feature extraction and data reconstruction in traditional recognition algorithms. Faster-RCNN [7] and YOLO [8] are two of the most representative target detection algorithms based on CNN. To improve the accuracy of crack detection and to meet the needs of real-world deployment, researchers have improved these networks in various ways. Sekar et al. [9] used the global average pool (GAP) and Region of Interest (RoI) alignment techniques to reduce information loss and detect road cracks. Li et al. [10] proposed a new proposal generation architecture to solve the problem of detecting small defects; Zhou et al. [11] solved the problem of YOLO v4 training by introducing residual blocks and an efficient channel attention module to address the problems of gradient vanishing and gradient explosion may occur during YOLO v4 training; Chu et al. [12] proposed a multiscale feature fusion network TCN with attention mechanism for learning the hierarchical features of tiny cracks. In addition, some feature fusion modules, such as the skip-squeeze-and-excitation module (SE) [13] and convolutional block attention module (CBAM) [14], can be added to the network to enhance the model's multiscale feature extraction capability. However, CNN models usually perform convolutional operations based on local receptive fields, which cannot establish the dependencies between global features, have difficulty dealing with complex crack patterns, and are less adaptable to different environmental conditions. Capturing targets in low light is still a difficult task.

Semantic segmentation is similar to target detection, with the difference that semantic segmentation is pixel-level classification, which assigns a label to each pixel in the image and can delineate crack boundaries. The full convolutional network (FCN) is the first deep learning-based segmentation method. There are also several segmentation models built using CNNs with an encoder-decoder architecture, such as SegNet, U-Net, PSPNet, and DeepLab. These native models can be applied in the field of crack detection. Huang et al. [15] used a full convolutional neural network model for iterative training based on deep learning principles as a way to perform intelligent semantic segmentation of defective images; Shang [16] et al. proposed the fusion of the dual attention mechanism and atrous spatial pyramid pooling (ASPP) of the U-Net network. The dual-encoder structure shows advantages in capturing local and global contextual information. Xiang [17] et al. proposed dual-encoder structure crack segmentation algorithms fusing transformers and convolutional neural networks. However, there is still a need to address the limitations of the existing segmentation methods, such as the difficulty in handling cracks of varying complexity and the inability to meet the challenge of segmentation under low-light conditions.

Accurately determining the severity of surface cracks on a structure helps to assess the condition of the structure to determine the repair plan, which is important for ensuring the safety of the structure. Existing methods for measuring crack severity are mainly based on the morphological characteristics of cracks, i.e., length, width, and area [18]. Using the crack skeleton (medial axis) obtained from the segmentation process, the crack is approximated into a series of curved segments, and the Euclidean distance between the two endpoints of each segment is counted to calculate the length [19], or the shortest path inside the crack is generated using the A^* algorithm [20]. The width and area are calculated based on the positional relationship and the number of pixel points after segmentation [21]. These methods cannot determine the direction of the crack, and the approximations used in the calculation are less accurate. The tunnel lining crack index (TCI) [22,23], which is drawn based on the crack tensor theory, can characterize the distribution and direction of cracks and is widely used in Japan; however, this method does not take into account the intersection of cracks.

Accurate detection, segmentation, and quantification of cracks, especially in low light conditions, remain challenging tasks. Incorporating advanced signal processing techniques like the fast Fourier transform (FFT) can significantly enhance detection effectiveness. The FFT method, known for its efficacy in expediting convolution [24], leverages the principle that convolution in the temporal domain is equivalent to convolution in the frequency domain. Additionally, the application of fractal dimension analysis for crack quantification has demonstrated effectiveness [25]. This study endeavors to integrate fractal dimension estimation methods into the crack detection network, thereby empowering professionals to assess crack severity with greater efficiency. Our main contributions are as follows:

1. Supplementary crack detection dataset with the open source crack segmentation dataset for network training and testing.
2. Designed an automated methodology capable of detecting, segmenting, and estimating the fractal dimension of cracks in an end-to-end manner called DSD-Net.
3. Designed a dual encoder structure based on FFT and CNN. The FFT-based target detection module effectively captures crack patterns and enhances crack localization. The CNN-based segmentation module accurately delineates the crack boundary by considering local and global context information.
4. Extensive performance evaluation of the proposed method using several evaluation metrics and comparison with mainstream sum detection and segmentation methods.

2. Methods

2.1. Crack Detection and Segmentation System

As shown in Figure 1, in our proposed deep learning model for locating and segmenting concrete cracks, FFT-based frequency-domain convolution is used as the first compiler module to acquire and locate the high-frequency and low-frequency information in the image. The spatial domain convolutional CNN branch has a powerful fine-grained feature extraction capability due to its unique structural properties [26], which can effectively extract feature information from crack images. To enhance its ability to extract global information from images, an information extraction module is designed in this paper to achieve contextual aggregation of different regions and is embedded in the CNN branch. A spatial attention mechanism feature fusion detection module (ADM) is designed to achieve the fusion of encoded training weight maps and decoded feature information.

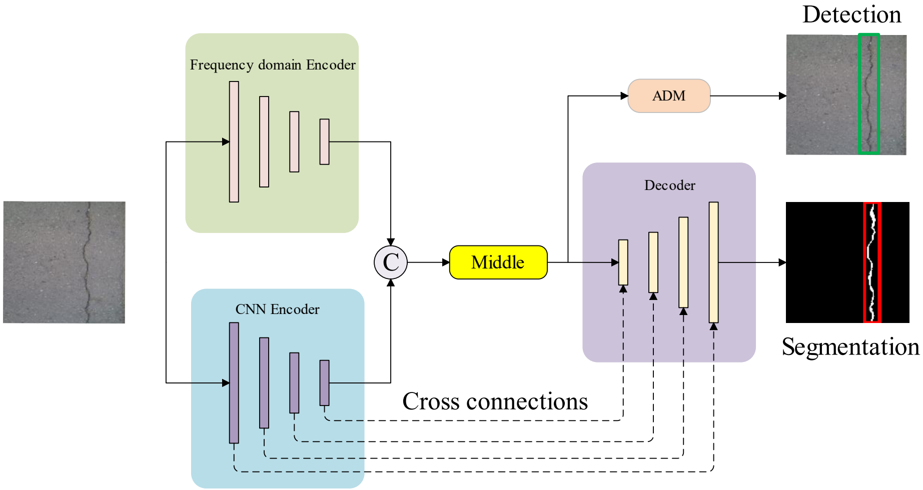

Figure 1. DSD-Net Network Architecture.

The role and detailed structure of each network component is described below.

2.1.1. Frequency Domain Encoder

Recent studies on recurrent neural networks have shown that complex numbers may have a richer ability to characterize information [27,28]. Complex number neural networks can better capture detailed image features and reduce the need for data enhancement by preserving phase information when processing images. To perform the equivalent of conventional 2-dimensional convolution in the complex domain, in the frequency domain encoder proposed in this paper, the image is first subjected to a downsampling operation to reduce the number of operations and then input into the frequency domain structure FFT Neck after a 3 × 3 convolution. The FFT Neck consists of a fast Fourier positive inverse transform and a complex convolution operation, as shown in Figure 2.

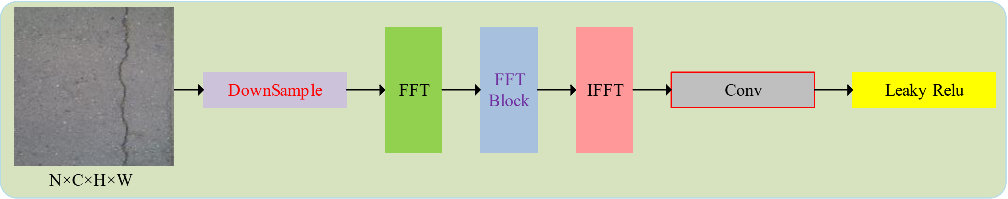

Figure 2. Diagram of FFT Neck.

The Fourier transform transforms the spatial domain representation and frequency domain representation of an image. The input 2D grayscale image can be considered as a grayscale matrix, which is transformed into a frequency distribution function by a 2D off Fourier transform, which is beneficial for extraction to achieve finer feature extraction and crack edge detection. The Fourier positive inverse transform is given by the following equation [29]:

$$F(u,v) = \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)e^{\frac{-2\pi i}{N}(ux+vy)} \tag{1}$$

$$f(x,y) = \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} F(u,v)e^{\frac{-2\pi i}{N}(ux+vy)} \tag{2}$$

where $f(x,y)$ is the preprocessed image, N is the number of pixels in the x and y directions ($N = 256$ in our model), and u and v are the new coordinates corresponding to the spatial frequencies in directions x and y, respectively. Matrix notation is used to represent the real and imaginary parts of the convolution operation [30]:

$$\begin{bmatrix}\Re(W \times h)\\ \Im(W \times h)\end{bmatrix} = \begin{bmatrix}A & -B\\ B & A\end{bmatrix} \times \begin{bmatrix}x\\ y\end{bmatrix} \tag{3}$$

where A and B are real matrices and x and y are real vectors. These $N \times N$ matrices containing complex numbers will be fed into the coding module FFT Block. The FFT block consists of two layers of complex convolutional layers (kernel size = 3, stride = 1, padding = 1) and two layers of complex residual structure as shown in Figure 3.

To perform backpropagation in complex-valued neural networks, it is a sufficient condition that the cost function and the activation are microscopic for the real and imaginary parts of each complex parameter in the network. Although some authors have shown that restricting the activation function to all-pure functions is not necessary [30], for computational efficiency purposes, the use of complex ReLU (ℂReLU) sharing gradient values that satisfy the Cauchy-Riemann equation is used. Complex activation of ReLU applied to real and imaginary parts of neurons separately, where $z \in \mathbb{C}$ [30]:

$$\mathbb{C}ReLU(z) = ReLU(\Re(z)) + iReLU(\Im(z)) \tag{4}$$

As in Figure 3, in Complex Res, jump connections are used between input and output feature mappings [31]. Similar to the spatial domain residual structure, residual networks constructed in this way are easier to optimize [32]. Formally, in this paper, we consider the building blocks defined as:

$$X_{out} = X + 3Complex\_Conv(X) \tag{5}$$

X is the feature mapping of the input, X_{out} is the feature mapping of the output, and $Complex\_Conv$ is a 3 × 3 complex convolutional layer.

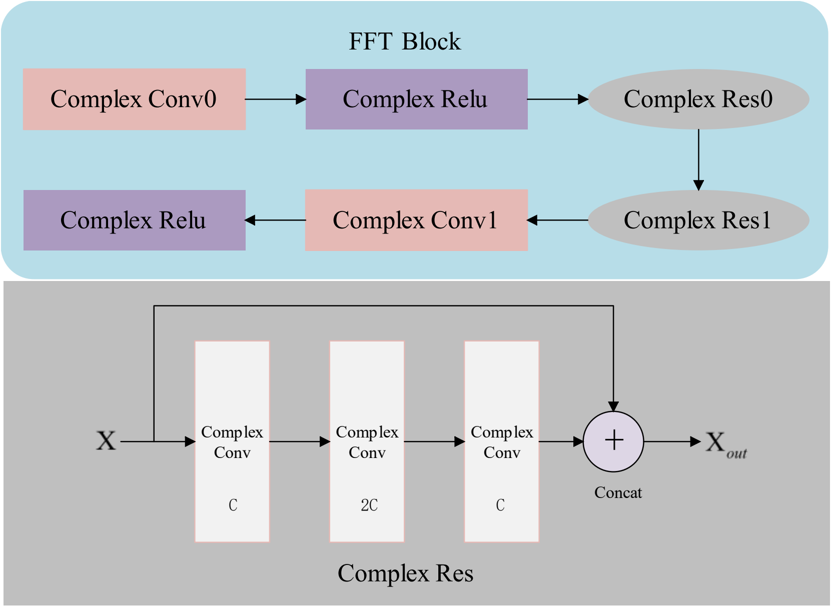

Figure 3. Diagram of the FFT block.

2.1.2. CNN Encoder

The feature pyramid is a computer vision technique for multiscale target detection and image segmentation. It extracts and fuses multiscale feature information in different levels of feature maps to capture target or image details at different scales. Inspired by the pyramid scene parsing network (PSPNet) [33] and strip pooling [34], in this paper, we constructed a feature pyramid-based information extraction module strip extractor, as shown in Figure 4.

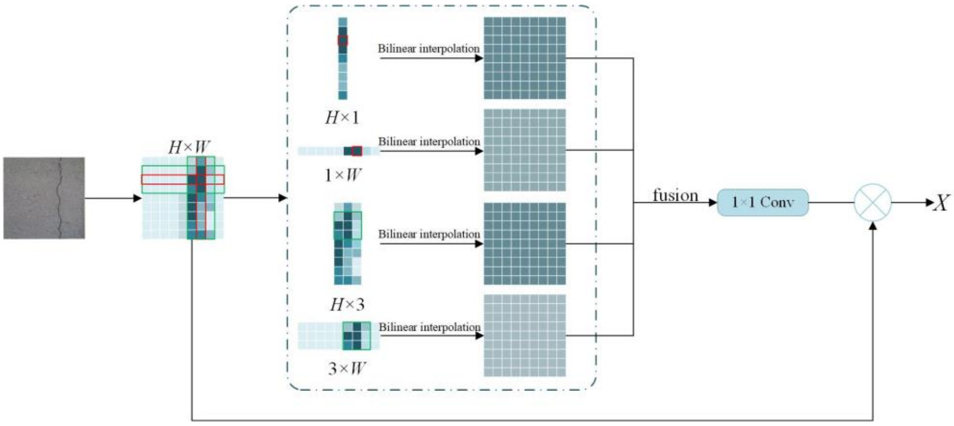

Figure 4. Diagram of the Strip-Extractor.

This module can mine global context information based on context aggregation in different regions. In the strip extractor, we make all the convolutional layers share the same number of channels with the input vectors, split the original input into four bin-sized feature maps (1 × H, 1 × W, H × 3, 3 × W) by a strip pooling layer, and then obtain the same size as the original feature maps by bilinear interpolation and splicing. Finally, it is superimposed with the original feature map. For a two-dimensional tensor, $X \in \mathbb{R}^{H \times W}$, the horizontal output $y^h \in \mathbb{R}^H$ is [34]:

$$y_i^h = \frac{1}{W} \sum_{0 \leq j < W} x_{i,j} \tag{6}$$

Similarly, the vertical output $y^v \in \mathbb{R}^W$ is [34]:

$$y_i^v = \frac{1}{H} \sum_{0 \leq i < W} x_{i,j} \tag{7}$$

The CNN encoder is an alternating stack of four strip extractors and maximum pooling layers, as in Figure 5:

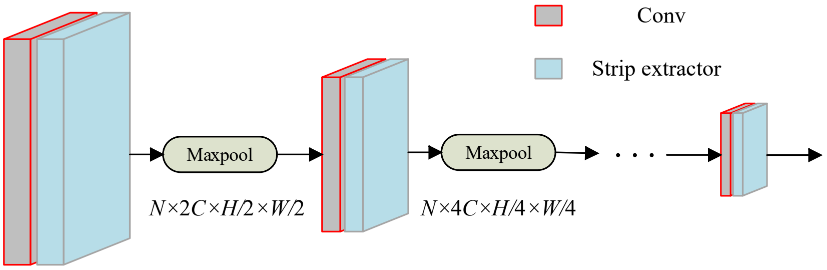
Conv
Strip extractor
Maxpool
Maxpool
$N{\times}2C{\times}H/2{\times}W/2$
$N{\times}4C{\times}H/4{\times}W/4$
...

Figure 5. Diagram of the CNN Encoder.

The feature map passes through each maximal pooling layer, where the number of channels C is doubled while the spatial height H and spatial width W are halved. The CNN encoder collects contextual global information by capturing remote connections in isolated regions through successive stripe pooling and extracts local semantic information from the original cracked image through stepwise maximal pooling.

2.1.3. Attention Fusion Detection Module

Conventional decoder-encoder architectures usually process the entire image as a whole output, which makes it tend to be heavily influenced by background and other unimportant information, leading to inaccurate crack detection. The nature of the attention mechanism is to calculate the attention weights to show the importance of different elements in the input, which is beneficial to mining the overall contextual and channel information and is beneficial for crack detection [35]. In this paper, a spatial attention mechanism feature fusion detection module (ADM) is constructed based on ResNet 34. By applying attention to the feature information, the feature fusion detection module can more accurately locate the crack category in the image, and its work is schematically shown in Figure 6.

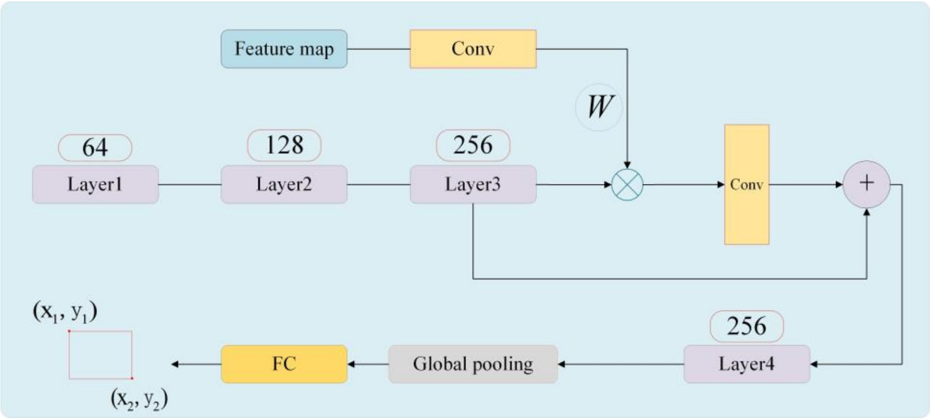

Figure 6. Attention mechanism feature fusion detection module.

Crack localization is based on a large global receptive domain. By sequentially applying linear transformations and nonlinear activation functions, the feature map x^l, which is progressively decoded by the decoder, is used as the header input and passes through three-layer layers, each of which doubles the number of channels, eventually becoming 256 channels. The undecoded feature weights, W, are increased in channels using convolutional layers, again becoming 256 channels. Both are then weighted and aggregated by the dot product method and, after convolution, are superimposed with the original feature map. The global pooling and fully connected layer (FC) maps the feature space to the sample labeling space by linear variation, outputting the crack localization coordinates.

2.1.4. Loss Functions

Choosing an appropriate loss function can help the model fit the data better and improve detection and segmentation performance. The cross-entropy loss function is a common loss function in image segmentation tasks, but the problem of positive and negative sample imbalance in crack images makes it less effective. Positive and negative sample imbalance means that a large number of predicted anchor frames select the background (negative samples), and a small number of anchor frames select the true target (positive samples). To address this problem, we view crack segmentation as a two-stage process. For an image containing a crack, the detection part of the model locates the crack for framing by global information analysis, and then the segmentation head segments the crack. We calculate L1 loss and focal loss based on cross-entropy improvement [36] for the two processes.

L1 loss, also known as mean absolute error (MAE), is the average of the absolute difference between the model's predicted value f(x) and the true value of y. The bounding box regression loss is defined as follows [36]:

$$MAE = \frac{\sum_{i=1}^{n} |f(x_i) - y_i|}{n} \tag{8}$$

The focal loss coordinates the sample imbalance by balancing the cross entropy and increasing the moderator to assign difficult sample training weights. It is defined as follows [36]:

$$FL(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t) \tag{9}$$

$p_t \in [0, 1]$ denotes the predictive confidence score of a candidate object, $(1 - p_t)^\gamma$ is the moderator, and α_t balances the importance of positive and negative samples.

2.2. Fractal Computing System

The fractal dimension (FD) is a mathematical concept used to describe complex geometric structures. Fractal shapes are fractal geometries and approximate self-similar shapes that have similar structures at all scales, and their complexity can be described using the concept of fractal dimension, which is a small number between 1 and 2 [37]. The larger the fractal dimension, the higher the complexity. The box-counting algorithm is a method to calculate the fractal dimension of complete and approximate self-similar patterns and is widely used in many systems in nature or man-made things [38–40].

Algorithm 1 shows the implementation details for estimating the fractal dimension of a crack image. To calculate the fractal dimension of a color crack image, it needs to be grayscaled. The bounding box is defined by the crack detection network section by placing the fixed point of the bounding box (in this case, the lower left corner) at the origin of the Cartesian coordinate system O. A grid of squares r of different sizes is created, and the number of these boxes containing the crack objects en is calculated. The fractal dimension estimate of the crack image is defined as [37]:

$$FD = \lim_{r \to 0} \frac{\log N_r}{\log\left(\frac{1}{r}\right)} \tag{10}$$

where $N(r)$ is the total number of boxes having an r size required to fill the curve totally, and FD stands for the fractal dimension characterizing the aforementioned curve.

Algorithm 1 Calculate Fractal Dimension

```
Require: image_path (image path), min_box_size (minimum box size), max_box_size
    (maximum box size)

Read the image from image_path and convert it to grayscale.
If max_box_size is not provided then
    Set max_box_size to half the minimum dimension of the image.
end if
Function box_count(box_size):
Initialize count to 0.
for each box with size box_size do
    if the box contains a crack then
        Increment count by 1
     end if
end for
return count
Function fractal_dimension():
Initialize an empty list counts.
for each box_size from min_box_size to max_box_size do
        Calculate the number of boxes that cover cracks using box_count(box_size)
and store it in counts.
end for
Fit a line to the pairs of box sizes and counts using the np.polyfit function.
Calculate the fractal dimension using the fitted line.
return Fractal Dimension
Function caculate_fractal_dimension (image_pathe, min_box_size, max_box_size):
Convert the image at image_path to grayscale.
if max_box_size is not provided then
        Set max_box_size to half the minimum dimension of the image.
end if
Call fractal_dimension() to calculate the fractal dimension.
return Fractal Dimension
```

3. Implementation

3.1. Public Crack Datasets

To evaluate the performance of the DSD-Net proposed in this paper, experimental validation was carried out using five open-source crack segmentation datasets, where some low-quality images were excluded from the original datasets. The CFD was 118 datasets reflecting the condition of urban pavements in Beijing, China, captured by an iPhone5, containing noises such as oil spots, water stains, shadows, etc., with the size of 480 × 320 pixels [41]. Concrete crack images for classification (CCIC) contains 445 pavements used in Temple University Smartphone collected pavement images with a pixel size of 3264 × 2448. Crack500 [42] is a dataset of 476 pavement cracks with pixel-level labeling of approximately 2000 × 1500 in size, collected from Temple University's main campus. German Asphalt Pavement Distress (GAPs384) describes German pavement distresses with different categories of pavement distresses such as cracks, potholes, and mosaic patches. We selected 509 of these images with crack distresses with an image resolution of 1920 × 1080 pixels. SDNET2018 [43] is an annotated concrete crack dataset with multiple hindrances, including shadows, surface roughness, scaling, edges, holes, and background fragments, acquired at Utah State University using a digital camera with a resolution of 256 × 256 pixels.

The cracks in these datasets are of varying widths, with different crack patterns, background materials, and a more complete representation of the various types of detection scenarios. The cracks were labeled using the image labeling tool X-AnyLabeling with box selection. To reduce the background pixel weight, only one labeling box was used for each image. The data image brightness was adjusted using gamma correction, i.e., each image (I) was normalized to [0–1], and a power law transformation was performed using the equation O = I^{γ} for performance testing [44]. The divided two datasets (Original, Low brightness) example images with labeling styles are shown in Figure 7.

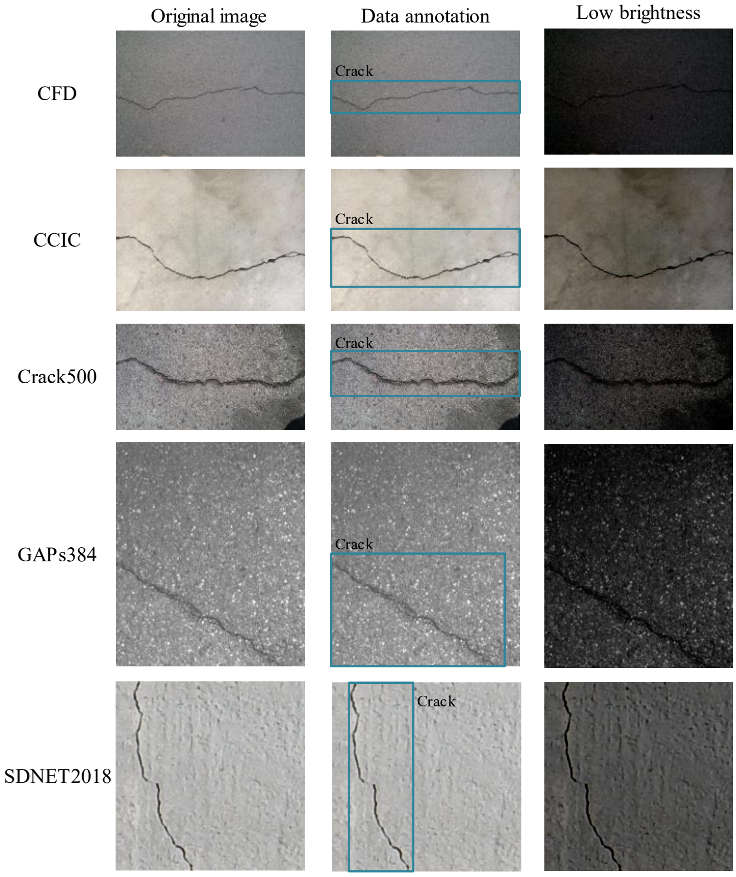

Figure 7. Sample images from the datasets used in this study.

3.2. Implementation Details

The model development language is Python, and the deep learning framework PyTorch 1.10 was used for training and testing. The operating system is Ubuntu 20.04. The graphics processor (GPU) is NVIDIA GeForce RTX 3090.

All experiments use the stochastic gradient descent (SGD) algorithm to update the network weights, with the learning rate initially set to 0.01 momentum to 0.9 and the weights decayed to 0.0005 to share the training pressure and accelerate network convergence. The learning rate decay strategy uses the cosine annealing algorithm. All input images were scaled to 256 × 256 pixels before training. Various data enhancement techniques, such as random rotation, horizontal flipping, and color dithering, are used to increase sample diversity, prevent overfitting, and enhance model generalization. The training loss was chosen to measure the predictive performance of the model and guide the network parameters (weights and biases) during the learning process.

3.3. Evaluation Metrics

DSD-Net's performance in crack detection, segmentation, and complexity recognition is comprehensively evaluated using precision (Pr), recall (Re), F1 score (F1), intersection-to-union ratio (IoU), and frames per second (FPS). Precision and recall provide insight into the accuracy and sensitivity of the model, while the F1 score provides a balance between the two. IoU measures spatial alignment and emphasizes the ability of the model to accurately segment cracks. Finally, model size (MB) and FPS are crucial for evaluating the computational efficiency of the algorithm, with smaller MB and higher FPS values ensuring that it can be deployed in mobile devices and run in real-time scenarios. Specific formulas for these metrics can be found in the literature [45].

4. Experiments and Analyses

We divided the training set and test set using random partitioning at a ratio of 9:1. We selected two classical algorithms in the field of target detection and semantic segmentation for comparison with DSD-Ne. For the crack detection algorithm, we compared it with Faster RCNN, YOLO v5, YOLO v7 [46] and SSD [47]. For crack segmentation algorithms, we chose to compare U-Net [48], Deeplabv3+ [49], PSPNet, and Seg-Net [50]. To be fair, all networks were trained using the same dataset and parameter configuration.

4.1. Crack Detection Results

Pretraining on open-source datasets and then fine-tuning the data can shorten the convergence time of the model [51]; however, the fine-tuning method is ineffective in preventing model overfitting. We want our model to apply to application requirements in various brightness contexts, and our training method does not use other open-source datasets for pretraining but uses a delineated training set. This training method bears the time cost of training the model, with the benefit of facilitating better satisfaction in the accuracy and localization sensitivity requirements of the crack detection task. Table 1 gives the model weight sizes and FPS values for mobile device deployment. The detection performance of each model is evaluated in Table 2. Figure 8 records the accuracy, recall, and F1 scores of the DSD-Net algorithm and other comparative algorithms under normal brightness and low light test sets.

Table 1. Comparison of different target detection algorithms for lightweight.

Method	MS (MB)	FPS (f/s)
Faster RCNN	97.6	25.9
YOLO v5	58.3	30.1
YOLO v7	69.8	28.6
SSD	65.9	57.8
Ours	35.3	80.4

Table 2. Crack Detection Performance Evaluation.

Method	Original			Low Brightness		
	Pr	Re	F1	Pr	Re	F1
Faster RCNN	85.80%	87.10%	86.45%	82.20%	83.60%	82.89%
YOLO v5s	89.20%	84.20%	86.63%	84.30%	80.60%	82.41%
YOLO v7-tiny	88.60%	85.60%	87.07%	84.70%	82.40%	83.53%
SSD	86.40%	85.90%	86.15%	79.80%	73.40%	76.47%
Ours	92.10%	91.40%	91.75%	88.50%	85.30%	86.87%

Figure 9 shows the results of our tests on the proposed method, where the present algorithm can detect the target crack lesions more accurately on both test sets without visual interference. However, concrete splice joints may still be incorrectly recognized as cracks when the light is low.

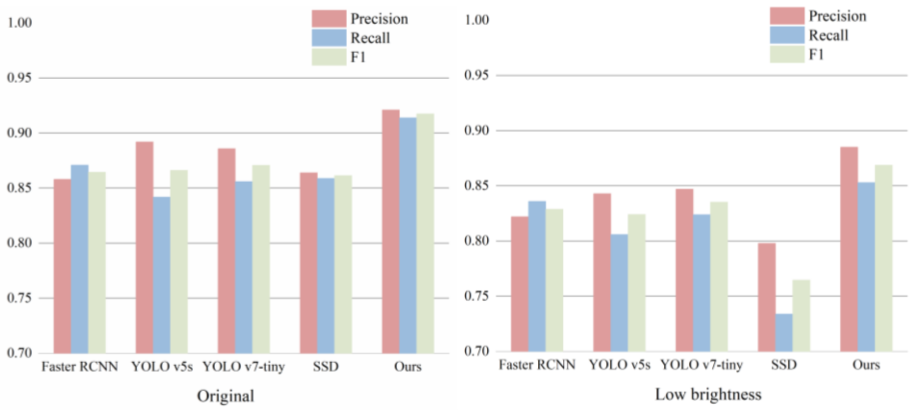

Figure 8. Results of crack detection and evaluation.

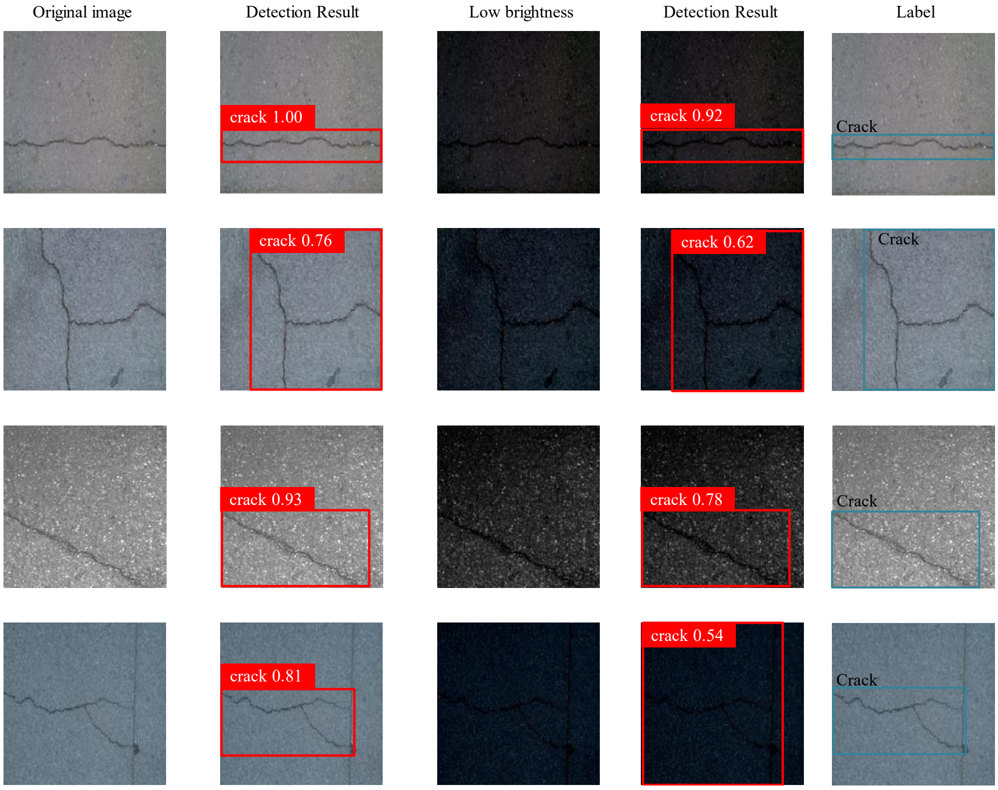

Figure 9. Test results of crack detection.

4.2. Crack Segmentation Results

Similarly, Table 3 evaluates the segmentation performance of each model. As shown in Figure 10, we tested the model's precision, recall, F1, and IoU over other mainstream semantic segmentation models on both test sets. As shown in Figure 11, we tested the proposed model for segmentation, giving the original crack image, the weak light crack image, the segmentation result, and the labeled image, respectively. Our model can accurately locate the crack boundary, produce clear and accurate segmentation masks, and extract most of the concrete crack pixels, and the test results are close to the real situation on the ground. It also shows stability under low-light conditions, as shown in the last set of Figure 11, and can accurately detect crack lesions when judging some cracks that are easily overlooked by human visual inspection.

Table 3. Crack Segmentation Performance Evaluation.

Method	Original				Low Brightness			
	Pr	Re	F1	IoU	Pr	Recall	F1	IoU
U-Net	83.90%	81.20%	82.53%	60.90%	70.40%	75.90%	73.05%	57.80%
Deeplabv3+	84.70%	85.00%	84.85%	63.70%	72.80%	74.60%	73.69%	53.40%
PSPNet	85.60%	81.80%	83.66%	67.00%	73.70%	72.10%	72.89%	51.60%
Seg-Net	82.40%	80.50%	81.44%	60.30%	71.50%	70.80%	71.15%	58.90%
Ours	86.30%	89.20%	87.73%	68.00%	76.30%	78.10%	77.19%	62.60%

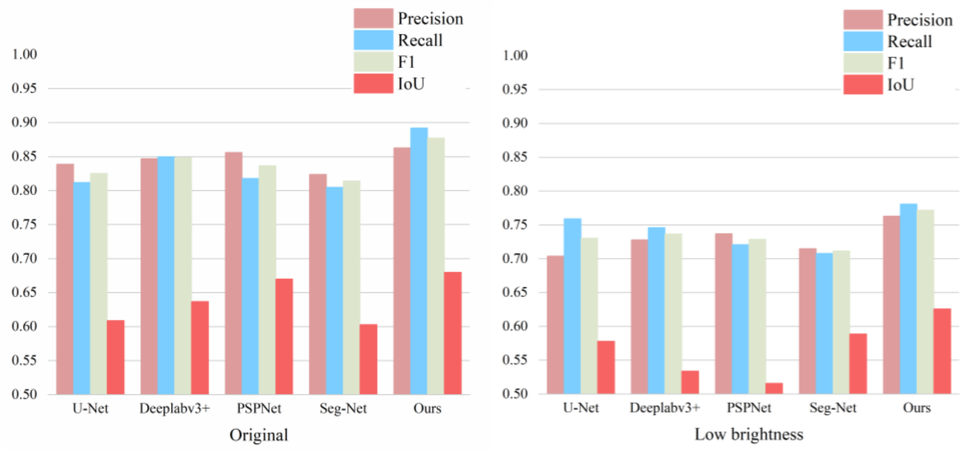

Figure 10. Results of crack segmentation assessment.

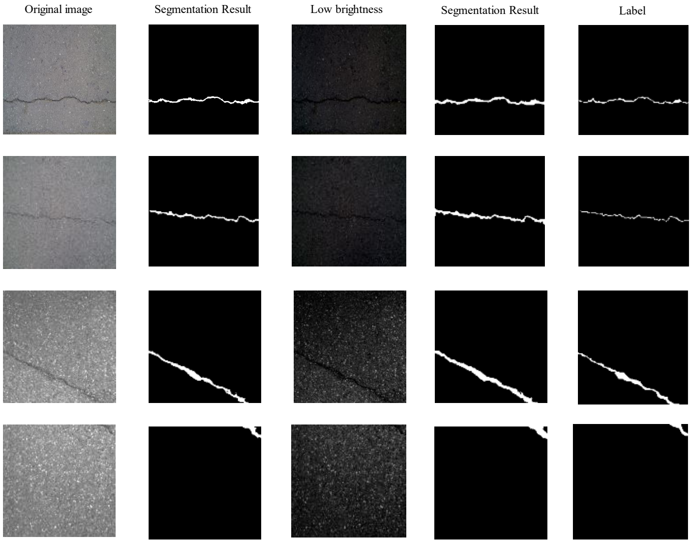

Figure 11. Test results of crack segmentation.

4.3. Fractal Dimension Estimate

Benoit 1.01 [52] software implemented box-counting methods to estimate the approximate fractal dimension (AFD) of 2D and 3D geometric patterns. We compared our proposed estimation method with it and tested the AFDs for straight lines, squares, and Koch curves, as shown in Figure 12. The results of the tests are shown in Table 4, and the error of the fractal method we used is less than 2%.

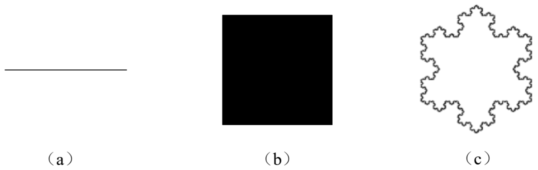

Figure 12. Fractal dimensions of different complex geometric patterns: (**a**) line segments, (**b**) squares, (**c**) sixth-order Koch curves.

Table 4. Fractal dimension estimation test.

Method	Line Segments	Squares	Sixth-Order Koch Curves
Benoit	0.973	1.996	1.268
Ours	0.986	1.994	1.267

Figure 13 demonstrates the results of fractal dimension estimation of crack disease using DSD-Net, which increases as the complexity of the crack increases and can be used to determine the severity of the disease.

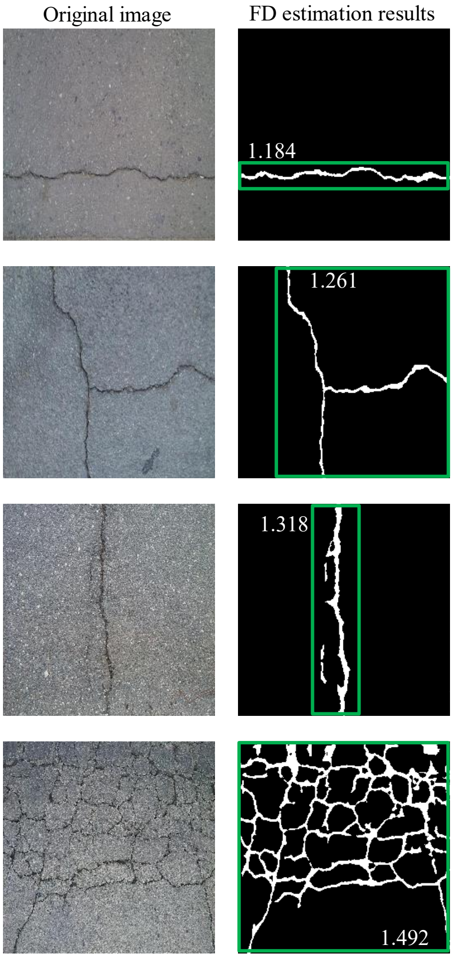

Figure 13. Estimation of the fractal dimension of cracks with different complexities.

5. Conclusions

To fulfill the tasks of crack detection, segmentation, and fractal dimension estimation under low-light conditions, this paper proposes a novel dual encoder network structure (DSD-Net) fusing fast Fourier transform and CNN. The frequency domain branch designs the FFT neck structure for image spatial domain and frequency domain conversion, which is conducive to mitigating the environmental impact of insufficient brightness. The CNN coding branch designs the information extraction module for mining contextual information. The feature fusion detection module is used to apply attention to specific information, which can significantly improve the accuracy of crack disease localization. The fractal system is based on the box-counting method, which estimates the fractal dimension of the crack image after localization and segmentation, providing the basis for crack severity discrimination.

The crack detection and segmentation ability of DSD-Net under low-light conditions was tested using a brightness-adjusted dataset. The results show that the crack detection and segmentation abilities of DSD-Net are better than those of the current mainstream target detection and semantic segmentation algorithms, and they can also have good performance under low-light conditions. In addition, as an end-to-end network integrating crack detection, segmentation, and fractal dimension estimation, DSD-Net has a small model footprint and fast processing speed of a single image, which can be deployed in edge devices to achieve real-time detection of crack damage.

In future work, we will further improve the generalization ability of the model to extend more application scenarios. In addition, we will also work on exploring more accurate fractal dimension estimation for crack segmentation images and researching explicit criteria for assessing the apparent damage of infrastructure using fractal dimension.

Author Contributions: Conceptualization, Q.C.; Methodology, J.C.; Formal analysis, X.H.; Investigation, J.C.; Data curation, J.C.; Writing—original draft, J.C.; Writing—review and editing, J.C.; Funding acquisition, Q.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos. 52078211 and 52279100), the Natural Science Foundation of Hunan Province, China (Nos. 2020JJ4021 and 2021JJ40201), Science and Technology Progress and Innovation Project of Transport Department of Hunan Province (Nos. 202009 and 202308), and Scientific Research Fund of Hunan Provincial Education Department (21B0463).

Data Availability Statement: Data sharing is not applicable to this article.

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

References

1. Chen, K.; Reichard , G.; Xu, X.; Abiola, A. Automated crack segmentation in close-range building façade inspection images using deep learning techniques. *J. Build. Eng.* **2021**, *43*, 102913. [CrossRef]
2. Chen, C.; Seo, H.; Jun, C.; Zhao, Y. A potential crack region method to detect crack using image processing of multiple thresholding. *Signal Image Video Process.* **2022**, *16*, 1673–1681. [CrossRef]
3. Nnolim, U.A. Automated crack segmentation via saturation channel thresholding, area classification and fusion of modified level set segmentation with Canny edge detection. *Heliyon* **2020**, *6*, e05748. [CrossRef] [PubMed]
4. Wang, G.; Peter, W.T.; Yuan, M. Automatic internal crack detection from a sequence of infrared images with a triple-threshold Canny edge detector. *Meas. Sci. Technol.* **2018**, *29*, 025403. [CrossRef]
5. Hong, Y.; Lee, S.J.; Yoo, S.B. AugMoCrack: Augmented morphological attention network for weakly supervised crack detection. *Electron. Lett.* **2022**, *58*, 651–653. [CrossRef]
6. Matlack, G.R.; Horn, A.; Aldo, A.; Walubita, L.F.; Naik, B.; Khoury, I. Measuring surface texture of in-service asphalt pavement: Evaluation of two proposed hand-portable methods. *Road Mater. Pavement Des.* **2023**, *24*, 592–608. [CrossRef]
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
9. Sekar, A.; Perumal, V. Automatic road crack detection and classification using multi-tasking faster RCNN. *J. Intell. Fuzzy Syst.* **2021**, *41*, 6615–6628. [CrossRef]
10. Li, R.; Yuan, Y.; Zhang, W.; Yuan, Y. Unified vision-based methodology for simultaneous concrete defect detection and geolocalization. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 527–544. [CrossRef]
11. Zhou, Z.; Zhang, J.; Gong, C.; Wu, W. Automatic tunnel lining crack detection via deep learning with generative adversarial network-based data augmentation. *Undergr. Space* **2023**, *9*, 140–154. [CrossRef]
12. Chu, H.; Wang, W.; Deng, L. Tiny-Crack-Net: A multiscale feature fusion network with attention mechanisms for segmentation of tiny cracks. *Comput.-Aided Civ. Infrastruct. Eng.* **2022**, *37*, 1914–1931. [CrossRef]

13. Zhang, Y.; Huang, J.; Cai, F. On bridge surface crack detection based on an improved YOLO v3 algorithm. *IFAC-PapersOnLine* **2020**, *53*, 8205–8210. [CrossRef]
14. Chen, L.; Yao, H.; Fu, J.; Ng, C.T. The classification and localization of crack using lightweight convolutional neural network with CBAM. *Eng. Struct.* **2023**, *275*, 115291. [CrossRef]
15. Huang, H.w.; Li, Q.t.; Zhang, D.m. Deep learning based image recognition for crack and leakage defects of metro shield tunnel. *Tunn. Undergr. Space Technol.* **2018**, *77*, 166–176. [CrossRef]
16. Shang, J.; Xu, J.; Zhang, A.A.; Liu, Y.; Wang, K.C.; Ren, D.; Zhang, H.; Dong, Z.; He, A. Automatic Pixel-level pavement sealed crack detection using Multi-fusion U-Net network. *Measurement* **2023**, *208*, 112475. [CrossRef]
17. Xiang, C.; Guo, J.; Cao, R.; Deng, L. A crack-segmentation algorithm fusing transformers and convolutional neural networks for complex detection scenarios. *Autom. Constr.* **2023**, *152*, 104894. [CrossRef]
18. Fan, Z.; Li, C.; Chen, Y.; Di Mascio, P.; Chen, X.; Zhu, G.; Loprencipe, G. Ensemble of deep convolutional neural networks for automatic pavement crack detection and measurement. *Coatings* **2020**, *10*, 152. [CrossRef]
19. Jang, K.; An, Y.K.; Kim, B.; Cho, S. Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot. *Comput.-Aided Civ. Infrastruct. Eng.* **2021**, *36*, 14–29. [CrossRef]
20. Zhao, S.; Zhang, D.; Xue, Y.; Zhou, M.; Huang, H. A deep learning-based approach for refined crack evaluation from shield tunnel lining images. *Autom. Constr.* **2021**, *132*, 103934. [CrossRef]
21. Xiang, C.; Wang, W.; Deng, L.; Shi, P.; Kong, X. Crack detection algorithm for concrete structures based on super-resolution reconstruction and segmentation network. *Autom. Constr.* **2022**, *140*, 104346. [CrossRef]
22. Koga, K.; Yasumura, N.; Shigeta, Y.; Shinjin, M.; Nakagawa, K. Examination of TCI for the quantitative integrity of tunnel lining. *Proc. Tunn. Eng. JSCE* **2004**, *13*, 371–376.
23. Shigeta, Y.; Tobita, T.; Kamemura, K.; Shinji, M.; Yoshitake, I.; Nakagawa, K. Propose of tunnel crack index (TCI) as an evaluation method for lining concrete. *Doboku Gakkai Ronbunshuu* **2006**, *62*, 628–632. [CrossRef]
24. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
25. Jiang, Y.; Zhang, X.; Taniguchi, T. Quantitative condition inspection and assessment of tunnel lining. *Autom. Constr.* **2019**, *102*, 258–269. [CrossRef]
26. Cha, Y.J.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
27. Arjovsky, M.; Shah, A.; Bengio, Y. Unitary evolution recurrent neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1120–1128.
28. Danihelka, I.; Wayne, G.; Uria, B.; Kalchbrenner, N.; Graves, A. Associative long short-term memory. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1986–1994.
29. Hirose, A.; Yoshida, S. Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 541–551. [CrossRef] [PubMed]
30. Trabelsi, O.; Bilaniuk, Y.; Zhang, D.; Serdyuk, S.; Subramanian, J.; Santos, S.F.; Mehri, N.; Rostamzadeh, Y.; Bengio, C.; Pal, J. Deep Complex Networks. In Proceedings of the ICLR, Vancouver, BC, Canada, 30 April–3 May 2018.
31. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
34. Hou, Q.; Zhang, L.; Cheng, M.M.; Feng, J. Strip pooling: Rethinking spatial pooling for scene parsing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4003–4012.
35. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
36. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
37. Rezaie, A.; Mauron, A.J.; Beyer, K. Sensitivity analysis of fractal dimensions of crack maps on concrete and masonry walls. *Autom. Constr.* **2020**, *117*, 103258. [CrossRef]
38. Li, L.; Sun, H.X.; Zhang, Y.; Yu, B. Surface cracking and fractal characteristics of bending fractured polypropylene fiber-reinforced geopolymer mortar. *Fractal Fract.* **2021**, *5*, 142. [CrossRef]
39. Wu, J.; Xie, D.; Yi, S.; Yin, S.; Hu, D.; Li, Y.; Wang, Y. Fractal Study of the Development Law of Mining Cracks. *Fractal Fract.* **2023**, *7*, 696. [CrossRef]
40. An, Q.; Chen, X.; Wang, H.; Yang, H.; Yang, Y.; Huang, W.; Wang, L. Segmentation of concrete cracks by using fractal dimension and UHK-net. *Fractal Fract.* **2022**, *6*, 95. [CrossRef]
41. Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z. Automatic road crack detection using random structured forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. [CrossRef]
42. 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. Transp. Syst.* **2019**, *21*, 1525–1535. [CrossRef]

43. Eisenbach, M.; Stricker, R.; Seichter, D.; Amende, K.; Debes, K.; Sesselmann, M.; Ebersbach, D.; Stoeckert, U.; Gross, H.M. How to get pavement distress detection ready for deep learning? A systematic approach. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2039–2047.
44. Yan, Y.; Zhu, S.; Ma, S.; Guo, Y.; Yu, Z. CycleADC-Net: A crack segmentation method based on multi-scale feature fusion. *Measurement* **2022**, *204*, 112107. [CrossRef]
45. Ali, R.; Chuah, J.H.; Talip, M.S.A.; Mokhtar, N.; Shoaib, M.A. Structural crack detection using deep convolutional neural networks. *Autom. Constr.* **2022**, *133*, 103989. [CrossRef]
46. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
47. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
48. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18. Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
49. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
50. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
51. Zhou, X.P.; Huang, X.C.; Zhao, X.F. Optimization of the Critical Slip Surface of Three-Dimensional Slope by Using an Improved Genetic Algorithm. *Int. J. Geomech.* **2020**, *20*, 04020120. [CrossRef]
52. TruSoft International Inc. *Benoit*™; 1.01; TruSoft International Inc.: St. Petersburg, FL, USA, 1997; Available online: https://www.trusoft-international.com/ (accessed on 15 September 2023).

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.