buildings

MDPI

Article

CGV-Net: Tunnel Lining Crack Segmentation Method Based on Graph Convolution Guided Transformer

Kai Liu [1,2,†,‡], Tao Ren [1,2,*,†], Zhangli Lan [1,*,†], Yang Yang [2], Rong Liu [3] and Yuantong Xu [1]

[1] School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China; 622230830024@mails.cqjtu.edu.cn (K.L.); 622220070018@mails.cqjtu.edu.cn (Y.X.)
[2] China Railway Changjiang Transport Design Group Co., Ltd., Chongqing 400067, China; yangyang87@crecg.com
[3] Institute of Future Civil Engineering Science and Technology, Chongqing Jiaotong University, Chongqing 400074, China; cqjtu_liurong@cqjtu.edu.cn
* Correspondence: rentao3@crecg.com (T.R.); lzl@cqjtu.mails.edu (Z.L.)
† These authors contributed equally to this work.
‡ "School of Information Science and Engineering, Chongqing Jiaotong University" and "China Railway Changjiang Transport Design Group Co., Ltd." are listed as co-first institutions.

Abstract: Lining cracking is among the most prevalent forms of tunnel distress, posing significant threats to tunnel operations and vehicular safety. The segmentation of tunnel lining cracks is often hindered by the influence of complex environmental factors, which makes relying solely on local feature extraction insufficient for achieving high segmentation accuracy. To address this issue, this study proposes CGV-Net (CNN, GNN, and ViT networks), a novel tunnel crack segmentation network model that integrates convolutional neural networks (CNNs), graph neural networks (GNNs), and Vision Transformers (ViTs). By fostering information exchange among local features, the model enhances comprehension of the global structural patterns of cracks and improves inference capabilities in recognizing intricate crack configurations. This approach effectively addresses the challenge of modeling contextual information in crack feature extraction. Additionally, the Detailed-Macro Feature Fusion (DMFF) module enables multi-scale feature integration by combining detailed and coarse-grained features, mitigating the significant feature loss encountered during the encoding and decoding stages, and further improving segmentation precision. To overcome the limitations of existing public datasets, which often feature a narrow range of crack types and simplistic backgrounds, this study introduces TunnelCrackDB, a dataset encompassing diverse crack types and complex backgrounds.Experimental evaluations on both the public Crack dataset and the newly developed TunnelCrackDB demonstrate the efficacy of CGV-Net. On the Crack dataset, CGV-Net achieves accuracy, recall, and F1 scores of 73.27% and 57.32%, respectively. On TunnelCrackDB, CGV-Net attains accuracy, recall, and F1 scores of 81.15%, 83.54%, and 82.33%, respectively, showcasing its superior performance in challenging segmentation tasks.

Keywords: tunnel crack segmentation; DMFF; vision transformer; CGV-Net

Academic Editor: Erwin Oh

Received: 10 December 2024
Revised: 28 December 2024
Accepted: 8 January 2025
Published: 10 January 2025

Citation: Liu, K.; Ren, T.; Lan, Z.; Yang, Y.; Liu, R.; Xu, Y. CGV-Net: Tunnel Lining Crack Segmentation Method Based on Graph Convolution Guided Transformer. *Buildings* **2025**, *15*, 197. https://doi.org/10.3390/buildings15020197

1. Introduction

Tunnel engineering represents a cornerstone of transportation infrastructure development and has been widely applied in highways, railways, and metro construction. Among the various forms of tunnel distress [1], lining cracking emerges as the most frequently encountered issue. These cracks accelerate the aging and deterioration of tunnel structures, posing significant challenges to their longevity and operational safety [2]. Consequently,

the maintenance and timely detection of tunnel lining cracks are of paramount importance for ensuring tunnel safety and extending their service life [3].

The segmentation of tunnel lining cracks is often hindered by challenges arising during data acquisition, such as uneven artificial lighting and other unpredictable factors. Lining cracks are predominantly slender, typically manifesting as circumferential or longitudinal cracks, and are mainly concentrated in higher structural areas such as the arch crown. These characteristics make manual segmentation of cracks a labor-intensive task. Unlike road or bridge cracks, tunnel lining cracks often lead to water seepage through the tunnel lining, further complicating the detection process [4]. Additionally, the image acquisition of tunnel lining cracks is highly susceptible to variations in lighting conditions, which increases the complexity of image segmentation. These challenges significantly affect the accuracy and recall rate of tunnel lining crack segmentation, necessitating more robust and adaptive segmentation approaches.

Current methods for detecting and segmenting tunnel lining cracks are predominantly manual, relying on visual inspection by technicians or the use of basic equipment to observe the location and size of cracks [5]. Alternatively, cracks are identified and documented one by one from tunnel lining images. However, such manual approaches are time-consuming, labor-intensive, and prone to human error, leading to inaccuracies and inconsistencies in segmentation results. This highlights the need for automated and more reliable segmentation techniques to address these limitations effectively.

With the continuous advancement of computer science, researchers have applied traditional algorithms to tunnel crack segmentation tasks [6]. For instance, Marco et al. [7] employed a two-dimensional Fourier transform as a preprocessing step to enhance crack segmentation performance. Jiang et al. [8] utilized a threshold-adaptive algorithm combined with the Canny edge detector to segment tunnel lining cracks. Long et al. [9] developed and integrated a novel stripe module to simulate long-range contextual dependencies, improving segmentation of long and narrow cracks. Lei et al. [10] proposed a differentiated noise filter and an improved segmentation method combining adaptive segmentation and thresholding techniques, achieving enhanced segmentation accuracy. While traditional algorithms have demonstrated promising performance in segmenting tunnel lining cracks, they often require extensive feature engineering and rely heavily on specific image processing steps. These requirements impose significant limitations on the flexibility and efficiency of such models, hindering their ability to extract features accurately. Consequently, traditional image processing methods exhibit poor robustness against noise, illumination variations, and complex backgrounds, making them less effective for crack recognition in challenging scenarios. Moreover, the need for manual parameter tuning further limits their practicality, rendering them inadequate for meeting the demands of tunnel lining crack segmentation in real-world applications.

With the continuous advancement of deep learning techniques, a surge of research has emerged to address the limitations of traditional image processing in crack segmentation tasks for tunnel linings. These studies have leveraged deep learning frameworks to significantly enhance segmentation accuracy and efficiency. Raja et al. [11] proposed a novel hybrid intelligent system that combines the Grey Wolf Optimization (GWO) algorithm with artificial neural networks to enhance the application of artificial intelligence in geotechnical engineering, inspiring other researchers to apply it to tunnel lining segmentation studies.Zhao et al. [12] improved PANet by incorporating the Growth Region Algorithm (GRA), effectively enhancing segmentation precision. Wang et al. [13] proposed the TunnelURes network by modifying the U-Net architecture, replacing the encoder with a ResNet-152 model to achieve high performance and efficient crack segmentation. Razveeva et al. [14] utilized U-Net and LinkNet architectures for concrete crack segmentation, with a static-

enhanced LinkNet showing the best segmentation performance. Zou et al. [15] introduced the DeepCrack network, based on SegNet's encoder structure, which fused multi-scale feature maps learned by various convolutional layers, significantly improving segmentation accuracy. Additionally, Zhao et al. [16] developed the Position-Channel Network (PCNet) by integrating a newly designed channel and positional attention module into the U-Net framework, addressing discontinuities in the channel and spatial dimensions of cracks for more refined segmentation. Lin et al. [17] proposed DeepCrackAT, a multi-scale crack feature learning model that incorporated convolutional attention modules into the encoder, enhancing crack perception and segmentation accuracy. Chen et al. [18] introduced an improved Swin-Unet model with a skip attention module and residual Swin Transformer blocks, assigning greater weight to crack feature channels and achieving substantial advancements in segmentation performance. Qin et al. [19] proposed a ViT-based defect segmentation model, employing an adapter and decoding head to enhance encoding capacity and improve the model's crack feature learning capabilities. Zhou et al. [20] presented MC-TLD, a segmentation algorithm leveraging multi-scale attention and contextual information to improve segmentation accuracy. Tao et al. [21] proposed a novel convolutional transformation network and designed the Dilated Residual Block (DRB) and Boundary-Aware Module (BAM). The DRB focuses on local detail features, and by combining the DRB with a lightweight Transformer to capture global information, it outperforms traditional algorithms in terms of performance. Other efforts include Pu et al. [22], who applied deep convolutional networks (DCNNs) and an improved encoder–decoder for semantic segmentation tasks, achieving better overall performance. Wang et al. [23] enhanced U-Net with a dilated spatial pyramid and an initial module, attaining high F1 scores and accuracy in crack instance segmentation. Kang et al. [24] developed a method using an improved Tuff algorithm to refine Faster R-CNN outputs for crack segmentation, though errors persisted in complex scenarios. Huang et al. [25] utilized the Mask R-CNN model for semantic segmentation of tunnel lining cracks, incorporating morphological closure operations, though challenges in segmentation accuracy remain unresolved. Despite these advancements, many models suffer from a lack of information exchange among local features, leading to suboptimal segmentation performance in complex crack backgrounds. Furthermore, the loss of feature layer information during the encoding and decoding stages in deep learning models exacerbates this issue, ultimately affecting segmentation accuracy. This underscores the need for more sophisticated models to address these challenges effectively.

This study proposes a Vision Transformer (ViT)-guided tunnel lining crack segmentation module, termed the CGV Module (CNN, GNN, and ViT Networks), which leverages graph neural networks (GNNs) to enhance the exchange of local information within feature layers. This design improves the network's ability to capture global crack features, thereby enhancing segmentation accuracy in complex backgrounds.Additionally, the Detailed-Macro Feature Fusion (DMFF) module integrates detailed-grained and coarse-grained feature maps through various fusion operations. This approach mitigates information loss during the encoding and decoding stages of the network, further improving the model's segmentation accuracy and recall rate.

The main contributions of this paper are as follows:

1. This paper proposes a graph neural network-guided Vision Transformer (ViT) tunnel crack segmentation module, CGV, designed for modeling tunnel lining cracks. The module is capable of simultaneously learning both local crack features and contextual information, thereby enhancing the accuracy of tunnel lining crack structure modeling. This approach further improves the performance of crack recognition and segmentation, leading to more precise and effective results in complex tunnel environments.

2. This paper proposes a graph-based representation method for tunnel lining cracks, which facilitates data interaction between different local regions by learning the features from the final layer of the encoder. This approach enhances the model's reasoning capability for complex crack structures, improving the accuracy and robustness of crack segmentation in challenging environments.
3. This paper proposes a multi-scale Detailed-Macro Feature Fusion (DMFF) module, which performs different feature fusion operations on feature layers of varying scales. This approach effectively compensates for the loss of critical data during the encoding and decoding stages, further enhancing the accuracy and robustness of crack segmentation.
4. A comprehensive dataset covering various crack types and complex backgrounds in operational tunnel linings has been constructed. This dataset is designed to provide a richer and more diverse set of training and testing samples for tunnel crack recognition and segmentation tasks. The aim is to enhance the model's generalization ability in complex scenarios, improving its performance in real-world applications.

2. Method

2.1. CGV-Net

This paper presents the CGV-Net segmentation model, which utilizes five layers of encoding operations. After encoding, the feature map from the final layer is passed into the CGV module, which enhances local feature information exchange. The feature map processed by the CGV module then undergoes five layers of decoding operations. The DMFF module is used to fuse different feature maps to compensate for feature loss during the encoding and decoding stages, generating a segmented binary image, as shown in Figure 1.

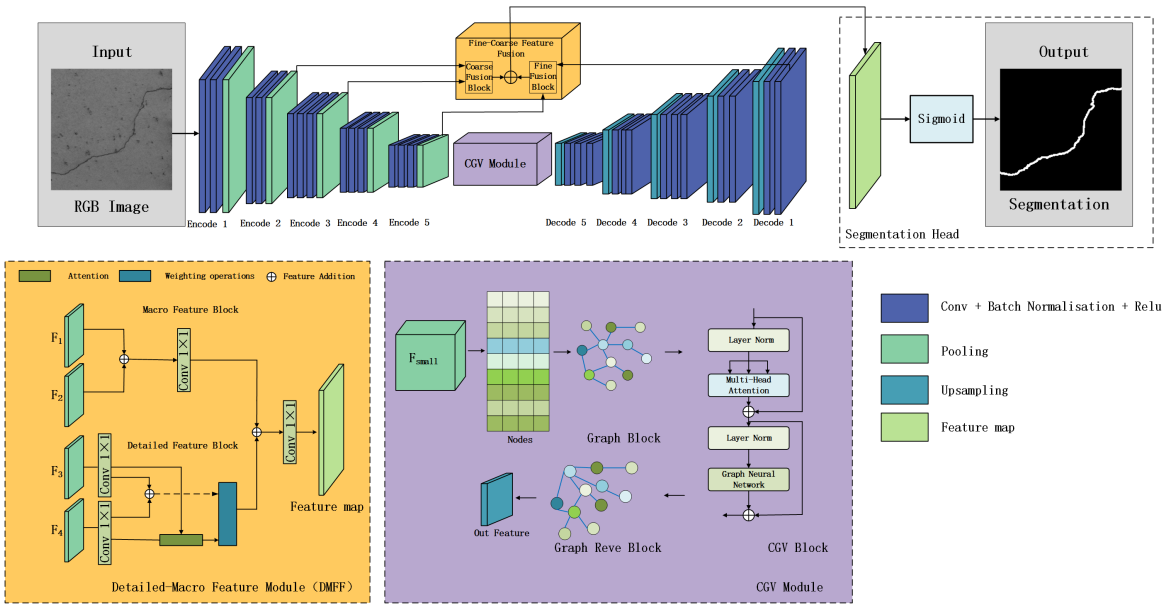

Figure 1. CGV net network model.

The CGV Module coordinates the relationship between graph neural networks (GNNs), convolutional neural networks (CNNs), and Vision Transformer (ViT), integrating the strengths of these three components to facilitate complex feature learning for tunnel lining crack image inference. It establishes interaction between local information and captures the global structural patterns of cracks, enabling global context awareness and enhancing the model's ability to extract crack information in complex backgrounds.

The DMFF module performs feature fusion across different scale feature layers. This module merges coarse and detailed features using an attention mechanism, integrating multi-scale feature information to compensate. This process significantly improves segmentation accuracy.

At the end of the network's encoding phase, the CGV module operates on the encoded features to enhance information exchange between local features. The DMFF module performs feature fusion on the decoded features and other feature layers at the end of the decoding phase, preventing the loss of crack feature information during both encoding and decoding stages.

2.2. Neural Network Guided Vision Transformer

In traditional convolutional neural networks (CNNs), the focus is primarily on local features, and the relationships between different local regions are limited, resulting in suboptimal performance for tunnel crack segmentation tasks. To address this limitation, the CGV module is proposed, which leverages graph neural networks (GNNs) to guide the Vision Transformer (ViT) for segmentation. This approach enhances the connectivity between different local features, enabling more complex pattern reasoning and improving the model's ability to learn intricate crack details.

2.2.1. The Structural Construction of the Diagram

This paper proposes a graph-based tunnel crack representation method, which takes the output feature map from the final layer of the encoder in the backbone network as the minimal-sized feature map. In traditional convolutional neural networks (CNNs), decoding operations typically follow the encoder to reconstruct crack features. However, due to the lack of information exchange between different regions, the overall image segmentation performance is affected. While each region contains rich local information, this locality limits the model's comprehensive understanding of the geometric structure of tunnel lining cracks. To address this limitation, the model treats each pixel block in the image as a node in the graph neural network, with the attributes of each pixel block serving as the features of the graph node. This enables the transmission of information between local regions, facilitating reasoning about complex geometric shapes and ultimately improving the segmentation performance of tunnel lining cracks.

The feature map F_{small} with dimensions $W \times S \times H$ is transformed into a node feature matrix D with dimensions $|D| \times C$, where $|D| = H \times W$. The matrix F_{small} represents the set of nodes in the graph, as shown in Figure 2. D can be expressed as the following formula:

$$D = [d_1, d_2, d_3, \ldots d_{|D|}]^T, D \in R^{|D| \times C} \tag{1}$$

d_i represents the feature vector of node i, and C represents the number of channels of F_{small}.

The model employs a self-attention mechanism to establish relationships between graph nodes, with edges representing the connections between nodes. The attention coefficient r_{ij} between nodes i and j is computed using the self-attention mechanism, which quantifies the importance of the relationship between these nodes. The attention coefficient r_{ij} is calculated by applying a linear transformation to the features of nodes i and j, followed by normalization of the attention coefficients across all neighboring nodes. This results in the final normalized attention distribution, expressed as

$$r_{ij} = y(d_i || d_j), y \in R^{|D| \times C} \tag{2}$$

$$r_{ij} = \frac{exp(r_{ij})}{\sum_{k \in N_i} exp(r_{ik})} \tag{3}$$

The generated attention matrix, composed of the attention coefficients r_{ij}, is denoted as $M_{atten} = [r_{ij}]_{|D| \times |D|}$, which can be viewed as the adjacency matrix of the graph. In other words, the attention coefficient r_{ij} represents the weight of the relationship between node i and node j.Through this self-attention mechanism, the model not only establishes the connectivity between graph nodes but also assigns different importance weights to the edges connecting these nodes. These attention weights reflect the strength of information flow between nodes, allowing the model to emphasize more important connections while diminishing the influence of less relevant ones. This dynamic weighting mechanism enables more effective and context-sensitive feature propagation across the graph.

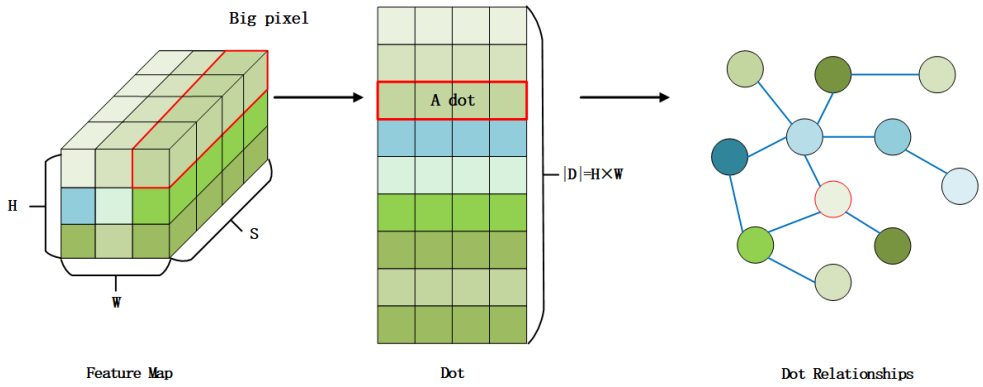

Figure 2. Dot feature initialization.

2.2.2. Graph Reasoning

To better capture the structural information of tunnel crack images, the CGV module employs a multi-head attention mechanism, as shown in Figure 3. The graph structure inference process of the CGV module can be expressed as follows:

$$x_l = M_{atten} \times D \tag{4}$$

$$Q = W_Q LN(x_l) \tag{5}$$

$$K = W_K LN(x_l) \tag{6}$$

$$V = W_V LN(x_l) \tag{7}$$

The matrix x_l represents the feature layer generated through the graph adjacency matrix weights. The matrices W_Q, W_K, and W_V are three learnable parameter matrices used for training. After the input matrix x_l undergoes Layer Normalization (denoted as $LN()$), it is multiplied with W_Q, W_K, and W_V to obtain the query (Q), key (K), and value (V) matrices, which are then used in the multi-head attention mechanism.

$$x_l^{'} = ReLU(Attention(Q, K, V)) + x_l, l = 1 \ldots L \tag{8}$$

$$x_l = GNN(LN(x_l^{'})) + x_l^{'}, l = 1 \ldots L \tag{9}$$

$$X_G = Reve(x_l) \tag{10}$$

Here, L represents the number of layers in the CGV module, and $Attention()$ denotes the multi-head attention mechanism. To obtain a feature map X_G that is compatible with the input to the CGV module, the inverse projection function $Reve()$ is applied. This results in the feature map, which serves as the output of the CGV module, as shown in Figure 3.

In each layer of the graph neural network (GNN), the attention weights are multiplied by different weight matrices, and the resulting vectors are combined through a concatenation operator to form new feature vectors.

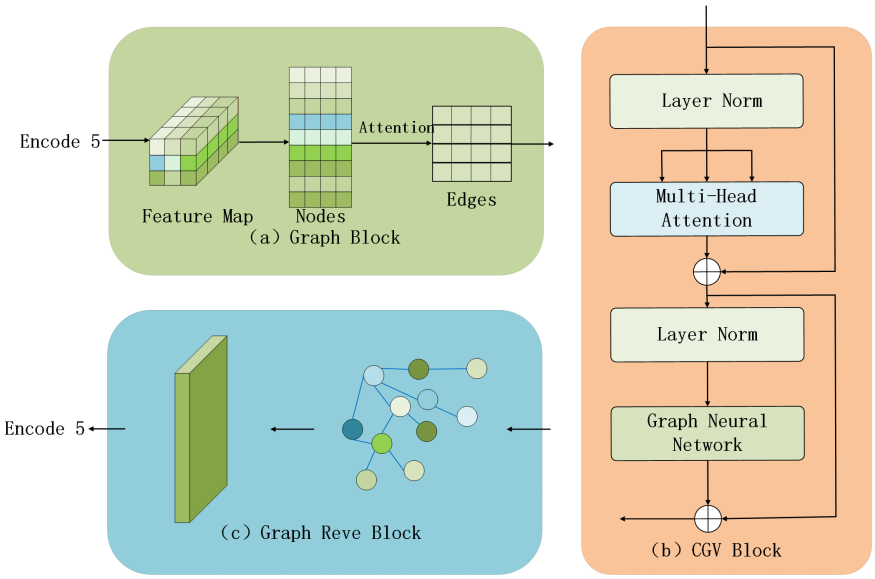

Figure 3. Structure and process flow. They should be listed as: (**a**) description of what is Graph Block in the first panel; (**b**) description of what is CGV Block in the second panel; and (**c**) description of what is Graph Block Reve Block in the third panel.

In the graph neural network (GNN), the features of each node are propagated to its neighboring nodes through the edges, enabling the node to gather information from its neighbors. This process allows different local regions of the feature map F_{small} to exchange information, enhancing the model's ability to capture fine details as well as to understand the global context, thereby improving the accuracy of crack segmentation. Formally, this propagation can be represented by the Equations (9) and (10).

2.2.3. Multi-Scale Feature Fusion

During the encoding and decoding phases of the network, the loss of critical information may occur, adversely affecting the segmentation of tunnel lining cracks. To mitigate the loss of feature information during these stages, a DMFF module is introduced at the end of the decoding phase. This module integrates an attention mechanism, effectively merging features at different scales, thereby enhancing the segmentation accuracy of the network.

The DMFF module consists of two components: the DFB (Detailed Fuse Block) and the MFB (Macro Fuse Block), as illustrated in Figure 4. For the MFB module, the operation is performed between the F_1 feature layer (the output feature of Encode 2) and the F_2 feature layer (the output feature of Encode 3), as represented by the following formula:

$$X_{MFB} = Conv_{1\times1}(F_1 + F_2) \tag{11}$$

X_{MFB} denotes the feature map output by the MFB (Macro Fuse Block) module. $Conv_{1\times1}()$ represents the operation of applying a 1×1 convolution to add the F_1 and F_2 feature maps. The resulting feature layer is then subjected to a 1×1 convolution, producing the fused feature map, X_{MFB}, from the coarse module.

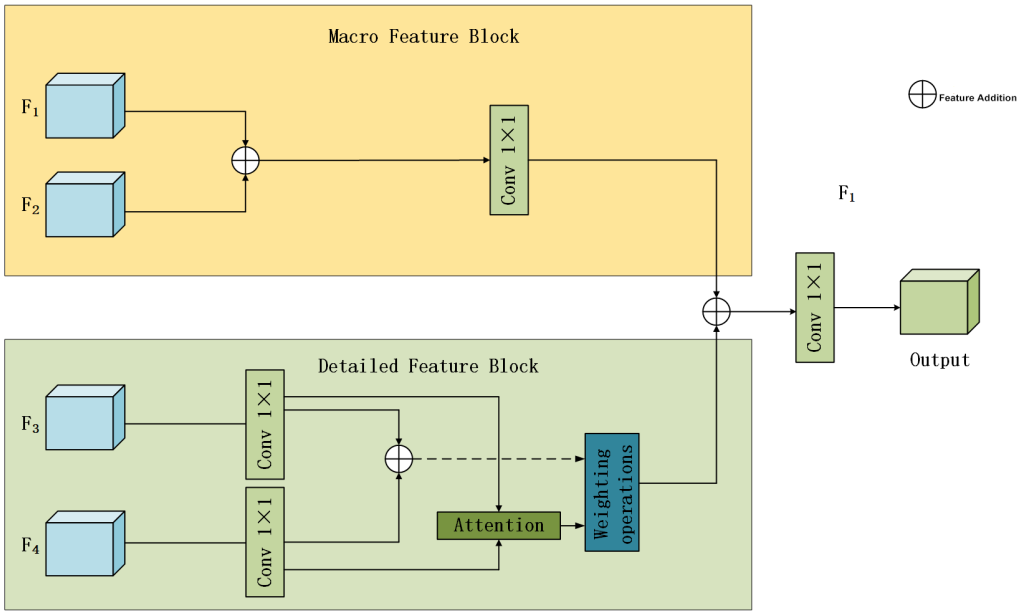

Figure 4. DMFF process structure diagram.

For the detailed module, operations are performed between the F_3 feature map (the output of Encode5) and the F_4 feature map (the output of Decode1). The specific operation process is outlined in Equations (15) to (16).

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{12}$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu)^2 \tag{13}$$

$$BN(x_i) = \gamma \left(\frac{x_i - \mu(x_i)}{\sqrt{\sigma(x_i)^2 + \epsilon}} \right) + \beta \tag{14}$$

$$Attw = attention(BN(Conv_{1\times1}(F_3)) + BN(Conv_{1\times1}(F_4))) \tag{15}$$

$$X_{DFB} = BN(Attw \times Conv_{1\times1}(F_3) + (1 - Attw) \times Conv_{1\times1}(F_4)) \tag{16}$$

$BN()$ represents Batch Normalization, The input feature maps of each layer are processed to reduce the distribution differences across layers, ensuring stable distribution for each layer. The $BN(\cdot)$ operation is represented by the equations from (12) to (14), where m denotes the batch size, and γ and β are trainable parameters. while $attention()$ denotes the computation of attention mechanism weights. $Attw$ refers to the weight matrix generated by the linear attention mechanism. X_{DFB} denotes the feature map output by the DFB (Detailed Fuse Block) module. The F_3 and F_4 feature maps undergo 1×1 convolutions individually, followed by a fusion operation. The resulting feature map is processed by the linear attention mechanism to obtain the attention weights. The 1×1 convoluted feature maps of F_3 and F_4 are then fused using the attention weights, producing the required feature map for the DFB module.

$$Sigmiod = \frac{1}{1 + e^{-x}} \tag{17}$$

$$F = Sigmiod(Conv_{1\times1}(X_{MFB} + X_{DFB})) \tag{18}$$

Through the MFB and DFB modules, different features from the network are fused. After fusion, the SegHead segmentation head, utilizing a 1×1 convolution, generates

the coarse-fine fused features. Equation (17) represents the application of the Sigmoid function to convert the output of each pixel into a probability value, and a threshold is used to convert the probability into the actual class label. Equation (18) represents applying the Sigmoid activation function to the generated fused features to obtain the tunnel crack segmentation map.

3. Experimental Setup

3.1. Experimental Environment

The experiments were implemented using the PyTorch framework on a Windows server. The server is equipped with an NVIDIA RTX 4070 Ti GPU (16 GB VRAM), and the operating system is Windows 11. The Python version used is 3.8.19, PyTorch version 1.9.1, and CUDA version 11.1. The learning rate was set to 1×10^{-4}, and the batch size for data processing was 16. The number of epochs is set to 200, with an early stopping strategy of 6 during the training process, and the Adam optimizer is used for parameter optimization.

3.2. Dataset

The Crack public dataset was collected from a tunnel, with images having a resolution of 512 × 512 pixels, and a total of 919 images. The dataset was split into training, validation, and test sets in a ratio of 8:1:1. The training set was used to train the model, the validation set was used for real-time evaluation of the model's training performance, and the test set was used for final evaluation of the model.

The custom-built dataset, TunnelCrackDB, consists of data collected from multiple operational tunnels of varying lengths in the Sichuan–Chongqing region. The data collection was conducted by the China Railway Changjiang Traffic Design Group Co., Ltd. This dataset contains 982 images (see Figure 5) of tunnel lining cracks captured under different lighting conditions. Different lighting conditions may affect the model's crack segmentation accuracy and recall rate, especially under complex lighting backgrounds. At the same time, the color and width of cracks in tunnel linings are influenced by the external environment. For example, cracks exposed to a humid environment may exhibit different colors and surface textures. We have incorporated crack images with seepage backgrounds into our custom-built dataset for model training. with a resolution of 512 × 512 pixels (see Table 1). The images are stored in PNG format, with an average size of 120 KB. The dataset includes both regular cracks (such as longitudinal cracks, circumferential cracks, and oblique cracks) and irregular cracks (such as networks of cracks and intersecting cracks) [26], as well as images of cracks in tunnel linings with complex backgrounds. The display of longitudinal cracks extends along the length of the structure and is often caused by structural joints or stress. The display of circumferential cracks extends along the circumference and is typically associated with stress concentration. The display of oblique cracks is related to structural forces or shape characteristics. The display of craze cracks refers to fine cracks caused by material shrinkage. The display of a network of cracks is a complex crack pattern caused by stress conditions or aging. The display of intersecting cracks results from the combined effect of multiple stresses or material defects. Understanding these displays is crucial for assessing structural health and formulating appropriate repair strategies, as shown in Figure 5. In the future, we will incorporate data that includes factors such as the presence of spider webs in the images, increase the resolution, and use wide-angle shots to capture long tunnel lining cracks. The data were manually annotated using the open-source software Labelme v4.5.13, with crack pixels labeled as 1 (white) and non-crack pixels labeled as 0 (black). A professional technician typically takes 7–10 min to annotate each 512 × 512 pixel image. The dataset was divided into training and test sets in a 1:9 ratio,

with 10% of the training set used as a validation set to evaluate model performance during the training phase.

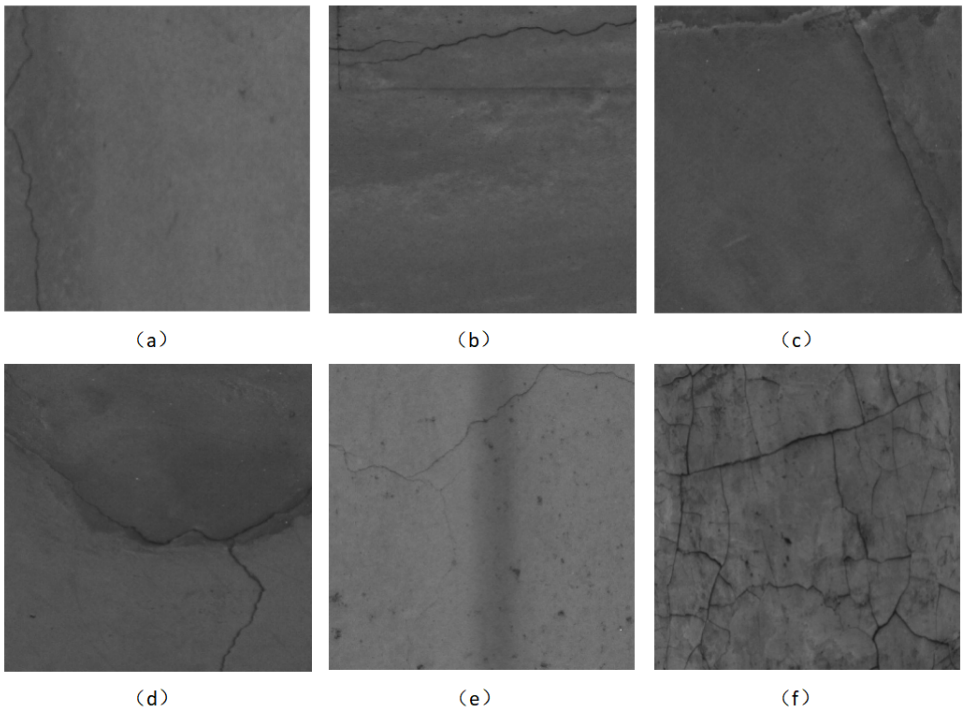

Figure 5. Crack morphology. They should be listed as (**a**) the display of longitudinal cracks; (**b**) the display of circumferential crack; (**c**) the display of oblique crack; (**d**) the display of a network of cracks; (**e**) the display of intersecting cracks; and (**f**) the display of craze cracks.

Table 1. Cracks in the TunnelCrackDB dataset.

	Total	**Longitude**	**Circular**	**Oblique**	**Network**	**Intersecting**	**Craze**
Train set	794	152	138	132	104	122	146
Valid set	89	19	17	13	11	11	18
Test set	99	23	20	16	14	12	14
Total	982	194	175	161	129	145	178

3.3. Evaluation Metrics

To evaluate the performance of the network model, this paper adopts the following evaluation metrics: precision, recall, F1 score, and mIoU (mean Intersection over Union). These metrics effectively measure the accuracy of semantic segmentation tasks. For binary classification problems, the distribution of model prediction results is described using a confusion matrix. The formulas for calculating precision, recall, F1 score, and mIoU are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{19}$$

$$Recall = \frac{TP}{TP + FP} \tag{20}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precison + Recall} \tag{21}$$

$$mIoU = \frac{1}{2}\left(\frac{TP}{TP + FP + FN} + \frac{TN}{TN + FP + FN}\right) \tag{22}$$

4. Results and Discussion

4.1. Comparative Experiments

CGV-Net was compared with recently published crack detection networks in terms of accuracy, recall, F1 score, and mIoU (mean Intersection over Union), using both the Crack dataset and TunnelCrackDB.

Public Crack dataset: Table 2 presents the results of CGV-Net and other network models on the Crack public dataset under the same parameter conditions. In the comparative experiments, the proposed CGV-Net was evaluated alongside publicly available network models (U-Net [27], SegNet [28], PSPNet [29], Deeplabv3 [15], Deeplabv3+ [30], and Deepcrack-net [31]) based on semantic segmentation metrics on the Crack public dataset. The experiments used precision, recall, F1 score, and mIoU as evaluation metrics to comprehensively assess the ability of each model to capture both global and local crack structures.

Table 2. Comparison of the performance of different network models on the Crack dataset.

	Precision	**Recall**	**F1**	**mIoU**
U-Net [27]	43.55%	55.78%	48.92%	45.32%
SegNet [28]	47.06%	66.35%	55.05%	52.74%
PSPNet [29]	35.64%	67.39%	46.62%	45.50%
Deeplabv3 [15]	36.36%	54.52%	43.63%	42.07%
Deeplabv3+ [30]	46.79%	71.94%	56.70%	55.43%
Deepcrack-net [31]	45.66%	63.19%	53.02%	51.26%
CGV-Net	47.11%	73.27%	57.32%	56.14%

The results indicate that, although traditional models provide some effectiveness in tunnel lining crack segmentation tasks, they exhibit limitations in feature generalization, particularly when dealing with complex crack shapes. In particular, the F1 score and mIoU of Deeplabv3 are only 43.63% and 42.07%, respectively. In contrast, CGV-Net shows significant improvement in both recall and F1 score by incorporating the CGV module, which merges CNN and GNN structural information extraction with the global and local modeling capabilities of the Vision Transformer (ViT). Specifically, CGV-Net achieves a recall of 73.27% and mIoU of 56.14%, demonstrating its superiority in capturing fine details. Meanwhile, the F1 score improves to 57.32%, showcasing a good balance between precision and recall.

Through innovative module design, CGV-Net effectively enhances its ability to model complex crack geometries, significantly improving the accuracy and robustness of tunnel lining crack segmentation. Figure 6 presents the segmentation results of different network models on the Crack test set, while Figure 7 shows that CGV-Net outperforms other models in terms of precision and recall on the Crack dataset.

Custom-built TunnelCrackDB dataset: The overall comparative results on the TunnelCrackDB dataset are shown in Table 3. CGV-Net demonstrates a significant advantage over traditional networks in the tunnel lining crack segmentation task. Comparative experiments with publicly available models such as U-Net [27], SegNet [28], PSPNet [29], Deeplabv3 [15], Deeplabv3+ [30], and Deepcrack-net [31] show that, although these traditional methods have some effectiveness in feature capture, they still have limitations in recognizing and segmenting complex crack structures. CGV-Net achieves the best performance on the three key metrics—precision, recall, F1 score, and mIoU—reaching 81.15%, 83.54%, 82.33%, and 81.24%, respectively. As shown in Figure 7, CGV-Net satisfies both precision and recall requirements on the TunnelCrackDB dataset, outperforming other models. Compared with Deepcrack networks, CGV-Net achieves a better balance between

precision and recall, with a particularly notable improvement in recall, demonstrating its superior robustness and generalization ability.

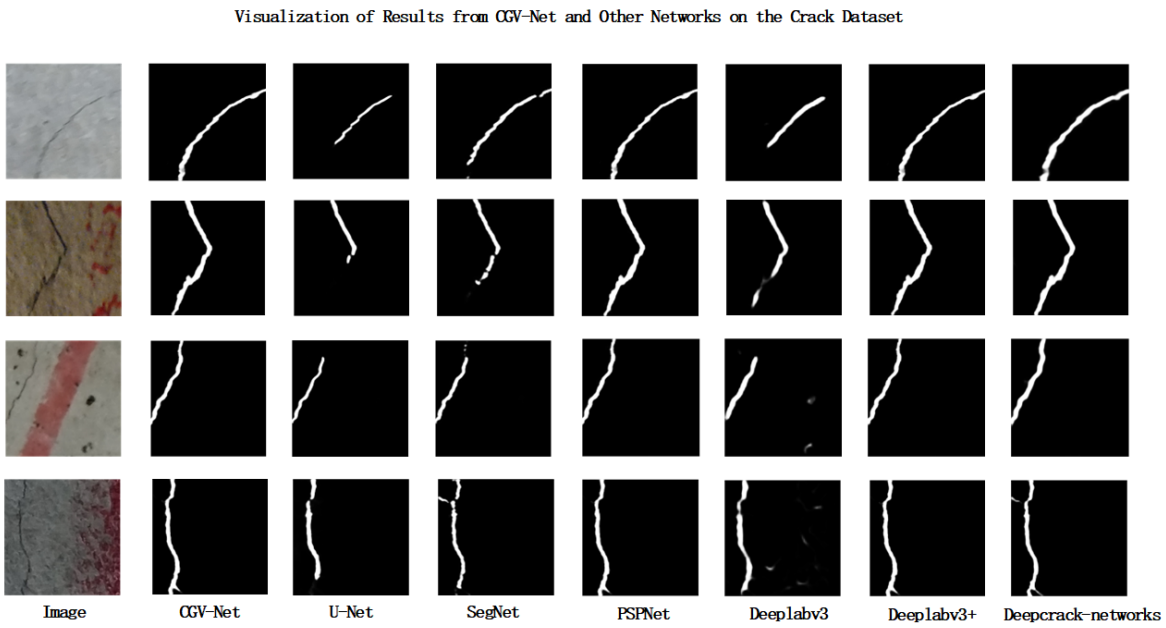

Figure 6. Visualization of results from CGV-Net and other networks on the Crack dataset.

Table 3. Comparison of the performance of different network models on the TunnelCrackDB dataset.

	Precision	Recall	F1	mIoU
U-Net [27]	63.56%	65.69%	64.61%	63.41%
SegNet [28]	79.43%	82.89%	81.12%	79.96%
PSPNet [29]	51.51%	71.59%	59.91%	58.32%
Deeplabv3 [15]	79.45%	81.81%	80.61%	7962%
Deeplabv3+ [30]	80.84%	83.03%	81.92%	80.83%
Deepcrack-net [31]	78.93%	83.00%	80.92%	79.51%
CGV-Net	81.15%	83.54%	82.33%	81.24%

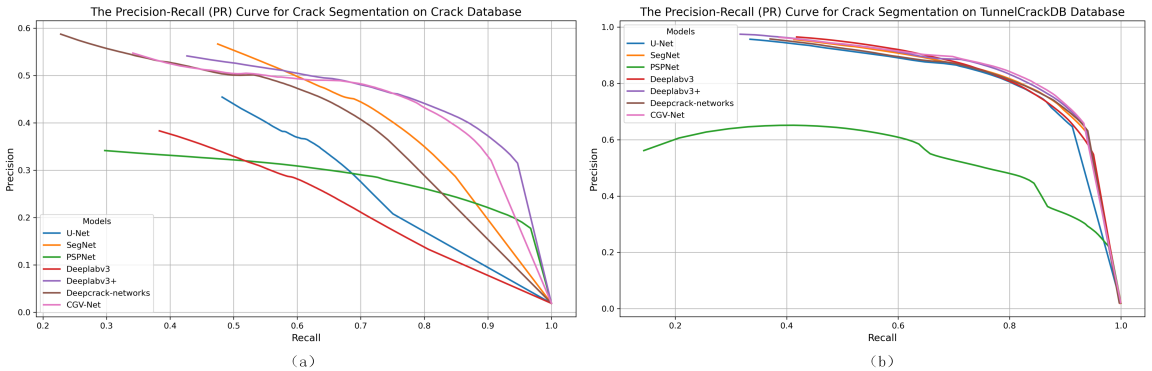

Figure 7. Visualization of results from CGV-Net and other networks on the TunnelCrackDB database. They should be listed as (**a**) description of the precision–recall (PR) curves for different models on the Crack dataset in the first panel and (**b**) description of the precision–recall (PR) curves for different models on the TunnelCrackDB dataset in the second panel.

The two innovative modules in CGV-Net, the CGV module and the DMFF module, play a crucial role in the performance enhancement. The CGV module strengthens the model's feature representation ability in complex and sparse scenarios by leveraging the local feature extraction advantage of CNN, the structural information modeling capability of GNN, and the global topology modeling ability of ViT. Meanwhile, the DMFF module

enhances the model's comprehensive understanding of crack geometries through multi-scale feature fusion, making full use of information from different layers.

Although SegNet and Deepcrack networks show some stability in overall performance (F1 scores of 81.12% and 80.92% and mIoUs of 79.96% and 79.51%, respectively), they still perform poorly when handling complex crack structures due to their limited ability to model global information. Deeplabv3 has an F1 score of 80.61% and an mIoU of 80.61%, both lower than CGV-Net's 81.24%, further demonstrating its limitations in feature extraction and upsampling strategies. In contrast, CGV-Net shows stronger performance in both metrics, indicating better robustness in handling complex crack structures. U-Net's performance is particularly underwhelming, with a precision of 63.56%, an F1 score of 64.61%, and an mIoU of 63.41%, highlighting the challenges this model faces in segmentation tasks involving complex geometric structures.

CGV-Net, through the innovative synergy of the CGV module and the DMFF module, effectively enhances the model's feature learning ability for both local details and global structures, demonstrating exceptional performance in the tunnel lining crack segmentation task. Figure 8 presents feature maps from different layers of the network model, validating the effectiveness of the proposed method on the TunnelCrackDB dataset. Notably, the method shows higher precision and robustness when handling complex crack shapes. Figure 9 displays the segmentation results of different network models on the TunnelCrackDB test set.

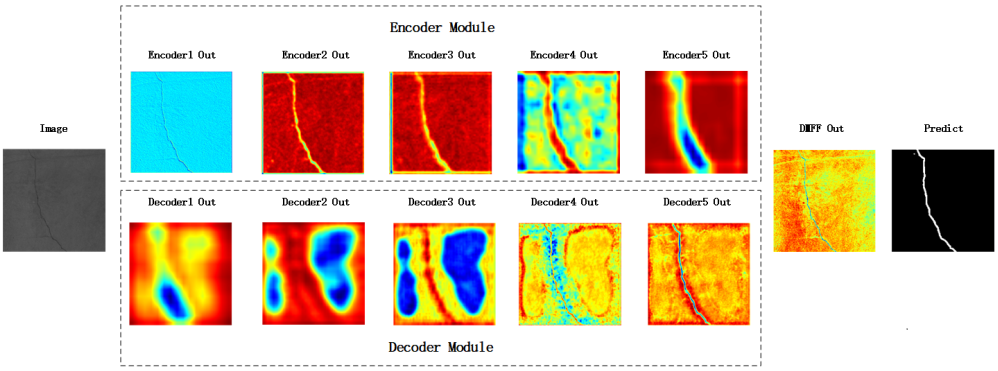

Figure 8. Diagram of the characteristics of the different network layers.

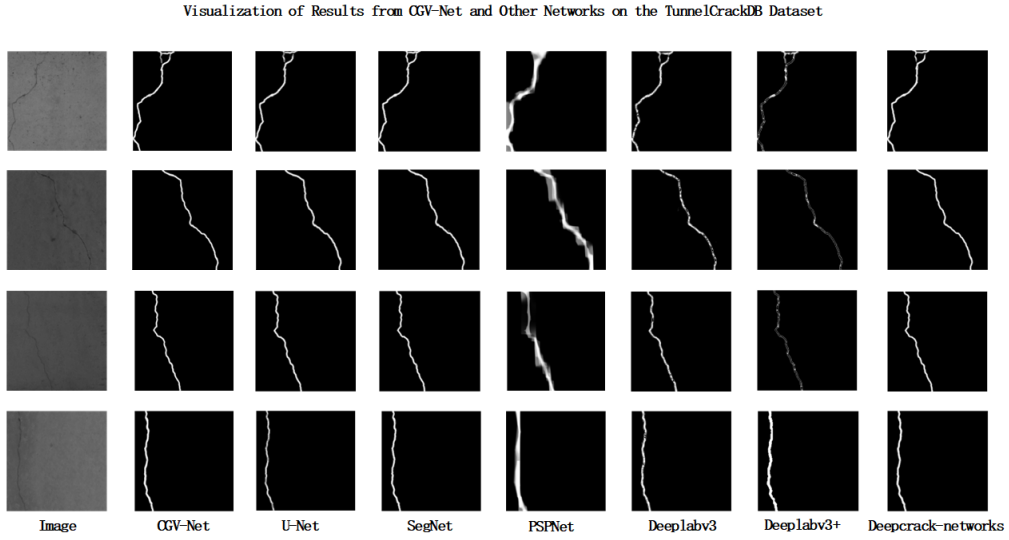

Figure 9. Visualization of results from CGV-Net and other networks on the TunnelCrackDB database.

4.2. Ablation Experiment

In the ablation study, precision, recall, F1 score, and mIoU (mean Intersection over Union) were used as segmentation evaluation metrics.On the TunnelCrackDB dataset, the ablation experiments compare the performance of the backbone network and the effect of adding each module, analyzing the contribution of each module to the network's performance. The results of the ablation experiments are shown in Table 4. The segmentation results of different network models are shown in Figure 10.

Table 4. Ablation experiment.

	Precision	Recall	F1	mIoU
Baseline	79.43%	82.89%	82.07%	80.36%
Baseline+CGV	80.11%	82.94%	81.51%	80.57%
Baseline+DMFF	80.81%	81.79%	81.30%	80.48%
Baseline+CGV+DMFF (CGV-Net)	81.15%	83.54%	82.33%	81.24%

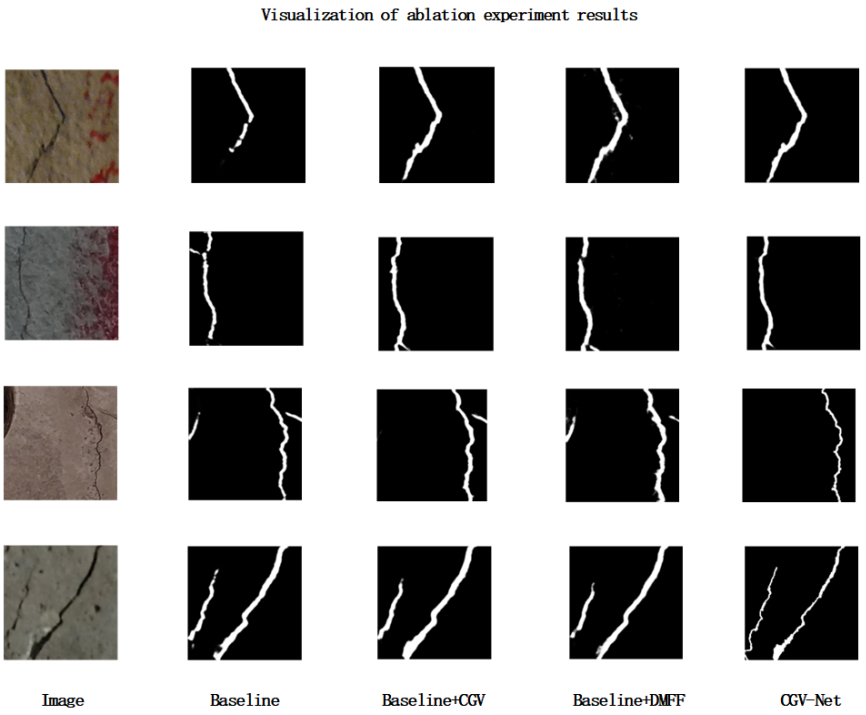

Figure 10. Visualization of ablation experiment results.

In the ablation study, precision, recall, F1 score, and mIoU (mean Intersection over Union) were used as segmentation evaluation metrics. The Baseline model achieved an mIoU of 80.36%, reflecting the basic performance of the model in tunnel lining crack segmentation without any additional modules.

After adding the CGV module, the precision of the Baseline slightly improved to 80.11%, while the recall remained nearly unchanged at 82.94%, and the F1 score decreased to 81.51%. The mIoU showed a slight increase, reaching 80.57%. This suggests that the CGV module has a limited effect on enhancing global information capture, contributing to some improvement in segmentation precision but showing relatively small effects on recall and F1 score.

With the addition of the DMFF module, the Baseline model's precision increased to 80.81%, but recall slightly decreased to 81.79%, and the F1 score was 81.30%. The mIoU rose to 80.48%, slightly higher than the Baseline's 80.36%. Although the DMFF module demonstrated some effectiveness in multi-scale feature fusion, improving segmentation

details and accuracy, its impact on recall improvement was still insufficient, and the mIoU increase was relatively limited.

Finally, the complete CGV-Net achieved a precision of 81.15%, recall of 83.54%, F1 score of 82.33%, and mIoU of 81.24%.CGV helps the model capture the global relationship between the target region and the background, making the segmentation task more coherent. On the other hand, DMFF enhances the segmentation of details through multi-scale feature fusion, particularly in handling complex edges and small region features, thus reducing the risk of mis-segmentation.Compared to the Baseline, CGV-Net's mIoU improved by 0.88 percentage points, validating their crucial role in enhancing segmentation precision, recall, and mIoU.

4.3. Experimental Performance

To comprehensively evaluate the performance of CGV-Net, we selected three key metrics for analysis: Floating Point Operations (FLOPs), the number of parameters, and precision, with the results shown in Figure 11. CGV-Net's performance on these metrics is as follows: Its FLOPs is 42.1 G, which is considered moderate; the number of parameters is 55.36 M, the highest among all the compared models, which may indicate that CGV-Net has strong model expressive power. Notably, in terms of precision, CGV-Net achieved 47.11%, the highest among all the compared models, indicating its high accuracy in image segmentation tasks.

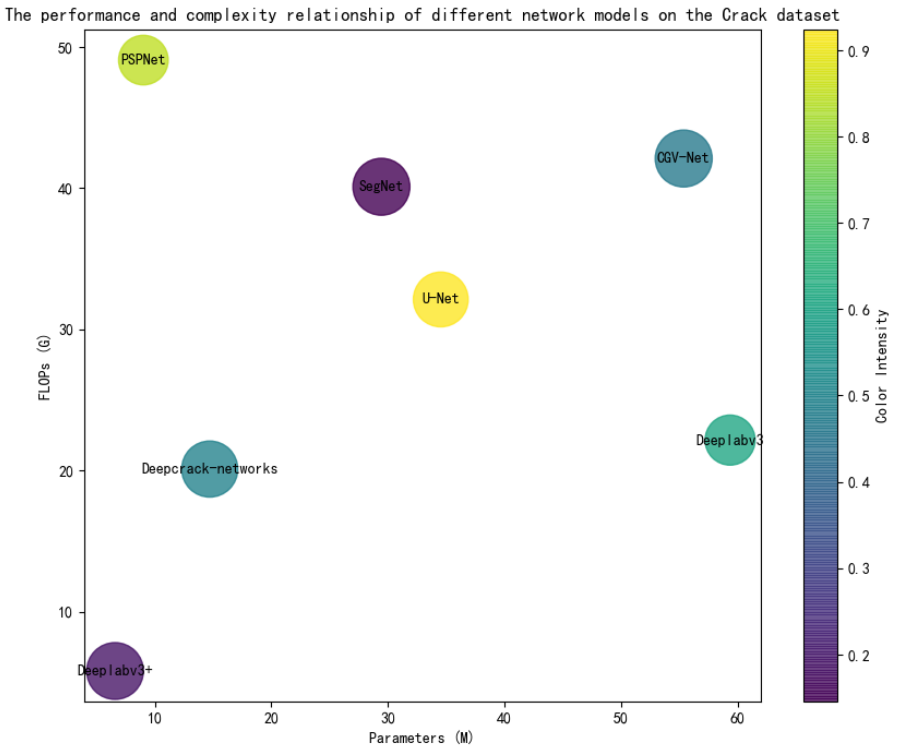

Figure 11. The performance and complexity relationship of different network models on the Crack dataset.

The advantage of CGV-Net lies in its integration of the features of convolutional neural networks (CNNs) and graph convolutional networks (GNNs), enabling it to effectively capture both local features and global contextual information. This allows CGV-Net to achieve higher accuracy when handling complex image segmentation tasks. Despite its larger number of parameters, the relatively low FLOPs value suggests that CGV-Net demonstrates good computational efficiency while maintaining high accuracy. Therefore, CGV-Net

not only has theoretical innovation but also exhibits strong performance advantages in practical applications.

5. Conclusions

This paper proposes a novel tunnel lining crack segmentation network, CGV-Net, which significantly improves crack segmentation accuracy and recall rate, especially in complex backgrounds, through the Vision Transformer module guided by graph neural networks and a multi-scale fine and coarse feature fusion module. Future work directions include not only further optimizing the segmentation of lining surface crack features but also quantifying the segmented cracks to obtain information such as crack width and length, which can be used for tunnel structural risk assessment. Additionally, efforts should be made to enhance the model's robustness under extreme lighting conditions. Furthermore, exploring the transfer learning potential of CGV-Net in other infrastructure-related tasks is also a promising research direction. For example, applying this model to structural health monitoring of bridges or buildings could effectively utilize existing crack segmentation technologies, providing broader applications and development opportunities in the engineering field.

These future research directions will help expand the applicability and benefits of CGV-Net in practical engineering applications, better serving the needs of tunnel operation and safety management.

Author Contributions: Conceptualization, K.L., T.R. and Z.L.; methodology, K.L., T.R. and Z.L.; software, K.L. and Y.X.; validation, Y.Y., R.L. and Y.X.; formal analysis, T.R. and R.L.; investigation, T.R. and Z.L.; writing—original draft preparation, K.L.; writing—review and editing, K.L. and T.R.; visualization, K.L.; supervision, Z.L., T.R. and R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: Authors Kai Liu, Tao Ren and Yang Yang were employed by the company China Railway Changjiang Transport Design Group Co., Ltd. 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. Qiu, J.; Liu, D.; Zhao, K.; Lai, J.; Wang, X.; Wang, Z.; Liu, T. Influence spatial behavior of surface cracks and prospects for prevention methods in shallow loess tunnels in China. *Tunn. Undergr. Space Technol.* **2024**, *143*, 105453. [CrossRef]
2. Xu, H.; Wang, M.; Liu, C.; Li, F.; Xie, C. Automatic detection of tunnel lining crack based on mobile image acquisition system and deep learning ensemble model. *Tunn. Undergr. Space Technol.* **2024**, *154*, 106124. [CrossRef]
3. Chen, L.L.; Li, J.; Wang, Z.F.; Wang, Y.Q.; Li, J.C.; Li, L. Sustainable health state assessment and more productive maintenance of tunnel: A case study. *J. Clean. Prod.* **2023**, *396*, 136450. [CrossRef]
4. Rosso, M.M.; Aloisio, A.; Randazzo, V.; Tanzi, L.; Cirrincione, G.; Marano, G.C. Comparative deep learning studies for indirect tunnel monitoring with and without Fourier pre-processing. *Integr. Comput.-Aided Eng.* **2023**, *31*, 213–232. [CrossRef]
5. Feng, Y.; Zhang, X.L.; Feng, S.J.; Zhang, W.; Hu, K.; Da, Y.W. Intelligent segmentation and quantification of tunnel lining cracks via computer vision. *Struct. Health Monit.* **2024**. [CrossRef]
6. Wang, R.; Chen, R.Q.; Guo, X.X.; Liu, J.X.; Yu, H.Y. Automatic recognition system for concrete cracks with support vector machine based on crack features. *Sci. Rep.* **2024**, *14*, 20057. [CrossRef]
7. Rosso, M.M.; Marasco, G.; Aiello, S.; Aloisio, A.; Chiaia, B.; Marano, G.C. Convolutional networks and transformers for intelligent road tunnel investigations. *Comput. Struct.* **2023**, *275*, 106918. [CrossRef]
8. Jiang, F.; Wang, G.; He, P.; Zheng, C.; Xiao, Z.; Wu, Y. Application of canny operator threshold adaptive segmentation algorithm combined with digital image processing in tunnel face crevice extraction. *J. Supercomput.* **2022**, *78*, 11601–11620. [CrossRef]

9. Long, S.; Yang, T.; Qian, Y.; Wu, Y.; Xu, F.; Tang, Q.; Guo, F. GPR Imagery Based Internal Defect Evaluation System for Railroad Tunnel Lining Using Real-time Instance Segmentation. *IEEE Sens. J.* **2024**, *24*, 35997–36010. [CrossRef]
10. Lei, M.; Liu, L.; Shi, C.; Tan, Y.; Lin, Y.; Wang, W. A novel tunnel-lining crack recognition system based on digital image technology. *Tunn. Undergr. Space Technol.* **2021**, *108*, 103724. [CrossRef]
11. A, R.M.N.; K, S.S. Predicting the settlement of geosynthetic-reinforced soil foundations using evolutionary artificial intelligence technique. *Geotext. Geomembr.* **2021**, *49*, 1280–1293.
12. Zhao, S.; Zhang, D.; Xue, Y.; Zhou, M.; Huang, H. A deep learning-based approach for refined crack evaluation from shield tunnel lining images. *Autom. Constr.* **2021**, *132*, 103934. [CrossRef]
13. Dang, L.M.; Wang, H.; Li, Y.; Park, Y.; Oh, C.; Nguyen, T.N.; Moon, H. Automatic tunnel lining crack evaluation and measurement using deep learning. *Tunn. Undergr. Space Technol.* **2022**, *124*, 104472. [CrossRef]
14. Razveeva, I.; Kozhakin, A.; Beskopylny, A.N.; Stel'makh, S.A.; Shcherban', E.M.; Artamonov, S.; Pembek, A.; Dingrodiya, H. Analysis of Geometric Characteristics of Cracks and Delamination in Aerated Concrete Products Using Convolutional Neural Networks. *Buildings* **2023**, *13*, 3014. [CrossRef]
15. Wang, H.; Li, Y.; Dang, L.M.; Lee, S.; Moon, H. Pixel-level tunnel crack segmentation using a weakly supervised annotation approach. *Comput. Ind.* **2021**, *133*, 103545. [CrossRef]
16. Zhao, S.; Zhang, G.; Zhang, D.; Tan, D.; Huang, H. A hybrid attention deep learning network for refined segmentation of cracks from shield tunnel lining images. *J. Rock Mech. Geotech. Eng.* **2023**, *15*, 3105–3117. [CrossRef]
17. Lin, Q.; Li, W.; Zheng, X.; Fan, H.; Li, Z. DeepCrackAT: An effective crack segmentation framework based on learning multi-scale crack features. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106876. [CrossRef]
18. Chen, S.; Feng, Z.; Xiao, G.; Chen, X.; Gao, C.; Zhao, M.; Yu, H. Pavement Crack Detection Based on the Improved Swin-Unet Model. *Buildings* **2024**, *14*, 1442. [CrossRef]
19. Qin, S.; Qi, T.; Deng, T.; Huang, X. Image segmentation using Vision Transformer for tunnel defect assessment. *Comput.-Aided Civ. Infrastruct. Eng.* **2024**, *39*, 3243–3268. [CrossRef]
20. Zhou, Z.; Yan, L.; Zhang, J.; Zheng, Y.; Gong, C.; Yang, H.; Deng, E. Automatic segmentation of tunnel lining defects based on multiscale attention and context information enhancement. *Constr. Build. Mater.* **2023**, *387*, 131621. [CrossRef]
21. Tao, H.; Liu, B.; Cui, J.; Zhang, H. A convolutional-transformer network for crack segmentation with boundary awareness. In Proceedings of the 2023 IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia, 8–11 October 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 86–90.
22. Pu, R.; Ren, G.; Li, H.; Jiang, W.; Zhang, J.; Qin, H. Autonomous concrete crack semantic segmentation using deep fully convolutional encoder–decoder network in concrete structures inspection. *Buildings* **2022**, *12*, 2019. [CrossRef]
23. Wang, A.; Togo, R.; Ogawa, T.; Haseyama, M. Defect detection of subway tunnels using advanced U-Net network. *Sensors* **2022**, *22*, 2330. [CrossRef]
24. Kang, D.; Benipal, S.S.; Gopal, D.L.; Cha, Y.J. Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning. *Autom. Constr.* **2020**, *118*, 103291. [CrossRef]
25. Huang, H.; Zhao, S.; Zhang, D.; Chen, J. Deep learning-based instance segmentation of cracks from shield tunnel lining images. *Struct. Infrastruct. Eng.* **2022**, *18*, 183–196. [CrossRef]
26. Jiang, Y.; Wang, L.; Zhang, B.; Dai, X.; Ye, J.; Sun, B.; Liu, N.; Wang, Z.; Zhao, Y. Tunnel lining detection and retrofitting. *Autom. Constr.* **2023**, *152*, 104881. [CrossRef]
27. 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]
28. Chen, T.; Cai, Z.; Zhao, X.; Chen, C.; Liang, X.; Zou, T.; Wang, P. Pavement crack detection and recognition using the architecture of segNet. *J. Ind. Inf. Integr.* **2020**, *18*, 100144. [CrossRef]
29. Zhu, X.; Cheng, Z.; Wang, S.; Chen, X.; Lu, G. Coronary angiography image segmentation based on PSPNet. *Comput. Methods Programs Biomed.* **2021**, *200*, 105897. [CrossRef]
30. Fu, H.; Meng, D.; Li, W.; Wang, Y. Bridge crack semantic segmentation based on improved Deeplabv3+. *J. Mar. Sci. Eng.* **2021**, *9*, 671. [CrossRef]
31. Chen, H.; Lin, H. An effective hybrid atrous convolutional network for pixel-level crack detection. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 5009312. [CrossRef]