# Real-Time Pavement Damage Detection With Damage Shape Adaptation

Yingchao Zhang and Cheng Liu

***Abstract*— Intelligent detection of pavement damage is crucial to road maintenance. Timely identification of cracks and potholes helps prolong the road service life. Current detection models fail to balance accuracy and speed. In this study, we propose a fast damage detection algorithm named FPDDN to achieve real-time and high-accuracy pavement damage detection. FPDDN integrates the deformable transformer, D2f block, and SFB module to predict pavement damage of different sizes in multiple branches. The deformable transformer allows the FPDDN to exhibit adaptability to geometric variations in road defects, thereby improving the detection accuracy of irregular defects such as cracks. D2f block is mainly used to lightweight the network and increase the inference speed. The SFB module can significantly decrease the loss of information during downsampling of small-sized objects. This integration enhances the model's ability to extract global damage features, reduces the loss of information on small-scale defects, and improves the synergy between deep and shallow feature layers. The model's performance was evaluated using the RDD2022 dataset, focusing on inference speed and detection accuracy. When compared to state-of-the-art models such as YOLO v8, FPDDN has a parameter count that is only one-fifth of that of YOLO v8x, yet it surpasses YOLO v8x in detection accuracy. The FPDDN achieved an F1 score of 0.601 and a mAP50 of 0.610 on the RDD2022 dataset, outperforming the compared models. Additionally, the algorithm achieved a balance between accuracy and speed with an inference speed of 1.8ms for pavement damage detection.**

***Index Terms*— Non-destructive testing, transformer, damage detection, real-time detection.**

## I. Introduction

TODAY, every country has a dense road network, which is extremely important for each country's economy, society, and defense. Efficient highway transportation can reduce the cost of transportation time for goods and significantly improve logistics efficiency. However, with the construction of highways in various countries, more and more roads begin to suffer from pavement damage, such as cracks, potholes, etc., which will affect driving comfort and may lead to traffic accidents. Therefore, it is essential to repair the damage at the early stage of its development, which will also improve the service life of the road.

There are a number of methods available for pavement damage inspection. Although manual detection is highly accurate, it requires traffic closures, which greatly impacts traffic efficiency. Multi-functional highway damage detection vehicles can detect pavement, subgrade, and other damage with high precision. However, the cost of the inspection vehicles is expensive, and they are not applicable to the frequent inspection of highway networks. Therefore, many scholars have developed intelligent detection algorithms [1], [2]. Current intelligent detection algorithms include three main categories: vibration-based sensors, 3D sensors, and cameras.

### *A. Detection Algorithms Based on Vibration Sensors*

The presence of cracks and potholes on the road surface can cause discomfort to the passengers, and we can determine the quality of the road surface according to the degree of discomfort [3]. Acceleration sensors, gyroscopes, and other devices were used to detect pavement unevenness [4], [5]. These sensors can capture vertical acceleration variations during driving with high precision. The higher the vibration amplitude, the worse the pavement. However, this method can only capture the area where the wheels pass and is ineffective for areas where the wheels do not pass. Due to its limited scope of application, it is rarely used in actual detection tasks.

### *B. Detection Algorithms Based on 3D Sensors*

3D sensors generally have higher accuracy when detecting pavement damage. Wang et al. [6] proposed a new approach for calculating the mean texture depth of pavement based on the 3D laser scanning sensors. This technique can also be used for damage detection. Reference [7] verified the feasibility of 3D sensors for detecting pavement distresses. Zhang et al. [8] applied the pavement condition detected by the 3D laser scanner and proposed the minimum cost spanning tree algorithm to improve the accuracy of crack detection to 98%. 3D sensors are highly accurate and not influenced by environmental conditions. However, the hardware and technical requirements for 3D sensor-based detection algorithms are more challenging. They are less suitable for lightweight detection devices.

### *C. Detection Algorithms Based on Cameras*

With the development of deep learning, convolutional neural networks (CNN) and Transformer structures have become

Manuscript received 27 February 2024; revised 21 April 2024; accepted 16 June 2024. This work was supported by the New Faculty Start-Up Fund from the City University of Hong Kong under Grant 9610612. The Associate Editor for this article was Y. Yu. *(Corresponding author: Cheng Liu.)*

The authors are with the Department of Systems Engineering, City University of Hong Kong, Hong Kong (e-mail: cliu647@cityu.edu.hk).

Digital Object Identifier 10.1109/TITS.2024.3416508

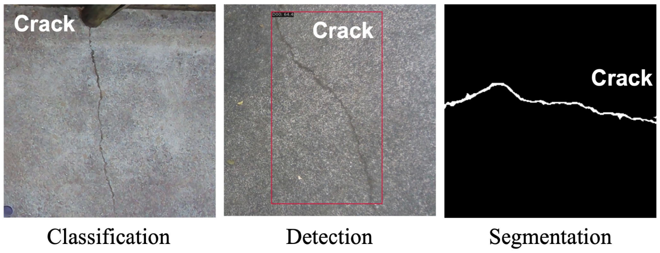


Fig. 1. Examples of different tasks.

extensively utilized in the domain of non-destructive damage detection. There methods can be divided into three main categories: damage classification [9], damage detection [10], and damage segmentation [11]. Fig.1 displays the three tasks.

AlexNet [12], and GoogleNet [13] were used by [14] to complete the classification of damage. Damage classification determines the presence of damage in the image with up to 99% accuracy in detecting pavement damage [9]. Merely determining the presence of defects in the image lacks practical application value, and we need to know the actual location of the damage in the image.

The object detection algorithm can detect the type and location of the damage in the image. Global Road Damage Detection Challenge (GRDDC) [15] was held in 2020 to search for high-precision models for pavement damage. This competition only took detection accuracy as evaluation metrics. Pham et al. [16] replaced the backbone of Faster RCNN with ResNet101, resulting in an F1-score of 0.51. The YOLO v4 model and the ensemble learning approach were used by [17] to improve the detection metric to 0.628. Using the state-of-the-art YOLO detection model, [18] used ensemble learning and data augmentation to improve the F1 score to 0.67. This result also won the competition. A lot of great models emerged from this challenge. In addition, some scholars employed the attention mechanism to improve the model accuracy. A multi-level attention mechanism [10] was proposed to enhance the ability to detect pavement damage in YOLO v3 [19]. In addition to cracks, potholes and other defects, high-precision detection of white line blur (D44) is also being carried out by [20]. Although these damage detection algorithms have high accuracy, they are insufficient in fast detection.

The damage segmentation is a more precise way to detect different types of damage, which can distinguish pixel classes [21]. Chen et al. [22] proposed a pixel-by-pixel trainable PCSN network to perform crack segmentation. Segmentation networks were applied to multiple injury categories immediately afterward [23]. However, the detection speed of the segmentation model is not fast enough, and the edge information of the damage is not processed well enough.

The above studies focused on how to improve the detection accuracy of the model, but neglected the importance of inference speed. Although the requirement for inference speed is not critical in some offline applications, increasing inference speed is still important for improving overall operational efficiency, minimizing costs, enhancing system scalability, and improving the user experience. Therefore, focusing on both accuracy and inference speed is important to improve the overall operational efficiency and user experience. Zhang et al. [24] introduced the convolution layer with a small kernel size in the segmentation network to reduce the parameters. To reduce the parameters of models based on the transformer, [25] proposed a multilayer cross-fertilization strategy that used depthwise separable convolution [26]. This network outperforms YOLO v4-tiny in terms of detection accuracy and inference speed. Liu et al. [27] used YOLO v5 [28] combined with Swin Transformer Block [29] to enhance the detection accuracy of pavement distress without significantly increasing the number of parameters. While these algorithms are superior in inference speed, they have lower accuracy and do not achieve a good balance between accuracy and speed.

Currently, there are several existing models for the detection or segmentation of pavement damage. CNN-based models are characterized by the limited number of parameters and fast detection speed but low detection accuracy [30], while Transformer is known for higher detection accuracy but slow detection speed [29]. Therefore, this study aims to propose a pavement damage detection model that effectively balances both accuracy and speed, considering the strengths of CNN and Transformer models. The contributions can be mainly summarized as follows:

1) A lightweight network for pavement damage detection called Fast Pavement Damage Detection Network (FPDDN) was proposed by us. This model focuses on irregularly shaped and small-scale pavement damage. It can adapt to irregularly shaped cracks and reduce the loss of information in the downsampling process for small-size targets. The structure of FPDDN can be seen in Fig.2.
2) Our proposed SFB module effectively magnifies small target weights in feature maps through subtraction and merging operations between feature maps.
3) We introduced vision transformer models with deformable attention into our model, which exhibits adaptability to geometric variations in road defects.
4) The D2f module based on depthwise separable convolution is proposed to lightweight the network and increase the inference speed.

The sections are divided as follows: Section I reviewed the research on detection accuracy and speed, Section II introduced currently commonly used pavement damage datasets, and Section III proposed our own model. Section IV introduced the experimental design, evaluation metrics, and results analysis. Finally, this paper was summarized.

## II. Pavement Dataset

Public datasets for pavement distress include RDD2020 [31] and UNFSI [32] for cracks classification and location, as well as Crack500 [33] and CrackForest [34] for damage segmentation. However, these databases have some limitations. For example, databases used for crack segmentation typically contain only a few hundred images, making it difficult to use them in practical inspections, even if they achieve high accuracy during training processes. These images rarely suffer from external noise when captured, and the image quality is

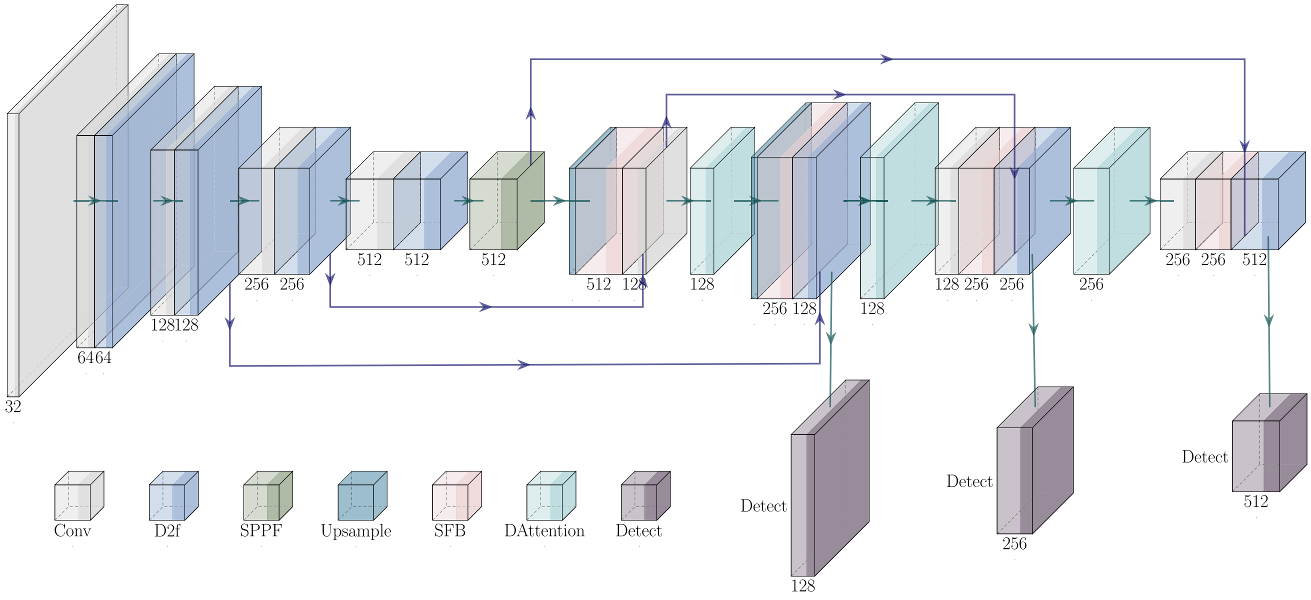


Fig. 2. The structure of FPDDN.

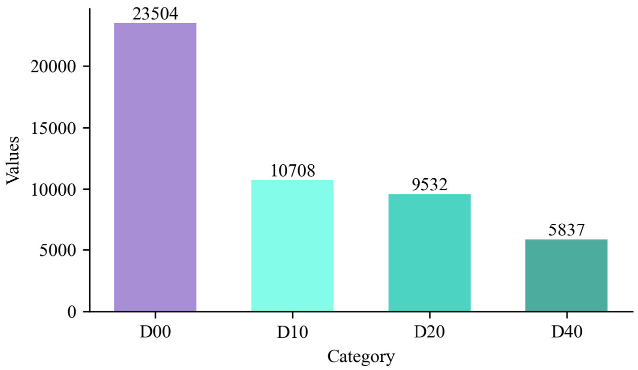


Fig. 3. Number of each category.

better. Therefore, we used the RDD2022 database proposed by Arya et al. [35] in 2022 to use the trained model for field detection. The RDD2022 database contains pavement damage images from six countries, including Japan, India, China, the United States, the Czech Republic, and Norway, and including transverse cracks (D10), longitudinal cracks (D00), alligator cracks (D20), and potholes (D40). Fig.3 and Fig.4 show the number and samples of each category of the RDD2022 database we collected.

We restricted the size of the input image to 640 × 640. If the area of the ground truth for the disease measures less than $32^2$ pixel$^2$, it is classified as a small target. Areas exceeding $96^2$ pixel$^2$ are considered large targets, while all other sizes are categorized as medium targets. The results of the division are shown in Fig. 5. The figure illustrates that RDD2022 contains a greater number of smaller targets, which can significantly affect the accuracy of the detection model.

## III. Proposed Network

### A. Overall Model of FPDDN

To utilize the lightweight property of the convolutional neural network and the high accuracy of the self-attention mechanism, we proposed the FPDDN model for rapid detection of pavement defects, as shown in Fig.2. FPDDN consists of a backbone, a feature fusion network, and three detection heads, similar to the YOLO structure [30].

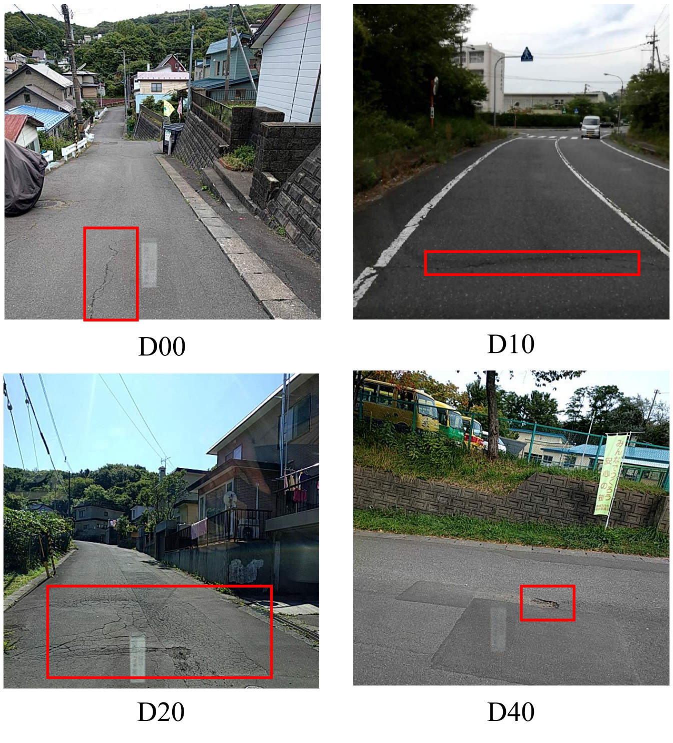


Fig. 4. Samples of different categories.

We use five convolution modules (Conv) and four D2f modules in the backbone. The Conv module comprises a convolution layer, a normalization layer, and an activation function layer. The convolution kernel size was set to 3 × 3 in the Conv module, and the stride was 2. After the feature maps pass through the convolution layer, their width and height become 1/2 of the original size. The mathematical

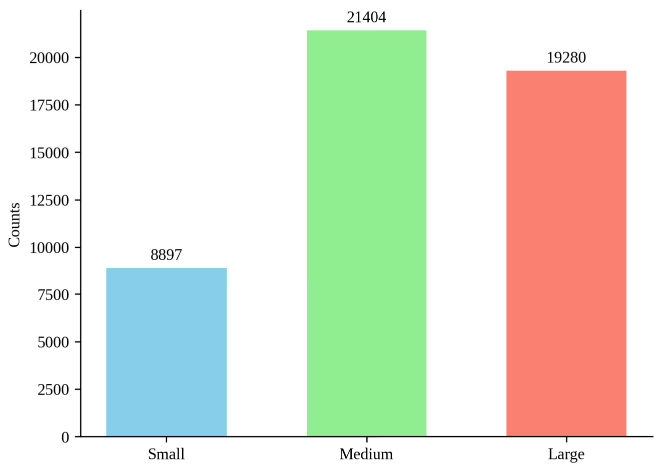


Fig. 5. Classification of defects size.

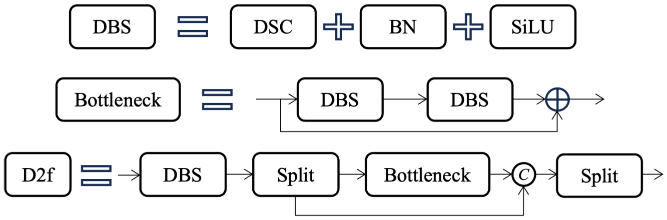


Fig. 6. D2f module.

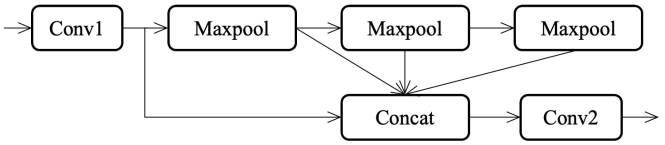


Fig. 7. SPPF module.

expressions for the normalization and activation function layers are presented in Eq.2 and Eq.3. Here, the SiLU function was used in the FPDDN. We proposed D2f based on the C2f module used in the YOLO v8 [30]. The convolutional layers in the C2f module were replaced by depthwise separable convolutions (DSC) [26] to further decrease the number of model parameters. The D2f module we adopted is shown in Fig.6. The DBS module consists of DSC, batch normalization (BN), and SiLU activation functions. Two DBS modules are adopted in the Bottleneck module, and the residual structure is used to prevent vanishing gradients and exploding gradients. In the D2f module, the DBS module processes the input feature map and splits it into two feature maps along the channel direction, one of which is input into the Bottleneck module. The three resulting feature maps are concatenated along the channel direction and subsequently processed through the DBS module. In addition, the SPPF module in the backbone is the same as that of YOLO v8, which can be seen in Fig.7.

$$(x * h)(i, j) = \sum_{c}\sum_{m}\sum_{n} x_c(m, n) \cdot h_c(i - m, j - n) \tag{1}$$

$$y = \frac{x - \mathrm{E}(x)}{\sqrt{\mathrm{Var}(x) + e^{-5}}} \tag{2}$$

$$y = \frac{x}{1 + e^{-x}} \tag{3}$$

where $h$ denotes the convolution kernel, $x$ is the input feature map, $c$ indicates the channel dimension. The variables $m$ and $n$ correspond to the traversal of the entire feature map along the height and width dimensions, respectively. The tuple $(i, j)$ specifies the position of the output feature while E and Var $(x)$ represent the mean and variance of the input feature map, respectively.

In the feature fusion stage, Feature Pyramid Network (FPN) [36] and Path Aggregation Network (PAN) [37] are used by FPDDN. In the fusion of shallow and deep feature maps, the FPDDN model employs the subtractive fusion block (SFB) module proposed by us to capture the feature information of small targets more effectively. In addition, we introduce the vision transformer with deformable attention to improve the feature extraction of irregular targets such as cracks. These modules will be described in subsequent sections. To increase the resolution of deep feature maps, the method of nearest neighbor upsampling is employed.

### *B. SFB Module*

The presence of many small-sized pavement defects in the RDD2022 dataset presents a significant challenge for detection. Improving the detection accuracy of small object defects can indirectly improve the detection accuracy of the overall dataset. FPN is a commonly used solution for small object detection problems. FPN improves small objects' classification and positioning accuracy by fusing the position information of shallow feature maps and the semantic information of deep feature maps. Since the number of pixels occupied by small targets is limited, small object information will be lost during the continuous downsampling process, and the upsampling process in the FPN cannot recover the lost information. Therefore, we proposed SFB module to enhance the weight of small target information during the feature fusion process. The structure of SFB is shown in Fig.8.

SFB is a fully convolutional module whose input is a shallow feature map $(f1)$ and a deep feature map $(f2)$. The sizes of these two feature maps are $(B, C_1, H, W)$ and $(B, C_2, H, W)$ respectively, where $B$, $H$, $W$ are the batch size, the height and the width of the feature maps, $C_1$ and $C_2$ represent the number of channels of the two feature maps. The symbol $C$ in Fig.8 denotes an arithmetic symbol, as shown in Eq.4.

$$F_c = C\begin{cases} \mathrm{Conv}(f1)_{c=128} - \mathrm{Conv}(f2)_{c=128} \\ \mathrm{Concat}(\mathrm{Conv}(f1), \mathrm{Conv}(f2)) \end{cases} \tag{4}$$

$$F_c = \{F_{c=128}, F_{c=256}\} \tag{5}$$

where $c$ is the number of channels in the feature maps.

### *C. Deformable Attention Block*

The deformable attention block (DAttention) introduces position bias based on the self-attention mechanism to improve the model's ability to detect irregular targets. The structure diagram is shown in Fig.9. The input feature map was multiplied by the projection matrix to obtain the query $(Q)$, and then the deviation of each position is calculated through

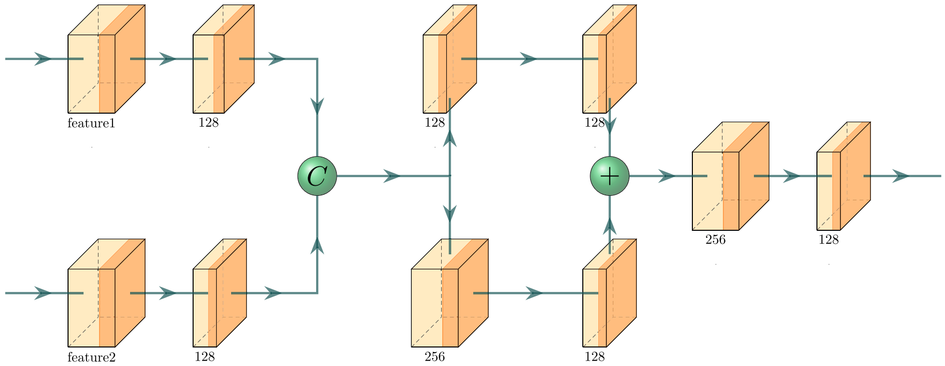


Fig. 8. SFB module.

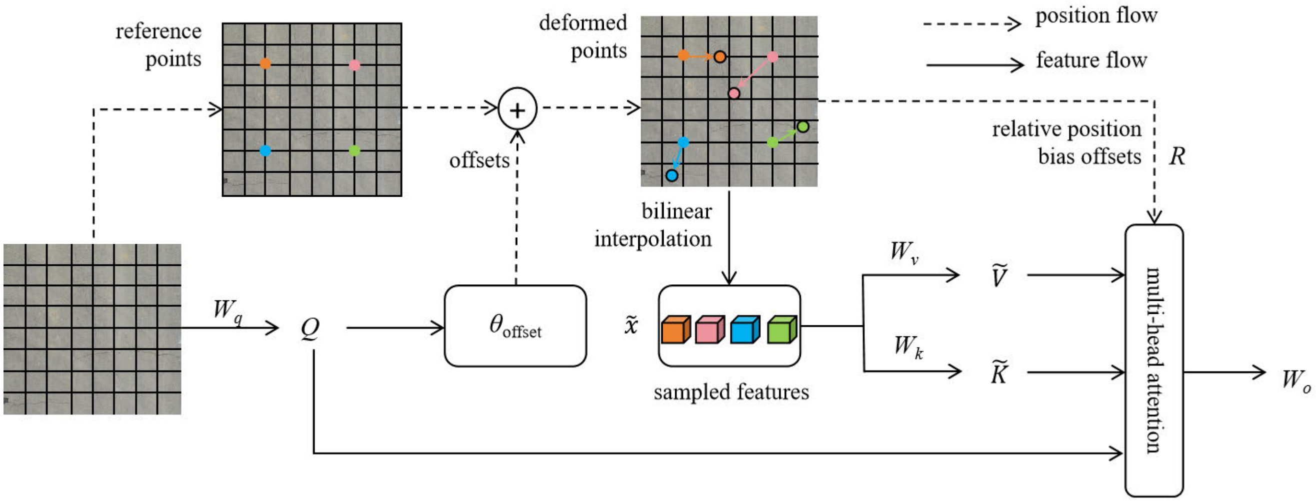


Fig. 9. Vision transformer with deformable attention.

$Q$, as shown in Eq.6-9.

$$Q = x \times W_q, K = \tilde{x} \times W_k, V = \tilde{x} \times W_v \quad (6)$$

$$\triangle p = \theta_{\text{offset}}(Q) \quad (7)$$

$$\tilde{x} = \phi(x; p + \triangle p) \quad (8)$$

$$\phi(z; (p_x, p_y)) = \sum_{(r_x, r_y)} g(p_x, r_x) g(p_y, r_y) z[r_y, r_x, :] \quad (9)$$

where $K$, $V$ are the key and value embeddings with offsets, $W_q$, $W_k$, $W_v$ represent the projection matrix, respectively, and $\triangle p$ indicates the offset of each point learned based on $Q$. $\phi(\cdot; \cdot)$ is the bilinear interpolation, and the function $g(a, b)$ represents the distance ratio between the new point and the original point, with a value between 0 and 1. The process of calculating the offset is shown in Fig.10. The $H \times W \times C$ feature map goes through several layers, including the convolution layer, Layer Normalization, and GELU activation function layer. Finally, the channel dimensions are adjusted to 2 by $1 \times 1$ convolution, representing the offsets of the feature points in the horizontal and vertical directions.

After obtaining $Q$, $K$, and $V$, the result can be calculated by applying the self-attention mechanism using Eq.10-13.

$$z^m = \sigma\left(Q^m (\tilde{K}^m)^T / \sqrt{d} + \phi(\hat{B}; R)\right) \tilde{V}^m \quad (10)$$

$$z = \text{concat}(z^{(1)}, \ldots, z^{(m)}) W_o \quad (11)$$

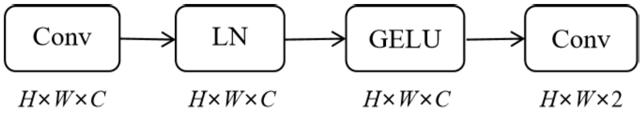


Fig. 10. The process of calculating the offset.

$$z_l' = \text{MHSA}(\text{LN}(z_{l-1})) + z_{l-1} \quad (12)$$

$$z_l = \text{MLP}(\text{LN}(z_l')) + z_l' \quad (13)$$

where $m$ denotes the $m$-th attention head, $z^{(m)}$ indicates the embedding output from the $m$-th attention head, $\sigma()$ is the softmax function, $W_o$ represents the projection matrix. Eq.12-13 is the function commonly used in multi-head attention block.

In DAttention block, learnable offsets are added to the self-attention mechanism so that the positions of reference points can be adaptively transformed to better fit the specific structure of the input data. This flexibility allows the model to better handle irregularly shaped targets such as curved cracks. The integration of DAttention allows the model to show adaptability to geometrical changes in road defects.

### *D. Loss Function*

In FPDDN, cross-entropy loss, distribution focal loss (DFL) [38], and CIoU loss are used. Cross-entropy loss is mainly

TABLE I
PLATFORMS FOR TRAINING

| Name | Value | Name | Value |
|---|---|---|---|
| CPU | Intel i9-13900K | Learning rate | 0.01 |
| GPU | Nvidia RTX 4090 | Batch size | 16 |
| System | Ubuntu 20.04 | Epochs | 100 |
| Python | 3.8.18 | Momentum | 0.937 |
| PyTorch | 2.1.0 | Image size | 640*640 |
| RAM | 128GB | Number worker | 8 |

used to calculate classification loss, DFL and CIoU loss are mainly used to calculate regression loss.

$$L_{ce} = -\frac{1}{N}\sum_{i}\sum_{c=1}^{M} y_{ic}\log p_{ic} \tag{14}$$

$$L_{ciou} = 1 - IoU + \frac{d_o^2}{c_o^2} + \alpha\upsilon \tag{15}$$

where $L_{ce}$ is the cross-entropy loss, $L_{ciou}$ denotes the CIoU loss, $M$ indicates the number of categories. $y_{ic}$ equal to 1 if the true category of sample $i$ is equal to $c$, and 0 otherwise. $p_{ic}$ denotes the predicted probability that sample $i$ belongs to category $c$. $d_o$ is the distance between the center point of the ground truth and the prediction box, and $c_o$ is the diagonal distance from the smallest outer bounding rectangle.

## IV. EXPERIMENTS AND RESULTS

### A. Implementation Details

Our hardware and software facilities for training neural networks are shown in Table I. RDD2022 database was divided into the training set, validation set, and test set according to the ratio of 7:2:1. The initial learning rate was set to 0.01, and the parameters of the model were updated using Stochastic Gradient Descent (SGD) [39] with the weight decay coefficient set to 0.0005.

### B. Evaluation Metrics

To access the algorithm's accuracy, we employ standard metrics from the domain of target detection, such as precision ($P$), recall ($R$), F1, mean average precision ($mAP50$) and mean average precision over the range of 0.5 to 0.95 intersection over union ($mAP50-95$). We consider the number of model parameters, Floating Point Operations (FLOPs), and the inference time to evaluate the algorithm's inference speed. The formulas for the above evaluation metrics are shown below.

$$P = \frac{TP}{TP+FP} \tag{16}$$

$$R = \frac{TP}{TP+FN} \tag{17}$$

$$F1 = \frac{2*P*R}{P+R} \tag{18}$$

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{19}$$

$$AP50-95 = \frac{AP50+AP55+\ldots+AP95}{10} \tag{20}$$

$$mAP50 = \frac{\sum_1^n AP50_i}{n} \tag{21}$$

$$mAP50-95 = \frac{\sum_1^n [AP50-95]_i}{n} \tag{22}$$

where $TP$ represents the number of samples correctly predicted as positive categories, $FP$ indicates the number of samples incorrectly predicted as positive categories, and $FN$ signifies the number of samples wrongly classified as negative classes. $n$ denotes the total number of classes, and $mAP50$ refers to the mAP calculated at an IoU threshold of 0.5.

### C. Results

*1) Comparison of Accuracy:* These models were trained until convergence based on the aforementioned settings. Table II displays the detection results for each baseline model. Due to the difficulty of detecting the RDD2022 dataset, only a relatively small number of studies have been conducted on the entire dataset. RDD2022 consists of several national pavement damage sub-datasets, so many scholars focus only on road diseases in a single country. In this paper, to demonstrate the performance of FPDDN, all models were trained on the entire RDD 2022 dataset. Therefore, we did not compare it with the algorithms proposed by other scholars but chose the algorithms proposed in the past two years with high accuracy and fast detection speed.

The SSD [40] is an earlier proposed one-stage detection algorithm that exhibits the lowest detection accuracy of all models. Faster R-CNN [41] is a two-stage detection algorithm that exhibits a detection accuracy comparable to that of YOLO v8, which was proposed in 2023. Among the YOLO series models, the YOLO v5s proposed in 2020 has an accuracy closer to that of the YOLO v6s proposed in 2022. The YOLO v8 series of algorithms proposed in 2023 are superior to YOLO v6 and YOLO v5. This is because YOLO v8 integrates the latest and most effective tricks in computer vision. To better compare each model's accuracy, we plot each model's parameters in Table III. YOLO v8x is the most accurate model in the YOLO v8 series and has the largest number of parameters, so its accuracy is higher than the YOLO v8s and YOLO v8m models. A model's detection accuracy is commonly believed to increase with the number of parameters. YOLO v8x achieves higher accuracy by stacking modules, resulting in longer inference times.

According to Table II, the Swin Transformer based on self-attention performs worse than all other compared models. The Swin Transformer structure may require a large dataset to achieve good results. Although RDD2022 has more than 40,000 images, it is still smaller than the ImageNet [42] and MS COCO datasets [43], resulting in lower performance for the Swin Transformer.

Considering accuracy alone, the FPDDN model outperforms the YOLO v8x in terms of both mAP50 and mAP50-90, demonstrating superior detection capabilities for pavement damage compared to the latest models in the YOLO series.

The FPDDN achieved a mAP50 of 0.610, indicating its high accuracy in detecting cracks and potholes under a relatively

TABLE II
DETECTION RESULTS

| Model | P | R | mAP50 | | | | | mAP50-95 | | | | | F1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | D00 | D10 | D20 | D40 | All | D00 | D10 | D20 | D40 | All | |
| SSD | 0.581 | 0.482 | 0.513 | 0.520 | 0.583 | 0.407 | 0.506 | 0.242 | 0.251 | 0.302 | 0.145 | 0.235 | 0.527 |
| Faster RCNN | 0.587 | 0.553 | 0.577 | 0.589 | 0.656 | 0.465 | 0.572 | 0.303 | 0.296 | 0.350 | 0.201 | 0.289 | 0.569 |
| YOLO v5s | 0.611 | 0.501 | 0.534 | 0.544 | 0.647 | 0.444 | 0.542 | 0.283 | 0.262 | 0.334 | 0.186 | 0.266 | 0.551 |
| YOLO v6s | 0.613 | 0.513 | 0.548 | 0.567 | 0.648 | 0.439 | 0.551 | 0.298 | 0.284 | 0.344 | 0.184 | 0.278 | 0.559 |
| YOLO v8s | 0.629 | 0.532 | 0.559 | 0.579 | 0.663 | 0.477 | 0.570 | 0.301 | 0.287 | 0.348 | 0.206 | 0.286 | 0.576 |
| YOLO v8m | 0.643 | 0.552 | 0.576 | 0.597 | 0.684 | 0.496 | 0.588 | 0.310 | 0.301 | 0.356 | 0.218 | 0.296 | 0.594 |
| YOLO v8x | 0.634 | 0.564 | 0.586 | 0.609 | 0.682 | 0.514 | 0.598 | 0.320 | 0.309 | 0.363 | 0.226 | 0.304 | 0.597 |
| Swin Transformer | - | - | 0.523 | 0.523 | 0.597 | 0.508 | 0.526 | - | - | - | - | 0.237 | - |
| FPDDN | 0.639 | 0.568 | 0.601 | 0.622 | 0.697 | 0.522 | 0.610 | 0.334 | 0.322 | 0.380 | 0.241 | 0.319 | 0.601 |

TABLE III
NUMBER OF PARAMETERS AND INFERENCE TIME

| Model | Parms | FLOPs | Time |
|---|---|---|---|
| YOLO v5s | 9.1M | 23.8 | 1.7ms |
| YOLO v6s | 16.3M | 44.0 | 1.1ms |
| YOLO v8s | 11.1M | 28.4 | 1.2ms |
| YOLO v8m | 25.9M | 78.7 | 2.8ms |
| YOLO v8x | 68.2M | 257.4 | 7.1ms |
| Swin Transformer | 44.8M | 147 | 22.7ms |
| FPDDN | 14.7M | 45.7 | 1.8ms |

loose IoU threshold. The mAP50-95 criterion has higher requirements for the positioning of bounding boxes. Only with high classification and positioning capabilities under each threshold can high values be achieved in this criterion.

*2) Comparison of Inference Speed:* Table III lists the comparison between FPDDN and other models regarding parameters and inference speed. Parms denotes the model's total number of trainable parameters, including weights and biases. FLOPs refers to the computational workload, while Time represents the time required to detect an image. According to Table III, we can see that YOLO v5s has the smallest number of parameters and FLOPs, but the inference speed of YOLO v6s is the fastest. The YOLO v5s, predicated on using anchor boxes, produces a series of anchor boxes and then refines the detection boxes by discarding extraneous boxes via the non-maximum suppression algorithm. This process results in a significant increase in calculation. Being an anchor-free algorithm, YOLO v6s only needs to detect key points of the targets, thereby facilitating a swifter inference speed. In addition, YOLO v6s has a high degree of parallelization and can better utilize the GPU to accelerate inference images. This phenomenon can also be seen from the comparison between Swin Transformer and YOLO v8x. The Parms and FLOPs of the Swin Transformer are smaller than YOLO v8x, but its inference time is longer than YOLO v8x. This is because the computational complexity of the self-attention mechanism in the transformer is quadratically related to the sequence length. The division of an image into numerous patches results in a significant increase in computational complexity, which can have a substantial impact on the inference time, particularly for high-resolution images. The proposed FPDDN model in this study possesses Parms that exceeds that of YOLO v8s yet remains less than that of YOLO v6s, with an inference time of 1.8 milliseconds per frame. The inference speed of FPDDN is not the fastest among all models. This is because using the SFB module and the DAttention makes the model more complex, but they are critical to improving accuracy. During practical detection operations, the inference time of 1.8ms per frame is adequate for real-time detection of pavement detection.

*3) Comparison of Detection Results:* Fig.11 compares the detection results of the YOLO v8x and the FPDDN. We can find that the detection results of FPDDN are similar to YOLO v8x. We can find that some defects are not labeled but still detected by FPDDN, which can prove that FPDDN has better detection performance. To further validate the robustness of FPDDN, we added detection images in environments such as heavy snow, moist pavement, bright light, and shadows, which can be seen in Fig.12. We can learn that even though shadows or moisture may cause the model to misclassify, FPDDN can still detect most of the defects that can be directly observed by the human.

Considering the accuracy and speed of the model, FPDDN can be considered the best model among those compared. It has only one-fifth of the Parms of YOLO v8x, yet it achieves better detection accuracy and FPS.

### *D. Ablation Study*

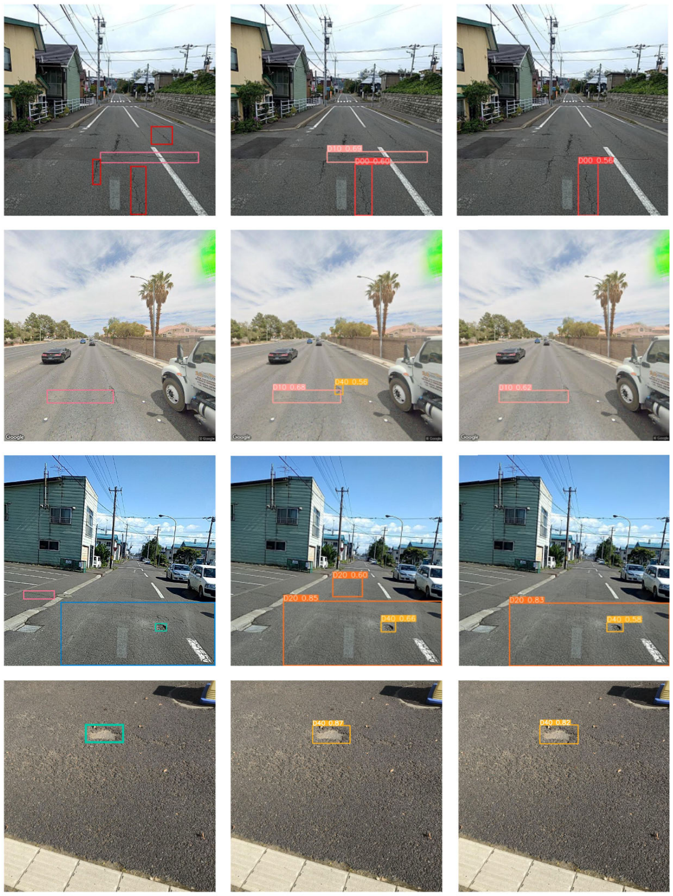
To validate the efficacy of each module, ablation studies were performed, with the outcomes presented in Table IV. FPDDN denotes the utilization of the comprehensive model proposed in this study. FPDDN+C2f denotes the baseline model without using the proposed module of this study.

*1) D2f Module:* As can be seen from Table IV, the accuracy of D2f is similar to that of C2f, which shows that both have the same ability to extract road disease features. However, model inference using the D2f module is faster, so this study proposes the D2f module.

*2) SFB Module:* It can be found from Table IV that when the SFB and DAttention are not used, the model has a low number of parameters and a fast inference speed, but its accuracy is low, even inferior to the YOLO v5s model. Incorporating the SFB module significantly enhanced the model's performance, most notably in the detection accuracy of potholes (D40). This is because the SFB module can help the model improve the detection accuracy of small targets. Most of the potholes in RDD2022 are small targets. In the field

TABLE IV
ABLATION EXPERIMENT

| Model | Parms | FPS | mAP50 | | | | | mAP50-95 | | | | | F1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | D00 | D10 | D20 | D40 | All | D00 | D10 | D20 | D40 | All | |
| FPDDN+C2f | 5.6M | 0.9ms | 0.530 | 0.535 | 0.621 | 0.434 | 0.530 | 0.274 | 0.256 | 0.330 | 0.181 | 0.260 | 0.545 |
| FPDDN+D2f | 4.7M | 0.8ms | 0.528 | 0.535 | 0.622 | 0.430 | 0.529 | 0.270 | 0.255 | 0.334 | 0.178 | 0.259 | 0.544 |
| FPDDN+SFB | 9.9M | 1.3ms | 0.567 | 0.588 | 0.674 | 0.499 | 0.582 | 0.309 | 0.294 | 0.355 | 0.215 | 0.293 | 0.579 |
| FPDDN+DAttention | 10.4M | 1.5ms | 0.578 | 0.611 | 0.675 | 0.478 | 0.586 | 0.321 | 0.309 | 0.358 | 0.199 | 0.297 | 0.583 |
| FPDDN | 14.7M | 1.8ms | 0.601 | 0.622 | 0.697 | 0.522 | 0.610 | 0.334 | 0.322 | 0.380 | 0.241 | 0.319 | 0.601 |



Ground Truth FPDDN YOLO v8x

Fig. 11. Detection results of FPDDN and YOLO V8X.

of object detection, shallow feature maps have rich location information and also contain a large number of small target features. However, downsampling gradually leads to the loss of these small target features. Upsampling deep feature maps does not recover features of small targets. The SFB module aims to improve small target detection accuracy by subtracting the deep feature map from the shallow feature map in one branch and splicing them in the other.

*3) DAttention Module:* Furthermore, using DAttention significantly enhances the mAP50 and mAP50-95 of both transverse (D10) and longitudinal cracks (D00). Table IV demonstrates that DAttention elevates the mAP50 of transverse cracks from 0.53 to 0.578 and increases the mAP50-95 from 0.274 to 0.321. The improvement is also substantial for longitudinal cracks.

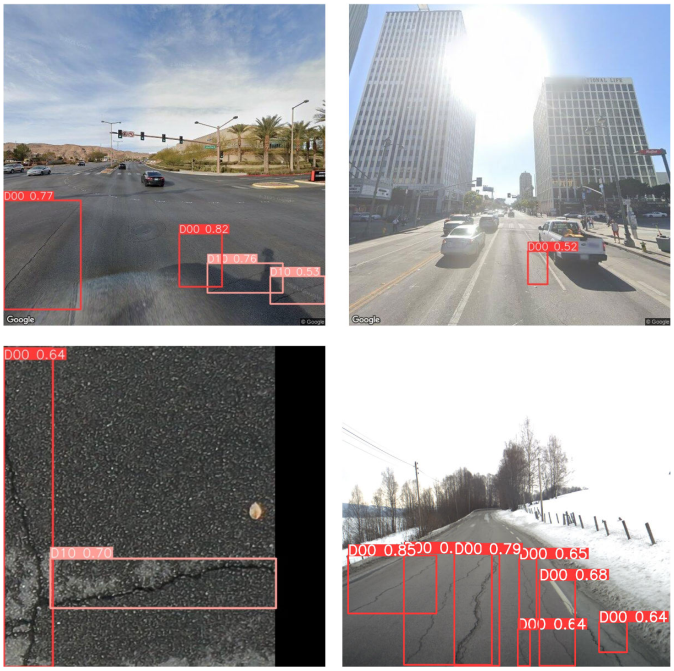


Fig. 12. Detection results in complex environments.

Transverse and longitudinal cracks have extremely irregular shapes and complex textures. Traditional convolution kernels cannot flexibly capture the irregular model of cracks, while the DAttention can adapt the self-attention mechanism to the geometrical shape of cracks by learning position offsets. Therefore, it has the most significant improvement in D00 and D10. In addition, the Transformer's self-attention mechanism allows the model to capture global information in a single layer, while CNN must gradually increase the receptive field through multiple convolution layers and pooling layers. Cracks are typically striped targets and a single crack may span the entire height or width of the image. This can create difficulties for CNN in extracting global features of the cracks. DAttention can extract the global information of cracks in a single layer, so the accuracy of D00 and D10 can be significantly improved.

*4) FPDDN:* Table IV shows a significant improvement in model accuracy when using the complete FPDDN network. The SFB module can improve the detection accuracy of small targets, such as potholes. Additionally, the DAttention is conducive to detecting irregular targets, significantly improving the accuracy of transverse and longitudinal cracks. The enhancements in accuracy for the detection of alligator cracks provided by the two modules are comparable; however, their integration elevates the mAP50 for alligator cracks from 0.621 to 0.697, while the mAP50-95 experiences an increase from 0.33 to 0.38. The observed improvements are chiefly due to the FPDDN's backbone network, which excels at extracting pertinent features of pavement damage. Feature maps derived from disparate locations are introduced into the SFB module, which not only emphasizes features of small targets but also fosters the synthesis of deep and shallow feature maps, culminating in a comprehensive fusion. Moreover, the self-attention mechanism within the DAttention captures the global features of the damage from a singular layer, obviating the necessity for additional convolution modules and consequently diminishing the model's parameters.

In conclusion, integrating the SFB and DAttention modules can significantly enhance the accuracy of the FPDDN network. Consequently, FPDDN demonstrates more robust pavement damage detection capabilities with a smaller parameter set than YOLO v8x.

## V. Conclusion

This study proposes the FPDDN model for real-time and high-accuracy pavement damage detection. This model mainly consists of convolution modules, deformable Transformer modules, SFB modules, D2f modules, and multiple detection branches. Depthwise separable convolutions are also used to reduce model parameters further. The SFB module in FPDDN mainly improves small target detection accuracy by reducing the loss of small target features, which can significantly improve the model's ability to detect potholes. The DAttention mainly adapts the self-attention mechanism to the geometry of irregular cracks by learning position offsets, thereby improving the ability to detect transverse and longitudinal cracks. By comparing with advanced models such as YOLO v5, YOLO v6, YOLO v8 and Swin Transformer, the F1 score of FPDDN is 0.601 and the mAP50 is 0.610, which are both larger than the current advanced object detection model.

YOLO v8x is currently a representative algorithm in the field of object detection with advanced performance in many datasets. However, our proposed FPDDN algorithm not only has higher detection accuracy than YOLO v8x, but also achieves four times faster detection speed than it. In addition, the detection results in complex environments such as bright light, moist pavement, and shaded environments further demonstrate the robustness and versatility of FPDDN in real-world detection environments.

Although the FPDDN model has a good balance between pavement damage detection accuracy and inference speed, it still has certain shortcomings. First, FPDDN cannot quantify the area, length, and other geometric characteristics of pavement damage and cannot quantitatively evaluate the quality of road sections. Additionally, damage detection accuracy is low in complex environments, such as those with shade or moisture. The above problems can be solved in the future by establishing a pavement damage database in complex environments and applying computer vision measurement technology to FPDDN.

## References

[1] Z. Zhang, F. Liu, Y. Huang, and Y. Hou, "Detection and statistics system of pavement distresses based on street view videos," *IEEE Trans. Intell. Transp. Syst.*, early access, pp. 1–10, May 2024.

[2] Y. Huang, Y. Liu, F. Liu, and W. Liu, "A lightweight feature attention fusion network for pavement crack segmentation," *Comput.-Aided Civil Infrastruct. Eng.*, early access, pp. 1–15, May 2024.

[3] A. Fox, B. V. K. V. Kumar, J. Chen, and F. Bai, "Multi-lane pothole detection from crowdsourced undersampled vehicle sensor data," *IEEE Trans. Mobile Comput.*, vol. 16, no. 12, pp. 3417–3430, Dec. 2017.

[4] Y. Zhang et al., "Road surface defects detection based on IMU sensor," *IEEE Sensors J.*, vol. 22, no. 3, pp. 2711–2721, Feb. 2022.

[5] R. Bajwa, E. Coleri, R. Rajagopal, P. Varaiya, and C. Flores, "Pavement performance assessment using a cost-effective wireless accelerometer system," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 9, pp. 1009–1022, Sep. 2020.

[6] Y. Wang, B. Yu, X. Zhang, and J. Liang, "Automatic extraction and evaluation of pavement three-dimensional surface texture using laser scanning technology," *Autom. Construct.*, vol. 141, Sep. 2022, Art. no. 104410.

[7] M. R. De Blasiis, A. D. Benedetto, and M. Fiani, "Mobile laser scanning data for the evaluation of pavement surface distress," *Remote Sens.*, vol. 12, no. 6, p. 942, Mar. 2020.

[8] D. Zhang et al., "Automatic pavement defect detection using 3D laser profiling technology," *Autom. Construct.*, vol. 96, pp. 350–365, Dec. 2018.

[9] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 12, pp. 1127–1141, Dec. 2018.

[10] Y. Zhang et al., "Road damage detection using UAV images based on multi-level attention mechanism," *Autom. Construction*, vol. 144, Dec. 2022, Art. no. 104613.

[11] H. Dong, K. Song, Y. Wang, Y. Yan, and P. Jiang, "Automatic inspection and evaluation system for pavement distress," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12377–12387, Aug. 2022.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–17.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[14] N. Wang, Q. Zhao, S. Li, X. Zhao, and P. Zhao, "Damage classification for masonry historic structures using convolutional neural networks based on still images," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 12, pp. 1073–1089, Dec. 2018.

[15] D. Arya et AL., "Global road damage detection: State-of-the-art solutions," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2020, pp. 5533–5539, doi: 10.1109/BIGDATA50022.2020.9377790.

[16] V. Pham, C. Pham, and T. Dang, "Road damage detection and classification with Detectron2 and faster R-CNN," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2020, pp. 5592–5601.

[17] K. Doshi and Y. Yilmaz, "Road damage detection using deep ensemble learning," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2020, pp. 5540–5544.

[18] V. Hegde, D. Trivedi, A. Alfarrajeh, A. Deepak, S. Ho Kim, and C. Shahabi, "Yet another deep learning approach for road damage detection using ensemble learning," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2020, pp. 5553–5558.

[19] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[20] O. Iparraguirre, N. Iturbe-Olleta, A. Brazalez, and D. Borro, "Road marking damage detection based on deep learning for infrastructure evaluation in emerging autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22378–22385, Nov. 2022.

[21] Y. Zhang and C. Liu, "Network for robust and high-accuracy pavement crack segmentation," *Autom. Construct.*, vol. 162, Jun. 2024, Art. no. 105375.

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

[23] Y. Zhang, J. Wu, Q. Li, X. Zhao, and M. Tan, "Beyond crack: Fine-grained pavement defect segmentation using three-stream neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14820–14832, Sep. 2022.

[24] T. Zhang, D. Wang, and Y. Lu, "ECSNet: An accelerated real-time image segmentation CNN architecture for pavement crack detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15105–15112, Dec. 2023.

[25] Z. Zhao et al., "MCANet: Hierarchical cross-fusion lightweight transformer based on multi-ConvHead attention for object detection," *Image Vis. Comput.*, vol. 136, Aug. 2023, Art. no. 104715.

[26] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[27] Y. Liu, F. Liu, W. Liu, and Y. Huang, "Pavement distress detection using street view images captured via action camera," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 738–747, Jan. 2024.

[28] (2022). *Ultralytics/YOLOv5: V7.0—YOLOv5 SOTA Realtime Instance Segmentation*. Accessed: May 7, 2023. [Online]. Available: https://github.com/ultralytics/yolov5.com

[29] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[30] G. Jocher, A. Chaurasia, and J. Qiu. (2023). *Ultralytics YOLOv8*. [Online]. Available: https://github.com/ultralytics/ultralytics

[31] D. Arya et al., "Transfer learning-based road damage detection for multiple countries," 2020, *arXiv:2008.13101*.

[32] X. He, Z. Tang, Y. Deng, G. Zhou, Y. Wang, and L. Li, "UAV-based road crack object-detection algorithm," *Autom. Construct.*, vol. 154, Oct. 2023, Art. no. 105014.

[33] 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.

[34] 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.

[35] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2022: A multi-national image dataset for automatic road damage detection," 2022, *arXiv:2209.08538*.

[36] T. Y. Lin, P. Dollàr, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.

[37] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[38] X. Li et al., "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 21002–21012.

[39] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.

[40] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37.

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

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2009, pp. 248–255.

[43] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf.*, 2015, pp. 740–755.

**Yingchao Zhang** received the B.S. and M.S. degrees in transportation engineering from Shandong University, Jinan, China, in 2020 and 2023, respectively. He is currently pursuing the Ph.D. degree with the City University of Hong Kong. His current research interests include computer vision and nondestructive testing.

**Cheng Liu** received the B.S. degree in aerospace engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013, and the M.S. degree in aeronautics and astronautics and the Ph.D. degree in mechanical engineering from Stanford University, Stanford, CA, USA, in 2015 and 2021, respectively. He is currently an Assistant Professor with the Department of Systems Engineering, City University of Hong Kong. His current research interests include physics-guided machine learning for structural health monitoring (SHM), smart structures, cyber-physical systems/digital twin, robotic tactile sensing, and mechanics of composite structures.