Research on Engineering Structures & Materials
www.jresm.org
MIM Research Group

Research Article

YOLOv8 algorithm improvement and lightweight system for ancient building defect detection

Haihong Jia [*,1,2,3,a], Mengyu Yu [1,b], Xiaofan Shi [1,c]

[1]*School of Smart City Engineering, Anyang Normal University, Anyang, China*
[2]*Henan Province Engineering Technology Research Center of Digital-Intelligent Building and Low-Carbon Building Materials, Anyang Normal University, Anyang, China*
[3]*Anyang City Key Laboratory for Digital Protection of Historic Buildings, Anyang Normal University, Anyang, China*

Article Info

Article History:

Received 01 Sep 2026

Accepted 12 Oct 2026

Keywords:

Ancient building defect detection;
YOLOv8;
Lightweight research;
Convolution block attention mechanism;
Coordinate attention mechanism

Abstract

To address the problems of low accuracy in slender defect feature extraction, insufficient multi-scale feature fusion, and difficult deployment of existing deep learning models, this study first improves the YOLOv8 algorithm using convolutional attention mechanism and coordinate attention mechanism, and carries out lightweight design; Furthermore, based on a lightweight model, a defect edge detection system for ancient buildings was constructed, and the model deployment and field verification were completed on the NVIDIA Jetson Orin Nano terminal. The results show that the mAP@0.5 increases from 80.4% to 95.8%, the parameter size decreases from 4.0MB to 2.7MB, and the running speed improves from 108FPS to 118FPS. Compared with other defect detection models, the proposed model achieves an average accuracy of 96.2%, a detection error of 1.2%, and a missed detection rate of 0.4%. Under normal lighting conditions, the detection accuracy for weathering and peeling, wall cracks, component defects, and paint aging reaches 98.5%, 96.7%, 94.6%, and 97.4%, respectively. Under six interference conditions, including low light, strong light overexposure, rainy blur, haze, and shadow obstruction, the accuracy remains above 90%. The proposed model effectively improves defect detection performance and provides technical support for ancient building protection.

© 2026 MIM Research Group. All rights reserved.

1. Introduction

1.1 Research Background and Significance

Ancient buildings are treasures of human civilization. They have rich historical, cultural and artistic values, and are non-renewable precious heritage [1]. Every ancient building contains the wisdom and creativity of the ancients. It can reflect the social, economic, cultural and technological levels of the time [2]. However, due to sustained weathering from the natural environment, the effects of human activities, and the aging of its own materials, various defects including cracks and weathering are prone to occur inside ancient buildings, which seriously threatens their structural safety and stability [3]. Therefore, detecting defects in ancient buildings can ensure the structural integrity of ancient buildings and promptly discover potential safety hazards, thereby providing a basis for repair decisions and improving the efficiency and accuracy of ancient building protection work [4].

To resolve the issues of low efficiency of traditional manual inspection and difficulty in identifying small defects, many studies have leveraged DNNs to detect defects in ancient buildings [5]. However, the reliability of the deep learning model in extracting the slender defect properties of

*Corresponding author: haihong_jia@outlook.com
[a]orcid.org/0009-0001-3932-8586; [b]orcid.org/0009-0002-8813-1162; [c]orcid.org/0009-0009-1611-6327
DOI: http://dx.doi.org/10.17515/resm2026-2279vk0901rs

ancient buildings is still low. The merging of features across different scales is insufficient. The high-precision model has a large amount of parameters and poses deployment challenges [6]. Therefore, it is urgent to design a method for detecting defects in ancient buildings with high-precision identification capabilities and lightweight characteristics.

As an instantaneous object detection method, You Only Look Once version 8 (YOLOv8) can meet the detection requirements for targets of different scales [7]. The Convolutional Block Attention Module (CBAM) can adaptively boost the response of key feature channels. It focuses on the defect spatial positions through the serial fusion of channel-wise attention and spatial-wise attention. This model suppresses complex background interference and strengthens the model's ability in feature extraction from defect areas [8]. Coordinate Attention (CA) mechanism can embed location priors into channel-focused attention. It learns extended dependencies and holds onto accurate positional representations via 1D feature encoding in both Horizontal and Vertical Directions (HVDs). This makes it suitable for precise positioning of slender defects such as cracks [9]. Moreover, the lightweight design can reduce the model's computational load and running time to satisfy the real-time deployment demands of terminal devices. Therefore, to address the problems of low extraction accuracy of slender defects, insufficient multi-scale fusion, and difficult model deployment in the defect detection of ancient buildings, this study chose YOLOv8n as the basic network and proposed an improved YOLOv8 algorithm and lightweight ancient building Defect Detection Model (DDM) based on CBAM and CA attention mechanisms. The reason for choosing YOLOv8n is that among the YOLOv8 series, YOLOv8n has the smallest number of parameters and the fastest inference speed, which can achieve a balance between accuracy and lightweight, in line with the constraints of ancient architectural site detection on model volume and real-time performance.

1.2 Research Status

Currently, many scholars conduct research on building defect detection. For example, Kim C's team designed a thermal defect detection method via thermal images and visible images to recognize the categories of thermal anomalies in the building enclosure. The F value of this method for different thermal defect detection was higher than 0.9000 [10]. Chen J et al. adopted binding registration algorithm and aerial image technology to detect concrete defects in buildings. This method reduced the false discovery rate to 56.8% [11]. Huo Z et al. utilized the GhostNet algorithm and bilateral filtering algorithm to locate and detect cracks in building components. The detection accuracy of this algorithm reached 0.990 [12]. To detect surface defects and damage of buildings, the Li W team proposed a detection method through the SELU Activation Function (AF) of YOLOv8 and normalization characteristics. Its mAP50 was 0.641 [13]. The Shin H team applied the layer attention network to optimize deep learning, for tackling the problem of limited accuracy in identifying building exterior wall defects. This method effectively represented the defect information of each layer of building surface images [14]. However, an analysis of the above research verified that the current construction defect detection still had problems including insufficient defect feature extraction and difficult model deployment in complex backgrounds, which still needed to be optimized.

YOLOv8 is a single-stage real-time object detection algorithm suitable for various defect detection [15]. For instance, the Liang Y team employed the YOLO model to detect the quality defects of building concrete with its best average accuracy reached 85.02% [16]. Karimi N et al. also adopted the YOLO model to detect the brick material defects of cultural heritage buildings. The average detection accuracy of this method for bricks reached 94.4% [17]. However, YOLOv8 has lower detection accuracy for defects in complex backgrounds. It cannot fulfill the demands for anomaly detection of ancient buildings. The CBAM and CAM will capture the directional features of elongated defects and can counter the challenges existing in YOLOv8. For example, Tang P et al. leveraged the attention mechanism to optimize the YOLOv8 algorithm to improve its accuracy for surface quality detection of industrial products. The results indicated that after optimization, its mAP50 reached 86.4% [18]. For improving the accuracy of part defect detection, F. Jiang's team utilized CBAM to optimize YOLOv5. The average accuracy increased by 1.5% after optimization [19]. Z. Ning's team aimed to decrease the computational burden of the YOLO, carried out

lightweight design on it, and the results verified that the lightweight model reduced 54% of the running time [20].

To sum up, the current building DDMs still have problems such as insufficient defect feature extraction capability and large model parameters that are difficult to deploy under complex background interference. To solve these issues, this study designs an IYOLOv8 ancient building defect detection algorithm based on CBAM and CA mechanisms, and achieves model light weighting through VanillaNet and simSPPF modules, thereby enhancing the defect feature extraction ability and reducing the model parameters.

1.3 Main Research Content and Innovation

Defect detection of ancient buildings faces the problems of low extraction accuracy of slender defect features, severe intricate background noise, inadequate Multi-scale Feature (MSF) integration and difficulty in deploying high-precision models. Therefore, this study employs YOLOv8 as the basic network and introduces CBAM and CA paradigm to enhance the defect feature encoding capabilities. This work then adopts VanillaNet and simSPPF modules to achieve model light weighting, and develops a visual inspection system to complete field verification. The innovation points and major contributions of the paper are enumerated below: (1) The CBAM is embedded in the YOLOv8 Backbone Network (BN) and Neck Network (NN) to adaptively enhance the defect feature response. It alleviates the disturbance of complicated masonry context to solve the problem of insufficient defect feature extraction in complex backgrounds. (2) This work introduces CA mechanism, embeds position information into channel-focused attention, and grasps remote dependencies along with exact position information using 1D feature encoding in both row-wise and column-wise directions. It improves the detection accuracy of slender defects like cracks. (3) The VanillaNet module serves to replace the Cross Stage Partial with 2 Convs fast (C2f) module to reduce computational complexity. The simSPPF module is utilized to replace the Spatial Pyramid Pooling Fast (SPPF) module, and ReLU serves to replace SiLU to eliminate the exponential operation overhead.

2. Construction of Ancient Building DDM

2.1 Ancient Building DDM

The ancient building detection model proposed in this study is based on the YOLOv8n network. YOLOv8n is the smallest version in the YOLOv8 series, with a parameter count of approximately 3.1M and low computational complexity. It can meet the real-time deployment requirements of edge terminals and is consistent with the lightweight design goal of this study. The basic structure of the model is shown in Figure 1.

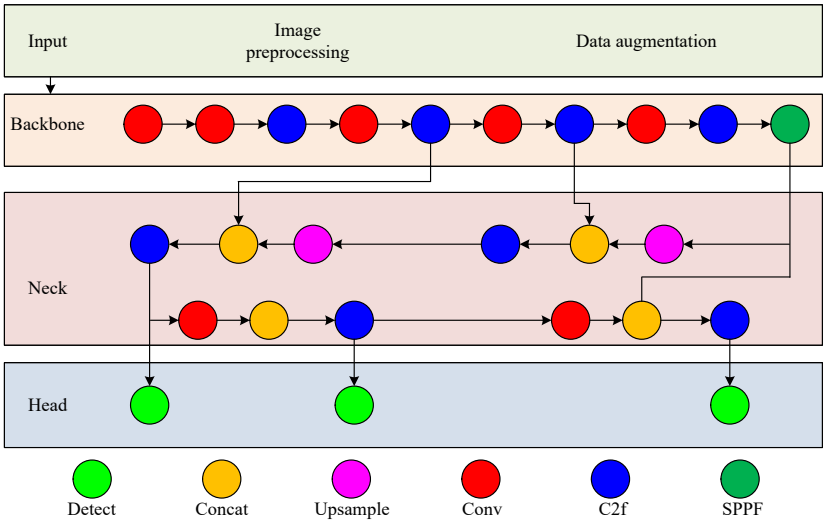


Fig. 1. Basic design of ancient building DDM

From Figure 1, the detection model consists of an input terminal, a BN, a NN, a Detection Head (DH) and an attention module. The input end is responsible for image preprocessing and data augmentation, comprising Mosaic stitching, random cropping, and color jittering and other operations to expand the variety of training instances and boost model generalization capabilities. The BN is built on the CSPDarknet53 architecture and extracts MSFs through the Convolution (Conv), the C2f and the SPPF design. The expression of its forward propagation process is displayed in Equation (1).

$$F_{backbone} = SPPF(C2f_n(\cdots(C2f_1(Conv(F_{in})))\cdots)) \tag{1}$$

In Equation (1), $F_{backbone}$ is the feature matrix output by the BN. $SPPF()$ is the SPPF module. $Conv()$ refers to the initial Conv downsampling operation. $C2f_n()$ is the nth C2f feature extraction module. F_{in} is the original image features of the input net. The NN uses the PAN-FPN framework to achieve complementary fusion of different levels of semantic information and position information through forward and backward bidirectional feature merging routes. The DH adopts a decoupled design and outputs defect category probabilities to reduce the optimization difficulty caused by multi-task coupling.

2.2 YOLOv8 Improvement Module

To address the issue of inaccurate defect recognition in YOLOv8, this study makes improvements to YOLOv8. CBAM is an efficient universal attention block consisting of two sub-modules: the Channel Attention Module (CAM) and the Spatial Attention Module (SAM). It adopts a sequential approach to adaptively weight the channel and spatial extents of the Feature Map (FM). CBAM can address the issues of severe texture-background interference and insufficient defect feature response in YOLOv8, thereby suppressing the effect of background clutter on detection precision. The basic framework of CBAM is given in Figure 2.

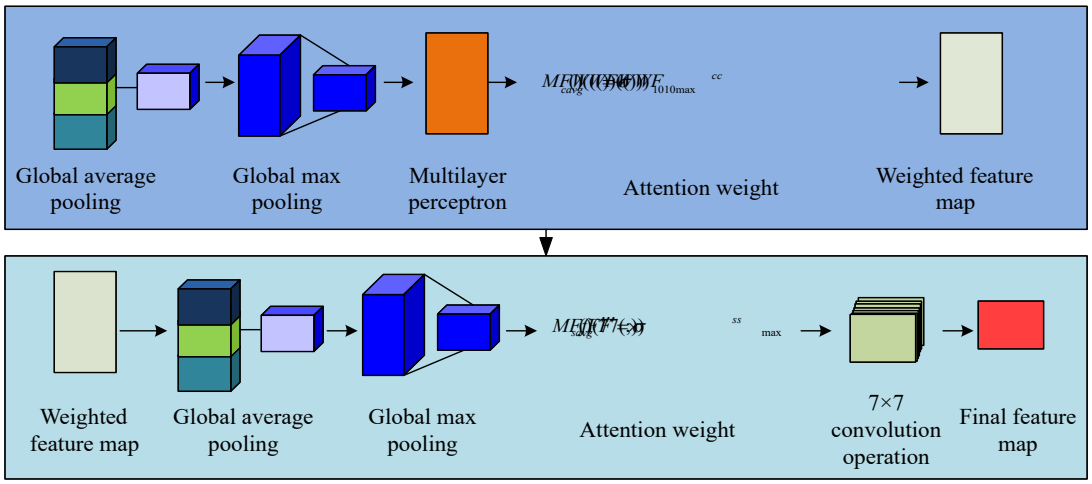


Fig. 2. Principal design of CBAM

In Figure 2, the CAM in CBAM initially applies global average pooling and global max pooling on the input FM to generate two channel descriptors, whose calculation is given in Equation (2) [21].

$$\begin{aligned} F_{avg}^{c} &= \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} F(i,j) \\ F_{max}^{c} &= \max_{i,j} F\ (i,j) \end{aligned} \tag{2}$$

In Equation (2), H is the height of the input FM. W is the width of the input FM. $F(i,j)$ refers to the pixel value of the input FM at position (i,j). F_{avg}^{c} represents the global average pooling descriptor of the channel axis. F_{max}^{c} denotes the global maximum pooled descriptor for the channel scope, and $max_{i,j}$ is the maximum value at all spatial positions (i,j) of the FM. To circumvent the loss of information induced by using only one pooling method, CAM inputs the two obtained descriptors

in parallel to a shared multi-layer perceptron, learns the nonlinear dependence between channels through axiality reduction-axiality enhancement operations. To avoid information loss caused by a single pooling operation, CAM inputs the two descriptors obtained in parallel into a shared multi-layer perceptron. The shared MLP consists of two fully connected layers, with the middle-hidden layer dimension compressed to C/r by a reduction rate of r. Nonlinear dependency relationships between channels are learned through dimensionality reduction and elevation operations, where C is the number of channels in the input feature map and r takes a value of 16. And then calculates the CAM weight, as denoted in Equation (3) [22].

$$M_c(F) = \sigma\left(W_1\left(W_0\left(F_{avg}^c\right)\right) + W_1\left(W_0\left(F_{max}^c\right)\right)\right) \tag{3}$$

In Equation (3), $W_0 \in R^{C/r}$ and $W_1 \in R^{(C\times C/r)}$ respectively represent the weight matrices of the shared MLP dimensionality reduction layer and dimensionality enhancement layer. σ is the Sigmoid AF, and $M_c(F)$ means the channel-wise attention weight map. The adaptive re-scaling of the channel scope is achieved by element-wise multiplication with the input FM. The calculation formula for the FM after CAM scaling is shown in Equation (4) [23].

$$F^{'} = M_c(F) \otimes F \tag{4}$$

F' is the FM after CAM recalibration, and F means the original input FM. After filtering the key feature channels in CAM, the SAM further filters the weighted feature again. In SAM, the weighted FMs in CAM are subjected to mean pooling and maximum pooling across the channel direction to generate two spatial signatures. The calculation formula is given in Equation (5).

$$\begin{aligned} F_avg^{s^{'}} &= 1/C\sum(c=1)^C F_c^{'} \\ F_max^{s'} &= \max_c F_c' \end{aligned} \tag{5}$$

In Equation (5), F'^{s}_{avg} denotes the average pooling descriptor of the spatial scope. F'^{s}_{max} is the maximum pooling descriptor of the spatial scope. C refers to the count of channels of the FM. F_c' represents the cth channel of the FM after CAM weighting. After concatenating the two spatial descriptors obtained, a 7×7 Conv layer and the Sigmoid AF are employed to generate the spatial attention weight map. The calculation formula is expressed in Equation (6).

$$M_s(F^{'}) = \sigma\left(f^{7\times7}\left(F_{avg}^{'s}; F_{max}^{'s}\right)\right) \tag{6}$$

In Equation (6), $M_s(F')$ is the spatial attention weight map, and $f^{7\times7}$ represents the Conv operation with a kernel size of 7×7. Finally, the output of CBAM is calculated through $F^{''} = M_s(F^{'}) \otimes F^{'}$. After optimizing YOLOv8 using CBAM, the basic framework of CBAM-YOLOv8 is visualized in Figure 3.

In Figure 3, the CBAM is inserted after the C2f module of the BN and in the feature fusion path of the NN. By embedding the CBAM at the output tail of the BN, the channel response of defect features at different scales can be enhanced. Then, the CBAM is embedded in the downward and upward blending branches of the PAN-FPN of the YOLOv8 NN to optimize the attention allocation during multi-scaled representation. By embedding the CBAM in the YOLOv8 network, the model's proficiency in extracting defect feature information and suppress the background is improved. The DD accuracy of YOLOv8 in crowded scenes is enhanced. Although the CBAM can effectively enhance the channel response and spatial positioning ability of defect features, it still has limitations when dealing with elongated defects [24]. Therefore, Given the introduction of CBAM, this study further introduces the CA mechanism. It inserts positional details into channel-focused attention, and learns long-range dependencies and preserves accurate positional details via 1D feature encoding in the HVDs. The basic process is presented in Figure 4.

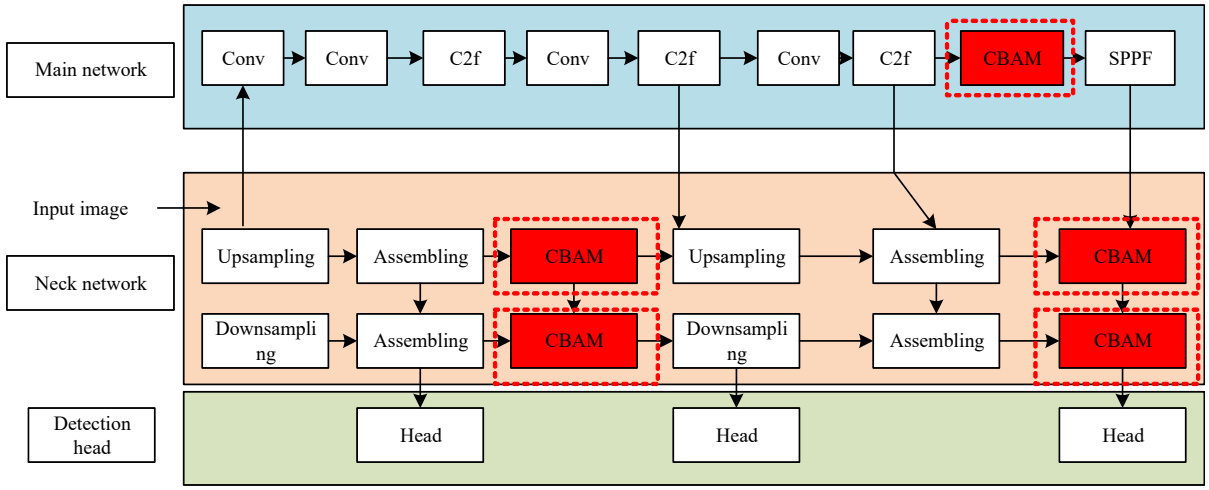


Fig. 3. Basic framework of CBAM-YOLOv8

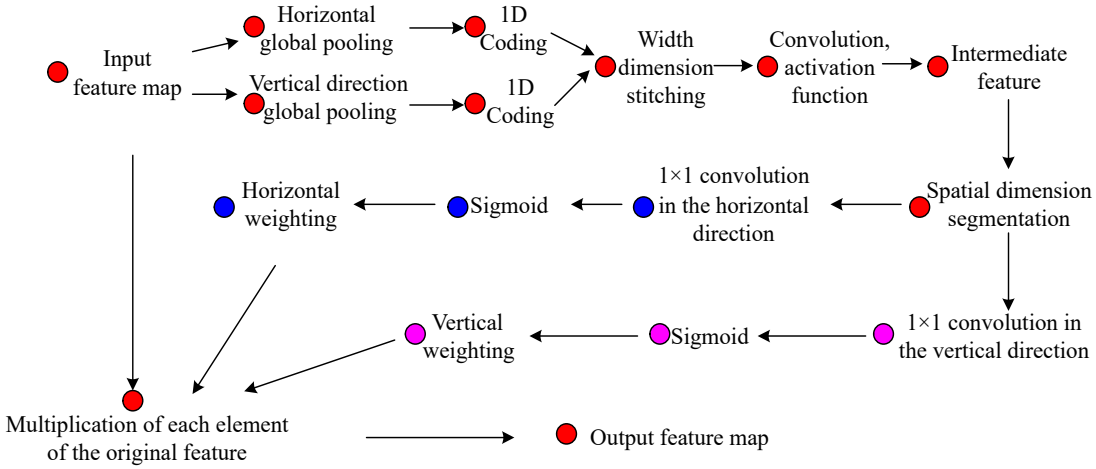


Fig. 4. Basic process of the CA mechanism

In Figure 4, after the introduction of the CA mechanism, CA first extracts spatial coordinate information through two-directional Global Pooling (GP) operations. The horizontal GP compresses the FMs of each channel in the height direction, while the vertical GP compresses the FMs of each channel in the width direction. The compression formula is displayed in Equation (7) [25].

$$\begin{cases} z_c^h = \dfrac{1}{W} \displaystyle\sum_{0 \le i \le W} x_c(h, i) \\ z_c^w = \dfrac{1}{H} \displaystyle\sum_{0 \le j \le H} x_c(j, w) \end{cases} \tag{7}$$

In Equation (7), z_c^h equals the 1D feature encoding of the cth channel in the horizontal axis. z_c^w is the 1D feature encoding of the cth channel in the vertical direction. $x_c(h, i)$ denotes the feature score of the cth channel at height h and width i. $x_c(j, w)$ means the feature score of the cth channel at height j and width w. The two calculated 1D feature encodings respectively retain the position distribution signals of the defect in the HVD, avoiding the loss of position data resulting from the traditional 2D GP. Subsequently, the two feature encodings are merged in the width scope to obtain the intermediate feature, as given in Equation (8).

$$f = \delta(F_1([z^h, z^w])) \tag{8}$$

In Equation (8), f is the fused intermediate feature, while z^h and z^w are 1D feature encodings in the HVDs across all channels. $F_1()$denotes the 1×1 Conv operation, and $\delta()$is the nonlinear AF. This Conv operation achieves channel domain reduction and feature fusion, enabling interaction between the positional information in the HVDs. Subsequently, the intermediate feature is split along the spatial domain into two independent features, which are transformed into attention weights with the same plane as the input FM through 1×1 Conv and the Sigmoid function. The expression is as displayed in Equation (9).

$$\begin{cases} g^h = \sigma(F_h(f^h)) \\ g^w = \sigma(F_w(f^w)) \end{cases} \tag{9}$$

In Equation (9), $F_h()$and $F_w()$are 1×1 Conv operations in the HVDs. σ is the Sigmoid AF. g^h and g^w are the attention weights in the HVDs. f^h and f^w is the HVD of the intermediate features after being split along the spatial plane. Finally, the output of the CA module is estimated, as expressed in Equation (10).

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{10}$$

In Equation (10), $y_c(i,j)$ is the feature score of the cth channel output by the CA module at position (i,j). $x_c(i,j)$ means the original feature score of the cth channel in the input FM at site (i,j). $g_c^h(i)$ denotes the attention weight of the cth channel at position i in the horizontal direction. $g_c^w(j)$ indicates the attention factor of c th channel at position j in the vertical direction. The CA mechanism explicitly embeds positional information into the CAM, enabling the model to perceive the precise position distribution of defects in both the HVDs. By embedding the CA mechanism into the deep feature fusion path of the NN PAN-FPN, it captures the long-range dependencies of slender defects like cracks. Through the collaborative effect of CBAM and CA, the model utilizes CBAM to augment the overall response of the defect area in the shallow architecture and CA to precisely locate the spatial distribution of slender defects in the deep network, thereby achieving precise detection of building defects.

2.3 Lightweight Improvement

After introducing the CBAM and CA mechanisms, the approach involves abundant parameters along with high temporal computational costs. To lower the parameter volume and boost computational efficiency, this study has carried out lightweight processing on the model and designed the IYOLOv8. The C2f module in YOLOv8 enhances the feature extraction ability through residual connections and multi-branch frameworks, but the complex branch topology leads to computational redundancy [26]. This study replaces the C2f module with a lightweight VanillaNet module to eliminate the computational redundancy caused by residual connections and reduce the model complexity. The primary composition of the VanillaNet module is given in Figure 5.

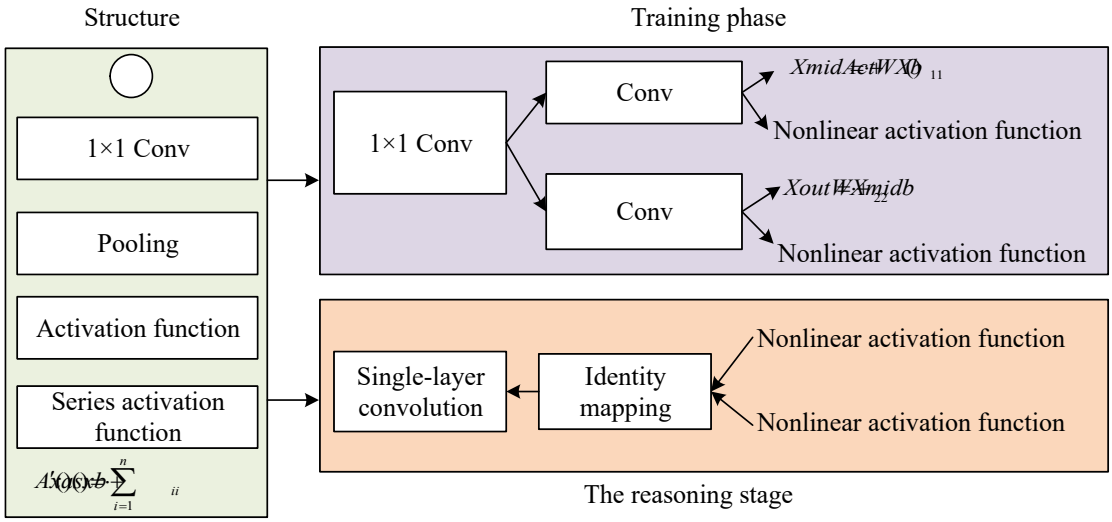


Fig. 5. Basic layout of the VanillaNet module

In Figure 5, the VanillaNet module is composed of a series of Convs, Pooling layers (Pool) and AFs. In the Convs, 1×1 Conv kernels are adopted instead of the standard 3×3 Conv kernels to reduce the overlap of receptive fields and thereby decrease the computational load. The input FM is passed through the VanillaNet module for forward propagation. The forward propagation process is expressed by Equation (11).

$$Y = Pool(Act(Conv_{1\times1}(X))) \tag{11}$$

In Equation (11), $Conv_{1\times1}$ is a 1×1 Conv operation, which is utilized to adjust the channel scope. $Act()$is a concatenation AF, and $Pool()$denotes a downsampling pooling operation. Moreover, to address the issue of the lack of nonlinearity caused by the extremely simple layout in the VanillaNet module, a concatenation AF is introduced. By stacking multiple AFs, the nonlinearity of feature expression is enhanced. The form of the concatenation AF is given in Equation (12).

$$A^{'}(x) = \sum_{i=1}^{n} a(s_i \cdot x + b_i) \tag{12}$$

In Equation (12), $A'(x)$ is the output of the series AF. x is the fed feature value. n is the count of stacked transfer functions. s_i and b_i are the scaling coefficient and bias term of the ith AF, this study used ReLU as the basis activation function. Moreover, the VanillaNet module adopts a deep training strategy to balance training accuracy and inference efficiency. During the training phase, each 1×1 Conv is split into two consecutive Convs, with a nonlinear AF inserted in the middle. The output calculation formula of the Conv is shown in Equation (13).

$$\begin{cases} Xmid = Act(W_1 \cdot X + b_1) \\ Xout = W_2 \cdot Xmid + b_2 \end{cases} \tag{13}$$

In Equation (13), $Xmid$ and $Xout$ are the intermediate output features of the first and second Convs. W_1 and W_2 are the weight matrices of the two Convs. b_1 and b_2 are the bias terms of the two Convs.

According to equation (12), when the weighting coefficients and bias terms converge to $A_i = 1$, $s_i = 1$, and $b_i = 0$, the concatenated activation function degenerates into an identity map, i.e. $A'(x) = x$. At this point, the two consecutive 1×1 convolutional layers in equation (13) can be mathematically compressed into a single convolutional layer during the inference stage. The derivation is as follows: assuming the input feature is X, the weight matrices and bias terms of the two convolutional layers are W_1, b_1, and W_2, b_2, respectively. After the activation function degenerates into an identity mapping, the forward propagation process is $Y = W_2(W_1X + b_1) + b_2$, and Y is the output feature of the compressed convolutional layer. Unfolding the parentheses yields $Y = (W_2W_1)X + (W_2b_1 + b_2) = WX + b$.

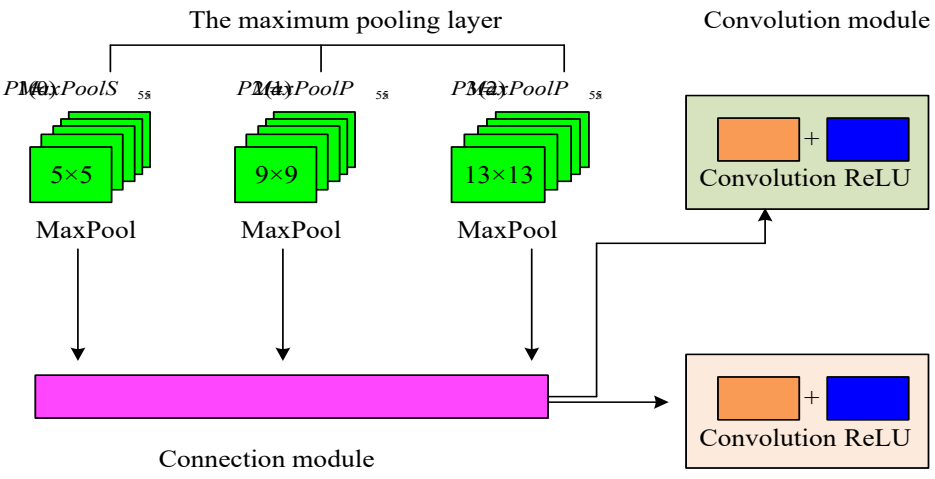


Fig. 6. Basic design of the simSPPF module

The composite operation of two consecutive 1×1 convolutions is equivalent to a single 1×1 convolution of the weight matrix $W = W_2W_1$ and bias $b = W_2b_1 + b_2$. Due to the fact that 1×1 convolution is only a linear transformation in the channel dimension, the size of the convolution kernel remains unchanged before and after the composite operation. Therefore, the number of network layers in the inference stage is halved, while the output features remain strictly unchanged. During the training phase, the dual layer convolution and nonlinear activation are retained to ensure the expressive ability of the model; In the inference stage, the structure is reparametrized and compressed into a single-layer convolution, eliminating redundant parameters and computational complexity.

VanillaNet performs only a single-layer Conv operation during inference, thereby reducing the computational complexity. However, merely replacing the C2f module is not sufficient to fulfill the live deployment specifications of terminal devices. The SPPF at the end of the BN also has a computing redundancy problem [27]. Moreover, substituting the SPPF in YOLOv8 with a more lightweight and efficient simSPPF module further reduces the model's parameter quantity. The layout of simSPPF is displayed in Figure 6.

In Figure 6, the simSPFF module is composed of 3 max Pools, 2 Conv modules and 1 connection module. In the max Pools, a pooling kernel with a size of 5×5 is utilized for serial stacking. Three consecutive downsampling operations are performed to achieve multi-scale receptive field fusion. The equivalent sensory spans are 5×5, 9×9 and 13×13. The output of the three Pools is given in Equation (14).

$$\begin{cases} P1 = MaxPool_{5\times5}(S0) \\ P2 = MaxPool_{5\times5}(P1) \\ P3 = MaxPool_{5\times5}(P2) \end{cases} \tag{14}$$

In Equation (14), $P1$, $P2$ and $P3$ denote the responses of the first, second and third max Pools. $MaxPool()$is the maximum pooling operation with a kernel size of 5×5. It refers to the output feature of the first Conv module. Among the two Conv modules, the first Conv module contains a 1×1 Conv and a ReLU AF. The input feature image is compressed through the first Conv module, with the compression calculation expressed in Equation (15).

$$S0 = Re\,L\,U(Conv_{1\times1}(X)) \tag{15}$$

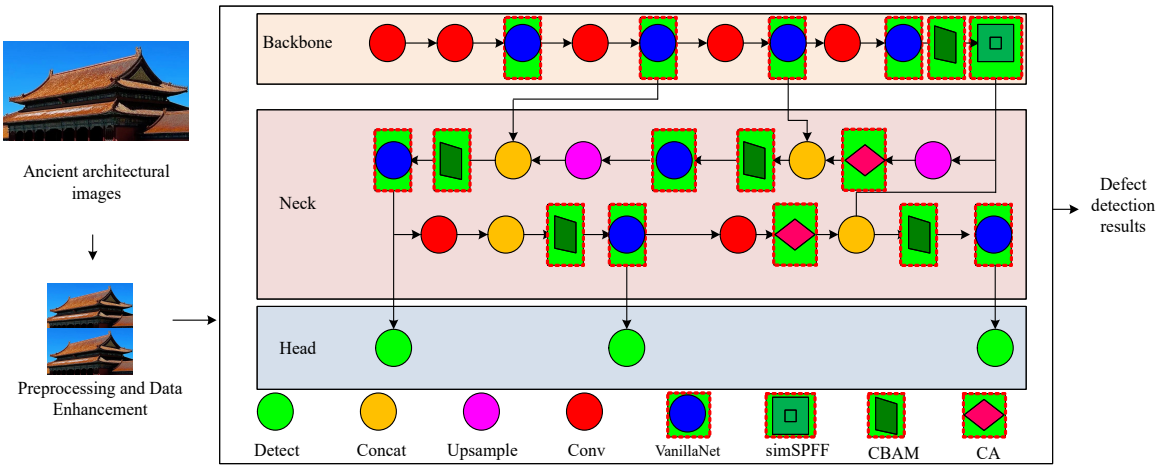


Fig. 7. Basic process of the ancient building defect recognition model based on CBAM-CA-IYOLOv8 (Image source: self taken)

In Equation (15), $Conv_{1\times1}()$is a Conv operation with a kernel size of 1×1. $Re\,L\,U()$is the AF. After MSF fusion, the second Conv module increases the channel number by performing 1×1 Conv, batch norm, and ReLU AF. In the connection module, the original features and three pooled features are fused using channel scope concatenation to generate a joint FM containing multi-scale context information. Through this connection module, the model can simultaneously capture fine-grained

details and multi-scale context information, providing rich feature representations for the subsequent DH. The standard ReLU has a zero gradient in the negative interval, theoretically posing a risk of 'death ReLU'. However, simSPPF is located at the end of the backbone and only contains 3 pooling layers and 2 convolution modules, with fewer ReLU layers and limited risk of gradient accumulation deactivation; And each ReLU is preceded by a batch normalization layer, with the output distribution concentrated around zero, which can prevent neurons from falling into the negative half region for a long time. Therefore, the designed module structure can effectively suppress the impact of the "death ReLU" problem on gradient propagation.

The output FM after scopeality increase by two Conv modules is input into the NN of YOLOv8. After the improvement and lightweight design of YOLOv8, the basic process for identifying ancient building defects is depicted in Figure 7.

In Figure 7, the red dotted line represents the optimized design of YOLOv8. When identifying the defects of ancient buildings, the input image of the ancient building defect is first preprocessed and data augmented. Then, the transformed image is introduced into the main framework of YOLOv8. In this main network, the simple Conv arrangement of the VanillaNet module is utilized to extract MSFs. The AF and serial pooling in the simSPPF module are used to achieve efficient MSF fusion. Moreover, the MSFs are adaptively weighted by the CBAM attention mechanism and spatially focused to enhance the defect feature response and suppress complex background interference such as brick and stone textures. Then, the CA mechanism is used to encode the 1D coordinates of the slender defects in the HVDs, model distant dependencies and maintain exact location cues, and achieve precise positioning of slender defects such as cracks. Finally, the YOLOv8 decoupled DH outputs the category, confidence, and bounding box coordinates of the defect, completing the defect detection of ancient buildings.

2.4 Lightweight Detection System

In order to apply the improved CBAM-CA-IYOLOv8 algorithm to the field inspection of ancient buildings, this study constructed a defect edge detection system for ancient buildings based on a lightweight model. The complete deployment architecture diagram is shown in Figure 8.

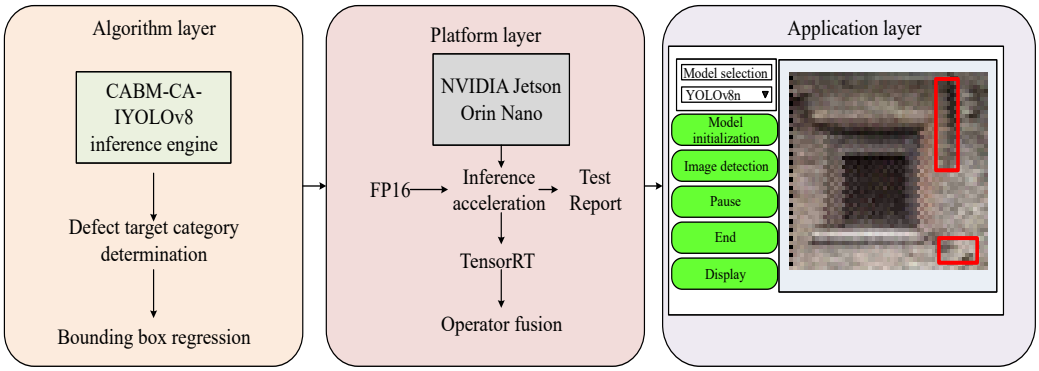


Fig. 8. Detection system architecture

As shown in Figure 8, the system consists of three parts: algorithm layer, platform layer, and application layer. Deploying CBAM-CA-IYOLOv8 inference engine at the algorithm layer, responsible for category determination and bounding box regression of defect targets; The platform layer takes NVIDIA Jetson Orin Nano as the core of edge computing, converts the model weight to FP16 precision, and uses TensorRT for reasoning acceleration and operator fusion; The application layer is based on PyQt to develop a detection interface, which enables the import of detection images, real-time visualization of defect boxes, statistical analysis of defect categories, and export of detection reports.

The basic process of defect detection in the system is as follows: first, obtain the image of the ancient building to be inspected through USB camera and drone shooting, and uniformly scale it to 640 × 640 pixels. Subsequently, the image is preprocessed by the platform layer and input into the

inference engine. CBAM-CA-IYOLOv8 outputs the defect category, confidence level, and bounding box coordinates; Finally, the application layer overlays the detection results on the original image and generates statistical information and detection records based on defect categories.

3. Results

3.1 Dataset and Experiment Setup

For analyzing the outcome of the proposed CBAM-CA-IYOLOv8, the StructuralCrackDataset open-source dataset (https://www.modelscope.cn/datasets/jiange1236/StructuralCrackDataset) from the ModelScope platform was selected as the test dataset. This dataset contained various annotations of building arrangement cracks. The dataset also provided bounding box annotations, polygon annotations, line segment annotations, and text descriptions, which were in line with the multi-class detection and fine positioning theme of ancient building defects in this study. This dataset is only used to test the general detection ability of the model for crack type defects. The dataset is divided into training, testing, and validation sets in a ratio of 7:2:1. The environment configuration during the test is displayed in Table 1.

Table 1. Environmental configuration

Project	Configuration	Project	Configuration
OS	Ubuntu 22.04 LTS	CPU	Intel Core i9-13900K @ 3.0GHz
GPU	NVIDIA GeForce RTX 4090	Deep learning framework	PyTorch 2.1.0
CUDA	CUDA 12.1	Python	Python 3.10

During the testing process, the parameter configuration of the algorithm was as follows: this study conducted comparative experiments using YOLOv8n as the baseline model, the input image resolution was configured to 640×640, the batch size was fixed at 16, the starting learning rate was established at 0.01, and the SGD optimizer was employed, the training rounds were 300, the learning rate scheduling adopted the cosine annealing strategy, and the warm-up rounds were 3, the channel reduction rate for shared MLP in CBAM is set to 16, and the size of the SAM space convolution kernel is set to 7 × 7. The actual test results of the model were analyzed using on-site pictures collected from ancient buildings. Each picture had a resolution of 1920×1080 and was in the JPG format.

The measurement of model running speed is based on the native PyTorch 1.10.0 inference framework, without TensorRT or FP16 acceleration enabled. When measuring, the input image size is 640 × 640, and the batch size is set to 1 to simulate the real-time detection scene of a single image. Each round of inference time includes the entire process of image preprocessing, network forward propagation, and non-maximum suppression post-processing; Preheat 100 images before testing, then iteratively infer and average the 500 images in the test set to obtain the final frame rate value.

Table 2. Class distribution of the ancient-building defect dataset

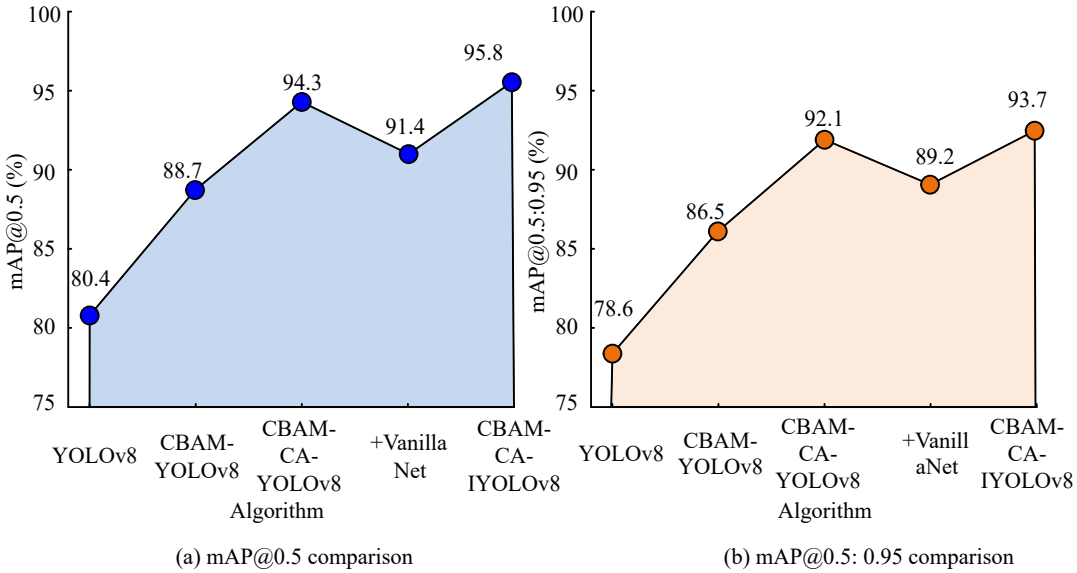
Defect category	Training set/piece	Verification set/piece	Test set/piece	Total/piece	Proportion/%
Weathering and peeling off	2680	760	380	3 820	42.4
Wall cracks	1560	440	225	2 225	24.7
Component defects	1130	320	160	1 610	17.9
Paint aging	920	260	130	1 310	14.5
Total bounding box	6290	1 780	895	8 965	100

During actual testing, the study selected the Ming and Qing style brick-wood arrangement ancient building complex as the object, covering various typical parts such as walls, wooden components

and paint surfaces. The test images were collected on-site using a Sony A7R IV digital camera and a DJI Mavic 3 drone. The shooting time was concentrated from September to November 2024, covering three natural lighting conditions of sunny, cloudy, and cloudy days. A total of 1,847 original images were obtained, and 1,200 valid samples were retained for testing after screening. The sample resolution was 1920×1080, the format was JPG, the shooting distance was 1.5-15m, including multi-angle viewing angles like front view, top view and strabismus. Defect samples were jointly marked by heritage restoration experts and inspectors, and were divided into four categories: weathering and peeling, wall cracks, component defects, and paint aging. The dataset is divided into a training set, a validation set, and a testing set in a ratio of 7:2:1, with 840 pieces in the training set, 240 pieces in the validation set, and 120 pieces in the testing set. The model is trained on the training set, and the validation set is used for hyperparameter selection and early stopping. When dividing, sample by defect category to ensure that the distribution of categories in each subset is consistent with the overall distribution. The statistics of bounding box instances for various defects in the dataset are shown in Table 2.

3.2 Ablation Test

First, an ablation test was conducted on the CBAM-CA-IYOLOv8 algorithm to verify that the added attention mechanism and lightweight design could improve the performance of YOLOv8. The Mean Average Precision at IoU threshold 0.5 (mAP@0.5) and mAP@0.5: 0.95 of the CBAM-CA-IYOLOv8 algorithm were compared, with the findings shown visualized Figure 9.



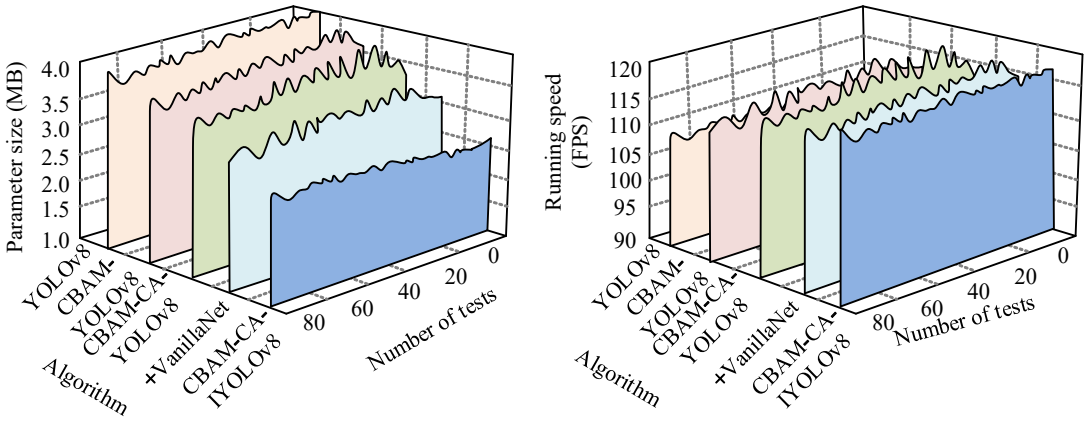
(a) mAP@0.5 comparison

(b) mAP@0.5: 0.95 comparison

Fig. 9. Contrast between mAP@0.5 and mAP@0.5:0.95

From Figure 9(a), the mAP@0.5 of the YOLOv8 was 80.4%. After introducing CBAM, it increased to 88.7%, and further introducing CA reached 94.3%. After using VanillaNet for light weighting, it dropped to 91.4%. Finally, CBAM-CA-IYOLOv8 recovered to 95.8%. Compared with the YOLOv8 algorithm, it improved by 15.4%. From Figure 9(b), mAP@0.5:0.95 from the YOLOv8 algorithm was 78.6%, successively increasing to 87.9% for CBAM-YOLOv8 and86.5% for CBAM-CA-YOLOv8. After light weighting with VanillaNet, it dropped to 89.2%. Finally, CBAM-CA-IYOLOv8 reached 93.7%.

The sequential introduction of CBAM and CA notably enhanced the feature extraction ability of YOLOv8 for ancient building defects. Although VanillaNet light weighting caused periodic accuracy fluctuations, the final model still achieved higher comprehensive performance through optimization methods such as simSPPF in the gradual addition module stage. Comparing the parameter quantity and running speed performance of CBAM-CA-IYOLOv8, the results are given in Figure 10.



(a) Comparison of Algorithm Parameters

(b) Comparison of Algorithm Running Speed

Fig. 10. Comparison of algorithm parameters and latency

From Figure 10(a), the parameter size of the YOLOv8 algorithm was approximately 3.5MB. After introducing CBAM, it increased to approximately 4.0MB. Continuing to introduce CA, it remained at approximately 4.0MB. After replacing the C2f module with VanillaNet, the parameter size dropped significantly to approximately 2.5MB. Finally, CBAM-CA-IYOLOv8 stabilized at approximately 2.7MB. The parameter size decreased from 4.0MB to 2.7MB. From Figure 10(b), the baseline running speed of YOLOv8 was approximately 108FPS, CBAM-YOLOv8 was approximately 110FPS, CBAM-CA-YOLOv8 was approximately 112FPS. After introducing the VanillaNet module, the running speed was 115FPS, and finally, CBAM-CA-IYOLOv8 reached approximately 118FPS. The running speed increased from 108FPS to 118FPS. From the above results, the VanillaNet and simSPPF introduced in this study reduced the approach parameter size and improved the inference speed while maintaining high reliability, which could meet the real-time deployment requirements of terminal devices. The summary of ablation tests for the above indicators is shown in Table 3.

Table 3. Summary of ablation tests

Algorithm	mAP@0.5 (%)	Parameter quantity (MB)	Running speed (FPS)	Calculation volume (GFLOPs)	Memory usage (MB)
YOLOv8n	80.4	3.5	108	8.7	1890
CBAM-YOLOv8	88.7	4	110	9.4	2140
CBAM-CA-YOLOv8	94.3	4	112	9.6	2180
CBAM-CA-YOLOv8+VanillaNet	91.4	2.5	115	5.9	1560
CBAM-CA-IYOLOv8	95.8	2.7	118	6.3	1620

According to Table 3, compared to the YOLOv8n baseline, CBAM-CA-IYOLOv8 mAP@0.5 Improved by 15.4%, the parameter count decreased from 3.5MB to 2.7MB, and the running speed increased from 108 FPS to 118 FPS, achieving optimization in detection accuracy, model size, and inference speed. And after introducing CBAM and CA attention mechanisms, the computational complexity of the model increased from 8.7 GFLOPs to 9.6 GFLOPs, an increase of about 10%, and the memory usage increased synchronously, which is a reasonable cost for improving accuracy; After adopting VanillaNet and simSPPF for light weighting, the computational load was reduced to 6.3 GFLOPs, a decrease of 27.6% compared to the baseline, and the memory usage was reduced by about 270 MB, which is consistent with the trend of parameter changes.

In order to further clarify the delay composition of the model, this study conducted decomposition testing on the inference process of CBAM-CA-IYOLOv8 on RTX 4090. The end-to-end delay was divided into three parts: image preprocessing, network forward inference, and NMS post-processing. The results are shown in Table 4.

Table 4. Latency breakdown of CBAM-CA-IYOLOv8 on RTX 4090

Delay component	Time consumption/ms	Proportion/%
Image preprocessing	1.2	14.2
Network forward reasoning	5.8	68.5
NMS post-processing	1.5	17.3
End to end total	8.5	100

According to Table 4, network forward inference accounts for 68.5% of the total delay and is the main time-consuming step; Post processing of NMS accounts for 17.3%, while pre-processing accounts for 14.2%. Lightweight design mainly compresses the forward inference time, while NMS and preprocessing time have a weak relationship with model size, which explains the phenomenon that the end-to-end frame rate improvement of lightweight models is smaller than the parameter reduction. And in order to determine the reasonable size of the SAM space convolution kernel, this study compared the detection performance of the model with convolution kernel sizes of 3 × 3, 5 × 5, 7 × 7, and 9 × 9, respectively. The results are shown in Table 5.

Table 5. Effect of SAM convolution kernel size on model performance

Convolutional kernel size	mAP@0.5 (%)	Parameter quantity (MB)	Running speed (FPS)
3×3	93.6	2.6	121
5×5	94.9	2.7	119
7×7	95.8	2.7	118
9×9	96.0	2.9	115

According to Table 5, as the size of the convolution kernel increases, the model's perception range of defect spatial distribution expands, mAP@0.5 From 93.6% of the 3 × 3 convolution kernel, it has increased to 94.9% of 5 × 5 and 95.8% of 7 × 7. After increasing the convolution kernel to 9 × 9, mAP@0.5 Only slightly increased to 96.0%, with a slower growth rate, while the parameter size increased from 2.7MB to 2.9MB, and the running speed decreased from 118 FPS to 115 FPS. The above results indicate that the 7 × 7 convolution kernel achieves the optimal balance between detection accuracy, parameter quantity, and inference speed, which meets the comprehensive requirements of detection accuracy and lightweight constraints in this study. Therefore, the spatial convolution kernel size of SAM is determined to be 7 × 7.

3.3 Comparison with Advanced Models

After conducting ablation tests, the CBAM-CA-IYOLOv8 DDM was compared with the DDMs through Distribution Shift Convolution-YOLOv 8 (DSConv-YOLOv8), the DDM via Multi-Scale Ensemble Booster-YOLOv 11 (MEB-YOLOv11), and the DDM via UNet-YOLOv 8 (UNet-YOLOv8). This was done to verify that the performance of the ancient building DDM proposed in this study was superior to that of other advanced models.

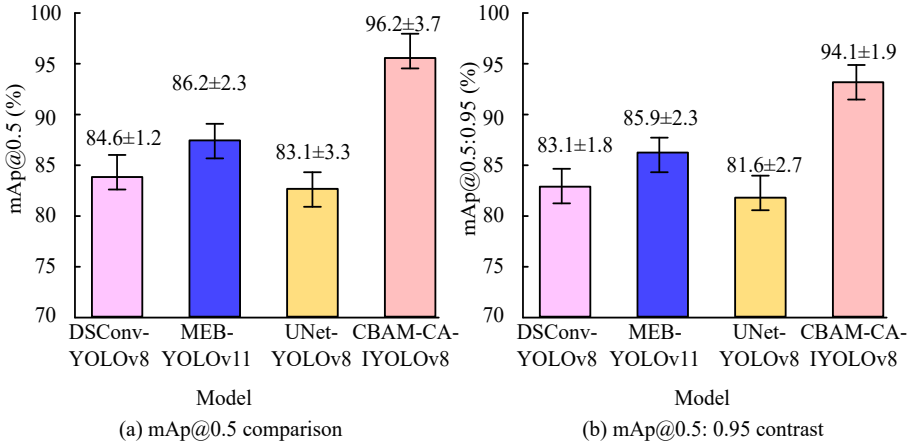
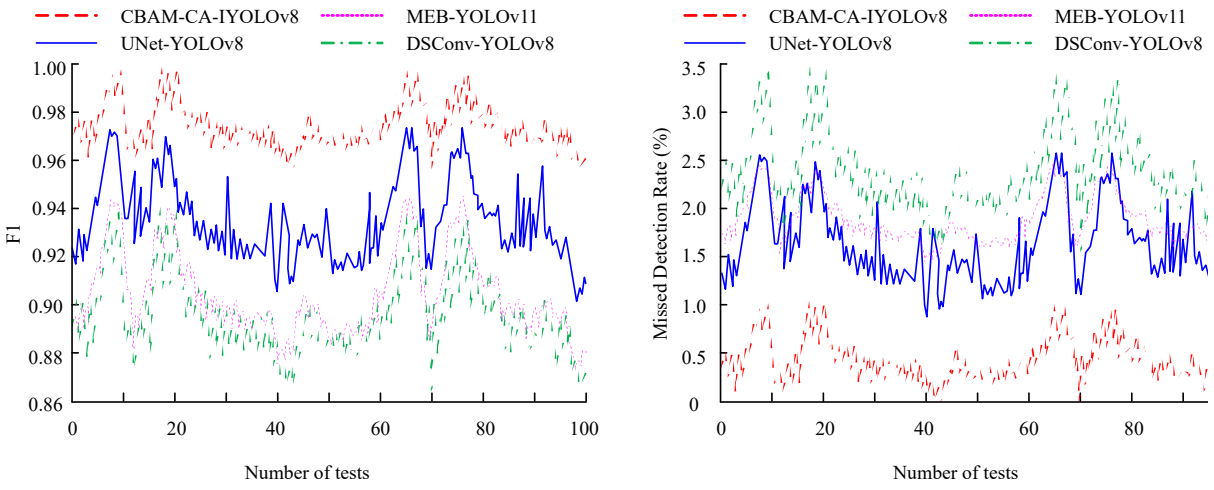


Fig. 11. mAP@0.5 and mAP@0.5:0.95 of the model

To ensure the statistical reliability of the comparison results, each model was independently trained 5 times using 5 different random seeds (0, 1, 2, 3, 4) and evaluated separately on the same test set mAP@0.5 and mAP@0.5 0.95, first for the four models mAP@0.5 and mAP@0.5 Comparing with 0.95, the result is shown in Figure 11.

From Figure 11(a), the mAP@0.5 of DSConv-YOLOv8 was 84.6% ± 1.2%, that of MEB-YOLOv11 was 86.2% ± 2.3%, that of UNet-YOLOv8 was 83.1% ± 3.3%, and that of CBAM-CA-IYOLOv8 was 96.2% ± 3.7%. CBAM-CA-IYOLOv8 had the highest mAP@0.5. From Figure 11(b), the mAP@0.5 distribution of DSConv-YOLOv8, MEB-YOLOv11, UNet-YOLOv8 and CBAM-CA-IYOLOv8 was 83.1% ± 1.8%, 85.9% ± 2.3%, 81.6% ± 2.7% and 94.1% ± 1.9%. These results indicated that CBAM-CA-IYOLOv8 outperformed the other three leading models for detection accuracy, verifying the outcomes of the CBAM and CA and lightweight design for the defect detection of ancient buildings. Additionally, the F1 score and False Negative Rate (FNR) of the models were compared, with the results visualized in Figure 12.

From Figure 12(a), the F1 score of CBAM-CA-IYOLOv8 remained between 0.96 and 1.00 throughout the entire testing process, with an average value of 0.97, having the smallest fluctuation and the highest average value. The average F1 scores of UNet-YOLOv8, MEB-YOLOv11, and DSConv-YOLOv8 models were 0.94, 0.92, and 0.89. From Figure 12(b), the FNR of CBAM-CA-IYOLOv8 remained between 0.2% and 1.0%, with an average FNR of 0.4%, being the lowest among the four models. From the above results, CBAM-CA-IYOLOv8 enhanced the response of defect feature channels through CBAM and suppressed complex background interference, and captured the long-distance dependency of slender defects through CA, effectively reducing the risk of false negatives. Finally, the Detection Errors (DEs) of the four models were compared, with the findings illustrated in Figure 13.



(a) Comparison of F1 scores

(b) Comparison of missed detection rates

Fig. 12. F1 score and missed detection rate

The normalized error value in the figure is calculated using $e_i = (FN_i + FP_i)/max(G_i, 1)$, where G_i represents the number of true defect targets in the image, FN_i is the number of missed targets, FP_i is the number of false targets, and e_i is the normalized error value. From Figure 13(a), the DE of DSConv-YOLOv8 was 5.7%. Among the 200 test pictures, the distribution of error recognition was dense and had a large amplitude, and correct recognition and incorrect recognition occurred alternately. From Figure 12(b), the DE of UNet-YOLOv8 was 4.8%, and the count of false recognitions was reduced compared with DSConv-YOLOv8. In Figure 13(c), the DE of MEB-YOLOv11 was 3.2%, the amount of false recognitions was further reduced, and the frequency of occurrence was considerably reduced. In Figure 12(d), the DE of CBAM-CA-IYOLOv8 was 1.2%. The number of incorrect identifications was the smallest and only occurred locally. Correct identifications absolutely dominated. The CBAM-CA-IYOLOv8 model proposed in this study had the lowest DE and its overall performance was better than other DDMs. And then compare the model with mature models YOLOv5n, YOLOv7-minily, baseline YOLOv8n, MobileNetV3-YOLOv8, and

YOLOv9-t designed for edge devices to verify the performance of the model. The results are shown in Table 6.

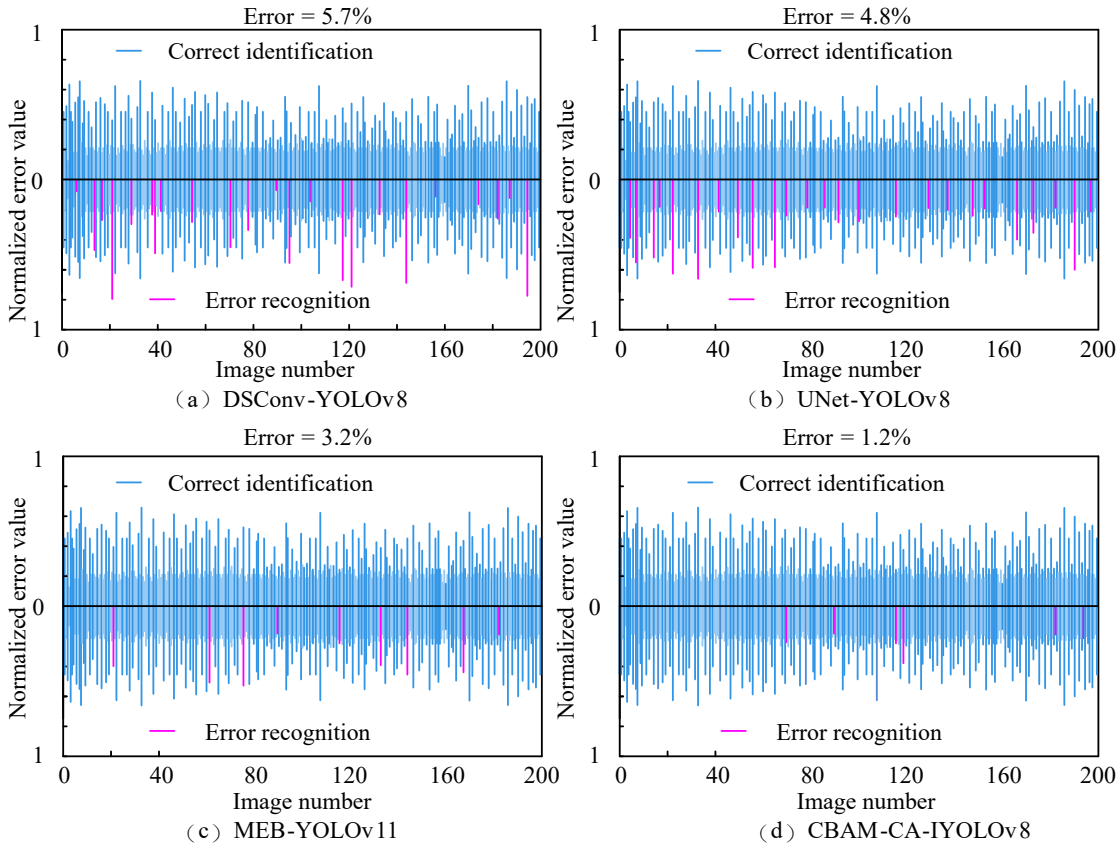


Fig. 13. Comparison of DEs

Table 6. Overall comparison results of model performance

Model	Parameter quantity (MB)	FLOP (G)	mAP@0.5 (%)	F1	Speed (FPS)
YOLOv5n	3.8	7.2	82.3	0.9	112
YOLOv7-tiny	6.2	13	84.7	0.91	105
YOLOv8n	3.5	8.7	80.4	0.89	108
MobileNetV3-YOLOv8	2.4	5.1	78.9	0.88	125
YOLOv9-t	5.1	10.8	86.5	0.92	110
DSConv-YOLOv8	4.6	9.8	84.6	0.89	106
MEB-YOLOv11	5.4	11.6	86.2	0.92	102
UNet-YOLOv8	8.9	15.2	83.1	0.94	95
CBAM-CA-IYOLOv8	2.7	6.3	96.2	0.97	118

According to Table 6, among the classic lightweight models, YOLOv9-t has the highest accuracy of 86.5% but a parameter size of 5.1MB, while MobileNetV3-YOLOv8 has the fastest speed at 125 FPS but insufficient accuracy; The accuracy of the three cutting-edge improved models ranges from 83.1% to 86.2%. CBAM-CA-IYOLOv8 achieved 96.2% accuracy with only 2.7MB of parameters mAP@0.5 Compared with F1 of 0.97, the accuracy of the optimal comparison model is improved by more than 10 percentage points, and the speed is also higher than DSConv-YOLOv8, MEB-YOLOv11, and UNet-YOLOv8, achieving the optimal balance between accuracy, model size, and inference speed.

3.4 Actual Test Results

After verifying that the model was better than the cutting-edge model, the actual test results of the CBAM-CA-IYOLOv8 ancient building DDM were analyzed. The CBAM-CA-IYOLOv8 model was

employed to detect different defect types, and the detection accuracy, F1 score, recall rate and FDR of various defects were compared. The results are displayed in Figure 14.

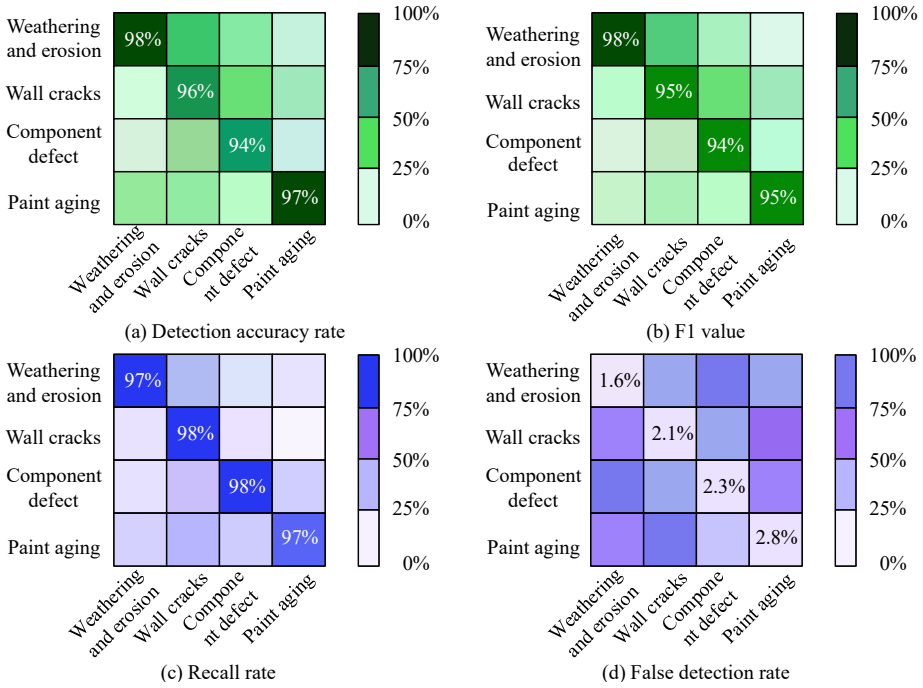


Fig. 14. Analysis of actual test results

From Figure 14(a), the detection accuracy of weathering and peeling was 98%, wall cracks were 96%, component defects were 94%, and paint aging was 97%. The accuracy of weathering and peeling was the highest, and the component defects were the lowest. The difference between the two was 4%. From Figure 14(b), the F1 scores for weathering erosion, wall cracks, component defects and paint aging defects were 98%, 95%, 94% and 95%. In Figure 14(c), when the model detected different types of defects, its recall rates were higher than 95%. In Figure 14(d), when the model detected four types of defect detection, its FDRs were 1.6%, 2.1%, 2.3% and 2.8%. Then the robustness of the CBAM-CA-IYOLOv8 model and its detection accuracy were compared. The findings are illustrated in Table 7.

Table 7. Assessment of detection accuracy

Interference conditions	Weathering erosion	Wall cracks	Component damage	Paint aging
Normal lighting	98.5%	96.7%	94.6%	97.4%
Low lighting	96.8%	94.2%	93.3%	95.1%
Strong light overexposure	95.2%	93.5%	92.7%	94.3%
Rainy day blurring	93.6%	91.8%	91.4%	92.5%
Haze obstruction	92.1%	90.3%	90.8%	90.7%
Shadow obstruction	94.8%	93.1%	90.1%	93.6%

Low light, overexposure to strong light, and shading were collected naturally on site, corresponding to the actual lighting conditions of strong light in the early morning, evening, noon, and backlit areas of buildings. Due to the lack of corresponding weather conditions during the collection period, rain blur and haze obstruction were injected into the test set using a synthetic method. Rain blur was generated by applying a 5 × 5 Gaussian blur kernel (σ=1.5) to the original image to simulate the detail degradation caused by raindrops; Haze obstruction is synthesized using an atmospheric scattering model, with a fog density factor of 0.6, corresponding to moderate concentration haze. The synthesized samples retain the original annotations and are used together with the field samples for robustness testing. In Table 7, as the interference conditions increased,

the accuracy of various types of DDM decreased in turn. Under low light conditions, the accuracy rates of weathering erosion, wall cracks, component damage and paint aging were 96.8%, 94.2%, 93.3% and 95.1%. Under strong light overexposure conditions, they were 95.2%, 93.5%, 92.7% and 94.3%. Under rainy day blur conditions, they were 93.6%, 91.8%, 91.4% and 92.5%. Under haze cover conditions, they were 92.1%, 90.3%, 90.8% and 90.7%. Under shadow obstruction conditions, they were 94.8%, 93.1%, 90.1% and 93.6%. Under different interference conditions, the defect detection accuracy of the model was higher than 90%. In summary, proposed model could accurately detect various defects in ancient buildings.

4. Discussion

To tackle the problems of poor precision in extracting fine-long defect features, severe interference from complex backgrounds, insufficient MSF fusion, and difficulties in deploying high-precision models in the defect detection of ancient buildings, this study designed an IYOLOv8 lightweight ancient building DDM via CBAM and CA mechanisms. This model aimed to strengthen defect feature extraction abilities and decrease model parameters, meeting the real-time deployment requirements of terminal devices. The ablation test results showed that after introducing CBAM and CA mechanisms, the model's mAP@0.5 increased from 80.4% to 94.3%, and finally, CBAM-CA-IYOLOv8 reached 95.8%; the parameter size decreased from 4.0MB to 2.7MB, and the running speed increased from 108FPS to 118FPS. This result was similar to that of Sakib Bin Islam M et al., indicating that introducing attention mechanisms in the YOLOv8 algorithm could improve the detection accuracy of YOLOv8, and after light weighting, its running speed was accelerated and the computational time was reduced [28]. Moreover, by introducing the VanillaNet module and the simSPPF module, the model's parameter size reduced, thereby reducing computational redundancy and improving inference efficiency. In the IYOLOv8, the VanillaNet module used 1×1 Conv kernels instead of standard 3×3 Conv kernels, reducing the parameter size of the Conv kernels and significantly reducing the parameter size and computational cost of the feature extraction layer. Meanwhile, the VanillaNet module adopted a deep training strategy, splitting each 1×1 Conv into two consecutive Conv and inserting nonlinear AFs during the training stage, gradually simplifying the AF to an identity mapping. In the inference stage, the two Conv layers were fused into a single layer Conv, further eliminating redundant parameters. The simSPPF module in the improved algorithm replaced the original SPPF arrangement with a serially stacked 5×5 max Pool and performed channel reduction and expansion through 1×1 Conv, replacing the SiLU AF with ReLU to eliminate the computational overhead of exponential operations, while maintaining the multi-scale receptive field fusion capability and reducing the module complexity. These improvements jointly eliminated the computational redundancy brought by the complex residual connections and multi-branch arrangement in the C2f module, reducing the model parameter size from 4.0MB to 2.7MB and significantly reducing the computational complexity.

Then the outcome of the introduced CBAM-CA-IYOLOv8 DDM was compared with the DSConv-YOLOv8, MEB-YOLOv11 and UNet-YOLOv8 DDMs. The mAP@0.5 of CBAM-CA-IYOLOv8 was 96.2%, which was higher than MEB-YOLOv11's 86.2%, DSConv-YOLOv8's 84.6% and UNet-YOLOv8's 83.1%; The DE of CBAM-CA-IYOLOv8 was 1.2%, lower than DSConv-YOLOv8's 5.7%, UNet-YOLOv8's 4.8% and MEB-YOLOv11's 3.2%. The above results were roughly the same as those of Xie Y's team. However, in the research of Xie Y's team, when performing defect detection, its defect detection accuracy in the NEU-DET dataset and DeepPCB dataset reached 83.9% and 98.9% respectively, and the detection accuracy in the NEU-DET dataset was lower. Although the Xie Y team adopted depth separation Conv and improved feature pyramid to improve defect detection accuracy and reduce detection speed, it paid less attention to slender defect features, which resulted in lower detection accuracy in the NEU-DET dataset [29]. The CBAM-CA-IYOLOv8 DDM not only took into account the characteristics of slender defects, but also embedded position information into CAM through the CA, and captured long-distance dependencies and retained precise position information through 1D feature encoding in the HVDs. This method employed CBAM to adaptively enhance the defect feature response and suppress complex background interference. Therefore, the proposed model had higher detection accuracy for various types of defects in complex backgrounds.

In addition, actual test results verified that the model's detection accuracy of weathering erosion, wall cracks, component damage and paint aging under normal lighting conditions were 98.5%, 96.7%, 94.6% and 97.4%. The accuracy of various defects under six interference conditions was higher than 90%. This result was roughly the same as that of the Sani A R team, both verifying that YOLOv8 improved its defect detection capabilities after optimization [30]. The suggested CBAM-CA-IYOLOv8 ancient building DDM accurately detected different defects, thereby enabling timely repair of ancient building defects. In summary, this new method achieved model lightweight while ensuring high accuracy, and solved the problems of low extraction accuracy of slender defects, serious complex background interference, and difficult model deployment in ancient building defect detection.

5. Conclusion

Targeting the issues of low extraction fidelity of slender defect features, severe complex background interference, and difficulty in model deployment in ancient building defect detection, this study proposes a YOLOv8 lightweight model CBAM-CA-IYOLOv8 via improvements in CBAM and CA attention mechanisms. The CBAM module enhances the defect feature channel response in the backbone and neck networks, while the CA module captures long-range dependencies of slender defects, improving the localization accuracy of cracks. Test results show that the introduced attention mechanism enhances the defect feature channel response and suppressses complex background interference, improving defect detection accuracy. And after using the VanillaNet module to replace the C2f and the simSPPF to replace the SPPF, the count of parameters in the approach is greatly reduced, the running rate is notably improved. The mAP@0.5 of the proposed model reaches 95.8%, the number of parameters is reduced to 2.7 MB, and the running speed reaches 118 FPS. It matches the real-time inference requirements of terminal equipment. The CBAM-CA-IYOLOv8 DDM is compared with DSConv-YOLOv8, MEB-YOLOv11 and UNet-YOLOv8. The model is superior to the above comparison models for detection accuracy, F1 score, FDR and DE. IT has the best overall performance. In addition, actual test outcomes verifies that the model has a strong detection precision for defects including weathering and erosion, wall cracks, component defects, and paint aging. This model maintains high robustness under a variety of complex interference conditions. The proposed CBAM-CA-IYOLOv8 model tackles the challenges of low slender defect extraction accuracy, severe complex background interference and difficult model deployment in ancient building defect detection, and can provide technical support for the protection of ancient buildings. However, this study still has shortcomings like insufficient detection capabilities for hidden defects. Future research will integrate multi-source sensing data including infrared thermal imaging and three-axisal point clouds, for further improving the model's detection capabilities for hidden defects and intricate working conditions.

References

[1] Ghida DB. Organic stabilization in earthen plaster: Eco-compatible architecture and ancient techniques in Tata Somba homes. Frontiers Architectural Research. 2024 Feb;13(3):625-638. https://doi.org/10.1016/j.foar.2024.02.004

[2] Halim MHA, Saidin M. Architecture of the ancient port management SB2G site at Sungai Batu Archaeological Complex (SBAC), Bujang Valley, Kedah, Malaysia. J. Maritime Archaeol. 2025 Sep;20(4):769-787. https://doi.org/10.1007/s11457-025-09468-0

[3] Miran FD, Husein HA. Evaluating deterioration causes and defect patterns in heritage buildings: A comprehensive analytical approach. Int. J. Archit. Heritage. 2025 Sep;19(9):1627-1646. https://doi.org/10.1080/15583058.2024.2367690

[4] Joyce HH. Antiquity in Gotham: The ancient architecture of New York City. J. Soc. Architectural Historians. 2023 Jun;82(2):214-216. https://doi.org/10.1525/jsah.2023.82.2.214

[5] Gil A, Arayici Y, Kumar B, Laing R. Machine and deep learning implementations for heritage building information modelling: A critical review of theoretical and applied research. ACM J. Computing Cultural Heritage. 2024 Apr;17(3):1-22. https://doi.org/10.1145/3649442

[6] Meklati S, Boussora K, Abdi MEH, Berrani SA. Surface damage identification for heritage site protection: A mobile crowd-sensing solution based on deep learning. ACM J. Computing Cultural Heritage. 2023 Mar;16(2):1-24. https://doi.org/10.1145/3569093

[7] Selgas-Cors M. Sociotechnical Transformation: A systematic review on the impact of Artificial Intelligence on society and organizations. FSI. 2025 Aug:1-16. https://doi.org/10.47852/bonviewFSI52026076
[8] Bhuyan P, Singh PK, Das SK. Res4Net-CBAM: A deep CNN with convolution block attention module for tea leaf disease diagnosis. Multimedia Tools Appl. 2024 Nov;83(16):48925-48947. https://doi.org/10.1007/s11042-023-17472-6
[9] Parlatini V, Itahashi T, Lee Y, Liu S, Nguyen TT, Aoki YY. White matter alterations in Attention-Deficit/Hyperactivity Disorder (ADHD): A systematic review of 129 diffusion imaging studies with meta-analysis. Mol. Psychiatry. 2023 Jul;28(10):4098-4123. https://doi.org/10.1038/s41380-023-02173-1
[10] Kim C, Park G, Jang H, Kim EJ. Automated classification of thermal defects in the building envelope using thermal and visible images. Quant. Infrared Thermogr. J. 2023 May;20(3):106-122. https://doi.org/10.1080/17686733.2022.2033531
[11] Chen J, Lu W, Lou J. Automatic concrete defect detection and reconstruction by aligning aerial images onto semantic-rich building information model. Comput.-Aided Civ. Infrastruct. Eng. 2023 Oct;38(8):1079-1098. https://doi.org/10.1111/mice.12928
[12] Huo Z, Wu X, Cheng T. Crack recognition and defect detection of assembly Building constructions for intelligent construction. J. Meas. Eng. 2024 Jun;12(3):485-501. https://doi.org/10.21595/jme.2024.23977
[13] Li W, Zhou H, Li J. Building surface defect and damage detection method based on YOLOv8-act. Int. Core J. Eng. 2026 Feb;12(2):93-100. https://doi.org/10.6919/ICJE.202602_12(2).0010
[14] Shin H, Lee S. An efficient explainable convolutional network with visualization of feature maps for enhanced understanding of Building Facade Defects. J. Comput. Civ. Eng. 2024 Jun;38(5):1-12. https://doi.org/10.1061/JCCEE5.CPENG-5857
[15] Chen C, Zhou Q, Li S, Luo D, Tan G. Fabric defect detection algorithm based on improved YOLOv8. Text. Res. J. 2025 Feb;95(3-4):235-251. https://doi.org/10.1177/00405175241261092
[16] Liang Y, Xu Z. Intelligent inspection of appearance quality for precast concrete components based on improved YOLO model and multi-source data. Eng. Constr. Archit. Manag. 2025 Mar;32(3):1691-1714. https://doi.org/10.1108/ECAM-07-2023-0705
[17] Karimi N, Mishra M, Lourenço PB. Automated surface crack detection in historical constructions with various materials using deep learning-based YOLO network. Int. J. Archit. Heritage. 2025 May;19(5):581-597. https://doi.org/10.1080/15583058.2024.2376177
[18] Tang P, Chen P, Ting TT, Shi Y, Zhao J, Liang Z. ACT-YOLO: An efficient multi-module fusion object detection algorithm for steel surface defect detection. Inf. Technol. Control. 2025 Dec;54(4):1227-1247. https://doi.org/10.5755/j01.itc.54.4.42676
[19] Jiang F, Ye W, Lu P, Zhang S, Wu Z, Zhu H. CNCR-YOLO: A comprehensive optimization strategy for small-target defect detection in injection-molded parts. IEEE Sensors J. 2024 Nov 1;24(21):35954-35964. https://doi.org/10.1109/JSEN.2024.3454311
[20] Ning Z, Wang H, Li S, Xu Z. YOLOv7-RDD: A lightweight efficient pavement distress detection model. IEEE Trans. Intell. Transp. Syst. 2024 Jul;25(7):6994-7003. https://doi.org/10.1109/TITS.2023.3347034
[21] Ahmed MR, Ashrafi AF, Ahmed RU, Shatabda S, Islam AM, Islam S. DoubleU-NetPlus: A novel attention and context-guided dual U-Net with multi-scale residual feature fusion network for semantic segmentation of medical images. Neural Computing Appl. 2023 Mar;35(19):14379-14401. https://doi.org/10.1007/s00521-023-08493-1
[22] Omega Boro L, Nandi G. CBAM-RIUnet: Breast tumor segmentation with enhanced breast ultrasound and test-time augmentation. Ultrasonic Imaging. 2025 Jan;47(1):24-36. https://doi.org/10.1177/01617346241276411
[23] Labiadh I, Boubchir L, Seddik H. Optimization of 2D and 3D facial recognition through the fusion of CBAM AlexNet and ResNeXt models. Visual Computer. 2025 Dec;41(8):5235-5250. https://doi.org/10.1007/s00371-024-03718-3
[24] Armijantoro GR, Nugroho B, Via YV. Optimization of tea leaf disease detection based on YOLOv8 using CBAM and BFP. bit-Tech. 2026 Apr;8(3):3225-3235. https://doi.org/10.32877/bt.v8i3.3394
[25] Manihira NR, Sethy PK. Transfer learning-based lightweight MobileNetV2 with CBAM attention for rice leaf nutrient deficiency detection. Neural Computing Appl. 2025 Sep;37(32):27179-27198. https://doi.org/10.1007/s00521-025-11665-w
[26] Afifah V, Erniwati. YOLOv8 for object detection: A comprehensive review of advances, techniques, and applications. IJACI: Int. J. Adv. Computing Inform. 2026 Aug;2(1):53-61. https://doi.org/10.71129/ijaci.v2i1.pp53-61
[27] Nemati N, Samet R. HR-YOLOv8: An innovative model to detect mitosis and identify cancer regions in histopathological images. Neural Computing Appl. 2025 Sep;37(29):24441-24460. https://doi.org/10.1007/s00521-025-11594-8

[28] Sakib Bin Islam M, Chowdhury MEH, Hasan-Zia M, Kashem SBA, Majid ME, Ansaruddin Kunju AK. VisioDECT: A novel approach to drone detection using CBAM-integrated YOLO and GELAN-E models. Neural Comput. Appl. 2025 Sep;37(24):20181-20204. https://doi.org/10.1007/s00521-025-11448-3
[29] Xie Y, Hu W, Xie S, He L. Surface defect detection algorithm based on feature-enhanced YOLO. Cogn. Comput. 2023 Oct;15(2):565-579. https://doi.org/10.1007/s12559-022-10061-z
[30] Sani AR, Zolfagharian A, Kouzani AZ. Automated defects detection in extrusion 3D printing using YOLO models. J. Intell. Manuf. 2026 Jan;37(1):351-371. https://doi.org/10.1007/s10845-024-02543-8