

Research Article

The Improvement of Automated Crack Segmentation on Concrete Pavement with Graph Network

Jiang Chen , Ye Yuan, Hong Lang , Shuo Ding, and Jian John Lu

The Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai 201804, China

Correspondence should be addressed to Hong Lang; honglang@tongji.edu.cn

Received 25 January 2022; Revised 7 April 2022; Accepted 23 May 2022; Published 11 June 2022

Academic Editor: Yong Zhang

Copyright © 2022 Jiang Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Crack is a common concrete pavement distress that will deteriorate into severe problems without timely repair, which means the automated detection of pavement crack is essential for pavement maintenance. However, automatic crack detection and segmentation remain challenging due to the complex pavement condition. Recent research on pavement crack detection based on deep learning has laid a good foundation for automated crack segmentation, but there can still be improvements. This paper proposes an automatic concrete pavement crack segmentation framework with enhanced graph network branch. First, the nodes of the graph and nodes' attributions are generated based on the image dividing. The edges of the graph are determined based on Gaussian distribution. Then, the graph from the image is input into the graph branch. The graph feature map of the graph branch output is fused with the image feature map of the encoder and then enters the decoder to recover the image resolution to obtain the crack segmentation result. Finally, the method is tested on a self-built 3D concrete pavement crack dataset. The proposed method achieves the highest F1 and IoU (Intersection over Union) in the comparison experiments. And the graph branch addition improves 0.08 on F1 and 0.06 on IoU compared with U-Net.

1. Introduction

Road infrastructure is an essential asset for a country, and it can contribute to the economic development and bring significant social benefits. Road density is adopted as a rating criterion by the World Bank to evaluate low-income, middle-income, and high-income economies [1]. Concrete pavement is one of the main pavement types. The concrete pavement in the United States highway network accounts for about 49 percent, and in Belgium they occupy 50 percent. However, due to the severe traffic loading and the variable environment, concrete pavement distress always appears over the road operation time. Maintaining an acceptable level of service for the whole road network is a challenge to the transportation agency officials.

Pavement distress evaluation is the essential work for pavement maintenance. The transportation agency officials regard pavement data collection as a regular work to grasp the evolution of road conditions and make opportune work to stop the deterioration of the distress. Efficient pavement condition inspections and reasonable repair strategies can

lead to a significant reduction in life-cycle pavement maintenance cost [2].

Pavement distress evaluation has undergone a long period of development with the continuous advancement of computer technology. Traditional distress inspections are based on the manual visual survey, which is time-consuming. After that, a collection vehicle equipped with a high-speed digital camera is invented to acquire the pavement surface images at a high speed [3, 4]. This method causes little influence on traffic operation and is widely accepted by the transportation agency officials. Recently, 3D technology has attracted much attention. Compared with the 2D technology, the pixels in the image captured by 3D technology describe the depth change relative to the reference surface. Therefore, the 3D images of concrete pavement can reduce the influence of surface oil and lighting conditions and present more information on pavement distress [5–7].

Once the concrete pavement surface images are obtained, the processing can be conducted to detect pavement distress using various algorithms. Over the past decade, there have been sufficient methods based on computer vision

proposed to detect pavement distress automatically and achieve excellent results, such as methods based on threshold [8, 9], methods based on edge detection [10, 11]. However, the effects of most methods are easily influenced by different pavement detection environments due to the feature extraction based on manual design. Therefore, semiautomatic pattern to pavement crack detection is in current practice. In the semiautomatic approach, crack detection algorithms are applied first, and then a series of human interventions are conducted to manually adjust the crack information and incorrect results. It is also time-consuming.

Recently, with the success of deep learning methods, especially Convolutional Neural Network (CNN) in computer vision tasks [12], applying deep learning to automatic pavement distress detection has become a spotlight. CNN can automatically extract the features of objects in the images with a structure similar to the human brain compared to manually designed feature extraction in traditional methods. In the current application, the deep learning based pavement crack detection method can be divided into three categories, e.g., patch classification [13], object detection [14], and semantic segmentation [15]. The patch classification methods divide the pavement image into several blocks of the same size and then classify each block into the corresponding category. The object detection methods frame the crack in the pavement image using a bounding box. Furthermore, the semantic segmentation methods classify each pixel in the pavement image. Hence, the semantic segmentation methods can achieve the pixel-level inspection result and obtain more detailed characteristics of distress, such as precise length and area of the crack. However, the low accuracy and high false positives of the semantic segmentation when the pavement conditions change limit the promotion in practice.

Concrete pavement is a rigid pavement, while asphalt pavement is a flexible pavement. Cracks have different characteristics in different pavements. The crack on the concrete pavement has a more obvious and complex boundary compared to the crack on the asphalt pavement and often has a jump down in pavement depth changes. And joints between the concrete blocks and the indentations in concrete pavement cause the more complex surface texture than asphalt pavement. The complex texture will bring interference to the identification of crack. Most of the research on pavement distress and the pavement datasets constructed nowadays focus on asphalt pavement distress [16, 17]. The effectiveness of transferring the method applied for asphalt pavement distress detection to the concrete pavement is substantially reduced. It is necessary to establish a concrete pavement distress dataset and a method to detect cracks in the concrete pavement.

In this work, a feature extraction branch based on graph neural network is added to a typically semantic segmentation network to form a new end-to-end network structure. And experiments are conducted on concrete pavement crack segmentation to evaluate the performance improvement. In this regard, the main contributions of this work can be summarized as follows:

- (i) A semantic segmentation network framework with graph neural network branch is proposed to segment the concrete pavement crack. The performance of crack segmentation is significantly improved based on the original segmentation network. In addition, the inclusion of the graph branch improves the continuity of crack segmentation.
- (ii) A generation method to convert images into graphs is designed, which enriches the feature map dimension of images.
- (iii) A new dataset of 3D concrete pavement crack images is established and applied to evaluate the proposed network.

The rest of this paper is organized as follows. Section 2 describes the related research on pavement crack detection and the development of graph neural networks. Section 3 introduces the detailed architecture of the segmentation network with graph neural network branch. Section 4 represents the experiment setting. Section 5 discusses the experiment results. Finally, Section 6 concludes the work and presents the findings of this research.

2. Related Work

2.1. Semantic Segmentation. The semantic segmentation method is the classification of the category for each pixel in the image. The fully convolution network (FCN) proposed by Long et al. is the milestone for semantic segmentation based on deep learning [18]. They apply a 1×1 convolution layer instead of a fully connected layer as a classifier. Hence, the output of the network is transformed from a vector to a matrix, where the value of each pixel represents the probability that the corresponding pixel of the input image belongs to a specific class. Moreover, an encoder-decoder structure is added to the network design for semantic segmentation [19]. This structure can improve computational efficiency and reduce the overfitting problem. In simple terms, the encoder process extracts the feature of the input image by convolutional computation and pooling, and the decoder process restores the feature map to a matrix with the same size as the input by upsampling.

The semantic segmentation method used in pavement crack detection is to classify each pixel of the image into two categories, crack pixel and none crack pixel. Due to the easy obtaining of pavement crack's geometric characteristics such as length, area, and bounding box, semantic segmentation is widely popular. Yang et al. offer an FCN-based method to segment crack pixel in the pavement image and acquire the length, width, and mean width of crack [20]. Liu et al. develop a U-Net based model to segment concrete cracks [15]. Qu et al. improve crack segmentation performance with attention mechanism and apply their model in asphalt pavement and concrete pavement crack segmentation [21].

2.2. Graph Neural Network. Graphs are all around us. The graph has two elements consisting of nodes and edges, which represent a set of objects and the connection between them,

respectively [22]. Anything with a connection relationship can be described as a graph, e.g., image, text, and social network. Motivated by the neural network and deep learning, new significant operations have rapidly developed over the past few years to handle the complexity of graph data. Compared to the other networks, the graph neural networks (GNNs) need two vectors or matrices as input, representing the node and the edge attributes of the graph. Sanchez-Lengeling et al. propose the Spectral network and use a learnable diagonal matrix as the filter to process graph [23]. However, the operation is computationally inefficient and the filter is nonspatially localized. Inspired by the 2D convolution in image computing, Kipf and Welling develop the graph convolution operation to alleviate the overfitting problem and promote the computational efficiency [24]. To address the large-scale graph computation problem, spatial approaches based on the graph convolution are developed to adjust to different sized neighborhoods and maintain the local invariance [25–27]. Zhang et al. improve the graph network's ability to extract node relationships by adding a cross attention module and apply the graph network to metro passenger flow prediction, achieving state-of-the-arts performance [28, 29].

2.3. 3D Technology in Pavement Detection. The 2D images describe the grey-scale feature of the pavement surface, which is the most used method in traditional pavement distress detection. However, detection on 2D images is susceptible to surface oil, pavement texture, lighting condition, etc. 3D images describe the depth changes of pavement surface, which can overcome the shortage above and usually present more detail of distress like depth. With the development of 3D sensors and image processing technology, the potential of 3D measurement in pavement detection has earned widespread attention. 3D technologies applied in pavement inspection include 3D structure light [30, 31], laser scanning [32], and binocular stereo vision [33]. There have been attempts to combine 3D techniques and deep learning methods for pavement inspection. Zhang et al. propose a model called CrackNet based on a convolution neural network to detect crack on 3D asphalt pavement image and significantly outperforms the traditional approaches in terms of F-measure [34]. Lang et al. develop a multiscale clustering model for detecting different types of cracks, including linear and netted types on the 3D pavement surface [35]. However, most of the existing studies have focused on detecting asphalt pavements and less on the detection of concrete pavement distress. In this work, a concrete pavement dataset with 3D images is built for pavement crack detection and validates the accuracy of our proposed method.

3. Methodology

In this section, the graph neural network feature extraction branch and the main body of semantic segmentation are first introduced, respectively. Then, the proposed network structure for crack detection on the concrete pavement is described.

3.1. Graph Neural Network Branch. Adding new feature extraction branches to enrich feature map information is a common approach to improving the accuracy of semantic segmentation networks. Image is similar to graph data, and each pixel in the image can be regarded as a node in the graph. The relationship between every pixel can be considered as an edge in the graph, as shown in Figure 1. Note that the generation of nodes and edges in a graph is designed according to the realistic task. In this work, the node and its attribute generate from a group of pixels in a region. The image with the size of 512×512 is divided into 1,024 (32×32) patches with 16×16 size. Each patch forms a node, and the mean value of the pixels in the patch is calculated as the attribution of the node. In general, the neighbors of a node can be the nearest neighbor node or the node in the interval, even all other nodes. In this work, we assume that each node connects to the node with the interval of D . The connection means that the nodes attribution can be transferred by the edge in the graph neural network. There is an instance to describe the neighbors of a node when the D is 2, as illustrated in Figure 2. The edge information will be respected by the adjacent matrix as the input to the graph neural network.

The transform methods from image to graph including the node generation and the edge generation are determined. Then, the feature extraction branch is described. This part of the work is related to the GraphSAGE (Graph SAmple and aggreGatE) method proposed by Hamilton et al. [25]. GraphSAGE is an inductive learning framework that can use the vertice attribution to generate unknown node embedding. The feature extraction based on GraphSAGE can be divided into three steps as illustrated in Figure 3. In the first step, sample the local neighborhood and generate the embeddings for nodes. Considering the computing efficiency, sampling range K is proposed to control the number of neighboring nodes sampled. According to the edge generation, a node has at least 5 neighbors, and at most 12 nodes in this work. When K is larger than the number of node neighbors, the sampling with put-back is completed until K nodes are sampled, when not, the sampling without put-back is used. In the second step, choose an aggregator and aggregate feature information from neighbors. Since the neighbors of the node in the graph are disordered, the aggregator function needs to be symmetric, which means that the output of the function remains the same when the order of the inputs changes. A mean aggregator is applied in this step to connect the node and its neighbors and calculate the mean value of each dimension of the node attribution vectors. Through an activate function layer, the target representing vector of node is obtained. This step is equivalent to the convolutional computing for feature extraction in a convolutional neural network. In the third step, the aggregate information output from step 2 can be applied to the downstream tasks such as classification and prediction.

3.2. Semantic Segmentation Main Body. The semantic segmentation task acquires a combination of local information based on high-resolution images and global information based on low resolution to classify each pixel. Common

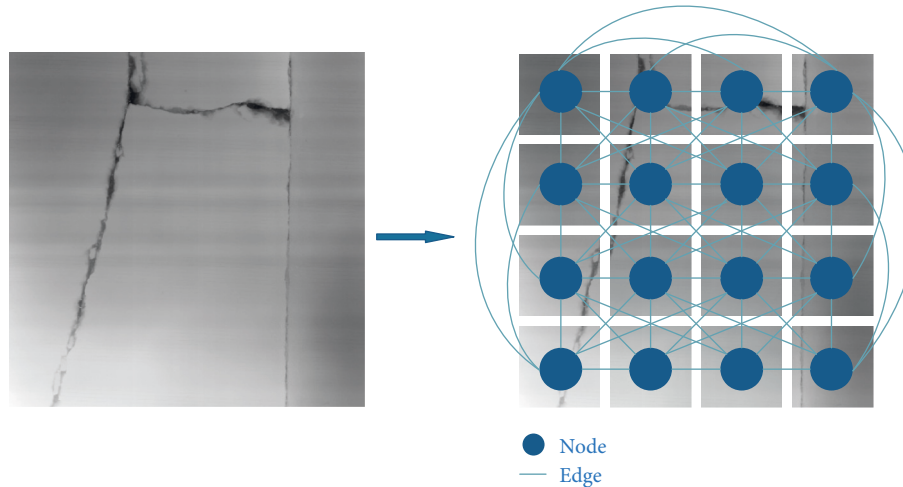


FIGURE 1: Schematic of transforming an image into a graph.

segmentation networks utilize an encoder-decoder framework to obtain the features of different levels of different scales. The main body of the network structure proposed in this work is related to U-Net with an encoder and decoder framework [19]. The U-Net structure is simple and easy to modify, as shown in Figure 4. Symmetry is one of the characteristics of U-Net. The left of Figure 4 is the encoder, while the right is the decoder. The encoder is responsible for the extraction of the image feature and the decoder is responsible for recovering the image resolution. In the encoder, there are five encoding blocks. Each encoding block consists of one convolutional layer with kernel size of 3×3 (deep blue arrow) and one maximum pooling layer (red arrow). The rotated numbers represent the width and the height of the images or the feature maps, while the normal number represents the number of channels. The convolutional layers do not change the sizes and channel numbers, but the maximum pooling layers do. After the maximum pooling layer, the output is halved in width and height but doubled in the number of channels compared to the input. The flow of the size and channel number is listed in order, $512 \times 512 \times 16$, $256 \times 256 \times 32$, $128 \times 128 \times 64$, $64 \times 64 \times 128$, $32 \times 32 \times 256$, $16 \times 16 \times 512$ (weight \times height \times channel number). In the decoder, there are five decoding blocks correspondingly. Each decoding block consists of one convolutional layer and one deconvolution layer (light blue arrow). The effect of the convolutional layer in the decoder is the same as in the encoder. The deconvolution layer is to recover the image resolution in the contrast to the pooling layers. After the deconvolution layer, the output is doubled in width and height but halved in channel number compared to the input. The flow of the size and channel number in 'decoder is list in order, $16 \times 16 \times 512$, $32 \times 32 \times 256$, $64 \times 64 \times 128$, $128 \times 128 \times 64$, $256 \times 256 \times 32$, $512 \times 512 \times 16$ (weight \times height \times channel number). Different from the encoding block, the input in the decoding block is not only the output of the upper decoding block but also includes the output from the encoding block at the same level. This design facilitates the integration of high-resolution detailed features and the low-resolution semantic feature to promote performance. And the ReLU activate

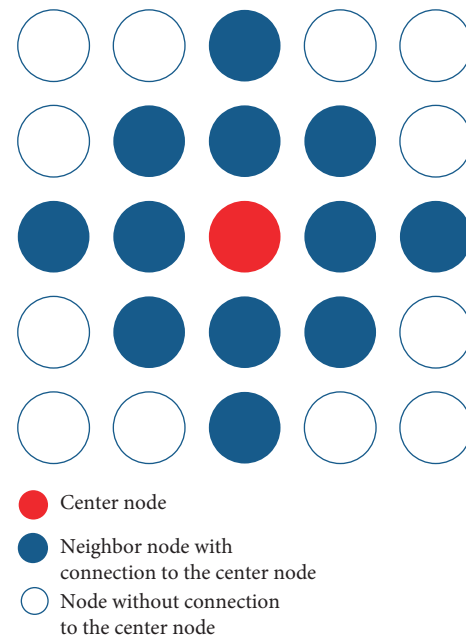


FIGURE 2: Nodes and neighbor nodes. Blue represents the node has edge to the red node. White represents the node has no edge to the red node.

function is added after each convolutional layer to boost the nonlinearity of the network. In the end, a convolutional layer with a kernel size of 1×1 is applied to classify the pixel into the corresponding class. The size of the output is 512×512 the same as the input. The number of channels is 2. The value of each pixel in output represents the probability of the corresponding pixel in the input belonging to a certain category.

3.3. Network Structure. By integrating the graph network branch in the U-Net, the network is developed, namely, GA-Net (Graph branch Added U-Net). The network structure is illustrated in Figure 5. Firstly, the input image is processed through the encoder in semantic segmentation main body

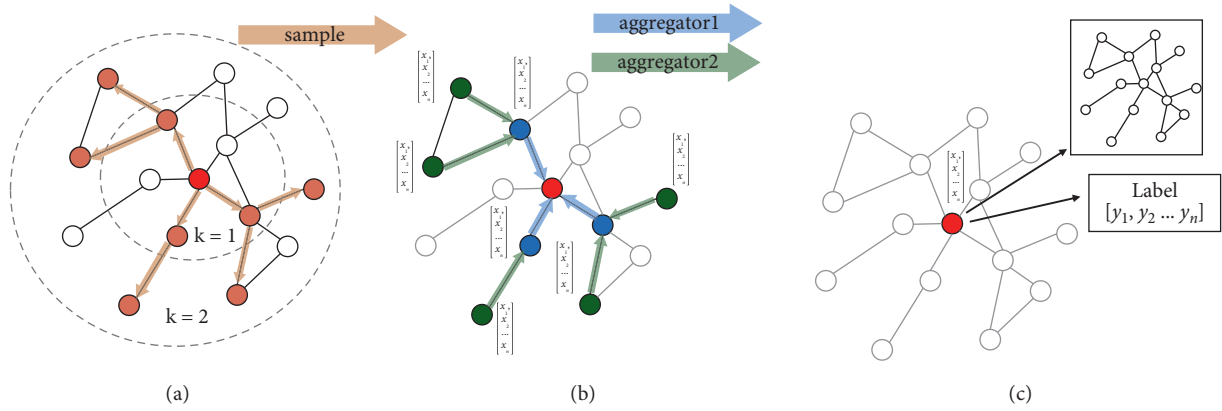


FIGURE 3: The process of feature extraction based on GraphSAGE(25). (a) Sample neighborhood, (b) Aggregate feature information from neighbors, (c) Predict graph context and label using aggregated information.

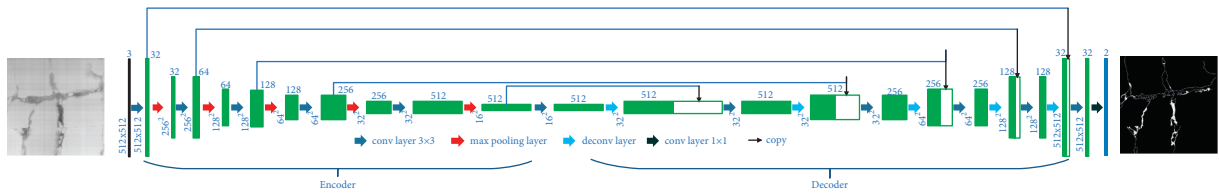


FIGURE 4: The structure of U-Net.

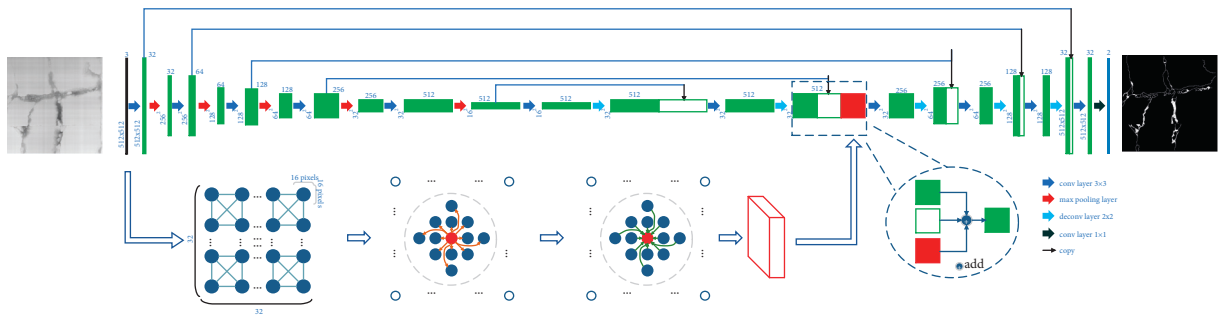


FIGURE 5: The structure of the whole network. The top-line describes the processing of the image in the U-Net. The number above the feature map presents the number of channels. The number beside the feature presents the size of the feature map. The following line describes the processing in the graph neural network branch. And the feature map in the U-Net and the graph network branch are fused after the first decoding.

and the graph network branch, respectively. In graph network branch, the image input with 512×512 size is transferred into a graph with 1024 (32×32) nodes and 5,174 edges. Through sampling, aggregating, and predicting in the graph network branch, the feature map with the size of 32×32 is obtained at the graph level. Then, the feature maps obtained by the graph branch and by the encoder are fused after the first decoding block and input to the subsequent decoding blocks.

4. Experiments and Results

The proposed method was evaluated on the self-captured concrete pavement crack dataset, namely, the CPC dataset. The performance of the proposed graph network branch was evaluated by comparing it with U-Net methods.

Furthermore, the proposed network was implemented using Pytorch on a personal computer with an Intel i7-11700K CPU @3.60 GHz, 64 GB memory, and an NVIDIA RTX3090 GPU with 24 GB memory.

4.1. CPC Dataset. The CPC dataset consisting of 3D concrete pavement crack images is built to train and test the proposed network in this work. The detection vehicle can scan the pavement at different collection speeds ranging from 35 to 100 km/h (20 to 60 mi/h). The pixel resolutions of the 3D pavement data are both 2048×2048 , covering pavement surfaces of more than 2 m in width and 2 m in length. Moreover, the CPC dataset contains images with various changes in pavement conditions aiming at the accuracy of crack recognition. There is no overlap between any

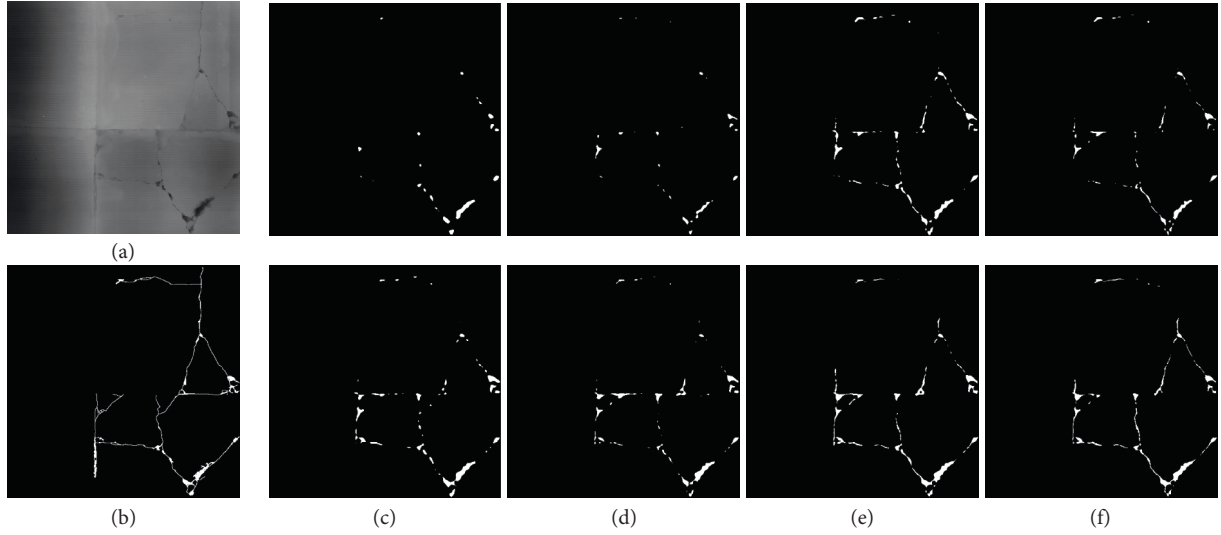


FIGURE 6: The result of GA-UNet and U-Net at different epoch. (a) original pavement image. (b) ground truth. (c) the results at epoch 50. (d) the results at epoch 100. (e) the results at epoch 200. (f) the results at epoch 300. In (c), (d), (e), (f) images, the upper line is the result of U-Net, and the down line is the results of GA-UNet.

TABLE 1: The comparison of the test results on the concrete pavement image dataset.

| | Precision | Recall | F1 | IoU |
|---------------|-------------|-------------|-------------|-------------|
| AutoEncoder | 0.38 | 0.53 | 0.42 | 0.28 |
| PSPNet [37] | 0.65 | 0.44 | 0.52 | 0.35 |
| U-Net [19] | 0.67 | 0.37 | 0.45 | 0.31 |
| KiU-Net [38] | 0.38 | 0.69 | 0.46 | 0.31 |
| GA-UNet(ours) | 0.63 | 0.49 | 0.53 | 0.37 |

2 images, and no more than 50 images are from the same pavement section. The 3D pavement image input into the network will be resized into 512×512 to reduce the computational effort. The final dataset consists of 1,452 images. After collection, the labeling work is conducted. The ground truth of cracks on all pavement images is manually labeled on pixel level by our research team. To ensure the accuracy of the ground truth, three rounds of labeling work were applied. In the first round, several well-trained operators label cracks manually on the pavement image. In the second round, the operators in the first round exchange their labeling results and check them. In the third round, the experts further confirm the availability of ground truth in each pavement image of the entire dataset. And finally we get accurate ground truth images. The ground truth image is a binary image, in which 0 represents the pavement background pixel and 1 represents the crack pixel. Then, the CPC dataset with ground truth is divided into two parts, 1,352 images for training and 100 images for testing.

4.2. Training Settings. The input image size of the network is resized into 512×512 . The epoch number is 300. And Adam [36] is chosen as the optimizer with a batch-size of 1 and weight decay of 0.00001. Training is started with a learning rate of 0.00005. The cross-entropy loss function is chosen as

the loss function in training, and the definition is shown in the following equation

$$\text{loss function} = -\frac{1}{N} \sum_i^N Y_i \log(y_i), \quad (1)$$

where N means the category number which is 7, Y_i is the ground truth representing whether the pixel belongs to category i , 1 for yes and 0 for no. And y_i is the prediction probability that the pixel belongs to category i .

4.3. Evaluation Criteria. Four metrics are introduced to evaluate the performance of crack semantic segmentation, Precision (Pr), Recall (Re), F1, and Intersection over Union (IoU). Precision describes the ratio of all pixels predicted to be the crack type that is actually positive. Recall shows the completion of crack prediction, which is a ratio of all crack pixels in the image which is predicted to be crack. F1 is the metric combining precision and recall. IoU calculates a ratio between the number of true positives and the sum of the true positives, false negatives, and false positives. The definition of Pr, Re, F1, and IoU are as follows:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP}, \\ \text{Recall} &= \frac{TP}{TP + FN}, \\ \text{F1} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \\ \text{IoU} &= \frac{\text{GroundTruth} \cap \text{Prediction}}{\text{GroundTruth} \cup \text{Prediction}}, \end{aligned} \quad (2)$$

where TP (True Positive) means the number of crack pixel predicted to be cracks, FP (False Positive) means the number of pavement pixel wrongly predicted to be cracks, FN (False

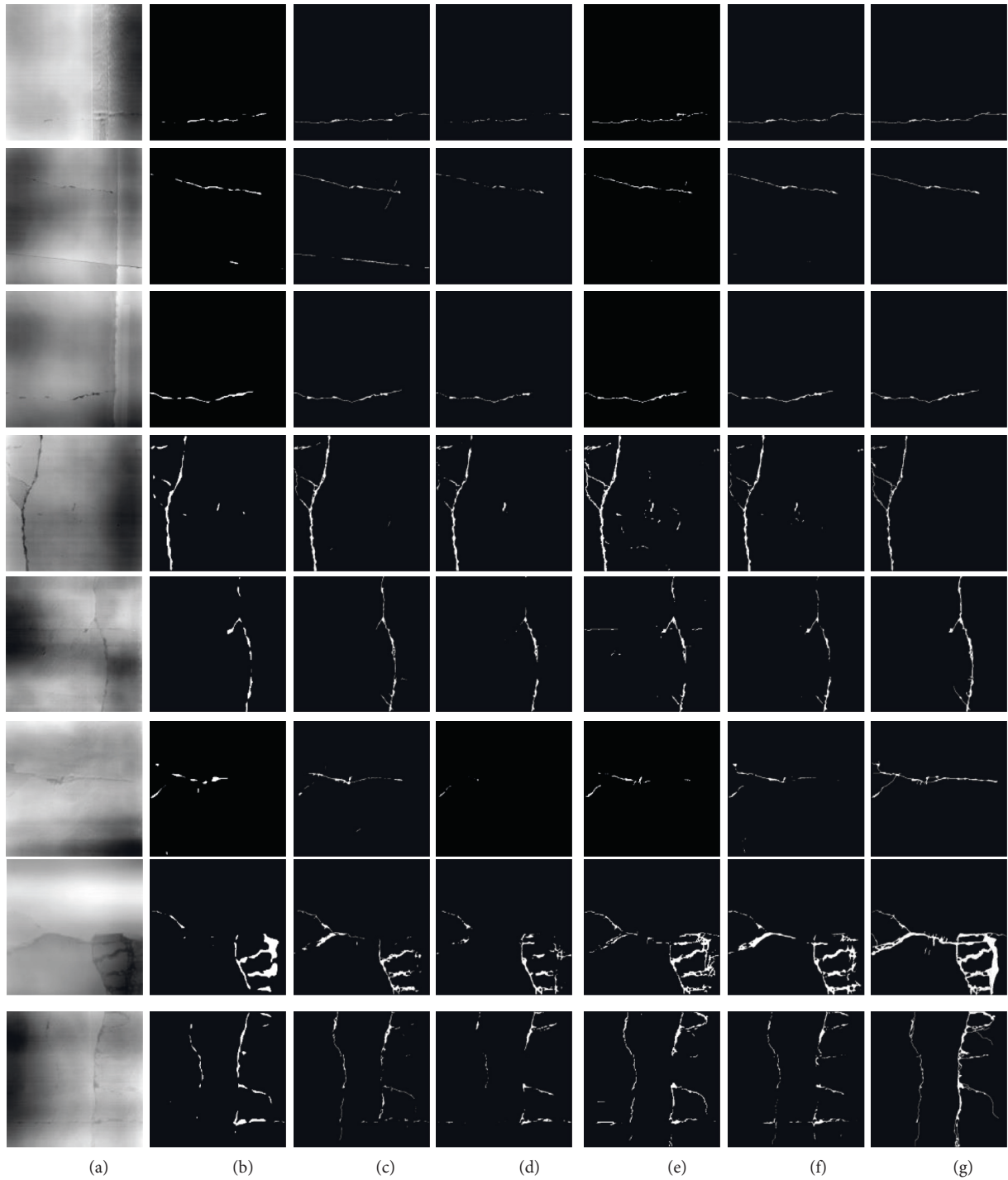


FIGURE 7: The segmentation results. (a) the original image, (b) the results of AutoEncoder, (c) the results of PSPNet, (d) the results of U-Net, (e) the results of KiU-net, (f) the results of GA-Unet, (g) ground truth of crack in pavement image.

Negative) means the number of crack pixel wrongly predicted to be pavement pixel.

5. Results and Discusses

To evaluate the performance of the proposed GA-Unet, the test dataset selected from the CPC dataset is applied to

evaluate the network. The following are the results and discussion of the experiment.

5.1. Learning Process Experiment. Figure 6 shows the result of U-Net and GA-Unet at different epochs. The effort of concrete pavement crack segmentation is improving and the

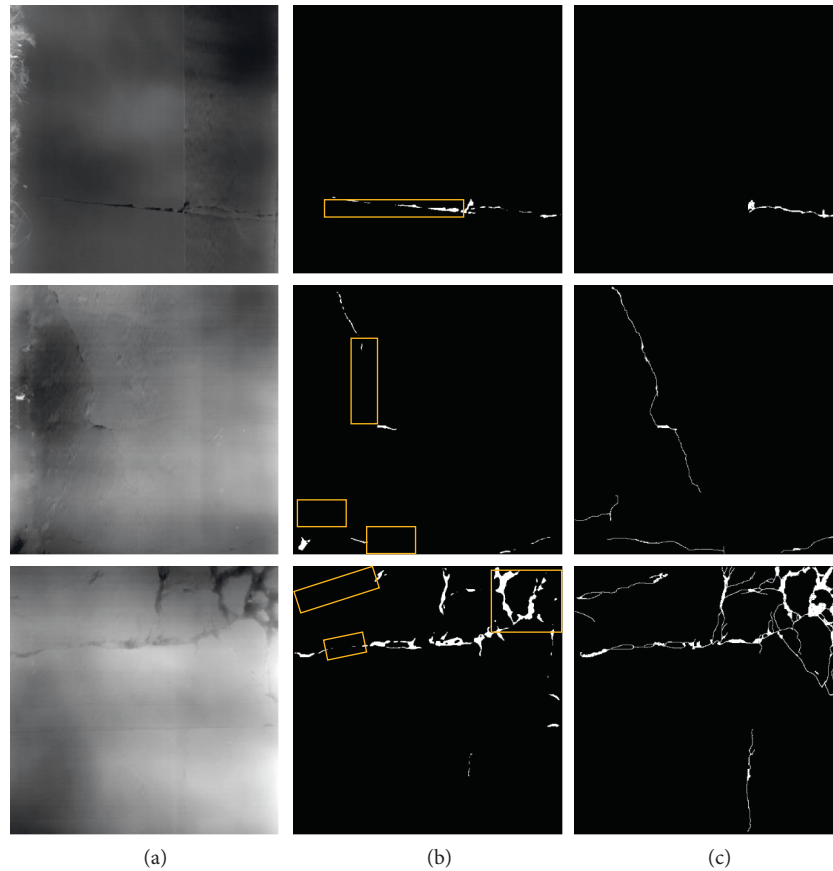


FIGURE 8: Typical errors resulting from PDSNet. (a) pavement images, (b) GA-UNet, and (c) ground truth.

results become closer to the ground truth with the epoch increasing regardless of the U-Net or GA-UNet. However, the GA-UNet is more accurate than U-Net for the same training epoch. The addition of the graph branch can improve the learning ability, enhance feature extraction capability, and boost the convergence speed.

5.2. Comparison Experiment. The comparison experiment between the AutoEncoder, PSPNet [37], U-Net [19], KiU-Net [38], and GA-UNet is conducted, and the results are illustrated in Table 1 and Figure 7. AutoEncoder is the simplest segmentation network with only an encoder and decoder structure. U-Net is the segmentation backbone in KiU-Net and GA-UNet. KiU-Net adds an over-complete representation branch based on U-Net to promote the performance. GA-UNet adds the graph network branch to enrich the feature represents. The U-Net can be regarded as the original semantic segmentation network compared to the GA-UNet. The comparison result between U-Net and GA-UNet can verify the validity of graph network branch. The performance is represented by four metrics, and the optimal results have been highlighted in bold in Table 1. GA-UNet achieves the optimal results in the metrics of F1, and IoU, which are 0.53 and 0.37, respectively. In addition, GA-UNet has a significant improvement in Recall, F1, and IoU metrics compare to the U-Net, which is increased by 0.12, 0.08 0.06. Although GA-UNet is weaker than U-Net in terms of

Precision and KiU-Net in terms of Recall, GA-UNet achieves better performance in segmenting cracks in concrete pavement in general. Figure 7 shows the comparison between the segmentation image of PSPNet, U-Net, and GA-UNet. The quality of the crack segmentation conducted by GA-UNet achieved better results than U-Net under different conditions.

5.3. Discussion. Convolutional computation is a common image processing method widely used in computer vision as a feature extractor for images. However, the convolutional network often uses the convolutional kernel with a small size (usually 3×3), which may lead to the problem of large and long object detection such as crack. The graph represents the relationship between nodes. Transforming the image into a graph can generate the connection between every region of the image. Then, the feature maps processed by the graph branch represent the relationships between regions and describe the characteristic of cracks at large scales, such as whether the cracks span multiple regions. This design is validated in Figure 7. Note the first four images are typical cases of single long crack segmentation, including transverse cracks and longitudinal cracks. The segmentation results by GA-UNet are more continuous than the U-Net model, which means the relationship between the regions extracted by the graph branch can enhance the detailed segmentation of continue cracks.

Moreover, it is impressive that adding a graph branch can improve the robustness of the network. The fifth and last two images show the results in light crack segmentation and the multiple cracks segmentation. The result of GA-Unet is significantly outstanding than the other methods. Although there is a disparity between the results of GA-Unet and the ground truth, the potential of adding graph branches has been validated experimentally.

However, compared to the ground truth, the GA-Unet can still be improved. In the first row of Figure 8, the concrete joints are identified as cracks, due to the similar feature between joint and crack. The joints can be considered as a separate category for detection to reduce the mistake of cracks. In the second row in Figure 8, the pixels of the shallow crack are ignored by GA-Unet method, and the same situation appears in the left crack in the last row. This indicates that the feature extraction branch in the GA-Unet is not effective enough in extracting shallow cracks, and the next step can be considered to enhance the feature of shallow cracks and improve the feature extraction branch. In the last row of Figure 8, the performance of GA-Unet is worse at the junction of shallow and heavy cracks and inside the severely broken area, which may be influenced by the deeper crack, and the accuracy of the surrounding shallow crack is inhibited, so we can consider the post-processing methods to make corrections, for example, using the CRF (Conditional Random Field) method to cluster the surrounding pixel with similar feature to improve the effect.

6. Conclusion

In this work, an end-to-end concrete pavement crack segmentation network called GA-Unet is proposed, for which a graph feature extraction branch is developed. The image can be processed as a graph through the graph generation method. The graph branch extracts the detailed graph features of the concrete pavement crack. The graph feature is fused with the image feature extracted by encoder structure, which is helpful to improve the continuity of crack segmentation. After the feature fusion, the new feature map is applied to the decoder to complete the segmentation.

The crack segmentation methods based on deep learning need sufficient data to support training. Hence, a concrete pavement 3D image dataset has been built. Furthermore, we evaluate our method on the dataset. The results of experiments prove that the graph branch can significantly improve the performance of crack segmentation based on the existing network.

Data Availability

All data and program files included in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. NSFC71871165) and the Fundamental Research Funds for the Central Universities (Grant no. TTS2021-03).

References

- [1] A. V. Cesar, Queiroz and Surhid Gautam, *Road infrastructure and economic development: some diagnostic indicators*, World Bank Publications, Washington, DC, USA, 1992.
- [2] L. Pellecuer, G. J. Assaf, and M. St-Jacques, "Life cycle environmental benefits of pavement surface maintenance," *Canadian Journal of Civil Engineering*, vol. 41, no. 8, pp. 695–702, 2014.
- [3] G. W. Flintsch and K. McGhee, *Quality management of pavement condition data collection*, Transportation Research Board, Washington, DC, USA, 2009.
- [4] A. Cubero-Fernandez, F. J. Rodriguez-Lozano, R. Villatoro, J. Olivares, and J. M. Palomares, "Efficient pavement crack detection and classification," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, pp. 1–11, 2017.
- [5] J. Laurent, M. Talbot, and M. Doucet, "Road surface inspection using laser scanners adapted for the high precision 3d measurements of large flat surfaces," in *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No. 97TB100134)*, pp. 303–310, IEEE, Ottawa, ON, Canada, May 1997.
- [6] S. Mathavan, K. Kamal, and M. Rahman, "A review of three-dimensional imaging technologies for pavement distress detection and measurements," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2353–2362, 2015.
- [7] K. C. P. Wang, Q. J. Li, G. Yang, Y. Zhan, and Y. Qiu, "Network level pavement evaluation with 1 mm 3d survey system," *Journal of Traffic and Transportation Engineering*, vol. 2, no. 6, pp. 391–398, 2015.
- [8] J. Wang and R. X. Gao, "Pavement distress analysis based on dual-tree complex wavelet transform," *International Journal of Pavement Research and Technology*, vol. 5, no. 5, p. 283, 2012.
- [9] Y. Zhang and H. Zhou, "Automatic pavement cracks detection and classification using radon transform," *Journal of Information and Computational Science*, vol. 9, no. 17, pp. 5241–5247, 2012.
- [10] A. Ayenu-Prah and N. Attoh-Okine, "Evaluating pavement cracks with bidimensional empirical mode decomposition," *EURASIP Journal on Applied Signal Processing*, vol. 1–7, 2008.
- [11] S. Sorncharean and S. Phiphobmongkol, "Crack detection on asphalt surface image using enhanced grid cell analysis," in *Proceedings of the 4th IEEE International Symposium on Electronic Design, Test and Applications (delta 2008)*, pp. 49–54, IEEE, Hong Kong, January 2008.
- [12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [13] A. Zhang, K. C. P. Wang, R. Ji, and Q. J. Li, "Efficient system of cracking-detection algorithms with 1-mm 3d-surface models and performance measures," *Journal of Computing in Civil Engineering*, vol. 30, no. 6, Article ID 04016020, 2016.
- [14] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, "Pavement distress detection and classification based on yolo network," *International Journal of Pavement Engineering*, vol. 22, no. 13, pp. 1659–1672, 2021.

- [15] Z. Liu, Y. Cao, Y. Wang, and W. Wang, "Computer vision-based concrete crack detection using u-net fully convolutional networks," *Automation in Construction*, vol. 104, pp. 129–139, 2019.
- [16] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016.
- [17] R. Stricker, D. Aganian, M. Sesselmann et al., "Road surface segmentation - pixel-perfect distress and object detection for road assessment," in *Proceedings of the International Conference on Automation Science and Engineering (CASE)*, Lyon, France, August 2021.
- [18] J. Long, E. Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.
- [19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science*, Springer, in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, October 2015.
- [20] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, "Automatic pixel-level crack detection and measurement using fully convolutional network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1090–1109, 2018.
- [21] Z. Qu, W. Chen, S.-Y. Wang, T.-M. Yi, and L. Liu, "A crack detection algorithm for concrete pavement based on attention mechanism and multi-features fusion," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2021.
- [22] B. Sanchez-Lengeling, E. Reif, P. Adam, B. Alexander, and Wiltschko, *A Gentle Introduction to Graph Neural Networks*, Distill, 2021, <https://distill.pub/2021/gnn-intro>.
- [23] J. Bruna, W. Zaremba, Arthur Szlam, and Y. LeCun, "Spectral Networks and Locally Connected Networks on Graphs," 2013, <https://arxiv.org/pdf/1312.6203>.
- [24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, <https://arxiv.org/abs/1609.02907>.
- [25] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, California, CA, USA, December 2017.
- [26] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proceedings of the 24th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining, KDD '18*, New York, NY, USA, 2018.
- [27] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proceedings of the International Conference on Machine Learning*, PMLR, South Korea, July 2016.
- [28] G. Huo, Y. Zhang, J. Gao, B. Wang, Y. Hu, and B. Yin, "Caegcn: cross-attention fusion based enhanced graph convolutional network for clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [29] J. Wang, Y. Zhang, Y. Wei, Y. Hu, X. Piao, and B. Yin, "Metro passenger flow prediction via dynamic hypergraph convolution networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7891–7903, 2021.
- [30] J. Laurent, D. Lefebvre, and E. Samson, "Development of a new 3d transverse laser profiling system for the automatic measurement of road cracks," in *Proceedings of the Symposium on Pavement Surface Characteristics, 6th*, Portoroz, Slovenia, October 2008.
- [31] J. Laurent, J. F. Hébert, D. Lefebvre, and Y. Savard, "Using 3d laser profiling sensors for the automated measurement of road surface conditions," in *Proceedings of the 7th RILEM International Conference on Cracking in Pavements*, pp. 157–167, Springer, Netherlands, June 2012.
- [32] R. Kaushik, J. Xiao, W. Morris, and Z. Zhu, "3d laser scan registration of dual-robot system using vision," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4148–4153, IEEE, St. Louis, MO, USA, October 2009.
- [33] A. Cohen, C. Zach, S. N. Sinha, and M. Pollefeys, "Discovering and exploiting 3d symmetries in structure from motion," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1514–1521, IEEE, Rhode Island, USA, June 2012.
- [34] A. Zhang, K. C. P. Wang, B. Li et al., "Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 805–819, 2017.
- [35] H. Lang, J. J. Lu, Y. Lou, and S. Chen, "Pavement cracking detection and classification based on 3d image using multi-scale clustering model," *Journal of Computing in Civil Engineering*, vol. 34, no. 5, Article ID 04020034, 2020.
- [36] P. Diederik, "Kingma and Jimmy Ba. Adam: a method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, May 2015.
- [37] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, Honolulu, HI, USA, July 2017.
- [38] Jeya Maria Jose Valanarasu, V. A. Sindagi, I. Hacihaliloglu, and V. M. Patel, "Kiu-net: towards accurate segmentation of biomedical images using over-complete representations," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 363–373, Springer, Lima, Peru, October 2020.