IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 November 2023, accepted 8 January 2024, date of publication 12 January 2024, date of current version 26 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3353729

RESEARCH ARTICLE

CrackHAM: A Novel Automatic Crack Detection Network Based on U-Net for Asphalt Pavement

MIANQING HE AND TZE LIANG LAU
School of Civil Engineering, Universiti Sains Malaysia, Nibong Tebal, Pulau Pinang 14300, Malaysia
Corresponding author: Tze Liang Lau (celau@usm.my)

ABSTRACT The maintenance of pavements takes considerable time and poses a significant task, especially when it comes to detecting cracks at the pixel level. Due to the complexity of pavement conditions, such as road markings, shadows, and oil stains, deep learning techniques are still a challenge in automating crack detection. This paper presents a novel methodology termed as CrackHAM, which is an encoder-decoder network founded on the U-Net architecture. The primary objectives of CrackHAM are twofold: to achieve accurate and robust pavement crack detection while reducing the parameters of the network. Our study introduces two significant improvements to the existing neural network architecture, namely the phased multi-fusion module and the dual attention mechanisms. These improvements improve the process of defect extraction, resulting in an improved level of performance. Furthermore, a novel module named HASPP is devised to augment the network's capacity to acquire more comprehensive receptive fields. In order to lower the number of network parameters, a technique is employed whereby only use half of the number of input channels and output channels in the VGG16 are utilized as U-Net encoder modules. The empirical findings demonstrate that in the Deepcrack, Crack500, and FIND public datasets, CrackHAM achieves superior segmentation performance compared to the FCN, Deeplabv3, Swin-Unet, and U-Net models while utilizing only one-third of the computational resources.

INDEX TERMS Pavement crack detection, deep learning, fusion module, attention mechanism, atrous convolution.

I. INTRODUCTION

Cracks represent a major type of initial damage on pavement, causing significant degradation in service performance and lifespan [1]. Ensuring well-maintained roads necessitates the diligent survey and maintenance of cracks, which are critical responsibilities for transportation departments. The initial step in this process involves crack detection. Nevertheless, manual crack detection is both time-consuming and subjective in nature. Consequently, there is a rising need for automatic crack detection techniques, which can increase detection efficiency and lower expenses. Missions for pavement crack inspection have been enhanced by using mobile devices in conjunction with remote sensing observation and imaging techniques in recent decades. Mobile mapping systems typically use laser scanners or optical sensors. For the purpose of analyzing pavement distress, images or point clouds can quickly and completely record the conditions and appearances of the pavement surface [2].

Point clouds of pavement can accurately reflect cracks' three-dimensional properties, making them useful for direct crack parameter measurements [3]. However, shallow, narrow, or low-reflectivity cracks are unfriendly. Pavement images are better for crack detection because they have a multitude of textural intricacies and notable disparities in intensity between cracks and other focal points [4], [5]. There has been a significant focus on computer vision, with numerous researchers devoting considerable attention to this field. Numerous techniques have been devised, including image processing techniques (IPT) and machine-learning techniques. A wavelet transform was used by [6] to automatically identify and emphasize cracks. Reference [7] utilized entropy and image dynamic thresholding to automatically

The associate editor coordinating the review of this manuscript and approving it for publication was Essam A. Rashed.

segment pavement cracks into crack and non-crack pixels. Despite this, these algorithms often encounter difficulties operating effectively due to factors such as inhomogeneous distress, shadows, and other related issues.

On the other hand, machine-learning methods use manual features to identify areas in the image that contain cracks. Several crack detection algorithms, [8], [9], and [10] have been proposed that rely on feature engineering. However, conventional machine-learning methods usually depend on shallow learning techniques and heavily rely on handcrafted features extracted from input images, which makes it challenging to produce semantic segmentation outcomes.

Deep convolutional neural networks (CNNs), popularized by AlexNet's [11] win in the 2012 ImageNet championship, are widely used in computer vision for pavement crack detection. Deep learning approaches possess the capability to automatically learn features from data and optimize model parameters. This feature's learning ability streamlines the model design and adjustment process. Therefore, there are algorithms applied for pavement crack detection. For pavement crack detection, [12] proposed a novel approach for detecting different sorts of structural defects by utilizing a Faster R-CNN-based algorithm [13]. Reference [14] utilized a sliding window approach to analyze photos with greater dimensions in order to detect concrete crack. For pavement crack segmentation, [15] utilized U-Net [16]. Using CNN DeepLabv3+, [17] presented a method for integrated crack detection. Deep full convolutional networks (FCN) were proposed by [18] as a crack detection method and successfully used to perform semantic segmentation of pavement crack images. In [19], an encoder network architecture was designed that shares the same topology as the 13 convolutional layers found in the VGG16 network [20]. By deep learning with improved statistical segmentation, it becomes possible to accomplish automated crack detection and segmentation more swiftly, leading to improved detection efficiency and decreased maintenance expenses. However, many of these algorithms are frequently designed to extract more intricate characteristics by constructing more complex network architectures. Consequently, these models can be relatively large and require a substantial amount of hardware resources for computing power.

To reduce computer resource consumption, a target segmentation architecture based on U-Net is employed to segment pavement cracks. The primary findings of our study can be succinctly outlined as follows:

1) The U-Net architecture has been enhanced by our team to develop a novel crack detection network. Despite being over 50% smaller than U-Net, our network exhibits superior detection performance.
2) In order to tackle the matter of potential loss of image details resulting from the utilization of the Atrous Spatial Pyramid Pooling (ASPP) [21] module, the HASPP module is proposed. The module is designed to proficiently capture both multi-scale features and fine-grained details.
3) A phased multi-fusion module has been implemented to allow for the integration of high-level features and facilitate the merging of low-level features, enhancing the overall feature representation. Distinct attention mechanisms are utilized for the two types of features. A channel attention module is implemented to provide more discriminative information for high-level features, whereas a spatial attention module is utilized to concentrate on pertinent details and eliminate extraneous information for low-level features.
4) The method exhibits superior performance compared to four other CNN models while necessitating fewer computational resources for training.

II. RELATED WORKS

This section begins with an overview of traditional approaches used for crack detection. Subsequently, it introduces a more recent approach based on deep learning.

A. TRADITIONAL METHODS

Before the advent of deep learning, conventional image processing-based methods and machine learning were used for crack detection. The rapid progress in image processing and machine learning has significantly enhanced the efficacy and precision of crack detection. For instance, a heuristic threshold method was proposed by [22] to detect cracks where crack pixels are usually darker than background pixels under normal illuminance. However, such threshold-based methods are not effective in realistic environments due to uneven illumination and the difficulty of determining a suitable threshold. To overcome this limitation, [23] proposed a percolation-based model that builds connectivity between neighboring pixels' gray values to ensure crack continuity. Reference [24] utilized two post-processing steps to improve the precision and reliability of crack detection results. Moreover, in [25], random structure forest [26] was employed to leverage structured information in crack patches.

B. DEEP LEARNING METHODS

Deep learning has seen significant breakthroughs in computer vision tasks, including crack detection, recently. Some researchers have used CNNs to perform classification of crack images [27], while others [17], [18] have employed semantic segmentation algorithms to classify each pixel of an image, assigning it a label of either crack or non-crack. Other authors have combined these approaches, such as [28], who implemented a two-phase sequence processing method for attaining crack detection at the pixel level on pavement crack. The CNNs generate multi-scale features by utilizing pooling operations, and information from these multi-scale features [29] and [30] is blended to enhance the network's crack feature representation. Numerous studies [31], [32], [33] have developed innovative network architectures that

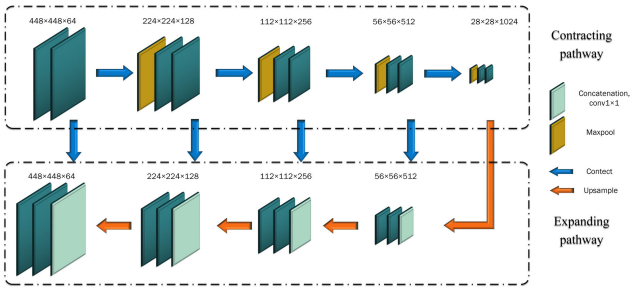

FIGURE 1. U-Net architecture.

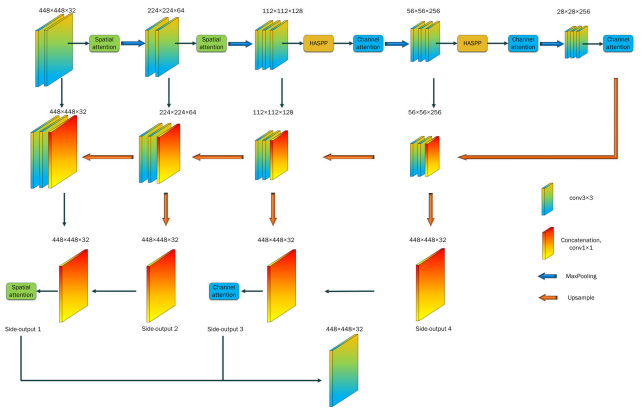

FIGURE 2. Illustration of the proposed CrackHAM architecture.

effectively capture both spatial and channel-based semantic interdependencies by leveraging attention mechanisms. The attention-based Generative Adversarial Network (GAN) [34] developed by [35] is proposed as a solution for the real-time detection of internal defects in concrete members. Reference [36] utilizes a squeeze-and-excitation attention-based encoder and a multi-head attention-based decoder to attain enhanced crack segmentation outcomes. By incorporating these attention mechanisms, these studies have successfully enhanced contextual information, resulting in improved performance and a richer understanding of the data. To detect scattered, tiny cracks, the ASPP module is adapted to improve the receptive field [37]. Reference [38] integrates the ASPP module with separable convolution modules to construct a computationally efficient network suitable for the purpose of segmenting concrete cracks.

III. METHODOLOGY

A. OVERVIEW OF THE PROPOSED METHOD

The U-Net network has been used for numerous segmentation tasks, which have been demonstrated effectively. According to Fig. 1, the structure is made up of two primary constituents: a contracting pathway that acquires information about the context of the image inputted and an expanding pathway that produces the segmentation map. The U-Net framework incorporates skip connections that establish connections between the corresponding layers in the contracting and expanding pathways. This enables the network to exploit the high-resolution features of the contracting pathway while producing the segmentation map of the expanding pathway.

The architectural design, named CrackHAM, is founded on the U-Net framework. The VGG16 network was selected, and the fully connected layers are omitted in the contracting path. Additionally, the VGG16 network underwent a 50% reduction in the number of input channels and output channels in each of its layers.

CrackHAM, illustrated in Fig. 2, leverages a hierarchical feature extraction approach that combines features from multiple layers. The third and fourth layers of the contracting path apply the HASPP module to introduce additional detailed information to the network, which is described below in detail. In light of the characteristics of different level features, we apply different attentions (spatial attention and channel attention) to select effective features. Besides, our expanding path utilizes side-output features differently. After the initial side-output layer, the subsequent three side-output layers are accompanied by deconvolutional layers to ensure the feature maps align with the original input image size and a 1×1 convolutional layer, which adjusts the number of channels. Side-output 1 and Side-output 2 are combined together and then processed through a spatial attention mechanism. Similarly, Side-output 3 and Side-output 4 were combined together, but with the use of a channel attention mechanism. Then these feature maps are combined through concatenation to generate the final features.

B. HASPP MODULE

The size of the receptive field in deep neural networks can indicate the extent to which context information is utilized [39]. However, in semantic segmentation tasks, simply expanding the receptive fields by combining convolutional and pooling layers can result in decreased resolution and computational inefficiency. Furthermore, research by [40] has shown that the receptive field of CNNs is often found to be significantly smaller than the theoretical receptive field, particularly in higher layers. This limitation greatly hinders CNNs' ability to accurately predict using contextual information. To address this, by utilizing dilated convolutions with varying dilation rates, it becomes possible to extract contextual information at different scales. This technique allows for a significantly larger receptive field, enabling the capture of extensive details while also preserving the resolution of the input data. The ASPP module, which employs dilated convolution, was first proposed in DeepLab [41].

However, the ASPP module causes a loss of spatial information. At present, the most popular dilated rates are (6, 12, and 18) in the ASPP module. It means that the convolutional kernels have many ''holes''. The receptive field of a convolutional kernel is restricted to regions with square patterns due to the filling of ''holes'' with zeros. This implies that only those positions with non-zero values are taken into account, resulting in the loss of some neighboring

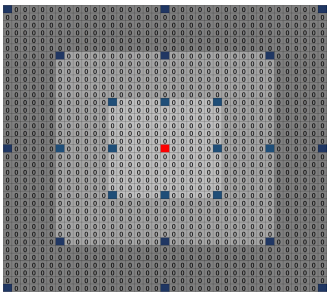

FIGURE 3. Illustration of the ASPP problem.

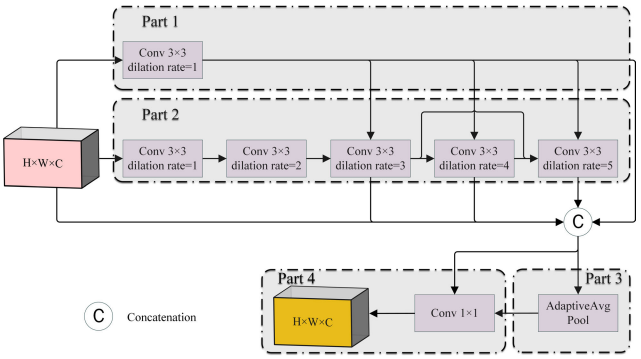

FIGURE 4. Illustration of HASPP module.

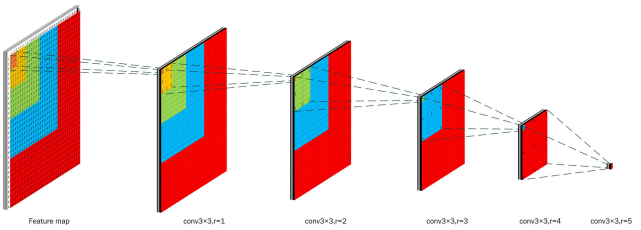

FIGURE 5. Illustrates part two which shows the cascading of several dilated convolutional layers.

information. This issue becomes more pronounced as the dilation rate increases, particularly in higher layers where the receptive field is larger. The center pixel (marked in red) is computed through three convolutional layers with a kernel size of 3 × 3, where the pixels contributing to this computation are identified by the blue markings. Convolutional layers have dilation rates of r = 6, 12, and 18, respectively, in Fig. 3. Take r = 6 as an example, where only 9 out of 169 pixels in the region are utilized for computation (the lightest gray area in Fig. 3). As r increases to larger values, such as r = 12 or r = 18, the input information can become extremely sparse, which is not ideal for effective learning as the local information is entirely missing.

In this study, a novel network model, called the HASPP module, draws inspiration from the work of [42] and [43]. The objective of this model is to tackle the problem of spatial information degradation. The HASPP's operation, divided into four parts, is presented in a schematic diagram in Fig. 4. In part one, a convolutional layer is executed on a feature map utilizing a kernel size of 1. In part two, the feature map undergoes a cascading process that involves five layers of CNNs with increasing rates of dilation, specifically 1, 2, 3, 4, and 5 times the original dilation rate, but the outputs of the layers of CNNs with 2, 3, and 4 times the original dilation rate will be concatenated with the output of part one, respectively. Additionally, a skip connection will be applied between the layer of CNNs 3 times the original dilation rate and the layer of CNNs with 5 times the original dilation rate. In part three, the output of part one is concatenated with the feature maps obtained by layering CNNs with 3, 4, and 5 times the rate of dilatation in part two, as well as the input feature map. Then AdaptiveAvgPool is applied on the resulting feature maps. In part four, the outputs from the first, second, and third parts are concatenated with the input feature map. A 1 × 1 convolution is then applied to adjust the number of channels to match the channel number of the input feature map.

In cascading mode, an upper dilated layer accepts the output of a lower dilated layer, enabling the efficient production of large receptive fields, as shown in Fig. 5. Each color in the figure represents a different dilated convolutional dilation rate of 1, 2, 3, 4, and 5, respectively. The range of the receptive field for each dilated inference is showcased on the feature map using different colors to represent it. Each layer applies padding equal to its corresponding dilation rate, and the dilated layers obtain all the information from the input feature map.

C. PHASED MULTI-FUSION MODULE

To address the issue of losing important information and the absence of linkage between global and local information in fully convolutional neural networks with pooling layers, [44] proposed the Multi-Fusion U-Net architecture. This approach aggregates contextual information from feature maps of varying sizes during the downsampling phase, allowing for the extraction of detail-space features from global images. However, the architecture of Multi-Fusion U-Net overlooks the distinct characteristics of features at different levels. Edge information is plentiful in low-level features, but they also contain considerable noise. Similarly, high-level features capture extensive contextual information, yet they may produce less detailed outcomes.

Therefore, based on the idea of a multi-fusion module, the proposed improved module, called a phased multi-fusion module, extracts features at multiple scales from different layers of the network, facilitating more accurate predictions. Fig. 6 illustrates how the proposed phased multi-fusion module applies upsampling in a top-to-bottom approach with different factors of 8, 4, 2, and 1. Next, a 1 × 1 convolution is used to directly adjust the feature channels to a size of 32. Different from the architecture of Multi-Fusion U-Net, the outputs of the high-level features (the first and second layers) are concatenated, and then the number of channels can be adjusted using a 1 × 1 convolutional kernel. Then the

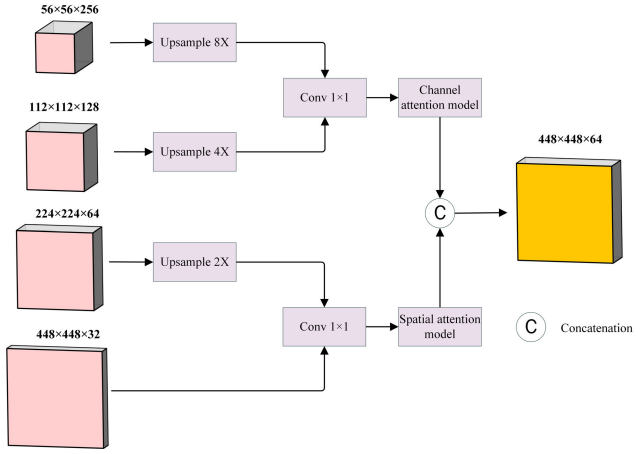

FIGURE 6. The proposed phased multi-fusion module.

channel attention mechanism can be applied. Similarly, the outputs of the low-level features (the third and fourth layers) can be concatenated, and then the number of channels can be adjusted using a 1×1 convolutional kernel. Finally, the spatial attention mechanism can be applied. Lastly, the two output results can be concatenated and then outputted.

D. ATTENTION MECHANISMS

In the context of crack detection, the low-level feature encompasses a significant amount of detailed information, but accompanied by a considerable amount of noise. Conversely, the high-level feature encompasses a wealth of abstract semantic information, which might result in indistinct outcomes. Inspired by the CBAM attention module [45], taking into consideration the distinctive characteristics of different-level features, spatial attention is utilized in convolution stages 1-2 to concentrate on low-level features, while channel attention is employed in convolution stages 3-5 to accentuate high-level features.

1) SPATIAL ATTENTION MODULE

The structure of the spatial attention module is depicted in Fig. 7. Firstly, the input features, which have a size of H × W × C, undergo separate max pooling and average pooling operations, generating two feature maps of size H × W × 1. Subsequently, these feature maps are concatenated by means of channel splicing, thereby creating a unified feature map of size H × W × 2. After a 7 × 7 convolution operation, this unified feature map is compressed into a single feature map of size H × W × 1. Finally, the feature map obtained is fed into a sigmoid activation function, producing spatial attention features.

$$S(F) = \sigma C^{7\times7}[(F_{avg} + F_{max})] \tag{1}$$

where σ means the sigmoid function, $C^{7\times7}$ denotes the convolution operation with a filter size of 7×7, F_{avg} refers the feature map after average pooling, and F_{max} refers the feature map after max pooling. S(F) denotes spatial attention weight.

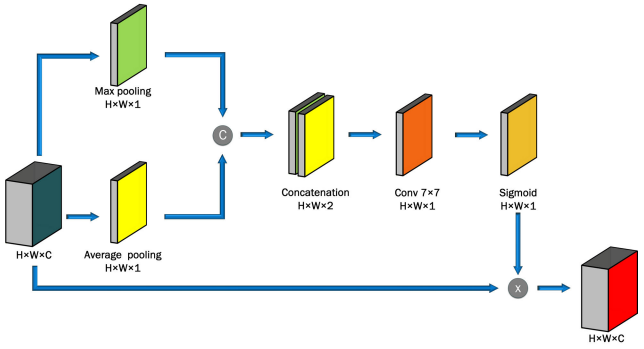

FIGURE 7. The spatial attention.

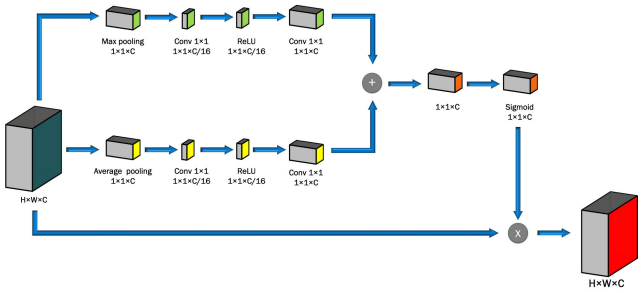

FIGURE 8. The channel attention.

2) CHANNEL ATTENTION MODULE

Fig. 8 illustrates the structure of the channel attention module. The input features of size $H \times W \times C$ for spatial attention are subjected to global max pooling and global average pooling operations, resulting in two feature maps. These feature maps are then compressed using a 1×1 convolution operation to a size of $1 \times 1 \times C/R$, where R is the compression rate set to 16. The Rectified Linear Unit (ReLU) activation function is then applied to each compressed feature map. Following this, the two feature maps are again compressed to a size of $1 \times 1 \times C$ individually through another 1×1 convolution operation. The resulting output features from the ReLU activation function are added together and passed through the sigmoid activation function to generate the channel attention features.

$$C(F) = \sigma\left\{C_2(R(C_1(F_{avg}))) + C_2(R(C_1(F_{max})))\right\} \tag{2}$$

where σ means the sigmoid function, C_1 denotes the convolution operation with a filter size of 1×1 and input channels to C, which outputs channels to C/R channels. Similarly, C_2 signifies convolution with a filter size of 1×1, input channels to C/R, which outputs to C channels. F_{avg} refers the feature map after applying global average pooling and F_{max} refers to the feature map after applying global max pooling. Finally, C(F) denotes channel attention weight.

IV. EXPERIMENT

A. DATASETS

For three public pavement datasets: the Crack 500 [46], DeepCrack [30], and FIND [47]. There are some samples for

TABLE 1. The detail of public pavement crack datasets.

Dataset	Train	Valid	Test	Size	Using
Crak500	2420	606	337	448 × 448	Comparison
DeepCrack	387	97	53	448 × 448	Comparison
FIND	1600	400	500	256 × 256	Ablation and comparison

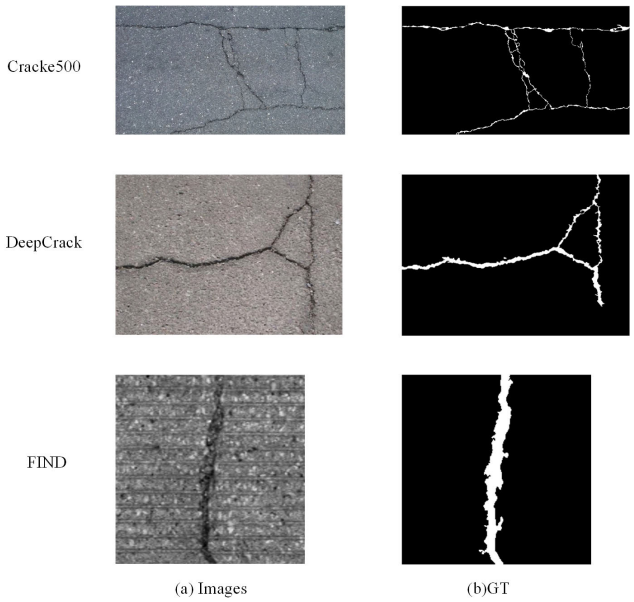

FIGURE 9. Dataset sample.

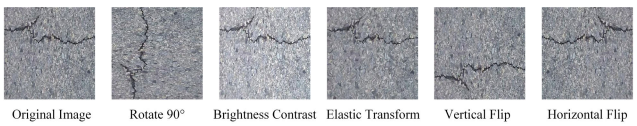

FIGURE 10. Techniques sample.

datasets, shown in Fig. 9. We randomly select 80%–90% of the samples from each dataset as the training set and leave the remaining samples as the testing set. During the training process, a validation set is selected by randomly choosing 20% of the training dataset. The detailed information can be found in TABLE 1. Due to the relatively small scales of the three public datasets, they might not provide enough training data to achieve effective results. To address this challenge, various techniques are employed to augment the dataset and expand the number of image patches. These techniques include rotation (anticlockwise 90°), random adjustments of brightness and contrast, elastic deformation, and mirroring (left-right and top-bottom), shown in Fig. 10.

B. TRAINING SETTINGS

This experiment used PyTorch deep learning software and Python 3.6.0. The experimental environment has an Intel(R) i5-12600KF CPU, 12GB memory, an NVIDIA GeForce RTX3080 Ti GPU, and Windows 10. Due to computer memory constraints and the need for fair model testing, the

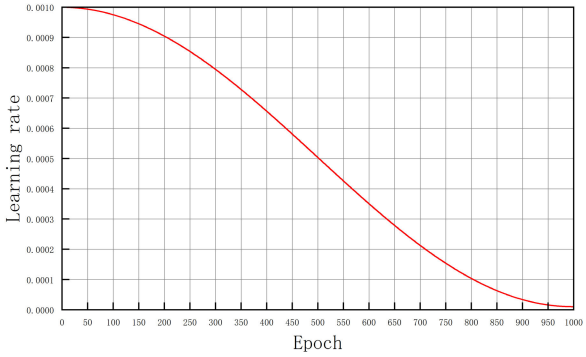

FIGURE 11. Learning rate.

epoch was set to 1000, batch size was 8 for 448 × 448, 24 for 256 × 256, and shuffle was True. Adam [48] was used as the optimizer. Training begins with a learning rate of 0.001 and is reduced by simulated annealing with the Cosine Annealing Schedule [49] to 0.00001, as shown in Fig. 11. The loss function utilized in this study was BCE-Dice loss [50]. BCE-Dice loss is a method that combines binary cross-entropy loss and Dice coefficient. Pretraining is not considered in our network to ensure fairness in comparison with other networks. The paper's training indicator is the validation set's Intersection over Union (IoU) result. The model weights are saved when the validation set yields the best IoU value.

C. EVALUATION CRITERIA

Multiple metrics are employed in our semantic segmentation evaluation to evaluate its performance. These metrics include Precision(P), Recall(R), F1-measure (F1), IoU, and Average Precision (AP). The ratio of positive pixels predicted as crack types is precision. Recall is the percentage of crack pixels in the image that are correctly predicted as crack. The precision-recall curve's area is calculated by AP. The F1-measure is the arithmetic mean of Precision and Recall. The predicted crack region's overlap with the ground truth crack region is measured by IoU. Definitions of Precision, Recall, F1, IoU and AP:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{5}$$

$$IoU = \frac{TP}{TP + FP + FN} \tag{6}$$

$$AP = \int_{R=0}^{1} P(R)\, dR \tag{7}$$

where TP (True Positive) refers to the count of crack pixels that are correctly predicted as cracks. FP (False Positive) represents the number of pavement pixels that are incorrectly

classified as cracks. FN (False Negative) denotes the count of crack pixels that are mistakenly predicted as non-crack pixels.

In addition, parameters and floating point of operations (FLOPs) are key indicators of model complexity. Calculating convolutional layer parameters:

$$Parameter = K^2 \times C_{in} \times C_{out} \tag{8}$$

where K is the size of convolution kernel. And C_{in}, C_{out} represent the number of channels for the input and output feature maps, respectively.

D. ABLATION EXPERIMENT

Table 2 is utilized to compare various module combinations to evaluate the performance of the proposed network. Four model structures, named CrackHAM _B, CrackHAM _F, CrackHAM _F _A, and CrackHAM, are created by incorporating the phased multi-fusion module, attention mechanisms, and HASPP into the backbone network. To assess the performance differences between the HASPP and HAPP modules, a network called CrackHAM _ASPP is constructed. The hyperparameter settings for network training are in accordance with the guidelines outlined in Section IV-B. The suitability of the FIND dataset for ablation experiment is attributed to its large quantity and small size.

From Fig. 12, it is evident that each of the proposed components contributed to the enhancement of the effectiveness of the detection during the training and validation processes. The absence of overfitting and the achievement of convergence can be inferred from the performance of models in both the training and validation processes. Furthermore, the comparison between CrackHAM _F _A and CrackHAM revealed significant improvements in IoU, demonstrating that extending the receptive field by adding HASPP can significantly improve detection performance.

As indicated in Table 3, the CrackHAM achieves the highest IoU, Precision, Recall, and F1 values on the test images, which are 0.765, 0.864, 0.850, and 0.854, respectively. When comparing CrackHAM_B to CrackHAM_F, there is an improvement of 0.7% in precision, 0.8% in recall, 0.9% in F1, and 0.9% in IoU score. Similarly, when comparing CrackHAM_F_A to CrackHAM_F, there is an improvement of 0.5% in precision, 0.5% in recall, 0.5% in F1, and 0.7% in IoU score. Furthermore, comparing CrackHAM to CrackHAM_F_A, there is an improvement of 0.1% in precision, 0.6% in recall, 0.4% in F1, and 0.8% in IoU score. These findings highlight the effectiveness of phrase fusion modules, attention mechanisms, and the HASPP module in enhancing segmentation performance. After replacing the HASPP structure with the ASPP structure, the network's performance degraded. As a result, the decision is made to opt for the HASS module. To provide a clearer depiction, please consult Fig. 13.

E. COMPARISON OF DIFFERENT NETWORKS

The CrackHAM model has demonstrated superior performance in crack segmentation when compared to recently developed networks that have attained cutting-edge outcomes in the task of semantic segmentation, including FCN [51], Deeplabv3 [21], Swin-Unet [52], U-Net, CrackSeg [53], ATT- Unet [54] and U-net++ [55]. Details of their parameters and FLOPs are provided in Table 4. It is worth noting that both FCN and Deeplabv3 incorporate ResNet50 [56] as the backbone architecture for their networks and Swin-Unet utilizes Swin transformer [57] as the backbone. The FLOPs of the CrackSeg, ATT-Unet and U-net++ exhibit a notable

TABLE 2. Network alternatives considered for training.

Methods	Phrase multi-fusion module	Attention mechanisms	HASPP	ASPP
CrackHAM_B	-	-	-	-
CrackHAM_F	✓	-	-	-
CrackHAM_F_A	✓	✓	-	-
CrackHAM	✓	✓	✓	-
CrackHAM_ASPP	✓	✓	-	✓

Notes: "✓" means the module is introduced and "-" means the structure is not introduced.

TABLE 3. Results of ablation experiment for testing FIND dataset.

Methods	Precision	Recall	F1	IoU
CrackHAM_B	0.851	0.831	0.836	0.741
CrackHAM_F	0.858	0.839	0.845	0.750
CrackHAM_F_A	0.863	0.844	0.850	0.757
CrackHAM	0.864	0.850	0.854	0.765
CrackHAM_ASPP	0.857	0.843	0.845	0.752

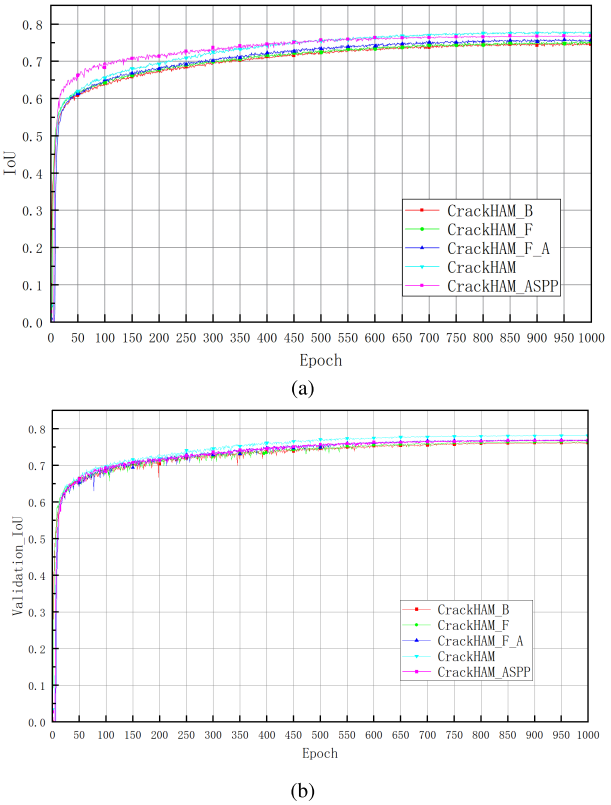

FIGURE 12. The performance of different modules.

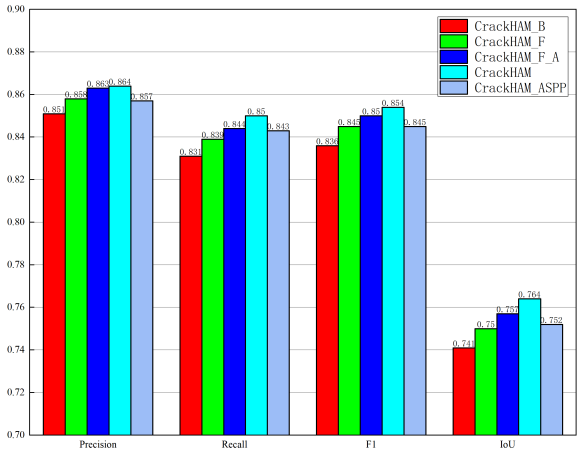

FIGURE 13. IoU, precision, recall, and F1 results.

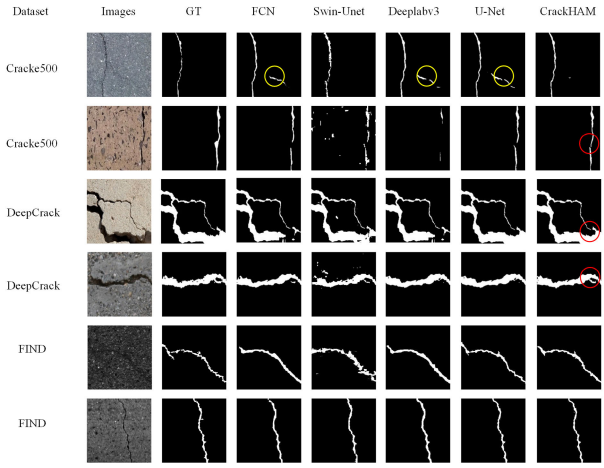

FIGURE 14. Visualization results of different methods of the FCN, Swin-Unet, Deeplabv3, U-net, CrackHAM on different datasets.

superiority over the remaining models. As a result, in the succeeding trials, the channels are halved in size throughout the training process. To ensure equitable comparison, the networks underwent training and testing using identical sets of training, validation, and test data. Additionally, the same data augmentation technique was employed throughout the process. The training phase involved applying Loss to track the progress of parameter learning, while the model's performance during validation was measured using IoU.

Fig. 14 shows visual comparisons of crack identification for the proposed approach and the comparative methods using six common input photos. Crack500, DeepCrack, and FIND provide input images. The first two rows represent images from Crack500, while the middle two rows showcase images from DeepCrack. Finally, the last two rows exhibit input images from FIND. All deep models are capable of generating satisfactory results. In cases where the presence of pavement traces influences the results, the models FCN, DeeplabV3, and U-Net may exhibit error detection, while

TABLE 4. The complexity of different networks.

	FLOPs (G)	Parameters (M)
FCN	110.65	32.95
Deeplabv3	125.65	39.63
Swin-Unet	23.64	27.15
U-net	171.45	31.39
CrackHAM	81.58	13.90
CrackSeg	340.10	10.80
ATT-Unet	203.62	34.88
U-net++	424.65	36.63

Notes: The FLOPs arecalculated with 448×448 input

CrackHAM and Swin-Unet make the correct decision in the first row. The models FCN, DeeplabV3, Swin-Unet and U-Net tend to produce under-segmented crack regions, meaning that they may not accurately identify the boundaries of the cracks. On the other hand, the CrackHAM model demonstrates better precision in determining the crack paths, as evident in the second and third rows of the results. In particular, in the fourth row, the CrackHAM model correctly detects small localized features. This can be attributed to the inclusion of the HASPP module, which makes the network more sensitive to detailed information so that it captures fine-grained features and improves its overall performance in detecting cracks, even in challenging cases. In the last two rows of the table, although all five models successfully detect the cracks, the segmentation results of FCN and Deeplabv3 are wider than the ground truth. This may be due to the fact that FCN and Deeplabv3 lack skip connection layers and phased multi-fusion modules, which help capture multi-scale and multi-level information and enhance the precision and resilience of the model.

1) THE RESULTS OF DEEPCRACK

Fig. 15 illustrates the loss curve during training and the IoU curve during validation over the 1000 epochs. It can be found that the CrackHAM, U-Net, Swin-Unet, Deeplabv3, and FCN perform similarly in terms of Loss and IoU, respectively. At Fig. 15(b), the IoU values fluctuate greatly on the validation set, but they finally reach convergence, which is maybe caused by a too high learning rate. Because in the early stage, the learning rate is relatively large, close to 0.001, and after it reaches 700 epochs, the learning rate is smaller than 0.0002, as shown in Fig. 11. We don't modify hyperparameters since, as the training proceeds, the model gradually approaches the optimal solution, the parameter updates become more stable, and the loss and IoU values gradually stabilize and eventually converge to a fixed value. Another reason is keeping the hyperparameters the same to evaluate different models accurately.

CrackHAM exhibits a curve that is closest to the upper-right corner, indicating its superior performance. It achieves the highest precision and recall values among the compared methods, as shown in Fig. 16. The performances of CrackHAM, U-Net, and CrackSeg are very close, which is about

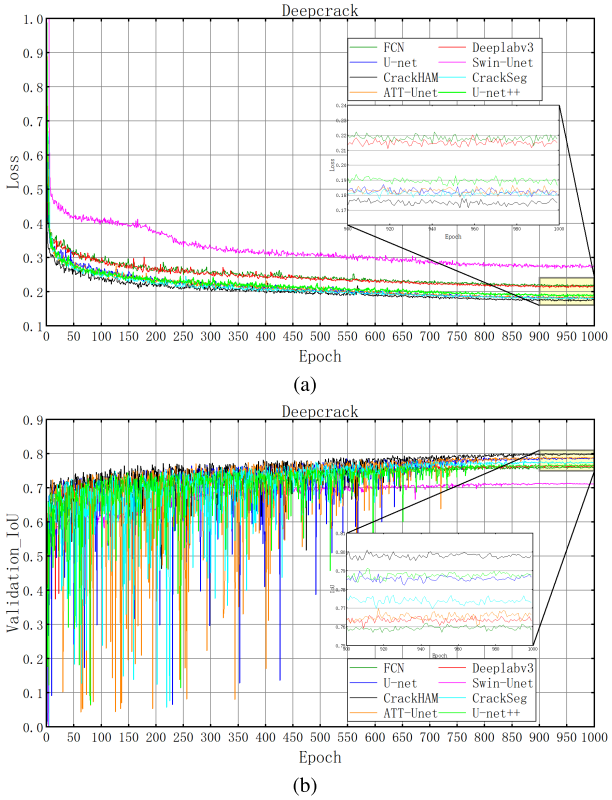

FIGURE 15. Loss (a) curves during training and IoU (b) curves during validation.

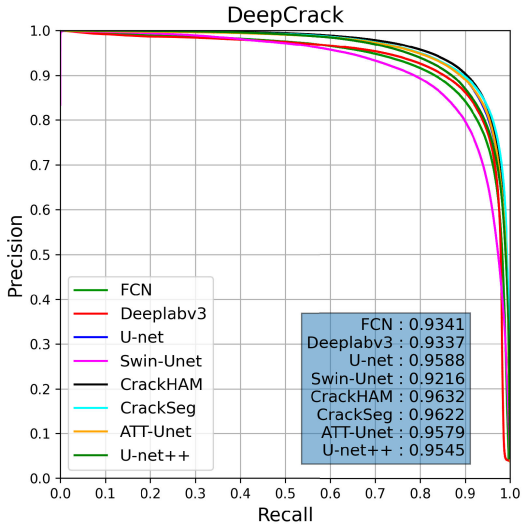

FIGURE 16. Precision-Recall curves on the DeepCrack dataset.

0.96. Similarly, Swin-Unet, Deeplabv3, and FCN exhibit closely aligned performances, while ATT-Unet and U-net++ also demonstrate similar performance levels.

The quantitative results in Table 5 show that our method gets the best results on the Precision, Recall, F1, IoU, and AP evaluation criteria. The most important point is that our model has the smallest number of parameters, only 1/3 of other models, which is 13.90M. Obviously, the CrackHAM significantly reduces the computational cost, and splitting performance has also been improved.

TABLE 5. Test results of compared methods for DeepCrack.

	Precision	Recall	F1	IoU	AP
FCN	82.01%	86.32%	83.30%	0.7260	0.9341
Deeplabv3	81.64%	87.09%	83.19%	0.7288	0.9337
Swin-Unet	82.52%	82.07%	81.01%	0.6955	0.9216
U-net	85.65%	87.10%	85.84%	0.7603	0.9588
CrackHAM	85.98%	88.23%	86.59%	0.7724	0.9632
CrackSeg	85.76%	87.70%	86.38%	0.7671	0.9622
ATT-Unet	76.13%	87.01%	85.78%	0.7613	0.9579
U-net++	84.32%	86.99%	84.82%	0.7484	0.9545

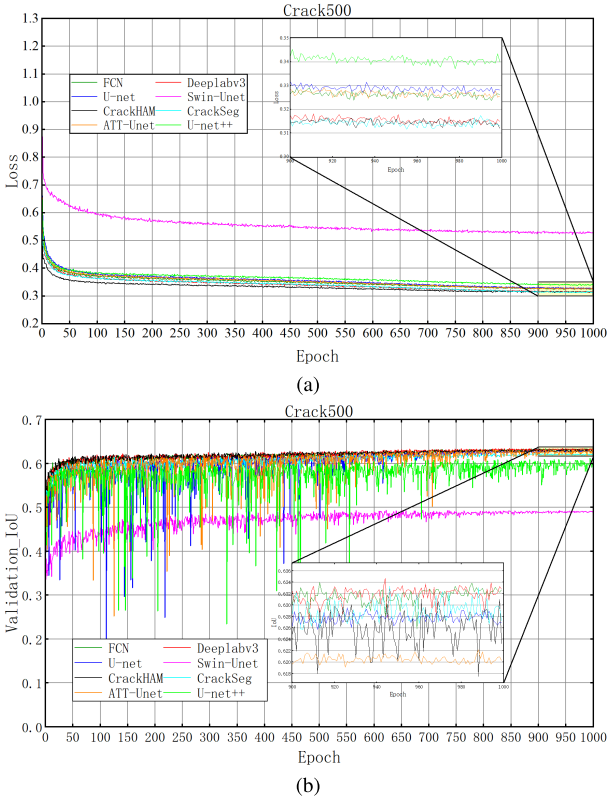

FIGURE 17. Loss (a) curves during training and IoU (b) curves during validation.

2) THE RESULTS OF CRACK500

Fig. 17 depicts the loss curve observed during the training process and the IoU curve representing the model's performance during validation. Analysis reveals that CrackHAM, U-Net, Deeplabv3, FCN, CrackSeg, ATT-Unet and Unet++ exhibit comparable performance in terms of IoU. However, when it comes to Swin-Unet, its performance is notably inferior. From Fig. 17(a), each model has reached convergence after 900 epochs. The IoU finally reaches convergence although its values fluctuate greatly on the validation set (Fig. 17(b)).

Fig. 18 shows precision-recall curves for five different models. The curves of FCN, Deeplabv3, U-Net, CrackHAM, CrackSeg, ATT-Unet, and Unet++ overlap for the most part, indicating that the models have similar performance on the dataset being evaluated.

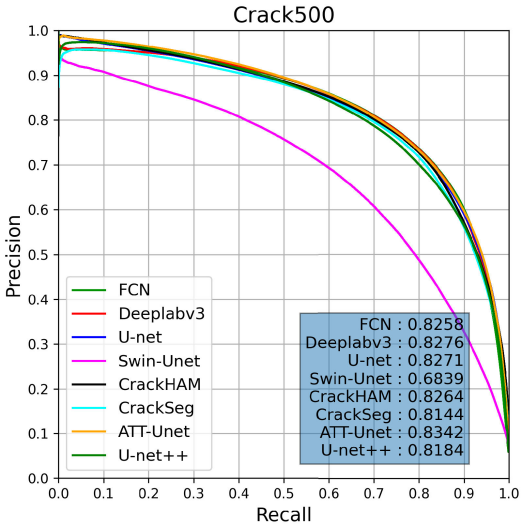

FIGURE 18. Precision-Recall curves on the Crack500 dateset.

TABLE 6. Test results of compared methods for Crack500.

	Precision	Recall	F1	IoU	AP
FCN	71.63%	77.68%	72.32%	0.5916	0.8258
Deeplabv3	72.16%	78.01%	72.49%	0.5938	0.8276
Swin-Unet	61.50%	65.97%	59.40%	0.4489	0.6839
U-net	71.26%	78.26%	72.40%	0.5902	0.8271
CrackHAM	73.40%	77.31%	73.13%	0.5973	0.8264
CrackSeg	71.26%	78.68 %	72.50 %	0.5901	0.8144
ATT-Unet	71.54%	78.43%	72.52%	0.5921	0.8342
U-net++	70.27%	76.59%	70.85%	0.5755	0.8184

TABLE 7. Test results of compared methods for FIND.

	Precision	Recall	F1	IoU	AP
FCN	74.02%	78.29%	75.45%	0.6328	0.8511
Deeplabv3	69.18%	84.83%	75.48%	0.6283	0.8566
Swin-Unet	75.17%	72.71%	73.09%	0.6154	0.8404
U-net	88.03%	84.80%	85.98%	0.7724	0.9461
CrackHAM	86.41%	84.97%	85.38%	0.7644	0.9440
CrackSeg	67.34%	72.41%	68.18%	0.5568	0.7605
ATT-Unet	84.75%	83.63%	83.70%	0.7395	0.9301
U-net++	83.66%	83.51%	83.12%	0.7299	0.9264

Table 6 presents the quantitative results of five models evaluated on the test set. The performance of the CrackHAM model is very impressive, as it achieves the highest scores for evaluation metrics such as IoU, Precision, and F1 among all five models.

3) THE RESULTS OF FIND

Fig. 19 displays the loss and IoU curves during training and validation across 1000 epochs. The results show that U-Net and CrackHAM exhibit comparable performance, while FCN, Swin-Unet, and Deeplabv3 exhibit similar results in terms of IoU and loss.

Fig. 20 shows the precision-recall curves, where U-Net achieves the highest performance on FIND with a score of 0.9461, outperforming all compared methods. CrackHAM ranks second with an AP value of 0.9440.

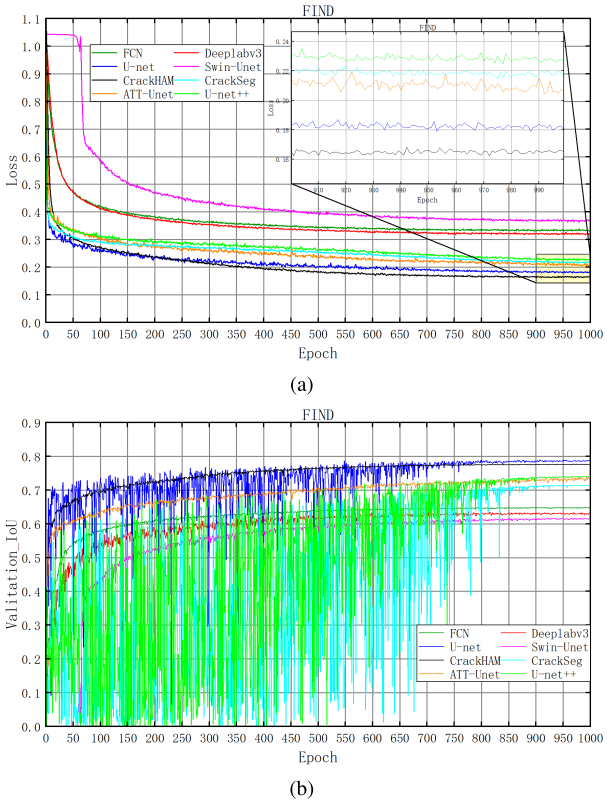

FIGURE 19. Loss (a) curves during training and IoU (b) curves during validation.

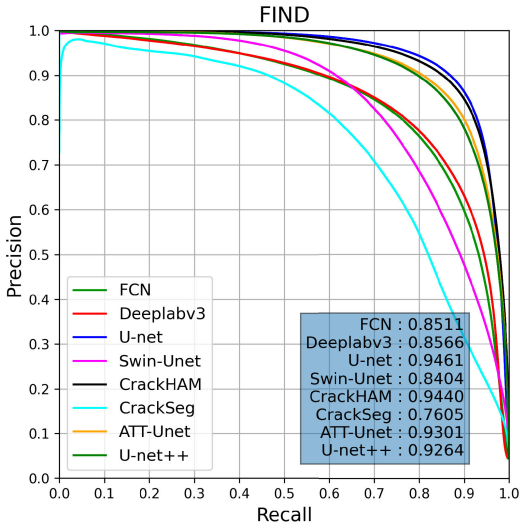

FIGURE 20. Precision-Recall curves on the FIND dataset.

However, Swin-Unet, Deeplabv3, FCN, CrackSeg, ATT-Unet and U-net++ yield poor results, as they can not fully segment the test dataset, as demonstrated in Fig. 19.

Table 7 presents the quantitative results, which demonstrate that our model achieves the highest performance on the Recall evaluation metric. Meanwhile, the other metrics

show similar values to the U-Net and our model stands out since our model distinguishes itself by virtue of its reduced parameter count. The performance of the ATT-Unet and U-Net++ models, specifically CrackSeg, is noticeably lower when compared to the performance of the CrackHAM model.

V. CONCLUSION

In this paper, we introduce a novel automatic crack detection network that incorporates several innovative modules to enhance performance while reducing computational costs. Our network architecture consists of the HASPP module, phased multi-fusion module, and attention mechanisms, which collectively contribute to improved segmentation results compared to existing mainstream network structures.

The HASPP module comprises cascading and parallel dilated convolutional layers with varying dilation rates, which expand the receptive field and capture more detailed spatial information. To capture abundant contextual information for high-level features, we employ the channel attention module. For low-level features, we leverage the spatial attention module to extract rich edge information. Meanwhile, the phased multi-fusion module is designed to extract features at varying scales from distinct network layers to facilitate prediction.

Moving forward, our future plans involve delving deeper into the extraction of specific and valuable information from crack images during the feature extraction stage, such as quantifying the length, width, and even depth of cracks. Furthermore, we intend to leverage the segmentation results to enable continuous monitoring of pavement conditions over an extended period. Additionally, we aim to apply this method to lightweight devices like unmanned aerial vehicles (UAVs), enhancing their capabilities in this domain.

This paper does not address the common challenge of operating on handheld devices or embedded platforms, which stands as a limitation. Future research endeavors will focus on investigating and addressing this challenge.

REFERENCES

[1] S. Li, Y. Cao, and H. Cai, "Automatic pavement-crack detection and segmentation based on steerable matched filtering and an active contour model," *J. Comput. Civil Eng.*, vol. 31, no. 5, Sep. 2017, Art. no. 4017045.

[2] Z. Chen, L. Deng, Y. Luo, D. Li, J. M. Junior, W. N. Gonçalves, A. A. M. Nurunnabi, J. Li, C. Wang, and D. Li, "Road extraction in remote sensing data: A survey," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, Aug. 2022, Art. no. 102833.

[3] L. Ma and J. Li, "SD-GCN: Saliency-based dilated graph convolution network for pavement crack extraction from 3D point clouds," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 111, Jul. 2022, Art. no. 102836.

[4] 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, Feb. 2022, Art. no. 126162.

[5] F. Panella, A. Lipani, and J. Boehm, "Semantic segmentation of cracks: Data challenges and architecture," *Autom. Construct.*, vol. 135, Mar. 2022, Art. no. 104110.

[6] 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.

[7] H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *Proc. 17th Eur. Signal Process. Conf.*, Aug. 2009, pp. 622–626.

[8] 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.

[9] T. Merazi-Meksen, M. Boudraa, and B. Boudraa, "Mathematical morphology for TOFD image analysis and automatic crack detection," *Ultrasonics*, vol. 54, no. 6, pp. 1642–1648, Aug. 2014.

[10] P. Prasanna, K. J. Dana, N. Gucunski, B. B. Basily, H. M. La, R. S. Lim, and H. Parvardeh, "Automated crack detection on concrete bridges," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 591–599, Apr. 2016.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[12] 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.

[13] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

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

[15] S. L. H. Lau, E. K. P. Chong, X. Yang, and X. Wang, "Automated pavement crack segmentation using U-Net-based convolutional neural network," *IEEE Access*, vol. 8, pp. 114892–114899, 2020.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[17] A. Ji, X. Xue, Y. Wang, X. Luo, and W. Xue, "An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement," *Autom. Construct.*, vol. 114, Jun. 2020, Art. no. 103176.

[18] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Autom. Construct.*, vol. 99, pp. 52–58, Mar. 2019.

[19] 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.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[21] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[22] 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.

[23] Z. Qu, L. Lin, Y. Guo, and N. Wang, "An improved algorithm for image crack detection based on percolation model," *IEEJ Trans. Electr. Electron. Eng.*, vol. 10, no. 2, pp. 214–221, Mar. 2015.

[24] 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.

[25] 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.

[26] 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.

[27] Y. Que, Y. Dai, X. Ji, A. K. Leung, Z. Chen, Z. Jiang, and Y. Tang, "Automatic classification of asphalt pavement cracks using a novel integrated generative adversarial networks and improved VGG model," *Eng. Struct.*, vol. 277, Feb. 2023, Art. no. 115406.

[28] N. H. T. Nguyen, S. Perry, D. Bone, H. T. Le, and T. T. Nguyen, "Two-stage convolutional neural network for road crack detection and segmentation," *Expert Syst. Appl.*, vol. 186, Dec. 2021, Art. no. 115718.

[29] W. Wang and C. Su, "Automatic concrete crack segmentation model based on transformer," *Autom. Construct.*, vol. 139, Jul. 2022, Art. no. 104275.

[30] 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.

[31] Y. Pan and L. Zhang, "Dual attention deep learning network for automatic steel surface defect segmentation," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 37, no. 11, pp. 1468–1487, Sep. 2022.
[32] 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.
[33] Z. Qu, C.-Y. Wang, S.-Y. Wang, and F.-R. Ju, "A method of hierarchical feature fusion and connected attention architecture for pavement crack detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16038–16047, Sep. 2022.
[34] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2672–2680.
[35] 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.
[36] 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.
[37] J. Yang, H. Li, J. Zou, S. Jiang, R. Li, and X. Liu, "Concrete crack segmentation based on UAV-enabled edge computing," *Neurocomputing*, vol. 485, pp. 233–241, May 2022.
[38] W. Choi and Y.-J. Cha, "SDDNet: Real-time crack segmentation," *IEEE Trans. Ind. Electron.*, vol. 67, no. 9, pp. 8016–8025, Sep. 2020.
[39] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.
[40] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," 2014, *arXiv:1412.6856*.
[41] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[42] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1451–1460.
[43] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.
[44] J. Shang, J. Xu, A. A. Zhang, Y. Liu, K. C. P. Wang, D. Ren, H. Zhang, Z. Dong, and A. He, "Automatic pixel-level pavement sealed crack detection using multi-fusion U-Net network," *Measurement*, vol. 208, Feb. 2023, Art. no. 112475.
[45] S. Woo, J. Park, J.-Y. Lee, and I. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2018, pp. 3–19.
[46] 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.
[47] S. Zhou, C. Canchila, and W. Song, "Deep learning-based crack segmentation for civil infrastructure: Data types, architectures, and benchmarked performance," *Autom. Construct.*, vol. 146, Feb. 2023, Art. no. 104678.
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[49] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.
[50] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham, Switzerland: Springer, 2017, pp. 240–248.
[51] 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.
[52] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 205–218.
[53] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, and X. Shen, "Image-based concrete crack detection in tunnels using deep fully convolutional networks," *Construct. Building Mater.*, vol. 234, Feb. 2020, Art. no. 117367.
[54] O. Oktay, J. Schlemper, L. Le Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-Net: Learning where to look for the pancreas," 2018, *arXiv:1804.03999*.
[55] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Proc. 4th Int. Workshop Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support (DLMIA), 8th Int. Workshop, ML-CDS, Held Conjunct With MICCAI*. Granada, Spain: Springer, Sep. 2018, pp. 3–11.
[56] 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.
[57] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

MIANQING HE received the B.E. degree in civil engineering from Northeastern University, Shenyang, China, in 2017. He is currently pursuing the master's degree with the School of Civil Engineering, Universiti Sains Malaysia, Pulau Pinang, Malaysia.

His research interests include deep learning, the segmentation of structural cracks, and the detection of surface defects.

TZE LIANG LAU received the B.Eng. degree in civil engineering and the M.Sc. degree in structure engineering from the School of Civil Engineering, Universiti Sains Malaysia, and the joint Ph.D. degree in civil engineering from Chulalongkorn University, Thailand, and the Tokyo Institute of Technology, Japan.

He is currently an Associate Professor with the School of Civil Engineering, Universiti Sains Malaysia. His research interests include structural dynamics, microtremor observation, earthquake and tsunami disaster mitigation, and hydrodynamic forces on structures.