

Article

Superficial Defect Detection for Concrete Bridges Using YOLOv8 with Attention Mechanism and Deformation Convolution

Tijun Li ¹, Gang Liu ^{1,2,*}  and Shuaishuai Tan ¹ 
¹ School of Civil Engineering, Chongqing University, Chongqing 400045, China; litijun@cqu.edu.cn (T.L.); shuaishuai_tan@cqu.edu.cn (S.T.)

² Key Laboratory of New Technology for Construction of Cities in Mountain Area, Chongqing University, Ministry of Education, Chongqing 400045, China

* Correspondence: gliu@cqu.edu.cn

Abstract: The accuracy of detecting superficial bridge defects using the deep neural network approach decreases significantly under light variation and weak texture conditions. To address these issues, an enhanced intelligent detection method based on the YOLOv8 deep neural network is proposed in this study. Firstly, multi-branch coordinate attention (MBCA) is proposed to improve the accuracy of coordinate positioning by introducing a global perception module in coordinate attention mechanism. Furthermore, a deformable convolution based on MBCA is developed to improve the adaptability for complex feature shapes. Lastly, the deformable convolutional network attention YOLO (DCNA-YOLO) detection algorithm is formed by replacing the deep C2F structure in the YOLOv8 architecture with a deformable convolution. A supervised dataset consisting of 4794 bridge surface damage images is employed to verify the proposed method, and the results show that it achieves improvements of 2.0% and 3.4% in mAP and R. Meanwhile, the model complexity decreases by 1.2G, increasing the detection speed by 3.5/f·s⁻¹.

Keywords: concrete surface defects; deep learning; YOLO; attention mechanism; deformable convolution



Citation: Li, T.; Liu, G.; Tan, S. Superficial Defect Detection for Concrete Bridges Using YOLOv8 with Attention Mechanism and Deformation Convolution. *Appl. Sci.* **2024**, *14*, 5497. <https://doi.org/10.3390/app14135497>

Academic Editors: Roque A. Osornio-Rios, Athanasios Karlis and Andres Bustillo Iglesias

Received: 21 May 2024
Revised: 14 June 2024
Accepted: 18 June 2024
Published: 25 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Bridges, which are crucial to national economic development, play an important role in the transport infrastructure. The number of constructed and operational bridges has exceeded 1.2 million in China, of which more than 70% are concrete bridges. As the service life increases, concrete bridges are inevitably subjected to various superficial defects such as cracking, concrete spalling, and rebar corrosion due to the combined effects of material aging, external loads, and the working environment [1]. These defects can affect the load-bearing capacity, service life, and overall safety of bridge structures [2]. Due to the large number and scale of concrete bridges, traditional manual inspections are inefficient to meet the demands of routine inspections. In addition, structures such as high piers and long spans require expensive auxiliary equipment for close-up inspection, which further increases inspection costs. With the improvements in resolution and accuracy of cameras, there is growing interest in using computer vision pattern recognition, including machine learning and deep learning, [3] to automatically detect bridge defects [4–7].

Machine learning methods mainly rely on template matching for defect detection, so such methods rely heavily on manual experience for sample feature extraction. In addition, single-layer features constructed by these methods have limited recognition ability when dealing with complex features and noise. Therefore, deep learning methods that can adaptively learn and extract image features have become mainstream, which are categorized into one-stage and two-stage methods based on the overall training conditions. One-stage detection methods extract features directly in the network to predict object classification and location, including the You Only Look Once (YOLO) [8] and Single Shot Multibox Detector (SSD) algorithms [9]. Two-stage methods typically use a Region

Proposal Network (RPN) to extract potential target area which typically exhibit higher detection accuracy and stability as well as more computing resources, such as Regions with Convolutional Neural Network (RCNN) [10], Fast RCNN [11], Faster RCNN [12], and Mask RCNN [13].

The YOLO algorithm has been continuously improved since it was proposed by Joseph Redmon et al. in 2016, whose detection accuracy and computing speed have been further improved [8,14–21]. The main feature of the YOLO algorithm is that it transforms object detection into a regression problem, dividing the image into $s \times s$ grids and directly predicting the class probabilities and position information within the corresponding bounding box of each grid. YOLOv8 [21] adopts an anchor-free approach, combines the Task-Aligned assigner positive sample assignment strategy [22] and decoupled head, and introduces a C2F structure with a richer gradient flow and distribution focal loss, further improving detection accuracy and speed. However, the YOLO algorithm was primarily designed for general image classification. To improve its effectiveness in identifying concrete cracks, Zhang XB et al. [23] proposed a YOLOv5 model enhanced with a fusion of spatial pyramid pooling cross-stage partial connections (SPPCSPCs) and a transposed convolution to detect cracks on bridge surfaces from different angles, demonstrating superior detection performance compared to other models on the ZJU SYG dataset (a crack data set for object detection based on deep learning provided by Zhejiang University). Yu Z et al. [24] introduced a concrete structure crack detection method based on an improved YOLOv5, using a threshold segmentation method based on Otsu's maximum inter-class variance to remove background noise in images, and optimizing the initial anchor box sizes with the K-means method. The improved average accuracy in complex environments increased by 6.87%. Jin Q et al. [25] proposed an improved YOLOv5 algorithm based on transformer heads and the self-attention mechanism, which effectively improved the detection and classification capabilities of concrete cracks, with a mean accuracy (mAP) of up to 99.5%. WU Y et al. [26] presented a lightweight LCANet backbone and a novel efficient prototype mask branch for crack detection based on the YOLOv8 instance segmentation model, reducing model complexity. Specifically, under conditions of 129 frames per second (FPS), the results of the case study showed the accuracy reached 94.5%, while the computational complexity decreased by 51%, compared to the original model.

The improvements made to the YOLO algorithm focus on optimizing the model structure, improving the loss functions and the feature extractors, and optimizing the data pre- and post-processing methods. These improvements have enhanced detection accuracy, speed, and robustness. However, in engineering applications, the underside of concrete bridge structures, where significant forces are applied, is prone to cracking, but these areas often have inadequate lighting conditions. In addition, structural corners and edges are susceptible to defects such as honeycombing and exposed rebar due to casting problems, but these areas have complex backgrounds and weak surface textures. In such cases, the YOLO algorithm can suffer from missed detections and false positives. To address these challenges, the deformable convolutional network attention YOLO (DCNA-YOLO) algorithm based on YOLOv8s is proposed in this study. A multi-branch coordinate attention mechanism (MBCA) is introduced to simultaneously incorporate spatial position information and global information. The attention weights for direction perception, position sensitivity, and global awareness are optimized to comprehensively improve the accuracy of coordinate localization. This effectively highlights features of target defect areas with uneven reflections and weak textures, thereby improving the representation and detection effects of the target detection algorithm. Thus, the balance between detection performance, speed, and model parameter size is achieved using MBCA. Furthermore, a deformable convolution method, named MBCADC, based on MBCA is presented. By embedding MBCA attention, this method improves the adaptability of the deformable convolution to significant illumination changes and complex feature shapes in regions with uneven reflections. As a result, it better accommodates different image structures and texture

features. The complexity of the model is reduced, while recall (R) and average precision are improved, and missed detections and false positives are reduced.

The paper is organized as follows. Firstly, an overview of the YOLO algorithm is given for the problem under study, and the improvements of existing methods are compared. Subsequently, an improved framework that incorporates deformable convolution (DC) modules with the multi-branch coordinate attention (MBCA) mechanism is introduced. Finally, this new framework is validated with a dataset containing 4794 damage images and compared with other algorithms. Flowchart of study as shown in Figure 1.

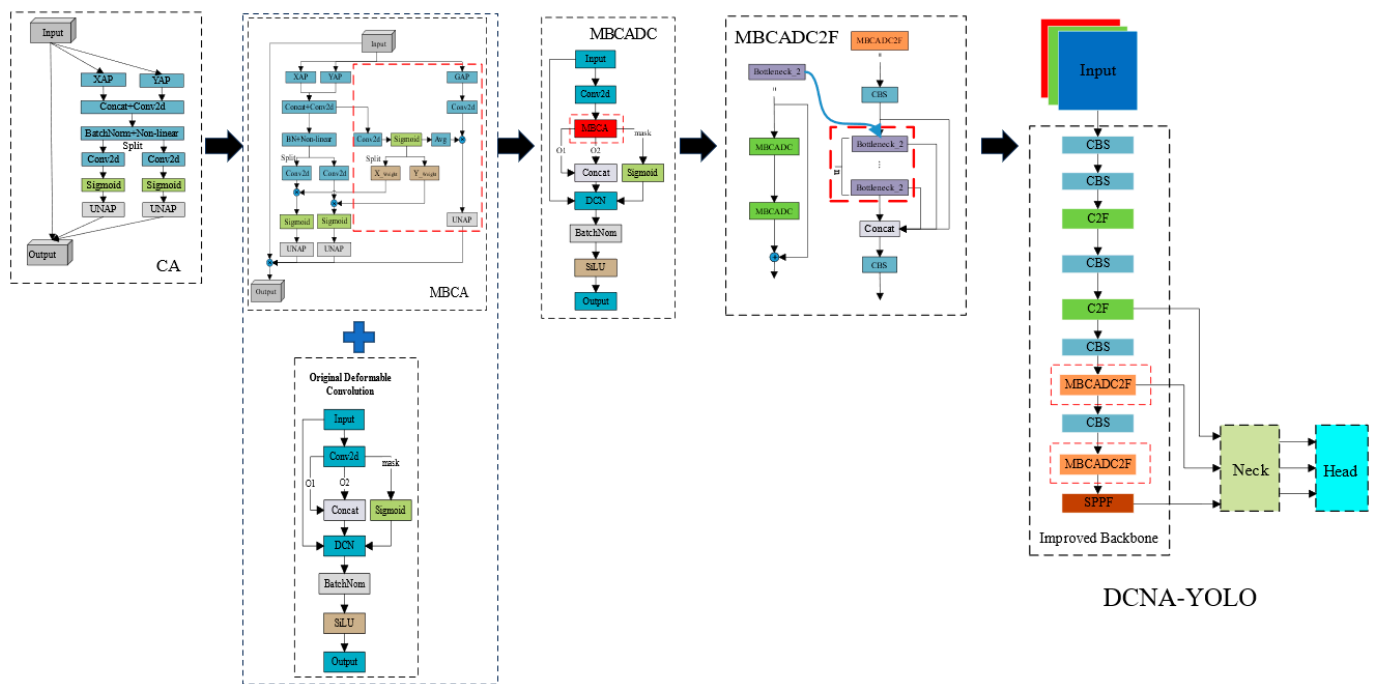


Figure 1. Flowchart of study.

2. Basic Theory of the YOLOv8 Network

YOLOv8s was introduced in 2023 as the latest version of YOLO, which supports image classification, object detection, and instance segmentation tasks. The model structure consists of three main components: backbone, neck, and head, as shown in Figure 2. The received training data are pre-processed using mosaic data augmentation before entering the backbone network for feature extraction. Then, the backbone outputs three feature maps of different scales to the neck structure for bidirectional feature fusion. Finally, the head uses convolutional layers to scale the fused feature maps, producing outputs at three different scales.

The backbone network consists of convolution batch normalization sigmoid linear unit (CBS), cross-stage partial fusion (C2F), and spatial pyramid pooling fast (SPPF) modules [27], where the CBS module is primarily used to extract features from the input image, the C2F module retains lightweight properties while capturing richer gradient flow information, and the SPPF module employs spatial pyramid pooling by serially computing three MaxPool2d operations with 5×5 convolutional kernels. The optimizations made in this paper focus on this component; further information on YOLOv8s can be found in reference [21].

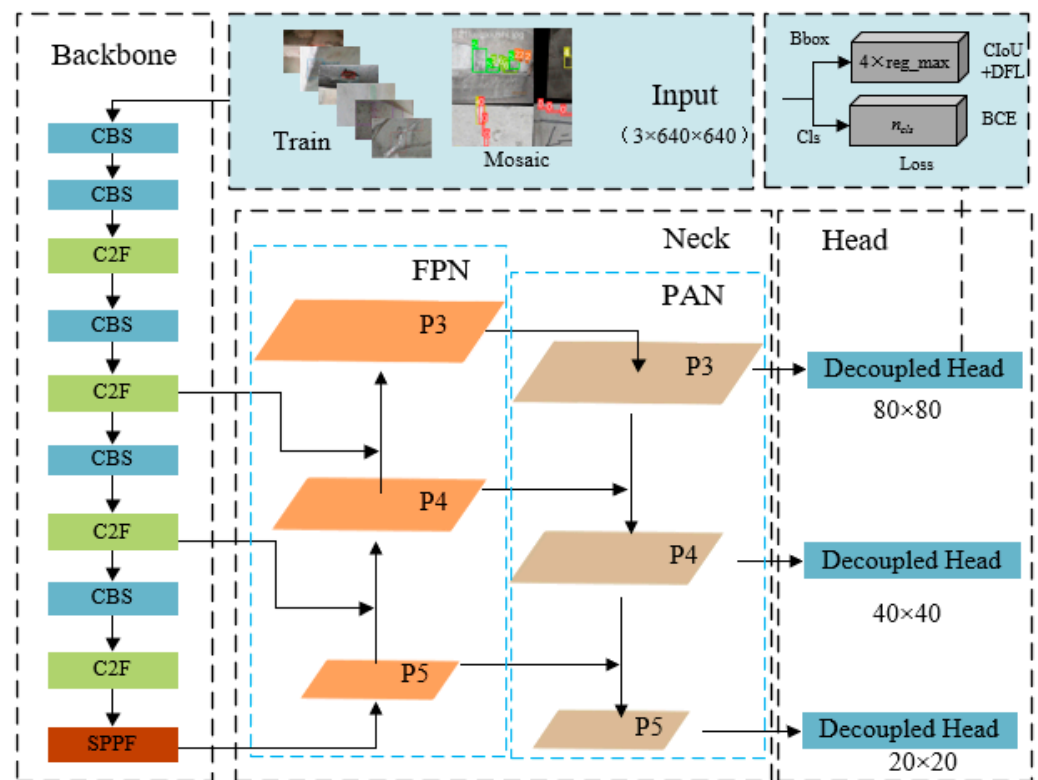


Figure 2. Network structure diagram of YOLOv8s.

3. DCNA-YOLO Method Construction

3.1. Multi-Branch Coordinate Attention

Critical information about objects may be obscured by noise, image backgrounds, and uneven lighting due to complex, blurry, and poorly lit environments. Therefore, enhancing the positional information of features is a significant challenge. To address this issue, attention mechanisms [28] have been introduced in recent years. Among these methods, the coordinate attention mechanism (CAM) [29] effectively enhances the extraction of structural information about objects by using two one-dimensional average pooling operations to aggregate the feature maps vertically and horizontally into two separate orientation-aware feature maps, which are subsequently encoded into an attention tensor containing orientation–position information, and ultimately decomposed into a pair of attention maps that are both orientation- and position-aware.

Although the CAM is effective in capturing long-range dependencies in local spatial information, it overlooks the global dependencies necessary for understanding spatial features. To address this limitation, a global context perception module is introduced into the CAM, aiming to help the network acquire global information by considering the overall context comprehensively. This results in more precise and comprehensive image representations for processing tasks. Additionally, by optimizing attention weights for direction awareness, position sensitivity, and global perception, the network can selectively focus on key areas of the target, significantly improving coordinate localization accuracy.

As shown in Figure 3, the multi-branch coordinate attention (MBCA) principle is described, which consists of two steps:

Step 1: In the information embedding phase, each channel of the input feature map X is encoded using two spatial range pooling kernels: $(H, 1)$ and $(1, W)$ to embed coordinate information. The kernels operate along the horizontal (width W) and vertical (height H) coordinates, respectively.

$$\begin{cases} z_c^h(h) = \frac{1}{W} \sum_{0 \leq i < W} x_c(h, i) \\ z_c^w(w) = \frac{1}{H} \sum_{0 \leq j < H} x_c(j, w) \end{cases} \quad (1)$$

where h and w represent the height and width of the current input feature map, respectively. c denotes the current input feature map's channel. $z_c^h(h)$ denotes the output of channel c at height h , and $z_c^w(w)$ denotes the output of channel c at width w .

The global information is additionally embedded in the CAM by encoding each channel of the input feature map X through global average pooling (GAP).

$$Z_c = \frac{1}{h \times w} \sum_{i=1}^h \sum_{j=1}^w x_c(i, j) \quad (2)$$

where z_c represents the global information output of channel c .

Step 2: At the coordinated and global attention generation phase, the aggregated feature maps Z^h and Z^w generated from Equation (1) are firstly concatenated, and then a 3×1 convolution operation is applied for information fusion, compressing feature channels. After batch normalization and non-linear activation functions, the intermediate feature map is split along the spatial dimension into two independent tensors f^h and f^w . Subsequently, two 1×1 convolutions F_h and F_w are used to transform f^h and f^w into tensors with the same number of channels as the input X . Furthermore, to fully utilize the expressive representation of the aggregated feature maps and accurately highlight the regions of interest, a feature map optimization module is proposed. This involves passing the intermediate feature map f through a 1×1 convolutional transformation function F_1 to adjust the number of channels. After applying a non-linear activation function and a sigmoid function, a feature map optimization weight matrix g_1 is obtained. The optimization weight matrix is then split along the spatial dimension into independent weights g_1^h and g_1^w for the vertical and horizontal directions, respectively. Finally, the two independent weights g_1^h and g_1^w are applied to the corresponding tensors, and after passing through a sigmoid function, the outputs g^h and g^w are expanded and used as attention weights.

$$\begin{cases} f = \delta(F_{3 \times 1}([Z^h, Z^w])) \\ g_1 = \sigma(\delta(F_1(f))) \\ g_1^h = F_h(g_1) \\ g_1^w = F_w(g_1) \\ g^h = \sigma(F_h(f^h) \times g_1^h) \\ g^w = \sigma(F_w(f^w) \times g_1^w) \end{cases} \quad (3)$$

where $[\cdot, \cdot]$ represents the concatenation operation along the spatial dimension. $F_{3 \times 1}(\cdot)$ denotes the 3×1 convolution transformation function, $\delta(\cdot)$ represents the non-linear activation function hard_swish. f represents the intermediate feature map with horizontal and vertical spatial information, where $f \in \mathbb{R}^{C/r \times (W+H)}$. r is the reduction ratio, taken as 16. F_h and F_w represent 1×1 convolution operations in the vertical and horizontal directions, respectively. f^h and f^w represent intermediate feature maps in the vertical and horizontal directions, where $f^h \in \mathbb{R}^{C/r \times 1 \times H}$ and $f^w \in \mathbb{R}^{C/r \times 1 \times W}$. σ denotes the sigmoid activation function. g_1 represents the feature map optimization weight matrix. g_1^h and g_1^w represent optimization weights in the vertical and horizontal directions, respectively. g^h and g^w represent attention weights in the vertical and horizontal directions, respectively.

For global attention generation, the global information feature map generated from Equation (2) is multiplied element-wise by the average-weighted optimization weight matrix g_1 . Subsequently, the result passes through a sigmoid function to obtain the global attention weights g^G .

$$\begin{cases} f_c = \delta(F_1(Z_c)) \\ g^G = \sigma(\text{mean}(g_1) \cdot f_c) \end{cases} \quad (4)$$

where $F_1(\cdot)$ represents the 1×1 convolution transformation function. $mean(\cdot)$ represents the mean function. f_c represents the intermediate feature map with global information, where $f_c \in R^{C \times 1 \times 1}$. g_1 represents the feature map optimization weight matrix. g^G represents the global attention weights.

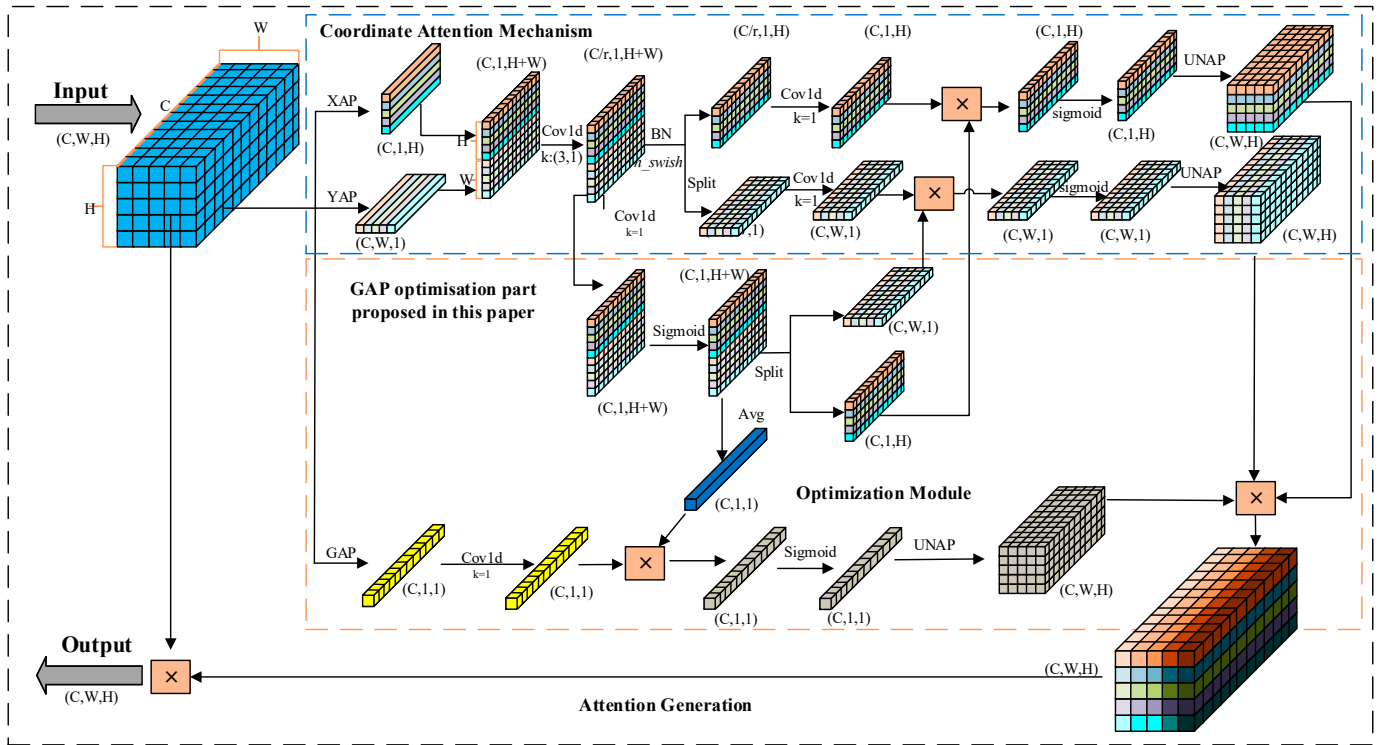


Figure 3. The principle of the multi-branch coordinate attention (MBCA) algorithm.

The attention weights g are calculated by multiplying the vertical and horizontal direction attention weights by the global attention weights.

$$g_c = g_c^h(i) \times g_c^w(j) \times g_c^G(i, j) \quad (5)$$

where g_c represents the attention weight at channel c . g_c^h represents the vertical direction attention weight at channel c . g_c^w represents the horizontal direction attention weight at channel c . g_c^G represents the global attention weight at channel c .

It computes the average of all elements within each feature map of the input, resulting in a feature map with a size of 1×1 . When direct multiplication or addition operations are needed for the original input, reverse average pooling (UNAP) can be used to expand the feature map to the desired size. The specific pooling operations are illustrated in Figure 4.

The multi-branch coordinate attention (MBCA) mechanism is established by the two steps above, introducing global-level information and optimizing the perception of specific target positions at the local level.

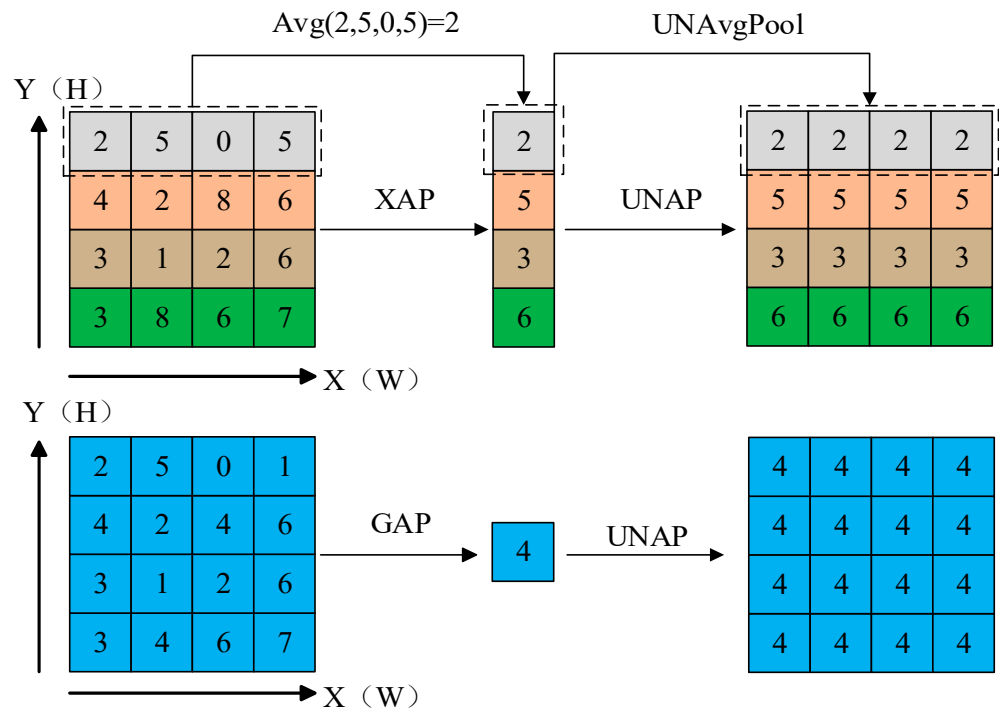


Figure 4. Schematic diagram of XAP, GAP, and UNAP.

3.2. Deformable Convolution Based on MBCA

The features of the image become more complex and irregular due to weak texture or lighting changes on the target surface with the detection of weak texture areas and uneven reflection areas, making the fixed receptive field kernel insufficient to adapt complex features. To address this issue, deformable convolution (DCNv2) [30] is applied to learn the offset and modulation parameters for each pixel to better adjust to the sampling position of the convolution kernel and to adapt to different image structures and texture features.

For each position p_0 in the output feature map y , the deformable convolution structure is defined as follows:

$$y(p_0) = \sum_{p_n \in R} w(p_0) \cdot x(p_0 + p_n + \Delta p_n) \cdot \Delta m_n \quad (6)$$

where the grid R defines the size and expansion rate of the receptive field. For a receptive field of size 3×3 and a dilation rate of 1, $R = \{(-1, -1), (1, 0), \dots, (0, 1), (1, 1)\}$. p_0 corresponds to mapping each point of the output feature map y to the center of the convolution kernel, and then mapping it to the coordinates in the input feature map x ; p_n represents the relative coordinates in R for p_0 . $w(\cdot)$ denotes the sampling point weight, and $x(\cdot)$ denotes the mapping of the coordinates in the input feature map x to feature vectors. The offset $\{\Delta p_n | n = 1, 2, \dots, N\}$, $N = |R|$, and the modulation parameter Δm_n lies within $[0, 1]$.

The offsets Δp_n and modulation parameters Δm_n of a deformable convolution are obtained by applying a separate standard convolution layer to the same input feature map, which results in insufficient spatial support range. As a consequence, the effective receptive field of foreground nodes and the prominent region constrained by errors may include background areas irrelevant to detection. The proposed multi-branch coordinate attention (MBCA) is embedded during the process of generating the offsets Δp_n and modulation parameters Δm_n , which is named MBCADC, to further enhance the ability of deformable convolution DCNv2 to manipulate spatial support regions, as illustrated in Figure 5.

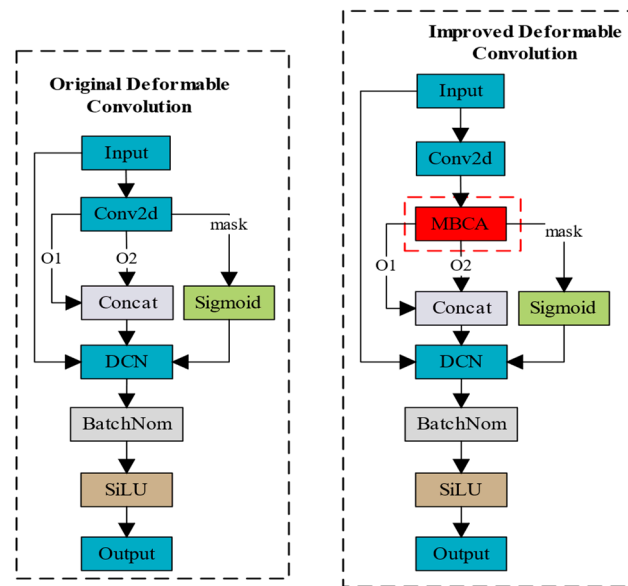


Figure 5. Structure of original deformable convolution (DCNv2) and improved deformable convolution (MBCADC).

The MBCADC module consists of a Conv2d, MBCA attention, a DCN, BatchNorm2d, and a SiLU activation function, where $o1$ and $o2$ represent learned offsets in the x - and y -coordinate directions, respectively, and $mask$ denotes the sampling weights at different positions. The structure of the deformable convolution MBCADC is defined as follows:

$$y(P_0) = \sum_{p_n \in R} w(p_0) \cdot x(p_0 + p_n + \Delta p_n \cdot g) \cdot \Delta m_n \cdot g \quad (7)$$

where g represents the attention weights generated by multi-branch coordinate attention (MBCA).

Figure 6 illustrates the process of MBCADC deformable convolution. From Figure 6, it can be observed that the sampling matrix of deformable convolution is non-fixed and deformable, with the offsets determined by algorithms that can better learn the geometric properties of the objects to be detected.

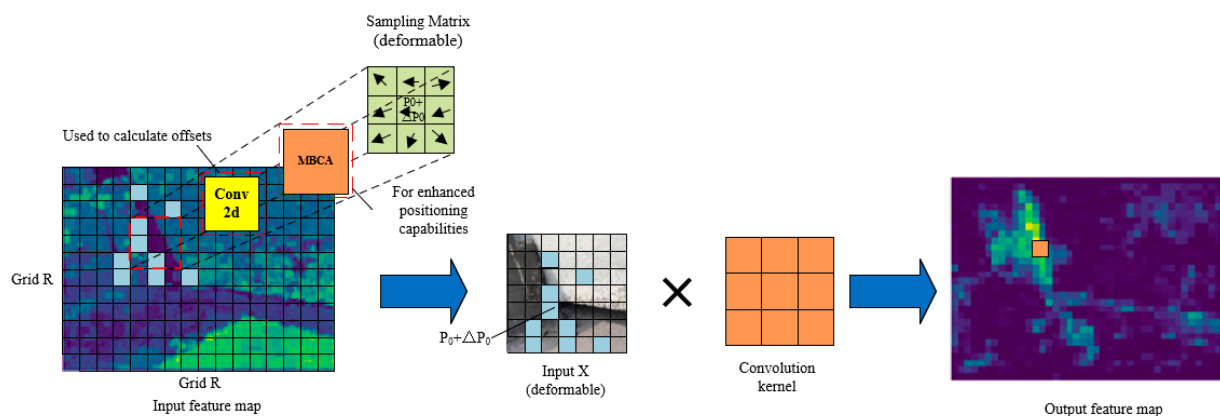


Figure 6. Process of MBCADC deformable convolution.

3.3. MBCADC2F Module

The MBCAC2F module is an improvement over the YOLOv8s backbone network's C2F module, integrating the multi-branch coordinate attention deformable convolution (MBCADC) module. This module comprises two CBS modules and n Bottleneck modules, where the Bottleneck module contains a residual structure with two MBCADC modules, as

illustrated in Figure 7. By learning the parameters of deformable convolution, the model can dynamically adjust the sampling positions of the convolution kernel based on the actual shape and positional information of the target, allowing for more precise capture of target features.

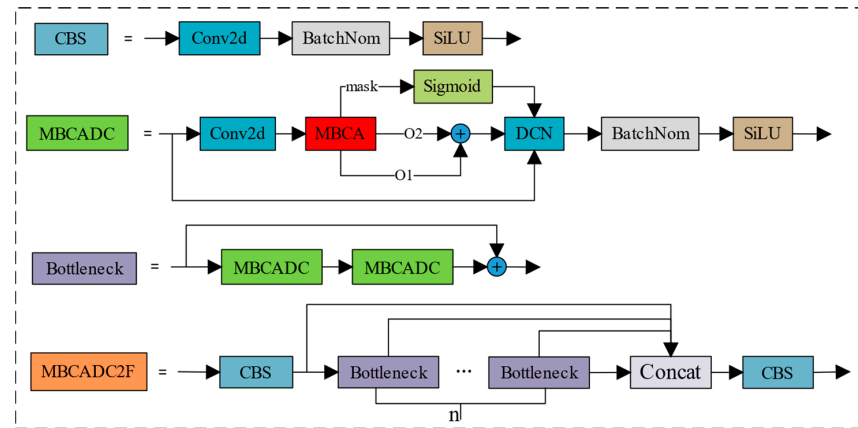


Figure 7. The structure of the MBCADC2F module.

3.4. Deformable Convolutional Network Attention YOLO Object Detection Network

The deformable convolutional network attention YOLO (DCNA-YOLO) algorithm improves upon the YOLOv8s baseline model by modifying the backbone network. The neck and head structures of the DCNA-YOLO model remain the same as those of the baseline model. The model's overall structure is illustrated in Figure 8. The DCNA-YOLO backbone network comprises five CBS modules, two C2F modules, two MBCADC2F modules, and one SPPF module. The structures of modules such as CBS, C2F, and SPPF in the MBCADC2F module are identical to those of the corresponding modules in the YOLOv8s baseline model. Each Bottleneck unit in the MBCADC2F module contains a residual connection structure with two MBCADC modules.

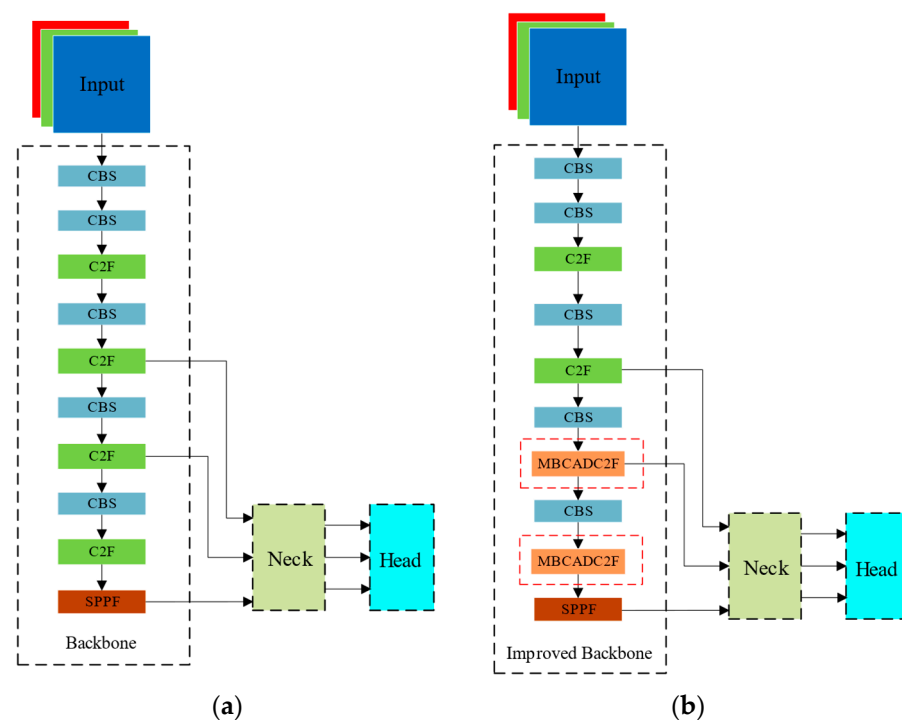


Figure 8. Original YOLOv8s and proposed DCNA-YOLO overall framework. (a) Original YOLOv8s overall framework; (b) proposed DCNA-YOLO overall framework.

4. Example Verification

4.1. Experimental Dataset

We created a dataset of apparent damage to concrete bridges, consisting of 4794 images. Domain experts annotated the dataset, which we used to evaluate the effectiveness of the proposed method in identifying apparent damage to concrete bridges. According to the regulations of China's road and bridge maintenance standards and inspection standards, apparent concrete damage was classified into seven types: cracks, spalling, honeycombing, exposed reinforcement, water seepage, and voids. The constructed dataset included at least one type of damage in each image. Augmenting the dataset enhances its diversity and richness, making the model's detection more effective. The original dataset was randomly divided into training, validation, and testing sets with a ratio of 8:1:1. Data augmentation techniques, such as flipping, rotation, and HSV (hue, saturation, and value) enhancement, were then applied to the divided dataset. The dataset sizes were as follows: 14,528 images in the training set, 1816 images in the validation set, and 1816 images in the testing set. Table 1 shows the statistical results of the number of annotated boxes for each type of damage.

Table 1. Number of Labels for Each Damage in the Dataset.

Labels (Damage)	Number	Labels (Damage)	Number
liefeng (crack)	17,636	shenshui (seepage)	9244
boluo (spalling)	11,875	fengwo (comb surface)	8330
kongdong (cavity)	7082	lujin (steel exposed)	6584
mamian (pockmark)	8274		

The images of the bridge's apparent damage collected in the experimental dataset were affected by the lighting environment, resulting in variations in image quality. Statistical analysis was conducted on the grayscale histograms of the images, which allowed for the categorization of the images into four lighting conditions:

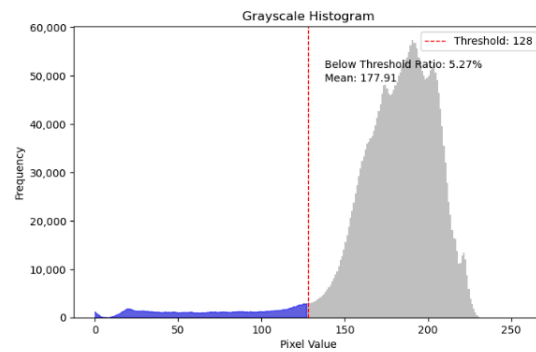
(1) Well-lit images, which exhibit high clarity and rich details. The histogram of the grayscale image displays a unimodal distribution, with grayscale values primarily ranging from 150 to 250. The average grayscale value of the image is greater than 170.

(2) Partial shadow or occlusion images: The grayscale distribution is complex, with areas of varying brightness. The histogram of the grayscale image displays a bimodal distribution, with grayscale values primarily ranging from 50 to 100 and from 150 to 250. The image's average grayscale value is approximately 150.

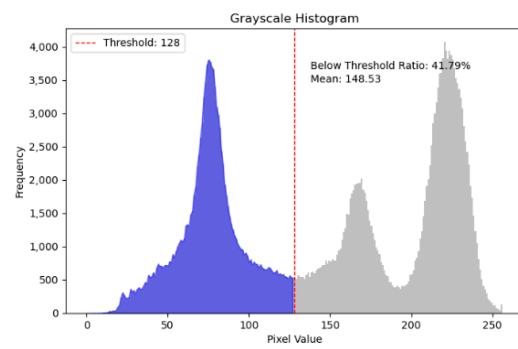
(3) Low-lighting images: The overall low brightness can result in blurry or confusing areas in the apparent damaged regions. The grayscale histogram of the image displays a unimodal distribution, with grayscale values primarily distributed between 50 and 100. The image's average grayscale value is approximately 100.

(4) Dark-lighting images: The overall low brightness can make it challenging to distinguish details of the apparent damage. The image's grayscale histogram displays a unimodal distribution, with grayscale values ranging from 0 to 50. The average grayscale value of the image is below 30.

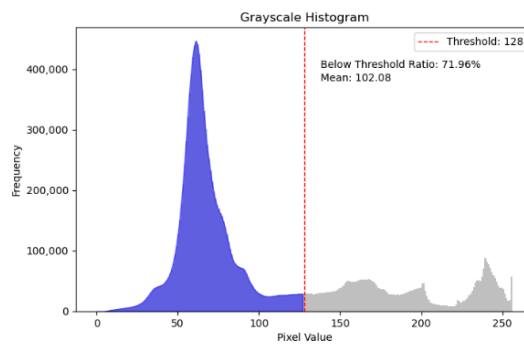
Figure 9 shows the grayscale histograms for the four distinct illumination conditions. Table 2 presents a statistical study of the number of well-lit images, partial shadow or occlusion images, low-lighting images, and dark-lighting images. Figure 10 illustrates some examples of picture data.



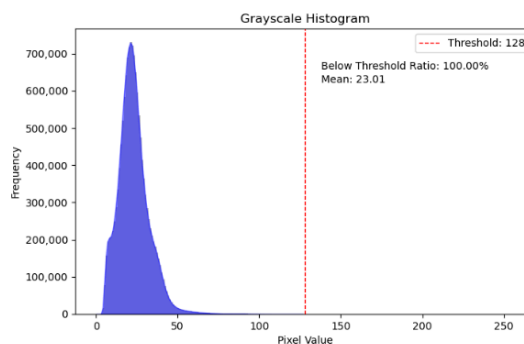
(a)



(b)



(c)



(d)

Figure 9. Images and grayscale histograms captured under varying lighting conditions. (a) Well-lit image and its gray histogram; (b) partial shadow or occlusion image and its gray histogram; (c) low-lighting image and its gray histogram; (d) dark-lighting image and its gray histogram.

Table 2. Summary of Image Quantities Under Different Illumination Conditions.

Data Set	Well-Lit Images	Partial Shadow or Occlusion Images	Low-Lighting Images	Dark-Lighting Images	Total
Train	6697	1798	2608	3425	14,528
Val	923	239	298	356	1816
Test	876	225	326	389	1816

**Figure 10.** Partial images of the training set. (a) Examples of well-lit images; (b) examples of partial shadow or occlusion images; (c) examples of low-lighting images; (d) examples of dark-lighting images.

4.2. Environmental Design and Evaluation Metrics

The computer hardware setup included an Intel Core i5-13600K CPU, 48 GB of RAM, and an NVIDIA GeForce RTX4070 with a 12,282 Mib GPU. The experimental environment consisted of Windows 10, CUDA 11.8, PyTorch 2.0.1, and Python 3.8.18. The training parameter settings had an initial learning rate of 0.01, with a learning rate strategy that employed cosine annealing. The number of training epochs was set to 300, with an initial input size of the model at 640×640 and a batch size of 16. The optimization algorithm used was SGD [31], with the loss function being cross-entropy. To prevent overfitting, early stopping criteria were employed. The network training was halted if the validation accuracy did not improve after 50 epochs.

To evaluate the model's detection performance for seven types of visual damage, we used precision (P), recall (R), mean average precision (mAP), model parameter quantities (parameters), floating-point operations (FLOPs), and frames per second (FPS) as the evaluation metrics.

4.3. Experimental Results and Analysis

4.3.1. Ablation Experiment

To further validate the effectiveness of the proposed improvements in terms of the number and placement of different enhancement modules, ablation experiments and quantitative and qualitative analyses were performed using the generated dataset in the same experimental environment to evaluate the benefits of key components in the model. Among them, "MBCA" refers to the addition of MBCA attention after the last layer of the backbone network SPPF; "DCNv2" refers to the replacement of the Bottleneck in the C2F module of the eighth layer of the backbone network with deformable convolution DCNv2; "Proposed method" refers to replacing the C2F module of the sixth and the eighth layer of the backbone network with the MBCADC2F module proposed in this paper. The experimental results are presented in Table 3.

Table 3. Results of ablation experiments.

Model	Parameters/M	FLOPs/G	FPS/f·s ⁻¹	P/%	R/%	mAP _{0.5} /%	mAP _{0.5:0.95} /%
YOLOv8s	11.1	28.7	70.9	89.1	82.0	87.4	68.9
+MBCA	11.2	28.7	68.9	90.2	82.7	87.9	68.9
+DCNv2	11.2	27.5	73.8	90.0	82.5	88.1	70.2
proposed method	11.3	27.5	74.4	91.3	85.4	89.4	73.3

A comparison of the data in the table shows that the MBCA attention mechanism proposed in this study did not significantly increase the number of network parameters or model complexity, but improved the model's mAP_{0.5} value by 0.5%. The deformable convolution DCNv2 was able to reduce the model complexity and improve the model mAP_{0.5} without significantly increasing the model parameter count. The method proposed in this paper achieved the best experimental results, with only a 0.2 M increase in model parameters compared to the baseline model. Furthermore, floating-point operations were reduced by 1.2G, precision increased by 1.8%, recall improved by 3.4%, and mAP_{0.5} and mAP_{0.5:0.95} increased by 2.0% and 4.4%, respectively. In addition, the model's speed of detection increased by 3.5/f·s⁻¹.

Figure 11 shows a comparison between the test results and the heatmap analysis of the baseline model and the proposed method for well-lit images. The heatmaps were generated using the Grad-CAM method. They show that both models achieved good detection results for all seven types of surface defects in images with good lighting conditions. The proposed method detected all defects, while the baseline model missed a small piece of exposed rebar (fourth-row image). Furthermore, the accuracy of the detection results obtained by the proposed method was consistently higher than that of the baseline model. A comparison of the heatmaps shows that the proposed method provided a better representation, with

the heatmaps better conforming to the shape of the target. These results indicate that the proposed method was effective in improving the accuracy of detecting images with good lighting conditions.

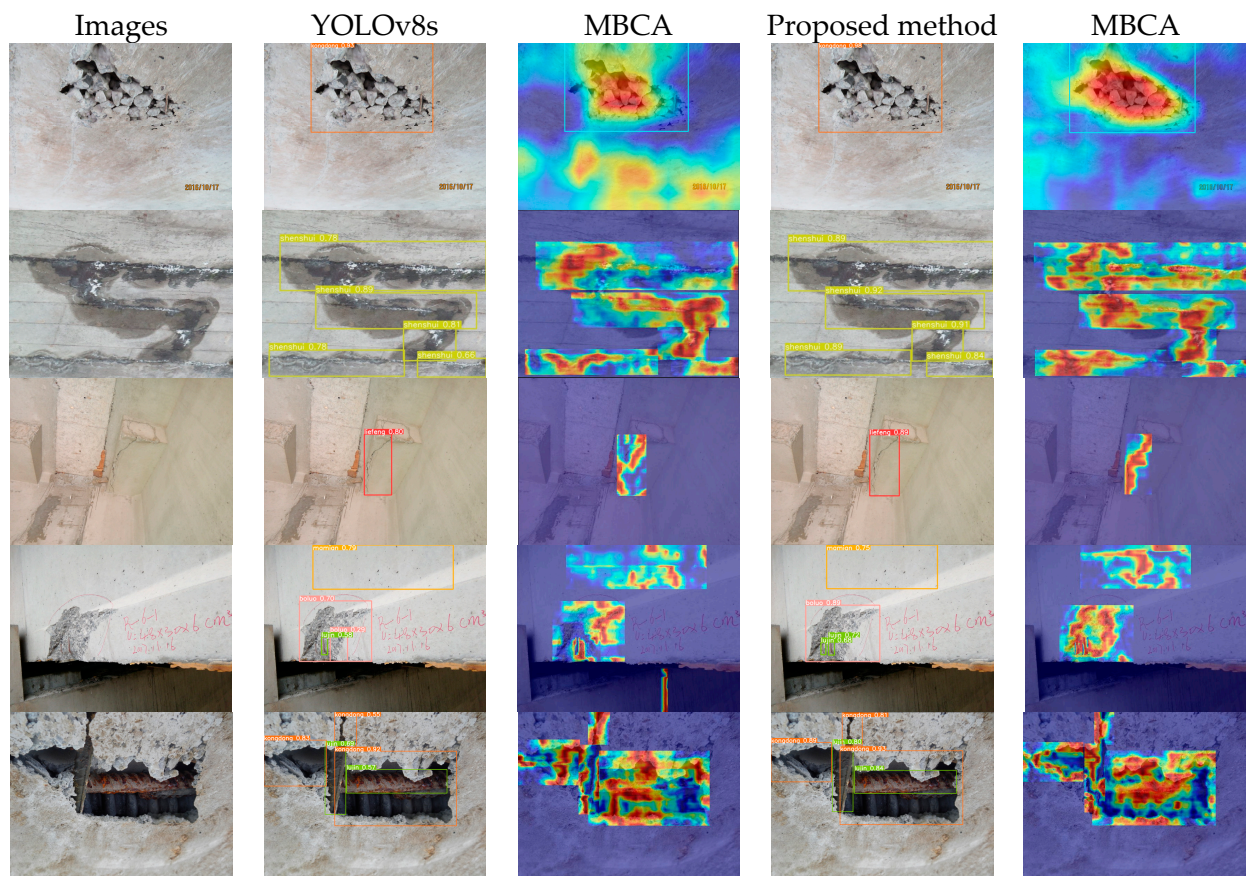


Figure 11. Detection results and heatmaps of seven types of damage in well-lit images.

Figure 12 shows a comparison between the test results and the heatmap analysis of the baseline model and the proposed method for images with partial shadows or occlusions. It can be seen that the proposed method achieved better detection results than the baseline model for images with partial shadows or occlusions. In images with partial shadows or occlusions, there were areas of varying brightness due to significant changes in illumination and uneven reflections on the target surface. The baseline model with fixed geometric structures of convolutional kernels was not effective in capturing the spatial information of the target in these regions. The proposed method used deformable convolution based on multi-branch coordinate attention, which allowed the model to dynamically adjust the sampling positions of the convolutional kernels according to the actual shape and position information of the target. A comparison of the heatmaps shows that the proposed method provided a better representation, with the heatmaps focusing more on the edge features of the target, which effectively reduced the rates of missed detections and false positives in images with partial shadows or occlusions.

Figure 13 shows the comparison between the test results and the heatmap analysis of the baseline model and the proposed method. It can be seen that both the baseline model and the proposed method could detect the class and location of defects in the image under low-light conditions. However, the detection accuracy of the proposed method was higher compared to the baseline model. The texture of the target region in the image may have been relatively weak, resulting in the lower detection accuracy of the model. A comparison of the heatmaps shows that the proposed method could effectively highlight the features of the defective region, thus improving the detection accuracy of the model.

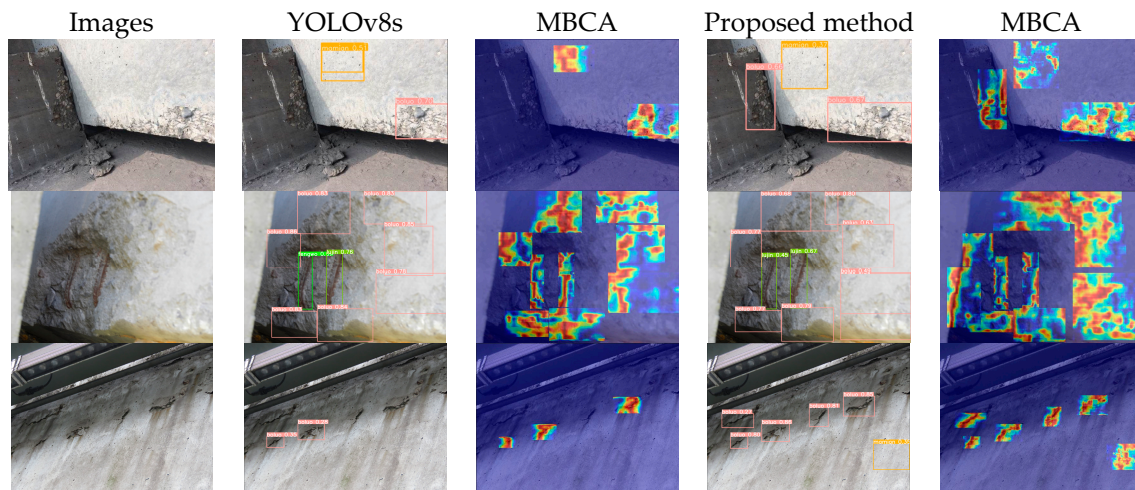


Figure 12. Detection results and heatmaps of damage in partially shaded or occluded images.

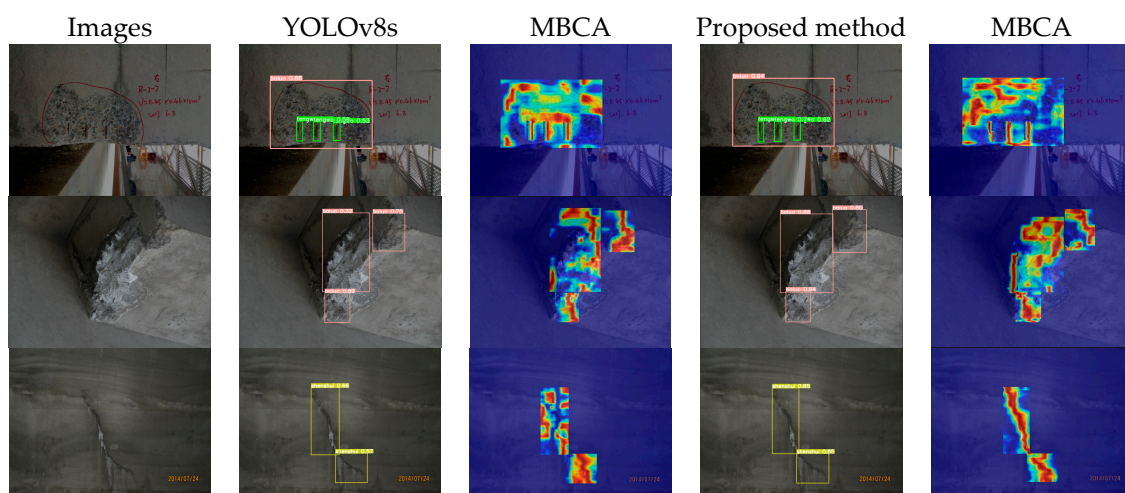


Figure 13. Detection results and thermograms of damage in low-light images.

Figure 14 illustrates a comparison between the test results and the heatmap analysis of the baseline model and the proposed method for images with dark lighting conditions. From Figure 14, it can be seen that both the baseline model and the proposed method performed poorly on images with dark lighting conditions. In images with dark lighting conditions, the overall brightness was low, making it difficult to detect surface defect details. The models failed to learn useful information from the images, resulting in incorrect target category detection or no defect detection.

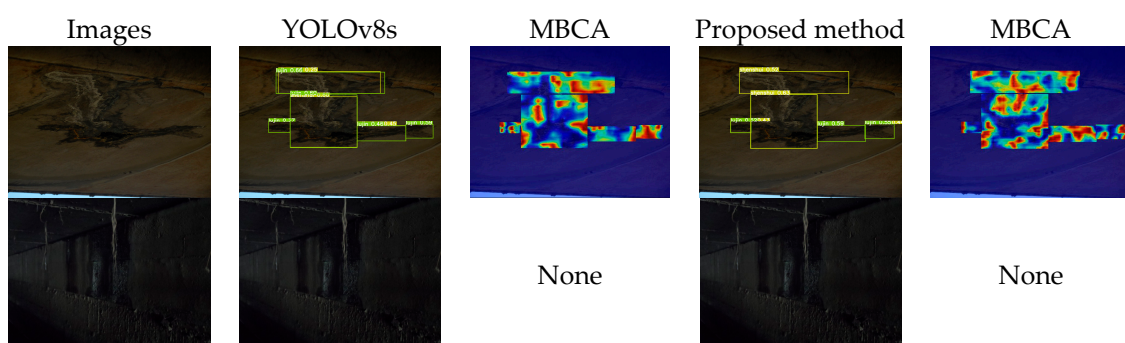


Figure 14. Detection results and thermograms of damage in dark and light images.

In summary, the proposed method effectively improved the accuracy of detecting images with good lighting conditions and those with poor lighting conditions, effectively mitigating the problems of missed detections and false positives.

4.3.2. Comparison Experiment of Different Detection Algorithms

Considering the real-time performance and accuracy requirements for concrete bridge surface defect detection tasks, the two-stage object detection models of the RCNN series and the outdated SSD models were not included in the comparative experiments. Instead, more widely used and advanced models from the YOLO series were selected as the benchmark models. The results are presented in Table 4.

Table 4. Comparative experimental results of different network models.

Model	Parameters/M	FLOPs/G	FPS/f.s ^{−1}	P/%	R/%	mAP _{0.5} /%	mAP _{0.5:0.95} /%
YOLOv3-tiny	12.1	19.1	76.9	81.7	73.4	78.6	53.4
YOLOv5s	7.0	16.8	78.3	91.2	84.9	88.7	67.4
YOLOv6s	16.3	44.2	69.4	90.2	81.5	87.7	69.6
YOLOv8s	11.1	28.7	70.9	89.1	82.0	87.4	68.9
Proposed method	11.3	27.5	74.4	91.3	85.4	89.4	73.3

Comparing the data in Table 4, it is evident that the model proposed in this paper exhibited more effective performance in terms of detection accuracy compared to the current state-of-the-art (SOTA) models. The mAP_{0.5} value was improved by 11.1%, 0.7%, and 1.7% compared to the classical YOLOv3-tiny, YOLOv5s, and YOLOv6s models, respectively. Compared to the baseline model YOLOv8s, the proposed model achieved a 2.0% and 4.4% improvement in average precision (mAP_{0.5} and mAP_{0.5:0.95}, respectively), a 3.4% increase in recall rate, an increase of 3.5/f.s^{−1} in detection speed, and a reduction in model complexity by 1.2G. These results demonstrate that the proposed model had better detection performance in concrete bridge surface defect detection.

5. Conclusions

The current work adopts a novel object detection algorithm termed DCNA-YOLO based on multi-channel attention mechanisms and deformable convolutions. It is proposed to address problems such as missed detections, false positives in regions with insufficient illumination, complex backgrounds, and weak surface textures on concrete bridge surfaces. The major findings of this work are concluded below:

(1) Multi-branch coordinate attention (MBCA) is adopted on the basis of CA with a supplementation of the global information branch. MBCA is applied to obtain spatial coordinate information and global information simultaneously, which improves the accuracy of coordinate information in the attention mechanism.

(2) The MBCA mechanism is embedded with a deformable convolution, then used to enhance the adaptability of the convolution kernel. This novel coordinated model contributes the coordinate localization ability for better adaptation to different image structures and texture features.

(3) The proposed framework (Figure 8) is validated through a self-constructed concrete surface defect dataset. Our results effectively highlight the accuracy of detecting regions with significant light variations, uneven reflections, and weak textures without increasing the complexity. All of these mitigate the problems of missed detections and false alarms.

(4) The next plan is to prune and distill the knowledge of the DCNA-YOLO model to develop a lighter model that can be deployed on resource-constrained concrete bridge health inspection drones for efficient real-time inspection. This will improve the safety, efficiency, and accuracy of the inspection and provide a scientific basis for the maintenance and management of concrete bridges.

Author Contributions: Conceptualization, T.L.; methodology, T.L.; software, T.L.; validation, T.L. and S.T.; formal analysis, G.L.; investigation, G.L.; resources, G.L.; data curation, T.L.; writing—original draft preparation, T.L.; writing—review and editing, S.T.; visualization, T.L.; supervision, G.L.; project administration, G.L.; funding acquisition, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (52221002, 52078084), Fundamental Research Funds for the Central Universities (2023CDJKYJH093), and the 111 project of the Ministry of Education and the Bureau of Foreign Experts of China (Grant No. B18062).

Data Availability Statement: The data used in this paper can be obtained through the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xing, S.; Ye, J.; Jiang, C. Review the study on typical diseases and design countermeasures of China concrete curved bridges. In Proceedings of the 2010 International Conference on Mechanic Automation and Control Engineering, Wuhan, China, 26–28 June 2010; pp. 4805–4808.
2. Ni, F.; Zhang, J.; Chen, Z. Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. *Comput. Aided Civ. Infrastruct. Eng.* **2019**, *34*, 367–384. [CrossRef]
3. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
4. Cha, Y.-J.; Choi, W.; Büyükoztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput. Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
5. Ni, F.; Zhang, J.; Chen, Z. Pixel-level crack delineation in images with convolutional feature fusion. *Struct. Control Health Monit.* **2019**, *26*, e2286. [CrossRef]
6. Dung, C.V.; Anh, L.D. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58. [CrossRef]
7. Ali, R.; Cha, Y.-J. Subsurface damage detection of a steel bridge using deep learning and uncooled micro-bolometer. *Constr. Build. Mater.* **2019**, *226*, 376–387. [CrossRef]
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016, Lecture Notes in Computer Science*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
11. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
13. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef] [PubMed]
14. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
15. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
16. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
17. Ultralytics YOLOv5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 20 August 2023).
18. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
19. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
20. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
21. Ultralytics YOLOv8. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 12 October 2023).
22. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-aligned One-stage Object Detection. In Proceedings of the International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021.
23. Zhang, X.; Luo, Z.; Ji, J.; Sun, Y.; Tang, H.; Li, Y. Intelligent Surface Cracks Detection in Bridges Using Deep Neural Network. *Int. J. Struct. Stab. Dyn.* **2023**, *24*, 2450046. [CrossRef]
24. Yu, Z. YOLO V5s-based Deep Learning Approach for Concrete Cracks Detection. *SHS Web Conf.* **2022**, *144*, 03015. [CrossRef]

25. Jin, Q.; Han, Q.; Su, N.; Wu, Y.; Han, Y. A deep learning and morphological method for concrete cracks detection. *J. Circuits Syst. Comput.* **2023**, *32*, 2350271. [[CrossRef](#)]
26. Wu, Y.; Han, Q.; Jin, Q.; Li, J.; Zhang, Y. LCA-YOLOv8-Seg: An Improved Lightweight YOLOv8-Seg for Real-Time Pixel-Level Crack Detection of Dams and Bridges. *Appl. Sci.* **2023**, *13*, 10583. [[CrossRef](#)]
27. Elfving, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [[CrossRef](#)] [[PubMed](#)]
28. Tsotsos, J.K. *A Computational Perspective on Visual Attention*; The MIT Presse Books: Cambridge, MA, USA, 2011.
29. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 15–25 June 2021.
30. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable ConvNets v2: More Deformable, Better Results. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
31. Robbins, H.; Monro, S. A Stochastic Approximation Method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.