


Article

Vision Transformer-Based Automatic Crack Detection on Dam Surface

Jian Zhou ¹, Guochuan Zhao ^{1,2,*} and Yonglong Li ^{3,4} 

¹ School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China; jianzhou@swust.edu.cn

² Sinograin Chendu Storage Research Institute Co., Ltd., Chendu 610000, China

³ Sichuan Energy Internet Research Institute, Tsinghua University, Chengdu 610213, China; liyonglong@tsinghua-eiri.org

⁴ Tianfu Yongxing Laboratory, Chengdu 610213, China

* Correspondence: guochuanzhao@163.com

Abstract: Dam is an essential structure in hydraulic engineering, and its surface cracks pose significant threats to its integrity, impermeability, and durability. Automated crack detection methods based on computer vision offer substantial advantages over manual approaches with regard to efficiency, objectivity and precision. However, current methods face challenges such as misidentification, discontinuity, and loss of details when analyzing real-world dam crack images. These images often exhibit characteristics such as low contrast, complex backgrounds, and diverse crack morphologies. To address the above challenges, this paper presents a pure Vision Transformer (ViT)-based dam crack segmentation network (DCST-net). The DCST-net utilizes an improved Swin Transformer (SwinT) block as the fundamental block for enhancing the long-range dependencies within a SegNet-like encoder–decoder structure. Additionally, we employ a weighted attention block to facilitate side fusion between the symmetric pair of encoder and decoder in each stage to sharpen the edge of crack. To demonstrate the superior performance of our proposed method, six semantic segmentation models have been trained and tested on both a self-built dam crack dataset and two publicly available datasets. Comparison results indicate that our proposed model outperforms the mainstream methods in terms of visualization and most evaluation metrics, highlighting its potential for practical application in dam safety inspection and maintenance.

Keywords: Swin Transformer; Vision Transformer; feature fusion; concrete dam; crack detection



Citation: Zhou, J.; Zhao, G.; Li, Y. Vision Transformer-Based Automatic Crack Detection on Dam Surface. *Water* **2024**, *16*, 1348. <https://doi.org/10.3390/w16101348>

Academic Editor: Chin H Wu

Received: 30 March 2024

Revised: 25 April 2024

Accepted: 8 May 2024

Published: 9 May 2024



Copyright: © 2024 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

Dams are essential hydraulic structures of hydropower stations, playing a significant role in economic development and societal advancement. However, prolonged exposure to environmental erosion, internal chemical reactions, and various loads can precipitate damage within the concrete dam structure [1]. Among the foremost concerns are cracks, which pose a significant threat to the integrity, durability, strength, and stability of concrete dams [2,3]. For instance, both the Koelnbrein Dam in Australia and the Chencun Dam in China have experienced breaches in their grouting curtains due to the presence and propagation of dam cracks [4]. Hence, conducting routine and methodical inspections to identify and assess cracks in concrete dams is paramount to guaranteeing their safe and stable operation. The current primary approach for detecting dam surface cracks relies on traditional visual inspection [5], which is inherently subjective in both quantitative and qualitative assessments of cracks. Additionally, this method presents drawbacks such as time-consuming procedures, labor-intensive efforts, safety concerns, and limitations in monitoring accuracy and coverage due to the environmental conditions and scale of dams. Moreover, achieving continuous crack monitoring and longitudinal comparisons over time proves unrealistic under manual monitoring practices. Tragically, errors in crack

detection contributed to the failure of the Canyon Lake Dam, resulting in the deaths of 236 individuals, as documented [6]. Additionally, the tragic loss of 76 lives occurred in the failure of the Austin Dam due to insufficient attention and maintenance of cracks and defects [7]. Therefore, the formulation and implementation of a highly precise and efficacious crack detection methodology is essential for the autonomous evaluation of dam integrity.

With the rapid advancements in computer hardware and computer vision technology, vision-based automated detection methods have demonstrated notable success in detecting concrete surface cracks in civil infrastructure [8–10]. In these vision-based approaches, a crucial step is to extract features from the images that are sensitive to cracks. Previous studies [11–13] have proposed utilizing image processing techniques to extract crack-sensitive features. For instance, Fan et al. [12] designed CrackLG for underwater dam crack detection. First, a k-means clustering algorithm was employed to distinguish the image blocks including cracks. Then, the final crack areas were extracted using global feature information. However, handcrafted feature-based crack detection approaches suffer from limitations in terms of accuracy and generalization.

Deep learning (DL), with its ability to automatically extract relevant features from data, has become a preferred approach for crack detection [14], yielding promising results in various applications [15]. DL-based concrete surface crack detection methods can be classified into classification-oriented, object-oriented, and segmentation-oriented methods. For example, Zhang et al. [16] proposed a dam crack detection method based on an improved ResNet algorithm using knowledge distillation. Initially, the improved residual neural networks were trained on mini-ImageNet for multi-classification. Subsequently, a parameter/model transfer method was employed to achieve crack detection. However, the image classification-based algorithms just ascertain whether cracks are present or absent within images, without providing specific information about the structural characteristics of the cracks [17], such as their width, length, or orientation.

One potential solution is the object recognition-based method that enables the direct acquisition of both the positional coordinates and categorical labels of objects through the utilization of bounding boxes. The most popular architecture used for performing object recognition in crack detection area is faster R-CNN [18], such as for concrete crack detection [19] and road crack detection [20]. For dam crack detection, Xu et al. [21] proposed AF-RCNN (Attention-based Faster-RCNN), achieving an mAP (mean Average Precision) of 81.07% on an expanded dam crack dataset, surpassing the performance of the original Faster-RCNN. YOLO families are also used as the main architectures in object recognition tasks. For example, YOLOX [22] and YOLOv5 [23] are used for dam crack detection. However, these object recognition-based methods, with the aid of bounding boxes, still suffer from the drawback of coarse crack localization and struggle to encompass a full longitudinal crack within a single bounding box [5].

Another alternative approach performing crack classification is the semantic segmentation-based method. By distinguishing between the background and the cracks at the pixel level, the semantic segmentation-based method inherently confers a distinctive merit in terms of achieving superior accuracy in demarcating the spatial boundaries of cracks within images. Considering the crack images with high resolution, Zhang et al. [24] introduced a dam crack detection method. Initially, a CNN was trained for crack classification, followed by the utilization of an FCN to achieve crack segmentation. Similarly, Pang et al. [25] firstly employed a target detection method to identify cracks using bounding boxes. Subsequently, crack segmentation was performed using image-processing techniques. Within the domain of crack detection research, the majority of crack segmentation tasks are predominantly executed through the utilization of encoder–decoder architectures [26–28]. Various well-known architectures and their improvements such as UNet [29,30] and DeepLabv3+ [31,32] were also proposed to conduct dam crack detections. These deep learning-driven semantic segmentation methodologies have been found to offer enhanced detection results under challenging and noisy environmental scenarios, and precision measurement in

cracks [27,31]. However, due to the inherent bias in the convolutional structure, it cannot fully understand global semantic information, which to some extent limits the accuracy and robustness of crack detection.

Comparative to CNNs, the transformer architecture has emerged as a formidable paradigm, exhibiting notable prowess in the domain of Natural Language Processing (NLP) by leveraging its capacity to capture global and long-range information. Google introduced the Vision Transformer (ViT) in 2020 [33], employing transformer structures for image classification and establishing itself as the leading network at that juncture [34]. The ViT-based methods introduce novel model design concepts for CV tasks. In the research field of crack detection, it has been demonstrated that ViT-based methods can be employed for autonomous and efficient dam crack segmentation [35]. For example, CrackFormer [36] is designed for fine-grained crack detection. Ref. [37] introduced a ViT-based framework for crack segmentation on concrete surfaces, showcasing the robust performance of ViT across various types of noise signals. ViT utilizes a self-attention mechanism to extract and integrate contextual information. Nevertheless, ViT's tokens are fixed in number and dimensions, limiting its ability to train and predict at multiple scales. Additionally, employing the ViT algorithm for crack identification entails substantial computational expenses. To address these constraints, Liu et al. [38] introduced the Swin Transformer, which utilizes non-overlapping shift windows for self-attention computation, enabling distinct windows to interact with one another. Swin Transformer has demonstrated significant success in pavement crack detection tasks [39–41]. Some studies utilized the Swin Transformer as an encoder for extracting deep feature representations [42,43]. However, unlike pavement cracks, dam surface cracks in the real world exhibit distinct characteristics such as significant image noise, high background complexity, and considerable scale diversity. When applying these advanced methods directly for dam crack detection purposes, there is a tendency for suboptimal performance outcomes [44].

This paper introduces a new semantic segmentation network for dam crack detection, named DCST-net. To the best of our knowledge, DCST-net is a pioneering instance of a pure ViT-based SegNet-like structure for dam surface crack segmentation. Specifically, its symmetric encoder and decoder are both constructed based on the improved Swin Transformer blocks (SwinT blocks). In addition, it employs a weighted attention block on the encoder and the corresponding decoder features for activating crack features and suppressing the non-crack ones. The principal contributions of this paper can be summarized as follows:

1. It is the first attempt to perform dam surface crack detection with a pure ViT-based encoder–decoder network (DCST-net); our approach yields superior crack segmentation performance on the dam crack dataset collected from a real dam surface as well as two open benchmark crack datasets, outperforming state-of-the-art models;
2. To establish long-range pixel interaction, we propose an improved SwinT-block as the fundamental unit of the DCST-net; this block efficiently extracts contextual information across feature channels through the utilization of depth-wise separable convolution kernels (DWConv); moreover, it integrates spatial domain contextual information through a proficient position-encoding scheme, thereby capturing a wide receptive field;
3. To alleviate the loss of semantic details, we introduce a weighted attention module; it utilizes features from the encoder to produce an attentive mask, which serves as attention coefficients; these coefficients are then multiplicatively applied element-wise to the corresponding features in the decoder, thus suppressing non-crack features while enhancing crack features.
4. To facilitate the training of deep networks, we propose a multi-level label supervision training method, which directly supervises different depth feature layers with crack labels; in addition, we design a hybrid loss function to overcome the problem of class imbalance in crack images.

The rest of the paper is organized as follows: Section 2 presents the proposed DCST-net and elaborates its each block; Section 3 introduces the implementation details, datasets

and evaluation metrics; Section 4 demonstrates the extensive experiments results and companions; and Section 5 summarizes the findings and the superiority of our model.

2. Methodology

Generally, crack segmentation networks use feature extractors to reduce feature resolution and extract high-dimensional semantic features to identify cracks. Subsequently, the low-resolution and high-dimensional features are restored to the original image resolution through interpolation or deconvolution, thus producing semantic segmentation results. Convolutional structure-based crack segmentation methods extract features by stacking convolutional layers [16,17,26,32]. In shallower layers, convolution kernels have a smaller receptive field, observing only local features of cracks, which is not favorable for crack orientation detection. Larger receptive fields are only available in deeper layers. In contrast, ViT-based crack segmentation networks [35,44] utilize a self-attention mechanism [21,27,36] to construct a feature extractor and acquire the global semantic features of cracks. This approach addresses the issue of inadequate global information in low-dimensional features, thereby enhancing the prediction of crack direction. In addition, the encoding–decoding structure [26,29–31,36] is a commonly employed design in semantic segmentation networks, where encoding corresponds to feature extraction, and decoding involves resolution recovery. Moreover, skip connections [16,26,32] are implemented between the encoding and decoding layers of the identical resolution to mitigate the loss of fine-grained details and boost crack segmentation. Motivated by these effective mechanisms, we propose DCST-net to enhance the segmentation performance for dam cracks in real-world scenarios.

2.1. Architecture of DCST-Net

The DCST-net embraces a SegNet-like encoder–decoder design, as depicted in Figure 1, comprising an encoder, decoder, and a weighted attention module. To effectively capture global and long-range semantic interdependencies, the encoder–decoder module is built upon an improved SwinT-block (presented in Section 2.2), serving as its fundamental unit. In order to refine the segmentation results in detail-rich regions, a weighted attention module (presented in Section 2.4) is adopted between the symmetric encoder and decoder, serving as a filter suppressing other interfering features.

Firstly, the input image, dimension as $[H, W, 3]$, undergoes a patch-embedding module, dividing it into non-overlapping patches (size of 4×4). This results in each patch having a feature dimension of $4 \times 4 \times 4 = 48$. Meanwhile, a local detail module (the upper right of Figure 1) is designed to down-sample the original image by a factor of 2 to obtain local detail features with a dimension of $[H/2, W/2, 48]$. Subsequently, these patch tokens are inputted into a five-stage Swin Transformer-based encoder network, to extract multi-scale features. Within this process, the patch-merging layer [38,45] is tasked with down-sampling and expanding the feature dimension, while the improved SwinT-block focuses on feature representation learning. In contrast, the corresponding Swin Transformer-based decoder network works to enable the restoration of the resolution of the encoder output feature. Specifically, a patch-expanding layer [38,45] is employed for the purpose of up-sampling. This is achieved stage by stage, until the feature map resolution is regained as one-half of the authentic image size. Finally, at each stage, the weighted attention module is designed to concatenate the features from the corresponding encoder and decoder, generating an attentive mask to refine the predicted crack. Subsequently, the fused feature at the current stage undergoes up-sampling by a factor of 2 before being passed to the next stage. This process iterates five times, yielding the final predicted results with identical dimensions as the original input images. Each block of the DCST-net is elaborated in the following.

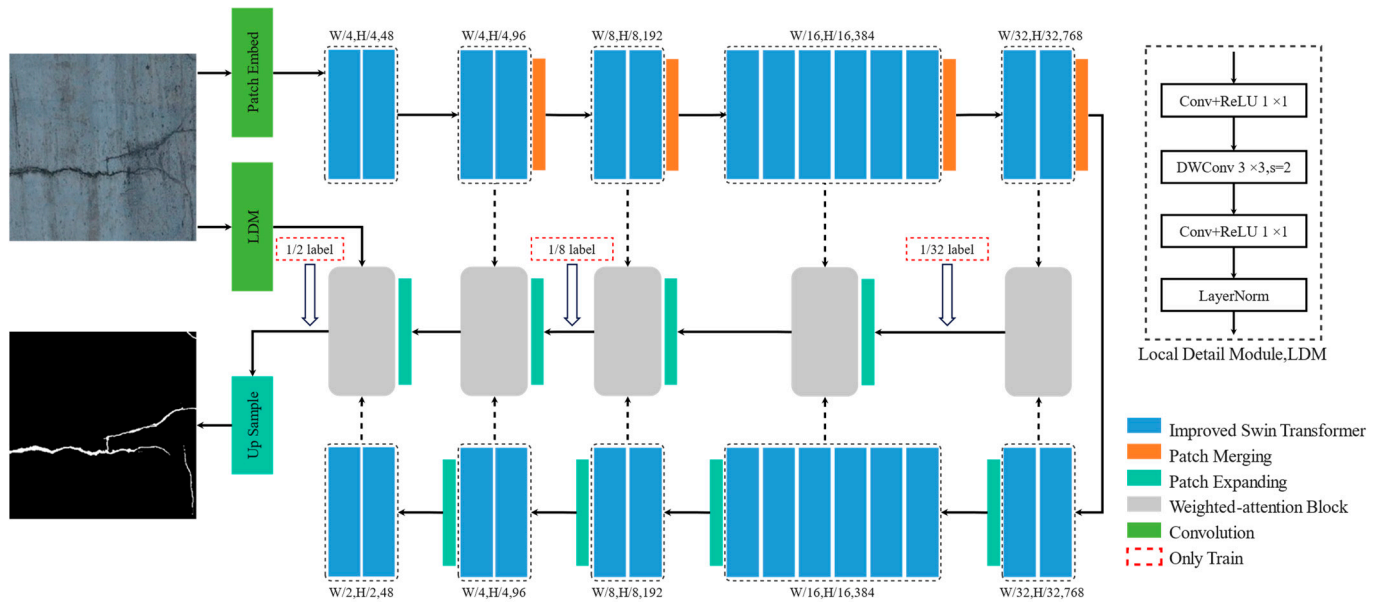


Figure 1. Overview of DCST-net architecture.

2.2. The improved Swin Transformer Block

In Figure 2, the replacement of the MLP in the original SwinT-block [38] with a 1×1 DW-Conv results in the two sequential improved SwinT-blocks. Each block comprises a multi-head self-attention module, layer normalization (LN), residual connection and 1×1 depth-wise separable convolutions (DW-Conv). The W-MSA module and the SW-MSA module, adopted in the two consecutive improved SwinT-blocks, utilize improved self-attention computation, as illustrated in Figure 3. These modifications lead to an enlarged receptive field and the acquisition of cross-channel semantic features.

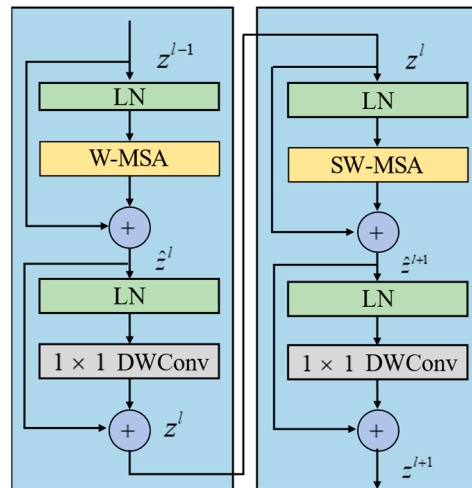


Figure 2. Two successive improved Swin Transformer blocks.

Let $X \in R^{d_{in} \times W \times H}$ be the input tensor, where W , H and d_{in} are the width, height and dimension of the input tensor, respectively. Similar to the work [45], the computation of self-attention proceeds as follows:

$$F^c = SoftMax\left(\frac{qk^T}{\sqrt{d}} + b\right)v \tag{1}$$

where q , k and v , are generated by 1×1 DW-Conv and denote the query, key and value matrices, respectively. The dimension of the query or key is set as d ; b presents a relative position vector and is a learnable weight parameter.

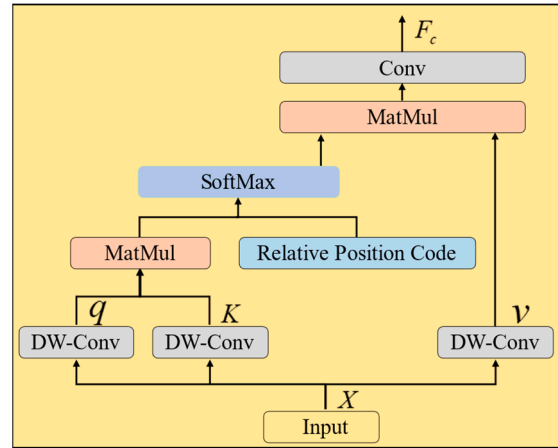


Figure 3. The improved self-attention block.

The two successive improved SwinT-blocks utilize *W*-MSA and *SW*-MSA in pairs. *W*-MSA efficiently reduces the computational cost during self-attention computation, while *SW*-MSA conducts cross-region self-attention computation to obtain a global perspective, effectively enhancing the segmentation performance of dam surface cracks. The computation of continuous SwinT-blocks is given as follows:

$$\hat{z}^l = W - MSA\left(LN\left(z^{l-1}\right)\right) + z^{l-1}, \quad (2)$$

$$z^l = DWConv\left(LN\left(\hat{z}^l\right)\right) + \hat{z}^l, \quad (3)$$

$$\hat{z}^{l+1} = SW - MSA\left(LN\left(z^l\right)\right) + z^l, \quad (4)$$

$$z^{l+1} = DWConv\left(LN\left(\hat{z}^{l+1}\right)\right) + \hat{z}^{l+1}. \quad (5)$$

where \hat{z}^l and z^l are the outputs of the (S)W-MSA module and the DW-Conv module of the l^{th} block, respectively.

2.3. Swin Transformer-Based Encoder

The encoder of DCST-net comprises improved SwinT-blocks and patch merging layers, organized into 5 stages based on the {2,2,2,6,2} layout. As depicted in Figure 1, except for the initial stage, which lacks patch-merging layers, the subsequent stages consist of either 2 or 6 improved SwinT-blocks succeeded by a patch-merging layer. The patch tokens undergo representation learning within the two successive improved blocks, preserving the consistency of feature dimension and resolution. Simultaneously, the patch-merging layer [38,45] down-samples the token number by a factor of 2 and increases the feature dimension to double its original dimension. Therefore, the feature sizes outputted by the 5 stages are, respectively, 1/4, 1/4, 1/8, 1/16 and 1/32 of the original image size.

2.4. Swin Transformer-Based Decoder

Referring to the encoder, the decoder of the DCST-net has a symmetrical layout of {2,6,2,2,2}. On the contrary, the patch-merging layer located behind the improved SwinT-blocks is replaced by the patch-expanding layer. It serves as the inverse process of patch merging, doubling the resolution of the input feature tensor while halving the number of channels. Finally, the feature sizes outputted in decoder are, respectively 1/32, 1/16, 1/8, 1/4 and 1/2 of the original image size.

2.5. Weighted Attention Block

The SwinT-blocks have been demonstrated to be effective for modeling global dependencies, making them valuable for segmenting long cracks [21]. However, relying solely on SwinT-blocks may not be sufficient for addressing cases involving fine-grained cracks with strong background noise. To address this problem, drawing inspiration from the attention gate in U-Net [46] and the scaling attention in SegNet [36], it is evident that suppressing irrelevant regions while highlighting salient features crucial for the segmentation task offers a straightforward yet effective solution. To achieve this, similar to [36], we adopt a weighted attention model between the symmetric encoder and decoder at the same stage. The features from the encoder are utilized to generate a self-attention mask as attention coefficients ranging from 0 to 1, which are then element-wise multiplied with the related features in the symmetric decoder. This process, depicted in Figure 4, functions as a filter that activates crack-related features while suppressing the non-crack ones.

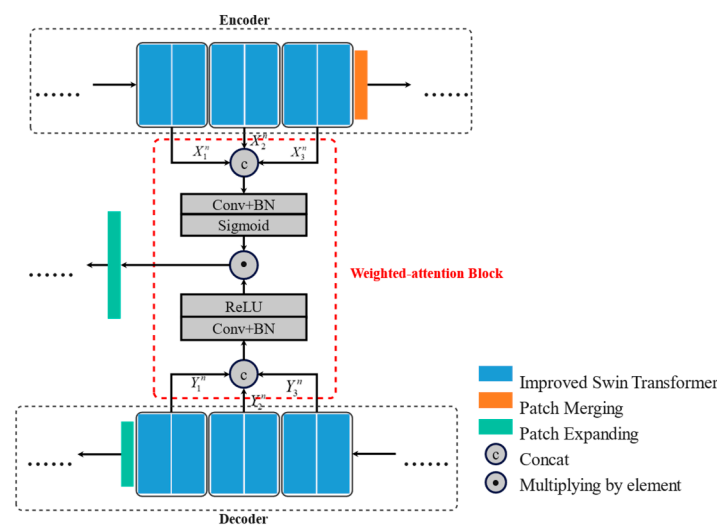


Figure 4. The weighted attention block.

Let us take the feature fusion with the two continuous Swin Transformers in stage n as an example. Features in the symmetric encoder and decoder are denoted as $\{X_1^n, X_2^n, X_3^n\}$ and $\{Y_1^n, Y_2^n, Y_3^n\}$, respectively. As presented in Figure 4, the mask L_{mask}^n is then given by (Equation (6)) as follows:

$$L_{mask}^n = \delta(BN(\otimes_{3 \times 3}(Concat(X_1^n, X_2^n, X_3^n)))) \tag{6}$$

where $Concat()$ is the concatenation operation, $\otimes_{3 \times 3}$ represents the convolution operation with a 3×3 convolution kernel, BN is the batch normalization operation, δ denotes the sigmoid activation function. Subsequently, the fusion result, S_{side}^k denoted as side output, is obtained by the weighted attention block as follows:

$$S_{side}^k = L_{mask}^k \odot BN(\otimes_{3 \times 3}(Concat(Y_1^n, Y_2^n, Y_3^n))), \tag{7}$$

where \odot denotes an element-wise multiplication operation. The side output is then up-sampled by a factor of 2 using the patch-expanding operation. As shown in Figure 1, this procedure is repeated 5 times, ultimately obtaining a semantic segmentation prediction image that matches the dimensions of the original image.

2.6. Loss Function

In general, crack segmentation tasks suffer from a severe imbalance between positive and negative samples [47]. For example, through the analysis of our self-built dataset (Section 3.1), it can be found that the average proportion of crack pixels is only 1.34%, while

background pixels reach 98.66%. A substantial quantity of background samples facilitate the model's fast learning of background prediction, resulting in a more rapid reduction in background prediction loss compared to that of crack samples. Additionally, despite the small loss for each background pixel, the overall background loss is substantially higher than the crack loss due to the high proportion of background samples. Consequently, the network prioritizes learning background information, leading to challenges such as low accuracy and suboptimal performance in crack segmentation. To tackle the issue of class imbalance between crack and background, one intuitively simple method is to randomly exclude some background pixels from the training process. However, its effectiveness in improving the crack segmentation is limited. Essentially, the poor performance in crack segmentation is caused by the overwhelming effect of crack loss being dominated by background loss. Therefore, we seek the help from hybrid loss function.

In this work, Weighted Cross-Entropy loss and Dice loss are employed to form the hybrid loss function. It is calculated as follows:

$$loss = loss_{wce} + \lambda * loss_{dice} \quad (8)$$

where $loss_{wce}$ and $loss_{dice}$ present Weighted Cross-Entropy loss and Dice loss, and λ is the weight ratio between these two loss functions. As the network progresses through multiple epochs and gains preliminary understanding of both background and crack foreground information, the Weighted Cross-Entropy loss tends to be a certain multiple of the Dice loss. Accordingly, λ is set to an appropriate multiple to optimize the performance of the hybrid loss function.

By introducing weight coefficients for positive and negative samples separately, the Weighted Cross-Entropy loss function becomes more sensitive to the higher-weighted parts, thereby enhancing the learning focus on the target category. Weighted Cross-Entropy loss is defined as follows:

$$loss_{wce} = \frac{1}{n} \sum_i^n w_f true_f \log(pred_f) + w_b true_b \log(pred_b), \quad (9)$$

where w_f is the weight for crack loss; w_b is the weight for background loss; $true_f$ and $true_b$ refer to the crack label and the background label, respectively; $pred_f$ and $pred_b$ denote the predicted probabilities of cracks and background, respectively; and n presents the total number of pixels.

The Dice loss quantifies the dissimilarity between crack prediction values and the true values by directly utilizing Intersection over Union (IoU), aiming for model optimization. Its formulation is represented by Equation (10) as follows:

$$loss_{dice} = \frac{2 * true \cap pred}{true \cup pred + true \cap pred} \quad (10)$$

where $true$ is the ground truth value and $pred$ is the predicted value.

2.7. Multiple-Level Label Supervision

To enhance the utilization of features at different levels, the implementation in this paper adopts three additional low-resolution labels during the training phase to supervise the parameter updates of deep networks. These low-resolution labels are obtained by down-sampling the original image labels by 1/2, 1/8, and 1/32, respectively (as shown in Figure 1). The rationale behind incorporating these low-resolution labels is based on their unique information content; the 1/2 scale contains richer details of crack information, the 1/32 scale encompasses broader global crack backbone information, and the 1/8 scale lies between these two, exhibiting larger discrepancies. This comprehensive approach enhances model performance and generalization ability, striking a balance between accuracy and speed.

Training the model at different resolutions facilitates a better understanding and adaptation to images of various scales and resolutions, thereby improving the model's generalization ability. Additionally, introducing extra low-resolution labels can increase the training data volume, reducing the risk of overfitting. Implementing multi-level label supervision during training increases the training burden and diminishes training speed, while the multi-level label structure is removed during testing, thus exerting no influence on inference speed.

3. Experiment Preparation

3.1. Dataset Description

In order to evaluate the effectiveness of our proposed crack segmentation network in real-world engineering scenarios, we compiled a dataset named DamSCrack, specifically focusing on dam surface cracks. Furthermore, two openly accessible datasets, namely DeepCrack [48] and Crack500 [49], were employed to validate the generalization capability of the proposed network.

DamSCrack: The dam surface crack images were initially captured using a drone at a hydropower station located in Sichuan, China. As shown in Figure 5, the hydroelectric power station employs a concrete gravity dam design, incorporating eight surface spillways in the overflow section and five bottom outlets in the non-overflow section. Spanning a length of 995.4 m, the dam rises to a height of 465 m. Then, these real-world images underwent cropping and selection. We further meticulously screened, diagnosed, and manually annotated them under the guidance of domain experts, utilizing the ImgAnnotation software as our annotation tool. The annotation process is detailed as follows: firstly, we employed the 1×1 pixel-sized annotation pen from ImgAnnotation to outline the crack's edge and refine its shape; secondly, based on the crack width, we utilized an annotation pen ranging from 3×3 to 10×10 pixels to fill the crack, achieving pixel-level labeling; finally, DamSCrack was generated, comprising 1000 crack images with a resolution of 448×448 , and each crack image had a binary label image, with red denoting the crack and black representing the background.

In our experiments, the DamSCrack dataset was randomly divided into training, validation, and test set with the ration of 8:1:1, respectively. Then, five image augmentation strategies, including random brightness variation, random rotation, erasing, blending, and shear blending, were applied to the training set images. These strategies aimed to provide the network with more challenging samples, thereby improving the model's robustness. Regarding every crack image within the training dataset, the five strategies independently occurred with a probability of 0.5. This way, the training dataset expanded from 800 images to 2800 images.

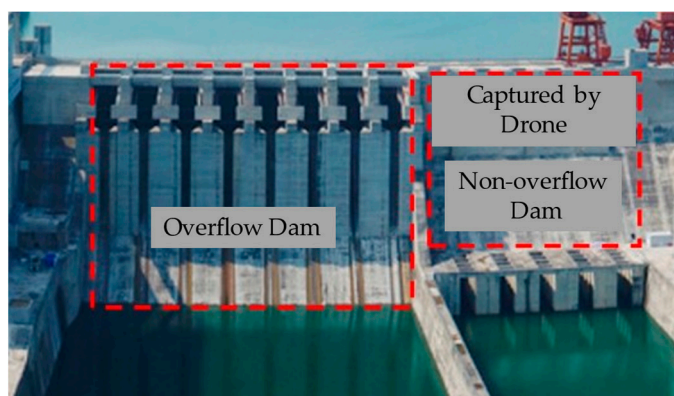


Figure 5. The on-site situation of the dam.

Crack500: This dataset consists of 500 RGB images depicting pavement cracks. Each image with a resolution of 2000×1500 is accompanied by a pixel-level binary label image,

featuring a black background and a white representation of the crack. Due to the limited number and high resolution of this dataset, we cropped each image into 12 non-overlapping sub-images with a resolution of 448×448 . Following the approach in [49], we retained the sub-images containing a minimum of 1000 pixels of cracks. Subsequently, a total of 3368 crack images were selected and subsequently partitioned into distinct subsets for training, validation, and testing purposes, with the ratio of 8:1:1, in our experiments.

DeepCrack: There are 537 RGB crack images with a resolution of 544×384 in this dataset, covering concrete surface cracks with various scenes and multiple scales. Each crack image corresponds to a manually annotated pixel-level mask label, where the background is white, and the cracks are black. In our experiments, the original images in the DeepCrack dataset underwent a scaling operation, where the long side was scaled to 448, and the short side was padded with black pixels to achieve an RGB image with a resolution of 448×448 . Similarly, training images, verification images, and test images were set with the ratio of 8:1:1.

3.2. Experiment Settings

All experiments in this paper operate in an identical hardware and software environment. The details of experiment settings are listed in Table 1. The DCST-net was optimized by using the stochastic gradient descent (SGD) method. The initial learning rate was 1×10^{-5} and the batch size was 8. In addition, the parameters of the comparative methods (Section 4.3) were set to be the same as those in the original paper.

Table 1. Software and hardware configuration.

Hardware/Software	Parameters/Version
CPU	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz
GPU	2 × NVIDIA GTX TITAN Xp, 24 GB memory
RAM	DDR4, 32 GB
Operating System	Ubuntu18.04
Python	3.6
CUDA	10.2
Cudnn	8.0.2
TensorRT	7.0

3.3. Evaluation Metrics

Three commonly used evaluation metrics for binary classification, namely precision (*Pre*), recall (*Rec*), and F1 score (*F1s*) were employed in this work. Meanwhile, the commonly used intersection over union (*Iou*) metric for measuring the segmentation effect was also adopted in this work. Similar to other works, calculation formulas of these four evolution metrics are as follows:

$$Pre = \frac{TP}{TP + FP}, \quad (11)$$

$$Rec = \frac{TP}{TP + FN}, \quad (12)$$

$$F1s = \frac{2 \times Pre \times Rec}{Pre + Rec}, \quad (13)$$

$$Iou = \frac{TP}{TP + FN + FP}. \quad (14)$$

where *TP* (True Positive) and *FP* (False Positive) mean that a crack is correctly predicted as a crack and a non-crack is wrongly detected as a crack at pixel level, respectively; *FN* (False Negative) denotes that a crack is wrongly detected as a non-crack at pixel level.

4. Experimental Results and Discussion

4.1. Analysis of Training Results

Figure 6 illustrates the loss curve of training and validation during the training process of the DCST-net on DamSCrack dataset. It can be observed that the model's loss rapidly decreased from an initial value of 51 to below 10 in just over 10 epochs, with a decrease of over 80%. This indicates that the selection of training parameters and the design of the loss function in this paper are appropriate, leading to a rapid convergence of the network. As the training epoch hits the 600th round, the network has essentially reached the convergence limit, maintaining stability in loss with an average intersection over union of 67.21%. Throughout the entire training process, the validation loss curve maintains a consistent and proximate alignment with the training loss curve, indicating that the model avoids overfitting and possesses strong generalization capabilities.

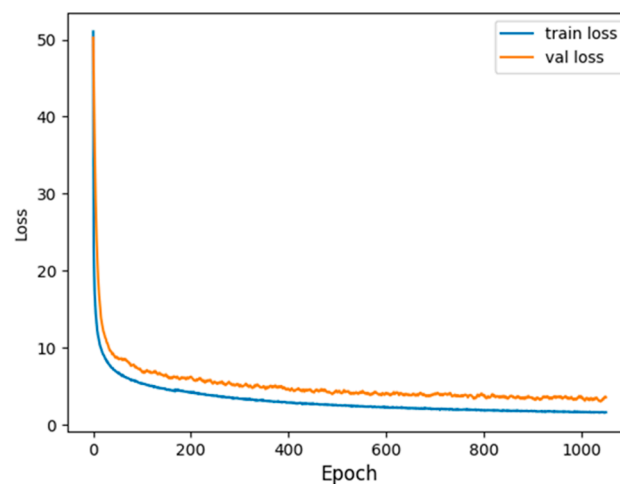


Figure 6. Loss decay curve during training.

Figure 7 presents the curves of precision, recall, and average intersection over the union on validation dataset during the training process. These curves clearly show that in the early stages of network training, precision and average intersection over union rapidly increase. Meanwhile, recall, starting from a relatively high value, quickly optimizes downwards. This indicates that the network is rapidly learning crack segmentation knowledge, improving its ability to segment cracks and outlining the general framework of cracks. By the 600th training round, although the network loss remains stable, the three evaluation metrics continue to improve. This suggests that the class-balanced loss function designed in this paper continues to play a role, allowing the network to continuously learn and optimize for crack details, enhancing the segmentation of crack details. By the 900th training round, the three evaluation metrics essentially stop changing, indicating that the network has completed training without overfitting and exhibits high performance in crack segmentation.

Figure 8 shows the prediction results of the DCST-net on the typical dam surface crack images in the test dataset of DamSCrack. These results clearly indicate that the DCST-net can accurately address the practical problem of dam surface crack detection, even when faced with challenging cracks, such as those with complex branching structures, slender characteristics, intensity in-homogeneity, and strong background interference.

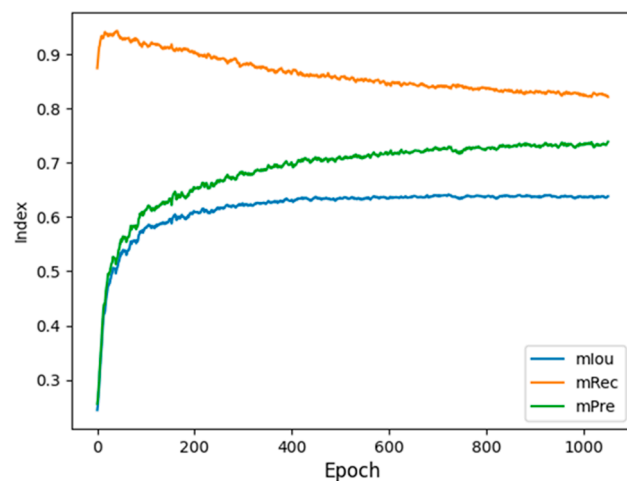


Figure 7. Evaluation index transformation curve during training.

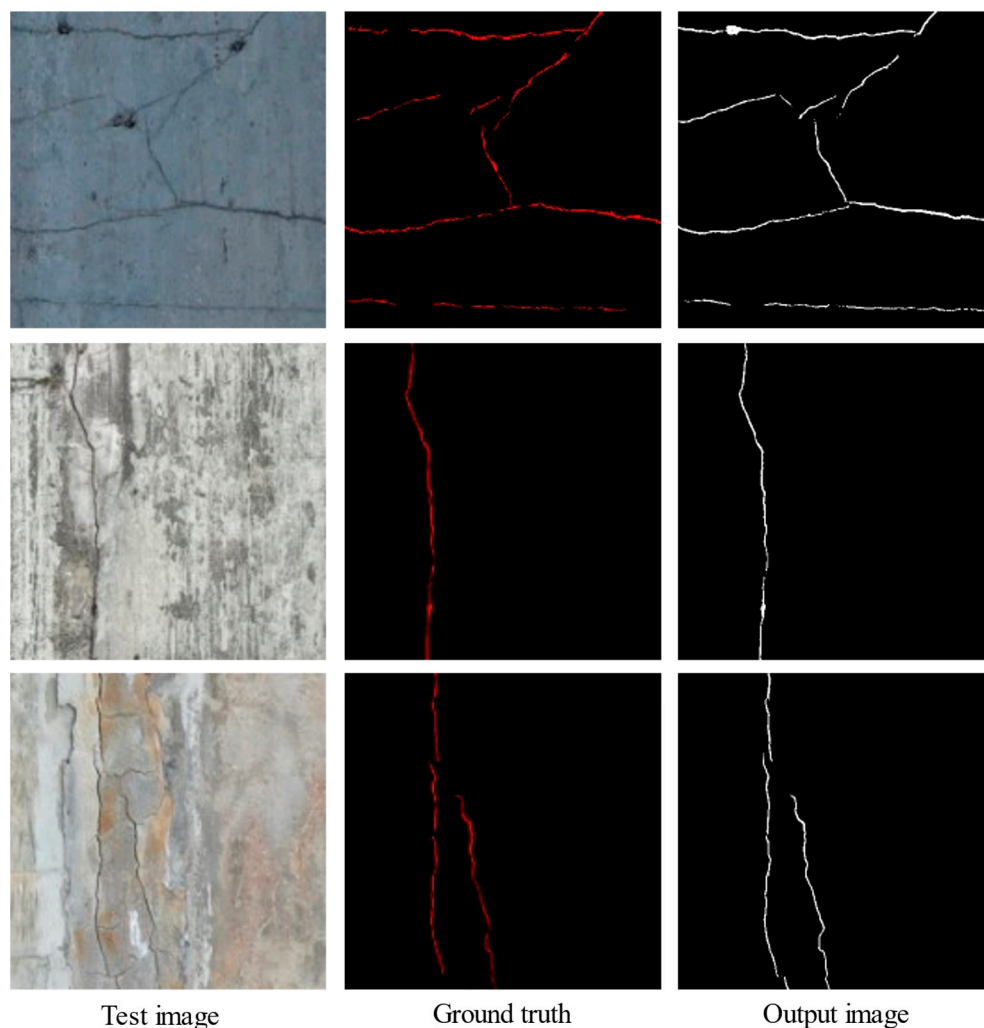


Figure 8. Segmentation results of the DCST-net for typical cracks in the DamSCrack dataset.

4.2. Ablation Study

To further validate the advantage derived from each improved module in DCST-net, an ablation analysis is conducted on the DamSCrack dataset. Specifically, the proposed improvement modules in this paper are added one by one to the Swin Transformer network, resulting in a total of 6 ablation methods (method a, b, . . . ,f), as presented in Table 2. The

resulting evaluation metrics are presented in Table 3, and the related predictions are visually shown in Figure 9.

Method a: It utilizes the Swin Transformer [38] serving as the backbone for feature extraction, employs up-sampling to restore feature resolution, and thus, outputs the semantic segmentation results. Benefiting from the powerful global representation capability of the self-attention mechanism, method a can segment the crack backbone. However, its segmentation of the local parts of cracks is not satisfactory due to the loss of low-level features, as illustrated in the third column of Figure 9.

Method b: It replaces all SwinT-blocks in method a with improved SwinT-blocks. As indicated in the third row of Table 3, the improved SwinT-blocks considerably boosts the evaluation metrics *Pre*, *Rec*, *F1s* and *mIoU* by 4.50%, 0.80%, 2.92% and 3.24%, respectively, achieving better segmentation performance.

Method c: It introduces only the weighted attention block to method a. The weighted attention module generates a self-attention mask to suppress non-semantic features, leading to the optimized segmentation of the local parts of cracks (the fifth column in Figure 9), with an increase in *Pre*, *Rec*, *F1s* and *mIoU* by 4.78%, 1.69%, 3.30%, and 3.66%, respectively.

Method d: Both the improved SwinT-blocks and the weighted attention module are introduced to method a. These two modules jointly significantly boost the crack segmentation performance, with a corresponding to an increase in *mIoU* by 8.42% and the maximum value of 86.40% for *Rec*.

Method e: Methods a, b, c, and d utilize the cross-entropy loss function provided by PyTorch for loss computation without any weight coefficients. By substituting the loss function in method d with the hybrid loss function, method e demonstrates superior performance over method d (the seventh column in Figure 9), indicating that the problem of class imbalance in crack images is effectively alleviated.

Method f: Finally, the multi-level label supervision training method is implemented to train method e. Through the combined effect of the improved SwinT-blocks, weighted attention module, hybrid loss function and multi-level label supervision training method, method f achieves the optimum segmentation performance, with the best evaluation metrics values.

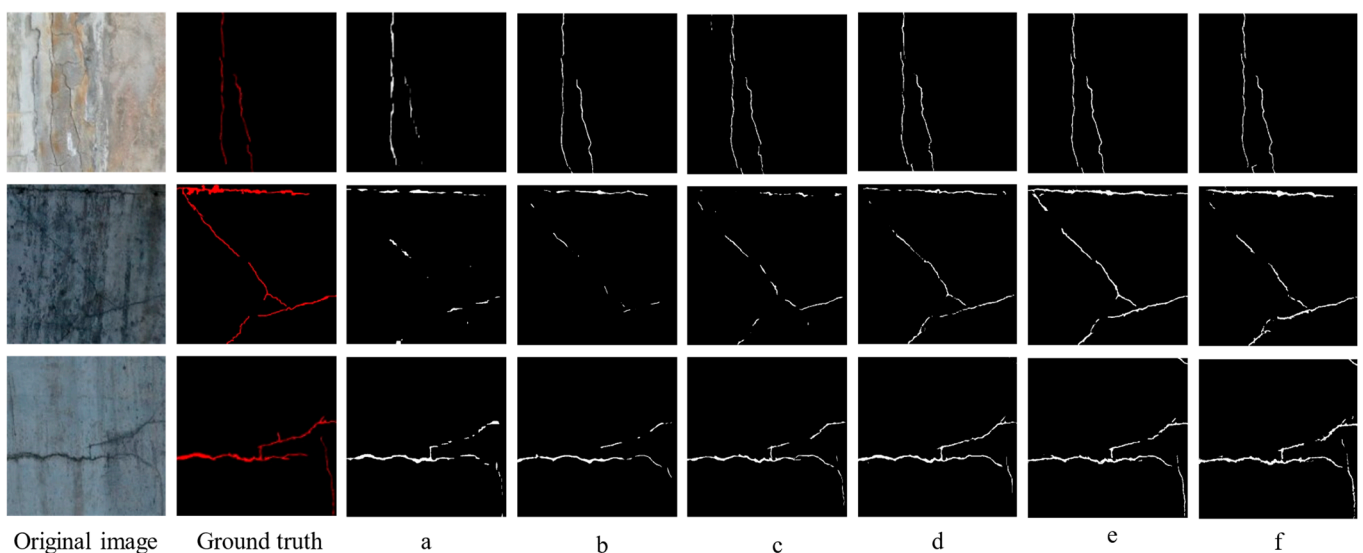


Figure 9. Segmentation results of ablation study of DamSCrack dataset.

Table 2. Methods in ablation study.

Method	Improved SwinT-Blocks	Weighted Attention Block	Hybrid Loss Function	Multiple-Level Label Supervision
a	×	×	×	×
b	✓	×	×	×
c	×	✓	×	×
d	✓	✓	×	×
e	✓	✓	✓	×
f(DCST-net)	✓	✓	✓	✓

Table 3. Evaluation metrics of methods in ablation study on DamSCrack dataset.

Method	Pre (%)	Rec (%)	F1s (%)	mIou (%)
a	68.16	73.46	69.10	54.42
b	72.66	74.26	72.02	57.66
c	72.94	75.15	72.40	58.08
d	69.80	86.40	76.43	62.84
e	77.81	81.73	79.48	66.65
f	78.41	82.02	79.96	67.21

In conclusion, the experimental results of the ablation study indicate that the crack segmentation network presented in this work, by progressively encoding crack features using the improved SwinT-blocks, enhances the global perception of cracks. In addition, the utilization of the weighted attention module to cross-integrate features from various levels of the encoder and decoder improves the segmentation of local crack parts. Moreover, the combined effects of the crack segmentation loss function and multi-level label supervision effectively alleviate the issue of class imbalance and optimize the network's weight parameters.

4.3. Comparative Study

The performance of the DCST-net is empirically validated on the DamSCrack dataset and compared with some classical segmentation models, such as the SegNet [50], FCN-8s [51], DeepLab v3+ [52], U-Net [53], and LR-ASPP [54]. Table 4 presents the quantitative comparison results of performance indices, and Figure 10 demonstrates some typical prediction outcomes generated by the DCST-net and other comparative methods.

Table 4. Evaluation metrics of comparison methods in DamSCrack dataset.

Method	Pre(%)	Rec(%)	F1s (%)	mIou (%)
SegNet [50]	61.05	58.87	57.53	42.26
FCN-8s [51]	54.90	71.17	61.98	44.91
DeepLab v3+ [52]	69.75	79.69	73.49	58.86
U-Net [53]	72.32	76.47	73.02	58.80
LR-ASPP [54]	54.15	66.76	58.18	42.11
DCST-net (ours)	78.41	82.02	79.96	67.21

It can be seen from Table 4 that SegNet [50] and LR-ASPP [54] exhibit the poorest segmentation outcomes for dam surface cracks, achieving an *mIoU* score of merely 42.26% and 42.11%, respectively. Obvious deficiencies are observed in their segmentation results (column 3 and 7 of Figure 10), such as incorrectly identifying many non-crack areas as crack areas and mis-detecting some of the crack regions. FCN-8s [51] delivers suboptimal segmentation results, with an *mIoU* of 44.91 and a high recall of 71.17%. While FCN-8s [54] can predict the rough outline of cracks, its performance on fine details remains unsatisfactory. DeepLab v3+ [52] and U-Net [53] offer improved segmentation results for performance

metrics, particularly achieving an *mIoU* score of 58.86% and 58.80%, respectively. From columns 5 and 6 of Figure 10, one can discern that these two approaches can effectively predict the crack backbone and provide relatively adequate, detailed information. These observations align well with the existing research literature [55–59], which highlights the superior accuracy of U-Net and DeepLabv3+ compared to other SOTA models (such as FCN and SegNet) in semantic segmentation tasks related to road pavement and concrete cracks. However, CNNs rely on convolutional and pooling layers to process input images, granting them translational invariance by uniformly filtering each patch. While this characteristic is essential for CNNs' effectiveness compared to fully connected networks in vision tasks, prioritizing local pixel connectivity compromises a global context [60,61]. Consequently, CNNs are susceptible to image distortions such as translation and scaling, making them less robust [62]. Hence, U-Net and DeepLabv3+ still face challenges in achieving continuous predictions and struggle in complex regions when dealing with the DamSCrack. On the contrary, ViTs are not limited by local pattern operations and can instead concentrate on information from various distances around the input target area [61]. Therefore, the proposed DCST-net equipped with a ViT-based encoder and decoder demonstrates superior segmentation results, presenting continuous and detailed outputs without the misidentification of crack segmentation. It outperforms the other five mainstream algorithms, as illustrated in the visual comparison figures (last column of Figure 10). In the quantitative comparisons presented in Table 4, the proposed DCST-net achieves superior scores across all performance metrics compared to those related to the classical algorithms. Specifically, its precision is 6.09% higher than that of U-Net [53], and its recall, F1 score, and *mIoU* are 2.33%, 6.47%, and 8.35% higher than those related to DeepLab v3+ [52], respectively. To sum up, in addressing the practical challenge of dam crack detection, the current mainstream crack segmentation algorithms appear unsuitable and cannot be directly applied.

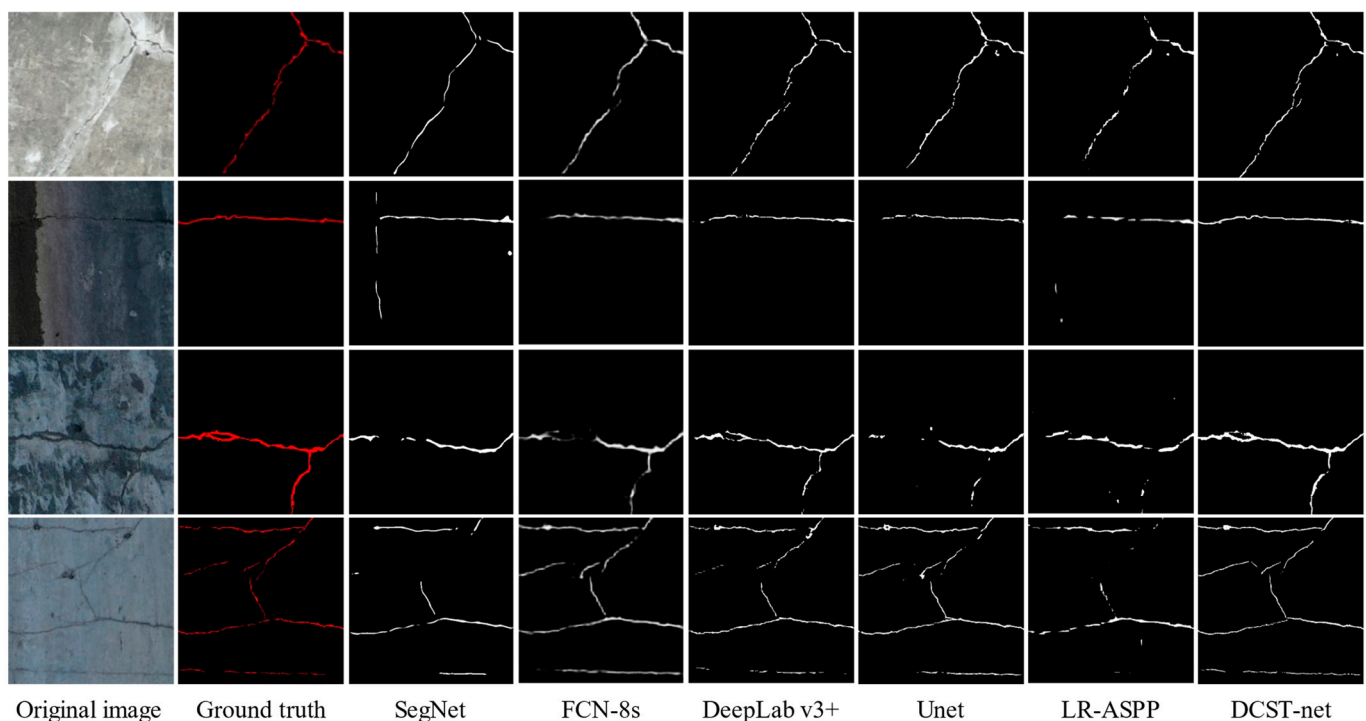


Figure 10. Segmentation results of comparison methods of DamSCrack dataset.

4.4. Generalization Study

To validate the generalizability and further demonstrate the effectiveness of the proposed DCST-net, comparison experiments were performed on openly accessible datasets, DeepCrack [48] and Crack500 [49], against five mainstream segmentation networks.

Comparative results on the DeepCrack dataset: Figure 11 illustrates the prediction results of some typical sample cracks of DeepCrack [48]. Evaluation metrics including inference time are presented in Table 5. It is observable that LR-ASPP [54] has the lowest crack detection accuracy, with an *mIoU* of 65.19%, but the fastest forward inference speed of 18.27 ms. U-Net [53] achieves more consistent performance across evaluation metrics, compared with those obtained by FCN8s [51] and SegNet [50]. Obviously, DeepLab v3+ [52] demonstrates the strongest crack segmentation performance on DeepCrack [48], achieving *Pre* of 84.87%, *Rec* of 82.89%, *F1s* of 82.46%, and *mIoU* of 71.41%, respectively, but with the longest inference time of 31.29 ms. The notable performance of DeepLabv3+ can be attributed to its extensive utilization of separable convolutions and spatial pyramid-pooling modules which enable the model to effectively capture multi-scale contextual information while maintaining a wide field of view [63]. However, the proposed DCST-net achieves a balance between performance and speed on the DeepCrack dataset. While its evaluation metrics are slightly lower than those of DeepLab v3+ [52], the forward inference time is approximately 30% faster than that of DeepLab v3+. This suggests that DeepLab v3+ prioritizes higher accuracy at the expense of reduced inference speed. Consequently, the proposed DCST-net achieves better overall performance compared to DeepLab v3+ in the dataset of DeepCrack.

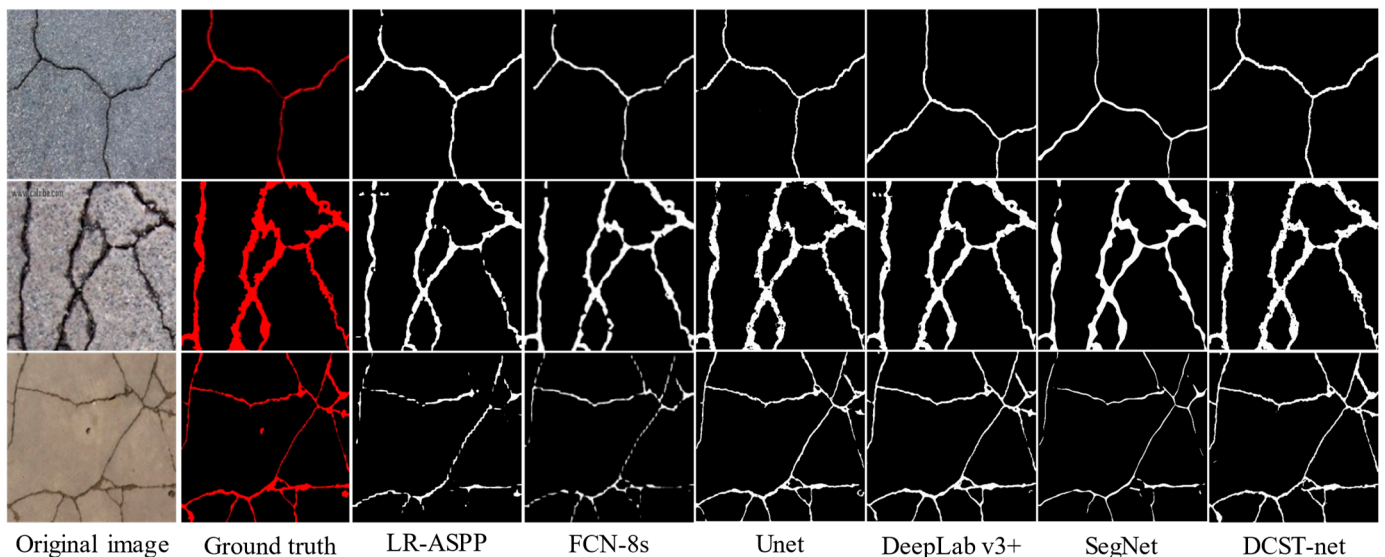


Figure 11. Segmentation results of comparison methods in DeepCrack dataset.

Table 5. Evaluation metrics of comparison methods in DeepCrack dataset.

Method	Pre (%)	Rec (%)	F1s (%)	mIoU (%)	Time (ms)
LR-ASPP [54]	81.61	77.34	77.79	65.19	18.27
FCN8s [51]	87.10	74.15	80.10	66.81	20.15
U-Net [53]	83.50	82.58	81.30	68.31	19.52
DeepLab v3+ [52]	84.87	82.89	82.46	71.41	31.29
SegNet [50]	82.71	73.57	75.93	62.98	18.17
DCST-net (ours)	84.64	81.31	80.36	69.46	19.77

Comparative results on the Crack500 dataset: Figure 12 presents the crack segmentation results derived by DCST-net and the comparison methods. It can be observed, in Figure 12, that LR-ASPP [54] achieves the poorest performance, while the detection effect of the DCST-net is better compared with that of comparison methods from visualization aspects. However, all the algorithms incorrectly predict non-crack areas as cracks (row 2 of Figure 12). This is attributed to the presence of image background noise that is challenging

even for the human eye to distinguish. As evidenced in Table 6, the DCST-net has achieved the best scores over most of the evaluation indices.

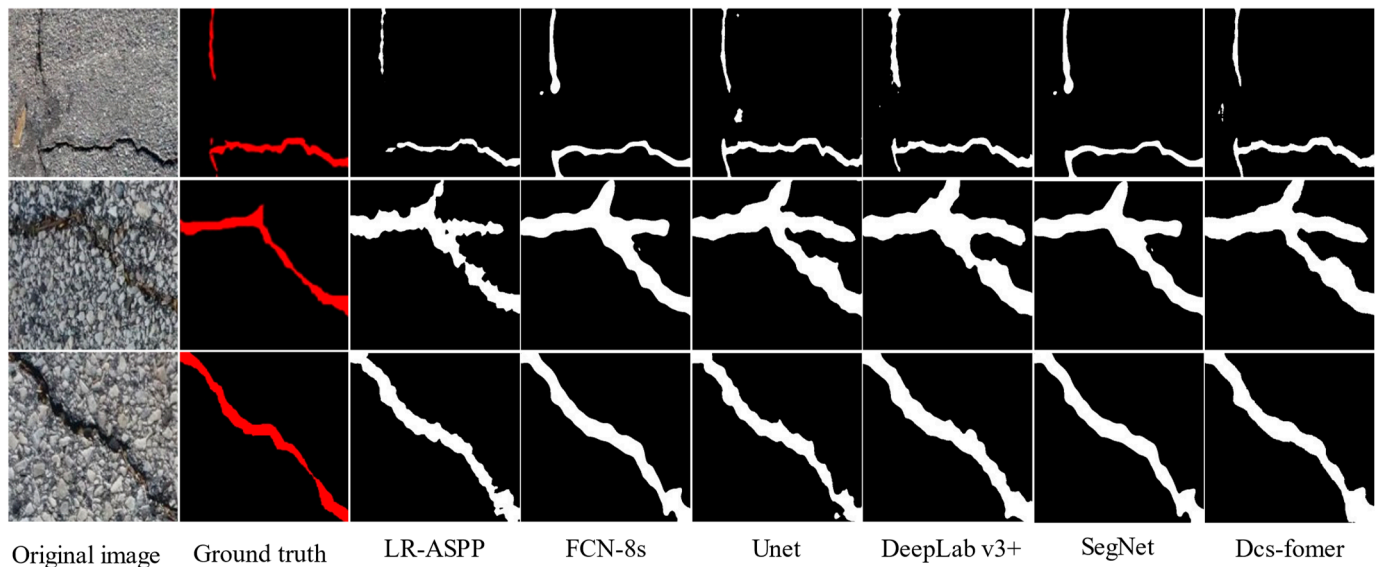


Figure 12. Segmentation results of comparison methods in Crack500 dataset.

Table 6. Evaluation metrics of comparison methods in Crack500 dataset.

Method	Pre (%)	Rec (%)	F1s (%)	mIou (%)	Time (ms)
LR-ASPP [54]	65.68	73.04	65.85	51.20	16.03
FCN8s [51]	65.44	79.21	69.24	54.83	19.97
U-Net [53]	60.98	80.05	67.03	52.10	18.88
DeepLab v3+ [52]	63.64	81.15	69.13	54.55	26.11
SegNet [50]	64.64	75.86	67.41	52.78	18.14
DCST-net (ours)	70.78	75.35	73.00	57.48	22.42

The typical sample cracks in DamSCrack, DeepCrack, and Crack500 are presented in the first column of Figures 10–12, respectively. Obviously, DamSCrack, collected from dam scenes, presents more challenges compared to the other two datasets. It includes weak cracks, diverse crack patterns, low contrast, and significant noise. In contrast, DeepCrack possesses relatively high contrast, making crack regions visually distinguishable even in complex scenes. The primary challenge in Crack500 is that some cracks closely resemble the background. From the evaluation metrics, it is evident that LR-ASPP [54], FCN8s [51], U-Net [53], and SegNet [50] perform adequately when the background noise is weak (Table 5), but exhibit relatively lower performance in the presence of strong background noise (Tables 4 and 6). In contrast, DCST-net and DeepLabv3+ demonstrate distinctive capabilities in extracting global contextual layers, resulting in higher precision in crack segmentation. However, due to dilated convolutions leading to the loss of very fine-grained information, DeepLabv3+ falls short of achieving the high accuracy attained by DCST-net equipped with a ViT-based encoder–decoder architecture in precisely discerning the extremely sharp boundaries of cracks and restoring the connectivity among crack pixels. It is worth mentioning that the proposed algorithm outperforms the other five advanced algorithms on these three datasets, even though it is specifically designed for dam surface crack segmentation.

5. Conclusions

In this paper, a pure ViT-based dam surface crack segmentation network, named DCST-net, is developed for automatic crack detection at pixel level to enhance dam assessment and maintenance in practical engineering applications. Specifically, the improved SwinT-block

serving as the basic component of the DCST-net is proposed for characterizing features and learning cross-channel and long-range semantic information. Additionally, to enhance segmentation performance and sharpen crack detection, a weighted attention block is incorporated to combine features between the symmetric encoder and decoder. Moreover, to address the challenge of class imbalance in crack segmentation, we propose a hybrid loss function combining Weighted Cross-Entropy loss and Dice loss. Furthermore, in the training process, we adopt a multi-level label supervision training method to enhance the utilization of features at different levels.

Based on the methodology and experimental findings presented by this work, several main conclusions can be outlined. (1) The comparison results for the self-built dam crack dataset demonstrate that our DCST-net outperforms mainstream methods such as SegNet [50], FCN-8s [51], DeepLab V3+ [52], U-Net [53], and LR-ASPP [54] for pixel-level crack segmentation. In particular, its precision surpasses that of U-Net by 6.09% [53], while its *Rec*, *F1s* and *mIoU* exceed those of DeepLab v3+ by 2.33%, 6.47%, and 8.35%, respectively [52]. This superiority is particularly evident when dealing with various categories of dam cracks, such as elongated, slender, and complex geometry cracks under heavy noise. These findings highlight the potential of the adoption of ViT-based approaches for dam crack detection at pixel level. (2) The comparison results of ablation study reveal that the improved strategies adopted in this work, including the improved SwinT-block, the weighted attention block, the hybrid loss function and the multiple-level label supervision training method, contribute to the performance enhancement of the model to varying degrees. (3) The experimental comparison results both on pavement cracks (Crack500 [49]) and concrete surface cracks (DeepCrack [48]) further demonstrate that the DCST-net possesses good generalization, which is a necessity for real-world applications [37].

The precise segmentation of crack width and the effective elimination of crack-like patterns pose significant challenges in crack detection automation [56,57,64]. Despite the proposed DCST-net demonstrating the best performance in both visualization and quantitative indicators across three crack datasets, it occasionally misidentifies defects or backgrounds as cracks under conditions characterized by extremely weak or significantly disturbed backgrounds. In future work, it is essential to consider the impact of other crack-like defects on crack detection to further enhance accuracy. One possible approach would be to create a more diverse and extensive dataset by collecting images from various dams worldwide to meet the requirements of ViTs [65]. In our study, due to the lack of a rigorous calibration procedure and the shaking of the drone during the collection of crack images, the quantification of segmented cracks cannot be achieved with a unified model. However, precise geometric features of dam cracks are crucial for monitoring and assessing dam safety. Additionally, adopting cost-effective, rapid, and highly precise online systems for crack segmentation in real-world scenarios is imperative. Therefore, developing an encoder–decoder architecture integrated with a Vision Transformer (ViT) capable of enabling real-time applications on edge devices is another research direction.

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

Funding: This research was funded by Sichuan Science and Technology Program, grant number 2022YFG0242, and the Open Fund of Robot Technology Used for Special Environment Key Laboratory of Sichuan Province, grant number 21kftk01.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [<https://github.com/yhleo/DeepCrack> (accessed on 9 May 2023)]; [<https://github.com/fyangneil/pavement-crack-detection> (accessed on 11 June 2023)].

Conflicts of Interest: Author Guochuan Zhao was employed by the company Sinograin Chendu Storage Research Institute Co., Ltd., Chendu, China. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Kang, F.; Li, J.; Zhao, S.; Wang, Y. Structural health monitoring of concrete dams using long-term air temperature for thermal effect simulation. *Eng. Struct.* **2019**, *180*, 642–653. [\[CrossRef\]](#)
2. Zhang, G.; Liu, Y.; Zheng, C.; Feng, F. Simulation of influence of multi-defects on long-term working performance of high arch dam. *Sci. China Technol. Sci.* **2011**, *54*, 1–8. [\[CrossRef\]](#)
3. Ye, X.W.; Jin, T.; Li, Z.X.; Ma, S.Y.; Ding, Y.; Ou, Y.H. Structural crack detection from benchmark data sets using pruned fully convolutional networks. *J. Struct. Eng.* **2021**, *147*, 04721008. [\[CrossRef\]](#)
4. Li, Y.; Bao, T.; Shu, X.; Gao, Z.; Gong, J.; Zhang, K. Data-driven crack behavior anomaly identification method for concrete dams in long-term service using offline and online change point detection. *J. Civ. Struct. Health* **2021**, *11*, 1449–1460. [\[CrossRef\]](#)
5. Hamishebahar, Y.; Guan, H.; So, S.; Jo, J. A comprehensive review of deep learning-based crack detection approaches. *Appl. Sci.* **2022**, *12*, 1374. [\[CrossRef\]](#)
6. Graham, W. *A Procedure for Estimating Loss of Life Caused by Dam Failure*; Bureau of Reclamation, Dam Safety Office: Denver, CO, USA, 1999; p. 10.
7. Rich, T.P. Lessons in social responsibility from the Austin dam failure. *Int. J. Eng. Educ.* **2006**, *22*, 1287–1296.
8. Chen, B.; Zhang, H.; Wang, G.; Huo, J.; Li, Y.; Li, L. Automatic concrete infrastructure crack semantic segmentation using deep learning. *Autom. Constr.* **2023**, *152*, 104950. [\[CrossRef\]](#)
9. Shi, P.; Shao, S.; Fan, X.; Zhou, Z.; Xin, Y. MCL-CrackNet: A Concrete Crack Segmentation Network Using Multi-level Contrastive Learning. *IEEE T. Instrum. Meas.* **2023**, *72*, 5030415. [\[CrossRef\]](#)
10. Bhowmick, S.; Nagarajaiah, S.; Veeraraghavan, A. Vision and deep learning-based algorithms to detect and quantify cracks on concrete surfaces from UAV videos. *Sensors* **2020**, *20*, 6299. [\[CrossRef\]](#)
11. Shi, P.; Fan, X.; Ni, J.; Wang, G. A detection and classification approach for underwater dam cracks. *Struct. Health Monit.* **2016**, *15*, 541–554. [\[CrossRef\]](#)
12. Fan, X.N.; Wu, J.J.; Shi, P.F.; Zhang, X.W.; Xie, Y.J. A Novel Automatic Dam Crack Detection Algorithm Based on Local-Global Clustering. *Multimed. Tools Appl.* **2018**, *77*, 26581–26599. [\[CrossRef\]](#)
13. Mohan, A.; Poobal, S. Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **2018**, *57*, 787–798. [\[CrossRef\]](#)
14. Cao, W.; Liu, Q.; He, Z. Review of Pavement Defect Detection Methods. *IEEE Access* **2020**, *8*, 14531–14544. [\[CrossRef\]](#)
15. Li, B.; Wang, K.; Zhang, A.; Yang, E.; Wang, G. Automatic classification of pavement crack using deep convolutional neural network. *Int. J. Pavement. Eng.* **2020**, *21*, 457–463. [\[CrossRef\]](#)
16. Zhang, J.; Bao, T. An improved resnet-based algorithm for crack detection of concrete dams using dynamic knowledge distillation. *Water* **2023**, *15*, 2839. [\[CrossRef\]](#)
17. Li, Y.T.; Bao, T.F.; Xu, B.; Shu, X.S.; Zhou, Y.H.; Du, Y.; Wang, R.J.; Zhang, K. A deep residual neural network framework with transfer learning for concrete dams patch-level crack classification and weakly-supervised localization. *Measurement* **2022**, *188*, 110641. [\[CrossRef\]](#)
18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [\[CrossRef\]](#)
19. Deng, J.; Lu, Y.; Lee, V.C.S. Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Comp. Aided Civ. Infrastruct. Eng.* **2020**, *35*, 373–388. [\[CrossRef\]](#)
20. Ciaparrone, G.; Serra, A.; Covito, V.; Finelli, P.; Scarpato, C.A.; Tagliaferri, R. A deep learning approach for road damage classification. In *Proceedings of Advanced Multimedia and Ubiquitous Engineering*; Springer: Singapore, 2018; pp. 655–661.
21. Xu, G.; Han, X.; Zhang, Y.; Wu, C. Dam crack image detection model on feature enhancement and attention mechanism. *Water* **2022**, *15*, 64. [\[CrossRef\]](#)
22. Ben, H.; Fei, K.; Yu, T. A real-time detection method for concrete dam cracks based on an object detection algorithm. *J. Tsinghua Univ.* **2023**, *63*, 1078–1086.
23. Li, Y.; Bao, T. A real-time multi-defect automatic identification framework for concrete dams via improved YOLOv5 and knowledge distillation. *J. Civ. Struct. Health Monit.* **2023**, *13*, 1333–1349. [\[CrossRef\]](#)
24. Zhang, J.; Zhang, J. An improved nondestructive semantic segmentation method for concrete dam surface crack images with high resolution. *Math. Probl. Eng.* **2020**, *2020*, 5054740. [\[CrossRef\]](#)
25. Pang, J.; Zhang, H.; Feng, C.; Li, L. Research on crack segmentation method of hydro-junction project based on target detection network. *KSCE J. Civ. Eng.* **2020**, *24*, 2731–2741. [\[CrossRef\]](#)
26. Feng, C.; Zhang, H.; Wang, H.; Wang, S.; Li, Y. Automatic pixel-level crack detection on dam surface using deep convolutional network. *Sensors* **2020**, *20*, 2069. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Chen, B.; Zhang, H.; Li, Y.; Wang, S.; Zhou, H.; Lin, H. Quantify pixel-level detection of dam surface crack using deep learning. *Meas. Sci. Technol.* **2022**, *33*, 065402. [\[CrossRef\]](#)

28. Kang, D.; Cha, Y. Efficient attention-based deep encoder and decoder for automatic crack segmentation. *Struct. Health Monit.* **2022**, *21*, 2190–2205. [[CrossRef](#)]
29. Lv, Z.; Tian, J.; Zhu, Y.; Li, Y. Automatic crack detection of dam concrete structures based on deep learning. *Comput. Concr.* **2023**, *32*, 615.
30. Li, J.; Lu, X.; Zhang, P.; Li, Q. Intelligent Detection Method for Concrete Dam Surface Cracks Based on Two-Stage Transfer Learning. *Water* **2023**, *15*, 2082. [[CrossRef](#)]
31. Wu, Z.; Tang, Y.; Hong, B.; Liang, B.; Liu, Y. Enhanced precision in dam crack width measurement: Leveraging advanced lightweight network identification for pixel-level accuracy. *Int. J. Intell. Syst.* **2023**, *2023*, 9940881. [[CrossRef](#)]
32. Zhu, Y.; Tang, H. Automatic damage detection and diagnosis for hydraulic structures using drones and artificial intelligence techniques. *Remote Sens.* **2023**, *15*, 615. [[CrossRef](#)]
33. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
34. Paul, S.; Chen, P.Y. Vision Transformers Are Robust Learners. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 22 February–1 March 2022; Volume 36, pp. 2071–2081.
35. Zhao, S.; Kang, F.; Li, J. Intelligent segmentation method for blurred cracks and 3D mapping of width nephograms in concrete dams using UAV photogrammetry. *Autom. Constr.* **2024**, *157*, 105145. [[CrossRef](#)]
36. Liu, H.; Miao, X.; Mertz, C.; Xu, C.; Kong, H. Crackformer: Transformer network for fine-grained crack detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 3783–3792.
37. Shamsabadi, E.A.; Xu, C.; Rao, A.S.; Nguyen, T.; Ngo, T.; Dias-da-Costa, D. Vision transformer-based autonomous crack detection on asphalt and concrete surfaces. *Autom. Constr.* **2022**, 104316. [[CrossRef](#)]
38. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
39. Huang, H.; Hao, X.; Pei, L.; Ding, J.; Hu, Y.; Li, W. Automated detection of through-cracks in pavement using three-instantaneous attributes fusion and Swin Transformer network. *Autom. Constr.* **2024**, *158*, 105179. [[CrossRef](#)]
40. Sun, Z.; Zhai, J.; Pei, L.; Li, W.; Zhao, K. Automatic Pavement Crack Detection Transformer Based on Convolutional and Sequential Feature Fusion. *Sensors* **2023**, *23*, 3772. [[CrossRef](#)] [[PubMed](#)]
41. Luo, H.; Li, J.; Cai, L.; Wu, M. STrans-YOLOX: Fusing swin transformer and YOLOX for automatic pavement crack detection. *Appl. Sci.* **2023**, *13*, 1999. [[CrossRef](#)]
42. Guo, F.; Qian, Y.; Liu, J.; Yu, H. Pavement crack detection based on transformer network. *Autom. Constr.* **2023**, *145*, 104646. [[CrossRef](#)]
43. Guo, F.; Liu, J.; Lv, C.; Yu, H. A novel transformer-based network with attention mechanism for automatic pavement crack detection. *Constr. Build. Mater.* **2023**, *391*, 131852. [[CrossRef](#)]
44. Zhang, E.; Shao, L.; Wang, Y. Unifying transformer and convolution for dam crack detection. *Autom. Constr.* **2023**, *147*, 104712. [[CrossRef](#)]
45. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-unet: Unet-like pure transformer for medical image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 205–218.
46. Ozan, O.; Jo, S.; Loic, L.F.; Matthew, L.; Mattias, H.; Kazunari, M.; Kensaku, M.; Steven, M.; Nils, Y.H.; Bernhard, K.; et al. Attention u-net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.03999.
47. Zhao, F.; Chao, Y.; Li, L. A Crack Segmentation Model Combining Morphological Network and Multiple Loss Mechanism. *Sensors* **2023**, *23*, 1127. [[CrossRef](#)]
48. Liu, Y.; Yao, J.; Lu, X.; Renping, X.; Li, L. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing* **2019**, *338*, 139–153. [[CrossRef](#)]
49. 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](#)]
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](#)]
51. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
52. Chen, L.C.; Papandreo, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587.
53. 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), Munich, Germany, 5–9 October 2015; pp. 234–241.
54. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Adam, H. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
55. Dung, C.V. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58.

56. Dais, D.; Bal, I.E.; Smyrou, E.; Sarhosis, V. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Autom. Constr.* **2021**, *125*, 103606.
57. Hsieh, Y.-A.; Tsai, Y.J. Machine learning for crack detection: Review and model performance comparison. *J. Comput. Civ. Eng.* **2020**, *34*, 04020038. [[CrossRef](#)]
58. Alipour, M.; Harris, D.K.; Miller, G.R. Robust pixel-level crack detection using deep fully convolutional neural networks. *J. Comput. Civ. Eng.* **2019**, *33*, 04019040. [[CrossRef](#)]
59. Liu, Z.; Cao, Y.; Wang, Y.; Wang, W. Computer vision-based concrete crack detection using U-net fully convolutional networks. *Autom. Constr.* **2019**, *104*, 129–139. [[CrossRef](#)]
60. Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F.A.; Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv* **2018**, arXiv:1811.12231.
61. Tuli, S.; Dasgupta, I.; Grant, E.; Griffiths, T.L. Are Convolutional Neural Networks or Transformers more like human vision? *arXiv* **2021**, arXiv:2105.07197.
62. Azulay, A.; Weiss, Y. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv* **2018**, arXiv:1805.12177.
63. 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. 833–851.
64. Mei, Q.; Gül, M.; Azim, M.R. Densely connected deep neural network considering connectivity of pixels for automatic crack detection. *Autom. Constr.* **2020**, *110*, 103018. [[CrossRef](#)]
65. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jegou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; Volume 139, pp. 10347–10357.

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.