applied sciences

MDPI

Article

Highway Crack Detection and Classification Using UAV Remote Sensing Images Based on CrackNet and CrackClassification

Yingxiang Zhao, Lumei Zhou, Xiaoli Wang, Fan Wang and Gang Shi *

School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; zhaoyingxiang@stu.xju.edu.cn (Y.Z.); zhoulumei007@xju.edu.cn (L.Z.); wxl@xju.edu.cn (X.W.); xmilu0315@stu.xju.edu.cn (F.W.)
* Correspondence: shigang@xju.edu.cn

Abstract: Cracks are a common type of road distress. However, the traditional manual and vehicle-borne methods of detecting road cracks are inefficient, with a high rate of missed inspections. The development of unmanned aerial vehicles (UAVs) and deep learning has led to their use in crack detection and classification becoming an increasingly popular topic. In this paper, an aerial drone is used to efficiently and safely collect road data. However, this also brings many challenges. For example, flying too high or too fast may produce poor quality images, with unclear cracks that may be ignored or misjudged as other features and increased environmental noise that may make it difficult to distinguish between cracks and other noise features. To address the above challenges, this paper proposes the CrackNet model and CrackClassification algorithm. The CrackNet network is an encoder–decoder architecture. Low- and high-level semantic information are combined through the skip feature fusion layers between the encoder and decoder to enhance the model's expression and ability to recover image details. Additionally, the MHDC module at the bottom of the network can significantly increase the receptive field without reducing the feature map resolution. The MHSA module can simultaneously capture features from multiple subspaces. The average precision (AP) scores of the CrackNet network on three datasets, namely UAVRoadCrack, CRKWH100, and CrackLS315, were 0.665, 0.942, and 0.895, respectively. In addition, values of the other two evaluation metrics, ODS and OIS, were the highest among the compared methods. Meanwhile, the proposed CrackClassification algorithm in this paper achieves 85% classification accuracy for transverse and longitudinal cracks and 78% classification accuracy for block cracks and reticulated cracks. Overall, the CrackNet algorithm provides a new baseline model for crack detection in UAV remote sensing image scenes. The CrackClassification algorithm provides a new approach for batch classification of highway cracks. The detection and classification algorithm proposed in this paper were applied to 108 km of road sections.

Keywords: UAV remote sensing images; road crack detection; road crack classification; deep learning

check for updates

Citation: Zhao, Y.; Zhou, L.; Wang, X.; Wang, F.; Shi, G. Highway Crack Detection and Classification Using UAV Remote Sensing Images Based on CrackNet and CrackClassification. *Appl. Sci.* **2023**, *13*, 7269. https://doi.org/10.3390/app13127269

Academic Editor: Yosoon Choi

Received: 19 April 2023
Revised: 1 June 2023
Accepted: 13 June 2023
Published: 18 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Cracks on the surface of roads account for a significant proportion of road defects. Therefore, how to efficiently and safely detect and classify highway cracks is of great significance for maintaining highway safety. We conducted on-site visits and surveys of relevant personnel from the highway management department, and we learned that the current method of road inspection still mainly relies on manual detection. A team of four to five people working a full day can inspect approximately 3 km, and the inspection process also requires relevant road sections to be closed. This demonstrates that the manual inspection method is inefficient, laborious, and has a high rate of missed inspections and misclassification of cracks, which can result in serious consequences.

If cracks are not detected and repaired in a timely manner, they can worsen and lead to more severe damage, such as potholes or even complete pavement failure. This can result in safety hazards for drivers as well as increased maintenance costs for repairing the damage. In addition, misclassification of cracks can lead to incorrect maintenance decisions, such as allocating resources to repair non-existent cracks or neglecting actual cracks that require attention. This can result in wasted resources and increased risk of accidents. The methods to solve the problems of road crack detection and classification include timely detection and repair of cracks, improvement of existing mainstream crack detection networks, creation of algorithms suitable for batch classification of cracks, and establishment of efficient crack detection and classification systems, among others.

Over the past few decades, the automatic detection of cracks using ultrasonic detection has been proposed [1] to solve the problems associated with manual visual inspection classification. However, ultrasonic testing requires specialized technicians for operation and may be affected by the surface conditions of the object being tested. Moreover, ultrasonic testing equipment is expensive and requires calibration within a certain range to ensure accuracy. Later, crack detection methods based on images and image processing have gradually appeared [2,3]. However, image-based detection requires high-quality image data and processing of large amounts of data. Due to the complexity of the road surface, it is difficult to accurately detect and classify all types of cracks, especially small and shallow cracks, which may result in misidentification or missed detections. Then, with the development of artificial intelligence, various network models for crack detection appeared [4–6]. However, none of these networks can be applied to the detection of cracks in unmanned aerial vehicle (UAV) remote sensing images. Traditional machine learning algorithms, such as support vector machine and extreme learning machine, have also been used for crack classification [7]. The cracks are classified by calculating the horizontal projection, vertical projection, Euler number, and other metrics on a crack graph [8]. This method cannot classify cases where there are multiple cracks in the graph, and it cannot be applied to complex scenarios and batch processing.

To address the problem of low efficiency in manual and vehicle-borne inspection, this paper makes a bold attempt at data acquisition by using a professional mapping drone to collect road crack data through aerial photography at 80 m altitude. A single UAV sortie can shoot a 12 km road section, which takes about 30 min and is thus tens of times more efficient than the manual approach and has a higher safety factor. However, the flying altitude of the UAV is too high, which leads to a lot of environmental noise and unclear cracks in the captured image. Moreover, the shooting effect is affected by the lighting conditions. All of these issues make it challenging to detect cracks in the subsequent image analysis.

Considering the difficulties in detecting cracks in UAV remote sensing road images, we intend to construct a crack detection network based on two requirements. First, the network should be able to focus on the global features of cracks in the image and possess good noise immunity. Second, the network should be able to focus on more detailed information about cracks. To meet the first requirement, we design a multi-scale hybrid dilated convolution (MHDC) unit to be added to the bottom layer of the encoder to further increase the receptive field and obtain multi-scale contextual information. To meet the second requirement, feature fusion layers are added in each layer of the encoder, which can fuse the fusion results of all layers again at the end. We also design a multi-head self attention (MHSA) module to enhance the model's ability to acquire long-range structural information and obtain more detailed information about the cracks. In view of the above efforts, the CrackNet network can capture both the global features and the detailed information of cracks, achieving better crack detection performance for UAV remote sensing highway images.

To address the issue that traditional crack classification algorithms cannot classify complex cracks in UAV remote sensing road images and cannot exclude environmental noise, we propose a classification algorithm called CrackClassification. This algorithm is based on gridding, depth-first traversal, and minimum enclosing frame and is used to classify complex cracks in crack detection maps. In addition, a series of preprocessing

operations, such as identifying the roadbed and obtaining the coordinates of the roadbed, are used to exclude environmental noise.

The main contributions of our work can be summarized as follows.:

(1) In order to fill the gap of UAV remote sensing road crack dataset and facilitate subsequent research, we hand-labeled a UAV remote sensing road crack dataset, UAVRoadCrack.
(2) This paper proposes a novel network for UAV remote sensing road crack detection, CrackNet. The use of the MHDC module, MHSA module, GN+ELU, and skip feature fusion layer in the CrackNet network enables efficient and accurate detection of highway cracks, saving significant human and material resources, and facilitating subsequent data analysis. CrackNet achieved the best average precision (AP) scores of 0.665, 0.942, and 0.895 on the UAVRoadCrack, CRKWH100, and CrackLS315 datasets, respectively. Based on other evaluation metrics, it also outperforms other mainstream crack detection algorithms. These results demonstrate that the CrackNet network is not only applicable to UAV remotely sensed road images but can also be used in other scenes captured at close range.
(3) Our proposed preprocessing operation and CrackClassification algorithm can effectively exclude environmental noise and accurately classify complex cracks, which is essential for prioritizing maintenance and repair work. By calculating the confusion matrix of crack classification accuracy, we obtained precision rates of 0.839, 0.867, 0.740, and 0.778 for transverse cracks, longitudinal cracks, block cracks, and reticulated cracks, respectively. The recall for horizontal, vertical, block, and reticulated cracks is 0.868, 0.849, 0.738, and 0.772, respectively.

This paper is organized as follows: Section 2 presents the related work. Section 3 presents the proposed methodology in both highway crack detection and highway crack classification. Section 4 describes the details and results of each part of the performed experiments. Section 5 summarizes the paper and details the outlook for future research work.

2. Related Works

2.1. Highway Crack Detection

The approaches for crack detection can be broadly classified into two main categories of traditional image processing and machine learning algorithms and deep learning based on detection networks.

Oliveira et al. [9] used a sample-based learning-in-sample paradigm to select a subset of the existing image database for unsupervised learning, classifying image blocks into two categories with or without crack pixels. Zalama et al. [10] proposed a Gabor filter-based method to detect transverse cracks and longitudinal cracks. The data for their experiment was captured for a car equipped with an imaging system, an inertial profiler, a differential GPS, and a webcam. Zou et al. [11] proposed a crack detection scheme called CrackTree, in which geodesic shadows are first removed while preserving cracks, tensor voting is then used to build a crack probability map, and a set of crack seeds is finally drawn from the crack probability map to derive its minimum spanning tree, which is pruned at the edges of the tree to identify the desired cracks. Tang et al. [12] proposed a hybrid crack detection and segmentation algorithm, the idea of which is to first obtain the rough location of the crack using histogram thresholding and then use mathematical morphology techniques and snake modeling techniques to further refine the location. Avila et al. [13] proposed a method to find the minimum path from each path of length d through each crack pixel to detect cracks and proposed a corresponding dynamic programming implementation. Kapela et al. proposed a histogram of gradients (HoG) algorithm for crack detection, in which the idea is to perform local histogram computation by calculating the intensity and orientation of edges in each grayscale map. Shi et al. [14] proposed a random structured forest land-based road crack detection algorithm. Although the traditional image processing algorithm works better on datasets with high crack recognition and simple road conditions, it is difficult to

accurately and efficiently detect cracks in complex and diverse pavement environments with complex crack textures.

With the development of deep learning, many deep learning-based crack detection networks have emerged. Bang et al. [5] proposed an encoding–decoding network for crack detection, where a residual network is employed in the encoding part to extract features and migration learning is used to improve the network detection performance through data captured by a black box camera. Fan et al. [15] proposed a method to detect road cracks based on a deep convolutional neural network and adaptive thresholding method, where the deep convolutional neural network is used to determine whether the image contains cracks or not, and a bilateral filter is then used for smoothing. Nguyen et al. [16] proposed a two-stage convolutional neural network, where the first stage is used to denoise and isolate potential cracks to a region, and the second stage learns to detect the background of cracks in the region. Hac et al. [17] proposed the use of Fast R-CNN for crack detection, and Djenouri et al. [18] proposed a crack detection scheme using the scale invariant feature transformation (SIFT) algorithm to analyze the correlation between features to generate a series of graphs that are trained using a graph convolutional neural network and supervised using a super optimization algorithm. Jiang et al. [19] proposed an extended version of the U-Net framework, named MSK-UNet, for crack detection. They introduced selective kernel (SK) units to replace the standard convolution blocks in the U-shaped network to obtain receptive fields with different scales. Additionally, an image pyramid was established at the multi-scale input layer to preserve more image background information during the encoder stage. The datasets used in the above deep learning networks are mostly the data taken at close range. Although this study uses UAV remote sensing road data, which has fewer crack pixels and more environmental noise, it is difficult to reach the accuracy requirement of crack detection using the above neural networks.

2.2. Highway Crack Classification

Gavilan et al. [20] proposed a method using multiclass support vector machines to transform a single multiclass problem into multiple binary classification problems. Fernandes et al. [21] proposed a set of graphical features to effectively describe cracks that are highly expressive and robust in crack classification. Cubero-Fernandez et al. [8] used a decision tree heuristic algorithm for classification. Song et al. [22] used the minimum enclosing rectangular box to enclose each crack and classified the cracks according to the angle between the diagonal of the rectangular box and the horizontal direction in addition to the number of crack branches. Hoang et al. [23] proposed using the crack properties derived from the x-axis and y-axis projection integrals to determine the crack category. Li et al. [24] proposed a crack classification method using deep CNNs, training four CNNs with perceptual fields of different sizes. Li et al. [25] proposed an unsupervised crack classification algorithm fusing a convolutional neural network and a K-means clustering algorithm. Chen et al. [26] finally classified road cracks using support vector machines by extracting local binary pattern (LBP) features and reducing their dimensionality using principal component analysis.

The above crack classification methods are effective in cases where the dataset image has a single crack, a clear crack, and occupies a large proportion of the graph. However, when there are multiple cracks in the image and the environment is complex, the crack classification effect is poor, and the classification may not even be completed. In addition, the above crack classification scheme is slow and not suitable for batch processing of large amounts of data.

3. Proposed Methods

3.1. Highway Crack Detection Methods

In this section, we present CrackNet, a network model for highway crack detection, which is a pixel-level semantic segmentation model based on DeepCrack [27] with higher detection accuracy, higher noise immunity, and adaptability to more complex environments

for highway crack detection. The project on which this paper is based has used the CrackNet model to detect more than 100 km of road cracks using data from UAV remotely sensed road images. The structure of the CrackNet model, the MHDC module, the MHSA module, and the loss function are each described below.

3.1.1. The Structure of the Proposed CrackNet

As shown in Figure 1, the CrackNet network model is a typical encoding–decoding U-shaped architecture, divided into three parts: left–center–right. The encoder part is inspired by the VGG16 network [28]. However, only the first four layers of the VGG16 network are used, consisting of two convolutions with 64 channels of size 3×3 , two convolutions with 128 channels of size 3×3, three convolutions with 256 channels of size 3×3, and three convolutions with 512 channels of size 3×3. After each convolution operation, group normalization (GN) [29] and exponential linear unit (ELU) [30] activation functions are applied to the feature map. The reason for using GN + ELU (GE) instead of BN + ReLU (BR) is that the hardware environment in this experiment is limited, and the batch size set during training is small. BN increases rapidly when the batch size is small, and GN more effectively optimizes this problem. Additionally, each convolutional layer is paired with a max-pooling layer to reduce the input size of the next layer, computation, and the number of parameters. However, this also brings certain problems, such as reducing the spatial resolution of the feature map, which is not conducive to the segmentation of boundaries and lines. Therefore, the max-pooling index is used to capture and record the boundary information of the feature map in the encoding stage when downsampling in this network. The corresponding decoding layer uses the max-pooling index to perform nonlinear upsampling to avoid the lack of boundary details during the decoding process.

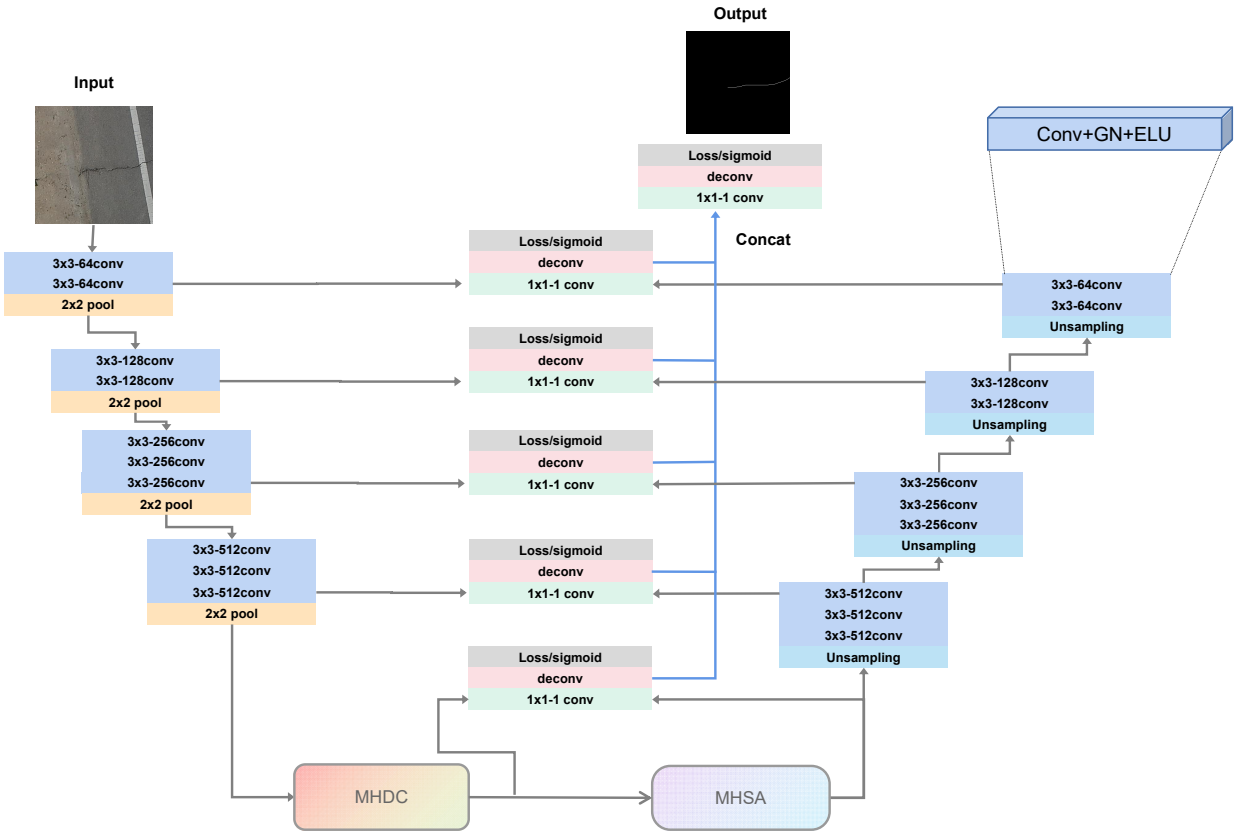


Figure 1. An illustration of the CrackNet network. The left side is the encoding part, the middle part is composed of the Skip-layer feature fusion layer, with detailed structure as in Figure 2, the MHDC module, with detailed structure as in Figure 3, and the MHSA module, with detailed structure as in Figure 4. The right side is the decoding part.

As shown in Figure 1, the middle part of the CrackNet network is composed of the Skip-layer feature fusion layer, MHDC module, and MHSA module. The MHDC module is composed of an dilated convolutional cascade conforming to the hybrid dilated convolution (HDC) design principle [31], in which the receptive field is increased without changing the feature map size and obtains richer contextual information. The details of the MHDC and MHSA modules are explained in Sections 3.1.2 and 3.1.3, respectively. Figure 2 illustrates the structure of the Skip-layer feature fusion layer, where the feature maps of the corresponding layers of the encoder and decoder are first stitched together, and the multiple channels are then mapped to a single channel using a 1×1 convolution with a channel number of 1. The feature maps are then processed to the size of the input image by deconvolution and crop layers. The feature maps are generated by the MHDC module, and the MHSA module is fused in the underlying Skip-layer. After these steps, the prediction maps of each layer have the same size as the ground truth fracture image. Finally, the prediction maps of each layer are stitched together, and all the prediction maps are fused by convolution with a channel number of 1×1 to obtain a multi-scale feature fusion map, i.e., a global prediction map.

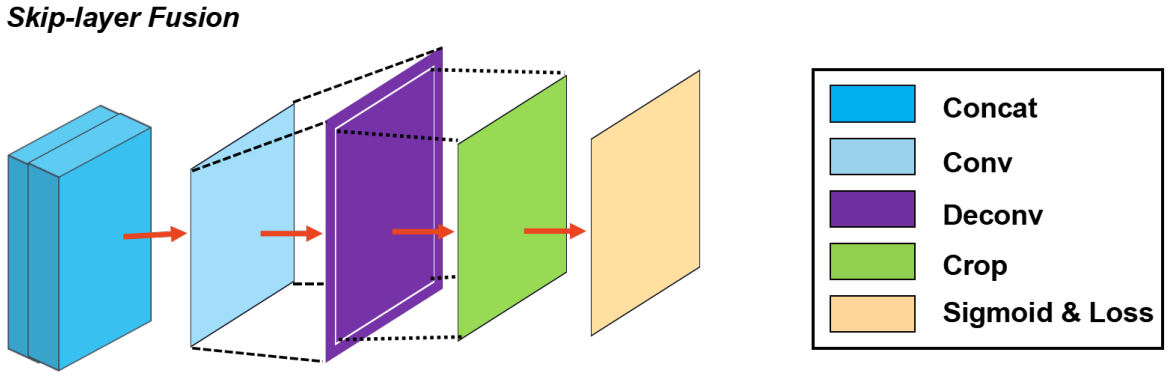


Figure 2. An illustration of the Skip-layer feature fusion layer.

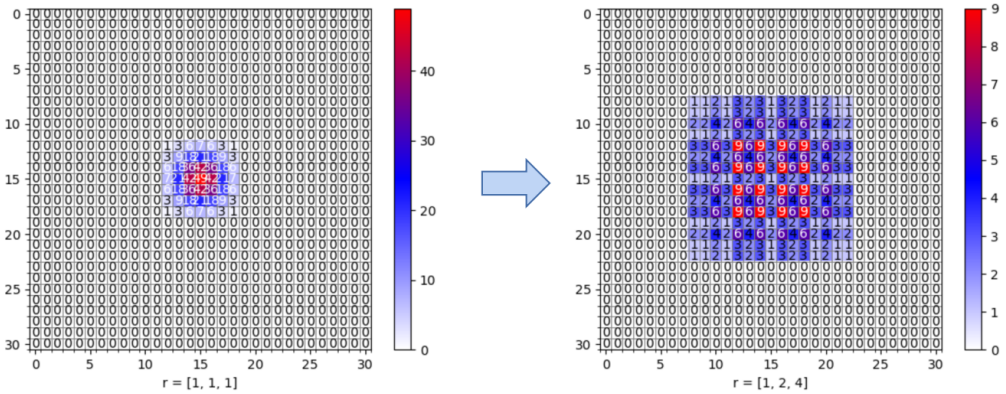


Figure 3. The change in the receptive field after using dilated convolution. Using 3 consecutive normal convolutions, the receptive field size can reach 7×7. Using three consecutive dilated convolutions with dilation rates of 1, 2, and 4, the receptive field size can reach 15×15.

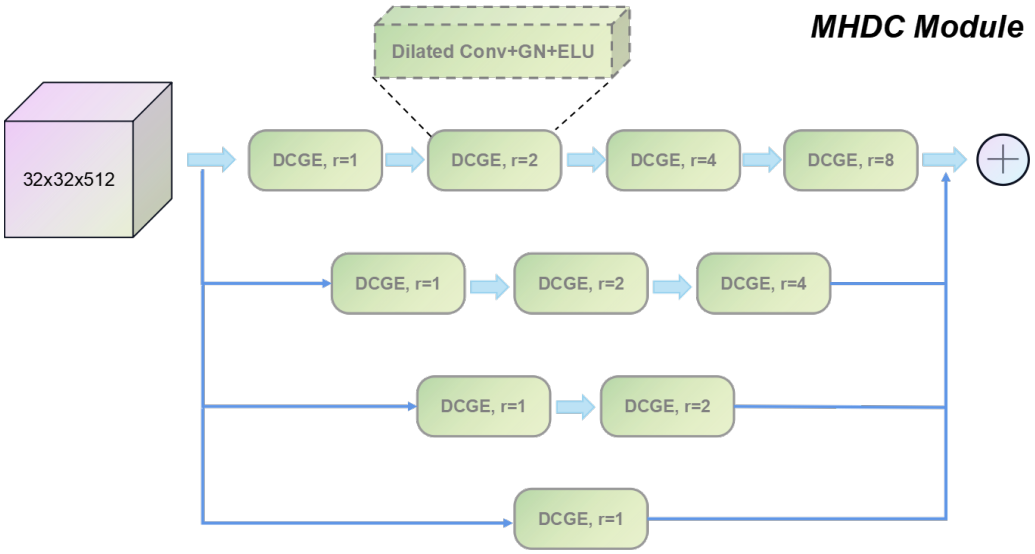


Figure 4. The MHDC module implementation details. DCGE is a combination of dilated convolution, GN, and ELU, and r represents the dilation rate of the dilated convolution.

The decoder part on the right side corresponds to the encoder part and is also divided into four layers. Each layer first goes through max-pooling upsampling and uses the max-pooling index to achieve nonlinear upsampling. After upsampling, it enters the convolution module of the layer, where the number of convolutions, the size of each convolution, and the number of channels are all consistent with the encoder convolution module of the corresponding layer. The prediction map from the last convolution of each layer is fed to the Skip-layer of the corresponding layer for fusion.

3.1.2. MHDC Module

Considering the complexity, connectivity, and narrowness of cracks, it is also important to increase the receptive field of feature points in the central part of the network while preserving detailed information. Dilated convolution is an ideal alternative to pooling in which the receptive field can expand and multi-scale contextual information captured without changing the feature map size.

Dilated convolution has an additional dilation rate parameter r to indicate the size of dilation compared with normal convolution. As shown in Figure 3, the receptive field size is 7 after successive stacks of three ordinary convolutions of size 3×3 and stride 1, while the receptive field can reach 15 after using three convolutions with dilation rates of 1, 2, and 4, respectively. Thus, it is easy to see that the dilated convolution can significantly expand the receptive field and cause the receptive field grow exponentially. However, when successively stacking the dilated convolutions, the dilation rate of each dilated convolution should be set according to the standardized design strategy, hybrid dilated convolution (HDC), to avoid the "gridding effect" problem. The HDC design strategy can be described as follows: (1) The dilation rate of each convolution of the stacked layers cannot have a convention greater than 1, such as $r = [2, 4, 6]$, there is a maximum convention of 2, and there will be a gridding effect. (2) Design the dilation rate as a sawtooth structure, such as $[1, 2, 3, 1, 2, 3]$. (3) Satisfy the following expression:

$$M_i = \max[M_{i+1} - 2r_i, M_{i+1} - 2(M_{i+1} - r_i), r_i] \quad (1)$$

where r_i is the dilation rate of layer i, and M_i means the maximum distance between two non-zero values in layer i. Then, assuming a total of n layers, the default $M_n = r_n$. The design goal is to make $M_2 <= k$. For example, for kernel size $k = 3$, an $r = [1, 2, 5]$ pattern works as $M_2 = 2$; however, an $r = [1, 2, 9]$ pattern does not work as $M_2 = 5$.

The usage modes of dilated convolution can be broadly classified into two, cascaded-mode-like [32] and parallel-mode-like [33], both of which have shown a powerful ability to improve segmentation accuracy. The MHDC module proposed in this paper then takes full advantage of both and combines the two modes by means of short connections.

As shown in Figure 4, if the expansion rate of the stacked dilated convolution is 1, 2, 4, 8, and 16, the receptive field of each layer is 3, 7, 15, 31, and 63, respectively. When the dilation rate is 1, the dilated convolution is the normal convolution. The encoder part of CrackNet network has four downsampling layers, and the image size for this experiment is 512 × 512, after four times of max-pooling. The uppermost HDC layer of the MHDC module designed in this paper uses dilated convolutional layers with dilation rates of 1, 2, 4, and 8, and its receptive field is 31 × 31, which covers most of the pixel points of the underlying feature map and can capture more contextual information. The other layers of HDC are 3, 7, and 15 in order of receptive field from bottom to top. By stitching the four layers of the multi-scale feature map together and reducing the dimensions through 1 × 1 convolution with 512 channels, multi-scale feature information is obtained, and deeper features are extracted without changing the size of the feature map.

3.1.3. MHSA Module

As shown in Figure 5, the MHSA module is divided into two parts, embedded patches and encoder block, and the encoder part in this experiment consists of $L = 6$ encoders with the same structure. For the image data, the 3D matrix in the format $[H, W, C]$ is not the input format required by the encoder block, so the input data format needs to be transformed by the embedding layer. Then, using linear projection, each patch is flattened into a vector called tokens, and the class token with the same dimension as the tokens is spliced before the tokens, and, finally, position embedding is added to complement the position information to obtain the input required by the final encoder block. The position encoding formula can be found in [34]:

$$\begin{aligned} PE_{(\text{pos},2i)} &= \sin\left(\text{pos}/10000^{2i/d_{\text{model}}}\right) \\ PE_{(\text{pos},2i+1)} &= \cos\left(\text{pos}/10000^{2i/d_{\text{model}}}\right) \end{aligned} \tag{2}$$

where pos is the position, i is a certain dimension at that position, and d_{model} is the dimension of the embedding vector, where the number must be a multiple of the number of heads. The even dimension of a position uses a sine function, and the odd dimension uses a cosine function.

From Figure 5, we can see that the tokens enter the encoder block and then pass through the multi-head self-attention (MSA) block and the MLP block in turn. The MSA block can capture different types of information from different perspectives. Specifically, the input feature map is divided into multiple subspaces, and each subspace is processed by a separate attention head. The outputs of all attention heads are concatenated and linearly transformed to obtain the final output. The MLP block is composed of two fully connected layers, GELU activation function and dropout. After the first fully connected layer, the dimensionality increases by a factor of four and the number of nodes becomes four times the original number, but after the second fully connected layer, the number of nodes returns to the original point. Dropout randomly sets a fraction of the input units to zero during training, which forces the network to learn more robust features. The attention formula is as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

where Q is the query matrix used to calculate the attention weights, K is the key matrix used to encode the structural information, V is the value matrix used to weight the importance of each feature, and d_k is the dimension of the matrix K.

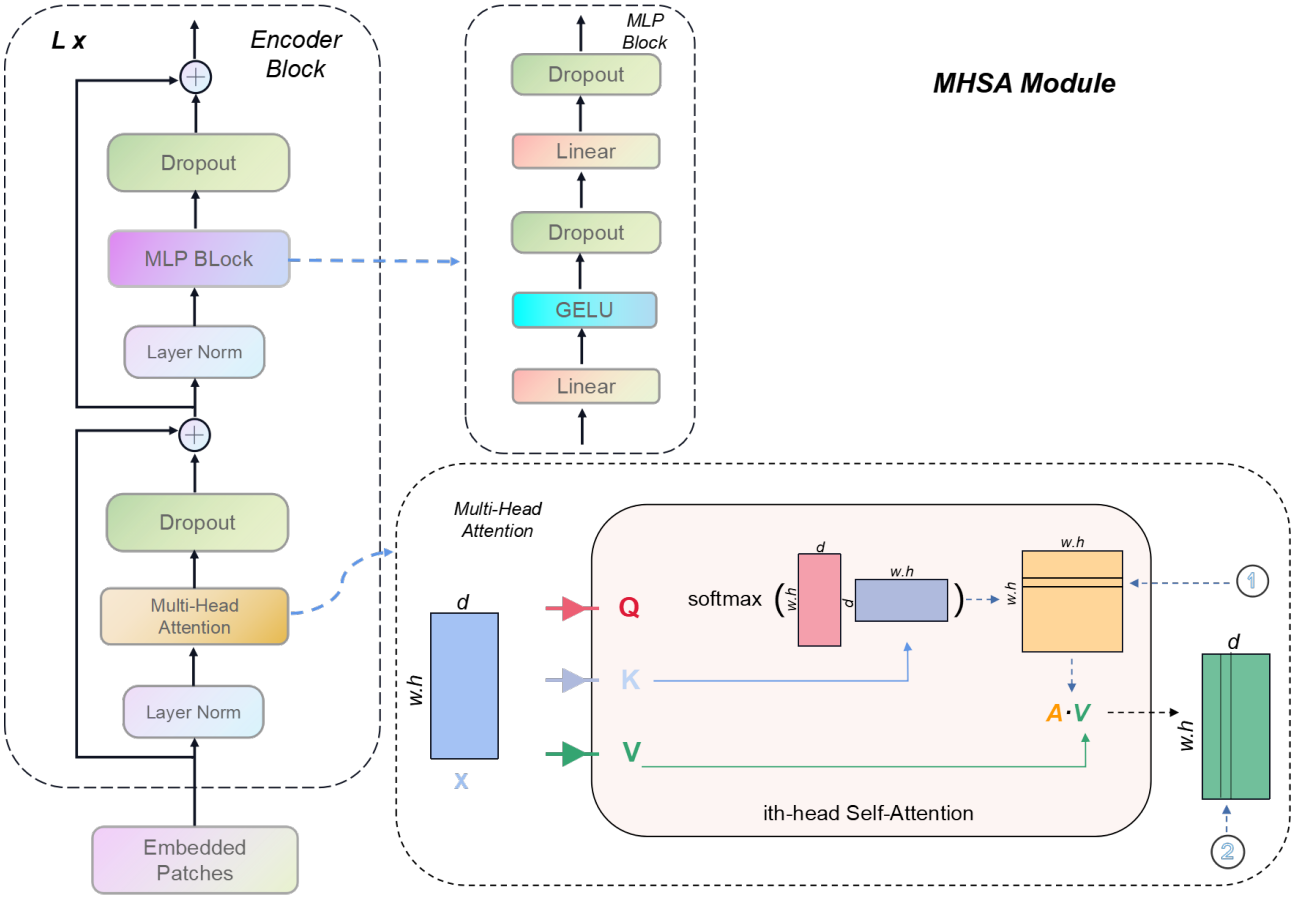


Figure 5. An illustration of the MHSA module. The left side is the backbone of the module, which consists of the embedding layer and the L-layer encoder block. In this paper, L is taken as 6. The structure of the MLP block is shown on the upper right. The lower right shows the structure of the $i-$ th head self-attention in multi-head attention, which finally splices the results of all heads. (**1**) A line in A corresponds to the attention of an element in Q to all elements in K. (**2**) A column in V corresponds to the column of the feature map obtained by weighting the attention of A.

A self-attentive module has three inputs, which are a query matrix Q, a key matrix K, and a value matrix V. As in Figure 5, the product of the transpose of matrix Q and matrix K is first computed to calculate the dot product of each query value and key value, and the larger the dot product result, the stronger the correlation between the two. Matrix A is obtained by dividing by$\sqrt{d_k}$ and using the softmax activation function for the quotient. $\sqrt{d_k}$ is divided because if the dot product is large, the gradient becomes small after softmax. The matrix A is multiplied by the matrix V to obtain the final output of the attention.

Multi-head self-attention is implemented based on self-attention, which divides the model into multiple heads to form multiple subspaces, allowing the model to focus on different aspects of information and finally splicing the results of multiple heads simply by satisfying the following equation.

$$\begin{aligned} \text{MultiHead}(Q,K,V) &= \text{Concat}(\text{head}_1,\ldots,\text{head}_\text{h})W^O \\ \text{where head} &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \end{aligned} \tag{4}$$

where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{mode}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$, h is the number of heads, and $d_k = d_{model}/h$.

The hyperparameters for the MHSA module are chosen based on empirical evaluation. The block size is set to 1, grid size is set to 12, hidden layer size is set to 12, encoder block has 6 layers, the number of attention heads is 12, the dropout rate for attention is set to 0.1, and the dimension of the multilayer perceptron (mlp) is set to 3072.

3.1.4. Loss

The dataset used in this study is a UAV remote sensing highway crack dataset, which presents challenges due to unclear cracks, with a low percentage of crack pixels in the dataset samples caused by the high flying height of the UAV, environmental noise, and complex crack patterns. To address these issues, a combination of binary cross-entropy loss and focal loss was chosen as the loss function for the CrackNet model. Binary cross-entropy loss is commonly used for binary classification problems, while focal loss is designed to address the issue of class imbalance in the dataset. The combination of these two loss functions helps to improve the model's ability to distinguish between positive and negative samples and to focus on hard-to-classify samples.

The dataset contains N images, denoted as $S = \{(X^n, Y^n), n = 1, \ldots, N\}$, where X^n denotes the original input image, Y^n denotes the ground truth crack label map corresponding to X^n, and I denotes the number of pixels in each image. The purpose of our training network is to generate prediction maps that are close to the ground truth. In this encoder–decoder architecture, let K be the number of convolution levels; then, at level k, the feature map generated by the jump layer can be formulated as $F^{fuse} = \left\{ f_i^{(k)}, i = 1, \ldots, I \right\}$, where $k = 1, \ldots, K$. In addition, the multi-scale feature fusion layer can be defined as $F^{fuse} = \left\{ f_i^{(fuse)}, i = 1, \ldots, I \right\}$.

There are only two categories in our crack detection, one for cracks and one for background, so it can be considered as a binary classification problem. We use cross-entropy loss to measure the prediction error. Usually, in a crack image, ground truth crack pixels belong to a few categories, which leads to detecting cracks as an unbalanced classification or segmentation. Some previous works addressed this problem by giving more weight to a few categories. However, we find that giving cracks a larger weight in crack detection generates more false positive samples, so we define the pixel-level prediction loss as:

$$l(F_i; W) = \begin{cases} log(1 - P(F_i; W)), & \text{if} \quad y_i = 0, \\ log(P(F_i; W)), & \text{otherwise}, \end{cases} \tag{5}$$

where F_i is the output feature map of pixel i after passing through the network model, W is the standard set of parameters in the network layer, and $P(F)$ is a standard sigmod activation function that converts the feature map into a crack probability map. The final total loss can be formulated as:

$$\mathcal{L}(W) = \sum_{i=1}^{I} \left(\sum_{k=1}^{K} l\left(F_i^{(k)}; W\right) + l\left(F_i^{fuse}; W\right) \right) \tag{6}$$

3.2. Highway Crack Classification Methods

In this section, we present a road crack classification scheme for UAV remote sensing road images. The scheme includes the preprocessing of crack detection result map and the proposed CrackClassification algorithm for highway crack classification.

3.2.1. Environmental Noise Removal Based on Roadbed Identification

As shown in Figure 6, in the UAV remote sensing highway image, the highway only occupies a part of the remotely sensed image, and the same is true in the highway crack detection result map. Due to the presence of errors in the manually labeled dataset, the model may be trained with the environmental noise as cracks during the training process.

Therefore, in addition to the road surface, some noise from the surrounding environment will also be detected as cracks in the detection result map. To avoid misclassifying the detected environmental noise as cracks in the classification process, it is necessary to first identify the roadbed and obtain the pixel coordinates of the roadbed. Only the pixel points within the pixel coordinates of the roadbed can be used for crack classification in the crack classification process. In this way, the impact of ambient noise can be reduced.



Figure 6. The sample images of the UAV remote sensing highway images. Roads make up a smaller portion of UAV remote sensing images. Highway cracks represent a much smaller proportion in the map.

In this paper, the DenxiDeepCrack [35] network model is used for roadbed recognition. Due to the lack of a roadbed dataset, this experiment manually labeled the roadbeds of UAV remote sensing road images with LabelMe. A series of data processing operations was then performed, resulting in a UAV remote sensing roadbed dataset containing 7025 images in the training set, 2356 images in the validation set and 2356 images in the test set. The DenxiDeepCrack network model was then trained using this dataset and used for roadbed detection.

The yellow line in the center of the road may be identified as the roadbed in the detection result, so it must be removed to obtain the coordinates of the roadbed. Given that the central yellow line identification effect is better, the roadbed identification map matrix is subtracted from the central yellow line identification map matrix to achieve the effect of removing the yellow line.

Then, the roadbed detection result map is binarized to find the roadbed pixel coordinates according to the pixel value, and the pixel coordinates of the roadbed are recorded.

Based on the roadbed pixel coordinates, the environmental noise in the full-size drone aerial image will be removed such that only the highway road surface is retained. Additionally, in this section, the crack detection algorithm proposed in Section 3.1 will be applied to detect cracks on the drone aerial image of the road surface for reducing the GPU workload. The resulting crack detection image will be used for subsequent crack classification.

3.2.2. CrackClassification Algorithm

According to the road technical condition standard, road cracks can be divided into four categories: transverse cracks, longitudinal cracks, block cracks, and reticulated cracks. The morphology of each crack is shown in Figure 7. The direction of transverse cracks is generally perpendicular to the center line of the road, the direction of longitudinal cracks is basically parallel to the direction of the route, the shape of block cracks is nearly rectangular in appearance, and the reticulated cracks are interlaced horizontally and vertically, resembling the lines of a tortoise shell.

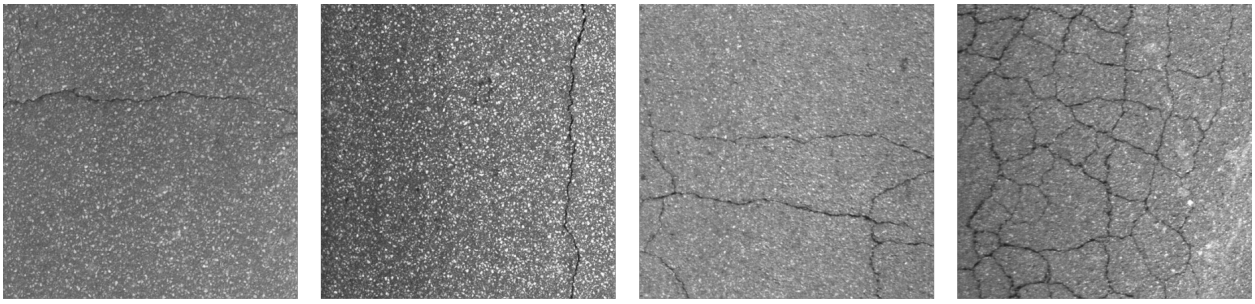


Figure 7. Diagram showing four kinds of cracks: (**a**) transverse cracks, (**b**) longitudinal cracks, (**c**) block cracks, and (**d**) reticulated cracks.

Different types of highway cracks have different causes, and the measures for their targeted repair also differ. Only through effective classification of highway cracks can we develop maintenance plans for different types of road defects, reasonably allocate maintenance resources, reduce maintenance costs, improve maintenance effectiveness and permanence, and enhance maintenance benefits.

The previous road crack classification algorithms cannot be used in UAV remote sensing road images, and the classification efficiency is low, making them unsuitable for batch processing of large-scale data. In contrast, the CrackClassification algorithm proposed in this paper can effectively solve these two problems. It was tested in the project and proved to be scientifically rational.

The overall idea of the algorithm is shown in the flow chart of the algorithm in Figure 8. First, the area within the pixel coordinates of the roadbed is divided into equal-sized grids, as shown in Figure 9a, and the number of white pixel points in each grid is calculated. The threshold α for the number of white pixels judged as block cracks and the threshold β for the number of white pixels judged as mesh cracks are set based on the comparison of statistical principles and manual labeling classification results according to the road technical condition assessment criteria. Second, each grid is classified as a reticulated crack when the number of white pixels in the grid is greater than or equal to β. The area of the reticulated crack is recorded and all pixels in the grid are set to 0. Similarly, when the number of white pixels in the grid is greater than or equal to α and less than β, the grid is determined as a block crack, the area of the block crack is recorded and all pixels in the grid are set to 0. The area of the block crack is recorded, and all pixels in the grid are set to 0. Third, the minimum enclosing frame algorithm based on depth-first traversal is applied to the image with block cracks and reticulated cracks removed, and the horizontal cracks and vertical cracks are enclosed in a rectangular frame, and the width-to-height ratio of the rectangular frame is calculated. Cracks are recorded as transverse when the aspect ratio is greater than or equal to 1 and longitudinal when the aspect ratio is less than 1, and their lengths are recorded. The visualization of the anchor frame is shown in Figure 9b. Finally, the information of crack category, length, area, and location is written into the database.

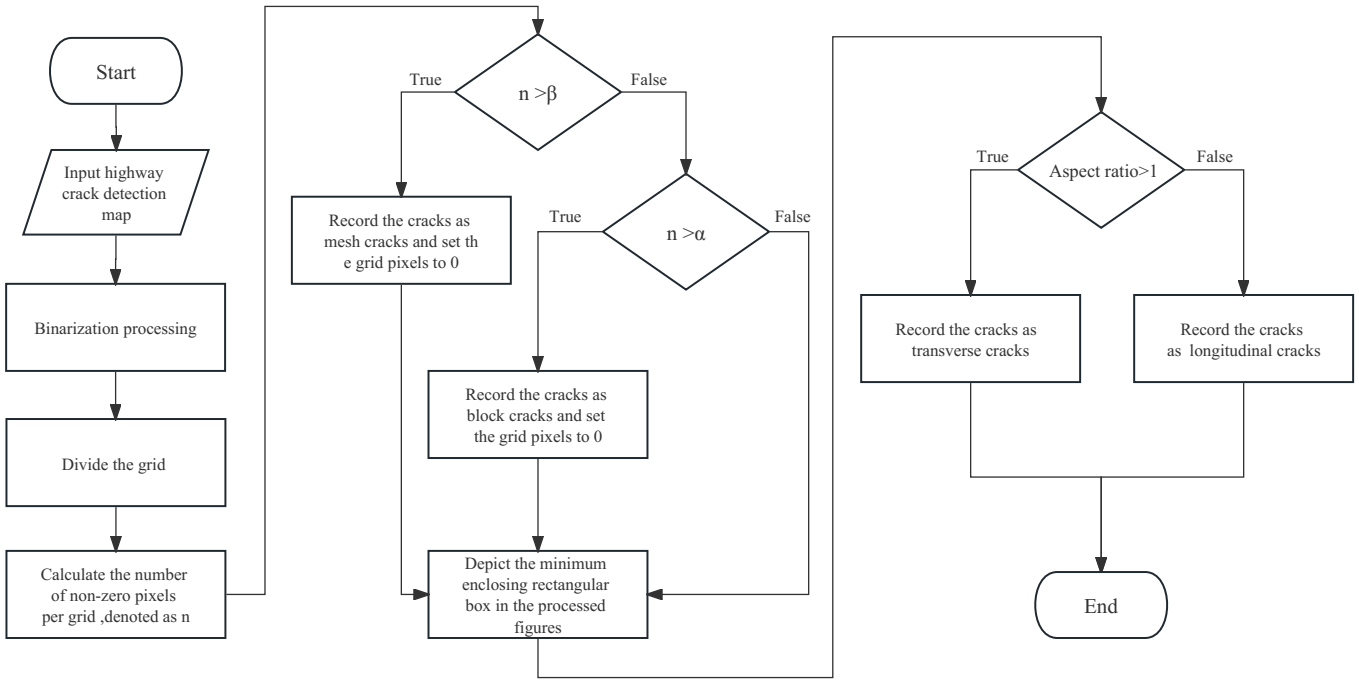


Figure 8. The CrackClassification algorithm flow chart.

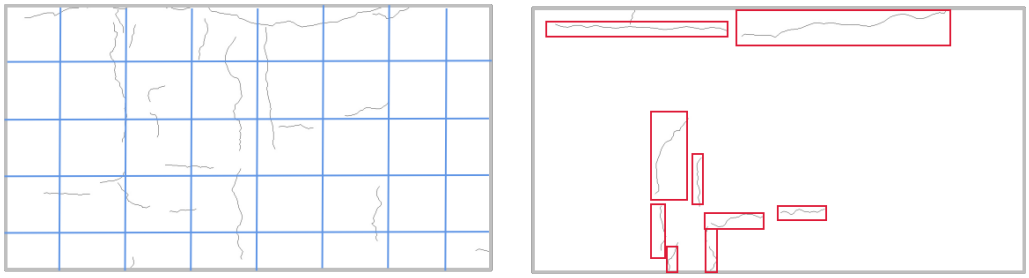


Figure 9. (**a**) The area within the pixel coordinates of the roadbed is divided into equal-sized grids. (**b**) Framing of transverse and longitudinal cracks with rectangular anchor frames.

4. Experiments

In this section, we will present experiments conducted in both highway crack detection and highway crack classification. For highway crack detection, to validate the crack detection performance of the CrackNet network on UAV remote sensing highway images, we conducted comparative experiments with other mainstream crack detection algorithms and ablation experiments on the improvements made to the original model. For highway crack classification, a series of data preprocessing operations were first conducted, including roadbed identification, central yellow line removal, roadbed coordinate acquisition, and environmental noise removal from full-size images. Then, crack classification was performed on the UAV remote sensing highway crack detection results from three roads, totaling 108 km. Finally, the results were compared with manual road inspection results to demonstrate the scientific soundness and feasibility of the method.

For all the experiments, training and testing was conducted on a server with an Intel(R) Xeon(R) CPU E5-2680 v3 processor, 128 RAM, and a single NVIDIA A40 GPU with 48 GB video memory. The software environment for the experiments in this paper was 18.04.1-Ubuntu OS, CUDA 11.3, PyTorch 1.12.1, and Python 3.9.

4.1. Highway Crack Detection Experiments

4.1.1. UAVRoadCrack Dataset

The original image data used in this experiment were all captured by a DJI Meridian M300 RTK UAV equipped with a 45-megapixel Zenith P1 camera as shown in Figure 10a. The UAV flight height is 60 m, flight speed is 18 m/s, and the resolution of the UAV images is 8192 × 5460. The horizontal and vertical resolutions of the image are both 72 dpi, the ground sampling distance (GSD) is 0.75 cm/pixel, the bit depth is 24, and the color representation is sRGB. The heading overlap of the flight line is 80%, and the sidelap is 70%. The left and right extension distance of the strip flight is 30 m. The camera's focal length for shooting is 35 mm, and the exposure time is 1/1000 s. The road remote sensing images used in the dataset include images from various sorties, various road sections, different road characteristics, different flight times, and different lighting conditions. Figure 6 shows some of the UAV aerial remote sensing images of roads.



(a) (b)

Figure 10. (**a**) The data acquisition equipment, a DJI M300 RTK drone. (**b**) The labeling results using LabelMe.

The dataset is labeled with LabelMe's line strip, and the labeling results are shown in Figure 10b, where the red curves represents cracks. Since the annotation generated with LabelMe is a Json file, it needs to be converted to a mask image. In addition, because its size is too large, the image needs to be cropped and split into images of 384 × 384 size. Then, the dataset is divided into a training set, validation set, and test set according to the ratio 6:2:2. Figure 11 shows some of the original images in the dataset with their corresponding label maps. The structural composition of the final generated UAV remote sensing road crack dataset is shown in Table 1.

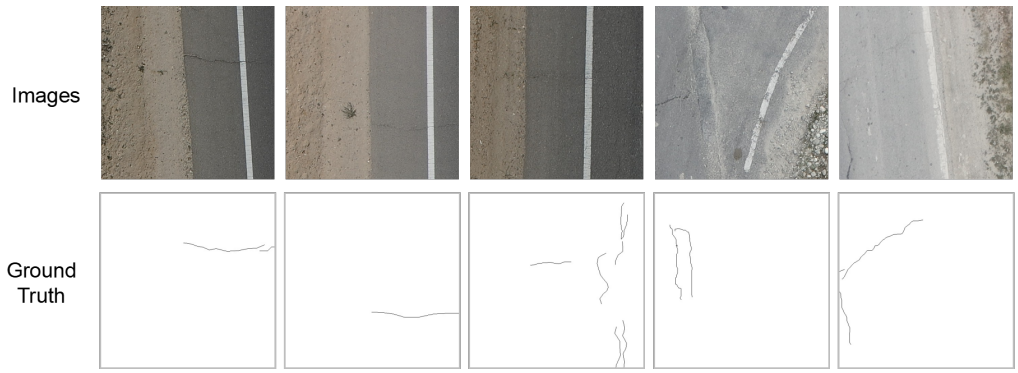


Figure 11. A part of the UAVRoadCrack dataset.

Table 1. UAVRoadCrack dataset.

Dataset	Original Images	Ground Truth
Training set	4443	4443
Validation set	1480	1480
Test set	1480	1480
Total	7403	7403

4.1.2. Evaluation Metrics and Experimental Parameter Settings

For highway crack detection, each image can be evaluated by comparing the crack detection map with the manually labeled ground-truth map to calculate the precision and recall rates. In addition, the F-Score can be used as an overall metric for performance evaluation. The F-Score is calculated as follows:

$$\text{F-Score} = \left(1+\beta^2\right) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{7}$$

where Precision is the proportion of true positive samples among the predicted positive samples, which may contain negative samples. Recall is the proportion of true positive samples without any negative samples that are correctly predicted. β is the weight to balance Precision and Recall in the F-score calculation. Precision and recall are calculated using the following formulas:

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

$$Recall = \frac{TP}{TP+FN} \tag{9}$$

where TP denotes true positives (pixels classified as cracks and correctly identified), TN denotes true negatives (pixels classified as non-cracks and correctly identified), FP denotes false positives (pixels classified as cracks but actually not), and FN denotes false negatives (pixels classified as non-cracks but actually are).

For subsequent evaluation of the results, probabilistic binarization of the crack detection result map is required to obtain a binary map. Therefore, a threshold value η, which can also be called the confidence level, is set to 0 for positive cases and 1 for negative cases, if it is greater than η.

In this experiment, three different metrics based on F-Scores are used to evaluate the experimental results: optimal dataset scale (ODS), optimal image scale (OIS), and average precision (AP) where ODS is also known as global optimal, fixed contour threshold, and optimal on detection metric dataset scale. That is, the same threshold is set for all images such that the F-Score of the whole dataset is the maximum. OIS is also known as single-image best, best threshold on each image, and optimal on image scale, i.e., a different threshold, η, is selected on each image that maximizes the F-Score of that image. AP, average accuracy, is the integral of the PR curve. Since it is difficult to integrate over the PR curve, AP is found by sampling the mean value over the PR curve.

After preliminary experiments, the final detailed settings of each hyperparameter are listed in the following Table 2. In addition, this experiment uses AdamW Optimizer to dynamically adjust the learning rate of each parameter and update the network weights. To minimize unnecessary computational resource loss, an early stopping strategy is used during the training process to stop the training when the loss on the test set is no longer decreasing.

Table 2. Hyperparameter settings.

Image Size	Batch Size	Epoch	Learning Rate	Learning Rate Decay	Weight Decay	Dropout Rate	MHSA Layer Num
384 × 384	8	100	1×10^{-3}	0.1	0.00001	0.1–0.5	6

4.1.3. Ablation Experiments

This subsection focuses on the impact of each proposed improvement to the original network and the effect of each added module on the effectiveness of the experiments on the UAVRoadCrack dataset.

The ablation experiment uses the DeepCrack network model as the baseline network to explore the effects of GE, MHDC, and MHSA on the baseline model. The "+" in Table 3 represents an improvement point of the model. The ODS, OIS, and AP of the baseline model on the UAV remote sensing highway crack dataset are 0.601, 0.613, and 0.608, respectively. When the MHDC module is added to the baseline model, the model further increases the receptive field in the underlying feature extraction process while obtaining multi-scale feature information, resulting in improvements in all indexes compared to the baseline. In the extreme model with the MHDC module, the BR in the original convolutional layer is replaced with GE, allowing the model to maintain a small training error when the batch size is small, and the ODS, OIS, and AP are 0.008, 0.009, and 0.009 higher, respectively, than those with the MHDC module only. When the CrackNet network with three improvements to the baseline model was used, the evaluation indexes of the network model on the UAV remote sensing highway crack dataset were improved by 10% compared with the baseline model, further demonstrating the reasonableness of the three improvements to the baseline model. The CrackNet model performs better on the UAVRoadCrack dataset than without the pretrained weights when using the training weight files on the public dataset CrackL315.

Table 3. The results of the ablation experiments. **MHDC**: multi-scale hybrid dilated convolution. **GE**: GN + ELU. **MHSA**: multi-head self-attention. **L315**: model pretrained on CrackL315.

Model	ODS	OIS	AP
Baseline	0.601	0.613	0.608
+MHDC	0.623	0.626	0.624
+MHDC + GE	0.631	0.635	0.633
+MHSA	0.628	0.631	0.627
+MHDC + MHSA	0.652	0.663	0.651
+MHDC + MHSA + GE	0.661	0.672	0.665
+MHDC + MHSA + GE + L315	0.673	0.688	0.682

4.1.4. Comparison Experiments

The comparison experiments not only compare the CrackNet network model with other mainstream image segmentation models based on the homemade UAV remote sensing highway crack dataset but also on the public datasets CRKWH100 and CrackLS315 to increase the credibility of the experiments.

As shown in Table 4, CrackNet has the best metric performance on both the UAVRoadCrack dataset and the CRKWH100 and CrackLS315 datasets and shows a more significant improvement compared to all other networks.

Table 4. The results of the comparison experiments.

	UAVRoadCrack			CRKWH100			CrackLS315		
Model	**ODS**	**OIS**	**AP**	**ODS**	**OSI**	**AP**	**ODS**	**OIS**	**AP**
DeepCrack [27]	0.601	0.613	0.608	0.909	0.917	0.931	0.845	0.867	0.877
ResNet [36]	0.442	0.505	0.402	0.844	0.853	0.869	0.804	0.813	0.826
FCN [37]	0.341	0.347	0.364	0.789	0.827	0.774	0.700	0.713	0.692
SegNet [38]	0.449	0.480	0.395	0.818	0.852	0.849	0.761	0.795	0.779
UNet [39]	0.352	0.413	0.355	0.846	0.854	0.902	0.672	0.702	0.740
CrackNet	0.661	0.672	0.665	0.933	0.936	0.942	0.863	0.882	0.895

On the UAVRoadCrack dataset, CrackNet network engages 0.06, 0.059, and 0.057 over the ODS, OIS, and AP, respectively, of the baseline model, all of which confirm that the CrackNet network is more suitable for road crack detection on UAV aerial road remote sensing images than other segmentation networks. The FCN network has poor detection performance. The ResNet and SegNet networks have similar results for each evaluation metric.

On the CRWH100 public dataset, the ODS, OSI, and AP of CrackNet reached 0.933, 0.936, and 0.942, respectively, showing varying degrees of improvement over other mainstream networks.

On the CrackLS315 public dataset, the three metrics of the DeepCrack network training results were 0.845, 0.867, and 0.877, while the CrackNet network still outperformed each of them by an average of 0.017. These experimental results show that the CrackNet network is also suitable for crack detection in other scenarios and has good generalization ability.

Figure 12 shows the detection results of different algorithms on the same UAV remotely sensed highway crack image. From the comparison of the detection results, it can be observed that CrackNet can detect more crack details and is closer to the ground truth than the other networks. Additionally, less environmental noise is also detected around the highway, indicating that the network also has better noise immunity.

The confusion matrix for the accuracy of CrackNet on the UAVRoadCrack dataset is shown in Table 5. For real cracks, 68.3% were detected as cracks and 31.7% were detected as non-cracks; for real non-cracks, 66.6% were detected as non-cracks and 33.4% were detected as cracks.

Table 5. The confusion matrix for the accuracy of CrackNet on the UAVRoadCrack dataset.

	Predicted Results	
Real Results	**Cracks**	**Non-Cracks**
Cracks	68.3%	31.7%
Non-Cracks	33.4%	66.6%

The confusion matrix for the accuracy of CrackNet on the CRWH100 dataset is shown in Table 6. For real cracks, 93.8% were detected as cracks and 6.2% were detected as non-cracks; for real non-cracks, 94.1% were detected as non-cracks and 5.9% were detected as cracks.

Table 6. The confusion matrix for the accuracy of CrackNet on the CRWH100 dataset.

	Predicted Results	
Real Results	**Cracks**	**Non-Cracks**
Cracks	93.8%	6.2%
Non-Cracks	5.9%	94.1%

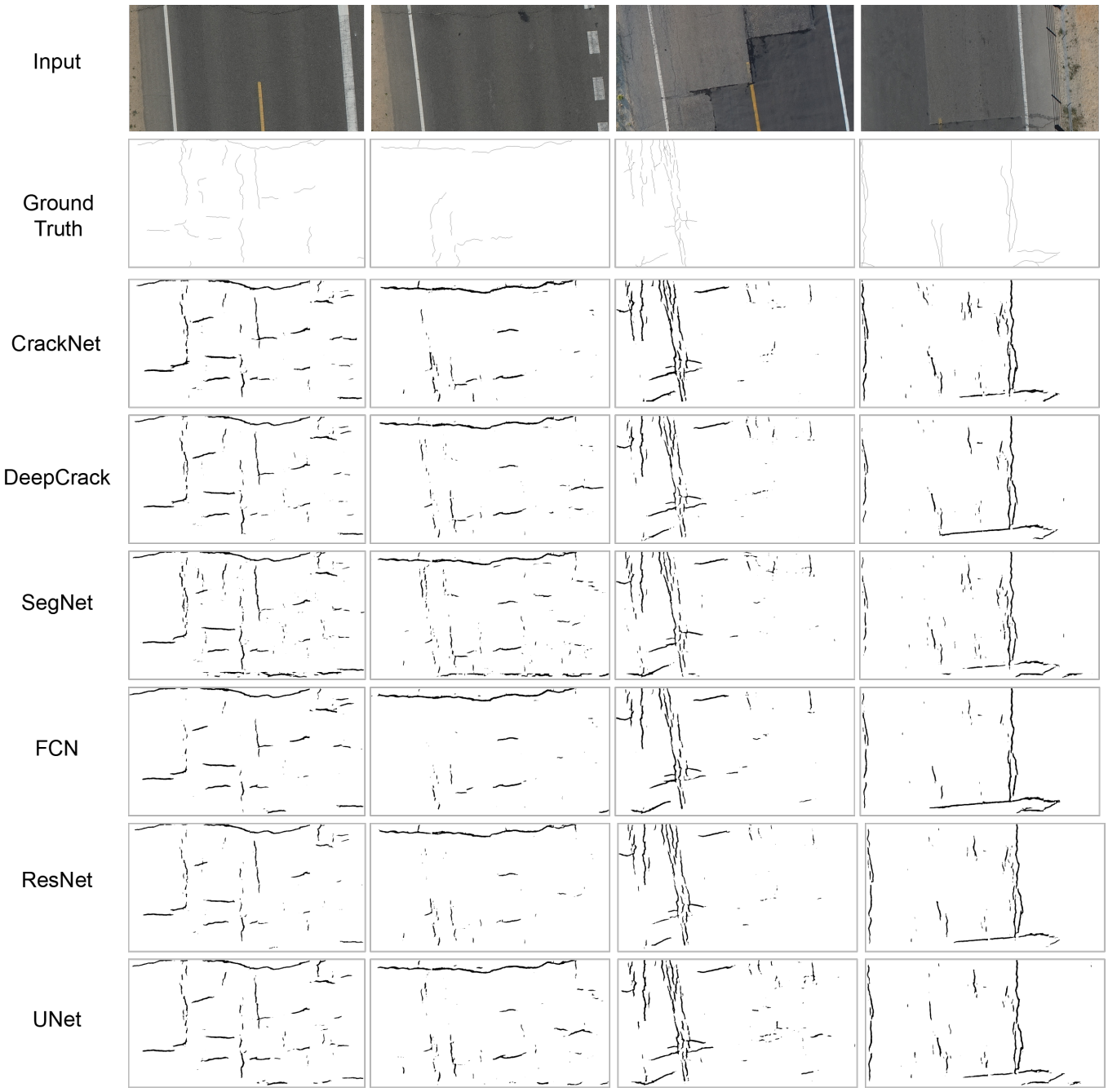


Figure 12. The comparison experiments on the UAVRoadCrack dataset.

The confusion matrix for the accuracy of CrackNet on the CrackLS315 dataset is shown in Table 7. For real cracks, 87.3% were detected as cracks and 12.7% were detected as non-cracks; for real non-cracks, 79.6% were detected as non-cracks and 20.4% were detected as cracks.

Table 7. The confusion matrix for the accuracy of CrackNet on the CrackLS315 dataset.

	Predicted Results	
Real Results	**Cracks**	**Non-Cracks**
Cracks	87.3%	12.7%
Non-Cracks	20.4%	79.6%

Based on comprehensive observation of the three confusion matrices, it is clear that the CrackNet network has a higher accuracy in identifying crack and non-crack pixels in two public datasets representing close-range shooting scenes. Additionally, in the UAVRoadCrack dataset representing high-altitude aerial shooting scenes by unmanned aerial vehicles, the lower clarity of crack images results in a lower accuracy in identifying crack and non-crack pixels compared to the public datasets. Nevertheless, the accuracy still falls within the error range required by the highway management department.

4.2. Highway Crack Classification Experiments

4.2.1. Failure Experiences of Highway Crack Classification Experiments

Crack classification is a challenging task due to the complex nature of crack patterns and the presence of noise in images. Using traditional crack classification methods, such as calculating the vertical projection, horizontal projection, Euler number, and other metrics of the split map to classify the cracks, classification cannot be achieved in cases where there are multiple cracks in the split map, as is shown in Figure 13. Additionally, these methods cannot adapt to complex scenes and are not suitable for batch processing. Therefore, it is a challenge to find a classification algorithm that is applicable to the UAV remote sensing highway crack image scenario.

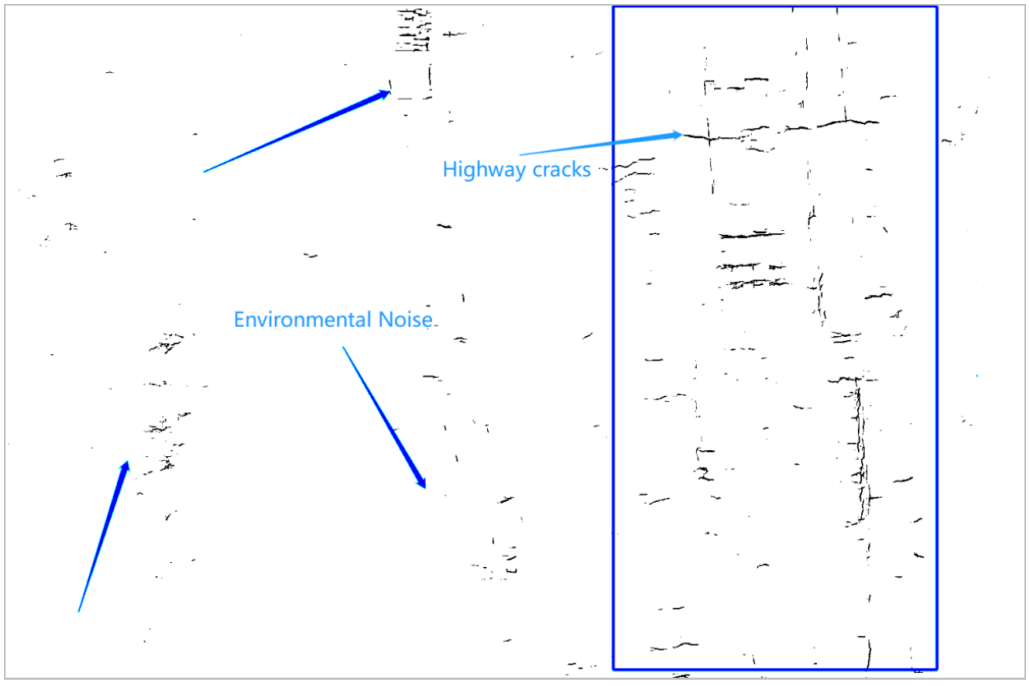


Figure 13. UAV remote sensing road crack detection map.

Classification on road crack detection result maps using deep learning networks such as YOLOv5 yields poor results, as shown in Figure 14. The road crack detection result map is a grayscale map consisting of black and white pixels. There are few crack features that can be obtained from the graph, and the highway pavement cracks are complex. Therefore, multi-category semantic segmentation and instance segmentation are very ineffective in application.

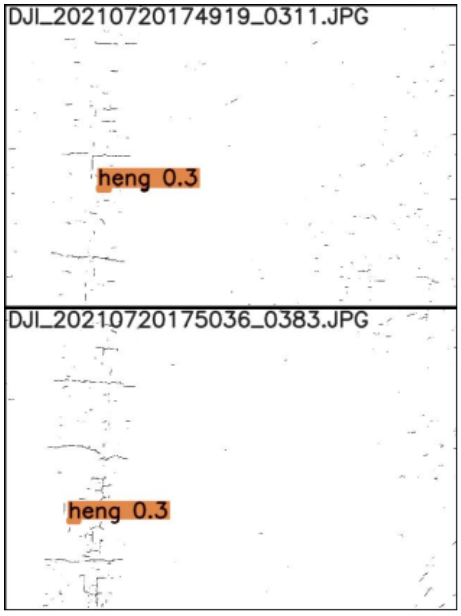


Figure 14. The classification effect of YOLOv5 on a UAV remote sensing highway crack detection map.

4.2.2. Environmental Noise Removal Based on Roadbed Identification

In order to avoid classifying the detected environmental noise as cracks in the classification process, we can first identify the roadbed and obtain its pixel coordinates. Based on the obtained pixel coordinates of the roadbed, the environmental noise in the full-size UAV aerial image can be removed, leaving only the highway pavement portion. Then, the trained CrackNet network can be used to detect cracks in the UAV aerial image of the highway pavement, resulting in a crack detection result map for the pavement portion. Finally, the CrackClassification algorithm can be applied to the crack detection result map of the pavement to achieve crack classification. In this way, the impact of ambient noise can be greatly reduced, and GPU resources can be greatly spared.

Due to the lack of a public dataset for UAV remote sensing roadbeds, a dataset, as is shown in Figure 15, was created by hand labeling, which is the second dataset created in this study. The creation process is similar to that of UAV remote sensing road crack dataset. The structure of the roadbed dataset is shown in Table 8.

Table 8. Highway roadbed dataset.

Dataset	Original Images	Ground Truth
Training set	7025	7025
Validation set	2356	2356
Test set	2356	2356
Total	11,737	11,737



Figure 15. A part of the UAV highway roadbed dataset.

Since the roadbed also belongs to the edge information and is less difficult to identify compared to the road cracks, the DenxiDeepCrack model [35] is trained to identify the roadbed. The results of the roadbed inspection are shown in Figure 16a.

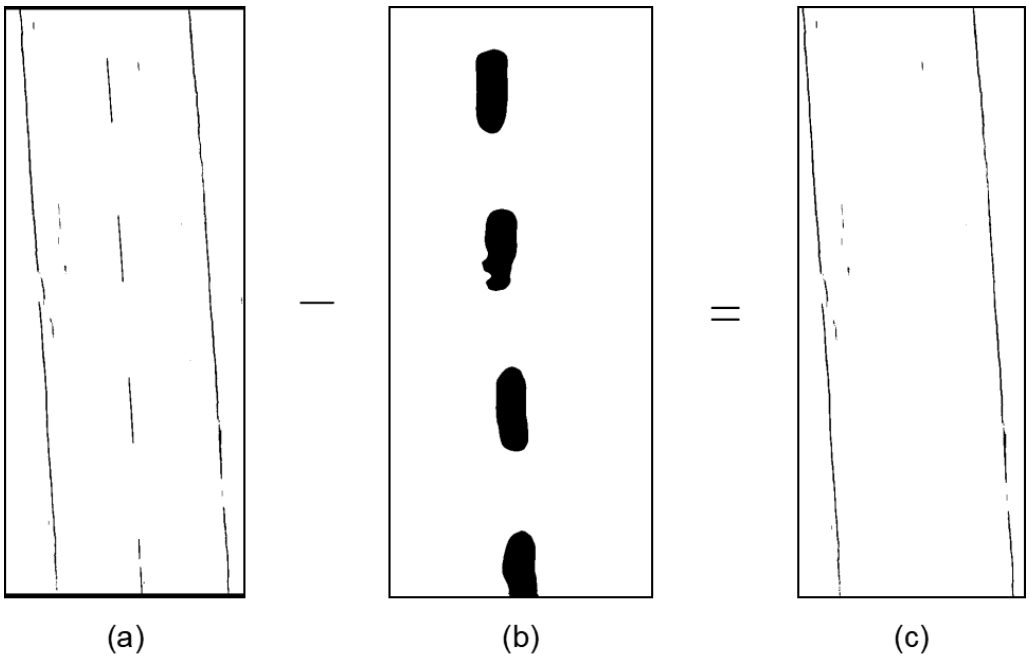


Figure 16. The yellow line removal experiments with diagrams showing (**a**) the roadbed inspection map with the central yellow line, (**b**) the central yellow line inspection map, and (**c**) the roadbed inspection map after processing with the yellow line removal algorithm, where the central yellow line has been removed.

Since the yellow wire in the center of the highway may be identified as the roadbed in the detection results, it must be removed to more accurately obtain the coordinates of the roadbed. The central yellow line removal algorithm can be described as follows: First, binarize the original roadbed inspection result map. Then, subtract the central yellow line detection result matrix from the original roadbed detection result matrix after binarization. Since the yellow line is very thick, as shown in Figure 16b, there will be negative values in the roadbed detection map matrix after obtaining the difference, and the image matrix is numerically processed with negative values set to 0. The image matrix after removing the central yellow line is rewritten, and the roadbed detection results after removing the central yellow line are shown in Figure 16c.

Then, the roadbed pixel coordinates are obtained and recorded in the roadbed inspection result map with the central yellow line removed. A text file with the same name as the inspection result map is created to record the pixel coordinates of the roadbed. The generated roadbed pixel coordinates txt file is shown in Figure 17.

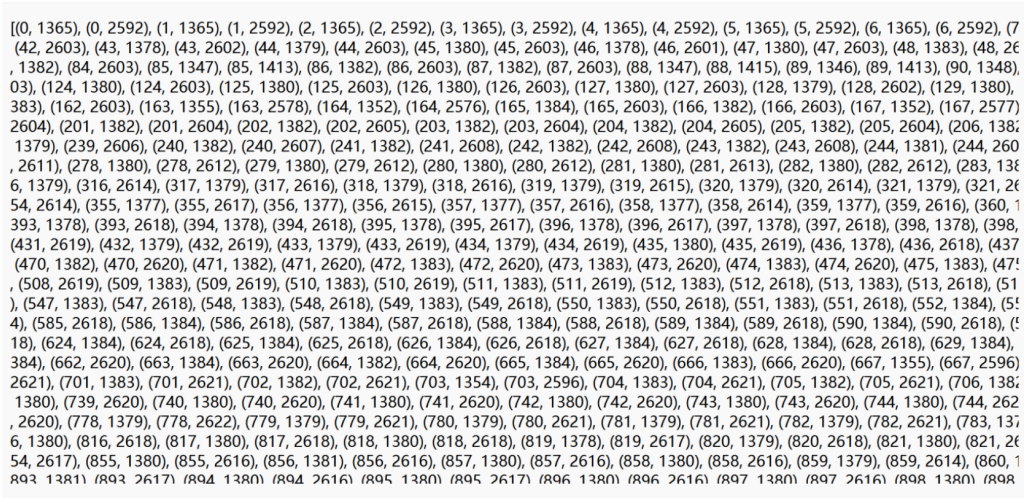


Figure 17. Highway roadbed pixel coordinates txt file.

After obtaining the pixel coordinates of the roadbed, the drone aerial image, as shown in Figure 18a, is cropped based on the roadbed pixel coordinates, removing the environmental part and retaining only the road surface, as shown in Figure 18b. Then, the crack detection network trained in Section 4.1 is used to detect cracks on the cropped image without the environmental noise, as shown in Figure 18c. This approach not only removes the environmental noise but also saves substantial GPU resources. Compared to directly detecting cracks on full-size images and then removing environmental noise based on roadbed recognition, this approach saves about 65% of GPU workload.

4.2.3. Highway Crack Classification

After a series of data preprocessing operations, the CrackClassification algorithm can be used to classify the cracks in the crack detection result image shown in Figure 18c. By comparing with the classification results of manual inspection during the same period, the confusion matrix of crack classification accuracy can be obtained, as shown in Table 9.

For real transverse cracks, 86.8% were detected as transverse cracks, 6.2% were detected as longitudinal cracks, 3.6% were detected as block cracks, and 3.4% were detected as reticulated cracks. For real longitudinal cracks, 7.9% were detected as transverse cracks, 84.9% were detected as longitudinal cracks, 4.8% were detected as block cracks, and 2.4% were detected as reticulated cracks. For real block cracks, 5.9% were detected as transverse cracks, 4.1% were detected as longitudinal cracks, 73.8% were detected as block cracks, and 16.2% were detected as reticulated cracks. For real reticulated cracks, 2.9% were detected as transverse cracks, 2.7% were detected as longitudinal cracks, 17.5% were detected as block cracks, and 77.2% were detected as reticulated cracks.

The evaluation metrics used for highway crack classification are precision and recall. The formulas for precision and recall of each crack category are shown below:

$$Precision_i = \frac{M_{i,i}}{\sum_{j=1}^{4} M_{j,i}} \tag{10}$$

$$Recall_i = \frac{M_{i,i}}{\sum_{j=1}^{4} M_{i,j}} \tag{11}$$

where $Precision_i$ represents the classification accuracy for type i cracks. $M_{i,i}$ represents the number of samples where the true class is type i crack and the predicted class is type i crack. $M_{j,i}$ represents the number of samples where the true class is type j crack and the predicted class is type i crack. $M_{i,j}$ represents the number of samples where the true class is type i crack and the predicted class is type j crack.

According to the confusion matrix, the precision and recall for each type of crack can be calculated, as shown in Table 10. The precision for horizontal, vertical, block, and reticulated cracks are 0.839, 0.867, 0.740, and 0.778, respectively. The recall for horizontal, vertical, block, and reticulated cracks are 0.868, 0.849, 0.738, and 0.772, respectively.



Figure 18. The environmental noise removal experiment with images showing (**a**) the full-size drone aerial image, (**b**) the drone aerial image with environmental noise removed, and (**c**) the crack detection result image for the road surface part. The two arrows from left to right represent the process of cropping the environmental parts from the full-size aerial image based on the roadbed pixel coordinates and detecting cracks on the road surface part of the aerial image.

Table 9. The confusion matrix for the accuracy of the crack classification.

Real Results	Predict Results			
	Transverse Cracks	**Longitudinal Cracks**	**Block Cracks**	**Reticulated Cracks**
Transverse Cracks	86.8%	6.2%	3.6%	3.4%
Longitudinal Cracks	7.9%	84.9%	4.8%	2.4%
Block Cracks	5.9%	4.1%	73.8%	16.2%
Reticulated Cracks	2.9%	2.7%	17.5%	77.2%

In this experiment, crack detection and classification were conducted on 108 km of three highways, and relevant information such as crack type and location were written into a database. By comparing our crack classification data with the corresponding manual road inspection data from that year, we found that the classification accuracy for transverse and longitudinal cracks reached about 85%, while the classification accuracy for block and reticulated cracks reached about 78%. These results confirm the scientific and rational nature of the classification scheme and algorithm used in this study.

Table 10. The precision and recall for four types of cracks.

Category	Precision	Recall
Transverse Cracks	0.839	0.868
Longitudinal Cracks	0.867	0.849
Block Cracks	0.740	0.738
Reticulated Cracks	0.778	0.772

5. Conclusions

The proposed scheme for UAV remote sensing image-based highway crack detection and classification in this paper uses a professional surveying and mapping UAV to collect highway data. A series of environmental noise removal operations were performed on the collected highway data, including creating a roadbed dataset, identifying the roadbed, removing the central yellow line, obtaining roadbed pixel coordinates, and removing environmental noise from the full-size image, resulting in an aerial image of only the highway surface. Then, the CrackNet model, trained on the UAVRoadCrack dataset, was used for crack detection on the highway surface images. Finally, the CrackClassification algorithm was applied to the crack detection result image of the highway surface for crack classification. The main contributions of the paper are as follows:

(1) Creation of a UAV remote sensing road crack dataset, UAVRoadCrack, which fills the gap in the field and facilitates further research.

(2) Proposal of a novel network for UAV remote sensing road crack detection, CrackNet, which uses a multi-scale hybrid dilated convolution (MHDC) module to obtain the global features and multi-scale contextual information of cracks, a multi-head self-attention (MHSA) module to capture the detailed features of the cracks, and group normalization instead of batch normalization to reduce training batch errors. The CrackNet achieved an AP of 0.665, 0.942, and 0.895 on the UAVRoadCrack, CRKWH100, and CrackLS315 datasets, respectively, which is better than for other mainstream crack detection networks. Other evaluation metrics also outperformed other mainstream crack detection algorithms, demonstrating the better generalization and robustness of the CrackNet algorithm.

(3) A preprocessing operation and CrackClassification algorithm were created for crack detection result maps based on roadbed identification, which can effectively exclude environmental noise and classify complex cracks. Compared to directly detecting cracks on full-size images and then removing environmental noise, this approach saves about 65% of GPU resources. The precision for horizontal, vertical, block, and reticulated cracks is 0.839, 0.867, 0.740, and 0.778, respectively. The recall for horizontal, vertical, block, and reticulated cracks is 0.868, 0.849, 0.738, and 0.772, respectively. These results confirm the scientific and rational nature of the classification scheme and algorithm used in this study.

Overall, in this paper, a comprehensive solution is proposed for UAV remote sensing road crack detection and classification, which can significantly improve the efficiency and accuracy of road crack detection and classification. In the future, we can use generative adversarial networks (GANs) to fuse images and remove duplicate areas in consecutive captured images that can then be used to augment the training data and improve the network performance.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z.; software, Y.Z.; validation, Y.Z.; formal analysis, G.S., L.Z., X.W. and F.W.; investigation, G.S.; resources, Y.Z.; data curation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z., L.Z., X.W., F.W. and G.S.; visualization, Y.Z.; supervision, G.S.; project administration, G.S. and Y.Z.; funding acquisition, G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China No.62162059 and supported by the Third Xinjiang Scientific Expediton Program under Grant No.2021xjkk1400.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Merazi Meksen, T.; Boudraa, B.; Drai, R.; Boudraa, M. Automatic crack detection and characterization during ultrasonic inspection. *J. Nondestruct. Eval.* **2010**, *29*, 169–174. [CrossRef]
2. Xu, X.j.; Zhang, X.n. Crack detection of reinforced concrete bridge using video image. *J. Cent. South Univ.* **2013**, *20*, 2605–2613. [CrossRef]
3. Qi, D.; Liu, Y.; Wu, X.; Zhang, Z. An algorithm to detect the crack in the tunnel based on the image processing. In Proceedings of the 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan, 27–29 August 2014; pp. 860–863.
4. Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3708–3712.
5. Bang, S.; Park, S.; Kim, H.; Kim, H. Encoder–decoder network for pixel-level road crack detection in black-box images. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 713–727. [CrossRef]
6. Laxman, K.; Tabassum, N.; Ai, L.; Cole, C.; Ziehl, P. Automated crack detection and crack depth prediction for reinforced concrete structures using deep learning. *Constr. Build. Mater.* **2023**, *370*, 130709. [CrossRef]
7. Zhang, W.; Zhang, Z.; Qi, D.; Liu, Y. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors* **2014**, *14*, 19307–19328. [CrossRef]
8. Cubero-Fernandez, A.; Rodriguez-Lozano, F.J.; Villatoro, R.; Olivares, J.; Palomares, J.M. Efficient pavement crack detection and classification. *EURASIP J. Image Video Process.* **2017**, *2017*, 39. [CrossRef]
9. Oliveira, H.; Correia, P.L. Automatic road crack detection and characterization. *IEEE Trans. Intell. Transp. Syst.* **2012**, *14*, 155–168. [CrossRef]
10. Zalama, E.; Gómez-García-Bermejo, J.; Medina, R.; Llamas, J. Road crack detection using visual features extracted by Gabor filters. *Comput.-Aided Civ. Infrastruct. Eng.* **2014**, *29*, 342–358. [CrossRef]
11. Zou, Q.; Cao, Y.; Li, Q.; Mao, Q.; Wang, S. CrackTree: Automatic crack detection from pavement images. *Pattern Recognit. Lett.* **2012**, *33*, 227–238. [CrossRef]
12. Tang, J.; Gu, Y. Automatic crack detection and segmentation using a hybrid algorithm for road distress analysis. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013; pp. 3026–3030.
13. Avila, M.; Begot, S.; Duculty, F.; Nguyen, T.S. 2D image based road pavement crack detection by calculating minimal paths and dynamic programming. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 783–787.
14. 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]
15. Fan, R.; Bocus, M.J.; Zhu, Y.; Jiao, J.; Wang, L.; Ma, F.; Cheng, S.; Liu, M. Road crack detection using deep convolutional neural network and adaptive thresholding. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 474–479.
16. Nguyen, N.H.T.; Perry, S.; Bone, D.; Le, H.T.; Nguyen, T.T. Two-stage convolutional neural network for road crack detection and segmentation. *Expert Syst. Appl.* **2021**, *186*, 115718. [CrossRef]
17. Hacıefendioğlu, K.; Başağa, H.B. Concrete road crack detection using deep learning-based faster R-CNN method. *Iran. J. Sci. Technol. Trans. Civ. Eng.* **2022**, *46*, 1621–1633. [CrossRef]
18. Djenouri, Y.; Belhadi, A.; Houssein, E.H.; Srivastava, G.; Lin, J.C.W. Intelligent Graph Convolutional Neural Network for Road Crack Detection. *IEEE Trans. Intell. Transp. Syst.* **2022**. [CrossRef]
19. Jiang, X.; Jiang, J.; Yu, J.; Wang, J.; Wang, B. MSK-UNET: A Modified U-Net Architecture Based on Selective Kernel with Multi-Scale Input for Pavement Crack Detection. *J. Circuits Syst. Comput.* **2023**, *32*, 2350006. [CrossRef]
20. 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. [CrossRef] [PubMed]
21. Fernandes, K.; Ciobanu, L. Pavement pathologies classification using graph-based features. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 793–797.
22. Song, W.; Jia, G.; Jia, D.; Zhu, H. Automatic pavement crack detection and classification using multiscale feature attention network. *IEEE Access* **2019**, *7*, 171001–171012. [CrossRef]
23. Hoang, N.D.; Nguyen, Q.L. A novel method for asphalt pavement crack classification based on image processing and machine learning. *Eng. Comput.* **2019**, *35*, 487–498. [CrossRef]
24. Li, B.; Wang, K.C.; Zhang, A.; Yang, E.; Wang, G. Automatic classification of pavement crack using deep convolutional neural network. *Int. J. Pavement Eng.* **2020**, *21*, 457–463. [CrossRef]
25. Li, W.; Huyan, J.; Gao, R.; Hao, X.; Hu, Y.; Zhang, Y. Unsupervised Deep Learning for Road Crack Classification by Fusing Convolutional Neural Network and K_Means Clustering. *J. Transp. Eng. Part B Pavements* **2021**, *147*, 04021066. [CrossRef]

26. Chen, C.; Seo, H.; Jun, C.H.; Zhao, Y. Pavement crack detection and classification based on fusion feature of LBP and PCA with SVM. *Int. J. Pavement Eng.* **2022**, *23*, 3274–3283. [CrossRef]
27. Zou, Q.; Zhang, Z.; Li, Q.; Qi, X.; Wang, Q.; Wang, S. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Trans. Image Process.* **2018**, *28*, 1498–1512. [CrossRef]
28. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
29. Wu, Y.; He, K. Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
30. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:1511.07289.
31. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1451–1460.
32. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
33. 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.
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
35. Li, Y.; Ma, J.; Zhao, Z.; Shi, G. A Novel Approach for UAV Image Crack Detection. *Sensors* **2022**, *22*, 3305. [CrossRef]
36. 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.
37. 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.
38. 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]
39. 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; Part III 18; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.

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.