Selective Feature Fusion and Irregular-Aware Network for Pavement Crack Detection

Xu Cheng, *Member, IEEE*, Tian He, Fan Shi, *Member, IEEE*, Meng Zhao, Xiufeng Liu, and Shengyong Chen, *Senior Member, IEEE*

***Abstract*— Road cracks on highways and main roads are among the most prominent defects. Given the inherent inaccuracy, time-consuming nature, and labor intensiveness of manual road crack detection, there's a compelling need for automated solutions. The irregular shape of cracks, along with complex background conditions encompassing varying lighting, tree shadows, and dark stains, poses a significant challenge for computer vision-based approaches. Most cracks exhibit irregular edge patterns, which are pivotal features for accurate detection. In response to recent advancements in deep learning within the realm of computer vision, this paper introduces an innovative neural network architecture termed the 'Selective Feature Fusion and Irregular-Aware Network (SFIAN)' designed specifically for crack detection on pavements. The proposed network selectively integrates features from multiple levels, enhancing and controlling the flow of valuable information at each stage while effectively modeling irregular crack objects. In an extensive evaluation, this paper conducts experiments on five distinct crack datasets and compares the results with twelve state-of-the-art crack detection methods, including the latest edge detection and semantic segmentation techniques. The experimental findings demonstrate the superior performance of the proposed method, surpassing baseline methods by a notable margin, with an increase of approximately 13.3% in the F1-score, all without introducing additional time complexity. Furthermore, the model achieves real-time processing, achieving a remarkable speed of 35 frames per second (FPS) on images at 320×480 pixels, facilitated by NVIDIA 3090 hardware.**

***Index Terms*— Deep learning, irregular-aware, pavement crack detection, selective feature fusion.**

I. Introduction

ROAD crack is one of the most representative road defects, which can damage the performance of the road surface, and in today's increasingly developed traffic, road cracks pose a severe threat to traffic safety. Therefore, it is necessary to perform necessary road inspections and to locate and repair cracks in time to prevent further deterioration of road conditions. The traditional method is to have inspections performed by professionals and to judge based on subjective experience. This method is time-consuming and labor-intensive. Therefore, it is necessary to perform automatic crack detection to reduce the workload of professionals.

In the early years, most methods tried to extract predefined features for crack detection to identify cracks. Based on the assumption that actual crack regions are darker than uncracked regions, Cheng et al. [1] propose a fuzzy logic algorithm and successfully detect fine cracks from noisy pavement images. However, they fail to detect cracks from low-contrast images between cracked and uncracked regions. In addition, Oliveira [2] propose a dynamic thresholding method based on information entropy, but their method is sensitive to noise in crack images. Since the crack is fine and can appear as an edge, Zhou et al. [3] propose a wavelet-based edge detection algorithm to identify the crack regions. However, it fails to correctly handle the cracks with high curvature or poor continuity, due to the anisotropic characteristics of wavelets. Abdel-Qader et al. [4] analyze Sobel, Canny, and fast Haar transformations that are suitable for crack detection as edge detectors. Based on the fact that cracks are similar to a series of neighboring pixels' paths, while the sum of intensities differs from others, Amhaz et al. [5] propose a crack linking method to improve crack continuity. However, its robustness is low and regularly produces intolerable false positives. Although the above rule-based methods are easy to implement, they have poor generality due to noise interference.

With machine learning attracting more and more attention, many methods exploit machine learning to extract crack features for pattern classification. For example, Gavilan et al. [6] extract an optimal texture-based feature vector to classify different pavement types by a linear SVM-based classifier. However, the classification results are not satisfactory when dealing with complex texture-mixed pavement images. Zalama et al. [7] propose a method using descriptors obtained by Gabor filters banks and identifying cracks using the AdaBoost algorithm. However, the information about the features can easily be tangled up in noise. As these machine learning-based methods focus on local patterns, they fail to robustly represent the features without a good overview.

Recently, due to its excellent representational capability, deep learning has been widely applied to the field of computer vision to address the challenge of object detection, including crack detection. In the beginning, the convolutional neural network [8] is used to find a boundary box for each possible crack object in an image by working in window sliding mode. Cha et al. [9] propose a modified Faster R-CNN architecture for

Manuscript received 7 May 2022; revised 13 October 2022, 19 May 2023, and 7 October 2023; accepted 16 October 2023. This work was supported by the National Natural Science Foundation of China under Grant 62306212, Grant 62020106004, and Grant 62272342. The Associate Editor for this article was K. C. Leung. *(Corresponding author: Fan Shi.)*

Xu Cheng, Tian He, Fan Shi, Meng Zhao, and Shengyong Chen are with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China (e-mail: shifan@email.tjut.edu.cn).

Xiufeng Liu is with the Department of Technology, Management and Economics, Technical University of Denmark, 2800 Kongens Lyngby, Denmark.

Digital Object Identifier 10.1109/TITS.2023.3325989

detecting five types of damages. Next, Zhang et al. [10] pre-classify the cracks by employing a CNN to remove most of the noise areas before proceeding with crack detection. However, the algorithms of these methods have a high time complexity, which makes it difficult to use large images. Gustavo et al. [11] propose a method to detect concrete spalling automatically with a depth sensor to quantify multiple instances simultaneously. Later, with the emergence of FCN [12], end-to-end crack detection becomes popular because it can generalize pixel-level crack prediction with much lower computational costs. It treats crack detection as a binary classification task for each image. Choi and Cha [13] propose a real-time crack segmentation method which can effectively negate various complex backgrounds and crack-like features. Cracks in an image can be considered curvilinear structures like the edges of the images from the global perspective, and they have a specific width that can reflect the degree of damage from the local perspective. In recent, Xie et al. [14] propose an edge detection method, called HED, by modifying VGG-16, which performs multi-scale side edge prediction maps in each pooling stage and fuses high-level and low-level features after upsampling for the final prediction. Unlike HED, RCF [15] extracts richer features from edge detection using all convolution layers at each stage. For crack characteristics from both perspectives, the most recent work combines edge detection [14] [15] and image segmentation methods [12] [16] and achieved good results. Inspired by HED, Liu et al. [17] propose a deep crack network for crack detection and reweight the contribution of side output feature maps for the final prediction.

However, there are two major challenges: the fusion process of multi-scale features by simple addition or concatenation process can degrade the performance, and the sensitivity to gradient information can lead to an unclear background. In an attempt to solve the former problem, Yang et al. [18] propose a pyramid-based crack detection method, which has shown better performance. The feature pyramid network obtains richer features by fusing top-down neighborhood stage features layer by layer, and hierarchical boosting pays more attention to complex samples by reweighting them during training. However, there is too much redundant information through concatenation or feature pyramiding, resulting in small objections missing and thick crack boundaries. Thus, designing a module to effectively improve the performance of multi-level fusion and to consider crack detection and localization by enhancing useful features and reducing useless features is a critical challenge. Gradient information is used for simple edge detection, which can reflect the object's boundary, especially for crack detection. Guo et al. [19] use the gradient of a real image as guidance to refine the coarse crack prediction and achieve accurate crack detection and localization. However, it is much more dependent on the accurate location of the crack on the coarse crack prediction map. Moreover, there is also too much redundant gradient information, resulting in an unclear background with more false positives. Therefore, how to utilize the gradient efficiently and model cracks attentively is another challenge.

This paper proposes a novel deep learning method of selective fusion and irregular-aware network(SFIAN) for pavement crack detection by fusing multi-scale useful features at each stage and modeling irregular crack detection. First, we specially develop a selective fusion module to integrate multi-level features containing high-resolution features with crack texture information and low-resolution features with crack semantic information. The selective fusion module selectively introduces crack localization representation at high levels to other lower levels and shares crack-specific representation at low levels to other higher levels, filling the gap between levels interactively and efficiently. Then, an irregular-aware module is designed to model cracked objects. Each irregular-aware block can build a non-rigid-aware region flexibly by learning the offset of feature maps, thereby focusing on crack gradient information and modeling cracks. Experiments show that our SFIAN can gain a much clear background with fewer false positives in challenge scenarios such as the gravel road, non-uniform illumination, and the small contract between the cracked and uncracked regions, etc., and the SFIAN performs favorably compared with the existing state-of-the-art methods on five datasets. The contributions of this work can be summarized as follows.

- We develop a novel neural network architecture for crack detection. In this architecture, we propose a selective multi-level feature fusion approach, which can fully exploit useful feature information while reducing redundant feature information.
- We introduce an irregular-aware module to the SFIAN, where each irregular-aware block can build a non-rigid aware region flexibly by learning the offset of feature maps, thereby focusing on crack gradient information and modeling crack.
- We conduct extensive experiments to evaluate the proposed method on five datasets and compare it with state-of-the-art methods. The results demonstrate the effectiveness and superiority of the proposed method.

The remainder of this paper is organized as follows. Section II reviews related work; Section III describes the details of the proposed method; Section IV conducts experiments to evaluate the model. Section V concludes the paper and presents future research directions.

II. Related Works

This section will first provide an overview of existing crack detection methods, including traditional and current deep learning-based methods. Then, we will discuss the techniques related to the proposed method.

A. Crack Detection

1) Traditional Crack Detection Methods: Over the past few decades, many attempts have been made to study pavement crack detection. Early work mainly focuses on wavelet transform or threshold-based crack detection methods.

From the perspective of vision, cracks are usually darker than the background under the same lighting conditions. In [2], Oiveirathe et al. propose an automatic segmentation of road cracks using the entropy and the dynamic threshold of an

image. However, uneven illumination will destroy its robustness, and selecting the threshold is another challenge. Since background noise is a heavy interference for crack detection, in [3], a wavelet transform is used to separate distress from noise by transforming them into high- and low-amplitude wavelet coefficients but fails to deal with low continuity cracks. On the other hand, text analysis methods have been studied in the past 30 years. In [20], they take advantage of the texture feature of local pixel-to-pixel information, but fail to fuse neighboring information and identify the intensity homogeneity of cracks.

Most recent non-deep learning methods focus on minimal path selection and traditional machine learning. Kass et al. [21] first propose a minimal path-based method to extract simple open curves, and Kaul et al. [22] propose a novel method to detect the same type of contour-like image structure with less prior knowledge about the endpoints of the desired curves and topology. In [5] and [23], an improved endpoint selection method is proposed to avoid false detection of assimilating loops.

With the increase in image data size, machine learning methods have become an important branch in crack detection with the principle of separating crack pixels from the background. In [24], a random structure forest is proposed to generate a high-performance crack detector for identifying arbitrarily complex cracks, and it uses integral channel features to obtain a better representation of cracks with intensity homogeneity. In [25], a general-purpose curvilinear structure detector is proposed, which takes advantage of B-COSFIRE for nonlinear filtering. This method is suitable for the delineation of curvilinear and elongated structures.

2) DCNN for Crack Detection: Due to the unprecedented success of deep learning in computer vision, much work has been done to use deep convolutional neural networks (DCNN) in crack detection. In [26], Zhang et al. first apply DCNN to detect cracks in a patch-based manner that trains the image of small cropped patches and classifies them as cracks or not. Moreover, this method proves that DCNN has better feature representation capability than handcrafted feature-based methods, but it is hard to use and sensitive to patch scale. Recently, most DCNN methods have treated crack detection as binary segmentation and considered crack edge features. In [27], Zou et al. present a crack segmentation method based on HED [14], which is employed in the edge detection domain, and combine the feature maps of each encoder and decoder stage to achieve a final prediction that is not sensitive to noisy crack labeling and has good crack processing ability. To handle the imbalance between easy and hard samples, Yang et al. [18] propose a feature pyramid and hierarchical boosting network (FPHBN) [18] for the model to focus on hard samples by re-weighting samples layer by layer. For thicker boundaries and blurred edges, in [19], Guo et al. introduce the original image gradient generated by Sobel into coarse crack detection and produce a more accurate crack boundary by refining the network for crack localization. Cha et al. introduce an efficient CNN-based method for automated concrete crack detection, outperforming traditional techniques [8]. Cha et al. extends this work, achieving high-accuracy multi-type structural damage detection in real-time videos with Faster R-CNN [9]. Beckman et al. develop a low-cost depth sensor-based method using Faster R-CNN for concrete spalling detection and volume quantification [11]. Kang et al. propose an autonomous UAV solution for structural health monitoring, demonstrating high sensitivity and specificity in concrete crack detection [28]. Ali et al. enhance defect detection with a modified Faster R-CNN in GPS-denied UAV environments, reducing false positives effectively [29].

B. Multi-Level Feature Fusion

It turned out that the deep layer lacks detailed information in the shallower layers. To address this issue, in [12], Long et al. exploit predictions from middle layers to obtain fine segmentation details. In [27], Zou et al. bridge the encoder layer and the corresponding decoder layer of the U-net and pairwise fuse the convolutional features generated from them, and then fuse the feature maps of all layers for the final prediction by simple concatenation. In [30], Lin et al. introduce contextual information into the lower-level layer by locally fusing each adjacent feature to generate a feature pyramid. These fusion methods operate a lack of feature utility awareness.

Therefore, useful features should be measured and propagated efficiently. To address this issue, Akilan et al. [31] proposed a new model that performs convolution of multiple filters with different scales on the same input for better FG(foreground) object/region identification. Zhang et al. [32] proposed a new feature extraction module based on the network model of `darknet19` and `darknet53` to improve the detection accuracy of small targets for inspection of rail surface defects. Hu et al. [33] proposed an FSDet module to extract deep features adaptively for the detection of stationary wear from railway switches. And Xu et al. [34] use gates for multi-model features fusion in multiple auxiliary tasks. In [35], Takikawa et al. leverage gates to shape stream by classical stream, allowing precise boundary information to be learned. In [36], Li et al. use gates to selectively fuse features from multi-levels in a fully connected way for small/thin objection segmentation. Inspired by these works, we develop a selective fusion module for fusing multi-level features interactively, which can make full use of all level features to gain a clear background.

C. Deformable Modeling Mechanism

In DCNN, most traditional convolutions are used as a priori knowledge to sample spatial locations, and the essential drawback of convolutional computation is that it does not take into account the unknown nature of convolutional networks for geometric changes, leading to inefficient utilization of model and data capacity. Some recent related works have also used deformation modeling to solve the problem. In [37], Jaderberg et al. propose a Spatial Transformation Network (STN) to train a spatially invariant model by learning a two-dimensional affine transformation. In [38], Roco et al. use Deep Geometric Matchers to learn a thin-plate spline transformation to solve the problem of determining the correspondences between two images in agreement with

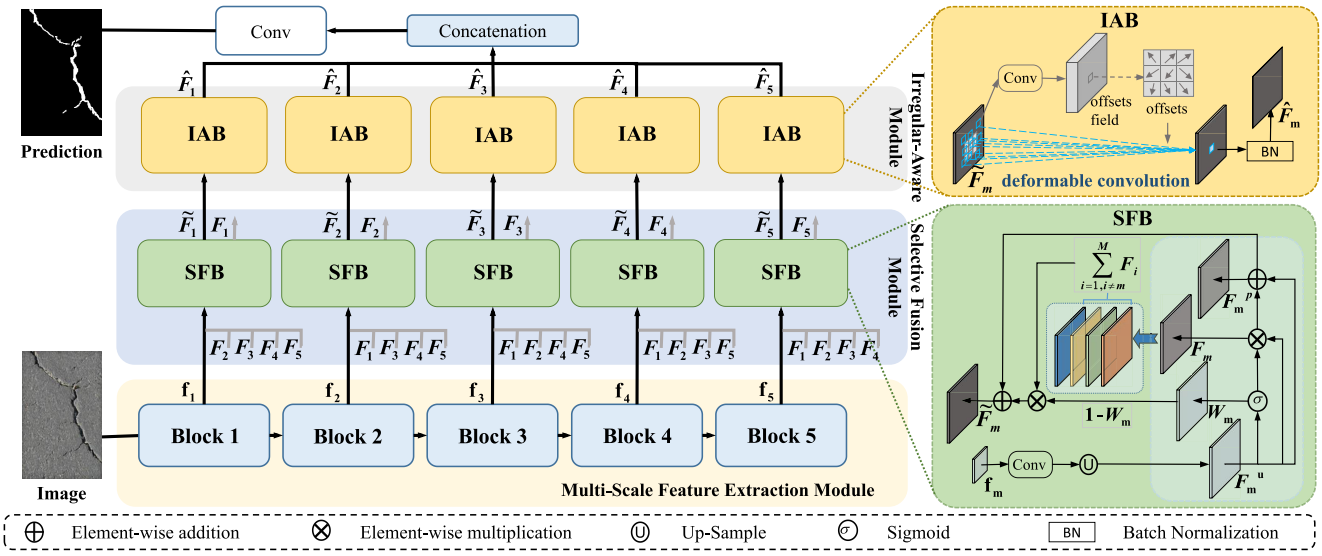


Fig. 1. Illustration of our proposed SFIAN road crack architecture. Details of the Irregular-aware Block (IAB) and Selective Fusion Block (SFB) are provided on the right. f_m and $\tilde{F}_m$ refer to the m-th outputs of multi-scale feature extraction module and the selective fusion module, respectively.

a geometric model. In [39], Dai et al. propose a deformable convolution based on conventional convolution for modeling nonrigid objects in target detection and semantic segmentation, which increases the spatial sampling position in the module using an additional offset that is relearned from the target task with almost no additional increase in model parameters. In addition, deformable convolution was used to model complex geometrically transformed tracking objects in [40] and as a feature alignment function to contextually align higher-level upsampled features in [41]. Thus, in this paper, we introduce the irregular-aware module to flexibly perceive crack gradient information and further model crack objects attentively for accurate crack prediction.

III. Proposed Method

A. Overview Network

In this paper, crack detection is designed as a pixel-wise binary classification task. A designed model produces a crack prediction map for a given crack image, where cracked regions have a high probability, and uncracked regions have a low probability. Fig. 1 is an overview of the proposed method, SFIAN. SFIAN consists of four major parts: Multi-Scale Feature Extraction Module, Selective Fusion Module, Irregular-Aware Module, and Side Supervision Strategy.

In particular, for a given crack image and the corresponding ground truth, the image is first fed into the multi-scale feature extraction module, which contains five stages. For the extracted feature map at each level, a weighted feature map F_m and a fused feature map $\tilde{F}_m$ are produced by the corresponding Selective Fusion Block (SFB). Then, the Irregular-Aware Block (IAB) models the irregular crack objects based on the fused image and generates a side coarse crack prediction compared to ground truth (GT) directly through a deep supervision network strategy [42]. Finally, the five features are concatenated to produce a final crack prediction.

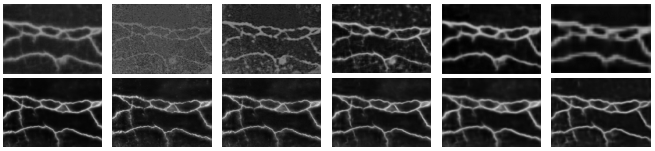


Fig. 2. Visual crack prediction of the side outputs and directly fused results from HED(the first row) and SFIAN(the second row).

B. Multi-Scale Feature Extraction Module

In our method, the multi-scale feature extraction module is based on holistic nesting edge detection (HED) [14]. The HED network uses the first 13 layers of VGG-16 to extract features. Specifically, it consists of stages 1-5, and the first four stages are followed by a max-pooling layer with a stride of 2, respectively, which is used to extract features of different scales. With the max-pooling layer, the size of feature maps is reduced by twice, stage by stage.

The multi-scale feature extraction network extracts features with different spatial resolutions. A high-resolution feature map contains more texture features, which can detect crack information, especially fine cracks; a fine-resolution feature map contains more obvious semantic information, which can better locate cracks. However, if these feature maps are fed directly into each side network, the output of different side networks will differ. Fig. 2 shows five side outputs and the output after fusion from HED and SFIAN, respectively. It can be seen that side-output 1 and side-output 2 contain a lot of texture information, such as fine cracks, but there are also other edge interference in the non-cracked area; while side outputs 4 and 5 contain richer semantic information and can position the cracks, but there is a problem of unclear edges. If these measurement outputs are directly fused, the fusion is blind, and the low-level edge interference and the high-level crack blur boundary will be merged into the fused image at the same time, resulting in disordered and blurred fused images.

C. Selective Feature Fusion Module

In order to address the above problems, we implement a selective feature fusion module to selectively aggregate contextual information, inspired by gated fully fusion [36]. Each feature map produced by each level is fed into SFB to aggregate useful information.

The general data flow of the proposed SFB is presented to the right of Fig. 1. Specifically, to begin with, the depth of the input feature map f_m is unified into 16 dimensions by a point-by-point convolution to highlight the importance of the feature map and suppress redundant feature maps. These multi-feature maps contain texture information and semantic information, respectively, and reflect the effect of each feature vector. To bridge the gap between feature maps of different sizes, we upsample them to the same size by bilinear interpolation. Next, the rescaled feature map F_m^u is weighted with the importance feature weight response map W_m which is activated by a sigmoid function, and then the re-weighted feature map F_m is added to F_m^u, known as plus feature maps F_m^p. Finally, F_m from other levels are introduced at the current level, selectively reinforcing important feature maps and removing unnecessary features. In general, the SFB process can be formulated as follows:

$$F_m = F_m^u \times W_m, \tag{1}$$

$$\tilde{F_m} = F_m^p + (1 - W_m) \sum_{i=1, i\neq m}^{M} F_i. \tag{2}$$

where $W_m \in [0, 1]^{H\times W}$ denotes as the m-th weighted responding map, $\sum_{i=1,i\neq m}^{M} F_i$ is the sum of all reweighted feature maps except m-th based on element-wise addition.

Conceptually, as shown in Eq.(2), for the feature vector at position (x, y) at level i, if the value of W_i is large and W_m is small, the useful feature at i can be fed to m level to reinforce the feature performance. On the other hand, if W_m is larger, it means that the feature vector at the same position of the m level is more useful, and redundant information about features at other levels can be avoided accordingly.

D. Irregular-Aware Module

Convolution aims to sample the input feature map at a fixed position through a regular grid. But for modeling geometry and multi-scale target, ordinal convolution is insufficient. It is necessary to model irregular crack targets for crack detection, which is often a challenge. Therefore, we design a module called Irregular-Aware Module(IAM) to model crack geometric targets from fused feature maps. Specifically, the 3×3 deformable convolution with dilation of 1 is applied to the filtered and fused features to generate a more accurate, discriminative, and robust final feature fusion map, followed by a batch normalization layer. Fig. 1 shows some important details of IAB implementation. First, we feed $\tilde{F_m} \in \mathbb{R}^{H\times W}$ as the input feature map to IAB. We define a $k \times k$ conventional convolution layer. Here, the inner convolutional layer has the same kernel size and dilation as the deformable convolution. For the output feature y at any location p, we can obtain the following:

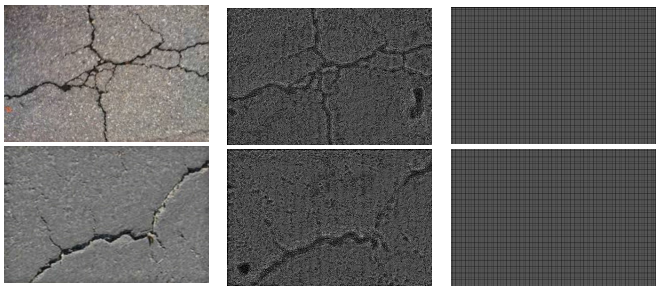


Fig. 3. Visual comparisons of the proposed Irregular-Aware Block (IAB) and Ordinary-Aware Mechanism (OAM).

$$y(p) = \sum_{n=1}^{N} w(p_n) \cdot x(p + p_n). \tag{3}$$

where $N = | k \times k |$ and n is the n-th convolution sample location; x is the input feature map; p_n and $w(p_n)$ are the enumerations of the locations in regular grid and the weight, respectively. For different sample locations, the offsets Δp_n are learned through the adaptive application of a deformable convolution. Eq. (3) can be reformulated as follows:

$$y(p) = \sum_{n=1}^{N} w(p_n) \cdot x(p + p_n + \Delta p_n). \tag{4}$$

where each Δp_n is implemented by the bilinear interpolation operation.

As shown in Fig. 3, the OAM uses original convolution to make aware the target objects of images regularly through a rigid grid, while the proposed IAB can perceive the gradient information of the irregular boundary of cracks and establish a non-rigid sensing area to model the target by learning the displacement of the irregular target. It is better to model the changes of target dense space after learning features sufficiently. Through our IAB, the fracture feature-aware map can more accurately focus on the object with more gradient change, making the cracks more distinguishable from the interference and the background.

E. Side Supervision Strategy

Due to the low feature discrimination of hidden layer learning and insufficient feature transparency in the current deep learning framework, Lee et al. [42] propose a side-supervised learning method. This side-supervised learning method adds middle-side supervision in the hidden layer and directly supervises the model layer by layer. The side-supervised method is beneficial in improving the robustness and discriminative nature of the model in feature learning, leading to faster convergence. It has been widely used in computer vision, especially crack detection, with high pixel positioning accuracy. In this paper, an improved binary cross-entropy focal loss function training model is used to implement the side-supervision, which is defined as follows:

$$L_{bce} = -ylogy' - (1 - y)log(1 - y'). \tag{5}$$

where the label y can be 1/0, and L_{bce} can be rewritten as follows:

$$L_{bce} = \begin{cases} -(1-y')logy', & if\ y=1, \\ -y'log(1-y'), & if\ y=0. \end{cases} \tag{6}$$

For this binary cross-entropy, for positive samples, the higher the output probability, the lower the loss; for negative samples, the lower the output probability, the lower the loss. At this point, the loss function is relatively slow in the iterative process of many simple samples and may not be optimized to the optimal state. The pixel difference between the positive and negative samples, i.e., cracked and non-cracked areas, is big for typical crack segmentation.

To balance the weights between classes, we calculate the loss using the focal loss by introducing a weight parameter that gives more attention to the difficult and misclassified samples. The formula is defined as follows:

$$L_{fl} = \begin{cases} -(1-y')^{\gamma}logy', & if\ y=1, \\ -y'^{\gamma}log(1-y'), & if\ y=0. \end{cases} \tag{7}$$

where γ represents the focus parameter to control the weights of easy-to-classify and hard-to-classify samples. Based on prior knowledge, in this paper we use $\gamma = 2$ to measure the contribution of hard-to-classify and easy-to-classify samples to the total loss.

Therefore, our side-output layers and the final fused layer all adopt the focal loss mentioned above. The overall loss function is the addition of those six loss.

IV. Experiments and Results

In this section, we first describe the experimental parameters of the proposed SFIAN approach, including implementation details, datasets, comparison methods, and evaluation metrics; then we present the experimental results by comparing baselines; and finally we discuss ablation analysis.

A. Experimental Settings

1) Datasets: We use the following five datasets for the evaluation, including CRACK500, GAPs384, Cracktree206, CFD, and AEL. The five datasets have different crack shape characteristics from different environments.

- **CRACK500:** This dataset contains 500 images of pavement cracks captured by a mobile camera and is approximately 2000×1500 in size. The dataset is divided into 250 images for training, 50 images for validation, and 200 images for testing. Due to the limited number of images, in this paper, we use 1,896 images for training and 1,124 images for testing as [18].
- **GAPs384:** This dataset contains 509 images (640×540-pixel and $440\times$ 540-pixel) pavement distress of cracks, which is selected from GAPs [43] dataset as [18] and renamed as GAPs384. Due to the brightness changes between images and among the areas of some images, it is challenging to match the true crack locations and boundaries.
- **Cracktree206:** This dataset contains 206 pavement images (800 $\times$ 600-pixel) obtained by [44] with fine cracks. In addition to the fact that the width of cracks is difficult to discriminate, it is even more difficult to detect the cracks under a complex background, such as the tree shadows on the cracks, low contrast, noise, etc.
- **CFD:** This dataset contains 118 images (480×320-pixel) collected using iPhone on pavement road with an smooth clean background. The crack shape in this dataset is fine, similar to Cracktree206, but with a different background, such as brighter sunlight and coverage of objects on the road.
- **AEL:** This is a composite dataset composed of Aigle-RN & ESAR & LCMs. It contains 38 images collected by the Aigle-RN system at traffic speed. ESAR contains 15 crack images collected by a static acquisition system in an environment without controlled lighting. LCMs contain five crack images.

For the last four data split, we randomly choose training-test data according to the split ratio of CRACK500 dataset. In our network, training data are first augmented by following [17].

2) Implementation Details: We implement our network using the deep learning framework, Pytorch v1.11, and run the experiments on the server with two Nvidia GTX 3090 GPUs. In training, the initial global learning rate is set to 1e-4. We employ the Stochastic Gradient Descent (SGD) method to optimize our network parameters. The momentum and weight decay are set to 0.9 and 2e-4, respectively, and the batch size is 1. We train the network with 90 epochs for the first four datasets, including CRACK500, GAPs384, Cracktree206, and CFD. The learning rate is reduced by 0.03 after 60 epochs. We train the network with 450 epochs for the AEL dataset, and the learning rate is reduced by 0.006 after 300 epochs due to the fewer AEL samples. The loss weight for each side-output layer and the final fused layer are all 1.0. The model is saved for every epoch, we choose the best performing model as final model. Besides, in the training stage, we apply similar data augmentation to our implementation as in [17], including rotation and flipping. The raw images and the augmented data are both used for training.

3) Comparison Methods: We compare the performance of SFIAN with state-of-the-art methods. We used the same datasets for all methods for a fair comparison.

- **HED [14]**: HED is a groundbreaking work for edge detection. The hyperparameters are set as default during the training process, except for the learning rate, which is set to 1e-5.
- **RCF [15]**: RCF is an extension work based on HED for edge detection. The hyperparameters are set as shown in RCF [15] except for the learning rate, which is set to 1e-5.
- **FCN [12]**: We adopt FCN-8s [12] and replace the loss faction with a sigmoid cross-entropy loss. The hyperparameters are set the same as in our work.
- **DC [17]**: It achieves the balance between five side-output features. We train DC [17] as default.
- **DC [27]**: The DeepCrack net was built on the encoder–decoder architecture of SegNet and the convolutional features which were generated in the encoder

network and in the decoder network at the same scale, were pairwise fused. We train DC [27] by default.

- **PidiNet [45]**: PidiNet is a simple and light effective edge detection architecture, which adopts novel pixel-different convolutions that decrease memory and energy consumption. We train PidiNet [45] as default.
- **BDCN [46]**: BDCN is proposed to extract edges at dramatically different scales. It supervises each individual layer by labeled edges at its specific scale and generates multi-scale features by using dilated convolution. The hyperparameters are set as default.
- **RIND [47]**: RIND is proposed to jointly detect different types of edges, taking into account the distinct attributes of each type of edge, and to capture the underlying relations between them. We train RINDnet [47] as default.
- **UCTNet [48]**: UCTNet is proposed an alternative of the skip connections for segmentation from the channel perspective, which conducts the multi-scale channel cross fusion with transformer and guides the fused information to connect the decoder features. We train UCTNet [48] as default.
- **Unet++ [49]**: Unet++ redesigns skip pathways to reduce the semantic gap based on Unet. We train Unet++ in our architecture, the hyperparameters and loss function are set the same as in our work.
- **DeeplabV3+ [50]**: DeeplabV3+ combines the advantages of the spatial pyramid pooling module and encode-decoder structure to encoder multi-scale contextual information and capture sharper object boundaries. We train DeeplabV3+ in our architecture, the hyperparameters and loss function are set the same as [50].
- **AttUnet [51]**: AttUnet proposes an attention gate model to focus of target structure of varying shapes and sizes automatically, which can highlight salient features useful and suppress irrelevant regions. We train AttUnet in our architecture, the hyperparameters and loss function are set the same as default.

4) Evaluation Metric: The final value of crack prediction is between 0 and 1. Therefore, we need to define a specific threshold for binary crack prediction. We use four main metrics in our evaluation, including precision (P), recall(R), F1-score($F1 = 2\frac{P \cdot R}{P+R}$) and mean intersection over union(mIoU). In addition, the Optimal Dataset Scale(ODS) and Optimal Image Scale (OIS) are introduced. Specifically, ODS means the best F1-score across the entire dataset for a fixed threshold to evaluate the model prediction performance over all images. OIS means an aggregate F1-score chosen for the best scale in each image to evaluate the model prediction performance for each image. More importantly, the F1-score is good at balancing the influence of precision and recall and is able to evaluate a classifier comprehensively.

$$ODS = max\{2\frac{P_t \cdot R_t}{P_t + R_t}\}, \tag{8}$$

$$OIS = \frac{1}{N_{img}} \sum_{i}^{N_{img}} max\{2\frac{P_t^i \cdot R_t^i}{P_t^i + R_t^i}\}. \tag{9}$$

TABLE I
QUANTITATIVE EVALUATION ON CRACK500 DATASET

Methods	ODS	OIS	P	R	F_1	mIoU
HED [14]	0.6032	0.6428	0.6015	0.7102	0.6514	0.7254
RCF [15]	0.6595	0.7003	0.6668	0.7410	0.7019	0.7378
FCN [12]	0.6835	0.7221	0.6925	0.7566	0.7237	0.7517
DC [17]	0.6184	0.6616	0.6162	0.6980	0.6546	0.7120
DC [27]	0.6319	0.6940	0.6606	0.7246	0.6911	0.7265
PidiNet [45]	0.6366	0.6821	0.6662	0.7108	0.6878	0.7261
BDCN [46]	0.2287	0.2361	0.3349	0.1822	0.2360	0.5384
RIND [47]	0.6482	0.7049	0.6798	0.7297	0.7039	0.7333
UCTNet [48]	0.6957	0.7210	0.6816	0.7861	0.7302	0.7576
Unet++ [49]	0.6569	0.6852	0.6697	0.7357	0.7012	0.7373
DeeplabV3+[50]	0.6760	0.7181	0.6769	0.7495	0.7113	0.7700
AttUnet[51]	0.7041	0.7257	0.6980	0.7851	0.7390	0.7769
Our	0.6977	0.7408	0.6983	0.7742	0.7343	0.7604

TABLE II
QUANTITATIVE EVALUATION ON GAPS384 DATASET

Methods	ODS	OIS	P	R	F_1	mIoU
HED [14]	0.3795	0.4245	0.4051	0.4224	0.4230	0.6186
RCF [15]	0.3937	0.4388	0.4152	0.4344	0.4249	0.6251
FCN [12]	0.4783	0.5122	0.4593	0.5521	0.5014	0.6579
DC [17]	0.4388	0.4879	0.4520	0.4967	0.4733	0.6361
DC [27]	0.3715	0.4297	0.3335	0.4664	0.3890	0.6133
PidiNet [45]	0.4656	0.5208	0.4279	0.5100	0.4652	0.6558
BDCN [46]	0.4693	0.5229	0.5043	0.5153	0.5098	0.6576
RIND [47]	0.5524	0.6054	0.5745	0.5898	0.5820	0.6979
UCTNet [48]	0.5795	0.6064	0.5568	0.6236	0.5884	0.7112
Unet++ [49]	0.5240	0.5610	0.5430	0.5550	0.5489	0.6837
DeeplabV3+[50]	0.5143	0.5656	0.6038	0.5929	0.5180	0.6950
AttUnet[51]	0.5273	0.5744	0.4986	0.5464	0.5214	0.7012
Our	0.5804	0.6213	0.5643	0.6172	0.5895	0.7080

where the threshold $t \in [0.01, 0.99]$, with an interval of 0.01, and precision and recall at a specific threshold are P_t and R_t, respectively. N_{img} and i are the total number of samples and the index of images, respectively. Moreover, P_t^i and R_t^i are computed over an image, i.

B. Comparison With the State-of-the-Art Methods

Table I∼V show the qualitative results of ten methods, respectively, which are all based on deep learning. Fig. 4 shows the visualization results of these methods in five data sets.

1) Results on CRACK500: According to Table I, our proposed SFIAN achieves outstanding performance on the CRACK500 dataset, boasting F1-score and mIoU values of 0.7343 and 0.7604, respectively. These metrics surpass the performance of other methods, trailing behind AttUnet by a margin of approximately 0.6% and 2.2%, respectively. Notably, FCN, UCTNet, and DeeplabV3+ also exhibit commendable results. In Fig. 4, the first and second rows depict probability images of samples extracted from the CRACK500 dataset. The first row illustrates SFIAN's ability to differentiate between deep-colored non-cracked marks and cracks. In the second row, SFIAN excels in providing a clearer background with fewer false positives, even in the presence of stone edge noise on the gravel road.

2) Results on GAPs384: Table II shows the quantitative results of the comparison methods. In terms of the F1-score, the proposed method achieves the best performance on GAPs384 with a value of 0.5895. Compared to the others,

TABLE III
QUANTITATIVE EVALUATION ON CRACKTREE206 DATASET

Methods	ODS	OIS	P	R	F_1	mIoU
HED [14]	0.3017	0.3164	0.2511	0.4324	0.3178	0.5865
RCF [15]	0.3280	0.3447	0.2859	0.4986	0.3397	0.5962
FCN [12]	0.2010	0.2032	0.1240	0.5740	0.2039	0.5486
DC [17]	0.3061	0.3168	0.2416	0.4768	0.3207	0.5827
DC [27]	0.3397	0.3578	0.2935	0.4071	0.3411	0.6006
PidiNet [45]	0.4450	0.4649	0.4613	0.4450	0.4529	0.6442
BDCN [46]	0.5681	0.5718	0.4991	0.6824	0.5765	0.6995
RIND [47]	0.4407	0.4426	0.3270	0.7027	0.4463	0.6388
UCTNet [48]	0.5806	0.5830	0.4738	0.7772	0.5887	0.7019
Unet++ [49]	0.4363	0.4493	0.4201	0.4737	0.4453	0.6385
DeeplabV3+[50]	0.2598	0.3649	0.4745	0.5208	0.4966	0.6153
AttUnet[51]	0.3054	0.3347	0.2242	0.3931	0.3067	0.5996
Our	0.5801	0.5906	0.5535	0.6373	0.5925	0.7045

TABLE IV
QUANTITATIVE EVALUATION ON CFD DATASET

Methods	ODS	OIS	P	R	F_1	mIoU
HED [14]	0.4978	0.5272	0.4593	0.5684	0.5080	0.6645
RCF [15]	0.5470	0.5752	0.5376	0.5779	0.5570	0.6909
FCN [12]	0.4494	0.4636	0.4212	0.3771	0.4674	0.6371
DC [17]	0.5794	0.6030	0.5347	0.6471	0.5856	0.6695
DC [27]	0.4357	0.4962	0.5372	0.4756	0.5045	0.6242
PidiNet [45]	0.5467	0.5883	0.5280	0.6238	0.5719	0.6898
BDCN [46]	0.5563	0.5834	0.5313	0.5831	0.5560	0.6948
RIND [47]	0.6221	0.6390	0.5885	0.6859	0.6317	0.7319
UCTNet [48]	0.6713	0.6822	0.6532	0.7160	0.6832	0.7771
Unet++ [49]	0.5976	0.6187	0.5695	0.6482	0.6063	0.7170
DeeplabV3+[50]	0.5702	0.6180	0.4835	0.6861	0.5673	0.7190
AttUnet[51]	0.6305	0.6456	0.6155	0.6197	0.6176	0.7342
Our	0.6401	0.6579	0.6200	0.6901	0.6532	0.7418

the proposed method demonstrates performance improvements on the F1-score as 16.65%, 16.46%, 8.81%, 11.62%, 20.05%, 12.43%, 7.97%, 0.75%, 0.11%, 4.06%, 7.15%, and 6.81%, respectively. And UCTNet holds an mIoU value of 0.7112, which is 0.32% higher than the proposed method. The third and fourth rows in Fig. 4 show the crack detection samples of GAPs384. With the interference of non-uniform illumination and the small contract between the cracked and uncracked regions, our proposed method obtains fewer false positives and accurately detects the cracks.

3) Results on Cracktree206: As shown in Table III, SFIAN outperforms the others, which has an F1-score of 0.5925 and an mIoU value of 0.7045. BDCN achieves a commendable result with an F1-score and mIoU value of 0.5765 and 0.6995, respectively, while UCTNet has the second-highest F1-score and mIoU value of 0.5887 and 0.7019, respectively. The visualization results of the crack image samples are shown in the fifth and sixth rows of Fig. 4 on this dataset. The results show that for the challenging scene where the cracks are almost shadowed by the tree (the fifth row), our method can still detect the cracks correctly and produces a much clearer crack prediction than the others on the fine crack image (the sixth row).

4) Results on CFD: The results in Table IV show that the proposed method obtains the second-highest performance in terms of F1-score in the CFD dataset, exceeding the F1-score of the baseline methods except UCTNet by 14.58%, 9.68%, 18.64%, 6.82%, 14.93%, 8.13%, 9.72%, 2.15%, 2.30%, 4.69%, 8.59%, and 3.56%, respectively, which is lower than UCTNet by 3%. The proposed method achieves 0.7418 on the mIoU, which is lower than UCTNet by about 3.53%. The probability values of the crack image samples are shown in the seventh and eighth rows of Fig. 4. This dataset has more challenging scenes, such as cracks in lanes (seventh row). SFIAN can generate a much clearer prediction image with a sharp boundary and more continuous cracks.

TABLE V
QUANTITATIVE EVALUATION ON AEL DATASET

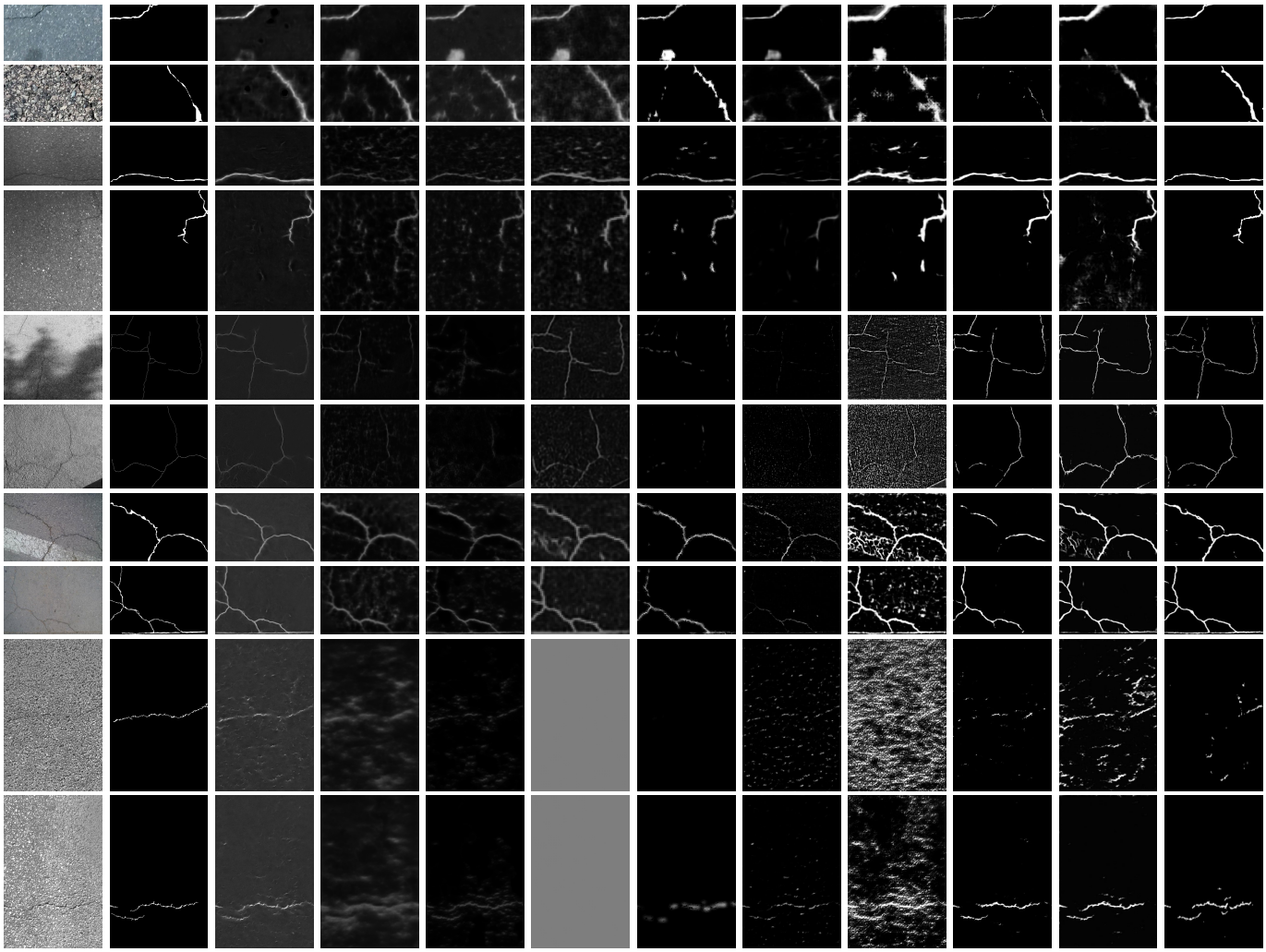
Methods	ODS	OIS	P	R	F_1	mIoU
HED [14]	0.2788	0.3040	0.3031	0.3597	0.3290	0.5808
RCF [15]	0.5341	0.5734	0.6099	0.5698	0.5892	0.6900
FCN [12]	N/A	N/A	N/A	N/A	N/A	N/A
DC [17]	0.2445	0.2653	0.2681	0.2765	0.2723	0.5578
DC [27]	0.4110	0.4488	0.4381	0.4602	0.4489	0.6333
PidiNet [45]	0.4331	0.4687	0.5760	0.4621	0.4876	0.6460
BDCN [46]	0.6013	0.6272	0.7174	0.6391	0.6760	0.7255
RIND [47]	0.5538	0.5840	0.5966	0.6523	0.6232	0.6977
UCTNet [48]	0.5676	0.5829	0.6185	0.6499	0.6338	0.7071
Unet++ [49]	0.4501	0.4564	0.4989	0.501	0.4999	0.6517
DeeplabV3+[50]	0.3592	0.3735	0.3478	0.4313	0.3851	0.6101
AttUnet[51]	N/A	N/A	N/A	N/A	N/A	N/A
Our	0.6231	0.6473	0.6910	0.6820	0.6865	0.7373

5) Results on AEL: The results of the small sample of the AEL dataset are shown in Table V and the last two rows in Fig. 4. Clearly, our proposed method outperforms the others, with an F1-score of 0.6865 and an mIoU value of 0.7373. The F1-score and mIoU values of BDCN are 0.6760 and 0.725 respectively, which outperform other compared methods. Surprisingly, FCN and AttUnet both produce gray crack prediction maps, and DC [17] also obtains a lower F1-score of 0.2723. This is probably due to the small number of training samples and the low contrast between the background and fine cracks. In other words, it is shown that SFIAN can perform well in a small sample dataset. The results of the small sample of the AEL dataset are shown in Table V and the last two rows in Fig. 4. Clearly, our proposed method outperforms the others, with an F1-score of 0.6865 and an mIoU value of 0.7373. The F1-score and mIoU values of BDCN are 0.6760 and 0.7255, respectively, which outperform other compared methods.

C. Complexity Analysis

The complexity analysis of deep learning-based methods can be found in Table VI. We assess computational complexity and model memory footprint by measuring floating-point operations per second (FLOPs) in gigaflops (GFLOPs) and the number of parameters in millions (M). These measurements are taken with tensor inputs sized at $512 \times 512 \times 3$ for all the previously mentioned deep learning methods as well as our proposed method.

Although our proposed method has more parameters than some of the comparison methods, such as Unet++, which has about 9.16M learnable parameters, the comparison results demonstrate our method's superior performance. Furthermore, our method has fewer learnable parameters than some other methods, such as RINDNet and UCTransNet, which have about 59M and 66M learnable parameters, respectively. Therefore, our method offers a reasonable trade-off between speed and accuracy. Our proposed method demonstrates its trade-off



(a) Raw (b) GT (c) Ours (d) HED (e) RCF (f) FCN (g) DC [17] (h) DC [27] (i) PidiNet (j) BDCN (k) RIND (l) UCTNet

Fig. 4. Qualitative results obtained by different methods on five datasets. Note that ten sample images (from top to bottom) are selected from CRACK500, GAPs384, Cracktree206, CFD and AEL in turn (with two images from each dataset).

TABLE VI
COMPLEXITY ANALYSIS OF EACH METHOD

Methods	FLOPs	Params
HED [14]	80.35G	14.72M
RCF [15]	115.54G	15.52M
FCN [12]	80.46G	14.72M
DC [17]	80.35G	14.72M
DC [27]	136.86G	30.9M
PidiNet [45]	11.60G	0.59M
BDCN [46]	143.86G	16.30M
RIND [47]	695.77G	59.39M
UCTNet [48]	171.73G	66.22M
Unet++ [49]	139.61G	9.16M
DeeplabV3+[50]	88.53G	59.34M
AttUnet[51]	541.34G	57.16M
Our	84.45G	14.75M

between timing and memory footprint, where it has less time (FLOPs) and space (Params) complexity while providing superior performance compared to the former methods.

D. Ablation Analysis

In this section, we conduct the ablation study on a mixed dataset to evaluate the validity of the proposed SFIAN models, i.e., SFM and IAM. The mixed dataset contains 2,594 images from CRACK500, Cracktree206, GAPs384, and DeepCrack [17], for training, and 314 images from CFD and DeepCrack, for testing. We compare the following three variants of SFIAN: 1) **Base(cat)**, which fuses multi-level features by concatenation without SFM and IAM based on SFIAN; 2) **Base(add)**, which fuses multi-level features by addition without SFM and IAM based on SFIAN; 3) **SFIAN_No_IAM**, where IAM was removed and SFM was reserved based on SFIAN.

The experiments use the settings in Section IV-A, and the results are shown in Table VII. From the results, we can observe that a naive application of addition (for feature fusion) adversely affects the performance, while our proposed SFM alone (see the row of SFIAN_No_IAM) can significantly improve the performance of the original **Base(cat)**, with an improvement of **5.84 points** in F1-score. Empirically, we observe that, compared to the SFM used alone, our proposed IAM further improves the F1-score by about **2.47 points** (see the row of SFIAN). One plausible reason why the feature of our methods can integrate better is related to the non-selective nature of commonly used fusion operations (i.e.,

TABLE VII
QUANTITATIVE RESULT OF ABLATIVE ANALYSIS EXPERIMENTS ON MIXED DATASET

Methods	ODS	OIS	P	R	F_1	FLOPs	Params
Base(cat)	0.5421	0.6016	0.5863	0.6694	0.6251	**80.352G**	**14.72M**
Base(add)	0.4892	0.5482	0.4861	0.5998	0.5370	80.353G	**14.72M**
SFIAN_No_IAM	0.6238	0.6692	0.6432	**0.7966**	0.6835	80.859G	14.742M
SFIAN	**0.6775**	**0.7376**	**0.6771**	0.7425	**0.7082**	84.451G	14.745M

TABLE VIII
QUANTITATIVE COMPARISON BETWEEN DIFFERENT NUMBER OF UNIFIED CHANNEL ON MIXED DATASET

	ODS	OIS	P	R	F_1	FLOPs	Params
C=4	0.6090	0.670	0.6283	0.7321	0.6763	**81.272G**	**14.725M**
C=8	0.6264	0.6944	0.6349	0.7223	0.6758	82.4G	14.732M
C=16	**0.6775**	**0.7376**	**0.677**	0.7425	**0.7082**	84.451G	14.745M
C=32	0.6620	0.7361	0.641	**0.7818**	0.7044	88.554G	14.772M

addition and concatenation). However, the simple addition or concatenation of multiple features is insufficient, suggesting the need for better-designed methods. From the results, we can conclude that the proposed selective fusion module and the irregular-aware module are critical to the performance of SFIAN, thus validating its effectiveness.

E. Different Number of Unified Channel Analysis

In Section III-C, we formulated the selective fusion block by setting the unified channel as 16, although the model performance is different with it. We will show the advantages of this setting with experiments. Specifically, we set the unified channel as 4/8/16/32. It can be seen in Table VI, SFIAN equipped with a small number of unified channels will have a lower performance, which indicates that it is not good to have too much dimension reduction. The computational complexity and the performance (F1-score) of the model increase with the growing number of unified channels, and it achieves a comparable performance when C=16 and 32, while when the dimension is reduced to 16, the model achieves a balance between performance and computational complexity.

F. Limitations and Discussions

Through the experimental results, we can find that our method can achieve comparative performance in various datasets. Nevertheless, there are several limitations to our proposed methods, which warrant consideration:

1) Our method has exclusively undergone testing on benchmark datasets characterized by simple backgrounds, such as pristine pavements. Consequently, there is an imperative for further assessments on datasets featuring more intricate background scenes. This is vital for evaluating the robustness and generalizability of our method. In contrast to previous works, such as SDDNet [13] and STRNet [52], which are evaluated within the context of complex background scenes rather than pure pavements, our method has yet to undergo such comprehensive scrutiny. Potential strategies to enhance our method's performance encompass employing a more advanced backbone network to extract richer features for crack detection, leveraging additional data augmentation techniques to augment diversity and robustness within the training data, and integrating attention mechanisms or adversarial learning strategies to refine the crack segmentation results. Moreover, the exploration of more adaptive evaluation metrics capable of adjusting to varying datasets and scenarios could be beneficial.

2) Our method exhibits a slower processing speed and demands lower-resolution images compared to previous works like SDDNet and STRNet. This reduced processing speed constrains the applicability of our proposed model, particularly on mobile or edge devices. One potential solution to address this limitation is model pruning, a technique that removes redundant weights or neurons from the model, thereby reducing its size and enhancing inference speed without significantly compromising accuracy.

V. CONCLUSION AND FUTURE WORK

The detection of irregular shape of cracks in pavements is challenging due to complex background conditions. This paper proposed a novel neural network method, SFIAN, to detect irregular cracks in pavement images. In this method, a Selective Fusion Module bridges the gap between high-resolution with low semantics and low-resolution with high semantics so that the missing low-level features can be fused into each feature level. This neural network structure can handle small and fine objects well and obtain clear crack boundaries. To further improve the detection capability, we incorporated an Irregular-Aware Module into the proposed SFIAN to model geometric cracks and proposed a multi-scale feature extraction module to extract features with different spatial resolutions. We conducted experiments to evaluate the proposed method comprehensively and compared it with state-of-the-art methods. The results demonstrate the superiority of the proposed method over others. In addition, the model processes in real-time (35 FPS in NVIDIA 3090) images at 320×480 pixels.

For future work, we intend to develop an adaptive loss function for the side output to improve training. Based on the characteristics of the different datasets, rather than a fixed contribution of each stage side-output to the final prediction, a flexible side-loss weighting mechanism will improve the potential of the proposed method. Furthermore, instead of being limited to static images, we intend to apply our model to video [53], [54]. We plan to investigate and implement

a lightweight video-based crack detection model for small device detection for physical inspections.

REFERENCES

[1] H. D. Cheng, J.-R. Chen, C. Glazier, and Y. G. Hu, "Novel approach to pavement cracking detection based on fuzzy set theory," *J. Comput. Civil Eng.*, vol. 13, no. 4, pp. 270–280, Oct. 1999.

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

[3] J. Zhou, "Wavelet-based pavement distress detection and evaluation," *Opt. Eng.*, vol. 45, no. 2, Feb. 2006, Art. no. 027007.

[4] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *J. Comput. Civil Eng.*, vol. 17, no. 4, pp. 255–263, Oct. 2003.

[5] 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.

[6] M. Gavilán et al., "Adaptive road crack detection system by pavement classification," *Sensors*, vol. 11, no. 10, pp. 9628–9657, 2011.

[7] E. Zalama, J. Gómez-García-Bermejo, R. Medina, and J. Llamas, "Road crack detection using visual features extracted by Gabor filters," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 29, no. 5, pp. 342–358, 2014.

[8] 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, Feb. 2017.

[9] 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.

[10] K. Zhang, H. D. Cheng, and B. Zhang, "Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning," *J. Comput. Civil Eng.*, vol. 32, no. 2, Mar. 2018, Art. no. 04018001.

[11] G. H. Beckman, D. Polyzois, and Y.-J. Cha, "Deep learning-based automatic volumetric damage quantification using depth camera," *Autom. Construct.*, vol. 99, pp. 114–124, Mar. 2019.

[12] 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.

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

[14] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[15] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5872–5881.

[16] 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.

[17] 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.

[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] J.-M. Guo, H. Markoni, and J.-D. Lee, "BARNet: Boundary aware refinement network for crack detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7343–7358, Jul. 2022.

[20] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Comput. Intell.*, vol. 20, no. 1, pp. 18–36, Feb. 2004.

[21] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, Jan. 1988.

[22] 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.

[23] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "A new minimal path selection algorithm for automatic crack detection on pavement images," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 788–792.

[24] H. Oh, N. W. Garrick, and L. E. Achenie, "Segmentation algorithm using iterative clipping for processing noisy pavement images," in *Proc. Imag. Technol., Techn. Appl. Civil Eng., 2nd Int. Conf. Eng. Found.; Imag. Technol. Committee Tech. Council Comput. Practices, Amer. Soc. Civil Eng.*, 1998, pp. 138–147.

[25] A. Cord and S. Chambon, "Automatic road defect detection by textural pattern recognition based on AdaBoost," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 27, no. 4, pp. 244–259, Apr. 2012.

[26] 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.

[27] 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.

[28] D. Kang and Y. Cha, "Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 10, pp. 885–902, Oct. 2018.

[29] R. Ali, D. Kang, G. Suh, and Y.-J. Cha, "Real-time multiple damage mapping using autonomous UAV and deep faster region-based neural networks for GPS-denied structures," *Autom. Construct.*, vol. 130, Oct. 2021, Art. no. 103831.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[31] T. Akilan, Q. M. J. Wu, and W. Zhang, "Video foreground extraction using multi-view receptive field and encoder–decoder DCNN for traffic and surveillance applications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 9478–9493, Oct. 2019.

[32] H. Zhang et al., "MRSDI-CNN: Multi-model rail surface defect inspection system based on convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11162–11177, Aug. 2022.

[33] X. Hu, Y. Cao, Y. Sun, and T. Tang, "Railway automatic switch stationary contacts wear detection under few-shot occasions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14893–14907, Sep. 2022.

[34] D. Xu, W. Ouyang, X. Wang, and N. Sebe, "PAD-Net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 675–684.

[35] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5228–5237.

[36] X. Li, H. Zhao, L. Han, Y. Tong, S. Tan, and K. Yang, "Gated fully fusion for semantic segmentation," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 34, no. 7, 2020, pp. 11418–11425.

[37] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2017–2025.

[38] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 39–48.

[39] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[40] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable Siamese attention networks for visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6727–6736.

[41] S. Huang, Z. Lu, R. Cheng, and C. He, "FaPN: Feature-aligned pyramid network for dense image prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 844–853.

[42] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. Artif. Intell. Statist.*, 2015, pp. 562–570.

[43] M. Eisenbach et al., "How to get pavement distress detection ready for deep learning? A systematic approach," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2039–2047.

[44] 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.

[45] Z. Su et al., "Pixel difference networks for efficient edge detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5097–5107.

[46] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "BDCN: Bi-directional cascade network for perceptual edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 100–113, Jan. 2022.

[47] M. Pu, Y. Huang, Q. Guan, and H. Ling, "RINDNet: Edge detection for discontinuity in reflectance, illumination, normal and depth," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6859–6868.

[48] H. Wang, P. Cao, J. Wang, and O. R. Zaiane, "UCTransNet: Rethinking the skip connections in U-Net from a channel-wise perspective with transformer," 2021, *arXiv:2109.04335*.

[49] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Granada, Spain: Springer, 2018, pp. 3–11.

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

[51] O. Oktay et al., "Attention U-Net: Learning where to look for the pancreas," 2018, *arXiv:1804.03999*.

[52] 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.

[53] D. Kang and Y. Cha, "Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 10, pp. 885–902, Oct. 2018.

[54] R. Ali, D. Kang, G. Suh, and Y.-J. Cha, "Real-time multiple damage mapping using autonomous UAV and deep faster region-based neural networks for GPS-denied structures," *Autom. Construct.*, vol. 130, Oct. 2021, Art. no. 103831.

Xu Cheng (Member, IEEE) received the Ph.D. degree in engineering from the Intelligent Systems Laboratory, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology (NTNU), Ålesund, Norway, in June 2020.

From June 2020 to March 2022, he was a Post-Doctoral Fellow and a Researcher with the Department of Manufacturing and Civil Engineering, Gjøvik, Norway. Since April 2022, he has been a Permanent Researcher with Smart Innovation Norway, Halden, Norway. He is currently a Full Professor with the Tianjin University of Technology, Tianjin, China. He has applied for and coordinated more than five projects supported by the Norwegian Research Council (NFR), the EU, and industry. He has published more than 60 articles as the first and coauthor in his research interests, including data analysis and artificial intelligence in maritime operations, time series analysis, and predictive maintenance of wind turbines.

Tian He received the B.Eng. degree in computer science and technology from Qinghai University, Xining, China, in 2019, and the M.Sc. degree from the Tianjin University of Technology, Tianjin, in 2023. Her research interests include data analysis, time series analysis, and predictive maintenance.

Fan Shi (Member, IEEE) received the Ph.D. degree from Nankai University, Tianjin, China, in 2012. From June 2018 to August 2019, he was a Research Scholar with West Virginia University. He is currently a Professor with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin. His research interests include machine vision, pattern recognition, and optics.

Meng Zhao received the B.S. degree in automation and the M.S. and Ph.D. degrees in control science and engineering from Tianjin University, Tianjin, China, in 2010 and 2016. Since 2016, she has been a Lecturer with the School of Computer Science and Engineering, Tianjin University. From May 2019 to April 2020, she was a Post-Doctoral Researcher supported by the European Research Consortium for Informatics and Mathematics (ERCIM) "Alain Bensoussan Fellowship Program." Her research interests include medical image processing, medical/biomedical engineering, and machine learning/deep learning in medical informatics.

Xiufeng Liu received the Ph.D. degree in computer science from Aalborg University, Denmark, in 2012. He was a Post-Doctoral Researcher with the University of Waterloo and a Research Scientist with IBM, Canada, from 2013 to 2014. He is currently a Senior Researcher with the Department of Technology, Management and Economics, Technical University of Denmark. His research interests include smart meter data analysis, data warehousing, energy informatics, and big data.

Shengyong Chen (Senior Member, IEEE) received the Ph.D. degree in computer vision from the City University of Hong Kong. He has been conducting research on vision sensors for robotics for more than 20 years. From 2006 to 2007, he received a Fellowship from the Alexander von Humboldt Foundation of Germany. He was with the University of Hamburg, Germany. From 2008 to 2012, he was a Visiting Professor with Imperial College London and the University of Cambridge, U.K. He is currently a Full Professor with the Tianjin University of Technology and the Director of the Engineering Research Center of Learning-Based Intelligent System (Ministry of Education). He has published over 300 scientific papers in international journals and conferences, including 80 articles in IEEE TRANSACTIONS. He also published more than ten books in the past years and applied more than 100 patents. He received the National Outstanding Youth Foundation Award of NSFC. He organized about 20 international conferences and serves as an Associate Editor for three international journals, such as IEEE TRANSACTIONS ON CYBERNETICS.