DOI: 10.1111/mice.12962

RESEARCH ARTICLE

COMPUTER-AIDED CIV. AND INFRASTRUCTURE ENGINEERING WILEY

A grid-based classification and box-based detection fusion model for asphalt pavement crack

Bao-Luo Li[1] | **Yu Qi**[1] | **Jian-Sheng Fan**[1] | **Yu-Fei Liu**[1] | **Cheng Liu**[2]

[1]Key Laboratory of Civil Engineering Safety and Durability of China Education Ministry, Department of Civil Engineering, Tsinghua University, Beijing, China

[2]Research Institute of Highway Ministry of Transport, Beijing, China

Correspondence
Yu-Fei Liu, Key Laboratory of Civil Engineering Safety and Durability of China Education Ministry, Department of Civil Engineering, Tsinghua University, Beijing, China.
Email: liuyufei@tsinghua.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 52192662, 51978376

Abstract

Crack identification is essential for the preventive maintenance of asphalt pavement. Through periodic inspection, the characteristics of existing and emerging cracks can be obtained, and the pavement condition index can be calculated to assess pavement health. The most common method to estimate the area of cracks in an image is to count the number of grid cells or boxes that cover the cracks in an image. Accurate and efficient crack identification is the premise of crack assessment. However, the current patch-based classification method is limited by the receptive field and cannot be used to directly classify cracks. Furthermore, the postprocessing algorithm in anchor-based detection is not explicitly optimized for crack topology. In this paper, a new model, which is the fusion of grid-based classification and box-based detection based on You Only Look Once version 5 (YOLO v5) is developed and described for the first time. The accuracy and efficiency of the model are high partly due to the implementation of a shared backbone network, multi-task learning, and joint training. The non-maximum suppression (NMS)–area-reduction suppression (ARS) algorithm is presented to filter redundant, overlapping prediction boxes in the postprocessing stage for the crack topology, and the mapping and matching algorithm is proposed to combine the advantages of both the grid-based and box-based models. A double-labeled dataset containing tens of thousands of asphalt pavement images is assembled, and the proposed method is verified on the test set. The fusion model has superior performance over the individual classification and detection models, and the proposed NMS-ARS algorithm further improves the detection accuracy. Experimental results demonstrate that the presented method effectively realizes automatic crack identification for asphalt pavement.

1 | INTRODUCTION

Pavement crack identification is a critical component of road maintenance. Roads paved over the previous decades have gradually deteriorated, and pavement distress accumulated over a long time now threatens vehicle safety and reduces driving comfort (Zaloshnja & Miller, 2009). The most common type of pavement distress is cracking (W. Wang et al., 2019). The pavement crack should be repaired as soon as possible. Otherwise, the pavement will deteriorate rapidly, resulting in shortened service life and increased maintenance costs (Adlinge & Gupta, 2013).

Therefore, periodic inspection and regular evaluation are necessary for maintaining pavement health (Koch et al., 2015).

Currently, early pavement crack identification is primarily accomplished through visual inspection. In the meantime, digital image processing (DIP) has been broadly researched and progressively applied with the advancement of imaging technology. DIP includes preprocessing and segmentation. Scholars have developed preprocessing techniques such as image enhancement (Sun et al., 2009) and image denoising (Zhou et al., 2006) to mitigate the effects of non-uniform illumination (Koutsopoulos & Downey, 1993), tire marks (Zakeri et al., 2017), and other disturbances. Researchers have also developed a wide variety of image segmentation techniques to extract the shape of cracks. The threshold method is the most commonly used method, and it operates by dividing the image into several different classes based on one or more gray threshold values (Otsu, 1979). Other widely deployed methods include the edge algorithm, which uses differential operators for gradient calculation (Abdel-Qader et al., 2003); the region algorithm for connectivity of regions and integrity of cracks (Huang & Tsai, 2011); and the matching algorithm, which uses filters to match linear patterns features (A. Zhang et al., 2013). Visual inspection for pavement defects can be labor-intensive and time-consuming. On the other hand, DIP overcomes such shortcomings through computer-aided feature identification and thus offers improvements in accuracy and savings in terms of man-hours. However, DIP requires hand-crafted features for specific scenarios, which can entail a heavy workload and the result generalizes poorly.

In the decade since AlexNet was proposed (Krizhevsky et al., 2017), the continuous improvement of computer hardware and the explosive popularity of artificial intelligence technology has generated much attention to the application of deep learning, especially convolutional neural networks (CNNs) toward pavement crack identification. Deep learning employs hierarchical networks to automatically extract discriminative features with decent generalization. Numerous studies have demonstrated that deep learning outperforms traditional methods in classification (Hoang et al., 2018), object detection (Basavaraju et al., 2019), segmentation (Dorafshan et al., 2018), and other tasks (Adeli, 2020; Rafiei et al., 2017).

The classification task in pavement crack identification involves the division of the image into patches and performing classification at each patch. The first application of deep learning in pavement crack identification is a supervised deep CNN for classifying image patches (L. Zhang et al., 2016). However, the different types of cracks are not classified by this method. Subsequently, principal component analysis was combined with CNN to classify the different types of cracks, and the different patch scale was discussed (X. Wang & Hu, 2017). In order to distinguish cracks from irrelevant pavement disturbances, a deep CNN-based unified approach incorporating transfer learning-based preclassification was proposed (K. Zhang, Cheng, & Zhang, 2018). Presently, various patch-based CNNs, such as the high-accuracy CNN (B. Li et al., 2020) and the adaptive lightweight CNN (Hou et al., 2021), have been developed to meet different requirements for pavement crack classification tasks.

Object detection involves the dual objective of object classification and localization. Most research on pavement crack detection directly borrows advanced models open-sourced by pioneers in computer vision, for example, single ahot multibox detector (SSD) (Maeda et al., 2018), YOLO) v1∼v5 (Du et al., 2021; Jeong, 2020; Mandal et al., 2018; Nie & Wang, 2019; Peraka et al., 2021), and faster regoin - convolutional neural network (Faster R-CNN) (Ibragimov et al., 2020; J. Q. Li et al., 2019). A number of researchers have improved upon and adapted such classical models for pavement crack detection. For example, in one model, a sensitivity detection stage is added to extract the sensitive part for cracks before the feature refinement stage of Faster R-CNN (Huyan et al., 2019). In more recent work, researchers have suggested that the pavement condition can be assessed more effectively by classifying cracks based on severity (V. P. Tran et al., 2021).

Pavement crack identification, instead of detecting individual instances of cracks in an image, involves pixel-level classification (i.e., semantic segmentation). Semantic segmentation has emerged as a hot topic in pavement crack identification in recent years (Chen et al., 2019; Guan et al., 2021; T. S. Tran et al., 2022; Zhang et al., 2018). However, preparing a dataset for segmentation is more complicated and time-consuming than object detection. In order to improve efficiency and accuracy, a two-step pavement crack detection and segmentation method is proposed (J. W. Liu et al., 2020). Additionally, detection and segmentation networks are fused to gain more information efficiently (Feng et al., 2020; C. Liu et al., 2021).

However, the above methods still have the following shortcomings. (1) The classical non-maximum suppression (NMS; Bodla et al., 2017) in object detection does not consider the topology of the detected crack, which is almost always different from that of other common objects. Other common objects in object detection tasks, such as faces and cars, are regions with a certain area, which is suitable for the bounding box to locate. Cracks can be difficult due to their simple appearance as an assembly of connected lines. Thus, while a large, branched crack may identify as one crack, its branches can also be considered individual cracks. As a result, multiple bounding boxes may overlap each other for the same object. To the best of the authors' knowledge, no studies have

specifically optimized this issue. (2) The receptive field of the patch is limited, leading to two main problems: First, local information cannot reflect the overall characteristics and crack type; second, focusing on the local area can lead to disturbances from pavement noise. Additional measures are needed to address these problems. (3) Patch-based classification adopts the sliding window method with a stride half of the window scale to avoid cracks located at the edge (Park et al., 2019), which at least doubles the computational load. (4) The current segmentation performance is insufficient to support practical applications. Segmentation is more complex than other tasks, making it harder to optimize. Besides, obtaining a large segmentation dataset is difficult, and thus most of the current studies are based on small sample sets (Bang et al., 2019; Yang et al., 2019; A. Zhang et al., 2017), which results in limited generalization even after data augmentation (Zou et al., 2018).

Furthermore, while most of the current research focuses on segmentation, classification and detection are often sufficient for pavement condition assessment. Cracking not only affects local pavement performance but also has a considerable influence on the pavement structure performance in the nearby area (L. Li & Jiang, 2021). This means that simply analyzing pixels overestimates the pavement health. At present, relevant assessment standards do not adopt pixel-based methods. Most specifications (e.g., JTG 5210-2018, JTG-T E61-2014) employ the pavement condition index (PCI), which requires a calculation of the distressed area, to assess pavement condition. The most common way to calculate the area is to divide the image based on a standardized grid and count the number of square cells containing cracks. In addition to the number of sections containing cracks, a bounding box, which is a rectangle that closely encapsulates the crack, can also be used to calculate the area (Ministry of Transport of the People's Republic of China, 2018). These two methods correspond to grid-based classification and box-based detection (GCBD). Considering that the accuracy, efficiency, cost, and implementation of classification and detection are currently superior to segmentation (Hu et al., 2021; Ronneberger et al., 2015; X. Wang & Hu, 2017), this research focuses on classification and detection.

The primary objective of this paper is the development of a GCBD fusion model. The fusion model outputs both grid and box results. The accuracy and efficiency of the model are improved through a shared backbone network, multi-task learning, and joint training. In this paper, a grid-based classification method is presented first, which can classify all grid cells on one image in one step. Second, a box-based detection method based on modified YOLO v5 is developed. An improved NMS method that considers crack topology in the postprocessing of the box-based method is proposed. Third, a fusion model that integrates GCBD is constructed to output both results simultaneously. Additionally, a large-scale double-labeled dataset for pavement crack identification covering various scenarios is assembled. Finally, the experimental results are discussed, and the model performance is analyzed.

2 | FRAMEWORK

The proposed methodology contains four parts: dataset construction, training, preprocessing, and postprocessing. The framework is presented in Figure 1.

The dataset consists of numerous asphalt pavement images labeled by grid cells and detection boxes. Data are divided into a training set, a validation set, and a test set. In the next step, a neural network architecture that integrates GCBD is introduced to identify asphalt pavement cracks. A combined loss function is adopted for joint training. The preprocessing step accelerates convergence and improves generalization. Finally, in the postprocessing stage, an improved NMS method filters redundant detection boxes. A mapping and matching (MaM) method classifies the type of crack captured in each grid cell by combining the outputs of the box and grid neural network modules.

The major contributions and highlights of this paper are as follows. First, the double-labeled dataset constructed for asphalt pavement crack identification in this research is so far the largest by a wide margin. The dataset covers diverse scenarios and guarantees generalization. Second, a fusion model is proposed to output both grid and box results. Research on multi-task learning of classification and detection is lacking, and the results of this paper bridge the gap. Meanwhile, the fusion network output conforms to the current road condition index. Third, grid-based classification performed by the fusion model avoids the high computation demand of the sliding window method and solves the limited receptive field. The method classifies all grids on one image through a single calculation. Fourth, classical NMS ignores crack topology, and thus NMS-area-reduction suppression (NMS-ARS) is proposed to filter redundant detection boxes. Fifth, the noise present in each grid cell is filtered, and the type of crack is classified through the MaM algorithm. The above contributions come together to enable automatic identification of asphalt pavement cracks.

3 | METHODOLOGY

3.1 | Dataset

A large-scale dataset is constructed for asphalt pavement crack identification. Numerous images acquired from various inspection vehicles were collected through multiple pavement inspection institutions. The image shape is 2048 × 2048 pixels, with about 1 mm ground sample

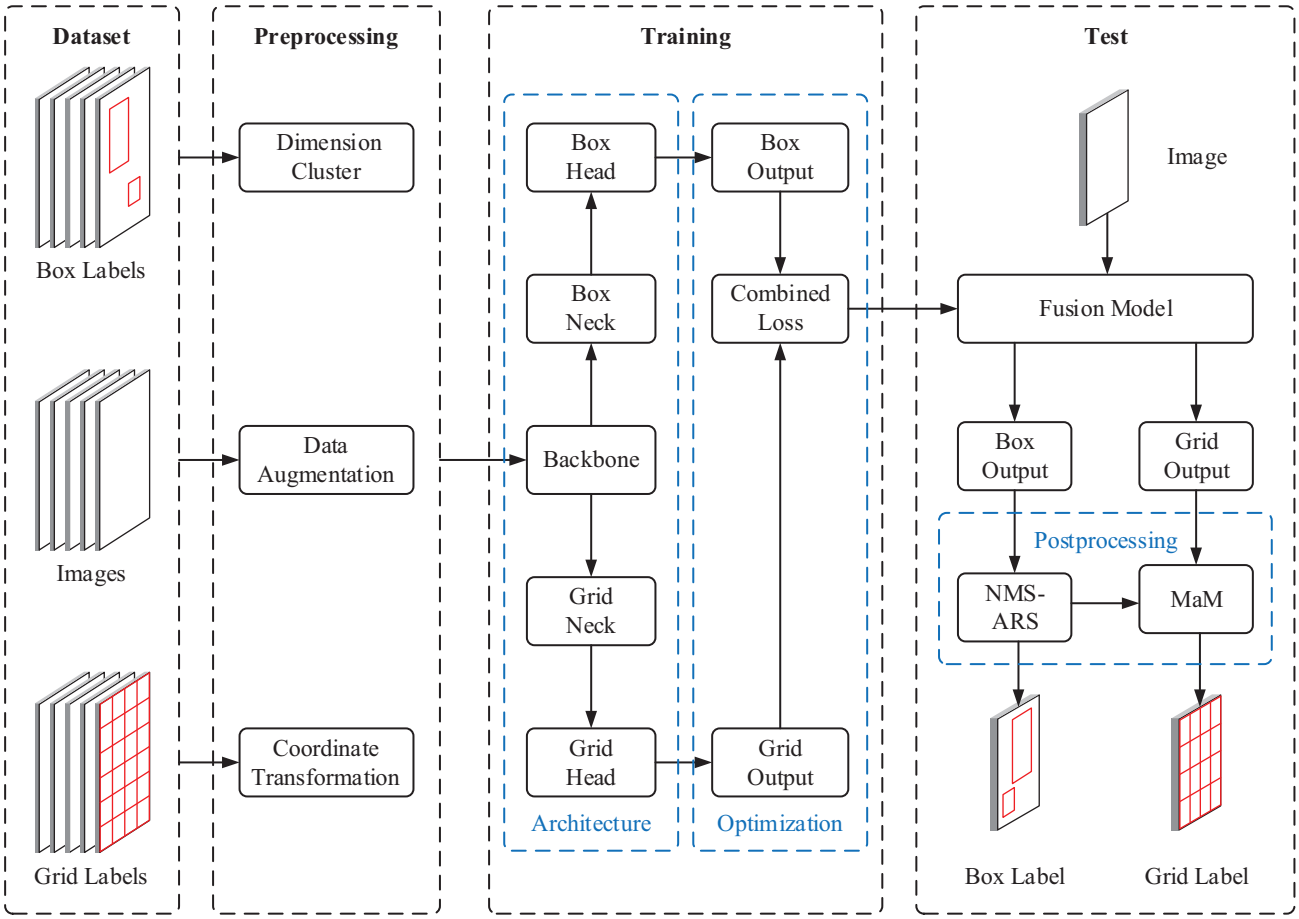


FIGURE 1 Methodology framework. ARS, area-reduction suppression; MaM, mapping and matching; NMS, non-maximum suppression

TABLE 1 Data annotation method

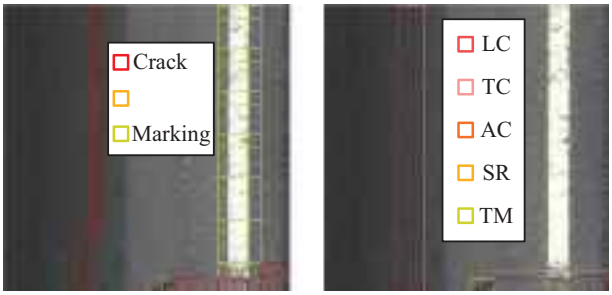
Method	Task object	Data category	Annotation principle
Grid-based	Classification	Crack, repair, marking	1. One category per grid 2. Crack first and marking last
Box-based	Detection	LC, TC, AC, SR, TM	1. Separate targets by connectivity 2. Minimum bounding box for positioning

Abbreviations: AC, alligator crack; LC, longitudinal crack; SR, strip repair; TC, transverse crack; TM, traffic marking.

distance in the top-down camera view. The images were acquired from several Chinese provinces and cities (e.g., Beijing, Hunan, Guangdong, Xinjiang) and from multiple complicated scenarios (e.g., leaves, sand, water, shadows). The massive dataset ensures data richness.

All images in the dataset are annotated using the dual approach of the proposed fusion model (as shown in Table 1). The grid-based method divides the image into 20 × 20 cells and classifies each cell. Simultaneously, the box-based method draws bounding boxes to locate and classify targets (as shown in Figure 2). The annotation follows three common and important categories. According to the crack morphology, cracks are divided into longitudinal cracks (LC), transverse cracks (TC), and alligator cracks (AC) in the box-based method. The presence of strip repairs (SR) is a factor to consider in pavement condition assessment (Ministry of Transport of the People's Republic of China, 2018). Since some SR can appear like cracks, SRs are divided into a separate category. Moreover, surface textures similar to cracks appear in traffic marking (TM) under local climatic conditions and other factors (Taheri et al., 2017). In order to distinguish the texture from the crack, TM is specially divided into its own category. Overall, 20,000 images are double-labeled to build the dataset.

The 20,000 images are randomly divided into the training set, validation set, and test set following a 6:2:2 ratio. In other words, there are 12,000 images for training, 4,000



(a) Grid-based method (b) Box-based method

FIGURE 2 Annotation examples. AC, alligator crack; LC, longitudinal crack; SR, strip repair; TC, transverse crack; TM, traffic marking

images for validation, and 4,000 images for testing. The population of each set are shown in Figure 3. The number of positive samples is almost balanced in both methods. In the grid-based method, the number of positive and negative samples are 1,098,181 and 6,901,819, respectively, and the ratio is about 1:6.3. Therefore, the sample imbalance is not severe.

3.2 | Training

3.2.1 | Architecture

The proposed model is named GCBD. GCBD is based on YOLO v5 and consists of five parts: backbone, box neck, box head, grid neck, and grid head (Figure 4). Compared to the base YOLO v5 architecture, the proposed model contains a grid neck and grid head as new additions to achieve the objectives described in this paper. The network is a deep CNN, in which the basic block is composed of a convolution layer, a batch normalization layer, and a sigmoid linear unit (SiLU) activation layer. For conciseness, this basic block is referred to as Convolution, Batch normalization, SiLU (CBS). Referring to the individual blocks in Figure 4, the default stride of the convolution kernel is 1. If the first row is marked with "/2," then the stride is 2. The second row shows the dimensions of the convolution kernel in order of height, width, and channel. If only the first two dimensions are displayed, then the output channel is the same size as the input channel. If the second row is marked with "/2," then the output channel is half the size of the input channel.

The cross stage partial darknet (CSPDarknet) architecture is implemented in the backbone for feature extraction (Bochkovskiy et al., 2020; C. Y. Wang et al., 2020). The backbone consists of CBS blocks and C3T blocks. The CBS blocks sample the image five times, and a feature map with a shape of 20×20 is received at the end of the backbone. The C3T block, composed of three convolutional blocks and one BottleneckT_N block, integrates the feature map and consumes less computational resources. The BottleneckT_N block in C3T consists of N residual units implemented with double-layer skips. The training speed and effectiveness are significantly improved through skip connections of residual units.

The neck enables feature fusion. In the grid neck, a bottom-up feature pyramid aggregates the information from the backbone. Meanwhile, spatial pyramid pooling (SPP) and path aggregation network (PAN) are employed in the box grid. Spatial pyramid pooling - fast (SPPF), an implementation technique for SPP, integrates multiscale features through a spatial pyramid pooling layer (He et al., 2015). PAN boosts information flow and enhances feature hierarchy by bottom-up path augmentation (S. Liu et al., 2018) after feature pyramid network (FPN) (Lin, Dollar, et al., 2017). C3F is a fundamental component in the neck composed of three convolution blocks and one BottleneckF_N block. The difference between BottleneckF_N and BottleneckT_N is that BottleneckF_N has no skip connection.

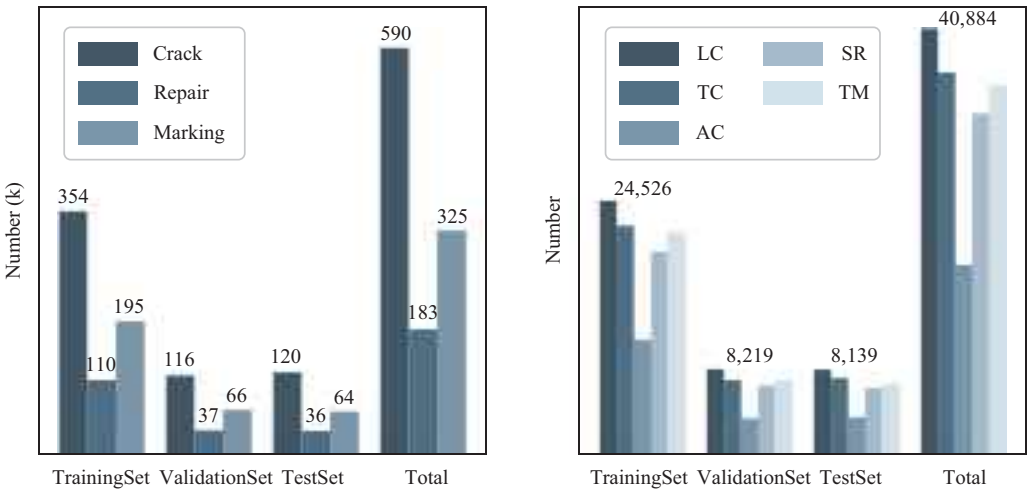
The head predicts the output. The grid head classifies each grid cell on the 20×20 feature map. The box head regresses the classes and location of the target through the anchor boxes at three scales. In order to balance the positive and negative samples, the two nearest cells of the target are also regarded as positive samples.

3.2.2 | Optimization

The loss function consists of the losses obtained from the grid cells and the box. The grid cell loss is BCEWithLogitsLoss, which combines sigmoid and binary cross-entropy loss. The focal loss is applied to balance positive and negative samples, and the gamma is 1.5 (Lin, Goyal, et al., 2017). The box loss consists of localization loss, confidence loss, and classification loss as described in Equation (1). Localization loss uses $CIoU$ loss for localization (Zheng et al., 2020). Meanwhile, confidence loss and classification loss use BCEWithLogitsLoss to distinguish objects from the background and classify them, respectively (Zhao et al., 2021).

$$\mathcal{L}_{box} = \mathcal{L}_{loc} + \mathcal{L}_{obj} + \mathcal{L}_{cls} \tag{1}$$

The fusion model adopts transfer learning and multitask learning. The detection network is pretrained on the common objects in context (COCO) dataset (Lin et al., 2014). The classification network and detection network are jointly trained with a mini-batch size of 64 for 100 epochs. The SGD optimizer started with a learning rate of



(a) Grid number (b) Box number

FIGURE 3 Breakdown of data from the grid-based classification sample set and box-based detection sample set

0.01 and a momentum of 0.9. Parameters are divided into three groups: bias, batch normalization weight, and convolution weight, where bias and convolution weights adopt a 0.0005 weight decay. The learning rate schedules utilize warm-up and cosine decay (Goyal et al., 2017; Loshchilov & Hutter, 2016) as shown in Figure 5. The bias learning rate reduced from 0.1 to 0.01, and all other learning rates rose from 0.0 to 0.01 linearly in three warm-up epochs. Meanwhile, the momentum rose from 0.8 to 0.9 linearly. The final learning rate decayed to 0.001 via cosine decay. The distribution coefficient (DC) of the combined grid cell and box loss function is 0.2 for joint training as shown in Equation (2).

$$\mathcal{L} = \alpha \mathcal{L}_{grid} + (1 - \alpha)\, \mathcal{L}_{box} \tag{2}$$

3.3 | Preprocessing

In order to accelerate convergence and improve generalization, preprocessing techniques are adopted. Since the grid cell label is marked according to the rows and columns of the grid, a coordinate transformation is performed so that the grid cell labels can be registered in the same coordinate system as the box label. Then, the image is resized to 640 × 640, and the pixel value is normalized between 0 and 1. Finally, image enhancement and adaptive anchors are applied as described in the following section.

3.3.1 | Data augmentation

The data loader generates mini-batch samples through online augmentation. Compared with offline augmentation, online augmentation does not excessively increase the size of datasets and is more suitable for larger datasets.

All images are geometrically transformed according to a probability. In consideration of maintaining the directionality of LC and TC, as well as the cell size, the image is only flipped horizontally and vertically with a probability of 50%. Translation, scaling, and other geometric transformations are not applied.

Color jittering is performed by adding disturbance to the H, S, and V channels in hue, saturation, and value (HSV) color space. The hue of the pavement crack image is relatively single, while the saturation and value varied significantly with environmental factors, such as illumination. Therefore, the disturbance coefficient for hue is small (0.015) but is large for saturation and value (0.7 and 0.4, respectively).

Mosaic augmentation is applied to enrich context by mixing four images as shown in Figure 6. Mixing different contexts greatly improves generalizability and robustness. This approach also reduces the difficulty of training.

3.3.2 | Dimension cluster

The proposed network is anchor-based, in which nine anchor boxes are evenly distributed to the detection head according to the receptive field. Anchor boxes affect the scale and quality of prediction boxes. Therefore, K-means clustering and genetic algorithms were employed to generate adaptive anchors (Redmon & Farhadi, 2017).

After 1000 generations to evolve the anchors, the sizes of the anchors are (53, 65), (206, 48), (49, 228), (52, 621), (496, 73), (80, 576), (555, 141), (201, 543), and (405, 577) as shown in Figure 7. The best possible recall is 99.3%, which means

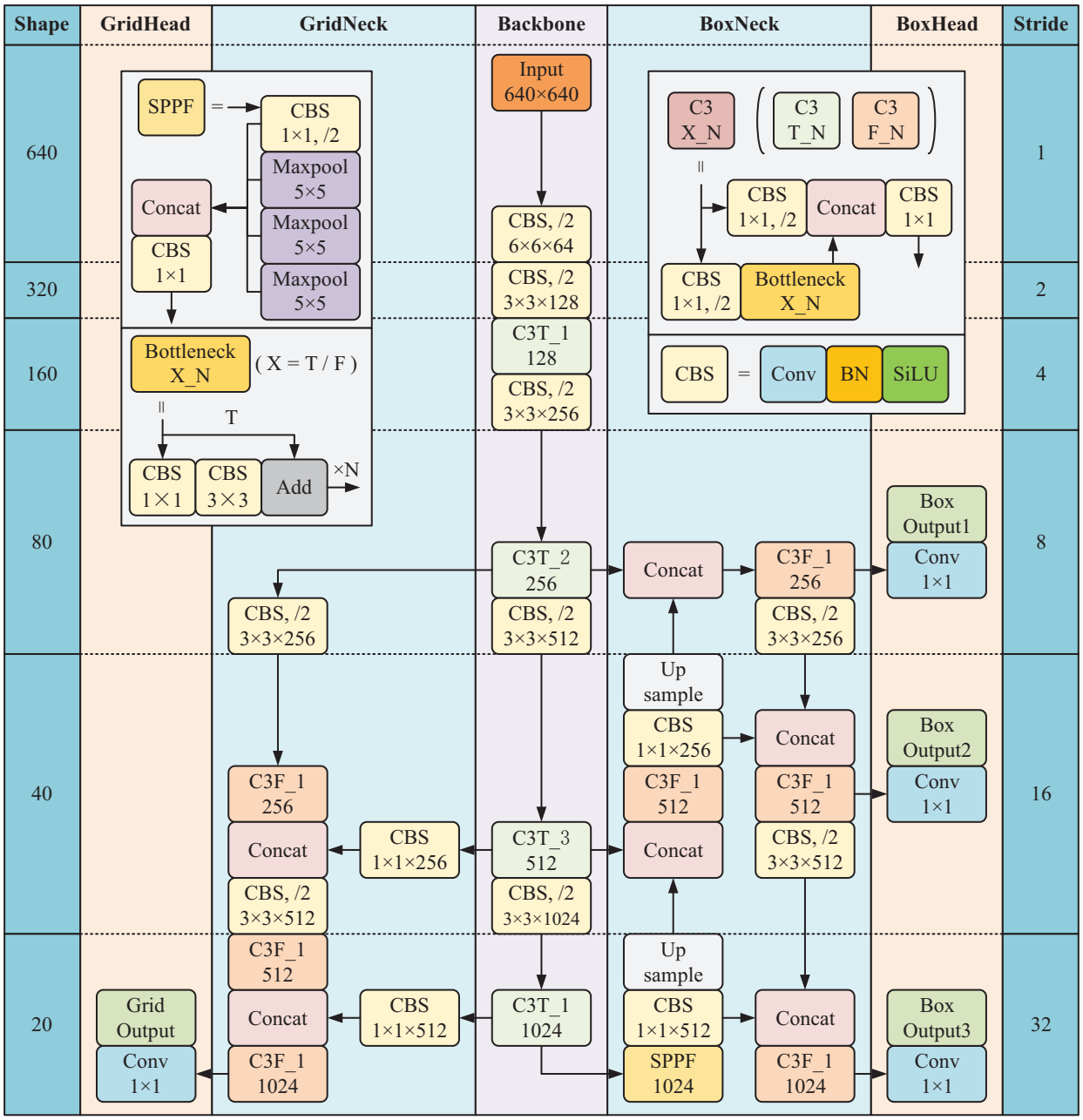


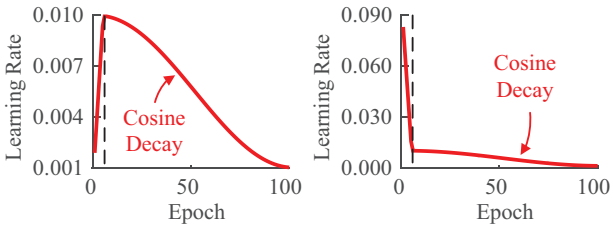
FIGURE 4 Network architecture. C3, Cross Stage Partial Bottleneck with 3 Convolutions; CBS, Convolution, Batch normalization, SiLU; Conv, Convolution; BN, Batch normalization; SiLU, Sigmoid Linear Unit; SPPF, Spatial Pyramid Pooling - Fast; T, True; F, False; N, Number

that theoretically a maximum of 99.3% of objects can be detected with the above anchors.

3.4 | Postprocessing

3.4.1 | NMS-ARS

NMS is a method for searching local maxima (Neubeck & Van Gool, 2006). Based on the classic NMS, multiple NMS have been developed and reported, such as Soft-NMS (Bodla et al., 2017), *GIoU*-NMS (Rezatof, 2019), *DIoU*-NMS, and *CIoU*-NMS (Zheng et al., 2020). However, these improved methods mainly solve the problem of object overlap, and no method explicitly considers crack topology. For example, in Figure 8, the NMS suggests three prediction boxes, A, B, and C, with high confidence. The *IoU* between any two boxes is below the usual threshold, and thus the predicted cracks are counted twice. If the *IoU* value is too low, other cracks would be filtered out incorrectly. Therefore, in this paper, area-reduction-suppression (ARS) is proposed to enhance NMS toward properly counting cracks.



(a) Weight (b) Bias

FIGURE 5 The learning rate schedules.

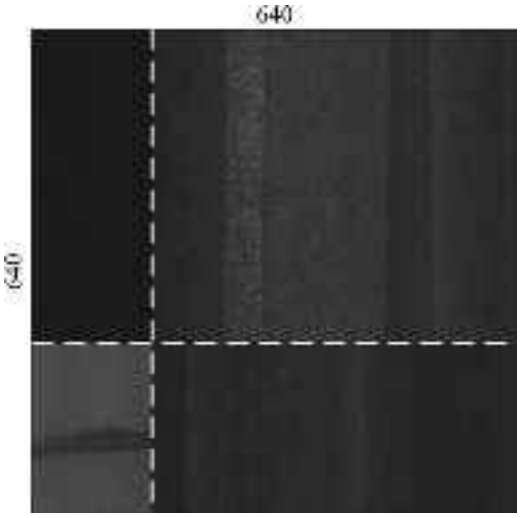


FIGURE 6 Example of a mosaic used for training

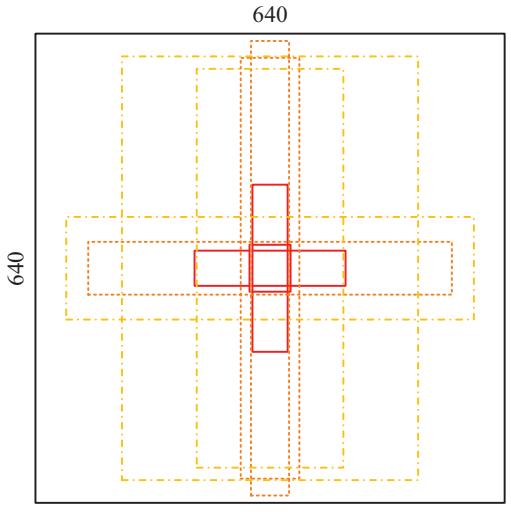


FIGURE 7 Anchor boxes

The concept behind the algorithm is to retain the prediction boxes with high confidence and suppress the confidence of the prediction boxes that have a high overlap area. The pseudocode is shown in Algorithm 1. First, it is optional to translate boxes by classes so that objects of different classes do not overlap. Then, in the algorithm, the loop is continued until $\mathcal{B}$ is empty. During the looping, the lowest confidence is assigned subscript m in $\mathcal{S}$, and corresponding elements in $\mathcal{B}$ and $\mathcal{S}$ are deleted. In the next step, the product p_m of $1 - IoA(b_m, b_i)$ of all boxes b_i with higher confidence than s_m is calculated. $IoA(b_m, b_i)$ is the ratio of the overlap area of b_m and b_i to the area of b_m. Finally, the updated confidence is obtained through the rescoring function as follows.

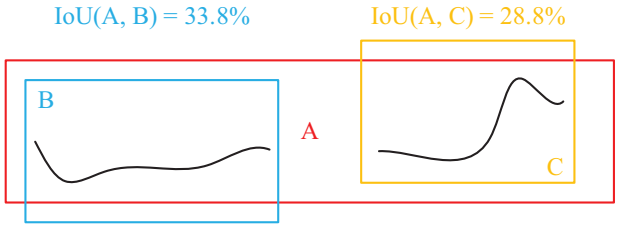


FIGURE 8 Crack detection via non-maximum suppression

$$f\ (x) = e^{\frac{x^2-1}{2}} \tag{3}$$

In the preceding method, IoA is employed instead of IoU to avoid abnormally low IoUs caused by a significant difference in the areas of the two boxes. $IoA(b_m,\ b_i)$ is calculated for all boxes b_i with higher confidence than s_m. Then $1 - IoA(b_m, b_i)$ is the probability $p_{m,i}$ that b_m is different from b_i and is an independent object. It is assumed that $p_{m,i}$ are independent of each other, and the product reflects the probability that b_m is an independent object. Considering the discontinuity of $1 - IoA$ as the penalty function, Equation (3) is proposed by referring to the gaussian penalty function for the pruning step (Bodla et al., 2017).

Algorithm 1. Area-Reduction-Suppression (ARS)

Input: after-NMS detection boxes $\mathcal{B} = \{b_1, \ldots, b_N\}$
after-NMS detection scores $\mathcal{S} = \{s_1, \ldots, s_N\}$

$\mathcal{D} \leftarrow \{\}, \mathcal{P} \leftarrow \{\}$
while $\mathcal{B} \neq \phi$ **do**
  $m \leftarrow argmin\ \mathcal{S}$
  $\mathcal{B} \leftarrow \mathcal{B} - b_m, \mathcal{S} \leftarrow \mathcal{S} - s_m$
  $p_m \leftarrow 1$
  for $b_i \in \mathcal{B}$ **do**
    $p_m \leftarrow p_m \times (1 - IoA(b_m, b_i))$
  end for
  $p_m \leftarrow s_m \times f(p_m)$
  $\mathcal{D} \leftarrow \mathcal{D} \cup b_m, \mathcal{P} \leftarrow \mathcal{P} \cup p_m$
end while
return $\mathcal{D}, \mathcal{P}$

Output: after-ARS detection boxes $\mathcal{D} = \{d_1, \ldots, d_{N'}\}$
after-ARS detection scores $\mathcal{P} = \{p_1, \ldots, p_{N'}\}$

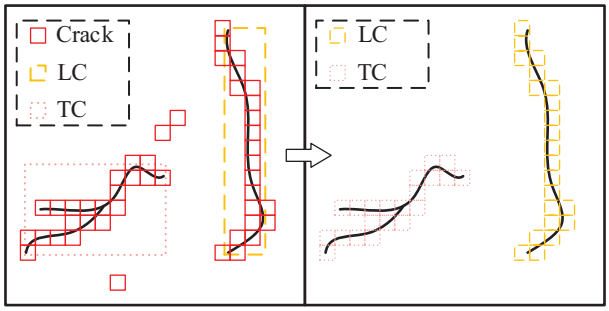


FIGURE 9 Mapping and matching. LC, longitudinal crack; TC, transverse crack

	Actual Positive P	Actual Negative N
Predicted Positive P	True Positive TP	False Positive FP
Predicted Negative N	False Negative FN	True Negative TN

FIGURE 10 Confusion matrix

3.4.2 | MaM

MaM is proposed to filter noise and determine the types of cracks captured in each grid cell. As shown in Figure 9, cells outside the detection box are filtered out as noise, and cells inside the box are classified into corresponding classes.

Algorithm 2. Mapping and Matching (MaM)

Input: detection boxes $\mathcal{B} = \{b_1, \ldots, b_N\}$
box classes $\mathcal{C} = \{c_1, \ldots, c_N\}$
classification grids $\mathcal{G} = \{g_1, \ldots, g_K\}$
grid classes $\mathcal{Q} = \{q_1, \ldots, q_K\}$

$\mathcal{H} \leftarrow \{\}, \mathcal{J} \leftarrow \{\}$
for $g_i \in \mathcal{G}$ **do**
  $\mathcal{T} \leftarrow \{\}$
  for $b_j \in \mathcal{B}$ **do**
    $t_j \leftarrow IoA(g_i, b_j)$
    $\mathcal{T} \leftarrow \mathcal{T} \cup t_j$
  end for
  $r \leftarrow max\ \mathcal{T}$
  if $r > 0$ **then**
    $m \leftarrow argmax\ \mathcal{T}$
    $\mathcal{H} \leftarrow \mathcal{H} \cup g_i, \mathcal{J} \leftarrow \mathcal{J} \cup c_m$
  end if
end for
return $\mathcal{H}, \mathcal{J}$

Output: after-MaM classification grids $\mathcal{H} = \{h_1, \ldots, h_{K'}\}$
after-MaM grid classes $\mathcal{J} = \{j_1, \ldots, j_{K'}\}$

The pseudocode is shown in Algorithm 2. The first step is to calculate the IoA between the cells and all boxes. If the maximum value is zero, then the cell is noise and is filtered; otherwise, the cell is assigned the class of box corresponding to the maximum value to the grid. In the case that the IoA of the grid and several boxes reach the maximum value simultaneously, the class of the cell is determined according to the principle that AC have the highest priority, while LC have the lowest priority. This principle considers the importance of various pavement cracks. Thus, the above algorithm yields higher quality and more accurate information by combining the output of boxes and cells.

4 | EXPERIMENTS

4.1 | Evaluation metrics

Precision (Pr), recall (Re), and F1 score (F1) are used as evaluation metrics. These metrics require four crucial parameters as shown in Figure 10. Pr measures the proportion of positive cases correctly predicted to positive cases predicted as shown in Equation (4). Re measures the proportion of positive cases correctly predicted to actual positive cases as shown in Equation (5). F1 is the overall evaluation of Pr and Re as shown in Equation (6).

$$Pr = \frac{TP}{TP + FP} \tag{4}$$

$$Re = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = \frac{2 \times Pr \times Re}{Pr + Re} \tag{6}$$

Additionally, average precision (AP) is also adopted as the evaluation metric for classification tasks. AP is obtained by calculating the area under the interpolated Pr/Re curve according to the new standard of Pattern Analysis, Statical Modeling and Computational Learning (PASCAL) visual object classes (VOC) (Everingham et al., 2015) as shown in Equation (7). For the detection task, the IoU threshold is used to determine whether the object is accurately located so as to determine whether the object is accurately predicted. AP_{50} is one of the most commonly used AP metrics and is implemented with an IoU

TABLE 2 Dependency version

Name	Version
Python	3.9.11
PyTorch	1.11.0
torchvision	0.12.0
CUDA	11.4
cuDNN	8.2
cudatoolkit	11.3

threshold of 0.5. Due to the fuzziness of crack boundaries, it is appropriate to choose AP_{50} as the evaluation metric for the detection task. The mean AP (mAP), which is the average APs of all classes, is adopted to measure the overall network performance for the classification task. Similarly, mAP_{50} is used for the detection task. To comprehensively evaluate the fusion model performance, $mAP_{grid\text{-}box}$ is defined as the weighted average of mAP and mAP_{50} as shown in Equation (8). Considering the learning difficulty of classification and detection tasks, the ratio between mAP and mAP_{50} is 1:9.

$$AP = \sum_{i=1}^{n-1} (Re_{i+1} - Re_i) \times Pr_{interp}(Re_{i+1}) \quad (7)$$

$$mAP_{grid-box} = \beta\, mAP + (1-\beta)\, mAP_{50} \quad (8)$$

4.2 | Development environment

The training process is performed on a workstation with high-performance graphic processing units (GPUs) and center processing units (CPUs). The two GPUs are NVIDIA Ampere A100 with 80 GB memory, and the two CPUs are Intel Xeon Gold 6342. The operating system is CentOS7, and coding was performed with Python and the PyTorch library. The versions of the dependencies are shown in Table 2.

4.3 | Fusion results

The weight with the best performance on the validation set in the training process served as the final weight. As shown in Figure 11, the weight of the 64th epoch was selected through $mAP_{grid\text{-}box}$.

The fusion model integrates micro-information and macro-information. As shown in Figure 12, the MaM algorithm enabled the classification of the crack type in each grid cell. Meanwhile, misidentified background textures were effectively removed.

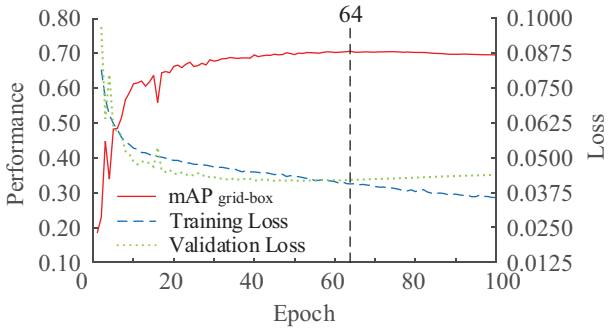


FIGURE 11 The overall training process. mAP, mean average precision

4.4 | Classification results

Variations in the evaluation metrics and the loss in the training process for the classification task are shown in Figure 13. Training loss decreased smoothly, while validation loss went down at first before rising slowly and smoothly after 30 epochs, indicating that overfitting has occurred. Pr, Re, and mAP oscillated before overfitting occurred and then remained nearly constant thereafter.

The weights obtained through training are verified on the test set, and the evaluation metrics of each class are shown in Figure 14. The markings on the pavement (i.e., "marking") are the easiest to learn, achieving the highest AP of 0.965 among the three classes in the test set. The second highest AP (0.919) belongs to pavement repairs, which is more complex than marking but is simpler than cracks. Last, cracks are the most complex among the three classes, and their AP reached 0.817. The mAP of the classification network on the test set reached 0.900.

The confidences corresponding to the maximum F1 score of the three classes exceed 0.8 and are relatively close. Therefore, the confidence of 0.8 is used for all categories. Meanwhile, it should be noted that the values of the evaluation metrics remained almost unchanged even after overfitting during the training process, indicating that the network in the overfitting stage mainly focused on learning confidence. As seen in the bottom row of Figure 15, 28%, 9%, and 4% of crack, repair, and marking, respectively, were not distinguished from the background. On the other hand, the Pr of crack, repair, and marking classifications are high, with little confusion between the categories.

The classification network accurately identifies cracks in the test set. It overcame disturbances from sources of noise, such as shadows, non-uniform illumination, SR, TMs, and water, as shown in Figure 16a, demonstrating the generalizability of the classification network across different types of asphalt pavement.

Furthermore, the receptive field of each cell in the grid-based classification method is larger than in the

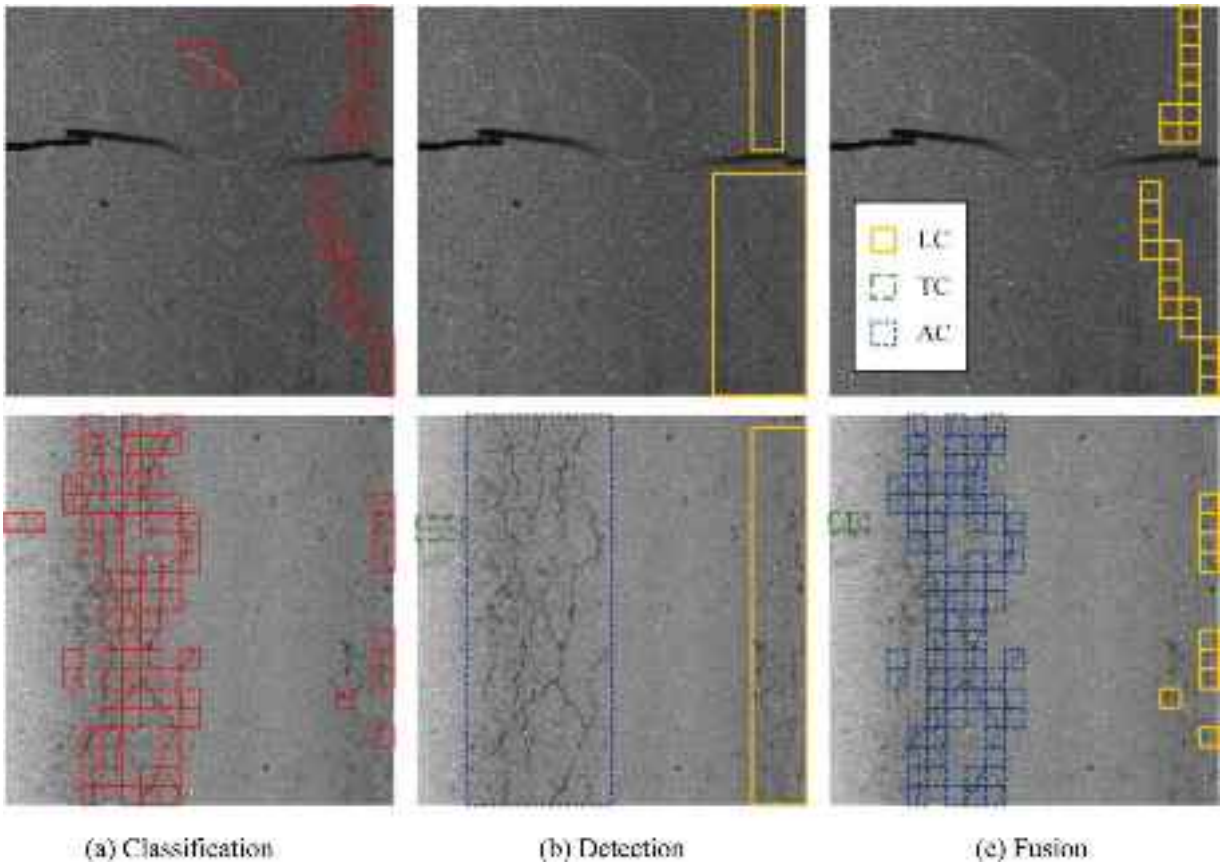


FIGURE 12 Fusion results of crack. AC, alligator cracks; LC, longitudinal crack; TC, transverse crack

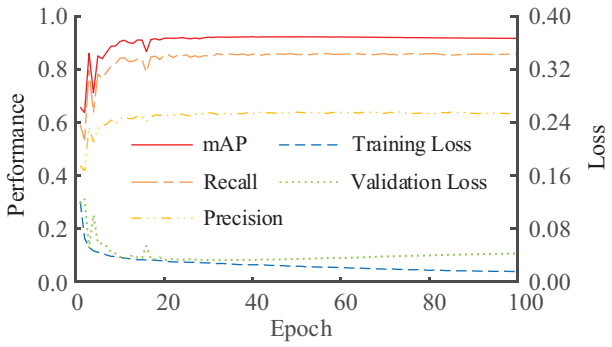


FIGURE 13 Performance and loss curve during the training process for classification. mAP, mean average precision

patch-based classification method. A sliding window with a stride half of the window scale (Figure 17) is unnecessary for avoiding cracks located at the edge of cells. In addition, the patch-based classification method needs to read patches many times. In contrast, the grid-based classification method only needs to read the image once so that the results of all cells can be obtained directly. As shown in Figure 17, the patch-based method needs to process 41 patches, while the grid-based method only needs one calculation step to obtain the categories of the 16 cells. Under the same network architecture, the efficiency of the grid-based classification method is more than double that of the patch-based classification method.

4.5 | Detection results

As shown in Figure 18, the overfitting of the detection network occurred later than that of the classification network, and the validation loss increased gradually after 60 epochs. $\mathcal{L}_{loc}$ and $\mathcal{L}_{cls}$ in the training set are generally lower than those in the validation set. On the contrary, $\mathcal{L}_{obj}$ is consistently higher on the training set than on the validation set. The higher training loss of objection is because $\mathcal{L}_{obj}$ smooths positive samples with IoU, while mosaic data augmentation reduces IoU through image clipping and splicing. Pr and Re fluctuate slightly in the overfitting stage and remain generally stable. mAP_{50} decreased slowly in the overfitting stage, indicating that generalizability gradually degraded.

Object detection requires both classification and localization, which, as demonstrated by the experimental results, is more complex than solely performing classification. The evaluation metrics for crack detection are much lower than the metrics for classification (Figure 19). The

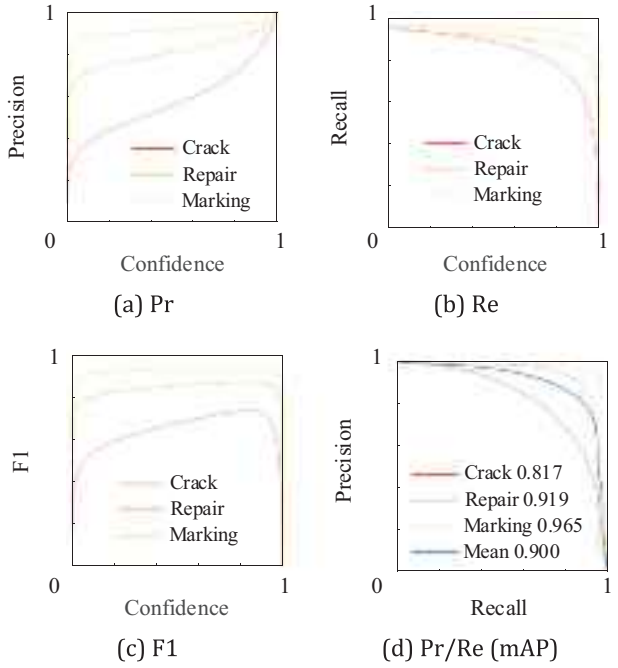


FIGURE 14 Performance on the test set of classification. F1, F1 score; mAP, mean average precision; Pr, precision; Re, recall

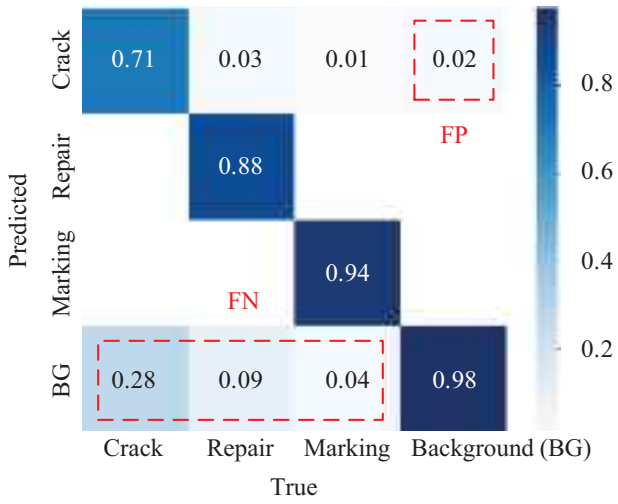


FIGURE 15 Confusion matrix at confidence 0.8 of classification

Pr curve is smooth when the confidence is low and oscillates when the confidence is high due to the low Re. mAP_{50} shows a similar behavior as the classification training; that is, markings are most easily detected, while cracks were the most difficult to detect. Among the crack types, AC are the easiest for detection network, while LC are the most challenging.

The maximum F1 scores of all classes indicate low confidence. From the confusion matrix with a confidence of 0.1 in Figure 20, it can be seen that the low scores of 0.45, 0.41, 0.22, 0.15, and 0.03 for LC, TC, AC, SR, and TM, respectively, indicate that these entities were not identified from among the background. AC were slightly confused with LC. In the actual classification of asphalt pavement crack, lighter AC can be similar in morphology to the LC and sometimes can be misclassified as LC. Therefore, this confusion is consistent with human practice. Except for the TM, all the other classes were mostly confused with the background. The characteristics of AC and SR are the most apparent, and the confusion is relatively light. In contrast, 0.31 of LC and 0.28 of TC were incorrectly detected from the background, respectively, suggesting that the characteristics of linear cracks are similar to the background. A robust feature extraction network is required to address this issue.

As shown in Figure 16b, the detection network overcomes visual interference from ruts, oil stains, fallen leaves, weeds, gravel, and soil. The results of the detection network are more intuitive and transparent and give more macro-information than the classification network.

5 | DISCUSSIONS

5.1 | Joint training and fusion model

Multi-task learning and joint training exploit the valuable information contained in multiple learning tasks to help each task learn more accurate features. To investigate whether joint training improves the model performance, the DC of the combined loss function is analyzed.

The results show that multi-task learning and joint training improve model accuracy. As shown in Figure 21, as the proportion of $\mathcal{L}_{grid}$ in the combined loss function increases, $mAP_{grid\text{-}box}$ first increases before decreasing to reach the maximum value of 0.684 when the DC is 0.2. mAP rapidly increases to 0.891 when DC increased from 0 to 0.01. mAP then peaks at DC of 0.2 before slowly decreasing and oscillating slightly near 1. As the DC decreases from 1 to 0, mAP_{50} increases slowly before peaking at DC of 0.2 and decreasing slowly thereafter. Compared with an mAP_{50} of 0.650 obtained by single detection network training, multi-task learning reached an mAP_{50} of 0.660 when DC is 0.2, which is an increase of 1.0% (Table 3). The fusion model's accuracy improved by sharing detection and classification information.

The fusion model is also highly robust. As shown in Figure 21, when DC increases from 0.01 to 0.6, the fusion model's accuracy approximates and even exceeds the accuracy of the individual classification and detection models. When DC increases from 0.01 to 0.99, the overall accuracy of the fusion model did not change appreciably. The fusion model is insensitive to changes in the DC and reliably improved the network performance by enabling

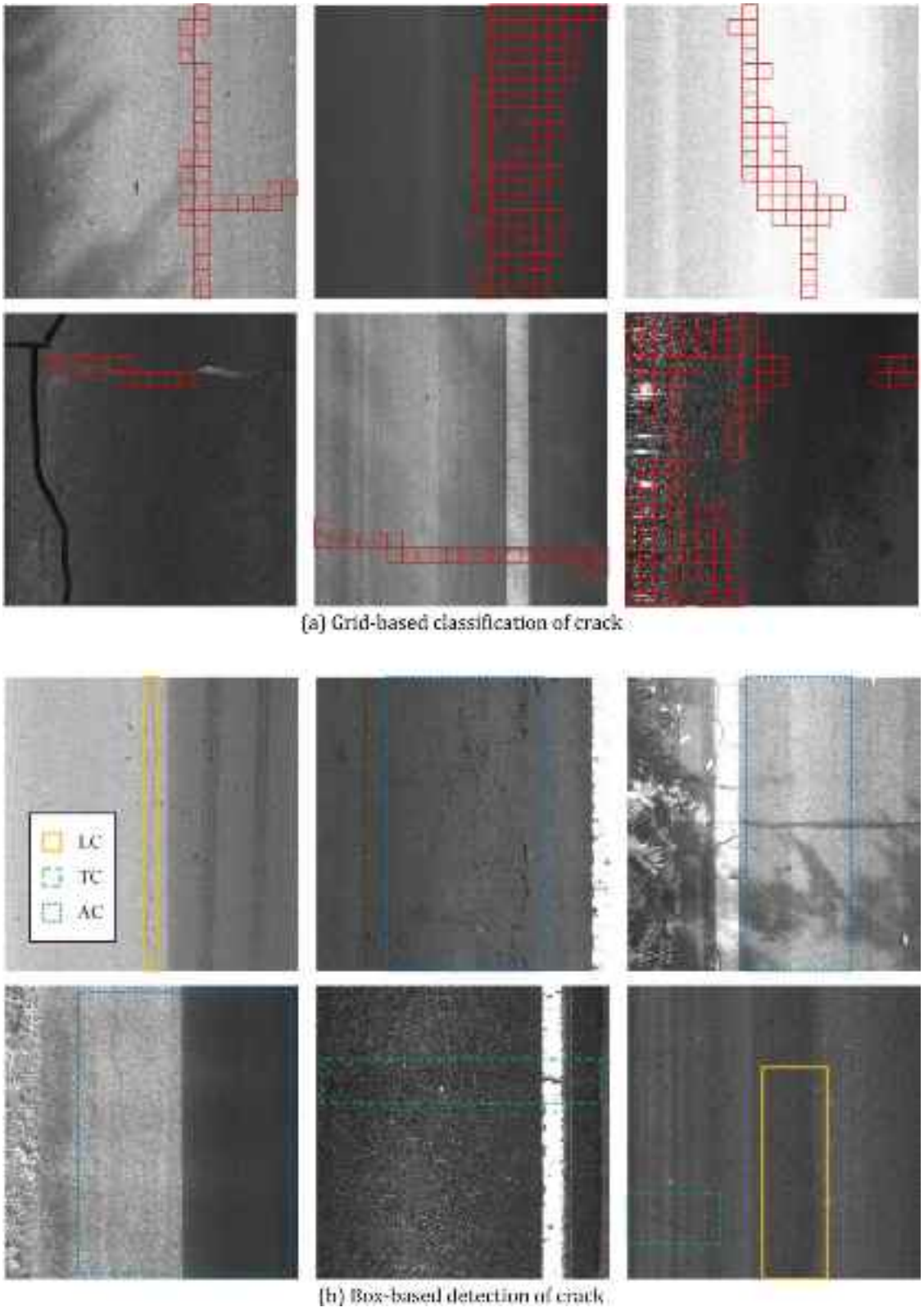
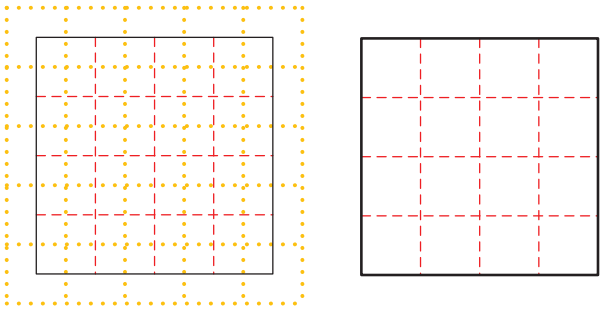


FIGURE 16 Grid-based and box-based crack identification. AC, alligator cracks; LC, longitudinal crack; TC, transverse crack

TABLE 3 Accuracy of fusion model and single model

Metrics	Detection (DC = 0.0)	Fusion (DC = 0.2)	Classification (DC = 1.0)
Crack AP	0.152	0.817	0.816
mAP	0.058	0.900	0.900
LC AP	0.405	0.411	0.001
TC AP	0.461	0.480	0.001
AC AP	0.628	0.634	0.008
mAP_{50}	0.650	0.660	0.002
$mAP_{grid\text{-}box}$	0.591	0.684	0.092

Abbreviations: AC, alligator crack; DC, distribution coefficient; LC, longitudinal crack; mAP, mean average precision; TC, transverse crack.



(a) patch-based (b) grid-based

FIGURE 17 Comparison of two classification methods

TABLE 4 The efficiency of the fusion and single models

Module	Parameters	GFLOPs
Backbone	3,514,560	10.4
Grid neck	2,456,320	2.1
Grid head	1539	–
Box neck	3,491,584	5.4
Box head	26,970	–
Grid-classification	5,972,419	12.5
Box-detection	7,033,114	15.9
Fusion model	9,490,973	17.9

Abbreviation: GFLOPs, giga floating point of operations.

cooperation between the classification task and the detection task.

The fusion model shares the backbone network for classification and detection and reduces the computational load. The number of parameters and giga floating point of operations (GFLOPs) of the fusion model are listed in Table 4. If the classification and detection networks were to be separate, the number of parameters will increase by 37%, and GFLOPs will increase by 58%, resulting in repeated feature extraction and a severe waste of computing resources. On the NVIDIA Ampere A100 with 80 GB memory, four thousand 640 × 640 images were tested with a batch size of 32. The average inference times per image for the fusion model, classification model, and detection model are 1.157, 0.812, and 1.025 ms, respectively. The fusion model increased the image processing capacity by 58% per unit of time.

Therefore, the fusion network improves accuracy and efficiency and has strong robustness. The fundamental operations of the crack classification and detection tasks are similar. Both networks can extract common features from objects portrayed in images. These common features guarantee the robustness of the fusion model. Beyond these universal features, these two tasks have different goals and distinct priorities. Classification tasks focus on local areas and extract micro-information. Detection tasks focus on the whole image and extract macro-information. It is believed that sharing information at different levels can effectively improve the model's feature extraction capability and generalizability. Meanwhile, sharing the backbone network avoids repeated extraction of common features and improves model efficiency.

5.2 | NMS

NMS is important in anchor-based objection detection. In order to investigate the effectiveness of the designed NMS-ARS algorithm, NMS-ARS is compared with the classical NMS algorithm using the presented asphalt pavement dataset.

The NMS proposed previously in the literature did not specifically target the topology of cracks. Table 5 shows that the performance gap between different methods narrowed after the addition of the ARS algorithm. Soft-NMS is specially designed for object overlap, but its strength is not applicable to the scene of crack detection, and thus Soft-NMS did not exhibit any obvious advantages in performance. The comparison also showed that NMS-ARS had improved detection accuracy over various NMS.

The ARS algorithm fully considers the crack topology. Compared with IoU, which is commonly used in

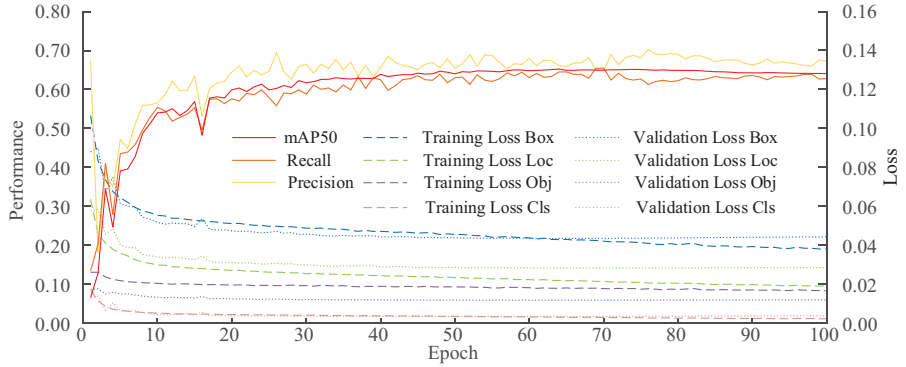


FIGURE 18 Performance and loss curve during the training process for detection. mAP, mean average precision

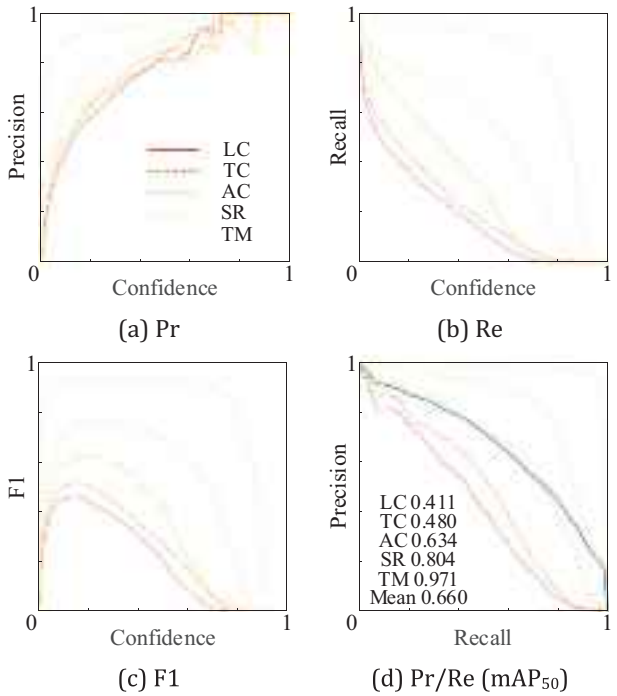


FIGURE 19 Performance on the test set of detection. (a) Precision (Pr), (b) recall (Re), (c) F1 score, and (d) Pr/Re (mAP_{50}). AC, alligator crack; LC, longitudinal crack; mAP, mean average precision; SR, strip repair; TC, transverse crack; TM, traffic marking

TABLE 5 AP_{50} of each NMS

NMS	LC	TC	AC	mAP_{50}
Classical NMS	0.405	0.472	0.631	0.655
NMS-ARS	0.411	0.480	0.634	0.660
GIoU-NMS	0.403	0.470	0.629	0.654
GIoU-NMS-ARS	0.410	0.479	0.633	0.659
DIoU-NMS	0.403	0.470	0.798	0.654
DIoU-NMS-ARS	0.410	0.479	0.633	0.659
Soft-NMS	0.392	0.460	0.620	0.644
Soft-NMS-ARS	0.402	0.472	0.627	0.651

Abbreviations: AC, alligator crack; ARS, area-reduction suppression; LC, longitudinal crack; mAP, mAP, average precision; NMS, non-maximum suppression; TC, transverse crack

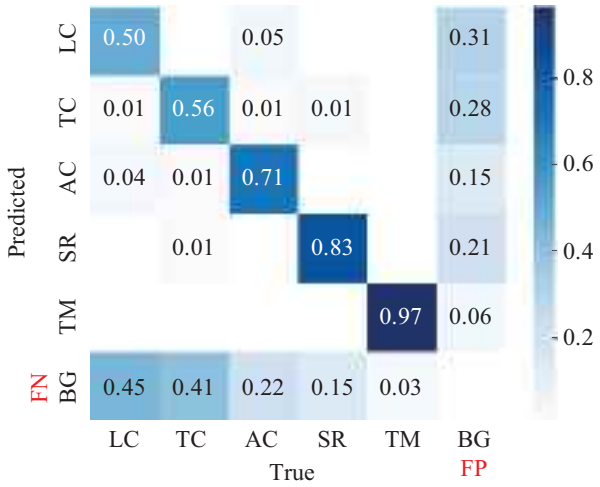


FIGURE 20 Confusion matrix at confidence 0.1 of detection. AC, alligator crack; BG, background; LC, longitudinal crack; SR, strip repair; TC, transverse crack; TM, traffic marking

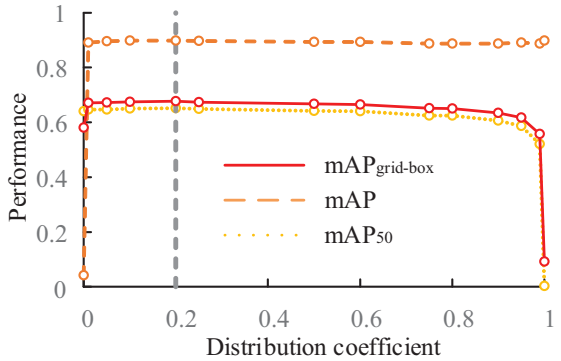


FIGURE 21 Changes in performance with respect to the distribution coefficient. mAP, mean average precision

NMS, *IoA* is employed to consider the linear characteristics of cracks in the postprocessing stage. The Gaussian penalty function is employed to smooth the pruning step and further suppress the generation of redundant

prediction boxes. At present, almost no postprocessing for crack topology in anchor-based detection has been reported in the literature, and the work in this paper can serve as a reference for future work.

5.3 | Label and prediction

The proposed model performs well on most of the images in the test set. However, some cracks were misidentified or were missed entirely. As shown in the first column of Figure 22a,b, some cracks were covered by TM paint, affecting identification and thus eluding detection. In the second column, manual inspection failed to detect the crack, whereas the proposed model successfully detected the crack albeit imperfectly due to noise. In the third column, manual inspection identified cracks but incompletely due to the fuzzy boundary of cracks; however, the proposed model detected cracks well. As shown in the first column of Figure 22c,d, manual detection missed transverse and LC in the image's upper right, while the proposed model successfully captured the cracks. In the second column, the proposed model failed to detect the longitudinal crack along the edge of the marking but successfully detected a transverse crack at the top of the image. Due to the fineness of the transverse crack, manual labeling missed the crack. In the third column, the image is full of noise. The proposed model identified the part of the noise near cracks as part of the cracks, resulting in longer identified LC than the labeled ones. Due to the high level of noise, the cracks in the noise at the lower-left corner were misidentified by both manual recognition and the proposed model.

Manual annotation errors and limited model generalization lead to decreased evaluation metrics. During visual inspection, an unavoidable amount of tagging errors may occur with human labelers due to human-specific issues such as weariness, but such errors will not occur with an automatic identification algorithm. As for model generalization, further exploration is needed in the future. For example, a potential direction is to use the entire, massive library of unlabeled data for unsupervised learning to train a powerful and effective feature extraction network.

5.4 | Result comparison

Many researchers have worked on pavement crack identification, but most of their models were trained and tested on private, unshared datasets. Thus, it is difficult to directly compare different models. This section summarizes recently reported research and discusses key issues. Since F1 is given in most studies or can be indirectly calculated by other metrics, F1 is used here as a benchmark to evaluate different methods (Table 6).

As seen in Table 6, most of the classification tasks are patch-based, and the shape of each patch is large. As the patch size becomes smaller, the task begins to resemble segmentation and thus becomes more difficult (X. Wang & Hu, 2017). The shape of each grid cell in this paper is only 20×20 and is thus in a complicated middle ground. In addition, the test set in this paper includes complex scenarios, which also increases the difficulty of classification. The presented model has a crack F1 of 0.74 in the proposed dataset. Considering the small shape of the grid cells, the large scale (i.e., 8 million images), and the complexity of the dataset, the F1 can be considered to have reached a relatively high level.

Furthermore, the positive and negative samples of the patch-based classification method are balanced, while those of the grid-based classification method are relatively unbalanced, resulting in an increase of false positives, a decrease of Pr, and a decrease of F1. The Receiver Operating Characteristic (ROC) curve and area under curve (AUC) can eliminate the influence of positive and negative sample imbalance. However, since most of the literature did not provide the ROC curve and AUC, F1 is used to compare different methods in this paper. The balance between positive and negative samples will significantly improve F1 scores in grid-based classification. The proportion of crack cells and background cells in the constructed dataset is about 1:12. If background samples are reduced to 1/12, assuming FPs are reduced equally to 1/12, then F1 is about 0.92 according to Equations (4)–(6).

In the detection task, the scale of most datasets is less than 10,000 images. When the test set is large, the requirement for model generalization is higher. In this paper, the F1 of LC, TC, and AC are 0.47, 0.53, and 0.64, respectively. Similar to the results of other publications, the F1 for AC detection is significantly higher than for LC and TC detection. TC was more readily identified than LC in the top-view datasets than in the wide-view datasets, likely because wide view better captures the characteristics of the LC. Meanwhile, transverse and longitudinal crack features were captured in a more balanced manner from a top-down view. Considering the scale and complexity of the constructed dataset, the F1 of the detection network is quite satisfactory.

6 | CONCLUSION

In this paper, a fusion model combining GCBD was proposed for the first time. The largest double-labeled asphalt pavement dataset was assembled to train and validate the presented model. The NMS-ARS algorithm was designed to consider the crack topology in the postprocessing stage. The MaM algorithm was developed to combine the

TABLE 6 Published research regarding deep learning for pavement crack identification

Reference	Task	View	Shape	Dataset (training: validation: test)	Method	F1
(L. Zhang et al., 2016)	Patch-classification	Wide view	99 × 99	640,000: 160,000: 200,000	ConvNet	Crack: 0.90
(Gopalakrishnan et al., 2017)	Classification	Top-down	2048 × 3072	760: 84: 212	VGG-16	Crack: 0.90
(X. Wang & Hu, 2017)	Patch-classification	Wide view	64 × 64	30,000: –: –	CNN	Crack: 0.95
(Park et al., 2019)	Patch-classification	Wide view	40 × 40	30,000: 6000: 16,486	CNN	Crack: 0.82
This study	Grid-classification	Top-down	20 × 20	4,800,000: 1,600,000: 1,600,000	GCBD	Crack: 0.74 Modified: 0.92
(Maeda et al., 2018)	Detection	Wide view	300 × 300	7240: –: 1813	SSD	LC: 0.52 TC: 0.33 AC: 0.68
(Ibragimov et al., 2020)	Detection	Top-down	1865 × 2000	2600: 600: 400	Faster R-CNN	LC/TC: 0.40 AC: 0.82
(Mandal et al., 2018)	Detection	Wide view	–	7240: –: 1813	YOLO v2	LC: 0.76 TC: 0.71 AC: 0.77
(Du et al., 2021)	Detection	Wide view	–	24,654: 10,566: 10,568	YOLO v3	LC/TC: 0.46 AC: 0.51
This study	Detection	Top-down	640 × 640	12,000: 4,000: 4000	GCBD	LC: 0.47 TC: 0.53 AC: 0.64

Abbreviations: AC, alligator crack; CNN, convolutional neural network; F1, F1 score; GCBD, grid-based classification and box-based detection; LC, longitudinal crack; TC, transverse crack.

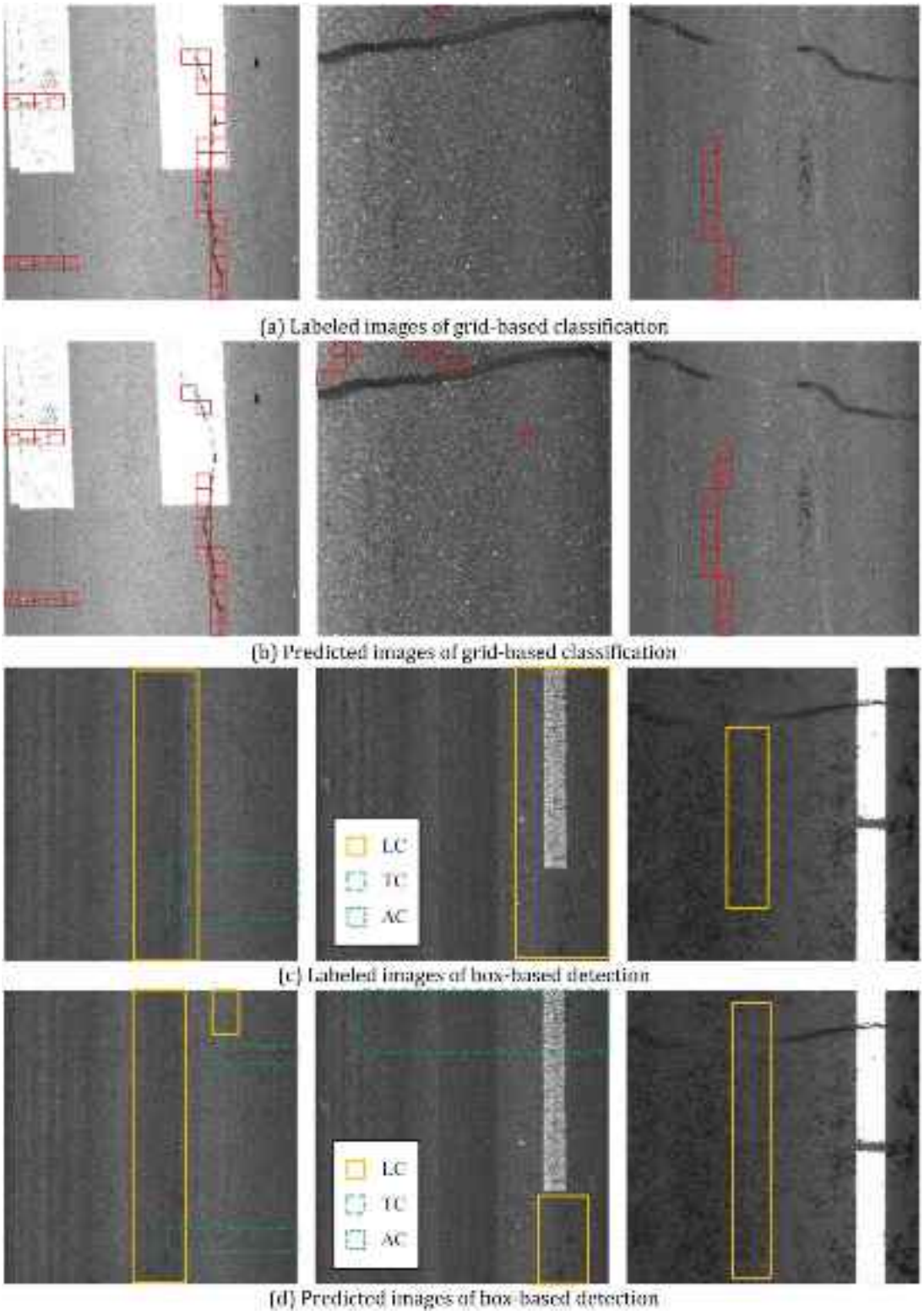


FIGURE 22 Comparison of labels and predictions. AC, alligator cracks; LC, longitudinal crack; TC, transverse crack

advantages of GCBD. The following conclusions can be drawn from the experiments and discussions.

1. The performance of the grid-based classification network is higher than the patch-based classification network due to the broader receptive field and the need for only one calculation step to classify all grid cells. The mAPs for classifying cracks, repairs, and markings are 0.817, 0.919, and 0.965, respectively. TMs are the easiest to identify, and cracks are the hardest. Since the

grid-based classification method has a large receptive field, a classification network that directly recognizes different types of cracks can be trained using a dataset with annotated cracks.
2. The detection network effectively classifies and locates cracks. The mAP_{50} of longitudinal crack, transverse crack, alligator crack, strip repair, and TMs are 0.411, 0.480, 0.634, 0.804, and 0.971, respectively. Among cracks, AC are the easiest to identify, while LC are the most difficult.
3. Multi-task learning and joint training improve the accuracy of pavement crack identification. Through sharing a backbone network, the fusion model reduces the number of parameters by 37% and GFLOPs by 58%, compared to the case in which classification and detection networks are deployed separately. The combination thus significantly improves efficiency. The fusion model also has strong robustness to the DC.
4. NMS-ARS, which considers crack topology, effectively filters the redundant prediction boxes and improves the accuracy of pavement crack detection. The optimized NMS while suited for other objection detection tasks, such as object overlap, may not be conducive to pavement crack detection. The gap between different NMS algorithms can be narrowed by adopting the ARS algorithm.

The proposed method realizes automatic crack identification for asphalt pavement. The fusion model simultaneously outputs grid-level classification and crack locations. Grid cells and boxes can be used to estimate the area of cracks. PCI can be used to evaluate pavement health condition and perform preventive maintenance. Furthermore, based on the proposed method, a number of other advanced approaches can be explored, including unsupervised learning performed on a massive pavement image database, network structure optimization through analyzing asphalt pavement crack features, or dynamic learning to improve model generalization.

ACKNOWLEDGMENT

This research is supported by the National Natural Science Foundation of China (No. 52192662, 51978376). The authors express sincere appreciation for their contribution to this research.

REFERENCES

Adeli, H. (2020). Four decades of computing in civil engineering. In C. Ha-Minh, D. Dao, F. Benboudjema, S. Derrible, D. Huynh, & Tang, A. (Eds.), *CIGOS 2019, Innovation for sustainable infrastructure* (pp. 3–11). Springer.

Abdel-Qader, I., Abudayyeh, O., & Kelly, M. E. (2003). Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering*, *17*(4), 255–263.

Adlinge, S. S., & Gupta, A. K. (2013). Pavement deterioration and its causes. *International Journal of Innovative Research and Development*, *2*(4), 437–450.

Bang, S., Park, S., Kim, H., & Kim, H. (2019). Encoder-decoder network for pixel-level road crack detection in black-box images. *Computer-Aided Civil and Infrastructure Engineering*, *34*(8), 713–727.

Basavaraju, A., Du, J., Zhou, F., & Ji, J. (2019). A machine learning approach to road surface anomaly assessment using smartphone sensors. *IEEE Sensors Journal*, *20*(5), 2635–2647.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS–Improving object detection with one line of code. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy (pp. 5561–5569).

Chen, H., Lin, H., & Yao, M. (2019). Improving the efficiency of encoder-decoder architecture for pixel-level crack detection. *IEEE Access*, *7*, 186657–186670.

Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, *186*, 1031–1045.

Du, Y., Pan, N., Xu, Z., Deng, F., Shen, Y., & Kang, H. (2021). Pavement distress detection and classification based on YOLO network. *International Journal of Pavement Engineering*, *22*(13), 1659–1672.

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The Pascal Visual Object Classes challenge: A retrospective. *International Journal of Computer Vision*, *111*(1), 98–136.

Feng, X., Xiao, L., Li, W., Pei, L., Sun, Z., Ma, Z., Shen, H., & Ju, H. (2020). Pavement crack detection and segmentation method based on improved deep learning fusion model. *Mathematical Problems in Engineering*, *2020*, 1–22.

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., & Agrawal, A. (2017). Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, *157*, 322–330.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., & He, K. (2017). Accurate, large minibatch SGD: Training ImageNet in 1 hour. arXiv, abs/1706.02677.

Guan, J. C., Yang, X., Ding, L., Cheng, X. Y., Lee, V. C. S., & Jin, C. (2021). Automated pixel-level pavement distress detection based on stereo vision and deep learning. *Automation in Construction*, *129*, 103788.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904–1916.

Hoang, N. D., Nguyen, Q. L., & Tran, V. D. (2018). Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. *Automation in Construction*, *94*, 203–213.

Hou, Y., Li, Q., Han, Q., Peng, B., Wang, L., Gu, X., & Wang, D. (2021). MobileCrack: Object classification in asphalt pavements using an adaptive lightweight deep learning. *Journal of Transportation Engineering, Part B: Pavements*, *147*(1), 04020092.

Hu, W., Wang, W., Ai, C., Wang, J., Wang, W., Meng, X., Liu, J., Tao, H., & Qiu, S. (2021). Machine vision-based surface crack analysis for transportation infrastructure. *Automation in Construction*, *132*, 103973.

Huang, Y. C., & Tsai, Y. C. (2011). Dynamic programming and connected component analysis for an enhanced pavement distress segmentation algorithm. *Transportation Research Record*, *2225*(1), 89–98.

Huyan, J., Li, W., Tighe, S., Zhai, J. Z., Xu, Z. C., & Chen, Y. (2019). Detection of sealed and unsealed cracks with complex backgrounds using deep convolutional neural network. *Automation in Construction*, *107*, 102946.

Ibragimov, E., Lee, H.-J., Lee, J.-J., & Kim, N. (2020). Automated pavement distress detection using region based convolutional neural networks. *International Journal of Pavement Engineering*, 1–12.

Jeong, D. (2020). Road damage detection using YOLO with smartphone images. *2020 IEEE International Conference on Big Data (Big Data)*, Atlanta, GA (pp. 5559–5562).

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P. (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, *29*(2), 196–210.

Koutsopoulos, H. N., & Downey, A. B. (1993). Primitive-based classification of pavement cracking images. *Journal of Transportation Engineering*, *119*(3), 402–418.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

Li, B., Wang, K. C. P., Zhang, A., Yang, E., & Wang, G. (2020). Automatic classification of pavement crack using deep convolutional neural network. *International Journal of Pavement Engineering*, *21*(4), 457–463.

Li, J. Q., Zhao, X. F., & Li, H. W. (2019). Method for detecting road pavement damage based on deep learning. *Health Monitoring of Structural and Biological Systems XIII*, *10972*, 517–526.

Li, L., & Jiang, R. (2021). Study on influence range of deflection on asphalt pavement cracks based on isometric determination. *Journal of Highway and Transportation Research and Development*, *38*(7), 17–21.

Lin, T. Y., c, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI (pp. 2117–2125).

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy (pp. 2980–2988).

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *European conference on computer vision* (pp. 740–755). Springer.

Liu, C., Li, J., Gao, J., Gao, Z., & Chen, Z. (2021). Combination of pixel-wise and region-based deep learning for pavement inspection and segmentation. *International Journal of Pavement Engineering*, *23*(9), 3011–3023.

Liu, J. W., Yang, X., Lau, S., Wang, X., Luo, S., Lee, V. C. S., & Ding, L. (2020). Automated pavement crack detection and segmentation based on two-step convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, *35*(11), 1291–1305.

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, (pp. 8759–8768).

Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1127–1141.

Mandal, V., Uong, L., & Adu-Gyamfi, Y. (2018). Automated road crack detection using deep convolutional neural networks. 2018 IEEE International Conference on Big Data (Big Data), *Seattle, WA* (pp. 5212–5215).

Miller, T. R., & Zaloshnja, E. (2009). Cost of crashes related to road conditions. *53rd Annual Scientific Conference of the Association for the Advancement of Automotive Medicine*, Baltimore, MD.

Ministry of Transport of the People's Republic of China. (2018). *Highway performance assessment standards: JTG 5210-2018*. People's Communications Publishing House.

Neubeck, A., & Van Gool, L. (2006). Efficient non-maximum suppression. *8th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, China pp. 850–855).

Nie, M., & Wang, C. (2019). Pavement crack detection based on YOLO v3. *2019 2nd International Conference on Safety Produce Informatization (IICSPI)*, Chongqing, China (pp. 327–330).

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66.

Park, S., Bang, S., Kim, H., & Kim, H. (2019). Patch-based crack detection in black box images using convolutional neural networks. *Journal of Computing in Civil Engineering*, *33*(3), 04019017.

Peraka, N. S. P., Biligiri, K. P., & Kalidindi, S. N. (2021). Development of a multi-distress detection system for asphalt pavements: Transfer learning-based approach. *Transportation Research Record*, *2675*(10), 538–553.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete. *ACI Materials Journal*, *114*(2), 237–244.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI (pp. 7263–7271).

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* Long Beach, CA (pp. 658–666).

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. Wells, & A Frangi, A. (Eds), *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Springer.

Sun, Y., Salari, E., & Chou, E. (2009). Automated pavement distress detection using advanced image processing techniques. *2009*

IEEE International Conference on Electro/Information Technology, Ontario, Canada (pp. 373–377).

Taheri, M., Jahanfar, M., & Ogino, K. (2017). Self-cleaning traffic marking paint. *Surfaces and Interfaces*, *9*, 13–20.

Tran, T. S., Tran, V. P., Lee, H. J., Flores, J. M., & Le, V. P. (2022). A two-step sequential automated crack detection and severity classification process for asphalt pavements. *International Journal of Pavement Engineering*, *23*(6), 2019–2033.

Tran, V. P., Tran, T. S., Lee, H. J., Kim, K. D., Baek, J., & Nguyen, T. T. (2021). One stage detector (RetinaNet)-based crack detection for asphalt pavements considering pavement distresses and surface objects. *Journal of Civil Structural Health Monitoring*, *11*(1), 205–222.

Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, Seattle, WA (pp. 390–391).

Wang, W., Wang, M., Li, H., Zhao, H., Wang, K., He, C., Wang, J., Zheng, S., & Chen, J. (2019). Pavement crack image acquisition methods and crack extraction algorithms: A review. *Journal of Traffic and Transportation Engineering (English Edition)*, *6*(6), 535–556.

Wang, X., & Hu, Z. (2017). Grid-based pavement crack analysis using deep learning. *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, Banff, Alberta, Canada (pp. 917–924).

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2019). Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, *21*(4), 1525–1535.

Zakeri, H., Nejad, F. M., & Fahimifar, A. (2017). Image based techniques for crack detection, classification and quantification in asphalt pavement: A review. *Archives of Computational Methods in Engineering*, *24*(4), 935–977.

Zhang, A., Li, Q., Wang, K. C., & Qiu, S. (2013). Matched filtering algorithm for pavement cracking detection. *Transportation Research Record*, *2367*(1), 30–42.

Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., Fei, Y., Liu, Y., Li, J Q., & Chen, C. (2017). Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, *32*(10), 805–819.

Zhang, A., Wang, K. C., Fei, Y., Liu, Y., Tao, S., Chen, C., Li, J Q., & Li, B. (2018). Deep learning-based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet. *Journal of Computing in Civil Engineering*, *32*(5), 04018041.

Zhang, K., Cheng, H. D., & Zhang, B. (2018). Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning. *Journal of Computing in Civil Engineering*, *32*(2), 04018001.

Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. *2016 IEEE international conference on image processing (ICIP)*, Phoenix, AZ (pp. 3708–3712).

Zhao, Z., Yang, X., Zhou, Y., Sun, Q., Ge, Z., & Liu, D. (2021). Real-time detection of particleboard surface defects based on improved YOLOV5 target detection. *Scientific Reports*, *11*(1), 1–15.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(07), 12993–13000.

Zhou, J., Huang, P. S., & Chiang, F. P. (2006). Wavelet-based pavement distress detection and evaluation. *Optical Engineering*, *45*(2), 027007.

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., & Wang, S. (2018). Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, *28*(3), 1498–1512.

How to cite this article: Li, B.-L., Qi, Y., Fan, J.-S., Liu, Y.-F., & Liu, C. (2023). A grid-based classification and box-based detection fusion model for asphalt pavement crack. *Computer-Aided Civil and Infrastructure Engineering*, *38,* 2279–2299. https://doi.org/10.1111/mice.12962