OUR-Net: A Multi-Frequency Network With Octave Max Unpooling and Octave Convolution Residual Block for Pavement Crack Segmentation

Pengtao Li, Meihua Wang, Zhun Fan, *Senior Member, IEEE*, Han Huang, *Senior Member, IEEE*, Guijie Zhu, and Jiafan Zhuang, *Member, IEEE*

***Abstract*— Cracks are among the most common, most likely, and earliest of all pavement distresses. Detecting and repairing cracks as early as possible can help extend the service life of pavements. However, Detecting cracks with precision can be challenging due to their varied structural characteristics and complex background interference. In this paper, a new convolutional neural network architecture, OUR-Net, is designed to more efficiently treat both high- and low-frequency visual image features. An Ocatve Convolution is incorporated into the proposed network as an enhancement to conventional convolution. In particular, an Octave Convolution Residual Block (OCRB) is embedded in the encoder to replace the convolutional layer of the classical encoder. Moerover, we propose Octave Max Unpooling (OMU) as the upsampling operation of the decoder, enabling the neural network to learn how to decode multi-spatial frequency features. Compared with models using traditional convolution, OUR-Net has better capability of processing multi-scale information, thus simultaneously improving model performance while saving computational costs by reducing spatial redundancy. We evaluate the superiority of the proposed method by comparing it to state-of-the-art crack segmentation methods on four public datasets (CrackLS315, CFD, Crack200, DeepCrack), which encompass cracks of various widths. Comprehensive experimental results reveal that the proposed method performs excellently, which achieves F1-score and mIoU of 0.9112, 0.9271, 0.8106, 0.9318, and 0.8369, 0.8644, 0.6815, 0.8723, respectively, on the four datasets. A lightweight version of the proposed network is constructed using depthwise separable convolution that achieves excellent performance with only 0.88M parameters.**

***Index Terms*— Crack segmentation, octave convolution, octave convolution residual block, octave max unpooling, multi-spatial frequency features.**

I. Introduction

CRACK is a typical infrastructure surface damage, frequently occuring on surfaces such as bridges, pavements, tunnels, metals, and dams. Among them, pavement cracks are typical engineering structural surface defects. Pavements are often subjected to fatigue stresses and cyclic loading, resulting in defects in the pavement structure. Cracks in the pavement reduce the local stiffness and thus lead to material discontinuity, which poses a significant threat to the service of the pavement and traffic safety. Appropriate pavement maintenance can extend pavement lifespan, reduce fuel consumption, and enhance roadway safety. Therefore, timely and accurate crack assessments are crucial for pavement maintenance. Manual inspections for pavement defects are inefficient, requiring significant labor and time, and can cause disruptions to traffic. As a result, automatic pavement crack detection using computer vision has gradually become a mainstream defect detection method with low cost and high accuracy, which can also effectively replace manual labor.

Since the rapid development of computer vision technology, several methods for detecting cracks automatically based on computer vision technology have been proposed. Kamaliardakani et al. [1] utilized a threshold-based technique to distinguish crack pixels from the background. In addition, methods such as edge detection [2], mathematical morphology [3], and minimal paths [4] are widely adopted as traditional image processing techniques. The crack detection also involves the use of filters, such as the Sobel [5] and Gabor filters [6]. Furthermore, support vector machine [7] and random forest [8], [9] were used to enhance crack segmentation accuracy. However, in real-world scenarios, the complexity of crack topology, various crack widths, weather-induced lighting conditions, skidproof stripes that closely resemble cracks,

Manuscript received 7 June 2023; revised 16 March 2024 and 19 April 2024; accepted 14 May 2024. This work was supported in part by the National Science and Technology Major Project under Grant 2021ZD0111501 and Grant 2021ZD0111502, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023B1515120020, in part by the National Natural Science Foundation of China under Grant 62176147 and Grant 62206064, and in part by the STU Scientific Research Foundation for Talents under Grant NTF22030. The Associate Editor for this article was H. Khayyam. *(Corresponding author: Meihua Wang.)*

Pengtao Li and Meihua Wang are with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou, Guangdong 510642, China (e-mail: lipengtao0524@163.com; wangmeihua@scau.edu.cn).

Zhun Fan is with the Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China, and also with the International Cooperation Base of Evolutionary Intelligence and Robotics of Guangdong Province, Shantou, Guangdong 515063, China (e-mail: zhun_fan@126.com).

Han Huang is with the School of Software Engineering, South China University of Technology, Guangzhou, Guangdong 510006, China (e-mail: hhan@scut.edu.cn).

Guijie Zhu and Jiafan Zhuang are with the Department of Electronic Engineering, Shantou University, Shantou, Guangdong 515063, China, and also with the International Cooperation Base of Evolutionary Intelligence and Robotics of Guangdong Province, Shantou, Guangdong 515063, China (e-mail: 16gjzhu@stu.edu.cn; jfzhuang@stu.edu.cn).

Digital Object Identifier 10.1109/TITS.2024.3405995

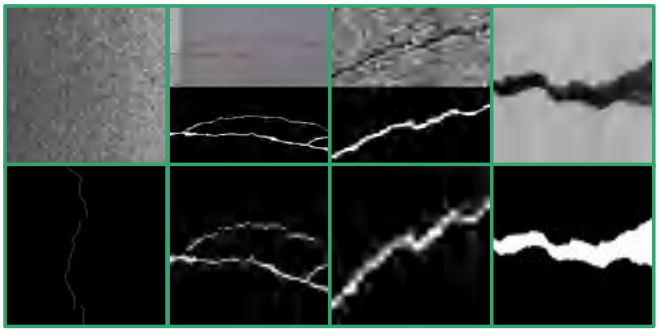


Fig. 1. Schematic diagram of cracks of different widths. From left to right, they are extremely thin crack, thin crack, thick crack and extremely thick crack.

and objects on the pavement, such as leaves, road markings, shadows, and maintenance hole covers, can make the task of crack segmentation extremely challenging.

Deep learning has proliferated recently duo to its robust feature representation capability. In particular, convolutional neural networks have achieved excellent performance in images. Inspired by deep learning, many groups [10], [11], [12], [13], [14], [15] have used convolutional neural networks to automatically extract deep crack features, enabling a breakthrough in crack detection and segmentation. Nevertheless, there are still several challenges that researchers and practitioners face in the crack segmentation task. A key issue that must be addressed is the significant imbalance between crack and non-crack pixels in crack images, which can cause the network to prioritize background information. To mitigate this issue, most existing methods adjust the loss function weights for the two categories. Although doing this can help improve model performance, detecting thin cracks with finer details still remains a challenge. Therefore, a finer segmentation of cracks requires more attention to the edge detail information of cracks and thin cracks, but at the same time, the main structure of cracks cannot be ignored. In order to retain more detailed information while extracting the main structure of cracks, DeepCrack [16] utilized SegNet [17] to fuse same-scale convolutional features in its encoder and decoder. Yang et al. [18] utilized a feature pyramid method to integrate contextual information for crack segmentation. Zhou et al. [19] developed ECDFFNet, a network that utilizes enhanced convolution and dynamic feature fusion to enhance performance. The DeepLabv3+ decoder by Sun et al. [20] incorporates a multi-scale attention module that utilizes attention masks to dynamically adjust the weights of high-level and low-level feature maps. Although multi-scale feature fusion can in general enhance model accuracy, most of these methods implemented it by simply stacking multi-level features. Therefore, the improvement of model performance is limited, which also generates more spatial redundancy, increases the consumption of computational resources, and prolongs the inference time.

As we can observe in an illustrative example shown in Figure 1, according to the width of the cracks, we can classify them into categories of extremely thin, thin, thick, and extremely thick ones, in which the extremely thin cracks are

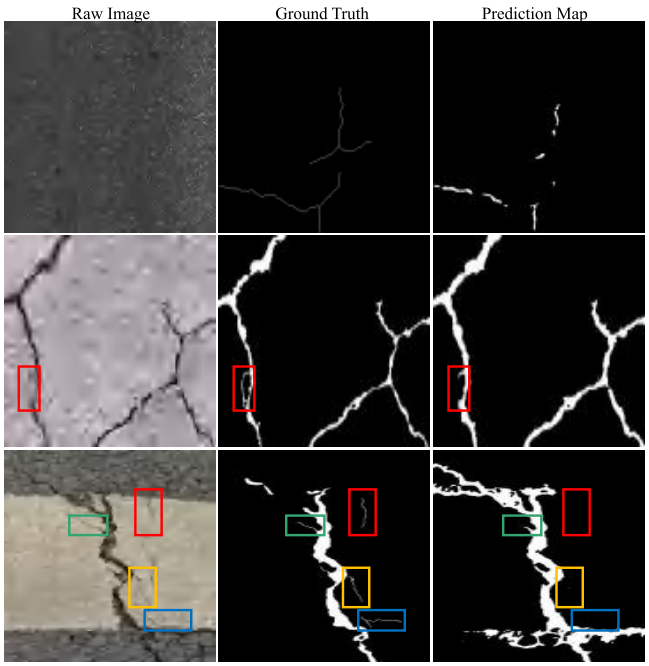


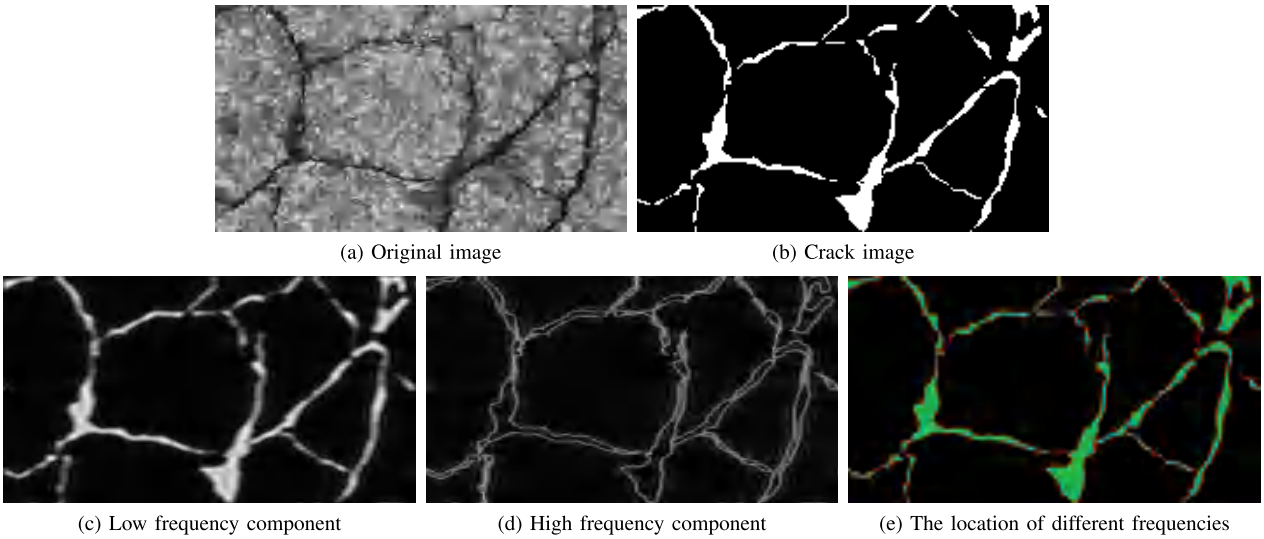
Fig. 2. Prediction maps of existing crack segmentation models. Only extremely thin cracks are present in the first row of the image. The second and third rows of images contain more than one type of crack. In the third column, the first row shows the prediction map for the method FPHBN [18], and the second and third rows show the prediction map for the method DMA-Net [20].

hardly observable by naked eyes. Few methods can simultaneously take into account cracks of various widths, and segment them all very well. As shown in Figure 2, when the image contains more than one type of cracks, the existing crack segmentation models tend to ignore the extremely thin ones.

Images in general can be divided into low- and high-spatial frequency components. Among them, the low-spatial frequency component describes smooth changes and the high-spatial frequency component describes the details of rapid changes [21], [22], [23], [24]. As shown in Figure 3, for a crack image, the low-spatial frequency component mainly relates to the backbone structure of the crack, while the high-spatial frequency component roughly relates to the edge details or thin cracks. Building on these observations, we proposed a novel method that implements multi-scale feature fusion by utilizing multi-frequency feature representation instead of simply stacking feature maps, which is capable of extracting and exploiting both high-frequency and low-frequency information of visual images in an effective and efficient way. Experimental results show that the proposed method improves the model accuracy while reducing the spatial redundancy, and achieves superior segmentation performance for cracks of different widths. This method is a multi-frequency network based on Octave Max Unpooling and Octave Convolution Residual Block called OUR-Net, where “O” represents the Octave Convolution used in both modules, “U” represents Octave Max Unpooling, and “R” represents Octave Convolution Residual Block.

In summary, this paper makes the following contributions.

- We proposed a new model called OUR-Net for crack segmentation, which basically applies Octave Convolution



(a) Original image (b) Crack image

(c) Low frequency component (d) High frequency component (e) The location of different frequencies

Fig. 3. (a) Original image. (b) The corresponding ground truth of the original image. (c) The low frequency component of (d) gained using low-pass filtering. (e) The high frequency component of (f) gained using high-pass filtering. (g) The crack image is decomposed into a low frequency component describing the backbone structure of the crack (green) and a high frequency component describing the edge details or thin cracks (red).

to replace the conventional convolution in the encoder and decoder building blocks of the classical SegNet architecture.
- In particular, we proposed an Octave Convolution Residual Block (OCRB) to be integrated in the OUR-Net to improve its performance.
- We also proposed Octave Max Unpooling (OMU) to replace the traditional up-sampling operation to enable the network to fully decode multi-frequency features and produce more accurate prediction results.

The remainder of this paper is structured as follows. Section II reviews the related work. Section III describes the proposed OUR-Net and the loss function used in detail. Section IV presents the experimental design and analyzes the experimental results. Section V concludes the paper.

II. Related Works

Pavement crack segmentation involves assigning each pixel in a pavement image to a binary class (crack or non-crack), making it a pixel-wise binary classification task. As computer vision technology has rapidly developed, the methods used for pavement crack segmentation can be classified into the following seven categories, among which the first six categories are traditional crack segmentation methods, which are sensitive to environmental noise.

A. Wavelet-Based Methods

Subirats et al. [25] used the continuous wavelet transform in three steps to determine whether cracks are present in the image. Chambon et al. [26] utilized a 2D matched filter to create a customized mother wavelet and applied it to a Markov Random Field process for crack segmentation. However, the anisotropic nature of wavelet-based methods makes them less effective in handling cracks with low continuity or high curvature.

B. Thresholding-Based Methods

In general, crack pixels have a lower grayscale value than non-crack pixels. Crack pixels can be extracted from the background by setting a reasonable threshold value. However, this thresholding-based method is sensitive to noise on the pavement, and a suitable threshold is difficult to find. Banharnsakun [27] first utilized a threshold-based approach to segment the crack images into distressed and non-distressed regions. Peng et al. [28] implemented an improved Otsu threshold segmentation algorithm and an improved adaptive iterative threshold segmentation algorithm to detect cracks in the runway surfaces of airport. Usually, researchers combined the threshold method with other methods to enhance the accuracy of crack segmentation.

C. Graph Theory-Based Methods

CrackTree, a fully automated crack detection method proposed by Zou et al. [29], segments cracks using a graph model, minimum spanning tree, and recursive edge pruning. This method can detect the location and shape of the cracks but is also sensitive to noises.

D. Edge Detection-Based Methods

A beam-based method was proposed by Ouyang and Wang [30] for extracting cracks, which perform well at low signal-to-noise ratios. Ayenu-Prah and Attoh-Okine [5] proposed a crack detection method utilizing a Sobel edge detector and bidimensional empirical mode decomposition. However, edge detection methods cannot distinguish between edges caused by cracks and those caused by changes in lighting or texture, which can lead to false positive problems.

E. Hand-Crafted Features-Based Methods

HOG [31] and LBP [32] are feature extractors that extract hand-crafted features from the crack patches, which are

later fed to the classifier for processing. The hand-crafted features-based methods require manual selection and design of image features, which requires a great deal of expertise and experience.

F. Minimal Path-Based Methods

Minimal path search is a branch of energy minimization methods that are frequently used in crack detection. Li et al. [33] extended the F* algorithm in two aspects and proposed FoSA for crack detection using a seed-growing strategy. According to Amhaz et al. [34], crack detection can be enhanced by selecting a set of minimal paths and introducing two post-processing steps.

G. Deep Learning-Based Methods

The rapid development of deep learning technology, especially deep convolutional neural networks(DCNNs), driven by advancements in computer hardware, has led to unprecedented achievements in the field of computer vision. DCNN-based methods have achieved breakthrough performance in structural health detection of road infrastructure [35], [36], [37], especially in pavement crack detection [38]. Cha et al. [39] proposed a structural visual detection method based on Fast Region-based Convolutional Neural Network (Faster R-CNN) in order to detect multiple types of damage simultaneously in quasi real-time. These DCNN-based methods are able to detect multiple damage types using bounding boxes. DCNN-based crack detection methods can efficiently capture the characteristics of cracks while effectively reducing noise interference on pavement images. Cha et al. [40] proposed a vision-based approach to detect concrete cracks without calculating defect features using a deep architecture of convolutional neural network. Deep learning-based crack detection tasks can be categorized into three types: image classification, object detection, and semantic segmentation. The image classification task is to determine whether the image or patch contains cracks [38], [41]. The object detection task is to localize where the crack is in the picture using the bounding box [42], [43]. The semantic segmentation task is to differentiate the pixels in an image into cracked pixels and non-cracked pixels based on a convolutional neural network [44], [45], [46], [47] or Transformer [48]. In particular, Kang et al. [49] provided a novel idea that firstly use bounding box to localize the crack region in the image and then segment the crack pixels in the detected crack region. In addition, AI-Huada et al. [50] proposed a hybrid deep learning pavement crack semantic segmentation method which is based on the knowledge transfer between class activation map (KTCAM) and encoder-decoder segmentation network (KTCAM-Net). Yang et al. [51] proposed a multiscale triple-attention network, MST-Net, for end-to-end pixel-level crack detection. These methods achieved improved results thanks to the proper features of the cracks extracted using deep convolutional neural networks.

III. Method

In this section, we first describe the general architecture of the proposed network, and then give detailed the multi-frequency feature representation based on OctConv, Octave Convolution Residual Block and Octave Max Unpooling. Finally, we elaborate on the loss function utilized.

A. Overall Architecture

Figure 4 illustrates the proposed architecture of OUR-Net. In this study, we introduce multi-frequency feature representation based on Octave Convolution (OctConv) [52], which aims to improve multi-frequency feature extraction and reduce spatial redundancy. We design a multi-frequency feature encoder based on Octave Convolution Residual Block (OCRB) to enable the network to encode multi-frequency features. A novel up-sampling operation called Octave Max Unpooling (OMU) is designed to provide the network with the capability of decoding multi-frequency features.

The multi-frequency feature encoder consists of five cascading encoder blocks. Each encoder block is made up of an OCRB (see Figure 5) that extracts both high- and low-frequency crack features and a particular down-sampling operation. Down-sampling operations like max-pooling, while increasing the receptive field and reducing the spatial resolution of the feature map, may at the same time ignore some detailed information and degrade the performance of fine-grained segmentation. We therefore proposed a particular down-sampling operation in this work, which is a max pooling operation performed separately for the high- and low-frequency components of the feature map, with the aim of enhancing crack segmentation performance. Thus, the particular down-sampling operation yield two max pooling indices, which will be utilized in the corresponding decoder up-sampling operation. In addition, the corresponding decoder is constructed according to the encoder. Each decoder block contains one OMU operation and two or three convolution blocks consisting of a 3×3 OctConv operation, a batch normalization operation (BN), and a rectified linear unit (ReLU), which correspond to the classical SegNet decoder structure. This encoder-decoder structure can capture smooth low-frequency components, like the backbone structure of a crack, as well as sharply changing high-frequency components, like details along the edges of a crack or thin cracks. A 1×1 convolutional layer follows the decoder's end to generate the crack prediction map.

B. Multi-Frequency Feature Representation Based On OctConv

The smooth structure of natural scene images can be represented by the low-spatial frequency components, while the fine details with rapid changes are represented by the high-spatial frequency components. In the case of crack images, the low-spatial frequency components represent the overall crack structure, and the high-spatial frequency components represent the edge details of the cracks or thin cracks. According to this characteristic, we speculate that utilizing multi-frequency feature representation [52] has a potential of increasing the effectiveness of crack segmentation. We utilize Octave Convolution (OctConv) [52] to achieve this multi-frequency feature representation that fully extracts the high- and low-frequency information of cracks. The diagram of

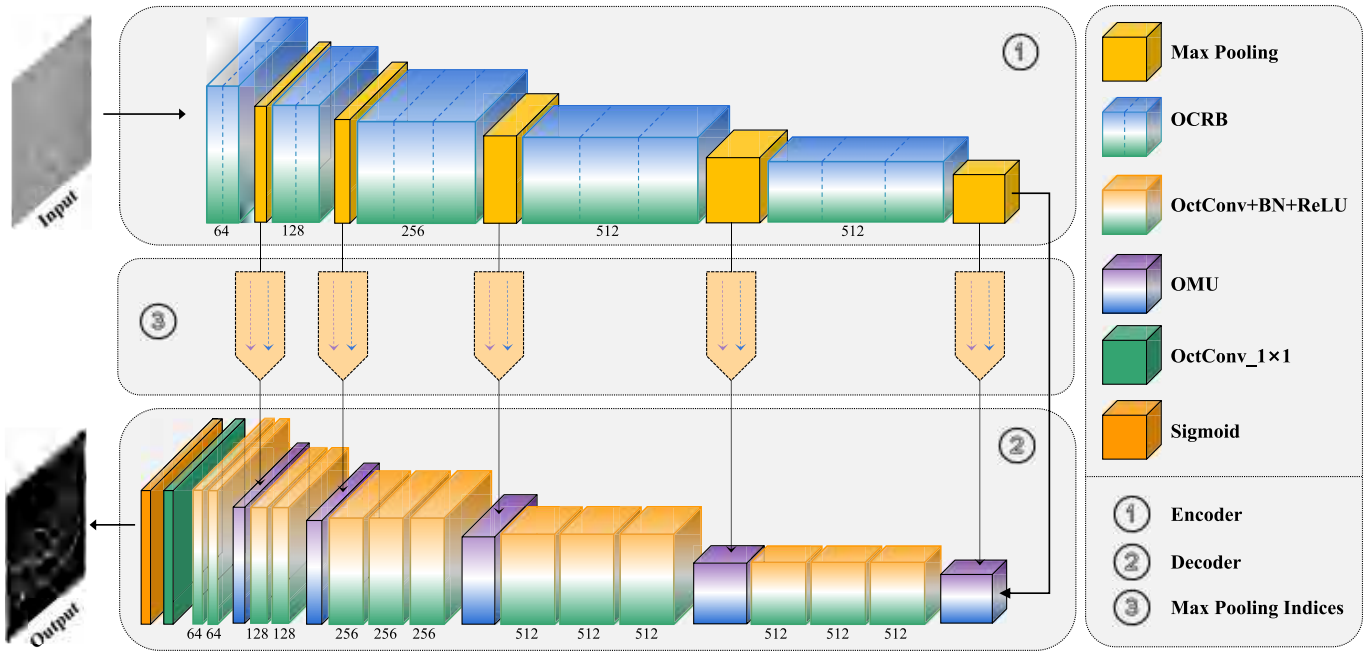


Fig. 4. The architecture of the proposed OUR-Net. The two colors in the module represent the high- and low-frequency components, respectively. The two different colored arrows in the third part represent the max pooling indices resulting from the max pooling operation on the high- and low-frequency components, respectively.

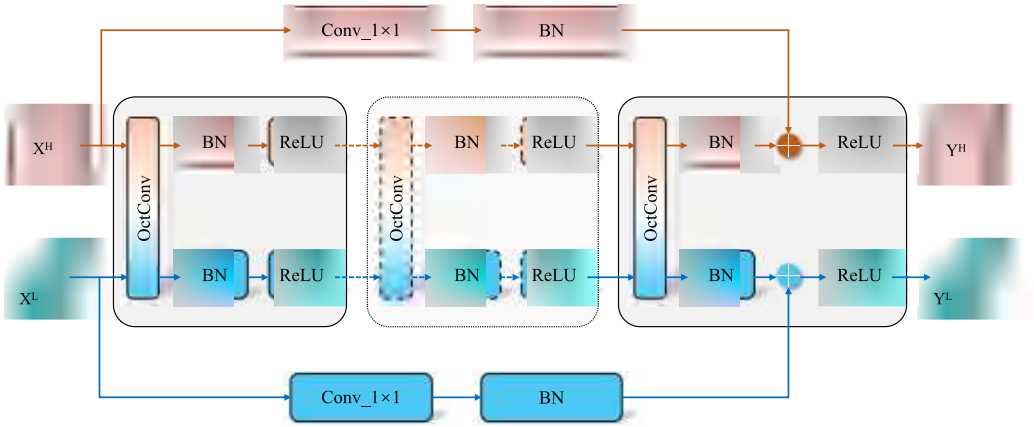


Fig. 5. The illustration of Octave Convolution Residual Block. The orange color represents the high-frequency component, and the blue color represents the low-frequency component.

OctConv operation is presented in Figure 6. The input feature map X is explicitly factorized into two components, X^H and X^L, along the channel dimension. The inputs for the high- and low-frequency feature maps are denoted by X^H and X^L, respectively. Compared to their high-frequency counterparts, the low-frequency feature maps have a frequency that is one octave lower. Then the high- and low-frequency feature maps of the output $Y = \{Y^H, Y^L\}$ will be given by

$$\begin{aligned} Y^H &= Y^{H\to H} + Y^{L\to H} \\ &= f^{H\to H}(X^H) + f^{L\to H}(X^L) \end{aligned} \tag{1}$$

$$\begin{aligned} Y^L &= Y^{L\to L} + Y^{H\to L} \\ &= f^{L\to L}(X^L) + f^{H\to L}(X^H) \end{aligned} \tag{2}$$

where $f^{H\to H}$ and $f^{L\to L}$ denote two regular convolutions which are utilized for intra-frequency information updates, and $f^{H\to L}$ and $f^{L\to H}$ denote a series of operations for inter-frequency information exchange. $f^{H\to L}$ involves both down-sampling and regular convolution operations, which folds the down-sampling of the feature map X^H into the regular convolution. $f^{L\to H}$ involves both up-sampling and regular convolution operations, which folds the up-sampling over the feature map X^L into the regular convolution.

C. Octave Convolution Residual Block

When extracting image features, the conventional encoder structure is implemented using multiple layers of progressive encoder blocks that consist of a series of regular convolutional operations, which lacks the capability of realizing multi-frequency feature representation. To address this issue, we propose the OCRB in this paper, as illustrated in Figure 5. OCRB uses OctConv instead of regular convolution

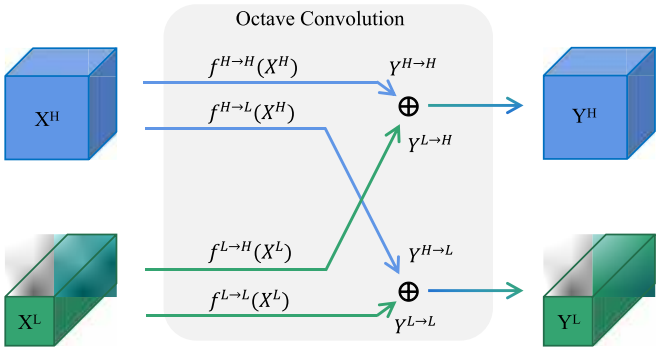


Fig. 6. The process of Octave Convolution. $f^{H\to H}$ and $f^{L\to L}$ represent intra-frequency information updates while $f^{H\to L}$ and $f^{L\to H}$ represent inter-frequency information exchange.

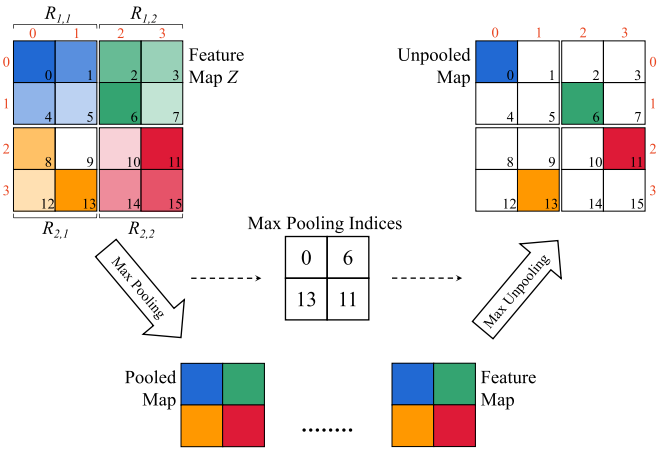


Fig. 7. The illustration of Max Pooling and Max Unpooling. The darker the color in the grid, the larger the value. The white color means the value is 0. The number in the lower right corner of each grid in the feature maps and unpooled maps represents its index.

to decompose the feature map into high- and low-frequency groups. Following OctConv, the feature maps of each group are processed by BN and ReLU, sequentially, to further extract the overall and detailed features of the cracks. Each OCRB is made up of two or three such combinations stacked in sequence, which corresponds to the encoder of the classical SegNet structure. In addition, the residual connection is adopted in each OCRB inspired by residual learning [53]. A residual block is considered as: $Y = \mathcal{F}(X, \{W_i\}) + X$, in which the function $\mathcal{F}(X, \{W_i\})$ represents the residual mapping being learned. In our method, residual mapping $\mathcal{F} = \{\mathcal{F}_H, \mathcal{F}_L\}$, $\mathcal{F}_H$ and $\mathcal{F}_L$ represent the residual mapping of the high-frequency group and low-frequency group, respectively. Then, an OCRB can be defined as:

$$\begin{cases} Y^H = \mathcal{F}_H\left(X^H, \{W_i^H\}\right) + X^H \\ Y^L = \mathcal{F}_L\left(X^L, \{W_i^L\}\right) + X^L \end{cases} \tag{3}$$

If the OCRB has two layers, $\mathcal{F} = \theta(W_2 \times \sigma(\theta(W_1 \times X)))$ in which θ and σ denote BN and ReLU, respectively. The addition operation is performed by element-wise addition. The dimensions of $\mathcal{F}$ and X are not necessarily the same. We can perform a regular convolution and BN operation to match the dimensions. Finally, the output is processed by ReLU. Thus, Eq. (3) can be rewritten as:

$$\begin{cases} Y^H = \sigma\left(\mathcal{F}_H\left(X^H, \{W_i^H\}\right) + \theta\left(W \times X^H\right)\right) \\ Y^L = \sigma\left(\mathcal{F}_L\left(X^L, \{W_i^L\}\right) + \theta\left(W \times X^L\right)\right) \end{cases} \tag{4}$$

D. Multi-Frequency Feature Decoding

Although multi-frequency feature representation can extract multi-frequency features of crack images, crack segmentation as a pixel-wise binary classification problem requires decoding the feature map to match the input image size. Therefore, a process is necessary that can recover spatial details from the encoded multi-frequency features and produce high-resolution prediction maps. We can simply conduct up-sampling operations for high-frequency and low-frequency features separately. However, doing this cannot allow inter-frequency information exchange, which may decrease the network's ability to recover spatial details of cracks. To overcome this limitation, we propose Octave Max Unpooling (OMU).

During the encoding stage, the feature map $Z \in \mathbb{R}^{h\times w}$ can be divided into a number of regions $R_{m,n}$, $1 \le m \le M, 1 \le n \le N$, as illustrated in Figure 7. Where M and N refer to the height and width of the pooled map obtained from the feature map Z after max pooling, respectively. For a region $R_{m,n}$, the maximum value of all items within the region is selected as the representation of this region. The max pooling indices $\mathcal{I} = \{i | 0 \le i \le h \times w - 1\}$ can be obtained after the max pooling operation, where i refers to the index of the max value within a region $R_{m,n}$ of the feature map Z. The $\mathcal{I} \in \mathbb{R}^{M\times N}$ is a matrix with the same size as the pooled map. In the corresponding decoding stage, the unpooling operation [54] utilizes the max pooling indices to up-sample the feature map $X \in \mathbb{R}^{M\times N}$ to generate the unpooled map $Y \in \mathbb{R}^{h\times w}$. Based on Figure 7 and the definition of the unpooling operation, we can summarize the formula for the unpooled map Y at position (p, q) as follows:

$$Y_{p,q} = \Psi_X(p,q) = \begin{cases} X_{\lfloor \frac{p}{2} \rfloor, \lfloor \frac{q}{2} \rfloor}, & pw + q = \mathcal{I}_{\lfloor \frac{p}{2} \rfloor, \lfloor \frac{q}{2} \rfloor} \\ 0, & else \end{cases} \tag{5}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation.

Our goal in designing OMU is not only to decode multi-frequency features, but also to efficiently process inter-frequency information exchange, as shown in Figure 8. X^H and X^L refer to the high- and low-frequency feature maps of the input, respectively. The output $Y = \{Y^H, Y^L\}$ of OMU can be decomposed into high- and low-frequency feature maps, represented by $Y^H = Y^{H\to H} + Y^{L\to H}$ and $Y^L = Y^{L\to L} + Y^{H\to L}$, respectively. Here, $Y^{H\to H}$ and $Y^{L\to L}$ correspond to intra-frequency information updates, while $Y^{H\to L}$ and $Y^{L\to H}$ correspond to inter-frequency information exchange. To compute the high-frequency feature map Y^H, we use a max unpooling operation to update intra-frequency information, while inter-frequency information exchange is achieved by folding the up-sampling of the max unpooling operation applied to the low-frequency input feature map X^L into the

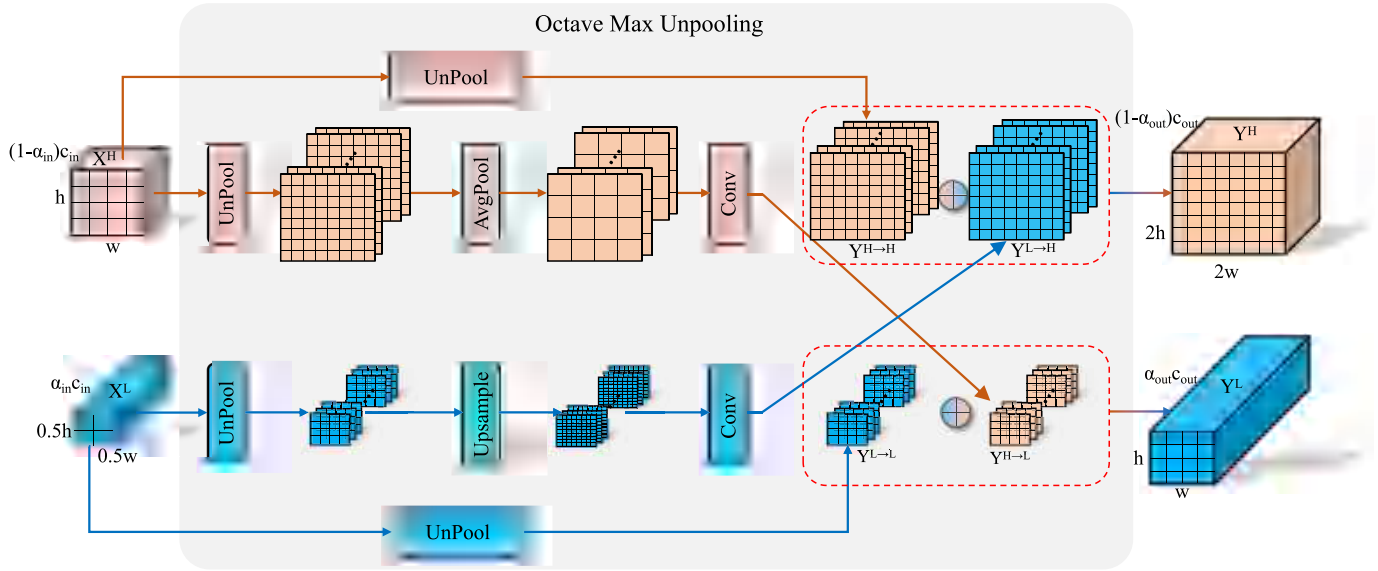


Fig. 8. The illustration of Octave Max Unpooling. Orange indicates high-frequency components, and blue indicates low-frequency components. The plus sign indicates the pixel-wise sum operation.

regular convolution process as follows:

$$\begin{aligned} Y_{p,q}^{H} &= Y_{p,q}^{H\to H} + Y_{p,q}^{L\to H} \\ &= \Psi_{X^H}(p,q) \\ &\quad + \sum_{i,j\in\Lambda_k} W^{\top}_{i+\frac{k-1}{2},j+\frac{k-1}{2}} \Psi_{X^L}\left(\lfloor\tfrac{p+i}{2}\rfloor, \lfloor\tfrac{q+j}{2}\rfloor\right) \end{aligned} \tag{6}$$

where (p,q) denotes the location coordinate and $\Lambda_k = \left\{(i,j)\,|i = \left\{-\frac{k-1}{2},\cdots,\frac{k-1}{2}\right\}, j = \left\{-\frac{k-1}{2},\cdots,\frac{k-1}{2}\right\}\right\}$ defines a local neighborhood, in which k denotes the size of the convolution kernel. In this work, $k = 3$. Similarly, for the low-frequency feature map Y^L, we also use the max unpooling to compute intra-frequency information update, while to exchange inter-frequency information, the average pooling of the max unpooling of the high-frequency input feature map (X^H) is folded into the regular convolution process as follows:

$$\begin{aligned} Y_{p,q}^{L} &= Y_{p,q}^{L\to L} + Y_{p,q}^{H\to L} \\ &= \Psi_{X^L}(p,q) \\ &\quad + \sum_{i,j\in\Lambda_k} W^{\top}_{i+\frac{k-1}{2},j+\frac{k-1}{2}} \\ &\quad \Psi_{X^H}(2(p+i)+0.5, 2(q+j)+0.5) \end{aligned} \tag{7}$$

where the location (p,q) is multiplied by a factor 2 to perform down-sampling. Because the down-sampling operation here is an average pooling that averages all four adjacent positions, the location is further shifted by half step to achieve the pooled maps are well aligned with the input.

In addition, α denotes the ratio of the number of channels assigned to low-frequency feature maps, where α ranges from 0 to 1. If not explicitly mentioned in this paper, the hyper-parameter α is set to 0.5 by default. Furthermore, in OMU, $\alpha_{in} = \alpha_{out} = \alpha$, where α_{in} and α_{out} denote the ratio of the number of channels of low-frequency feature maps in the input and output feature maps, respectively.

E. Loss Function

The pavement crack segmentation can be formulated as a pixel-wise binary classification problem, where each pixel in the crack image is classified as either a crack pixel or non-crack pixel. However, there is an extreme imbalance in the number of crack pixels and non-crack pixels in the crack image. In this paper, we utilize the loss function that combines weighted binary cross-entropy loss and dice loss to address this issue.

1) Weighted Binary Cross-Entropy: The training set containing M images is set to $S = \{(X^m, Y^m), m = 1,\cdots,M\}$, where $X^m = \left\{x_i^{(m)} | i = 1,\cdots,|X^m|\right\}$ refers to the crack image and $Y^m = \left\{y_i^{(m)} | i = 1,\cdots,|Y^m|;\, y_i^{(m)} \in \{0,1\}\right\}$ refers to the corresponding ground-truth crack map. To make the notation more straightforward, we will next omit the superscript m. The training aims to enable the network to generate prediction maps that approximate ground-truth. Additionally, we define W as the learnable parameters for the entire network. Then the definition of the weighted cross-entropy loss as follows:

$$\begin{aligned} L_{WBCE} = &- \sum_{i\in Y^+} w_p \log P_i\,(y_i = 1|X;W) \\ &- \sum_{i\in Y^-} \log P_i\,(y_i = 0|X;W) \end{aligned} \tag{8}$$

where Y^+ and Y^- refer to crack pixels and non-crack pixels in the ground-truth image, respectively. Then P_i denotes the probability that pixel i in the crack prediction map is a crack pixel or a non-crack pixel. Additionally, w_p denotes the weight allocated to the crack pixels, which when given a larger value allows the network to have more emphasis on the crack pixels during the training process. Let G_N^+ and G_N^- represent the number of crack pixels and non-crack pixels in the entire training set, respectively. In this work, we set $w_p = \frac{G_N^-}{G_N^+}$ for

TABLE I
ALL THE DATASETS

Dataset	Size	Number of Training	Number of Testing	Total Number
CrackLS315	512×512	252	63	315
CFD	480×320	94	24	118
Crack200	640×352	899	225	1124
DeepCrack	544×384	300	237	537

the sake of solving the class imbalance problem mentioned above.

2) Dice Loss: To calculate the similarity of two sets, the dice coefficient, which measures ensemble similarity, is typically employed. A larger dice coefficient indicates that the sets are more similar and vice versa. Accordingly, the dice loss in this work is formulated as follows:

$$L_{Dice} = 1 - \frac{2\sum_{n=1}^{N} p_n y_n + \varepsilon}{\sum_{n=1}^{N} p_n + \sum_{n=1}^{N} y_n + \varepsilon} \tag{9}$$

where N refers to the total number of pixels in crack image, p represents the prediction of network, y represents the ground-truth of the crack. $\varepsilon = 1e-5$, which prevents exceptions caused by a denominator of 0.

Finally, the weighted cross-entropy loss and the dice loss are integrated, yielding the total loss as follows:

$$L = \beta L_{WBCE} + \gamma L_{Dice} \tag{10}$$

where β and γ denote the weights assigned to the weighted cross-entropy loss and dice loss, respectively. In this work, $\beta = \gamma = 1$.

IV. Experiment and Results

A. Implementation Details

All experiments in this paper are performed on two NVIDIA GeForce GTX 3060 GPUs separately. PyTorch, a publicly available deep learning framework, is utilized to implement the proposed network. During the training phase, we utilize RAdam with a weight decay of 0.0002 as the optimizer to update the network parameters, while also setting the batch size to 8. A systolic schedule of $\eta_i = 0.9\eta_{i-1}$ is adopted for the current learning rate η_i after 10 periods of loss value saturation, with 1e-5 as the minimum learning rate. In addition, data augmentation operations are used, which include horizontal and vertical flip, random rotation, random affine transformation, and random adjustment of brightness, contrast, gamma, and saturation. Each operation is triggered with a 50% probability.

B. Datasets

In this paper, we validate the effectiveness of the proposed method using four datasets named CrackLS315, CFD, Crack200, and DeepCrack, respectively, as shown in Table I. These four datasets are described in detail as follows.

1) CrackLS315 [16]: This dataset contains 315 images of road surfaces taken under laser illumination. As the training set, we utilize 252 images and as the test set, 63 images. Since the cracks in the image are extremely thin, we denote the cracks in the CrackLS315 dataset as extremely thin cracks.

2) CFD [8]: It contains 118 manually labeled color images with a size of 480 × 320. There is noise in these images, like shadows, oil spots, and water stains, posing a challenge for crack segmentation. This dataset was captured in Beijing with mobile phones. A total of 94 images are utilized as training images and 24 images are utilized as test images in this dataset. Since the cracks in the CFD dataset are in general slightly thicker than the cracks in the CrackLS315 dataset, we call the cracks in the CFD dataset as thin cracks.

3) Crack200 [18]: Yang et al. [18] propose a dataset called Crack500, which contains 200 images as the test set, to validate their proposed model. We use the test set of Crack500 as our dataset and rename it Crack200. Each image is divided into 16 non-overlapping regions and regions containing fewer than 1000 pixels of cracks are filtered out. Through this operation, Crack200 consists of 1124 images. To facilitate training, each image was cropped again to 640 × 352 pixels. Then, this dataset was divided into 899 training images and 225 testing images. Since the cracks in the Crack200 dataset are normally thicker, we refer to the cracks in the Crack200 dataset as thick cracks.

4) DeepCrack [55]: This dataset contains 537 manually annotated RGB color images. Each image corresponds to a mask that precisely covers the crack regions. They all have a size of 544 × 384 pixels. This dataset was divided into 300 training images and 237 testing images. This dataset consists of various scenes, textures, and crack scales. And the crack widths exhibit a wide range, spanning from 1 to 180 pixels. Therefore, the DeepCrack dataset contains both thick cracks and extremely thick cracks.

C. Comparison Methods

1) FCN: FCN [56] is based on the VGG network, which defines a skip architecture to combine deep-level semantic information with shallow-level appearance information.

2) HED: HED [57] based on VGG16 is a breakthrough in edge detection. It utilize fully convolutional neural networks and deeply supervised networks to facilitate image-to-image prediction.

3) U-Net: U-Net [58] has achieved a significant breakthrough in medical image segmentation. Both its skip connection structure and encoder-decoder architecture are widely used in image segmentation tasks.

4) DeepCrack_Zou: Based on the encoder-decoder architecture of SegNet, Zou et al. [16] develop DeepCrack to integrate convolutional features of the same scale generated by the encoder and decoder. We denote it as DeepCrack_Zou.

5) DeepCrack_Liu: Based on HED, Liu et al. [55] propose the DeepCrack network, and the final crack detection results are refined by guided filtering and conditional random field methods. We denote it as DeepCrack_Liu.

6) FPHBN: Based on HED, FPHBN [18] integrates contextual information into low-level features through a feature

pyramid module and uses a hierarchical boosting module to balance loss among easy and hard samples.

7) ECDFFNet: Using enhanced convolution and dynamic feature fusion, ECDFFNet [19] achieves enhanced performance in capturing long-range dependencies and paying attention to local details.

8) DMA-Net: DMA-Net [20] enhances DeepLabv3+ and the decoder of DeepLabv3+ is enriched by a multi-scale attention module, in which feature maps of high-level and low-level are dynamically weighted.

D. Performance Evaluation Criteria

In this study, Precision(PR), Recall(RE), and F1-score($F1$), and mean Intersection over Union($mIoU$) are applied to assess the proposed model's performance, which are calculated as follows:

$$PR = \frac{TP}{TP + FP} \tag{11}$$

$$RE = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = \frac{2 \times PR \times RE}{PR + RE} \tag{13}$$

$$mIoU = mean\left(\frac{TP}{TP + FP + FN}\right) \tag{14}$$

where TP, FP, and FN represent the number of true positives, false positives, and false negatives, respectively. Taking into account the balance of PR and RE, $F1$ represents the harmonic mean of the two measures. As cracks have a certain width, a crack pixel etected within 2 pixels of the ground-truth is considered a true positive.

E. Experimental Results

1) Results on CFD: As shown in Figure 9(a), the curve obtained by our method is closest to the upper right corner of the chart, so we have obtained the highest precision and recall values on the CFD dataset. Out of all the compared methods, FCN performs the worst. The quantitative results in Table II show that our method achieves a precision, recall, F1-score, and mIoU of 0.9269, 0.9276, 0.9273, and 0.8644, respectively, on the CFD dataset, which outperforms the other methods. Compared with FCN, HED, U-Net, DeepCrack_Zou, DeepCrack_Liu, FPHBN, ECDFFNet, and DMA-Net, our method improves the performance of F1-score by percent of 12.17, 6.49, 1.36, 6.25, 2.84, 4.40, 0.46 and 2.89, respectively, and on mIoU by percent of 18.99, 10.63, 2.33, 10.26, 4.81, 7.34, 0.8 and 4.89, respectively.

2) Results on Crack200: It is clear from Figure 9(b) that our method outperforms all the other methods on the Crack200 too, among which U-Net and DeepCrack_Liu show the poorest performance. The data in the Table III shows that the F1-score (0.8106) and mIoU (0.6815) of our method are better than the other compared methods. DeepCrack_Liu scores the lowest F1-score and mIoU of 0.6769 and 0.5116. Compared to FCN, HED, U-Net, DeepCrack_Zou, DeepCrack_Liu, FPHBN, ECDFFNet, and DMA-Net, there are 8.7, 7.74, 12.04, 4.08, 13.37, 1.52, 2.55 and 0.62 percent of improvement on F1-score by our method, respectively. In addition, there are 11.46, 10.27, 15.46, 5.57, 16.99, 2.12, 3.53 and 0.86 percent of improvement on mIoU.

TABLE II
QUANTITATIVE EVALUATION ON CFD

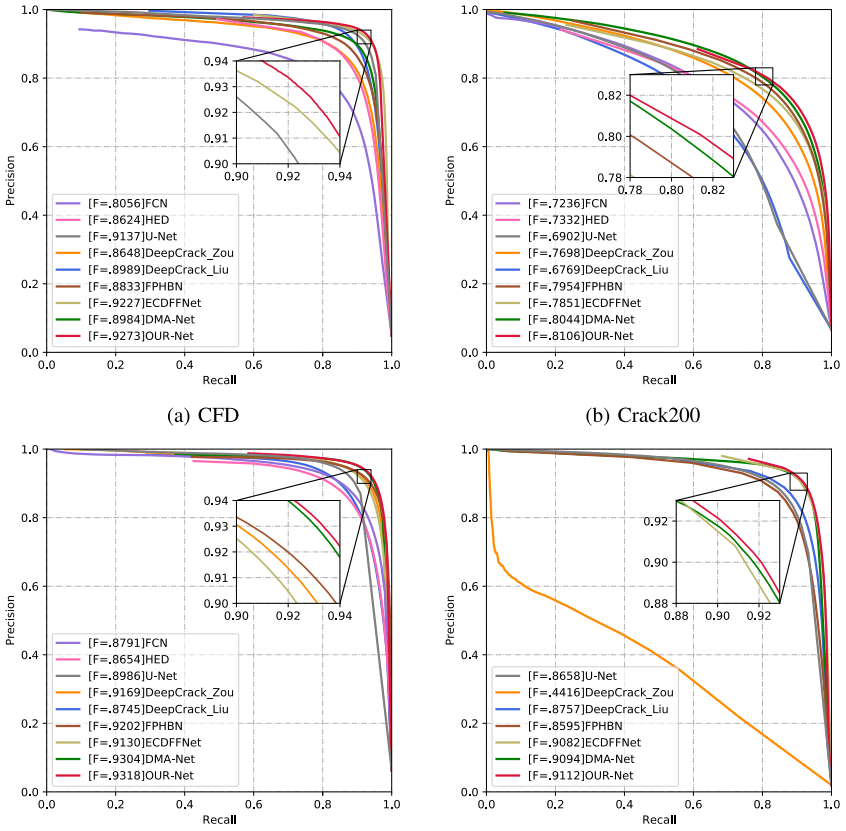
Method	Pr	Re	F1	mIoU
FCN [CVPR'2015]	0.7923	0.8194	0.8056	0.6745
HED [ICCV'2015]	0.8674	0.8574	0.8624	0.7581
U-Net [MICCAI'2015]	0.9118	0.9157	0.9137	0.8411
DeepCrack_Zou [TIP'2018]	0.8565	0.8733	0.8648	0.7618
DeepCrack_Liu [Neuro'2019]	0.9179	0.8806	0.8989	0.8163
FPHBN [TITS'2019]	0.8869	0.8797	0.8833	0.7910
ECDFFNet [TITS'2022]	0.9162	**0.9292**	0.9227	0.8564
DMA-Net [TITS'2022]	0.8926	0.9042	0.8984	0.8155
Ours	**0.9269**	0.9276	**0.9273**	**0.8644**

TABLE III
QUANTITATIVE EVALUATION ON CRACK200

Method	Pr	Re	F1	mIoU
FCN [CVPR'2015]	0.7039	0.7445	0.7236	0.5669
HED [ICCV'2015]	0.697	0.7733	0.7332	0.5788
U-Net [MICCAI'2015]	0.7108	0.6707	0.6902	0.5269
DeepCrack_Zou [TIP'2018]	0.7436	0.798	0.7698	0.6258
DeepCrack_Liu [Neuro'2019]	0.6774	0.6764	0.6769	0.5116
FPHBN [TITS'2019]	0.7702	0.8223	0.7954	0.6603
ECDFFNet [TITS'2022]	0.7435	0.8315	0.7851	0.6462
DMA-Net [TITS'2022]	0.7736	0.8378	0.8044	0.6729
Ours	**0.7742**	**0.8506**	**0.8106**	**0.6815**

3) Results on DeepCrack: As shown in Figure 9(c), our method achieves the best performance over other comprared methods on the DeepCrack dataset. The performance of DMA-Net follows closely behind and is on par with our method, achieving a significant lead over other methods. DeepCrack_Zou, FPHBN, and ECDFFNet perform similarly and rank in the second tier. From the Table IV, it can be concluded that our method obtain the best F1-score and mIoU of 0.9318 and 0.8723 on the DeepCrack dataset. Although the F1-score and mIoU of DMA-Net also reach 0.9304 and 0.8698, it is slightly lower than ours. The mIoU of FCN, HED, U-Net, DeepCrack_Zou, DeepCrack_Liu, FPHBN, ECDFFNet, and DMA-Net are lower than our method by percent of 8.8, 10.95, 5.65, 2.58, 9.54, 2.01, 3.24 and 0.25, respectively. The F1-scores are lower than our method by percent of 5.27, 6.64, 3.32, 1.49, 5.73, 1.16, 1.88 and 0.14.

4) Results on CrackLS315: The cracks in the CrackLS315 dataset images are extremely thin, which causes the task of segmenting the cracks so difficult that FCN and HED fail to identify the cracks. As shown in Figure 9(d), our method, DMA-Net, and ECDFFNet perform approximately the same, which as a group is far superior to the other compared methods. It is notable to point out that our method still outperforms the others in this group. Among the remaining methods, the performance of DeepCrack_Zou is the worst and far below



(a) CFD (b) Crack200

(c) DeepCrack (d) CrackLS315

Fig. 9. Precision-Recall curves on the four datasets.

TABLE IV
QUANTITATIVE EVALUATION ON DEEPCRACK

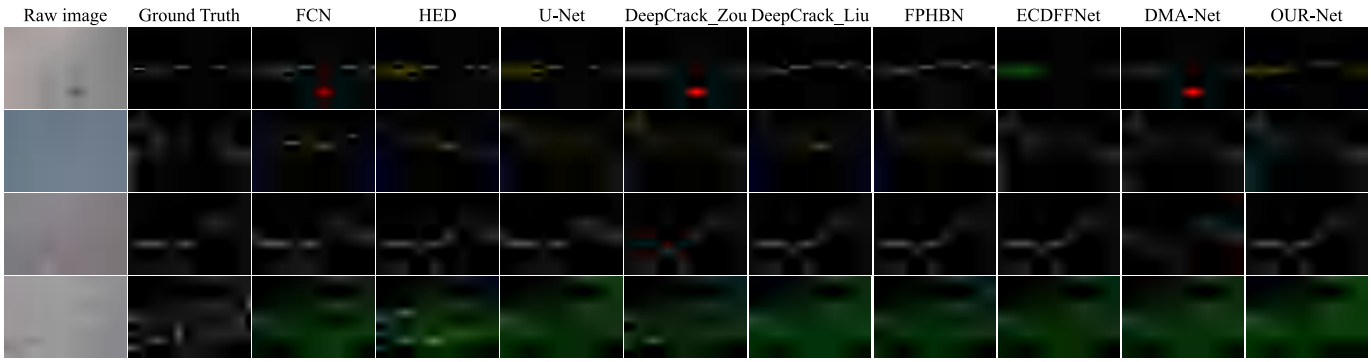
Method	Pr	Re	F1	mIoU
FCN [CVPR'2015]	0.8761	0.8821	0.8791	0.7843
HED [ICCV'2015]	0.8633	0.8676	0.8654	0.7628
U-Net [MICCAI'2015]	0.9137	0.884	0.8986	0.8158
DeepCrack_Zou [TIP'2018]	0.9144	0.9193	0.9169	0.8465
DeepCrack_Liu [Neuro'2019]	0.8891	0.8603	0.8745	0.7769
FPHBN [TITS'2019]	0.9142	0.9263	0.9202	0.8522
ECDFFNet [TITS'2022]	0.9171	0.9089	0.9130	0.8399
DMA-Net [TITS'2022]	0.9328	0.9280	0.9304	0.8698
Ours	**0.9339**	**0.9297**	**0.9318**	**0.8723**

TABLE V
QUANTITATIVE EVALUATION ON CRACKLS315

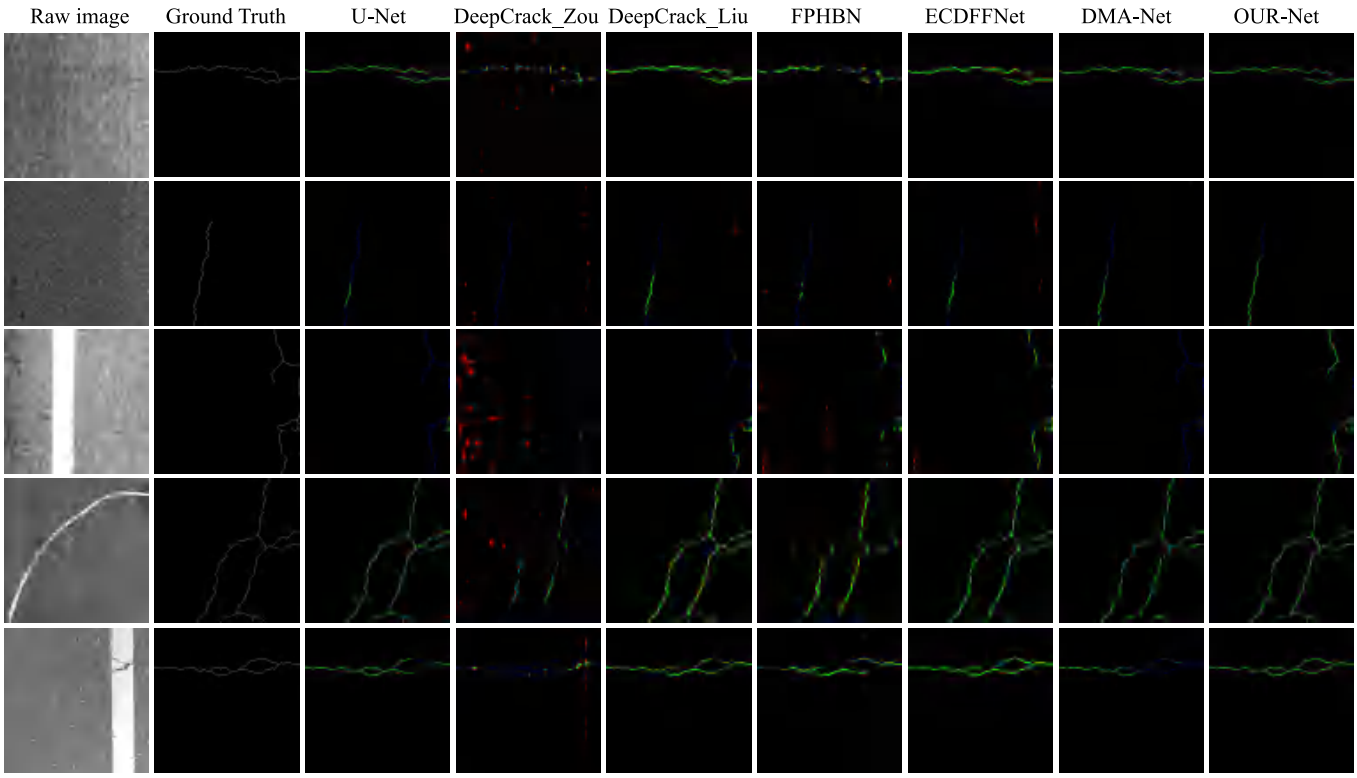
Method	Pr	Re	F1	mIoU
U-Net [MICCAI'2015]	0.8719	0.8598	0.8658	0.7634
DeepCrack_Zou [TIP'2018]	0.4115	0.4764	0.4416	0.2833
DeepCrack_Liu [Neuro'2019]	0.8716	0.8798	0.8757	0.7789
FPHBN [TITS'2019]	0.8751	0.8444	0.8595	0.7536
ECDFFNet [TITS'2022]	0.9079	0.9085	0.9082	0.8318
DMA-Net [TITS'2022]	0.9106	0.9082	0.9094	0.8339
Ours	**0.9138**	**0.9086**	**0.9112**	**0.8369**

the other methods. As shown in the Table V, the F1-score and mIoU of our method can still reach 0.9112 and 0.8369 on the extremely thin crack dataset, outperforming DMA-Net and ECDFFNet by a small margin. Furthermore, our method has a higher F1-score than those of U-Net, DeepCrack_Liu, and FPHBN by percent of 4.54, 3.55, and 5.17, respectively, and even higher value than that of DeepCrack_Zou by percent of 46.96. Similarly, the same is the case for mIoU.

According to Figure 9 and the quantitative results on the four datasets, it can be concluded that FCN, HED, and U-Net perform relatively poorly as general-purpose image segmentation methods. In contrast, other methods proposed specifically for crack segmentation perform better. The proposed method, U-Net, DeepCrack_Zou, and DMA-Net achieve better performance than FCN and HED, indicating that a decoder network can enhance the crack segmentation accuracy. Zou et al. [16] have demonstrated that multi-scale feature fusion can effectively improve crack segmentation accuracy. Therefore all comparison methods use multi-scale feature fusion. However, most of them stack multi-level features together to reduce the loss of crack details so that more redundant information is retained. The proposed method outperforms all the compared methods, which indicates that using multi-frequency feature representation to fuse multi-scale features is more effective than stacking multi-level features. For ECDFFNet and DMA-Net, ECDFFNet only adopts multi-scale feature fusion, while



(a) CFD

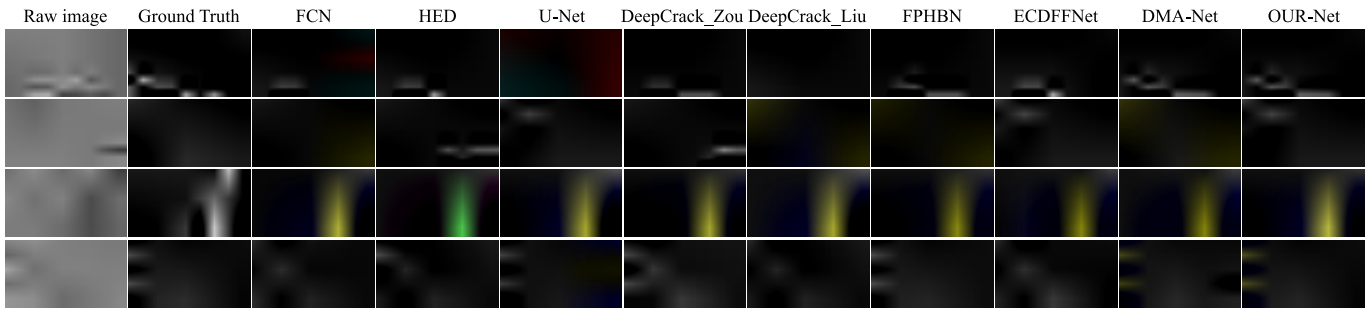


(b) CrackLS315

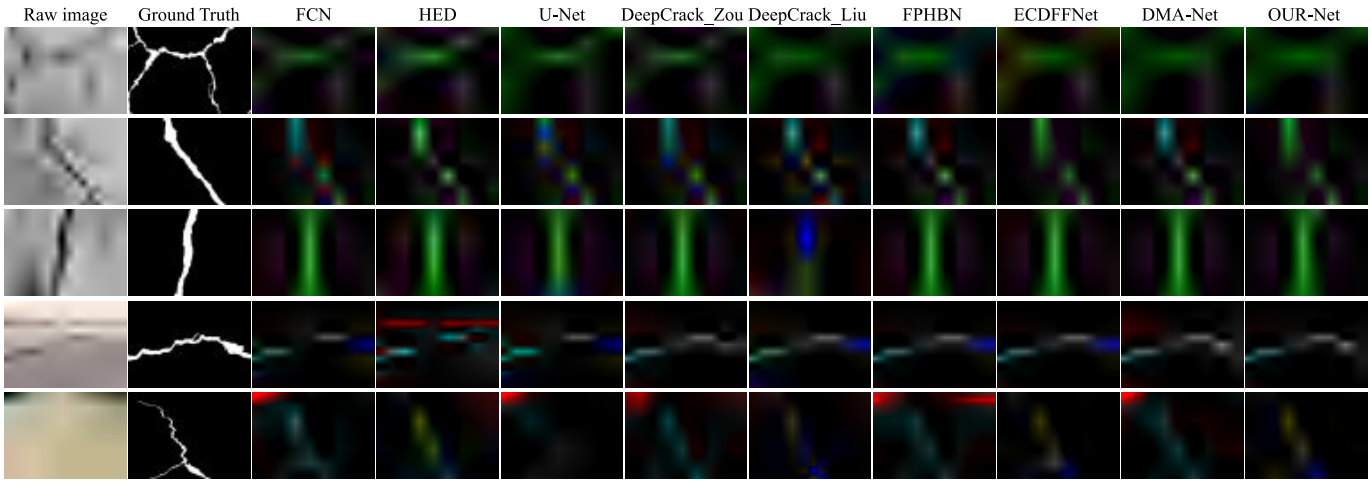
Fig. 10. Comparison results of different methods on CFD and CrackLS315 datasets. The results of the comparison are plotted as green for true positives, red for false positives, blue for false negatives, and black for true negatives.

DMA-Net employs multi-scale feature fusion and a decoder network. Thus the experimental results suggest that DMA-Net outperforms ECDFFNet in terms of performance in general. Meanwhile, the proposed method and DMA-Net, both of them utilize decoder networks. However, the proposed method uses multi-frequency feature representation to fuse multi-scale features, while DMA-Net stacks multi-level features. Our proposed method outperforms DMA-Net, as indicated by the experimental results. This further confirms the effectiveness of the proposed method. Additionally, in Figure 9, DeepCrack_Zou works well on thin, thick, and extremely thick cracks but badly on extremely thin cracks. DMA-Net is particularly effective on extremely thin, thick, and extremely thick cracks but is generally effective on thin cracks. The same goes for other comparison methods. Contrastingly, optimal performance is achieved by the proposed method on all types of cracks, which adequately demonstrates its robustness.

Figure 10 and Figure 11 display the visualization results. Figure 10 shows the resluts on thin and extremely thin cracks. In Figure 10(a), the input images are selected from the CFD dataset, which have thin cracks, and may contain stains and complex cracks. The proposed method yields optimal prediction results, as evident from the visualization. In Figure 10(b), the input images are selected from the CrackLS315 dataset, which consists of extremely thin cracks, and some cracks are influenced by lane lines. These cracks are so thin that they are even hard to be observed by the naked eyes. But the proposed method can still get the closest crack maps to the ground truth without interference. The visualization results of thick and extremely thick cracks are provided in Figure 11. In Figure 11(a), the input images are selected



(a) Crack200



(b) DeepCrack

Fig. 11. Comparison results of different methods on Crack200 and DeepCrack datasets. The results of the comparison are plotted as green for true positives, red for false positives, blue for false negatives, and black for true negatives.

from the Crack200 dataset, which features thick cracks. The proposed method produces better predictions in these images even when they contain apparent noise. In Figure 11(b), the input images are selected from the DeepCrack dataset, which includes thick cracks and extremely thick cracks. The second, third, and fourth rows of Figure 11(b) show the corresponding ground truth and the prediction maps when the input images are extremely thick cracks. The proposed method can reduce the background interference and, thus, false positives while predicting the cracks more completely to reduce false negatives. The last two rows of Figure 11(a) and the first row of Figure 11(b) show that when both thick and thin cracks are present in the input image, the proposed method accurately predicts the thick cracks while also more completely predicting the thin cracks, which reflects the superior ability of the proposed method to extract the edge details of cracks and thin cracks while extracting the main structure of cracks. In addition, in the last row of Figure 11(b), the input image comes with complex interference information from the environmental background. The proposed method can effectively remove the complex environmental interference, which will be detailed in the later sections.

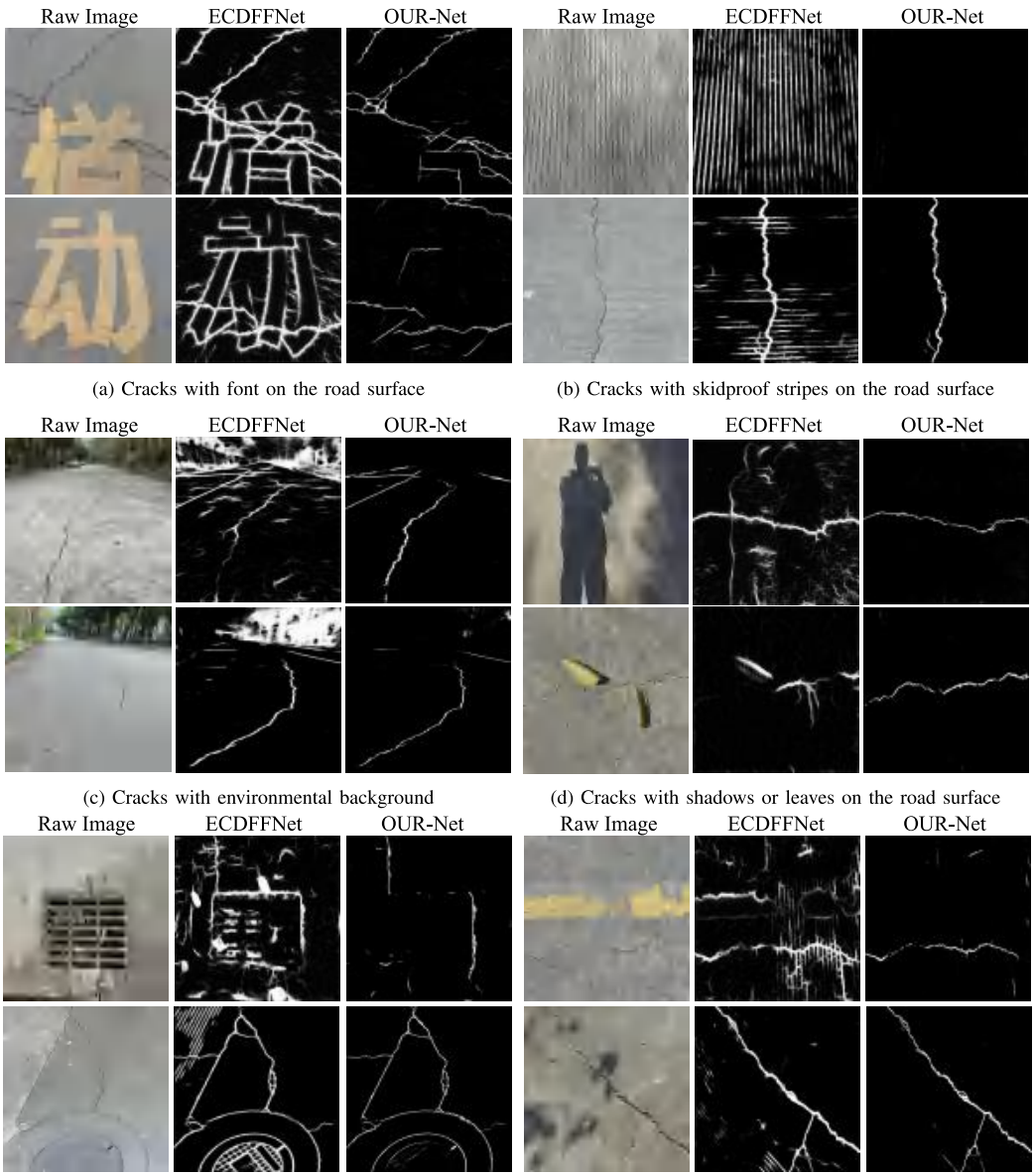
5) Ablation Experiments: In this study, the proposed network becomes SegNet when Octave Convolution (OctConv), Octave Convolution Residual Block (OCRB), and Octave Max Unpooling (OMU) are not utilized. Therefore, we use SegNet as the baseline. Since OCRB and OMU are designed based on OctConv, we used the network utilizing only OctConv as the octave baseline (OctBaseline), which differs from the baseline mainly by replacing the regular convolution with the Octave Convolution. Based on the OctBaseline, we added OMU and OCRB, separately. As shown in Table VI, on the DeepCrack dataset, the baseline model gets the lowest F1-score of 0.9101, and the OctBaseline model gets an F1-score of 0.9168, which

TABLE VI
THE ABLATION EXPERIMENTS ON DEEPCRACK

Method	Pr	Re	F1
Baseline	0.9015	0.9187	0.9101
OctBaseline	0.9230	0.9106	0.9168
OctBaseline+OMU	**0.9336**	0.9222	0.9279
OctBaseline+OCRB	0.9283	0.9290	0.9286
OctBaseline+OMU+OCRB (OUR-Net)	0.9325	**0.9312**	**0.9318**

TABLE VII
COMPARISON OF DIFFERENT LOSS FUNCTIONS ON DEEPCRACK

Method	Pr	Re	F1
OUR-Net (WBCE)	0.9279	0.9240	0.9260
OUR-Net (Dice)	**0.9523**	0.8571	0.9022
OUR-Net (WBCE+Dice)	0.9325	**0.9312**	**0.9318**



(a) Cracks with font on the road surface

(b) Cracks with skidproof stripes on the road surface

(c) Cracks with environmental background

(d) Cracks with shadows or leaves on the road surface

(e) Cracks with at least two types of interference

Fig. 12. Images of pavement cracks captured in a realistic environment using a smartphone and their prediction maps.

TABLE VIII

THE FLOPS, MODEL SIZE AND FPS OF ALL METHODS ON CFD DATASET

Methods	**FLOPs↓**	**Params↓**	**FPS↑**
FCN [CVPR'2015]	64.95G	134.27M	29
HED [ICCV'2015]	188.11G	29.43M	37
U-Net [MICCAI'2015]	128.09G	31.03M	23
DeepCrack-Zou [TIP'2018]	320.76G	30.91M	8
DeepCrack-Liu [Neuro'2019]	47.08G	**14.72M**	**38**
FPHBN [TITS'2019]	147.96G	34.92M	21
ECDFFNet [TITS'2022]	183.53G	58.34M	15
DMA-Net [TITS'2022]	53.40G	60.46M	23
Ours	**44.85G**	24.69M	23

TABLE IX

THE PERFORMANCE COMPARISON OF DIFFERENT α VALUES ON CRACK200 DATASET

α	**Pr**	**Re**	**F1**	**FPS**
0.25	0.7638	0.8225	0.7920	15
0.5	**0.7742**	**0.8506**	**0.8106**	20
0.75	0.7698	0.8368	0.8019	**22**

is 0.67 percent higher than that of the baseline model. This adequately verifies the effectiveness of Octave Convolution in the crack segmentation task. When OMU and OCRB are added separately, the F1-scores are increased by percent of 1.11 and 1.18, respectively, which indicates that the added modules are effective. The model obtained when OMU and OCRB are used simultaneously is the proposed OUR-Net with the highest F1-score of 0.9318. The results in Table VI demonstrate that each module benefits the crack detection task, with the best results achieved when both modules are used simultaneously.

TABLE X
COMPARISON OF LIGHTWEIGHT MODEL RESULTS ON CFD DATASET

Methods	Pr	Re	F1	mIoU	FLOPs↓	Params↓
MobileNetV3 [ICCV'2019]	0.6958	0.9139	0.7901	0.6530	**1.12G**	2.81M
BiSeNetV2 [IJCV'2021]	0.8684	0.9232	0.8950	0.8099	10.38G	5.19M
LinkCrack [TITS'2022]	0.9018	0.9294	0.9154	0.8440	35.65G	3.42M
RHACrackNet [CACAIE'2023]	0.8615	**0.9515**	0.9042	0.8252	5.38G	1.63M
OUR-Net*	**0.9186**	0.9236	**0.9211**	**0.8537**	1.59G	**0.88M**

In addition, in Table VII, OUR-Net (WBCE) and OUR-Net (Dice) denote the loss functions as the weighted binary cross-entropy loss function and the dice loss function, respectively. As seen from Table VII, when only the weighted binary cross-entropy loss function or the dice loss function is utilized, respectively, their F1-scores are 0.9260 and 0.9022, which are lower than 0.9318 when both are used simultaneously. These results indicate that superior results can be obtained by integrating the two loss functions.

6) Running Efficiency: From Table VIII, the proposed model gets the lowest FLOPs and ranks second in terms of Params after DeepCrack_Liu. The FPS of the proposed model also gets the second highest among the crack segmentation task-specific models, with FCN and HED achieving higher FPS because they have fewer network layers and convolution parameters. DeepCrack_Liu achieves the highest Params and FPS. The proposed model gets the best performance with a small model size, thanks to the fact that we do not simply pile up modules or feature maps, but take advantage of multi-frequency feature representation, and carefully design a network that both improves the model performance and reduces the model size.

7) The Performance Comparison of Different α Values: To compare the impact of different α values on the proposed model's performance, we design an experiment in which all the influencing factors are the same except for the hyperparameter α, which determines the ratio of the number of low-frequency feature channels to the total number of channels. The experimental results of the OUR-Net with different α values are shown in Table IX. The table demonstrates that the best F1-score is obtained for OUR-Net with α=0.5. The FPS of OUR-Net with α=0.5 significantly improves compared to OUR-Net with α=0.25. Although the FPS of OUR-Net with α=0.75 is slightly higher than that of OUR-Net with α=0.5, its F1-score is lower by 0.87%.

8) The Lightweight Model: We constructed a lightweight model (OUR-Net*) in which we replaced all convolution operations with depthwise separable convolution, and meanwhile, replaced the number of channels in each layer of the original model from [3, 64, 128, 256, 512, 512] to [3, 32, 64, 128, 256, 256]. From Table X, the learnable parameter of our lightweight model reaches 0.88M, and meanwhile, the F1 score and mIoU still reach 0.9211 and 0.8537, respectively. Compared with the current state-of-the-art lightweight models (MobileNetV3 [59], BiSeNetV2 [60], LinkCrack [61], and RHACrackNet [15]), both the F1-score and the mIoU reach the best with minimal learnable parameters.

9) Performance in Real World Experiments: Figure 12 shows images of actual road surfaces captured by a smartphone with features such as lettering, anti-skid stripes, shadows, leaves, and maintenance hole covers, besides partial pictures of cracks with the environment as a background. We use the models trained from the above four datasets to predict these real images. In particular, the ECDFFNet model is chosen as the compared model. As depicted in Figure 12(a), the proposed method can effectively eliminate the effect of traffic fonts on the road surface. Figure 12(b) demonstrates the effect of segmenting cracks with skidproof stripes. Although the skidproof stripes and the cracks are similar, the proposed method can accurately differentiate between the two. Crack segmentation of the images in Figure 12(c) is extremely challenging due to the presence of environmental backgrounds. The proposed method can eliminate the interference of the background environment well. Figure 12(d) shows features such as shadows or leaves on the road surface, which the proposed model proves to be insensitive to, according to the results. The crack images in Figure 12(e) contain at least two disturbances, which can be observed to have little effect on our method.

V. CONCLUSION

Pavement cracks accelerate damages to the road. Therefore, pavement crack segmentation is of great importance in road maintenance. As a result, an OctConv-based multi-frequency encoder-decoder network is proposed to improve the performance of pavement crack segmentation in this paper. The main contribution of the proposed method is the implementation of multi-scale feature fusion using multi-frequency feature representation instead of stacked feature maps. Therefore, the high- and low-frequency information of the crack images is fully taken advantage of to improve model accuracy further and reduce spatial redundancy. In particular, an Octave Convolution Residual Block (OCRB) is proposed to construct a multi-frequency feature encoder. In addition, we design a novel up-sampling operation called Octave Max Unpooling (OMU) to decode multi-frequency features. The proposed method achieves an impressive balance between segmentation performance and model efficiency, as demonstrated by comprehensive experiments, that confirm its superiority. Finally, a lightweight version of the proposed network is constructed using depthwise separable convolution with only

0.88M parameters. In the future, we will investigate the performance of the model in a larger database including more challenging cases encountered in real world experiments, and attempt to design models with better performance and improved robustness.

REFERENCES

[1] M. Kamaliardakani, L. Sun, and M. K. Ardakani, "Sealed-crack detection algorithm using heuristic thresholding approach," *J. Comput. Civil Eng.*, vol. 30, no. 1, Jan. 2016, Art. no. 04014110.

[2] H. Zhao, G. Qin, and X. Wang, "Improvement of Canny algorithm based on pavement edge detection," in *Proc. 3rd Int. Congr. Image Signal Process.*, vol. 2, Oct. 2010, pp. 964–967.

[3] T. S. Nguyen, S. Begot, F. Duculty, and M. Avila, "Free-form anisotropy: A new method for crack detection on pavement surface images," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 1069–1072.

[4] V. Kaul, A. Yezzi, and Y. Tsai, "Detecting curves with unknown endpoints and arbitrary topology using minimal paths," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1952–1965, Oct. 2012.

[5] A. Ayenu-Prah and N. Attoh-Okine, "Evaluating pavement cracks with bidimensional empirical mode decomposition," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, pp. 1–7, Dec. 2008.

[6] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, "Pavement crack detection using the Gabor filter," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2039–2044.

[7] H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 155–168, Mar. 2013.

[8] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016.

[9] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1558–1570, Aug. 2015.

[10] H. Tsuchiya, S. Fukui, Y. Iwahori, Y. Hayashi, W. Achariyaviriya, and B. Kijsirikul, "A method of data augmentation for classifying road damage considering influence on classification accuracy," *Proc. Comput. Sci.*, vol. 159, pp. 1449–1458, Jan. 2019.

[11] Z. Fan et al., "Ensemble of deep convolutional neural networks for automatic pavement crack detection and measurement," *Coatings*, vol. 10, no. 2, p. 152, Feb. 2020.

[12] Z. Fan et al., "Automatic crack detection on road pavements using encoder–decoder architecture," *Materials*, vol. 13, no. 13, p. 2960, Jul. 2020.

[13] Z. Qu, C. Cao, L. Liu, and D.-Y. Zhou, "A deeply supervised convolutional neural network for pavement crack detection with multiscale feature fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4890–4899, Sep. 2022.

[14] T. Chen et al., "Pavement crack detection and recognition using the architecture of segNet," *J. Ind. Inf. Integr.*, vol. 18, Jun. 2020, Art. no. 100144.

[15] G. Zhu et al., "A lightweight encoder–decoder network for automatic pavement crack detection," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 2023, pp. 1–23, Oct. 2023.

[16] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "DeepCrack: Learning hierarchical convolutional features for crack detection," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1498–1512, Mar. 2019.

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[18] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1525–1535, Apr. 2020.

[19] Q. Zhou, Z. Qu, S.-Y. Wang, and K.-H. Bao, "A method of potentially promising network for crack detection with enhanced convolution and dynamic feature fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18736–18745, Oct. 2022.

[20] X. Sun, Y. Xie, L. Jiang, Y. Cao, and B. Liu, "DMA-Net: DeepLab with multi-scale attention for pavement crack segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 18392–18403, Oct. 2022.

[21] F. W. Campbell and J. G. Robson, "Application of Fourier analysis to the visibility of gratings," *J. Physiol.*, vol. 197, no. 3, pp. 551–566, Aug. 1968.

[22] R. L. De Valois and K. K. De Valois, "Spatial vision," *Annu. Rev. Psychol.*, vol. 31, no. 1, pp. 309–341, 1980.

[23] S. Mallat, *A Wavelet Tour of Signal Processing*. Amsterdam, The Netherlands: Elsevier, 1999.

[24] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511–546, Mar. 1998.

[25] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba, "Automation of pavement surface crack detection using the continuous wavelet transform," in *Proc. Int. Conf. Image Process.*, Oct. 2006, pp. 3037–3040.

[26] S. Chambon, P. Subirats, and J. Dumoulin, "Introduction of a wavelet transform based on 2D matched filter in a Markov random field for fine structure extraction: Application on road crack detection," *Proc. SPIE*, vol. 7251, pp. 87–98, Jul. 2009.

[27] A. Banharnsakun, "Hybrid ABC-ANN for pavement surface distress detection and classification," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 2, pp. 699–710, Apr. 2017.

[28] L. Peng, W. Chao, L. Shuangmiao, and F. Baocai, "Research on crack detection method of airport runway based on twice-threshold segmentation," in *Proc. 5th Int. Conf. Instrum. Meas., Comput., Commun. Control (IMCCC)*, Sep. 2015, pp. 1716–1720.

[29] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 227–238, Feb. 2012.

[30] A. Ouyang and Y. Wang, "Edge detection in pavement crack image with beamlet transform," in *Proc. 2nd Int. Conf. Electron. Mech. Eng. Inf. Technol.*, 2012, pp. 2036–2039.

[31] R. Kapela et al., "Asphalt surfaced pavement cracks detection based on histograms of oriented gradients," in *Proc. 22nd Int. Conf. Mixed Design Integr. Circuits Syst. (MIXDES)*, Jun. 2015, pp. 579–584.

[32] Y. Hu and C.-X. Zhao, "A novel LBP based methods for pavement crack detection," *J. Pattern Recognit. Res.*, vol. 5, no. 1, pp. 140–147, 2010.

[33] Q. Li, Q. Zou, D. Zhang, and Q. Mao, "FoSA: F seed-growing approach for crack-line detection from pavement images," *Image Vis. Comput.*, vol. 29, no. 12, pp. 861–872, Nov. 2011.

[34] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2718–2729, Oct. 2016.

[35] M. Azimi, A. Eslamlou, and G. Pekcan, "Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review," *Sensors*, vol. 20, no. 10, p. 2778, May 2020.

[36] M. Flah, I. Nunez, W. Ben Chaabene, and M. L. Nehdi, "Machine learning algorithms in civil structural health monitoring: A systematic review," *Arch. Comput. Eng.*, vol. 28, pp. 2621–2643, Jan. 2021.

[37] Y.-J. Cha, R. Ali, J. Lewis, and O. Büyükoztürk, "Deep learning-based structural health monitoring," *Autom. Construct.*, vol. 161, p. 105328, May 2024.

[38] N. Kheradmandi and V. Mehranfar, "A critical review and comparative study on image segmentation-based techniques for pavement crack detection," *Construct. Building Mater.*, vol. 321, Aug. 2022, Art. no. 126162.

[39] Y. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 9, pp. 731–747, Sep. 2018.

[40] Y.-J. Cha, W. Choi, and O. Büyükoztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, 2017.

[41] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3708–3712.

[42] E. N. Ukhwah, E. M. Yuniarno, and Y. K. Suprapto, "Asphalt pavement pothole detection using deep learning method based on YOLO neural network," in *Proc. Int. Seminar Intell. Technol. Appl. (ISITIA)*, Aug. 2019, pp. 35–40.

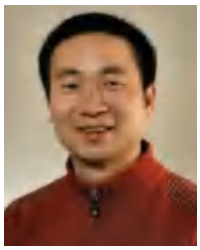
[43] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, "Pavement distress detection and classification based on YOLO network," *Int. J. Pavement Eng.*, vol. 22, no. 13, pp. 1659–1672, Nov. 2021.

[44] W. Choi and Y.-J. Cha, "SDDNet: Real-time crack segmentation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 8016–8025, Sep. 2020.

[45] D. H. Kang and Y.-J. Cha, "Efficient attention-based deep encoder and decoder for automatic crack segmentation," *Struct. Health Monitor.*, vol. 21, no. 5, pp. 2190–2205, Sep. 2022.
[46] R. Ali and Y.-J. Cha, "Attention-based generative adversarial network with internal damage segmentation using thermography," *Autom. Construct.*, vol. 141, Sep. 2022, Art. no. 104412.
[47] J. C. Ong, S. L. Lau, M.-Z. Ismadi, and X. Wang, "Feature pyramid network with self-guided attention refinement module for crack segmentation," *Struct. Health Monitor.*, vol. 22, no. 1, pp. 672–688, Jan. 2023.
[48] H. Liu, J. Yang, X. Miao, C. Mertz, and H. Kong, "CrackFormer network for pavement crack segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 9240–9252, May 2023.
[49] D. Kang, S. S. Benipal, D. L. Gopal, and Y.-J. Cha, "Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning," *Autom. Construct.*, vol. 118, Oct. 2020, Art. no. 103291.
[50] Z. Al-Huda, B. Peng, R. N. A. Algburi, M. A. Al-Antari, R. Al-Jarazi, and D. Zhai, "A hybrid deep learning pavement crack semantic segmentation," *Eng. Appl. Artif. Intell.*, vol. 122, Jun. 2023, Art. no. 106142.
[51] L. Yang, S. Bai, Y. Liu, and H. Yu, "Multi-scale triple-attention network for pixelwise crack segmentation," *Autom. Construct.*, vol. 150, Mar. 2023, Art. no. 104853.
[52] Y. Chen et al., "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3435–3444.
[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[54] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[55] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, Apr. 2019.
[56] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[57] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.
[58] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241.
[59] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.
[60] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiSeNet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3051–3068, Nov. 2021.
[61] J. Liao et al., "Automatic tunnel crack inspection using an efficient mobile imaging module and a lightweight CNN," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15190–15203, Sep. 2022.

Pengtao Li received the B.S. degree from North China University of Water Resources and Electric Power, Zhengzhou, China, in 2020. He is currently pursuing the M.S. degree with South China Agricultural University, Guangzhou, China. His current research interests include computer vision and image processing.

Meihua Wang received the M.S. degree from South China University of Technology, Guangzhou, China, in 1999. She is currently an Associate Professor with the Department of Computer Science and Engineering, South China Agricultural University, Guangzhou. Her research interests include deep learning, computer vision, and intelligent transportation systems.



Zhun Fan (Senior Member, IEEE) received the B.S. and M.S. degrees in control engineering from Huazhong University of Science and Technology, Wuhan, China, in 1995 and 2000, respectively, and the Ph.D. degree in electrical engineering from Michigan State University, East Lansing, USA, in 2004. He is currently a Full Professor with Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China. He is also the Director of the International Joint Research Center of Evolutionary Intelligence and Robotics. His research interests include artificial intelligence, computer vision, and machine learning.

Han Huang (Senior Member, IEEE) received the B.Sc. degree in information management and information systems from the School of Mathematics, South China University of Technology (SCUT), Guangzhou, China, in 2003, and the Ph.D. degree in computer science from SCUT in 2008. He is currently a Full Professor with the School of Software Engineering, SCUT. His research interests include the theoretical foundation and application of evolutionary computation and microcomputation. He is a Distinguished Member of CCF.

Guijie Zhu is currently pursuing the Ph.D. degree in structural engineering with Shantou University. His current research interests include computer vision and structural health monitoring.

Jiafan Zhuang (Member, IEEE) received the B.S. and Ph.D. degrees in control science and engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2017 and 2022, respectively. He is currently a Lecturer with Shantou University, Shantou, China. His current research interests include computer vision and robotic perception.