Structural Durability & Health Monitoring

Tech Science Press

DOI: 10.32604/sdhm.2024.042388

ARTICLE

# Rapid and Accurate Identification of Concrete Surface Cracks via a Lightweight & Efficient YOLOv3 Algorithm

**Haoan Gu[1], Kai Zhu[1], Alfred Strauss[2], Yehui Shi[3,4], Dragoslav Sumarac[5] and Maosen Cao[1,*]**

[1]College of Mechanics and Engineering Science, Hohai University, Nanjing, 211100, China

[2]Department of Civil Engineering and Natural Hazards, University of Natural Resources and Life Sciences, Vienna, 1180, Austria

[3]The First Geological Brigade of the Bureau of Geology and Mineral Resources of Jiangsu, Nanjing, 210041, China

[4]Control Technology Group Co., Nanjing, 210041, China

[5]Department of Technical Sciences, Civil Engineering, State University of Novi Pazar, Novi Pazar, 36300, Serbia

[*]Corresponding Author: Maosen Cao. Email: cmszhy@hhu.edu.cn

Received: 28 May 2023 Accepted: 01 November 2023 Published: 05 June 2024

## ABSTRACT

Concrete materials and structures are extensively used in transformation infrastructure and they usually bear cracks during their long-term operation. Detecting cracks using deep-learning algorithms like YOLOv3 (You Only Look Once version 3) is a new trend to pursue intelligent detection of concrete surface cracks. YOLOv3 is a typical deep-learning algorithm used for object detection. Owing to its generality, YOLOv3 lacks specific efficiency and accuracy in identifying concrete surface cracks. An improved algorithm based on YOLOv3, specialized in the rapid and accurate identification of concrete surface cracks is worthy of investigation. This study proposes a tailored deep-learning algorithm, termed MDN-YOLOv3 (MDN: multi-dilated network), of which the MDN is formulated based on three retrofit techniques, and it provides a new backbone network for YOLOv3. The three specific retrofit techniques are briefed: (i) Depthwise separable convolution is utilized to reduce the size of the backbone network; (ii) The dilated-down sampling structure is proposed and used in the backbone network to achieve multi-scale feature fusion; and (iii) The convolutional block attention module is introduced to enhance feature extraction ability. Results show that the proposed MDN-YOLOv3 is 97.2% smaller and 41.5% faster than YOLOv3 in identifying concrete surface cracks, forming a lightweight & efficient YOLOv3 algorithm for intelligently identifying concrete surface cracks.

## KEYWORDS

YOLOv3; concrete crack identification; MDN-YOLOv3; optimization retrofit techniques; depthwise separable convolution; dilated-down sampling; attention mechanism

## 1 Introduction

Concrete materials and structures have been extensively used in transportation infrastructure, typically highways and railways [1–3]. These materials and structures inevitably bear cracks during their long-term operation [4–6]. Owing to the large-scale characteristic of transportation infrastructure, the distribution of cracks in concrete materials and structures often exhibits divergence and dispersion [7,8]. The occurrence of cracks may impair the integrity and performance of related infrastructural systems [9,10]. The rapid

and accurate detection of cracks in concrete materials and structures becomes exceptionally crucial to the safety of transformation infrastructure [11–13]. Therefore, it is of great significance to develop an efficient technique to rapidly identify crack locations and patterns and evaluate the service performance of concrete structures [14,15].

Crack identification currently mainly relies on conventional routine-based visual inspection [16]. Unfortunately, visual inspection-based condition assessment is time-consuming and depends on the experience and knowledge of inspectors [17]. In general, visual inspection cannot authentically discover abrupt and subtle cracks and may miss potential risks in concrete structures [18]. To address this issue, computer vision-aided intelligent identification methods have become a research focus for the online detection of surface cracks. Numerous image processing methods have been utilized to detect surface defects or damage in structures. The main implementation methods can be divided into three categories, namely, digital image processing (DIP), machine learning (ML), and deep learning (DL) methods.

DIP methods usually conduct edge detection or pattern recognition of images to identify cracks or defects based on the abnormality of pixel points [19]. The locations of cracks or defects can be identified with advanced 2D signal processing techniques such as various types of transformations [20,21] and filtering methods [22]. In addition, DIP methods can identify the crack width when combined with the Laplacian method [23]. Many environmental or artificial interfering factors such as lighting, background layers, noise, and threshold setting may significantly impact the identified results [24–27]. However, ML algorithms can detect cracks or defects with numerous applications in practice [28–32]. The algorithm can accomplish crack identification and classification integrated with DIP methods, but the generality of algorithms needs further expansion for low-resolution images with background and complex multi-classification problems.

DL methods can effectively extract image characteristics with environmental interference and are especially powerful in processing large-scale training datasets and multi-classification problems compared with DIP and ML methods. Convolutional neural networks (CNNs), a notable method in DL, have garnered widespread utilization in the realms of crack identification due to their superior multi-scale feature extraction, noise resilience, and broad-spectrum recognition capacities [33–35]. Crack identification currently encompasses both object detection [36–38] and semantic segmentation [39–42]. For example, an initial CNN can detect whether there are cracks in pictures, which is a dichotomous classification problem [43]. However, an initial CNN cannot identify crack locations and is disabled to take a further assessment of structures. To address this issue, a multi-layered image preprocessing strategy has been proposed to locate cracks in concrete structures in the measured pictures [44,45]. The cracking region is filled up with a large number of bounding boxes, which are not flexible enough to directly determine the region of cracks [46]. Further, several two-stage identification methods, including regional-based CNNs [47] and faster regional-based CNNs [48], were developed for object detection in images. The crack region can be first identified and then classified based on the above two-stage methods. Additionally, crack width can be identified based on improved mask-RCNN methods [49]. You Only Look Once (YOLO) [50] was recently proposed to accomplish the one-stage identification of location and classification of surface cracks or defects with applications in bridges [51] and pavement [52–54] or tunnel engineering [55] and aircraft structures [56]. The architectures of models such as YOLOv3, other single-stage and dual-stage detection networks, and semantic segmentation frameworks [14,40,41,45] are notably intricate. These models, characterized by their voluminous parameter sets, mandate exhaustive training processes. As a result, ensuring precise identification outcomes demands not only potent computational prowess but also ample memory capacity [51]. A simple network structure is necessary to improve computation speed and reduce memory consumption in order to implement the CNN algorithm on mobile terminals or unmanned aerial vehicles. A series of YOLO-based networks with fewer training coefficients, such as MobileNetv2-YOLOv3 (M2-YOLOv3) [57] and MobileNetv3-YOLOv3

(M3-YOLOv3) [58], were proposed to achieve the purpose of real-time object detection in mobile terminals [59,60]. These networks are more efficient and designed to detect hundreds of different objects; however, there are only tens of concrete crack classes. Therefore, the YOLO algorithm still has room for optimization in concrete crack identification.

To address this deficiency, this study proposes a lightweight YOLOv3, called MDN-YOLOv3, for intelligently identifying the cracks of concrete structures using a self-built training set. The rest of this paper is organized as follows. Section 2 briefly summarizes the structures and imperfections of YOLOv3 in identifying cracks in concrete structures. Section 3 proposes the improved design methods of the backbone network in MDN-YOLOv3. Section 4 compares the performance of the concrete crack identification of MDN-YOLOv3 with existing algorithms. Section 5 compares the test results and the accuracy of the method with existing models. Section 6 concludes and provides remarks on this study.

## 2 Inadequacies of YOLOv3 in Identifying Concrete Cracks

### *2.1 Overview*

Fig. 1 illustrates the structure of YOLOv3, composed primarily of the Darknet-53 backbone and the feature pyramid network (FPN). The main identification process of YOLOv3 can be described as follows:

(1) Input process: The size of input images is adjusted to 416 × 416 × 3 for the feature extraction process.

(2) Feature extraction process: First, the images with 416 × 416 × 3 pixels are input into the backbone network of Darknet-53 [61] for feature extraction, and three effective feature layers with sizes of 13 × 13, 26 × 26, and 52 × 52 are output, as visualized in Fig. 2. The depth of the feature maps starts from the three channels of the RGB image and expands through subsequent layers with 32, 64, 128, 256, 512, and finally 1024 filters. Every layer is responsible for learning more complex features. Then, the three effective feature layers are input into the FPN. Through a series of operations, including continuous convolution operation, up-sampling, down-sampling, and feature fusion from the three effective feature layers, the enhanced feature maps of FPN are output.

(3) Output process: The output feature maps of FPN are used to predict three bounding boxes in every grid cell. The coordinate of the bounding box is shown in Fig. 3. Every bounding box predicts three key parameters based on extracted and multi-scale features: (i) Coordinate the dataset of the rectangular box, including horizontal coordinates $t_x$ and vertical coordinates $t_y$ of the centroid, width $t_w$, and height $t_h$ of the bounding box; (ii) Classification of detected objects; (iii) Confidence of predicted results.

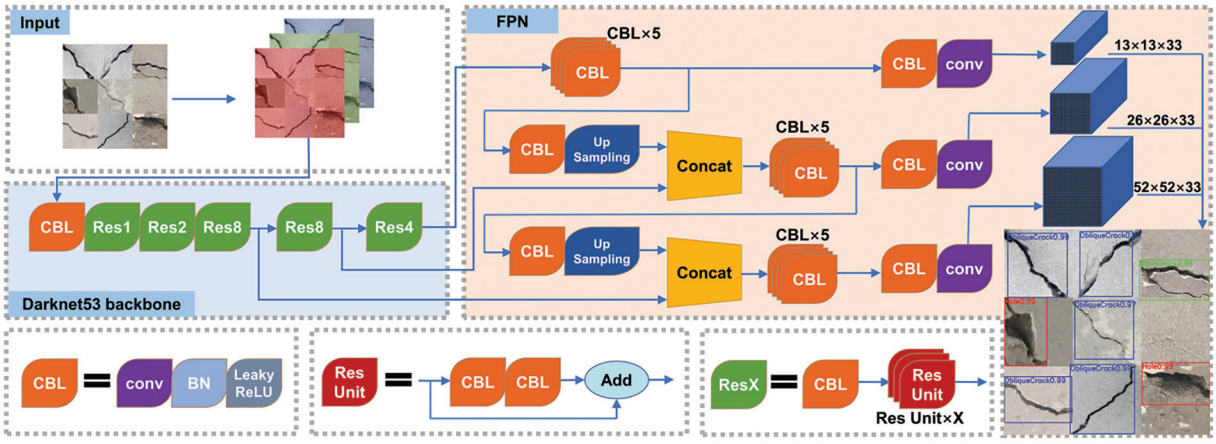


**Figure 1:** The structure of YOLOv3

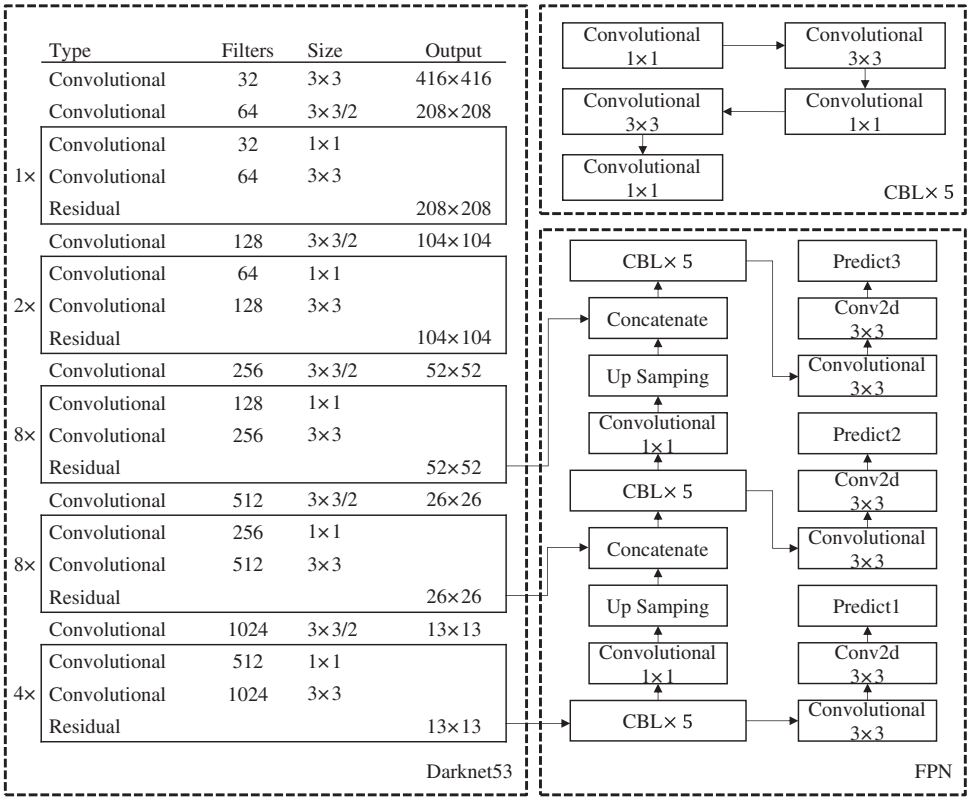


**Figure 2:** Network details of YOLOv3. Convolutional $k \times k$ = conv2d $k \times k$ + Nonlinear

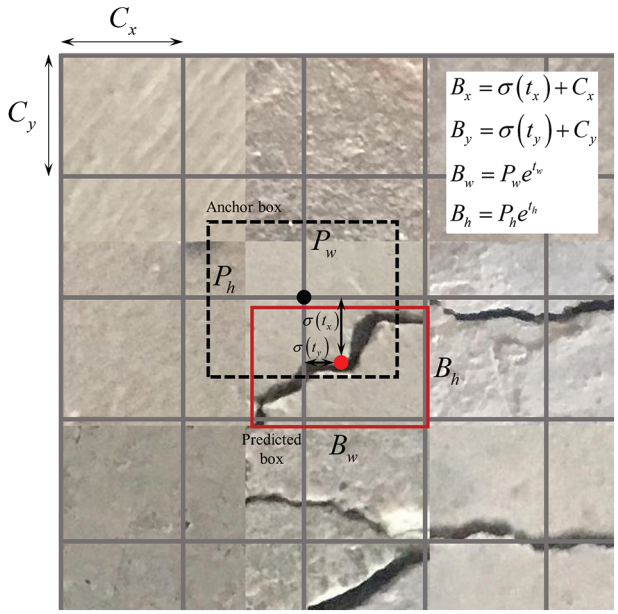


**Figure 3:** Anchor box to predict box process

The loss function can be calculated as follows:

$$\begin{aligned} Loss = {} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[ \left(x_i - \hat{x}_i^j\right)^2 + \left(y_i - \hat{y}_i^j\right)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[ \left( \sqrt{w_i^j} - \sqrt{\hat{w}_i^j} \right)^2 + \left( \sqrt{h_i^j} - \sqrt{\hat{h}_i^j} \right)^2 \right] \\ & - \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[ \hat{C}_i^j \log\left(C_i^j\right) + \left(1 - \hat{C}_i^j\right) \log\left(1 - C_i^j\right) \right] \\ & - \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{noobj} \left[ \hat{C}_i^j \log\left(C_i^j\right) + \left(1 - \hat{C}_i^j\right) \log\left(1 - C_i^j\right) \right] \\ & - \sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{c \in \text{classes}} \left( \left[ \hat{P}_i^j \log\left(P_i^j\right) + \left(1 - \hat{P}_i^j\right) \log\left(1 - P_i^j\right) \right] \right. \end{aligned} \tag{1}$$

where $\lambda_{\text{coord}}$ and $\lambda_{\text{noobj}}$ are weight coefficients and remain unchanged in the loss function. $I_{ij}^{obj}$ is the responsible anchor box that has the highest intersection over union (IoU) with the ground-truth box, which is actually used for predictions in the loss function. $I_{ij}^{noobj}$ represents the complementary set of responsible anchor boxes $I_{ij}^{obj}$. $S$ is the size of grid cells, and $B$ is the number of anchor boxes. $C_i^j$ and $P_i^j$ are the confidence and probability of belonging to a specific category, respectively. The superscript represents the prediction center of the corresponding parameters.

The IoU can be calculated as follows:

$$IoU(A,B) = \frac{Area_A \cap Area_B}{Area_A \cup Area_B} = \frac{Area_{Inter}}{Area_A + Area_B - Area_{Inter}} \tag{2}$$

where $IOU(A,B)$ is the intersection over the union of the ground-truth boxes and anchor boxes.

The loss function serves to adjust the generated anchor box. It is primarily composed of three components: the loss associated with the box's coordinates, the confidence, and the category. By minimizing this loss function, YOLOv3 can accurately pinpoint the location of objects in every output image.

### *2.2 Inadequacies*

YOLOv3 can detect multiple targets in a single forward pass, simultaneously predicting class scores and bounding boxes, which streamlines the object detection process. As a result, YOLOv3 is efficient and has found applications in various domains [50,61,62]. There are a series of CNNs used as the backbone networks of YOLO with fewer training coefficients, such as MobileNetv2 and MobileNetv3; however, the backbone networks in YOLO still do not meet the demand for swift crack identification with high accuracy. This is because the CNNs serving as the backbone network of YOLO in concrete crack identification are made for identifying hundreds of classes, whereas there are only tens of classes in concrete cracks. Therefore, the series of universal YOLOv3 models are large in size and time-consuming in identifying concrete cracks. In other words, the series of universal YOLOv3 models is relatively complex and unsuitable for concrete crack identification [14,63,64]. To this end, a tailored deep-learning algorithm termed MDN-YOLOv3 is proposed.

## 3 MDN-YOLOv3

### 3.1 Retrofit Techniques

#### 3.1.1 Depthwise Separable Convolution (DSC)

DSC was proposed by Sandler et al. [60], to improve the efficiency of a CNN [65]. Fig. 4 illustrates the difference between DSC and standard convolution. The parameter $C_{std}$ of the standard convolution is calculated as follows:

$$C_{std} = 3 \times 3 \times C_1 \times C_2 \tag{3}$$

where $C_1$ is the number of input channels and $C_2$ is the number of output channels of the structure, as shown in Fig. 4a.

As shown in Fig. 4b, the parameter $C_{dpt}$ of the depthwise convolution is calculated as follows:

$$C_{dpt} = 3 \times 3 \times C_1 + 1 \times 1 \times C_1 \times C_2 \tag{4}$$

The ratio of standard convolution parameters to those of DSC is as follows:

$$\frac{C_{dpt}}{C_{std}} = \frac{3 \times 3 \times C_1 + 1 \times 1 \times C_1 \times C_2}{3 \times 3 \times C_1 \times C_2} = \frac{1}{C_2} + \frac{1}{9} \tag{5}$$

As shown in Eq. (5), the parameter of DSC is 90% smaller compared with that of standard convolution. Therefore, the use of DSC instead of standard convolution in YOLOv3 can effectively improve efficiency.

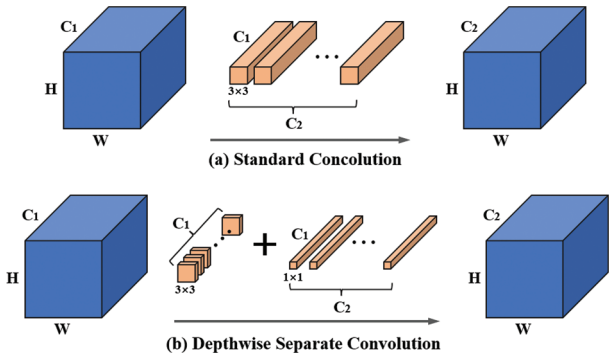


**Figure 4:** The difference between DSC and standard convolution

#### 3.1.2 Multi-Scale Feature Fusion by Dilated-Down Sampling (DDS)

The diagram of DDS is shown in Fig. 5, which is the main component of the backbone network in MDN-YOLOv3. As depicted in Fig. 5, a series of dilated convolution blocks are utilized in the network. There are three steps for DDS blocks to extract features. First, dilated convolution blocks with a stride of two and different dilation rates of one, two, and three are used to extract multi-scale features from the feature maps. Second, the dilated convolution block is combined using the concatenate operation to fuse features on different receptive scales. Finally, a 1 × 1 convolution layer is used to integrate every channel of the feature maps. The output of DDS contains features of various receptive fields to describe concrete cracks.

#### 3.1.3 Attention Module of CBAM

The attention module of the Convolutional Block Attention Module (CBAM) is introduced into the backbone network, and the structure of this module is shown in Fig. 6. CBAM compresses feature maps using max-pooling (MaxPool) along with average-pooling (AvgPool) for channel and spatial attention.

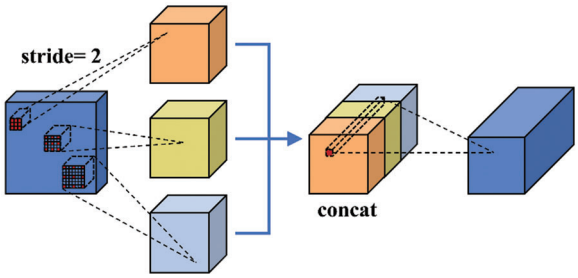


**Figure 5:** Block of the DDS

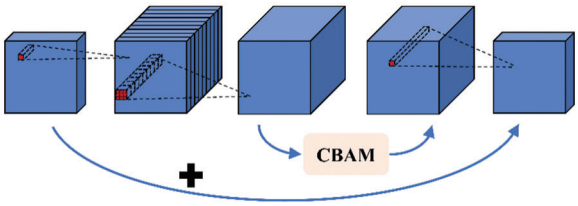


**Figure 6:** The CBAM in bottleneck

*3.1.4 Optimization of the Predictive Layer of the Backbone Module with a K-Means Cluster*

The classification of the anchor box is a key parameter in YOLOv3. To verify the best configuration of the anchor box in concrete crack detection, the K-means clustering algorithm is used to cluster the dataset of the concrete crack. The dataset of the concrete crack in this study contains 6066 figures with seven categories, including no crack, hole, mesh crack, oblique crack, transverse crack, vertical crack, and irregular crack. Average IoU is an index used to reflect the effect of clustering between anchors and real target boxes. For the dataset of crack detection in this study, the relationship between the average IoU and the clustering number K is shown in Fig. 7. The clustering number K = 4 is chosen because the average IoU has not changed significantly with the increase in clusters when K > 4 and the point after the great slope can be regarded as the optimal point. The K-means clustering results of the training set are shown in Fig. 8. The position of the four hollow marks is the center of the corresponding cluster, including (47, 137), (128, 133), (136, 89), and (137, 43). Thus, the anchor boxes of the MDN-YOLOv3 method are (47, 137), (128, 133), (136, 89), and (137, 43) for the concrete crack detection, while those in YOLOv3 are (35, 137), (51, 137), (83, 135), (109, 135) (136, 78), (136, 132), (136, 100), (137, 36), and (137, 54). In addition, the predictive layers of MDN-YOLOv3 are 8-fold and 16-fold downsampling layers, while the predictive layers are 8, 16, and 32-fold in YOLOv3. The optimization of the anchor configuration simplifies the prediction layer of the backbone module from three scales to two scales, resulting in higher efficiency compared with M3-YOLOv3.

### *3.2 Structure of MDN-YOLOv3*

Combining the superiority of DSC and DDS and the attention mechanism of CBAM, the multi-dilated network (MDN) is proposed. MDN is used as the backbone network of YOLOv3, which can be called MDN-YOLOv3. Furthermore, the number of effective feature layers is streamlined from three to two. The whole structure of the MDN-YOLOv3 is shown in Figs. 9 and 10. The main structures can be divided into three modules: an MDN backbone, an FPN, and a prediction module.

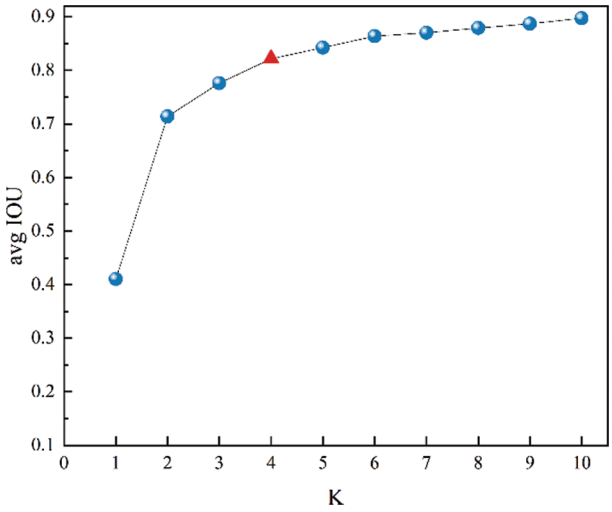


**Figure 7:** Clustering result of concrete-crack dataset

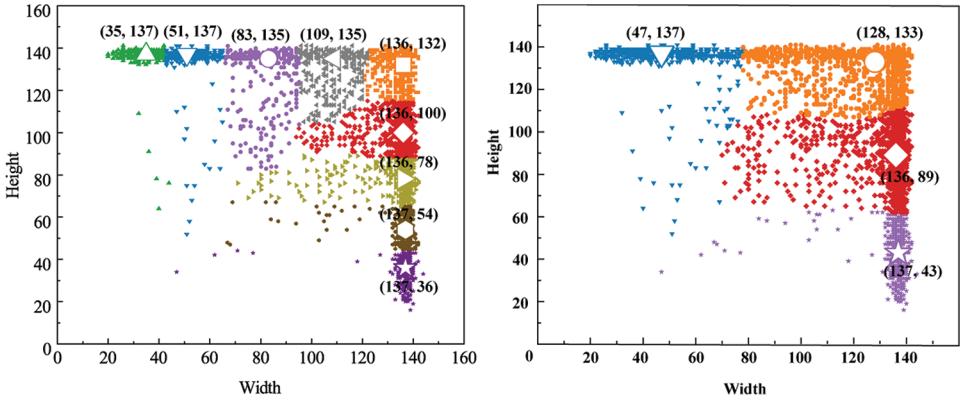


**Figure 8:** Clustering results of nine ground-truth boxes. (a) YOLOv3 (b) SciYOLOv3

The MDN serves as the backbone network for extracting multi-scale features. Specifically, crosswise MDN structures are added between bottleneck modules to extract and fuse multi-scale features from input images. In addition, the CBAM attention mechanism is introduced to enhance feature extraction ability. The output matrix dimensions of the effective feature layers in the MDN backbone are 1/16 and 1/8 of the original image size, resulting in feature maps of 52 × 52 and 26 × 26 pixels being output. These output feature maps can be used for prediction (see Fig. 9a). The resulting output data are a matrix with eleven rows and one column that includes four coordinate parameters, six classification parameters, and one confidence parameter.

## 4 Splicing Concrete-Crack Dataset (SCCD)

The existing crack datasets of concrete structures often focus on a single crack or several adjacent cracks. These crack patterns tend to be somewhat repetitive and are not suitable for identifying intricate patterns in real-world situations. To address this limitation, we develop a new dataset named the SCCD to capture more diverse and complex crack patterns. Overall, 6066 figures are collected and divided into seven categories: no crack, hole, mesh crack, oblique crack, transverse crack, vertical crack, and irregular crack, as shown in

Figs. 11a–11g. Moreover, every nine images are spliced as a new image to increase the complexity of crack patterns, as shown in Fig. 11h. Various types of crack models can be generated randomly at different locations in splicing images with variable backgrounds. The new 674 general splicing images are further divided into two parts: 500 images are used as the training set and 174 images are used as the test set.

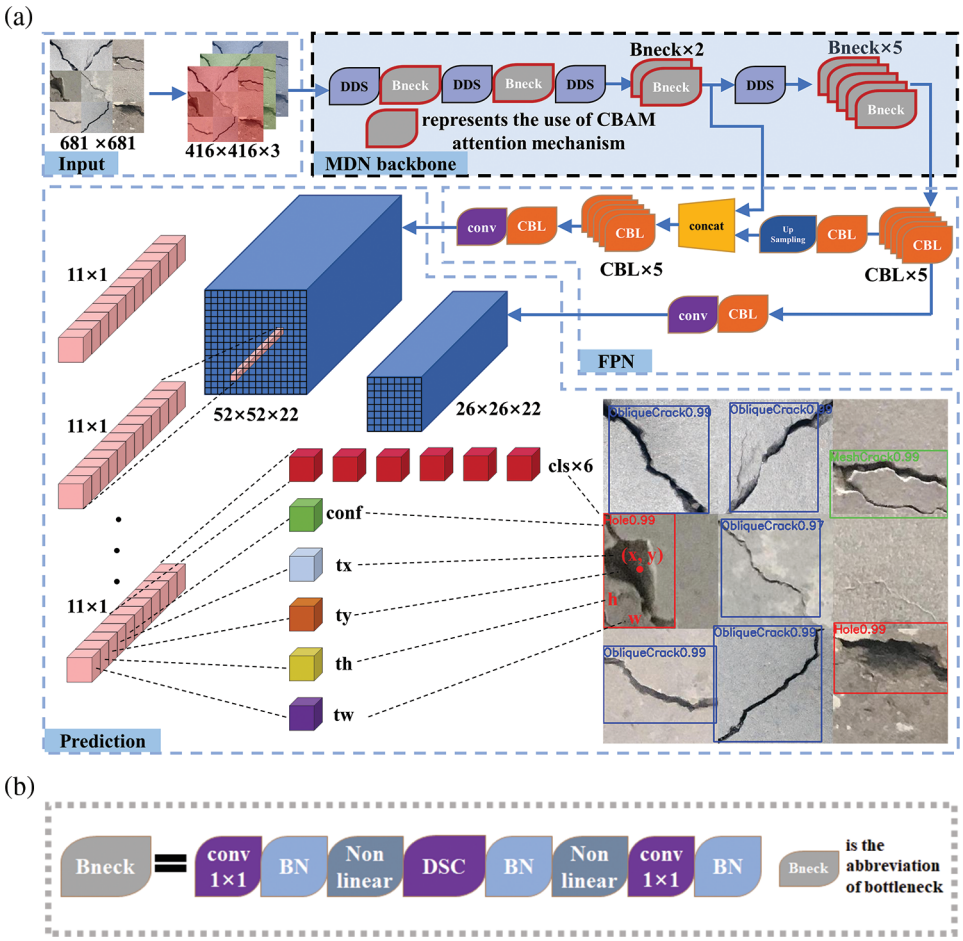


**Figure 9:** (a) The structure of MDN-YOLOv3. (b) The structure of the bottleneck

| Input | Operator | exp size | out | CBAM | NL |
|---|---|---|---|---|---|
| 416×416×3 | dilated downsampling | 5×3 | 16 | - | HS |
| 208×258×16 | bneck, 3×3 | 16 | 16 | √ | RE |
| 208×258×16 | dilated downsampling | 16×3 | 24 | - | - |
| 104×104×24 | bneck, 3×3 | 72 | 24 | √ | RE |
| 104×104×24 | dilated downsampling | 18×3 | 40 | - | - |
| 52×52×40 | bneck, 5×5 | 120 | 40 | √ | HS |
| 52×52×40 | bneck, 5×5 | 120 | 40 | √ | HS |
| 52×52×40 | dilated downsampling | 30×3 | 80 | - | - |
| 26×26×80 | bneck, 5×5 | 200 | 80 | √ | HS |
| 26×26×80 | bneck, 5×5 | 184 | 80 | √ | HS |
| 26×26×80 | bneck, 5×5 | 184 | 80 | √ | HS |
| 26×26×80 | bneck, 5×5 | 480 | 112 | √ | HS |
| 26×26×112 | bneck, 5×5 | 672 | 112 | √ | HS |

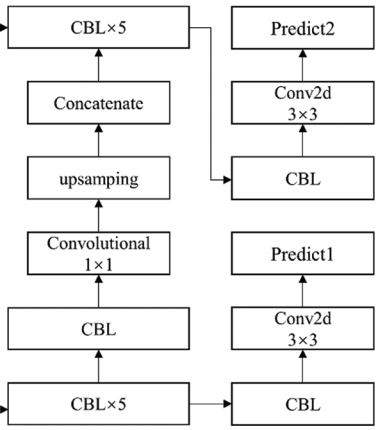


**Figure 10:** Network details of MDN-YOLOv3

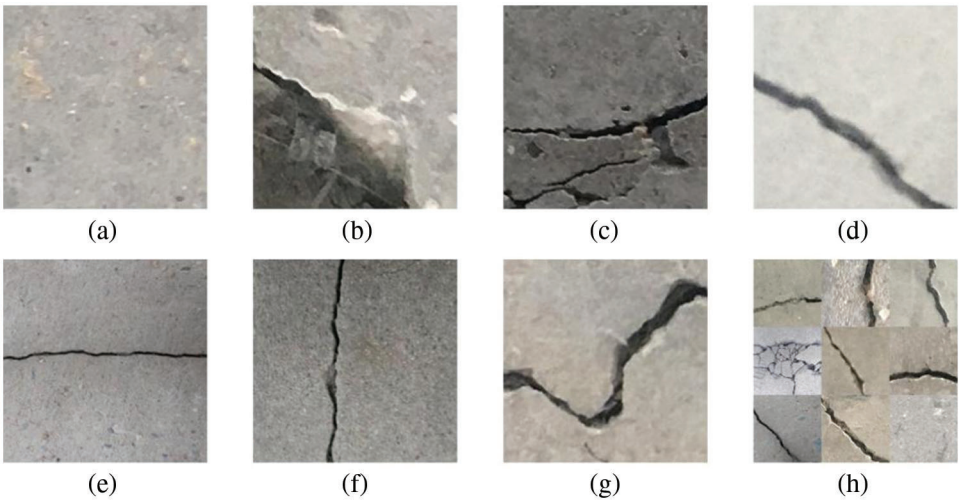


**Figure 11:** Concrete surface detect dataset. (a) no crack, (b) hole, (c) mesh crack, (d) oblique crack, (e) transverse crack, (f) vertical crack, (g) irregular crack, (h) splicing picture

## 5 Superiorities of MDN-YOLOv3

### *5.1 Training Environment*

This experiment is conducted on a Windows 10 system using PyTorch. The parameter settings for the training experiments on this platform are presented in Table 1.

**Table 1:** Experimental platform configuration

| Attribute | Value |
|---|---|
| Operating system | Windows 10 |
| CPU | Intel(R) Xeon(R) Gold 5222 CPU @ 3.80 GHz 3.79 GHz |
| GPU | NVIDIA Quadro P2200 |
| RAM | 64.0 GB |
| Programming environment | Anaconda3 |
| | CUDA10.2 |
| | Python3.6 |
| | PyTorch |

### *5.2 Training Parameters*

When the MDN-YOLOv3 is trained, the learning rate is set to 0.001 and kept constant throughout the training process. The epoch, batch size, and weight decay numbers are set to 50, 4, and 0.0005, respectively.

### *5.3 Evaluate Indicators*

The recall rate [66] and prediction rate [67] are calculated to evaluate the identification results of the MDN-YOLOv3. The definitions of these two parameters are

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

where $TP$ represents the number of positive samples with a correct prediction, $FP$ represents the number of wrong predictions, and $FN$ represents the number of positive samples with a failed prediction.

In addition, two indexes, the mean average precision ($mAP$) at an IoU threshold of 0.5 and the frames per second ($FPS$), are calculated to quantitatively evaluate the calculation accuracy and speed of the network, respectively. The definition of $mAP$ can be expressed as

$$mAP = \frac{\sum_{i \in N} AP_i}{N} \tag{8}$$

$$AP_i = \int_0^1 P_i(R_i) dR_i \tag{9}$$

where $AP_i$ is the average accuracy of the i-th crack pattern. Generally, a higher $AP$ value shows better classification results for a model.

The $FPS$ represents the identification speed and can be expressed as

$$FPS = \frac{N_{image}}{Time} \tag{10}$$

where $N_{image}$ is the number of images detected, and $Time$ is the sum time of the detected images.

### 5.4 Comparison with Existing Typical Models

Table 2 displays the structures utilized in every model. Fig. 12 illustrates the $P$–$R$ curves of six concrete cracks for various models, and their $AP$ is calculated using Eq. (9) and summarized in Table 3. The highest $AP$ among all of the compared models is observed for the hole, irregular crack, oblique crack, and transverse crack of MDN-YOLOv3; however, the mesh crack's $AP$ is slightly lower than that of YOLOv3 and M3-YOLOv3, while the vertical crack's $AP$ is a bit lower than that of YOLOv3, as shown in Table 3. The $mAP$ of the models is determined by calculating the respective $APs$, as depicted in Fig. 13. While MDN-YOLOv3 and YOLOv3 have the highest $mAP$ values, M2-YOLOv3 has the lowest value among them all.

**Table 2:** Comparison of model structures used in every model

| Structure | YOLOv3 | M2-YOLOv3 | M3-YOLOv3 | MDN-YOLOv3 |
|---|---|---|---|---|
| Backbone | Darknet-53 | MobileNetv2 | MobileNetv3 | MDN |
| Bottleneck | N/A | √ | √ | √ |
| DSC | N/A | √ | √ | √ |
| Attention mechanism | N/A | N/A | √ (SE) | √ (CBAM) |
| DDS | N/A | N/A | N/A | √ |

Table 4 summarizes the evolution indicators of $mAP$ and $FPS$. The MDN-YOLOv3 has a $mAP$ of 70.96%, which is almost identical to YOLOv3's 71.27%. This value is significantly higher than that of M2-YOLOv3 and M3-YOLOv3. In addition, the $FPS$ of MDN-YOLOv3 is 40.73 (f/s), making it faster by 41.5%, 8.25%, and 1.92% compared with YOLOv3, M2-YOLOv3, and M3-YOLOv3, respectively. Moreover, the model size of MDN-YOLOv3 is only 6.92 (MB), which makes it smaller by significant

margins of 97.2%, 27.8%, and 57.7% when compared with the YOLOv3, M2-YOLOv3, and M3-YOLOv3 models, respectively.

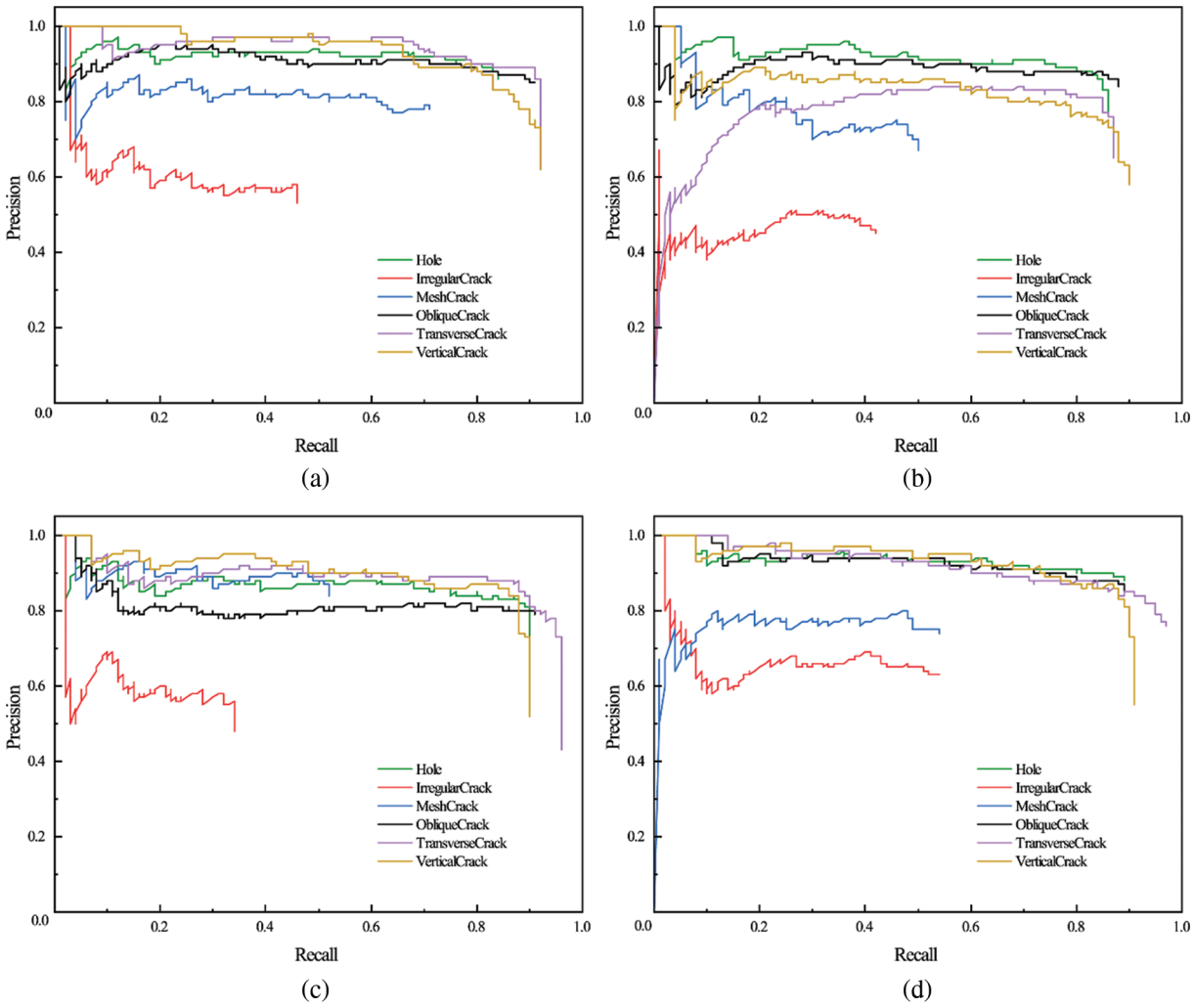


**Figure 12:** *P–R* curves of six concrete cracks for different models: (a) YOLOv3, (b) M2-YOLOv3, (c) M3-YOLOv3, (d) MDN-YOLOv3

**Table 3:** *AP* for every crack type identified by the models

| Crack type | AP (%) | | | |
|---|---|---|---|---|
| | YOLOv3 | M2-YOLOv3 | M3-YOLOv3 | MDN-YOLOv3 |
| Hole | 79.02 | 80.03 | 79.47 | 84.24 |
| Irregular crack | 29.95 | 21.23 | 21.74 | 37.63 |
| Mesh crack | 59.59 | 40.54 | 48.14 | 42.67 |
| Oblique crack | 83.51 | 79.71 | 75.90 | 85.43 |
| Transverse crack | 88.43 | 72.88 | 87.05 | 89.65 |
| Vertical crack | 87.13 | 76.15 | 82.62 | 86.12 |

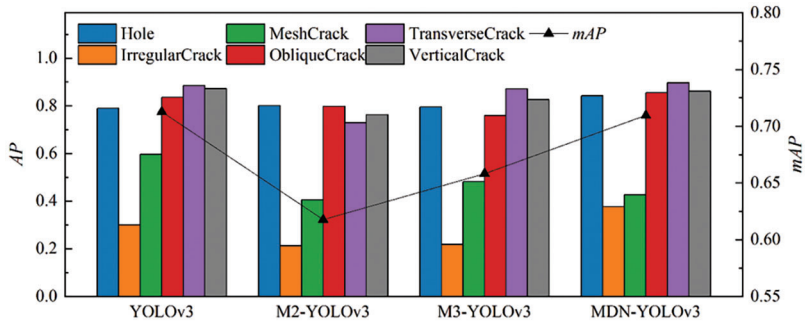


**Figure 13:** *AP* and *mAP* value comparison of different models

**Table 4:** Evaluate indicators of identification of the models

| Evaluate indicators | YOLOv3 | M2-YOLOv3 | M3-YOLOv3 | MDN- YOLOv3 |
|---|---|---|---|---|
| *mAP* (%) | 71.27 | 61.76 | 65.82 | 70.96 |
| FPS (f/s) | 23.81 | 37.37 | 39.96 | 40.73 |
| Parameter size (MB) | 246.42 | 9.58 | 16.35 | 6.92 |

To illustrate every model's performance vividly, a bubble diagram has been made in Fig. 14 where the bubble size represents every model's size, the abscissa shows the *mAP*, and the ordinate represents the corresponding *FPS*. As shown in Fig. 14, the bubble of MDN-YOLOv3 is the smallest and is located in the top right of the diagram. In other words, the MDN-YOLOv3 has the smallest model size and the fastest identification speed. Furthermore, the recognition accuracy of MDN-YOLOv3 is near that of M3-YOLOv3 and better than that of other comparison models.

In addition, the cracks in the test set are detected and classified with corresponding confidence, as shown in Fig. 15. The location and category of the cracks are clearly identified using the improved MDN-YOLOv3 method.

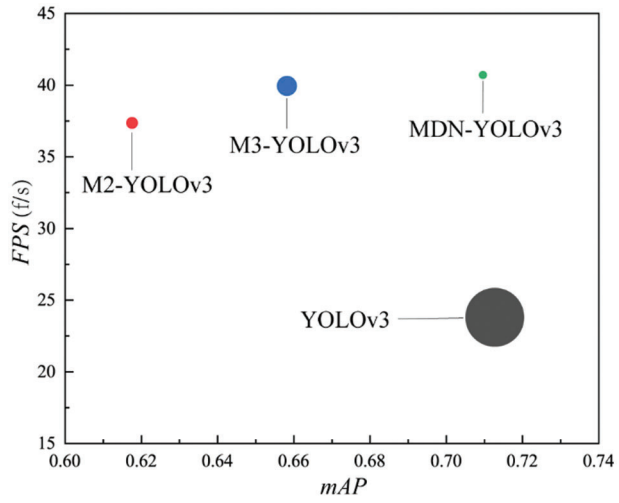


**Figure 14:** Comparison of *mAP*, *FPS*, and model size

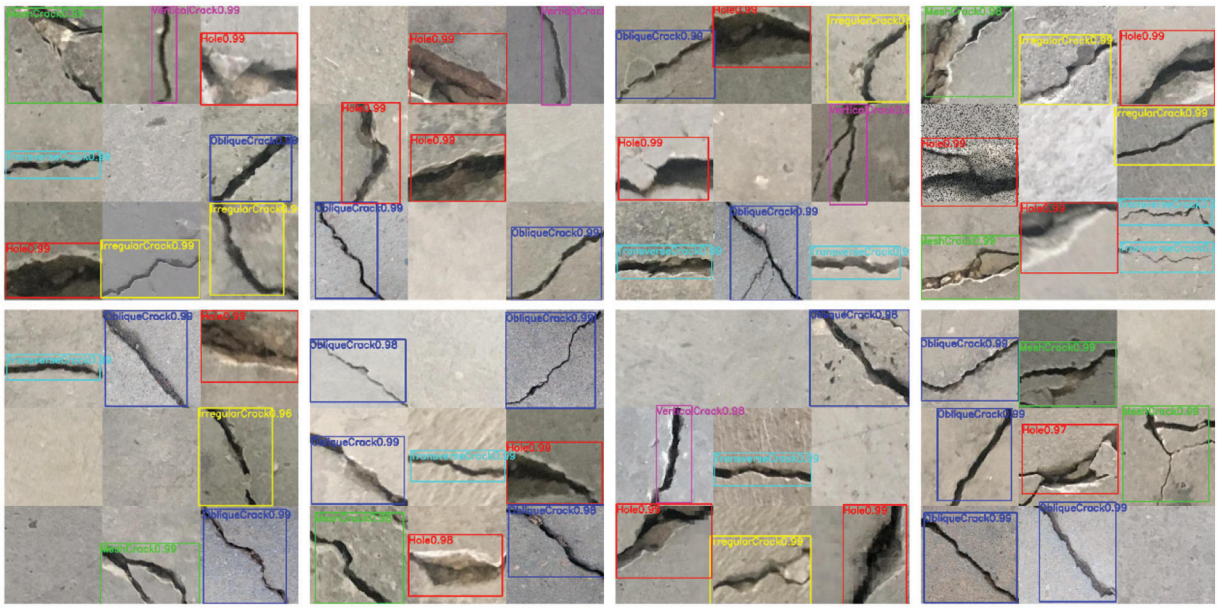

**Figure 15:** Crack detection results using MDN-YOLOv3

In summary, the proposed MDN-YOLOv3 identifies cracks quickly while maintaining high accuracy levels despite its small model size.

## 6 Conclusions

The identification of crack locations and patterns is of great significance in evaluating the service performance of concrete structures. To address this issue, this study proposes an improved YOLOv3 algorithm for the rapid and accurate identification of the surface cracks of concrete structures. The proposed model is verified using a self-built training set. The main conclusions are presented as follows:

(1) The proposed MDN-YOLOv3 has lightweight network structures and robust feature extraction capability. The DSC and proposed DDS structure can effectively extract and fuse the features of cracks.

(2) The proposed MDN-YOLOv3 has a model size of 6.92 M and an identification speed of 40.73 FPS, which is 97.2% smaller in size and 41.5% faster in speed than YOLOv3. This finding indicates that the proposed MDN-YOLOv3 is more suitable for applications on mobile devices due to the faster calculation speed and smaller memory of these devices.

In the future, a better crack identification scheme should be studied to evaluate the applications of proposed models, such as temperature cracks, concrete shrinkage cracks, and external load cracks. In addition, concrete crack images from complex scenes such as underwater concrete and tunnel concrete require further research and development.

**Acknowledgement:** None.

**Funding Statement:** The authors are grateful for the International Science & Technology Cooperation Project of Jiangsu Province (BZ2022010), the Jiangsu-Czech Bilateral Co-Funding R&D Project (No. BZ2023011), and "The Belt and Road" Innovative Talents Exchange Foreign Experts Project (No. DL2023019001L).

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: Maosen Cao, Dragoslav Sumarac; data collection: Alfred Strauss, Yehui Shi; analysis and

interpretation of results: Haoan Gu, Kai Zhu, Alfred Strauss; draft manuscript preparation: Haoan Gu, Maosen Cao. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The data that support the findings of this study are available from the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References


1. Srikanth, I., Arockiasamy, M. (2020). Deterioration models for prediction of remaining useful life of timber and concrete bridges: A review. *Journal of Traffic and Transportation Engineering (English Edition), 7(2),* 152–173. https://doi.org/10.1016/j.jtte.2019.09.005
2. Matsuoka, K., Tokunaga, M., Kaito, K. (2021). Bayesian estimation of instantaneous frequency reduction on cracked concrete railway bridges under high-speed train passage. *Mechanical Systems and Signal Processing, 161(9),* 107944. https://doi.org/10.1016/j.ymssp.2021.107944
3. Ren, J., Deng, S., Wei, K., Li, H. L., Wang, J. (2019). Mechanical property deterioration of the prefabricated concrete slab in mixed passenger and freight railway tracks. *Construction and Building Materials, 208(3),* 622–637. https://doi.org/10.1016/j.conbuildmat.2019.03.039
4. Dung, C. V., Le, D. A. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction, 99(4),* 52–58. https://doi.org/10.1016/j.autcon.2018.11.028
5. Zhong, X., Peng, X., Yan, S., Shen, M. Y., Zhai, Y. Y. (2018). Assessment of the feasibility of detecting concrete cracks in images acquired by unmanned aerial vehicles. *Automation in Construction, 89(1),* 49–57. https://doi.org/10.1016/j.autcon.2018.01.005
6. Liu, Z., Cao, Y., Wang, Y., Wang, W. (2019). Computer vision-based concrete crack detection using U-net fully convolutional networks. *Automation in Construction, 104,* 129–139. https://doi.org/10.1016/j.autcon.2019.04.005
7. Zerwer, A., Polak, M. A., Santamarina, J. C. (2005). Detection of surface breaking cracks in concrete members using Rayleigh waves. *Journal of Environmental & Engineering Geophysics, 10(3),* 295–306. https://doi.org/10.2113/JEEG10.3.295
8. Aghajanzadeh, S. M., Mirzabozorg, H. (2019). Concrete fracture process modeling by combination of extended finite element method and smeared crack approach. *Theoretical and Applied Fracture Mechanics, 101(4),* 306–319. https://doi.org/10.1016/j.tafmec.2019.03.012
9. Abdelkhalek, S., Zayed, T. (2020). Comprehensive inspection system for concrete bridge deck application: Current situation and future needs. *Journal of Performance of Constructed Facilities, 34(5),* 03120001. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001484
10. Wang, L. (2023). Automatic detection of concrete cracks from images using Adam-SqueezeNet deep learning model. *Frattura ed Integrità Strutturale, 17(65),* 289–299. https://doi.org/10.3221/IGF-ESIS.65.19
11. Ohtsu, M., Shigeishi, M., Iwase, H., Koyanagit, W. (1991). Determination of crack location, type and orientation ina concrete structures by acoustic emission. *Magazine of Concrete Research, 43(155),* 127–134. https://doi.org/10.1680/macr.1991.43.155.127
12. Goszczyńska, B., Świt, G., Trąmpczyński, W., Krampikowska, A., Tworzewska, J. et al. (2012). Experimental validation of concrete crack identification and location with acoustic emission method. *Archives of Civil and Mechanical Engineering, 12(1),* 23–28. https://doi.org/10.1016/j.acme.2012.03.004
13. Kim, H., Ahn, E., Shin, M., Sim, S. H. (2019). Crack and noncrack classification from concrete surface images using machine learning. *Structural Health Monitoring, 18(3),* 725–738. https://doi.org/10.1177/1475921718768747
14. Deng, J., Lu, Y., Lee, V. C. S. (2020). Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering, 35(4),* 373–388. https://doi.org/10.1111/mice.12497

15. Murao, S., Nomura, Y., Furuta, H., Kim, C. W. (2019). Concrete crack detection using UAV and deep learning. *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering*, Seoul, South Korea, ICASP.
16. Farhidzadeh, A., Dehghan-Niri, E., Salamone, S., Luna, B., Whittaker, A. (2013). Monitoring crack propagation in reinforced concrete shear walls by acoustic emission. *Journal of Structural Engineering, 139(12),* 04013010. https://doi.org/10.1061/(ASCE)ST.1943-541X.0000781
17. IAEA (2022). *Guidebook on non-destructive testing of concrete structures*. Vienna: International Atomic Energy Agency.
18. Karthick, S. P., Muralidharan, S., Saraswathy, V., Kwon, S. J. (2016). Effect of different alkali salt additions on concrete durability property. *Journal of Structural Integrity and Maintenance, 1(1),* 35–42. https://doi.org/10.1080/24705314.2016.1153338
19. Liu, Y., Cho, S., Spencer Jr, B. F., Fan, J. (2014). Automated assessment of cracks on concrete surfaces using adaptive digital image processing. *Smart Structures and Systems, 14(4),* 719–741. https://doi.org/10.12989/sss.2014.14.4.719
20. Jahanshahi, M. R., Kelly, J. S., Masri, S. F., Sukhatme, G. S. (2009). A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures. *Structure and Infrastructure Engineering, 5(6),* 455–486. https://doi.org/10.1080/15732470801945930
21. Feng, D., Feng, M. Q. (2018). Computer vision for SHM of civil infrastructure: From dynamic response measurement to damage detection–A review. *Engineering Structures, 156(12),* 105–117. https://doi.org/10.1016/j.engstruct.2017.11.018
22. 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. https://doi.org/10.1016/j.aei.2015.01.008
23. Wang, W. J., Zhang, A., Wang, K. C., Braham, A. F., Qiu, S. (2018). Pavement crack width measurement based on Laplace's equation for continuity and unambiguity. *Computer-Aided Civil and Infrastructure Engineering, 33(2),* 110–123. https://doi.org/10.1111/mice.12319
24. Yu, S. N., Jang, J. H., Han, C. S. (2007). Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. *Automation in Construction, 16(3),* 255–261. https://doi.org/10.1016/j.autcon.2006.05.003
25. Zhu, Z., German, S., Brilakis, I. (2011). Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation. *Automation in Construction, 20(7),* 874–883. https://doi.org/10.1016/j.autcon.2011.03.004
26. Lattanzi, D., Miller, G. R. (2014). Robust automated concrete damage detection algorithms for field applications. *Journal of Computing in Civil Engineering, 28(2),* 253–262. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000257
27. Yamaguchi, T., Hashimoto, S. (2010). Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Machine Vision and Applications, 21(5),* 797–809. https://doi.org/10.1007/s00138-009-0189-8
28. Moon, H. G., Kim, J. H. (2011). Intelligent crack detecting algorithm on the concrete crack image using neural network. *Proceedings of the 28th ISARC*, pp. 1461–1467. Seoul, South Korea. https://doi.org/10.22260/ISARC2011/0279
29. Zhang, W., Zhang, Z., Qi, D., Liu, Y. (2014). Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors, 14(10),* 19307–19328. https://doi.org/10.3390/s141019307
30. Prasanna, P., Dana, K. J., Gucunski, N., Basily, B. B., La, H. M. et al. (2016). Automated crack detection on concrete bridges. *IEEE Transactions on Automation Science and Engineering, 13(2),* 591–599. https://doi.org/10.1109/TASE.2014.2354314
31. Shi, Y., Cui, L., Qi, Z., Meng, F., Chen, Z. (2016). Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems, 17(12),* 3434–3445. https://doi.org/10.1109/TITS.2016.2552248

32. Li, G., Zhao, X., Du, K., Ru, F., Zhang, Y. (2017). Recognition and evaluation of bridge cracks with modified active contour model and greedy search-based support vector machine. *Automation in Construction, 78(4),* 51–61. https://doi.org/10.1016/j.autcon.2017.01.019
33. Lin, Y. Z, Nie, Z. H, Ma, H. W (2017). Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering, 32(12),* 1025–1046. https://doi.org/10.1111/mice.12313
34. Zhou, S., Song, W. (2020). Deep learning-based roadway crack classification using laser-scanned range images: A comparative study on hyperparameter selection. *Automation in Construction, 114(2),* 103171. https://doi.org/10.1016/j.autcon.2020.103171
35. Zhang, A., Wang, K. C. P., Fei, Y., Liu, Y., Chen, C. et al. (2019). Automated pixel-level pavement crack detection on 3D asphalt surfaces with a recurrent neural network. *Computer-Aided Civil and Infrastructure Engineering, 34(3),* 213–229. https://doi.org/10.1111/mice.12409
36. Nhat-Duc, H., 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(1),* 203–213. https://doi.org/10.1016/j.autcon.2018.07.008
37. Gupta, R., Goodman, B., Patel, N., Hosfelt, R., Sajeev, S. et al. (2019). Creating xBD: A dataset for assessing building damage from satellite imagery. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 10–17. Los Angeles, CA, USA. https://doi.org/10.48550/arXiv.1911.09296
38. Xu, J. Z., Lu, W., Li, Z., Khaitan, P., Zaytseva, V. (2019). Building damage detection in satellite imagery using convolutional neural networks. https://doi.org/10.48550/arXiv.1910.06444
39. Dong, C. Z., Catbas, F. N. (2021). A review of computer vision-based structural health monitoring at local and global levels. *Structural Health Monitoring, 20(2),* 692–743. https://doi.org/10.1177/1475921720935585
40. Ding, W., Yang, H., Yu, K., Zhang, J. W., Shao, Y. et al. (2023). Crack detection and quantification for concrete structures using UAV and transformer. *Automation in Construction, 152,* 104929. https://doi.org/10.1016/j.autcon.2023.104929
41. Zhao, W., Liu, Y., Zhang, J., Shao, Y., Shu, J. (2022). Automatic pixel-level crack detection and evaluation of concrete structures using deep learning. *Structural Control and Health Monitoring, 29(8),* e2981. https://doi.org/10.1002/stc.2981
42. Kang, D., Benipal, S. S., Gopal, D. L., Shao, Y., Cha, Y. J. (2020). Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning. *Automation in Construction, 118(4),* 103291. https://doi.org/10.1016/j.autcon.2020.103291
43. Cha, Y. J., Choi, W. (2017). Vision-based concrete crack detection using a convolutional neural network. *Conference Proceedings of the Society for Experimental Mechanics Series*, pp. 71–73. Rhodes, Greece. https://doi.org/10.1007/978-3-319-54777-0_9
44. Fu, R. H., Xu, H., Wang, Z. J., Shen, L., Cao, M. S. et al. (2020). Enhanced intelligent identification of concrete cracks using multi-layered image preprocessing-aided convolutional neural networks. *Sensors, 20(7),* 2021. https://doi.org/10.3390/s20072021
45. Shu, J., Ding, W., Zhang, J., Lin, F., Duan, Y. (2022). Continual-learning-based framework for structural damage recognition. *Structural Control and Health Monitoring, 29(11),* e3093. https://doi.org/10.1002/stc.3093
46. Bao, Y., Chen, Z., Wei, S., Xu, Y., Tang, Z. et al. (2019). The state of the art of data science and engineering in structural health monitoring. *Engineering, 5(2),* 234–242. https://doi.org/10.1016/j.eng.2018.11.027
47. Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Vision*, pp. 1440–1448. Santiago, Chile. https://doi.org/10.48550/arXiv.1504.08083
48. Ren, S. Q., He, K. M., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems, 28*. https://doi.org/10.48550/arXiv.1506.01497
49. Guo, P. W., Meng, W. N., Bao, Y. (2021). Automatic identification and quantification of dense microcracks in high-performance fiber-reinforced cementitious composites through deep learning-based computer vision. *Cement and Concrete Research, 148(1),* 106532. https://doi.org/10.1016/j.cemconres.2021.106532

50. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788. Las Vegas, USA. https://doi.org/10.48550/arXiv.1506.02640
51. Park, S. E., Eem, S. H., Jeon, H. (2020). Concrete crack detection and quantification using deep learning and structured light. *Construction and Building Materials, 252(5),* 119096. https://doi.org/10.1016/j.conbuildmat.2020.119096
52. Nie, M., Wang, C. (2019). Pavement crack detection based on yolo v3. *2nd International Conference on Safety Produce Informatization (IICSPI)*, pp. 327–330. Chongqing, China, IEEE. https://doi.org/10.1109/IICSPI48186.2019.9095956
53. Liu, Z., Gu, X., Chen, J., Wang, D., Chen, Y. et al. (2023). Automatic recognition of pavement cracks from combined GPR B-scan and C-scan images using multiscale feature fusion deep neural networks. *Automation in Construction, 146(4),* 104698. https://doi.org/10.1016/j.autcon.2022.104698
54. Liu, Z., Gu, X., Yang, H., Wang, L., Chen, Y. et al. (2022). Novel YOLOv3 model with structure and hyperparameter optimization for detection of pavement concealed cracks in GPR images. *IEEE Transactions on Intelligent Transportation Systems, 23(11),* 22258–22268. https://doi.org/10.1109/TITS.2022.3174626
55. Li, W., Shen, Z., Li, P. (2019). Crack detection of track plate based on YOLO. *2019 12th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, pp. 15–18. Hangzhou, China, IEEE. https://doi.org/10.1109/ISCID.2019.10086
56. Li, Y., Han, Z., Xu, H., Liu, L., Li, X. et al. (2019). YOLOv3-Lite: A lightweight crack detection network for aircraft structure based on depthwise separable convolutions. *Applied Sciences, 9(18),* 3781. https://doi.org/10.3390/app9183781
57. Liu, J., Wang, X. (2020). Early recognition of tomato gray leaf spot disease based on MobileNetv2-YOLOv3 model. *Plant Methods, 16(1),* 83. https://doi.org/10.1186/s13007-020-00624-2
58. Zhang, X., Kang, X., Feng, N., Liu, G. (2020). Automatic recognition of dairy cow mastitis from thermal images by a deep learning detector. *Computers and Electronics in Agriculture, 178,* 105754. https://doi.org/10.1016/j.compag.2020.105754
59. Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B. et al. (2019). Searching for MobileNetV3. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324. Seoul, South Korea. https://doi.org/10.48550/arXiv.1905.02244
60. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. (2018). MobileNetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520. Salt Lake City, USA. https://doi.org/10.48550/arXiv.1801.04381
61. Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. https://doi.org/10.48550/arXiv.1804.02767
62. Redmon, J., Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271. Hawaii, USA. https://doi.org/10.48550/arXiv.1612.08242
63. Cui, X., Wang, Q., Dai, J., Zhang, R., Li, S. (2021). Intelligent recognition of erosion damage to concrete based on improved YOLO-v3. *Materials Letters, 302(3–4),* 130363. https://doi.org/10.1016/j.matlet.2021.130363
64. Zhang, Y., Huang, J., Cai, F. (2020). On bridge surface crack detection based on an improved YOLO v3 algorithm. *IFAC-PapersOnLine, 53(2),* 8205–8210. https://doi.org/10.1016/j.ifacol.2020.12.1994
65. Zhou, Z., Zhang, J., Gong, C. (2022). Automatic detection method of tunnel lining multi-defects via an enhanced you only look once network. *Computer-Aided Civil and Infrastructure Engineering, 37(6),* 762–780. https://doi.org/10.1111/mice.12836
66. Bučko, B., Lieskovská, E., Zábovská, K., Zábovský, M. (2022). Computer vision based pothole detection under challenging conditions. *Sensors, 22(22),* 8878. https://doi.org/10.3390/s22228878
67. Li, Y., Zhao, Z., Luo, Y., Qiu, Z. (2020). Real-time pattern-recognition of GPR images with YOLO v3 implemented by tensorflow. *Sensors, 20(22),* 6476. https://doi.org/10.3390/s20226476