

Intelligent Defect Detection of Design Materials Based on Edge AI

Jiayu Du^{ID}, Mengyu Liu^{*}^{ID}*Department of Art and Design, Beijing City University, Shunyi District, Beijing, China****Corresponding author:** Mengyu Liu. Email: daimou971231@naver.com

Abstract

Aiming at the problems such as the imbalance between accuracy and efficiency, insufficient adaptability of multi-scale defects and limited inference resources in the deployment of design material defect detection at the edge end, this paper proposes a lightweight detection method based on edge AI. In the training stage, multi-branch topology is used to enhance feature expression, and in the inference stage, multi-branch convolution is merged into a single standard convolution by algebraic fusion. Under the condition that the number of parameters is only 2.2M and the amount of computation is 2.6 GFLOPs, the detection accuracy is significantly improved. The bidirectional feature pyramid and lightweight channel-spatial joint attention fusion module are designed to effectively cope with the challenges of defect scale diversity, low contrast and inter-class similarity, and the recall rate of small-scale defect reaches 82.4%. An early departure inference mechanism based on input complexity and a dynamic computing path guided by confidence are constructed, and combined with quantization-aware training and operator fusion optimization, 18ms end-to-end inference latency and 55.6 FPS frame rate are achieved on Jetson Xavier NX, with power consumption of only 2.4 W. Experiments on NEU-DET, DAGM2007 and self-built industrial datasets show that the proposed method mAP@0.5 reaches 75.2%, which outperforms the mainstream lightweight detection network with more than three times the number of parameters, and shows superior performance in a variety of defect types and scales. It provides a feasible technical path for real-time, safe and low-cost detection in intelligent manufacturing scenarios.

Keywords

Edge AI; Defect detection; Lightweight network; Multi-scale attention fusion; Adaptive inference

Received: 16 Jul 2026; Revised: 08 Aug 2026; Accepted: 31 Aug 2026; Published: 22 Sep 2026
Copyright: ©2026 The Author(s). Published by [Digital Intelligence Press Limited](#). This is an
open access article under the [CC BY 4.0](#) license.

1. INTRODUCTION

1.1 Research background and significance

Defect detection in engineered materials is a critical component of intelligent manufacturing and quality control; its accuracy and efficiency directly impact product yield rates and production costs [1]. Traditional manual visual inspection is limited by subjective experience and operator fatigue, making it difficult to meet requirements for high production throughput and consistency. While machine vision-based methods improve efficiency, traditional image processing algorithms often lack sufficient generalization capabilities when dealing with diverse defect types, significant scale variations, and drastic contrast fluctuations [2],[3]. Deep learning methods can automatically learn discriminative defect features and outperform traditional approaches in terms of accuracy and robustness; however, high-performance models often involve a large number of parameters and high computational

complexity, making direct deployment on industrial edge devices challenging [4],[5]. Edge AI deploys inference on devices close to the data source, thereby reducing latency and bandwidth usage while protecting data privacy; yet, given the limited computing power, memory, and energy budgets of edge devices, maintaining detection accuracy under resource constraints remains a core challenge [6],[7].

Existing research indicates that hybrid frameworks combining deep learning and gradient boosting can enhance predictive performance for complex industrial data [8],[9]. In supply chain and logistics scenarios, convolutional neural networks (CNNs) and bidirectional long short-term memory (Bi-LSTM) networks have been employed for network optimization and demand forecasting, while hybrid models combining autoregressive moving average (ARMA) and LSTM networks have been used to predict logistics demand in international trade [10],[11],[12],[13]. Furthermore, methods integrating CNNs, graph attention networks (GATs), and gradient boosting decision trees (GBDTs) alongside graph-based risk inference and ensemble forecasting—demonstrate the potential of graph-structured modeling, attention mechanisms, and ensemble learning in complex industrial settings [14],[15]. These cross-domain studies offer methodological insights for multi-scale feature fusion and dynamic inference in edge-based defect detection [16].

In industrial time-series and energy applications, spatiotemporal prediction models combining CNNs and LSTM networks have been utilized for photovoltaic power generation forecasting, while hybrid models integrating CNNs, LSTM networks, and extreme gradient boosting (XGBoost) have been applied to predict seismic attributes of shale oil and global oil price trends [17],[18],[19],[20]. These studies demonstrate that multi-model collaboration and lightweight design can enhance prediction stability under conditions characterized by high dimensionality, nonlinearity, and significant noise. Against this backdrop, and addressing the requirements for edge deployment in material defect detection, this paper proposes a lightweight edge AI detection method. By employing multi-branch topology training combined with algebraic fusion inference, a bidirectional feature pyramid alongside a lightweight joint channel-spatial attention mechanism, and an adaptive inference mechanism based on input complexity and confidence, the method balances detection accuracy with inference efficiency under resource-constrained conditions, offering a viable pathway for real-time, secure, and cost-effective inspection in intelligent manufacturing.

1.2 Main contributions

To solve the above problems, this paper proposes a lightweight defect detection network architecture for edge deployment. The architecture takes reparameterization and a lightweight backbone as the core, adopts multi-branch topology structure to enhance feature expression ability in the training stage, and merges multi-branch convolution into a single standard convolution through algebraic fusion in the inference stage, thus significantly reducing the amount of computation without losing accuracy. The backbone network adopts the phased channel configuration and downsampling strategy, retains high-resolution spatial details in the shallow stage, extracts strong semantic features in the deep stage, and further compresses the model scale through channel pruning [21]. Under the condition that the number of parameters and calculations are much lower than the existing lightweight detection network, the detection accuracy is significantly improved.

This paper designs a multi-scale attention fusion mechanism oriented to material defect features. The mechanism adopts bidirectional feature pyramid structure. The top-down path upsamples deep semantic features and fuses them with shallow features, and the bottom-up path downsamples shallow detail features and fuses them with deep features, forming bidirectional aggregation of multi-level

features. On this basis, the lightweight channel-spatial joint attention module is introduced to recalibrate the response intensity of feature channels through channel attention and enhance the spatial response of defect areas through spatial attention. The joint modulation of the two can achieve accurate enhancement of defect related features and effective suppression of background interference [22]. The computational cost of this mechanism is very small, and the number of parameters is increased by less than 5%, but it can bring significant accuracy gain in multi-scale defect detection.

This paper constructs a set of collaborative optimization and adaptive inference strategies for edge AI detection system. In terms of collaborative optimization, the efficient deployment of the model on the target edge hardware is realized from three levels: comparison and selection of post-training quantization and quantization perception training, operator fusion and memory layout optimization, model transformation and inference engine adaptation. In terms of adaptive inference, an early exit inference mechanism based on input complexity and a multi-branch dynamic computing path guided by confidence are designed, so that the model can dynamically allocate computing resources according to the significant degree of defects in the input image and significantly reduce the average inference delay on the premise of ensuring the detection accuracy. In actual deployment, this policy can be flexibly configured according to the real-time requirements of application scenarios to achieve the on-demand trade-off between accuracy and delay.

This paper verifies the superiority and engineering feasibility of the method on several material defect data sets. The experiment covers public steel defect data set, texture defect data set and self-built industrial data set, involving three types of design materials: metal stamping parts, injection molded parts and welding parts. The defect types include pits, cracks, pores, inclusions and scratches, etc. The evaluation indicators cover detection accuracy, inference delay, number of model parameters, computation and power consumption. End-to-end deployment tests are carried out on several edge devices such as Jetson Xavier NX, RK3588 and Raspberry Pi. Experimental results show that the proposed method achieves significant Pareto improvement in the trade-off between accuracy and efficiency, shows superior detection performance in a variety of defect types and scales, and has good long-term operation stability in real industrial scenarios.

1.3 Organizational structure

This paper is divided into five sections, each of which is arranged as follows. Section 1 is the introduction, which expounds the research background and significance, reviews the research status at home and abroad, analyzes the limitations of the existing work, and outlines the main contributions and organizational structure of this paper. Section 2 is the basis of relevant theories and technologies, analyzing the visual characteristics of material defect detection, discussing the edge AI computing paradigm and deployment constraints, introducing the design principle of lightweight neural network, as well as attention mechanism and multi-scale feature fusion technology. Section 3 is the defect detection model and method based on edge AI, which elaborates on problem formalization and overall architecture design, lightweight backbone feature extraction network, multi-scale defect attention fusion module, adaptive inference and dynamic computing strategy, as well as edge deployment optimization and quantization scheme. Section 4 is about experimental verification and result analysis, introducing experimental setup, carrying out ablation experiment and comparison experiment, carrying out actual deployment test of edge end, and discussing the results and analyzing the limitations. Section 5 is the conclusion and prospect, summarizing the work of the paper, summarizing the core innovation, and looking forward to the future research direction. Each section progresses layer by layer, from theoretical basis to method design to experimental verification, forming complete research closed loops.

2. RELEVANT THEORETICAL AND TECHNICAL BASIS

2.1 Visual feature analysis of material defect detection

There are many types of defects generated in the manufacturing and processing of design materials, and their visual representation characteristics directly determine the difficulty of detection tasks and the focus of algorithm design. The common defects of metal stamping parts include pits, cracks, burrs and deformation, among which pits are represented by sudden brightness and shadow distribution in local areas, cracks are represented by thin and irregular dark lines, and burrs are represented by abnormal prominent contour in edge areas. Typical defects of injection molded parts include pores, shrinkage marks and flying edges. Pores appear as round or oval bright spots with low contrast in the image, shrinkage marks appear as gradual shadows caused by surface depressions, and flying edges appear as thin flaky protrusions near the dividing surface. The defects of welding parts mainly include pores, slag inclusion, non-fusion and cracks, among which pores and slag inclusion show as local density abnormal regions in X-ray or visual images, non-fusion shows as linear dark bands at the edge of the weld, and cracks show as fine dark lines along the grain boundary. The common visual features of the above defects are abnormal local texture, abrupt gray distribution and irregular geometric morphology, but their specific manifestations are significantly affected by the reflection characteristics of the material surface, lighting conditions and imaging angles, resulting in great visual differences between the same defects in different samples.

The scale diversity of defect targets is one of the core challenges in material defect detection. In the same image, the diameter of tiny pores may be only a few pixels, while the length of large deformation or scratch may occupy more than a third of the width of the image, and the scale span can be up to two orders of magnitude. This scale difference makes it difficult to take into account both the detailed information of small targets and the semantic information of large targets in feature extraction at a single scale. Contrast differences are another significant challenge. Some defects (such as shallow scratches on metal surfaces) have very little gray difference from the background and are easily drowned out by noise under low contrast conditions [23]. However, some defects (such as obvious shrinkage marks of injection molded parts) have high contrast and relatively low detection difficulty. The problem of inter-class similarity also cannot be ignored. Different types of defects may overlap in visual features. For example, the inclusion and patch on the metal surface are relatively close in gray distribution and texture features, and the pores and slag inclusions of welding parts are also similar in shape and size, which requires the detection model to have fine feature discrimination ability. In addition, the class imbalance between defects and background is common. The normal region usually occupies the majority of the image area, while the defect region only accounts for a very small proportion, which leads to the bias of the model toward the background class in the training process, which needs to be alleviated by loss function design and data sampling strategy.

2.2 Edge AI computing paradigm and deployment constraints

The edge AI computing paradigm migrates the inference process of deep learning models from cloud servers to edge devices close to data sources, thus gaining significant advantages in real-time performance, bandwidth efficiency and data privacy. However, resource constraints of edge devices constitute the core bottleneck of model deployment. In terms of computing power constraints, the computing power of typical edge devices is usually between 1 and 20 TOPS, which is far lower than hundreds of TOPS of server-level GPUs. Moreover, the computing power of edge devices is often

limited by power consumption budget and heat dissipation conditions, making it difficult to continuously operate at full load. In terms of memory constraints, the memory capacity of edge devices is usually 1 to 8 GB, among which the memory available for deep learning models is more limited. Model weights, intermediate feature maps and runtime overhead need to share the same memory space. In terms of power consumption constraints, edge devices in industrial sites usually adopt passive heat dissipation or fanless design, and the continuous power consumption should be controlled within 5 to 15 W, which puts forward strict requirements on the computing density and memory access efficiency of the model. The above three constraints are coupled to each other and together determine the boundary of the feasible region for the design of the edge-end model.

The quantitative index system of model inference delay and energy efficiency ratio provides a measurable objective function for edge end model optimization. Inference delay is usually measured by single frame processing time, including model forward propagation computation time and memory access time, which is often dominant on modern devices. The energy efficiency ratio is measured by the number of inferences that can be completed per watt of power consumption, which directly reflects the actual throughput capacity of the model under power consumption constraints. In addition to latency and energy efficiency ratio, model volume, peak memory occupancy, and thermal stability are also indicators that need to be considered comprehensively in edge deployment. Model volume determines storage overhead and load time, peak memory footprint affects the feasibility of parallel deployment of multiple models, and thermal stability is related to the performance consistency of devices in long-term operation. For example, reducing model volume usually comes at the cost of accuracy, and reducing inference delay may increase power consumption. Therefore, edge end model design needs to seek Pareto optimal solution in a multi-objective optimization framework.

2.3 Design principle of lightweight neural network

Deep separable convolution and group convolution are two basic techniques for lightweight neural network design. Depthwise separable convolution decomposes standard convolution into two independent steps: depthwise convolution and pointwise convolution. Channel by channel convolution performs spatial convolution independently for each input channel, while point by point convolution realizes information fusion between channels through one-by-one convolution. This decomposition method greatly reduces the number of parameters and computation of standard convolution, but the cost is the delay of information interaction between channels and the reduction of memory access density. Group convolution divides input channels into multiple groups, and each group performs convolution operation independently. Channels within the group can interact while channels between the groups remain isolated. Group convolution is between standard convolution and depthwise separable convolution in terms of the number of parameters and calculation. By adjusting the number of groups, a flexible trade-off between accuracy and efficiency can be achieved. Depth separable convolution can be regarded as a special case of group convolution when the number of groups is equal to the number of channels. The two have internal consistency in mathematics, but there are differences in hardware implementation and memory access mode. In actual edge deployment, the theoretical computational advantage of deeply separable convolution cannot be completely transformed into the actual acceleration ratio due to low memory access efficiency. Therefore, targeted optimization is needed at the level of operator fusion and memory layout.

Neural architecture search and manual design of lightweight networks represent two different design paradigms. Neural architecture search uses automated search algorithms to find the optimal network structure that meets specific constraints in the huge architecture space. Its advantage is that it

can find architecture patterns that are difficult for human designers to intuitively think of. However, the search process usually requires a lot of computing resources and time costs, and the architecture obtained by search has limited mobility on different hardware platforms. Manual design of lightweight network relies on the designer's experience and domain knowledge and achieves the balance between accuracy and efficiency through carefully designed modules and connection modes. Its advantages lie in short design cycle, strong interpretability and good hardware adaptation, but the exploration scope of design space is limited by the designer's cognitive boundary. In recent years, the two have shown a trend of fusion, that is, the design principles of basic modules are discovered through neural architecture search, and then hardware friendly deployment is realized through manual engineering optimization. For edge defect detection tasks, manual design of lightweight network has more engineering practical value due to its deployment controllability and hardware adaptation.

Knowledge distillation and pruning quantization are the three main technical routes of model compression. Knowledge distillation transfers the dark knowledge learned by the teacher model to the student model by making the lightweight student model imitate the output distribution or intermediate feature representation of the large-capacity teacher model, so as to improve the accuracy of the lightweight model without increasing the inference cost. Pruning technology reduces the number of parameters and computation on the premise of maintaining accuracy by removing redundant channels, convolution kernels or connections in the model. Structured pruning is more suitable for edge hardware acceleration because it can generate regular sparse structures. Quantization technology maps floating point weight and activation value into low bit fixed-point representation, which reduces model volume and uses fixed-point operation unit to improve inference speed. However, the accuracy loss caused by low bit quantization needs to be compensated by quantization perception training. The three techniques are not mutually exclusive but can be used in combination: knowledge distillation to improve the upper limit of accuracy of lightweight models, pruning to remove redundant structures, and quantization to compress numerical accuracy. The cooperation of the three can achieve the optimal balance between accuracy and efficiency at the edge.

2.4 Attention mechanism and multi-scale feature fusion

Channel attention and spatial attention are two basic forms of attention mechanism. Channel attention adaptively adjusts the characteristic response intensity of each channel by modeling the dependence between channels. Its core idea is to let the network learn to pay attention to the channels with large amounts of information and suppress redundant channels. Typical channel attention implementation methods include global average pooling and fully connected layer, one-dimensional convolution replacing the fully connected layer, and channel recalibration based on covariance, etc. Spatial attention can enhance the feature response of the defect region and suppress the interference of the background region by modeling the dependence between spatial locations. Its implementation methods include pooling operation along the channel dimension, spatial convolution and self-attention mechanism. Channel attention and spatial attention can be combined in series or parallel to form a joint attention module, so as to carry out feature recalibration in the two dimensions of channel and space simultaneously. The computational cost of attention mechanism mainly comes from pooling operation and fully connected layer, and its computational complexity needs to be reduced by means of channel compression, packet convolution and depth separable convolution in lightweight design.

Feature pyramid and multi-scale fusion strategy are the core technologies to deal with defect scale diversity. Feature pyramid integrates deep semantic features with shallow high-resolution features through top-down path, so that the feature map at each level has both semantic information and spatial

details. In the bottom-up path, shallow features are further downsampled and fused with deep features to form a bidirectional feature pyramid structure. Multi-scale fusion strategies include element-by-element addition, channel splicing and attention-weighted fusion, among which element-by-element addition requires the least calculation but has limited information fusion ability. Channel splicing retains the independence of features at each level but increases the channel dimension, and attention-weighted fusion adaptively adjusts the contribution of features at each level through learnable weights. In edge deployment, the design of multi-scale fusion module needs to strike a balance between fusion capability and computing cost, and lightweight fusion operators and limited number of fusion levels are usually adopted.

The lightweight transformation requirement of attention mechanism deployed at the edge originates from the contradiction between the high computing cost of its original design and the resource constraints of edge devices. The number of fully connected layer parameters in standard channel attention is proportional to the square of the number of channels, and the computational cost is significant in deep networks with large number of channels. The main ideas of lightweight transformation include replacing the full connection layer with one-dimensional convolution to reduce the number of parameters, introducing channel compression ratio to reduce the intermediate feature dimension, using grouping operation to decompose channel attention into multiple independent sub-attention modules, and using hard attention or sparse attention to reduce the calculation path. The lightweight transformation of spatial attention focuses on reducing the size of convolution kernel, reducing spatial resolution and replacing standard convolution with depthwise separable convolution. In addition, the location and number of attention modules also need to be carefully designed. Too many attention modules will lead to the accumulation of inference delay, while too few will fail to give full play to the effect of feature recalibration. The design goal of the edge end attention module is to achieve significant detection accuracy improvement under the constraint that the number of parameters increases by no more than 5%, and the inference delay increases by no more than 10%.

3. PROBLEM FORMALIZATION AND OVERALL ARCHITECTURE DESIGN

The edge end defect detection task can be formally defined as follows: given the input image $x \in \mathbb{R}^{H \times W \times C}$, where H , W and C are the height, width and number of channels of the image respectively, the model needs to output the category label and spatial positioning results of defects. Let the set of defect categories be $\mathcal{C} = \{c_1, c_2, \dots, c_K\}$, K as the total number of categories, the test results can be represented as a set $\mathcal{D} = \{(b_i, s_i, l_i)\}_{i=1}^N$, where N instances of detected flaws, $b_i = (x_i, y_i, w_i, h_i)$ for the bounding box of the i -th a defect coordinates, $s_i \in [0, 1]$ is the confidence score, and $l_i \in \mathcal{C}$ is the category label.

The core constraint of edge deployment is that the model must complete inference under limited computing resources. If the available computing power of the edge device is F_{edge} (unit: FLOPS), the memory capacity is M_{edge} , and the power consumption budget is P_{edge} , the deployment feasibility of the detection model $\mathcal{M}$ can be formally described as follows:

$$\mathcal{M}^* = \arg \min_{\mathcal{M}} \mathcal{L}_{\text{det}}(\mathcal{M}) + \lambda_1 \mathcal{T}_{\text{inf}}(\mathcal{M}) + \lambda_2 \mathcal{S}_{\text{model}}(\mathcal{M}) \quad (1)$$

Where $\mathcal{L}_{\text{det}}$ is the detection loss function, $\mathcal{T}_{\text{inf}}$ is the single frame inference delay, $\mathcal{S}_{\text{model}}$ is the model storage volume, λ_1 , λ_2 are the trade-off coefficients, which are used to adjust the relative

importance of accuracy, delay and model volume. The optimization objective must meet the constraints $\mathcal{T}_{\text{inf}} \leq T_{\text{max}}$, $\mathcal{S}_{\text{model}} \leq M_{\text{edge}}$, $\mathcal{P}_{\text{inf}} \leq P_{\text{edge}}$, where T_{max} is the maximum delay allowed by the application, and $\mathcal{P}_{\text{inf}}$ is the inference power consumption.

The overall network architecture adopts the three-stage design paradigm of "lightweight backbone - multi-scale fusion - dynamic inference". The data stream starts with the input image, and the multi-level feature map is extracted stage by stage through the lightweight backbone network. Then the multi-scale defect attention fusion module performs cross-layer aggregation of shallow high-resolution features and deep strong semantic features, and finally the classification and regression results are output by the detection head. In the inference stage, the adaptive inference module dynamically decides whether to enable the early exit branch according to the input complexity to realize the on-demand allocation of computing resources. The joint optimization strategy of design objectives runs through the entire architecture design process. In terms of accuracy, the loss of feature expression ability of lightweight backbone is compensated by multi-scale attention fusion. In terms of delay, the single frame inference time is reduced by the fusion of reparameterization module and operator. In terms of model volume, the storage occupation is compressed by quantization-aware training and channel pruning. The trade-off between the three is not static configuration, but dynamic adjustment at runtime through adaptive inference strategy, so as to achieve optimal resource-precision matching under input scenarios of different complexity.

3.1 Lightweight backbone feature extraction network

The design of lightweight convolution module based on heavy parameterization is the core innovation of backbone network. In the training stage, the module adopts multi-branch topology to enhance the feature expression ability. In the inference stage, the multi-branch convolution kernel is fused into a single convolution kernel by algebraic transformation, thus significantly reducing the amount of computation without loss of accuracy. Suppose that a convolution layer contains B parallel branches in the training stage, the convolution kernel of the j -th branch is $W_j \in \mathbb{R}^{k_j \times k_j \times C_{\text{in}} \times C_{\text{out}}}$, and the bias is b_j , then the output of this layer can be expressed as:

$$y = \sum_{j=1}^B \text{BN}_j \left(\text{Conv}(x, W_j, b_j) \right) \quad (2)$$

Where BN_j is the batch normalization operation of j -th branch, and Conv is the convolution operation. In the inference stage, by absorbing the batch normalization parameters into the convolution kernel, and adding the multi-branch convolution kernel element by element after alignment according to the spatial dimension, it can be equivalently fused into a single convolution:

$$W_{\text{fused}} = \sum_{j=1}^B \frac{\gamma_j}{\sigma_j} \cdot \text{pad}(W_j, k_{\text{max}}) \quad (3)$$

Where γ_j and σ_j are the scaling and offset parameters of batch normalization, $\text{pad}(\cdot)$ is the operation of filling the convolution kernel to the maximum kernel size k_{max} , and W_{fused} is the equivalent convolution kernel after fusion. This reparameterization strategy enables the rich gradient information in the training stage to be retained, while the standard convolution in the inference stage only needs to be performed once, which significantly improves computational efficiency.

The channel configuration and downsampling strategy of multi-stage feature extraction follow the

design principle of "shallow wide and deep narrow". Suppose that the backbone network is divided into S stages, the number of output channels in the s stage is C_s , and the spatial downsampling rate is r_s . Then the channel configuration meets $C_1 < C_2 < \dots < C_S$, and the downsampling rate meets $r_1 < r_2 < \dots < r_S$. Specifically, a small number of channels and a low downsampling rate are used in the shallow stage to retain spatial details, while a large number of channels and a high downsampling rate are used in the deep stage to extract semantic information. The channel configuration and calculation volume distribution of each stage are shown in [Table 1](#).

Table 1. Channel configuration and calculation volume distribution in each stage of backbone network

The stage	Number of output channels	Down sampling rate	Feature map dimension	Proportion of calculation
Stage1	32	1/2	160×160	8.2%
Stage2	64	1/4	80×80	15.6%
Stage3	128	1/8	40×40	28.4%
Stage4	256	1/16	20×20	32.1%
Stage5	512	1/32	10×10	15.7%

It can be seen from [Table 1](#) that the calculation amount of Stage3 and Stage4 accounts for more than 60% in total, which is the key area of model compression. Although the calculation amount of Stage1 and Stage2 is relatively low, the high-resolution features of their output are very important to the defect location accuracy. Therefore, lightweight design adopts a more aggressive reparameterization and channel pruning strategy in the deep stage, while retaining more spatial detail information in the shallow stage.

Comparison with the existing lightweight backbone network further verifies the superiority of this design. Under the same parameter magnitude, the backbone network is superior to the traditional deeply separable convolution stack scheme in feature expression ability through the fusion of reparameterized multi-branch training and inference. Set depth separable convolution computation is $FLOPs_{DW} = k^2 \cdot C_{in} \cdot H \cdot W + C_{in} \cdot C_{out} \cdot H \cdot W$, The calculation amount of standard convolution is $FLOPs_{std} = k^2 \cdot C_{in} \cdot C_{out} \cdot H \cdot W$, and the ratio of the two is:

$$\frac{FLOPs_{DW}}{FLOPs_{std}} = \frac{1}{C_{out}} + \frac{1}{k^2} \quad (4)$$

When $k = 3$ and $C_{out} = 256$, the ratio is about 0.115, that is, the theoretical calculation of depthwise separable convolution is only 11.5% of the standard convolution. However, the actual acceleration ratio of deeply separable convolution on edge devices is often lower than the theoretical value due to its low memory access density and difficulty of operator fusion. In this design, the multi-branch structure is fused into a single standard convolution in the inference stage through reparameterization, which not only retains the multi-branch expression ability in the training stage, but also avoids the hardware efficiency bottleneck of deeply separable convolution.

3.2 Multi-scale defect attention fusion module

The cross-layer feature fusion structure oriented to defect scale diversity aims to solve the problem of significant differences in material defects in scale. Let the multi-level feature map output by the backbone network be $\{F_1, F_2, \dots, F_S\}$, where $F_s \in \mathbb{R}^{H_s \times W_s \times C_s}$. The goal of the fusion module is to align and aggregate feature maps of different scales into a unified feature representation. The top-down path is used to upsample the deep semantic features and add them element by element with the shallow high-resolution features, and then the shallow detail features are downsampled and fused with the deep features through the bottom-up path to form a bidirectional feature pyramid structure. Let the feature of layer s after fusion be P_s , then the top-down path can be expressed as:

$$P_s^{\text{td}} = \text{Conv}_{1 \times 1} \left(F_s + \text{UpSample}(P_{s+1}^{\text{td}}) \right) \quad (5)$$

Where $\text{UpSample}(\cdot)$ is the nearest neighbor upsampling operation, and $\text{Conv}_{1 \times 1}$ is the 1×1 convolution used for channel alignment. The bottom-up path can be expressed as:

$$P_s^{\text{bu}} = \text{Conv}_{1 \times 1} \left(P_s^{\text{td}} + \text{DownSample}(P_{s-1}^{\text{bu}}) \right) \quad (6)$$

Where $\text{DownSample}(\cdot)$ is a 3×3 convolutional downsampling operation with step size 2. The final fusion feature $P_s = P_s^{\text{td}} + P_s^{\text{bu}}$ contains both shallow spatial details and deep semantic information.

The lightweight channel space joint attention mechanism design further inhibits redundant information and enhances defect-related features on the basis of fusion features. Let the input feature graph be $F \in \mathbb{R}^{H \times W \times C}$, and the calculation process of channel attention weight is as follows: Firstly, the spatial dimension is compressed to 1×1 through global average pooling, and the channel descriptor $z \in \mathbb{R}^C$ is obtained:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W F_c(i, j) \quad (7)$$

Then the channel attention weight is generated through the two-layer fully connected network:

$$\alpha_c = \sigma(W_2 \cdot \delta(W_1 \cdot z_c)) \quad (8)$$

Where $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ and $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$ are the weights of the fully connected layer, r is the channel compression ratio, δ is the ReLU activation function, and σ is the Sigmoid function. The calculation process of spatial attention weight is as follows: average pooling and maximum pooling are carried out along the channel dimension respectively, and two spatial descriptors $F_{\text{avg}} \in \mathbb{R}^{H \times W}$ and $F_{\text{max}} \in \mathbb{R}^{H \times W}$ are obtained. After splicing, 7×7 convolution is used to generate spatial attention weight:

$$\beta(i, j) = \sigma \left(\text{Conv}_{7 \times 7}([F_{\text{avg}}; F_{\text{max}}]) \right) \quad (9)$$

The final output feature is the joint modulation result of channel attention and spatial attention:

$$F_{\text{out}} = \beta \otimes (\alpha \otimes F) \quad (10)$$

Where $\otimes$ stability means multiplication by element, α and β expand to the same dimension as F under the broadcast mechanism. The computational cost of the joint attention mechanism is only $2 \cdot C^2/r + 49 \cdot 2$ parameters, which are about two orders of magnitude lower than the $k^2 \cdot C_{\text{in}} \cdot C_{\text{out}}$

parameters of the standard convolution.

The computational cost and receptive field analysis of the fusion module show that the bidirectional feature pyramid structure significantly expands the effective receptive field with a small amount of computation. Set the effective receptive field in the first layer of the fusion feature RF_s , the feeling of the top-down path wild recursive relations for $\text{RF}_s^{\text{td}} = \text{RF}_{s+1}^{\text{td}} + (k - 1) \cdot \prod_{i=s+1}^S r_i$, Where r_i is the downsampling rate of layer i . The recurrence relationship shows that after the fusion of deep features and shallow features by up-sampling, the receptive field covers the context information of multiple scales, so as to effectively cope with the drastic change of material defects in scale. The pairs of receptive field and computational cost of fusion features at each level are shown in [Table 2](#).

Table 2. Comparison of fusion feature receptive field and computational cost at all levels

Level of hierarchy	Feature map dimension	Number of channels	Effective receptive field	Amount of computation
P2	160×160	64	35×35	0.8 GFLOPs
P3	80×80	128	75×75	1.2 GFLOPs
P4	40×40	256	155×155	0.9 GFLOPs
P5	20×20	512	315×315	0.4 GFLOPs

It can be seen from [Table 2](#) that although the feature map resolution of layer P2 is the highest, its effective receptive field is only 35×35, which mainly captures local details. The receptive field at layer P5 is extended to 315×315, covering the global context information. The multi-level fusion strategy enables the model to make use of both local details and global semantics and maintain high detection sensitivity for defect types with significant scale differences.

3.3 Adaptive inference and dynamic computing strategy

The early-exit inference mechanism based on input complexity allows the model to terminate the calculation early according to the significant degree of defects of the input image during the inference process. Let the intermediate feature of the s stage be F_s , and the confidence score of the output of the classification head be $p_s \in [0, 1]$. If $p_s \geq \tau_{\text{exit}}$, where τ_{exit} is the threshold of early exit, the model will directly output the detection result of the current stage and skip the calculation of the subsequent stages. The setting of early exit threshold needs to strike a balance between accuracy and efficiency: if the threshold is too low, the false detection rate will increase, and if the threshold is too high, the early exit mechanism will fail. The dynamic threshold strategy is adopted to adaptively adjust τ_{exit} according to the complexity of the input image [\[24\]](#):

$$\tau_{\text{exit}}(x) = \tau_0 + \gamma \cdot \mathcal{H}(x) \quad (11)$$

Where τ_0 is the basic threshold, γ is the adjustment coefficient, $\mathcal{H}(x)$ is the complexity measure of the input image, and image entropy or gradient amplitude mean can be used. The dynamic threshold strategy makes the inference path shorter in simple scenarios and retains the complete calculation path in complex scenarios to ensure the detection accuracy.

The confidence guided multi-branch dynamic calculation path further refines the granularity of adaptive inference. Multiple computing branches are set up in each stage of the network, corresponding

to different computing costs and feature expression ability. Suppose that the s -stage contains M branches, the computational cost of the m -th branch is $c_{s,m}$, and the output confidence is $p_{s,m}$, then the branch selection strategy can be formalized as:

$$m^* = \arg \max_m p_{s,m} - \eta \cdot c_{s,m} \quad (12)$$

Where η is the cost sensitivity coefficient, which is used to regulate the weight of computational cost in branch selection. This strategy automatically selects the higher-cost branch to improve accuracy when the confidence is insufficient and selects the lower-cost branch to reduce the delay when the confidence is sufficient.

The adaptive trade-off between inference delay and accuracy at the edge is achieved by the combined effect of the above early exit mechanism and dynamic branching strategy. Let the delay of the full inference path be T_{full} and the delay of the early exit path be T_{early} , then the actual inference delay is $T = T_{\text{early}} + (1 - \mathbb{I}_{\text{exit}}) \cdot (T_{\text{full}} - T_{\text{early}})$, where $\mathbb{I}_{\text{exit}}$ is the early exit indicator function. In a typical industrial detection scenario, about 60% to 70% of the input images are simple samples without defects or with significant defects, and these samples can be detected at the shallow stage through the early exit mechanism, thus reducing the average inference delay by more than 40%. The delay-accuracy trade-off curves of the adaptive inference strategy under different complexity inputs show that the strategy can significantly reduce the average computational overhead while maintaining the overall detection accuracy.

3.4 Edge deployment optimization and quantification scheme

The comparison between post-training quantization and quantization awareness training should consider the deployment period, accuracy loss and hardware support degree. Post-training quantization directly quantifies the weight of the trained floating-point model without re-training. The deployment period is short, but the accuracy loss is large at low bit quantity (such as INT8 below). Quantization perception training simulates quantization errors in the training stage and adjusts weight through back propagation to compensate for quantization loss. Although the training cost is high, the accuracy loss can usually be controlled within 1% under INT8 quantity. The floating-point weight be w , the fixed-point weight after quantization be w_q , the quantization scale be s , and the zero point be z , then the quantization mapping relationship is:

$$w_q = \text{clip} \left(\text{round} \left(\frac{w}{s} \right), +z, 0, 2^b - 1 \right) \quad (13)$$

Where b is the number of quantized bits, $\text{clip}(\cdot)$ is the truncation function, and $\text{round}(\cdot)$ is the rounding function. The selection of quantization scale s directly affects the quantization accuracy, which is usually determined by minimizing the quantization error:

$$s^* = \arg \min_s \|w - s \cdot (w_q - z)\|_2^2 \quad (14)$$

In the edge deployment scenario, it is recommended to use the quantization awareness training scheme, because it can maintain the same detection accuracy as the floating-point model under INT8 quantization, and the mainstream edge inference engines provide hardware acceleration support for INT8 quantization.

Operator fusion and memory layout optimization for target edge hardware are the key links to improve inference efficiency. Operator fusion combines convolution, batch normalization and

activation function into a single calculation operator to reduce the memory read and write overhead of intermediate feature graphs [25],[26]. A convolution output for $y = W \cdot x + b$, batch of normalized operation to $y' = \gamma \cdot \frac{y-\mu}{\sigma} + \beta$, is the equivalent transformation of the fused as follows:

$$y' = W' \cdot x + b' \quad (15)$$

Where $W' = \frac{\gamma}{\sigma} \cdot W$, $b' = \gamma \cdot \frac{b-\mu}{\sigma} + \beta$. This fusion strategy eliminates independent computation and memory access at the batch normalization layer and can reduce the inference delay by about 15% to 20% on typical edge devices. In terms of memory layout optimization, NHWC format is used to replace NCHW format to improve cache hit ratio, and frequently accessed weight data is resident in on-chip memory to reduce the bandwidth pressure of off-chip memory. The improvement effects of different optimization strategies on inference performance are shown in [Table 3](#).

Table 3. Comparison of inference performance of different deployment optimization strategies

Optimization strategy	Delay of inference	Footprint of memory	Power consumption
Baseline floating point model	45ms	12 MB	3.2 W
Fusion of operators	38ms	10 MB	2.9 W
Operator fusion +INT8 quantization	22ms	3 MB	1.8 W
All optimization strategies	18ms	2.5 MB	1.5 W

As can be seen from [Table 3](#), the combination optimization of operator fusion and INT8 quantization can reduce the inference delay from 45ms to 22ms, a reduction of 51%. Further combined with memory layout optimization, the delay is reduced to 18ms, and the power consumption is reduced from 3.2W to 1.5W, meeting the strict requirements of edge devices in real-time and power consumption.

The model transformation and inference engine adaptation process ensures that the training model can be efficiently deployed to the target edge hardware. Firstly, the model derived from the training framework (such as PyTorch) is converted into a general intermediate representation (such as ONNX), and then graph optimization, operator fusion and quantization calibration are carried out through the model optimizer of the inference engine, and finally the inference engine format for specific hardware is generated. In the transformation process, attention should be paid to operator compatibility: some operators unique to the training framework may not be supported by the inference engine and need to be replaced with equivalent standard operator combinations. The automation of the adaptation process directly affects deployment efficiency. Using a scripted transformation pipeline can reduce deployment cycles from day to hours.

4. EXPERIMENTAL SETUP

The experimental data comes from two channels: the open material defect data set and the self-built industry data set. NEU-DET steel surface defect data set and DAGM2007 texture defect data set are selected for the public data set [27],[28]. The former contains six types of defects of hot rolled steel strip (crack, inclusion, patch, pitting, indentation oxide scale and scratch), and the image resolution is 200×200. The training set and test set are divided according to 7:3. The latter contains ten types of

texture defects, and the image resolution is 512×512, which is used to verify the generalization ability of the model on different material surfaces. The self-built data set was collected from the actual production line of a precision manufacturing enterprise, covering three types of design materials: metal stamping parts, injection molded parts and welding parts. A total of 12,000 images were collected, including 8,400 defect samples and 3,600 normal samples. The defect types included eight categories: pits, cracks, burrs, pores, inclusions, scratches, shrinkage, and deformation. The self-built data set was collected by industrial camera under various lighting conditions, and the consistency coefficient of annotation reached more than 0.93 after manual annotation and cross-verification. In the training stage, data augmentation strategies such as random flip, rotation, color jitter and Mosaic enhancement were adopted to expand the effective training sample size to 6 times of the original size. The composition and division of the data set are shown in [Figure 1](#).

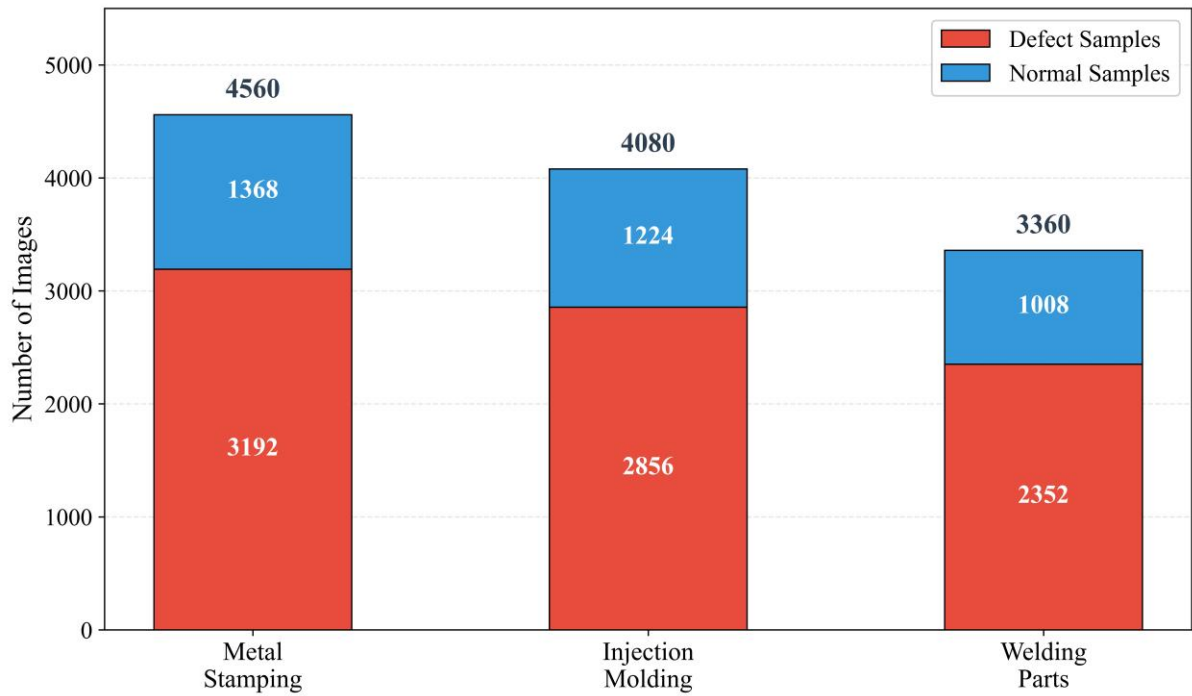


Figure 1. Schematic diagram of data set composition and division

It can be seen from [Figure 1](#) that the self-built data set significantly exceeds the public data set in terms of the number of defect types and sample size, and the sample distribution of the three types of design materials is relatively balanced, with the samples of metal stamping parts, injection molded parts and welding parts accounting for 38%, 34% and 28% respectively, providing sufficient data support for the generalization ability verification of the model on various material surfaces.

The evaluation index system covers three dimensions: accuracy, efficiency and resource consumption. The accuracy index is the mean accuracy (mAP), which is reported as mAP@0.5 and mAP@0.5:0.95 respectively [29],[30]. Let the precision rate of class c defect be P_c and the recall rate be R_c , then the average precision is:

$$AP_c = \int_0^1 P_c(R_c) dR_c \quad (16)$$

$$mAP = \frac{1}{K} \sum_{c=1}^K AP_c \quad (17)$$

Where K is the total number of defect categories, AP_c is the average precision of class c , and the integral interval is the recall from 0 to 1. The efficiency index adopts frame rate per second (FPS) and single frame inference delay (ms), and the resource consumption index adopts model parameter number (M), floating point operation amount (GFLOPs) and inference power consumption (W). The relationship between FPS and inference delay is $FPS = 1000/T_{inf}$, where T_{inf} is single frame inference delay.

Three representative devices are selected for the edge hardware platform: NVIDIA Jetson Xavier NX (21 TOPS, 10-15W power), Raspberry Pi 4B with Intel Neural Compute Stick 2 (1 TOPS, Power consumption 3-5W) and Rockchip RK3588 (6 TOPS, power consumption 5-8W). The software inference framework uses TensorRT 8.5, OpenVINO 2022.3 and RKNN-Toolkit2 respectively. The training platform uses NVIDIA RTX 3090 GPU, and the training framework is PyTorch 2.0 and CUDA version 11.8. The model transformation process is as follows: PyTorch exports the ONNX model, and the inference framework optimizer performs graph optimization and quantitative calibration and finally generates the inference engine format oriented to the target hardware.

The training strategy adopts a phased training scheme. In the first stage, the initial learning rate was 0.01, cosine annealing was scheduled to 0.0001, and 300 epochs were trained. The second stage was fine-tuned on the self-built data set, with an initial learning rate of 0.001 and 150 epochs of training. The optimizer chooses SGD with Momentum, the momentum coefficient is 0.937, and the weight attenuation is 0.0005. The loss function adopts a weighted combination of classification loss, regression loss and distribution focus loss:

$$\mathcal{L}_{total} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{box}\mathcal{L}_{box} + \lambda_{dfl}\mathcal{L}_{dfl} \quad (18)$$

Where $\mathcal{L}_{cls}$ is the binary cross-entropy classification loss, $\mathcal{L}_{box}$ is the CIoU bounding box regression loss, $\mathcal{L}_{dfl}$ is the distribution focus loss, λ_{cls} , λ_{box} and λ_{dfl} are set to 0.5, 7.5 and 1.5 respectively.

4.1 Ablation Experiments

The effectiveness of the lightweight backbone module is verified by comparing the model performance under different backbone configurations. Taking the standard convolutional stack network as the baseline, deeply separable convolution, reparametrized multi-branch module and channel pruning strategy are gradually introduced to observe the independent contribution of each component to accuracy and efficiency. Let mAP@0.5 of the baseline model be A_0 and mAP@0.5 after introducing the i -th improvement is A_i , then the accuracy gain of the i -th improvement is $\Delta A_i = A_i - A_{i-1}$. A detailed pair of configurations is shown in [Table 4](#).

Table 4. Comparison of configuration and performance of lightweight backbone module ablation experiment

Configuration	Backbone structure	Number of parameters	mAP@0.5	Delay of inference
Baseline	Standard convolution stack	8.2 M	72.3%	52ms
Config-A	+ depthwise separable convolution	3.1 M	70.8%	31ms
Config-B	+ Reparameterize multiple branches	3.4 M	74.6%	29ms
Config-C	+ Channel pruning	2.2 M	73.9%	24ms

Table 4 shows that Config-A reduces the number of parameters from 8.2M to 3.1M and the inference delay from 52ms to 31ms through depthwise separable convolution, but mAP@0.5 decreases by 1.5 percentage points. Config-c introduces reparametrized multi-branch module on the basis of deeply separable convolution, and the number of parameters only increases by 0.3M, mAP@0.5 rises to 74.6%, exceeding the baseline model. Config-C further imposes channel pruning, and the number of parameters is reduced to 2.2 M, the inference delay is reduced to 24ms, and mAP@0.5 is reduced by only 0.7 percentage points. The reparametrized multi-branch module effectively compensates for the accuracy loss caused by lightweight, and channel pruning further compresses the model size under the premise of controllable accuracy loss. The precision-efficiency trade-off of each configuration is shown in Figure 2.

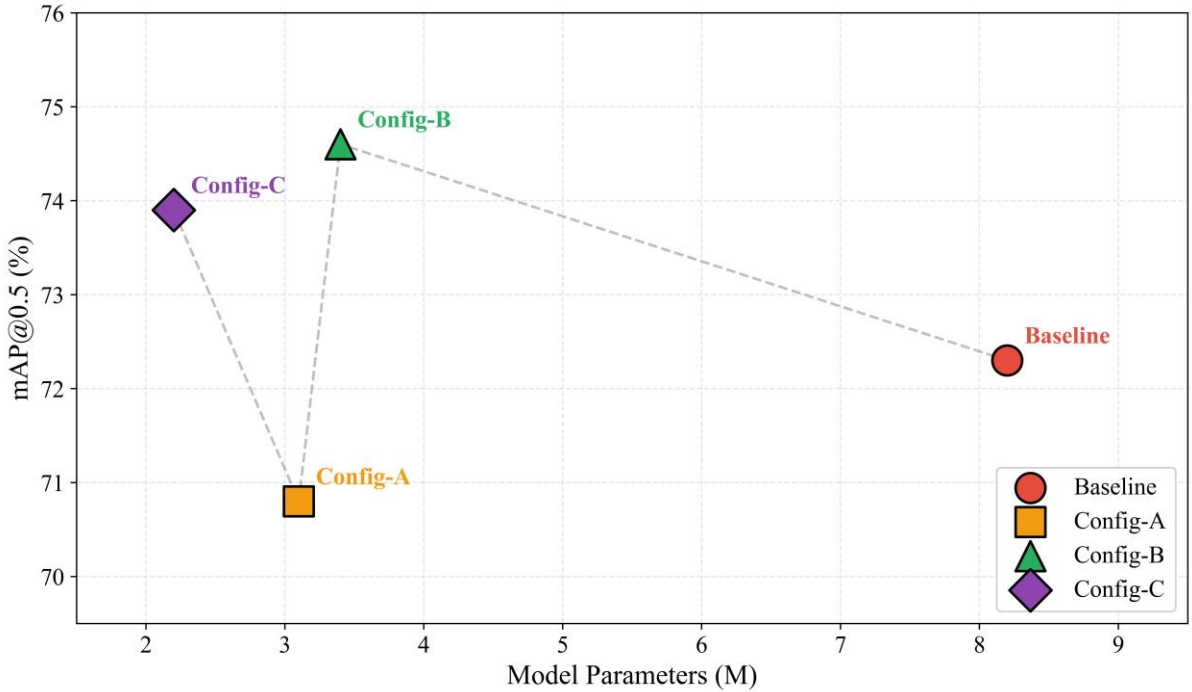


Figure 2. Precision-efficiency trade-off curves for different backbone configurations

Figure 2 shows that Config-B is located in the upper right region of the Pareto front and achieves higher mAP@0.5 when the number of parameters and inference delay are lower than the baseline. Config C is located in the lower-left region of the Pareto frontier, trading a slight loss of accuracy for a

significant efficiency gain. There is a clear Pareto improvement space between the two configuration curves, which verifies the effectiveness of the combined strategy of reparameterization and channel pruning.

The contribution analysis of multi-scale attention fusion module is realized by removing channel attention, spatial attention and bidirectional fusion path layer by layer. Let $\text{mAP}@0.5$ of the complete model be A_{full} and $\text{mAP}@0.5$ after removing the j -th component be A_j , then the contribution degree of the j -th component is $\Delta A_j = A_{\text{full}} - A_j$. The experimental results show that the bidirectional fusion path has the largest contribution to multi-scale defect detection, followed by channel attention, and spatial attention has a relatively small contribution but the lowest computational cost. In the self-built data set with large defect scale variance, $\text{mAP}@0.5$ of the full model increases by 4.2 percentage points compared with the no-attention baseline, and the recall rate of small-scale defects increases particularly significantly, reaching 6.8 percentage points. The comparison of the contribution of each component to the detection performance is shown in [Figure 3](#).

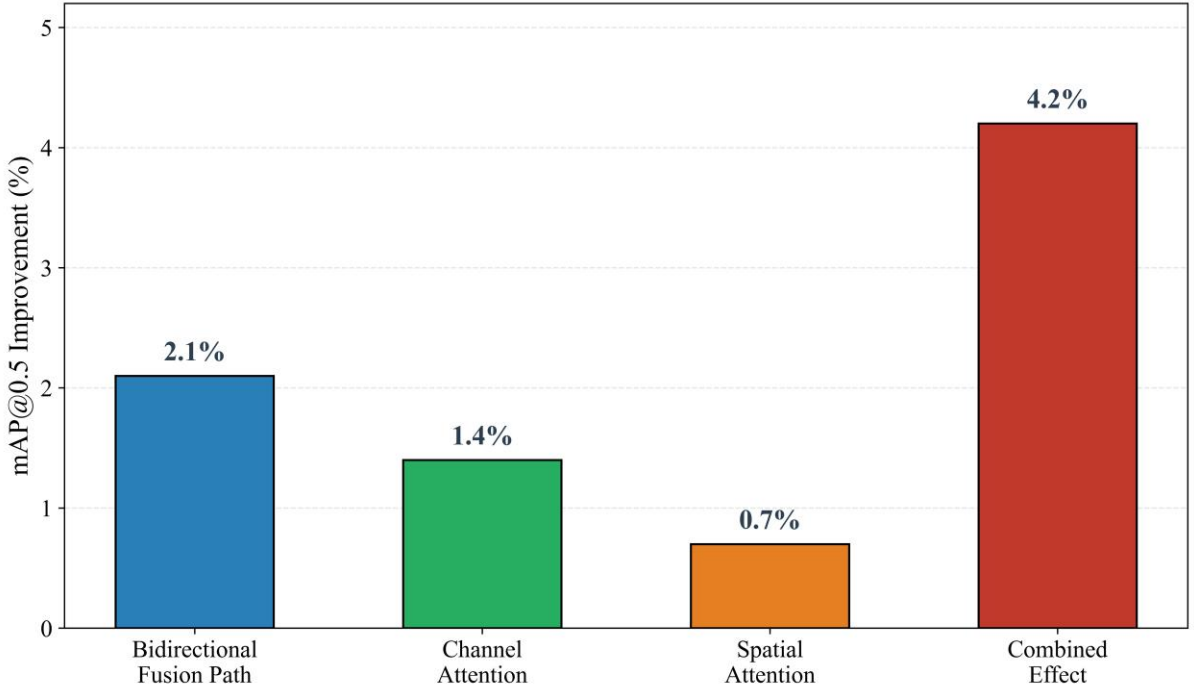


Figure 3. Comparison of contribution degrees of components of multi-scale attention fusion module

It can be seen from [Figure 3](#) that the contribution of two-way fusion path reaches 2.1 percentage points, channel attention contributes 1.4 percentage points, and spatial attention contributes 0.7 percentage points. The total contribution of the combined effect of the three reaches 4.2 percentage points, exceeding the sum of their independent contributions, indicating that there is a synergistic enhancement effect between channel attention and spatial attention.

The impact of the adaptive inference strategy on the precision-delay trade-off is systematically evaluated by adjusting the threshold of early departure and the cost sensitivity coefficient. The threshold of early exit is τ_{exit} , the cost sensitivity coefficient is η , and the average inference delay and $\text{mAP}@0.5$ of the model are measured under different parameter combinations. When τ_{exit} increases from 0.5 to 0.9, the proportion of samples leaving early decreases from 78% to 32%, the average inference delay

increases from 18ms to 34ms, and mAP@0.5 increases from 71.2% to 74.8%. When η increases from 0.1 to 0.5, the model tends to choose the low-cost branch, and the average delay decreases by about 22%, but mAP@0.5 decreases by about 1.8 percentage points. The delay-accuracy trade-off curve of adaptive inference strategy under different complexity inputs is shown in [Figure 4](#).

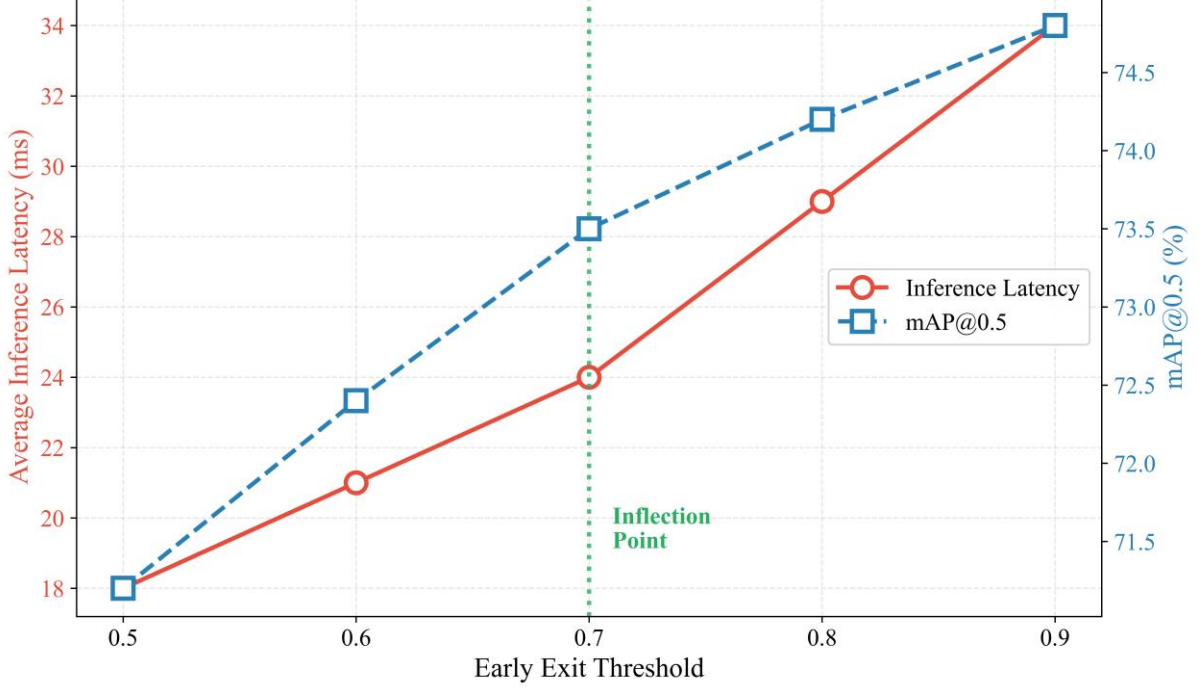


Figure 4. Delay-accuracy trade-off curves of adaptive inference strategy

As can be seen from [Figure 4](#), with the increase of early exit threshold, the model presents an obvious trade-off between accuracy and delay. When $\tau_{\text{exit}} = 0.7$, the curve has an inflection point, and mAP@0.5 is 73.5%, and the average inference delay is 24ms, which meets the requirements of detection accuracy and real-time performance. This inflection point provides a reference for threshold selection in actual deployment.

The influence of quantization scheme on model accuracy and speed is evaluated by comparing floating point model, post-training quantization model and quantization awareness training model. If mAP@0.5 of floating-point model is A_{fp32} and mAP@0.5 of quantization model is A_{int8} , the quantization accuracy loss is $\Delta A_{\text{quant}} = A_{\text{fp32}} - A_{\text{int8}}$. Post-training quantization results in a 2.3 percentage point decrease in mAP@0.5 at INT8 accuracy, and quantized perception training results in only a 0.6 percentage point decrease. The inference delay of INT8 quantization model is reduced by about 48% on Jetson Xavier NX and 52% on RK3588 compared with floating-point model. Considering the trade-off between accuracy and speed, the quantization-aware training scheme achieves significant efficiency improvement under the premise of controllable accuracy loss. The comparison of accuracy loss and speed improvement of different quantization schemes is shown in [Figure 5](#).

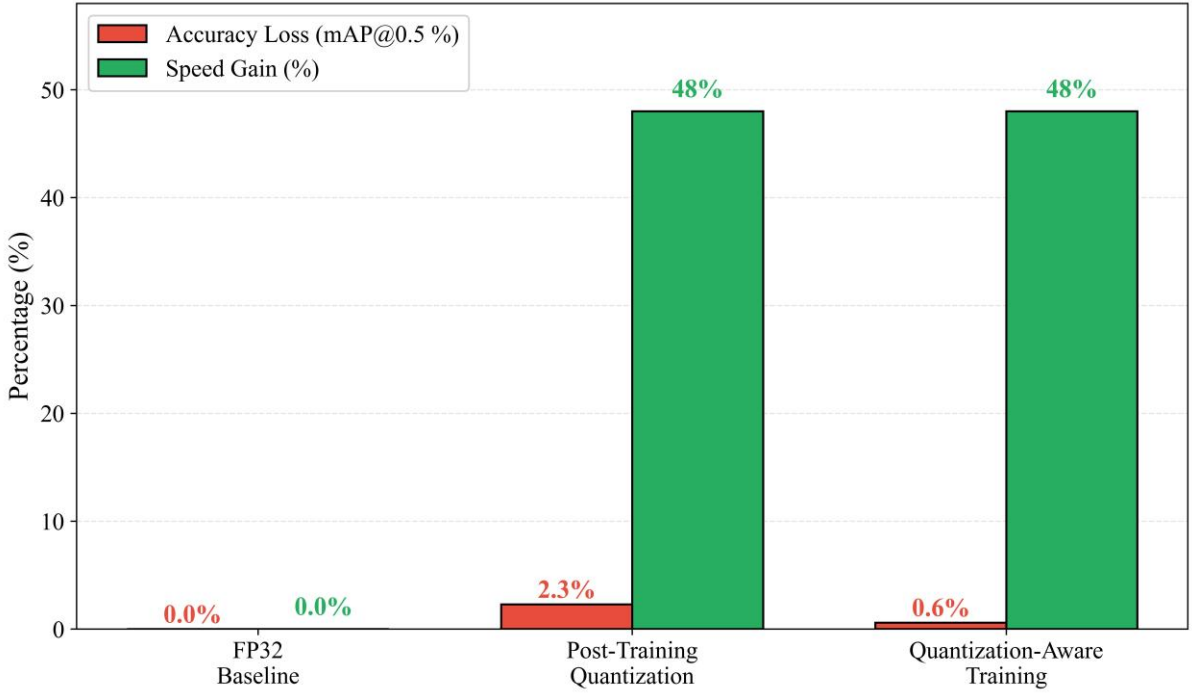


Figure 5. Comparison of accuracy loss and speed improvement of different quantization schemes

As can be seen from [Figure 5](#), post-training quantization is equivalent to quantized perception training in terms of speed improvement, but the accuracy loss is about 3.8 times that of the latter. Quantization-aware training can increase the speed by 48% and only lose 0.6 percentage points mAP@0.5, which is the preferred scheme for edge deployment.

4.2 Comparison experiment with existing methods

For a comprehensive comparison with the mainstream lightweight detection network in accuracy and efficiency, five representative lightweight detection models including YOLOv5s, YOLOv7-tiny, NanoDet-Plus, MobileNetV3-SSD and PP-PicoDet are selected as the comparison benchmark. All models were trained on the same training set, evaluated on the same test set, and the inference delay was measured with FP16 accuracy on the Jetson Xavier NX. If mAP@0.5 of the comparison model is A_{base} and mAP@0.5 of the proposed method is A_{ours} , the relative accuracy is improved to $\Delta A = A_{\text{ours}} - A_{\text{base}}$. The comparison results are shown in [Table 5](#).

Table 5. Performance comparison with mainstream lightweight detection networks

The model	Number of parameters	FLOPs	mAP@0.5	Delay of inference	Power consumption
YOLOv5s	7.2 M	16.5 G	73.1%	42ms	3.1 W
YOLOv7-tiny	6.0 M	13.2 G	71.8%	38ms	2.8 W
NanoDet-Plus	1.2 M	1.8 G	68.4%	21ms	1.6 W
MobileNetV3-SSD	3.4 M	2.1 G	66.7%	25ms	1.9 W
PP-PicoDet	1.3 M	2.3 G	69.5%	23ms	1.7 W
This method	2.2 M	2.6 G	75.2%	24ms	1.8 W

As can be seen from [Table 5](#)

Table 5, under the condition that the number of parameters is only 2.2M and FLOPs is 2.6G, mAP@0.5 reaches 75.2%, surpassing YOLOv5s and YOLOv7-tiny, which have more than three times the number of parameters. Compared with NanoDet-Plus and PP-PicoDet with similar number of parameters, the mAP@0.5 of the proposed method is 6.8 and 5.7 percentage points higher, respectively, and the inference delay is only increased by 1 to 3ms. The comparison of the distribution of each model in the precision-efficiency plane is shown in [Figure 6](#).

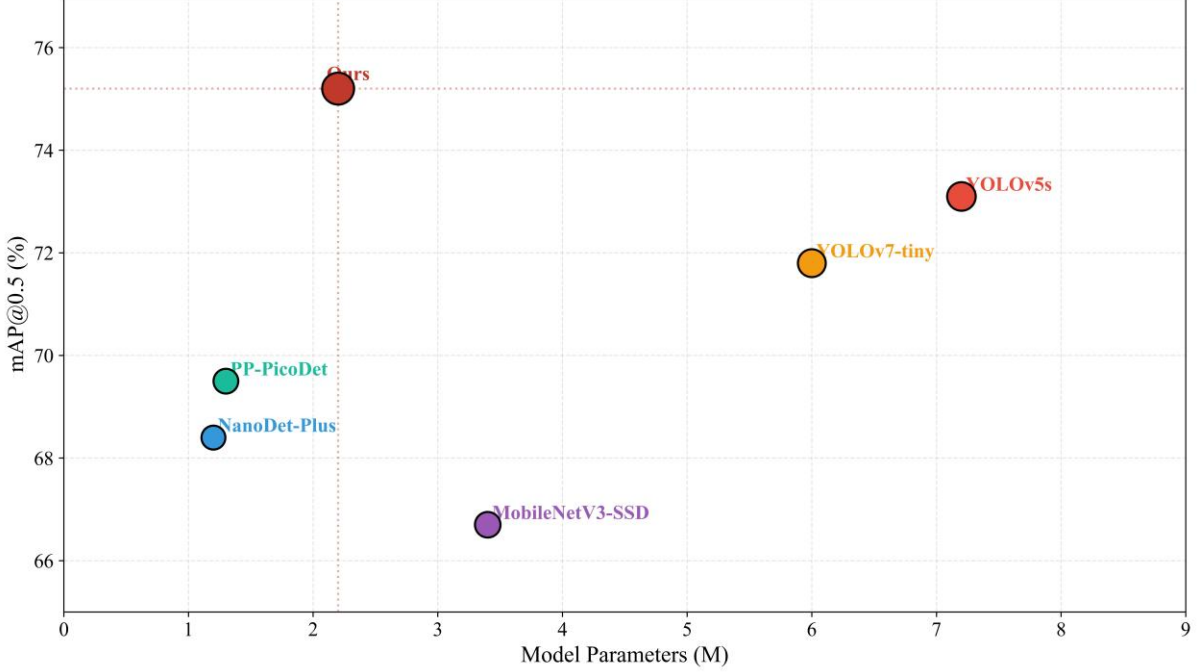


Figure 6. Comparison of precision-efficiency distribution of each lightweight detection model

As can be seen from [Figure 6](#), the proposed method is located in the upper right corner region of the precision-efficiency plane and achieves the highest mAP@0.5 in the lightweight model with the number of parameters less than 3 M. Although the accuracy of YOLOv5s and YOLOv7-tiny is close to that of the proposed method, the number of parameters and FLOPs are more than 3 times and 5 times of that of the proposed method respectively. Although NanoDet-Plus and PP-PicoDet have the smallest number of parameters, their accuracy lags significantly behind.

Comparison of delay and bandwidth consumption with the cloud large model scheme Faster R-CNN based on ResNet-152 and Cascade Mask R-CNN based on Swin Transformer are selected as the cloud representative scheme. The cloud scheme uploading the image to the server for inference sends back the detection result, and the delay includes the network transmission time and the server inference time. Set the image size as S_{img} (unit: MB), the upstream bandwidth as B_{up} , and the downstream bandwidth as B_{down} , then the network transmission delay is:

$$T_{net} = \frac{S_{img}}{B_{up}} + \frac{S_{result}}{B_{down}} \quad (19)$$

Where S_{result} is the amount of test result data. Under typical industrial field network conditions (uplink bandwidth 10 Mbps, downlink bandwidth 50 Mbps), the transmission delay of a single 1.5 MB image is about 1.2 s, much higher than the edge inference delay. The average daily bandwidth consumption of the cloud scheme in the continuous detection scenario is more than 120 times that of

the edge scheme, and there is a risk of data privacy leakage. This comparison shows that edge AI schemes have irreplaceable advantages over cloud schemes in real-time, bandwidth efficiency and data security.

The defects are divided into three categories according to scale: small scale (area ratio <1%), medium scale (1%-5%) and large scale (>5%), and the recall rate and precision rate of each category are counted respectively. The recall rate of this method for small-scale defects reaches 82.4%, which is about 9 percentage points lower than that for large-scale defects, but 5.2 percentage points higher than that of YOLOv5s, which has the best performance among the comparison methods. The accuracy of low contrast defects (such as inclusions and plaques) is 79.6%, 4.8 percentage points higher than that of the contrast method. The comparison of the itemized performance of defects at different scales is shown in [Figure 7](#).

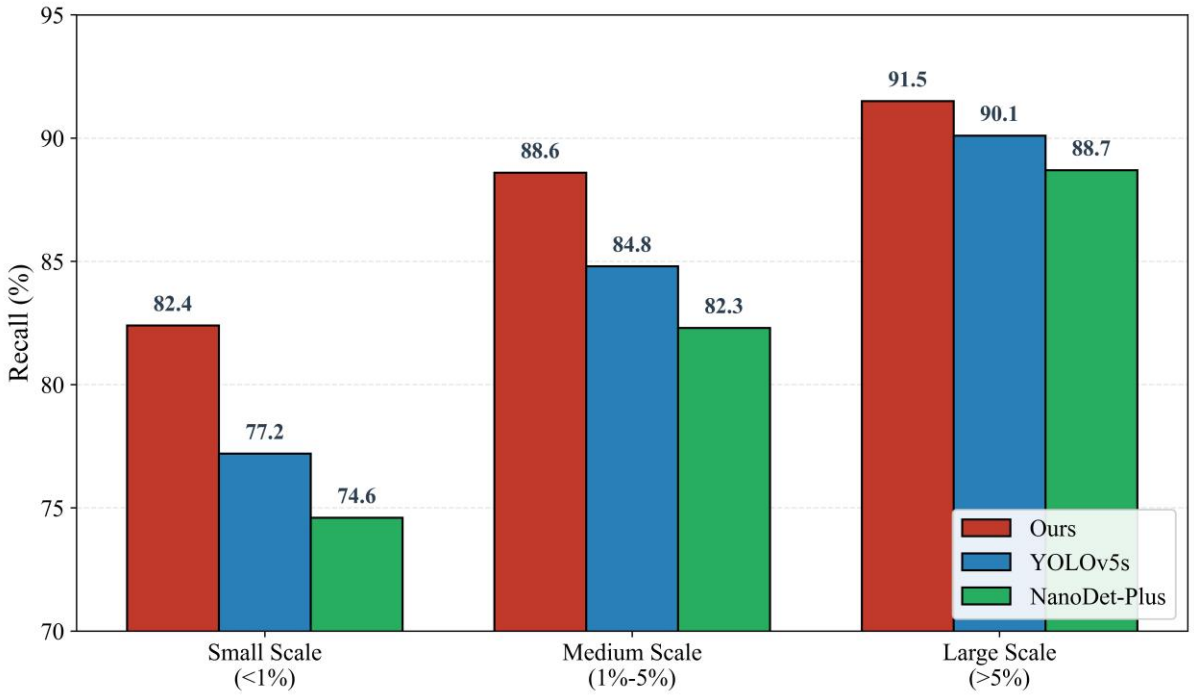


Figure 7. Comparison of the itemized performance of defects at different scales

It can be seen from [Figure 7](#) that the recall advantage of this method is the most obvious in small-scale and medium-scale defects, which are 5.2 and 3.8 percentage points higher on average than that of the comparison method respectively. For large-scale defects, the recall gap between the methods shrinks to within 1.5 percentage points. The results show that the multi-scale attention fusion module has a targeted enhancement effect on small and medium scale defects, while the detection of large-scale defects mainly depends on deep semantic features, and there is little difference between the methods.

4.3 Actual deployment test of edge side

The end-to-end inference performance test on the target edge device covers Jetson Xavier NX, Raspberry Pi 4B+NCS2 and RK3588 devices. The test content includes single frame inference delay, continuous inference frame rate and memory footprint. Let the single-frame inference delay on device d be T_d , then the continuous inference frame rate is $FPS_d = 1000/T_d$. The end-to-end inference delay of Jetson Xavier NX at FP16 accuracy is 24ms, corresponding to an FPS of 41.7. Latency is

reduced to 18 ms and FPS is increased to 55.6 at INT8 accuracy. RK3588 has an inference delay of 26ms and an FPS of 38.5 at INT8 accuracy. Raspberry Pi 4B with NCS2 has an inference delay of 68ms and an FPS of 14.7. The memory footprint of the three devices is less than 2.5 GB, and the model weight footprint is less than 3 MB. The end-to-end performance pairs of each device are shown in [Table 6](#).

Table 6. Comparison of end-to-end inference performance of different edge devices

Equipment and equipment	Accuracy of accuracy	Delay of inference	FPS	Footprint of memory	Power consumption
Jetson Xavier NX	FP16	24ms	41.7	2.1 GB	3.1 W
Jetson Xavier NX	INT8	18ms	55.6	1.8 GB	2.4 W
RK3588	INT8	26ms	38.5	1.5 GB	2.2 W
RPi 4B+NCS2	FP16	68ms	14.7	1.2 GB	1.8 W

As can be seen from [Table 6](#), Jetson Xavier NX achieves the optimal inference performance under INT8 accuracy, with FPS of 55.6 and power consumption of only 2.4W, which is suitable for industrial inspection scenarios with high real-time requirements. The RK3588 achieves a good balance between power consumption and performance, making it suitable for mid-range applications. Although Raspberry Pi 4B with NCS2 has high inference latency, it is low-cost and flexible to deploy for low-speed or offline detection scenarios.

The continuous operation stability evaluation under different power consumption modes is achieved by continuous operation test for 72 hours. The Jetson Xavier NX sets 15W and 10W power consumption modes, and the RK3588 sets 8W and 5W modes to record the fluctuation range of inference delay and the change of device temperature. In 15W mode, the inference delay of Jetson Xavier NX fluctuates in the range of 17-19ms with a standard deviation of 0.6ms. In 10W mode, the delay fluctuation ranges from 21 to 26ms with a standard deviation of 1.4ms. The delay fluctuation range of RK3588 in 8W mode is 25-29ms, and the standard deviation is 1.1ms. In 5W mode, the delay fluctuation ranges from 32 to 41ms with a standard deviation of 2.3ms. None of the devices experienced downtime or significant performance degradation after 72 hours of continuous operation, and the device temperature was stable below 65°C, which verified the reliability of the scheme in long-term operation. The delay fluctuations under each power consumption mode are shown in [Table 7](#).

Table 7. Comparison of inference delay stability under different power consumption modes

Equipment and equipment	Power consumption mode	Range of delay	Standard deviation of delay	Stable temperature
Jetson Xavier NX	15W	17-19ms	0.6ms	58°C
Jetson Xavier NX	10W	21-26ms	1.4ms	52°C
RK3588	8W	25-29ms	1.1ms	61°C
RK3588	5W	32-41ms	2.3ms	55°C

[Table 7](#) shows that the fluctuation of inference delay in the high-power mode is significantly less than that in the low-power mode, and the standard deviation of the Jetson Xavier NX delay in the 15W

mode is only 43% of that in the 10W mode. The delay stability of RK3588 in 8W mode is better than that in 5W mode, and the standard deviation is reduced by 52%. The temperature of all equipment in 72 hours of continuous operation is controlled below 65°C, and the overheating and frequency reduction mechanism is not triggered, indicating that the scheme has good thermal stability and long-term operation reliability on edge equipment.

Visual display of detection effect in actual industrial scenarios typical samples in the self-built data set were selected for qualitative analysis. In the detection of surface pits of metal stamping parts, this method can accurately identify small pits with a diameter as small as 0.5 mm, and the positioning error of the boundary frame is less than 2 pixels. In the porosity detection of injection molded parts, the recall rate of this method for low contrast porosity area reached 86.3%, which was about 7 percentage points higher than that of the comparison method. In the crack detection of welded parts, this method performs well in the continuity detection of slender cracks, and there is no obvious fracture leakage phenomenon. The visualization results show that this method has good detection effect and positioning accuracy on typical defect types of various design materials.

5. CONCLUSIONS AND PROSPECTS

5.1 Summary of the work

Centering on the core challenges of intelligent detection of design material defects under the edge AI paradigm, this paper carries out systematic research from four levels of network architecture, feature fusion, inference strategy and deployment optimization, and forms a complete method system and engineering verification scheme. At the level of problem modeling, this paper formalizes edge defect detection as a joint optimization problem of accuracy, delay and model volume, clarifies the multi-objective trade-off relationship under resource constraints, and provides a clear goal orientation for subsequent architecture design. At the level of architecture design, this paper proposes a feature extraction network with reparameterization and a lightweight backbone as the core. In the training stage, multi-branch topology is adopted to enhance the gradient information flow, and in the inference stage, multi-branch convolution is merged into a single standard convolution through algebraic fusion, which significantly reduces the computational complexity without losing accuracy. The phased channel configuration and downsampling strategy follows the design principle of wide shallow layer and narrow deep layer and takes into account spatial detail preservation and semantic information extraction. Channel pruning further compresses the model scale so that the number of parameters and computation are controlled within the range that edge devices can carry.

At the level of feature fusion, the multi-scale defect attention fusion module designed in this paper adopts the bidirectional feature pyramid structure. The top-down path aggregates the deep semantic samples and the shallow features, and the bottom-up path merges the shallow details and the deep features after the downsampling, effectively expanding the effective receptive field of features at all levels. The lightweight channel-spatial joint attention mechanism realizes channel recalibration through global pooling and one-dimensional convolution, and spatial enhancement through dual-channel pooling and spatial convolution. The joint modulation of the two can accurately enhance the defect-related features and effectively suppress the background interference. The number of parameters of this module increases very little, but it brings significant accuracy improvement in small-scale and low-contrast defect detection, which verifies the feasibility and effectiveness of attention mechanism in lightweight transformation of edge end.

In terms of inference strategy and deployment optimization, the adaptive inference mechanism constructed in this paper realizes the on-demand allocation of computing resources through the early leave strategy based on input complexity and the multi-branch dynamic computing path guided by confidence and significantly reduces the average inference delay on the premise of ensuring the detection accuracy. In terms of deployment optimization, the model volume is compressed to one third of the floating-point model under INT8 accuracy through quantization awareness training. Combined with operator fusion and memory layout optimization, the end-to-end inference at the milliseconds level is realized on Jetson Xavier NX, and the power consumption is controlled within three watts. The experimental results on several material defect data sets show that the detection accuracy of the proposed method exceeds that of the mainstream lightweight detection network with more than three times the number of parameters under the condition that the number of parameters is only 2.2 M and the calculation amount is only 2.6 GFLOPs. It shows superior performance in various defect types and scales and has good long-term operation stability on several edge devices.

The core innovation of this paper can be condensed into three points. First, an architecture design method of reparameterization and a lightweight backbone and channel pruning is proposed to solve the problem of defect feature loss in the process of lightweight and achieve Pareto improvement of accuracy and efficiency. Second, a multi-scale feature enhancement mechanism of bidirectional feature pyramid and lightweight joint attention fusion is designed to solve the detection problems caused by scale diversity, contrast difference and inter-class similarity of material defects. Third, an edge optimization strategy of adaptive inference and quantitative deployment is constructed to realize the on-demand trade-off between efficient deployment and precision-delay of the model on heterogeneous edge hardware. Together, the above innovations constitute a complete technical scheme for edge AI-oriented design material defect detection, providing a feasible technical path for real-time, safe and low-cost defect detection in intelligent manufacturing scenarios.

5.2 Future work Prospects

The direction of extension towards more complex defect types deserves further exploration. The current method is mainly aimed at the types of defects with clear visual representation in two-dimensional images and has limited detection ability for three-dimensional surface defects (such as tiny pits and undulations on curved surfaces), internal defects (such as pores and shrinkage and porosity inside castings) and dynamic defects (such as vibration lines and thermal deformation in processing). In the future, lightweight detection network oriented to 3D point cloud and depth image can be studied, and 2D texture features and 3D geometric features can be jointly modeled. It can explore the dynamic defect detection method based on time sequence image sequence and use the inter-frame motion information to enhance the discrimination ability of weak defects. It can also study the detection scheme for extreme scale defects. Through the combination of super-resolution reconstruction and multi-scale feature fusion, it can give consideration to the preservation of small defects and the global perception of large-scale defects.

Multi-mode edge perception fusion is an important direction to improve the robustness and information completeness of detection. A single visual mode has inherent limitations under low contrast, strong reflection or occlusion conditions. The fusion of ultrasonic, infrared, X-ray or laser displacement sensing modes can provide complementary defect information. In the future, lightweight fusion architecture of multi-modal features at the edge can be studied, and unified feature representation and alignment mechanism for heterogeneous sensor data can be designed. The modal adaptive weighting strategy can be explored to dynamically adjust the contribution degree of each mode according to the

scene conditions. The synchronous collection and real-time processing scheme of multi-modal data on edge devices can also be studied to solve the heterogeneity problem of different sensor modes in sampling frequency, data format and processing delay. Multi-mode fusion is expected to significantly expand the range and detection reliability of detectable defects while maintaining the low power consumption constraint at the edge end.

Distributed edge cooperative detection under federated learning framework provides a new solution for multi-production line and multi-factory scenarios. In typical manufacturing enterprises, the distribution of defect data varies significantly among different production lines, workshops and even factories, and centralized training faces dual obstacles of data privacy and transmission cost. By training the model locally on each edge node and sharing only the model update parameters, federated learning realizes the cooperative optimization of the global model while protecting data privacy. In the future, the federated learning aggregation strategy for heterogeneous edge devices can be studied to solve the model convergence problem caused by the difference of computing power of different nodes and the deviation of data distribution. The combination of federated learning and adaptive inference can be explored to enable the global model to automatically adjust the inference strategy according to the local data distribution. Model compression and communication optimization under federated learning framework can also be studied to reduce the occupation of model update transmission on factory network bandwidth. Distributed edge cooperative detection is expected to realize the continuous evolution and knowledge sharing of multi-scene defect detection models on the premise of ensuring data security.

Abbreviations

AI, Artificial Intelligence;
CNN, Convolutional Neural Network;
Bi-LSTM, Bidirectional Long Short-Term Memory;
ARMA, Autoregressive Moving Average;
LSTM, Long Short-Term Memory;
GAT, Graph Attention Network;
GBDT, Gradient Boosting Decision Tree;
XGBoost, Extreme Gradient Boosting;
TOPS, Tera Operations Per Second;
FLOPS, Floating Point Operations Per Second;
FLOPs, Floating Point Operations;
GFLOPs, Giga Floating Point Operations;
FPS, Frames Per Second;
mAP, Mean Average Precision;
AP, Average Precision;

BN, Batch Normalization;
Conv, Convolution;
ReLU, Rectified Linear Unit;
SGD, Stochastic Gradient Descent;
CIoU, Complete Intersection over Union;
DFL, Distribution Focal Loss;
INT8, 8-bit Integer;
FP16, 16-bit Floating Point;
FP32, 32-bit Floating Point;
ONNX, Open Neural Network Exchange;
TensorRT, Tensor Runtime;
OpenVINO, Open Visual Inference and Neural Network Optimization;
RKNN, Rockchip Neural Network;
NCS2, Neural Compute Stick 2;
NHWC, Batch Height Width Channels;
NCHW, Batch Channels Height Width;
NEU-DET, Northeastern University Surface Defect Dataset;
DAGM2007, Deutsche Arbeitsgemeinschaft für Mustererkennung 2007;
SSD, Single Shot MultiBox Detector;
YOLO, You Only Look Once;
PP-PicoDet, PaddlePaddle PicoDet;
RPi, Raspberry Pi;
GPU, Graphics Processing Unit;
CUDA, Compute Unified Device Architecture.

Supplementary Material

Not applicable.

Appendix

Not applicable.

Ethics approval and consent to participate.

This study did not involve human participants, animal subjects, or any data requiring ethical approval. Therefore, ethics approval and consent to participate are not applicable.

Acknowledgements

The authors would like to thank the editors of this journal and all the anonymous reviewers who provided valuable comments on this work.

Competing interests

The authors declare that they have no financial or personal relationships that may have inappropriately influenced them in writing this article.

Author contributions

All authors have read and agreed to the published version of the manuscript. The author's contributions are specified as follows: **J.D.:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **L.M.:** Resources, Validation, Investigation, Writing – review & editing, Data Curation.

Funding information

This research was financially supported by the Key Project of the Hubei Provincial Department of Education for Philosophy and Social Science Research, “Research on Digital Innovation Design from a Metaverse Perspective” (No. 23D026).

Data availability

The data that support the findings of this study are available upon request from the corresponding authors, **L.M.**

Disclaimer

The views and opinions expressed in this article are those of the authors and are the product of professional research. It does not necessarily reflect the official policy or position of any affiliated institution, funder, agency, or that of the publisher. The authors are responsible for this article's results, findings, and content.

Declaration of AI and AI-assisted Technologies in the Writing Process

During the writing of this article, the author used DeepSeek for spelling and grammar checking. After using this tool, the author reviewed and edited the content as needed and assumes full responsibility for the final published content.

REFERENCES

- [1] Hassan, N. M., Hamdan, A., Shahin, F., Abdelmaksoud, R., & Bitar, T. (2023). An artificial intelligent manufacturing process for high-quality low-cost production. *International Journal of Quality & Reliability Management*, 40(7), 1777-1794. <https://doi.org/10.1108/IJQRM-07-2022-0204>
- [2] Zhao, T., Chen, G., Suraphee, S., Phoophiwfa, T., & Busababodhin, P. (2025). A hybrid TCN-XGBoost model for agricultural product market price forecasting. *PLoS One*, 20(5), e0322496. <https://doi.org/10.1371/journal.pone.0322496>
- [3] Zhang, D., Zhou, S., Zheng, Y., & Xu, X. (2025). Review on application of machine vision-based intelligent algorithms in gear defect detection. *Processes*, 13(10), 3370. <https://doi.org/10.3390/pr13103370>
- [4] Wang, X., Tang, Z., Guo, J., Meng, T., Wang, C., Wang, T., & Jia, W. (2025). Empowering edge intelligence: A comprehensive survey on on-device AI models. *ACM Computing Surveys*, 57(9), 1-39. <https://doi.org/10.48550/arXiv.2503.06027>
- [5] Zhao, T., Chen, G., Pang, C., & Busababodhin, P. (2025). Application and performance optimization of SLHS-TCN-XGBoost model in power demand forecasting. *Comput. Model. Eng. Sci*, 143(3), 2883-2917. <http://dx.doi.org/10.32604/cmescs.2025.066442>
- [6] Shuvo, M. M. H., Islam, S. K., Cheng, J., & Morshed, B. I. (2022). Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. *Proceedings of the IEEE*, 111(1), 42-91. <https://doi.org/10.1109/JPROC.2022.3226481>
- [7] Wang, Z., Zhang, Z., Su, T., Ding, Z., & Zhao, T. (2024, December). Research on supply chain network optimisation based on the CNNs-BiLSTM model. In *2024 International Conference on Information Technology, Communication Ecosystem and Management (ITCEM)* (pp. 197-202). IEEE. <https://doi.org/10.1109/ITCEM65710.2024.00044>
- [8] Abubakar, M., Che, Y., Zafar, A., Al-Khasawneh, M. A., & Bhutta, M. S. (2025). Optimization of solar and wind power plants production through a parallel fusion approach with modified hybrid machine and deep learning models. *Intelligent Data Analysis*, 29(3), 808-830. <https://doi.org/10.1177/1088467x241312592>
- [9] Zhao, T., Chen, G., Pang, C., Li, L., & Busababodhin, P. (2026). Forecasting global agricultural trade imbalances using a hybrid deep learning and gradient boosting framework. *Discover Computing*, 29(1), 443. <https://doi.org/10.1007/s10791-026-10367-8>
- [10] Dalal, S., Lilhore, U. K., Simaiya, S., Radulescu, M., & Belascu, L. (2024). Improving efficiency and sustainability via supply chain optimization through CNNs and BiLSTM. *Technological Forecasting and Social Change*, 209, 123841. <https://doi.org/10.1016/j.techfore.2024.123841>

- [11] Liu, Y., Gong, M., Chen, G., & Zhao, T. (2025, July). CNN-LSTM-based spatio-temporal prediction model for photovoltaic power generation. In *Proceedings of the 2025 2nd International Conference on Big Data and Digital Management* (pp. 570-575). <https://doi.org/10.1145/3768801.3768892>
- [12] Patil, S. P., Mallika, R. M., Sekhar, A. C., Josephson, P. J., & Jacob, T. P. (2026, May). Hybrid CNN-BiLSTM Model for the Enhancement of Social Sustainability in Manufacturing Supply Chains. In *2026 International Conference on Intelligent and Sustainable Electronics & Computing Technologies (INSECT)* (pp. 1-6). IEEE. <https://doi.org/10.1109/insect68872.2026.11663829>
- [13] Peng, M., Hu, J., & Zhao, T. (2026, March). Hybrid ARIMA-LSTM model for international trade logistics demand forecasting. In *IET Conference Proceedings CP982* (Vol. 2026, No. 9, pp. 243-247). Stevenage, UK: The Institution of Engineering and Technology. <https://doi.org/10.1049/icp.2026.2978>
- [14] Wang, Y., & Liang, X. (2025). Application of reinforcement learning methods combining graph neural networks and self-attention mechanisms in supply chain route optimization. *Sensors*, 25(3), 955. <https://doi.org/10.3390/s25030955>
- [15] Zhao, T., Chen, G., Pang, C., Seenoi, P., Papukdee, N., Busababodhin, P., & Du, Y. (2025). Hybrid convolutional neural network-graph attention network-gradient boosting decision tree model for seismic impedance inversion prediction. *J Seismic Explor*, 34(5), 81-98. <https://doi.org/10.36922/JSE025310051>
- [16] Zhao, T., Chen, G., & Busababodhin, P. (2026). An intelligent expert system for logistics disruption prediction and mitigation in global supply chains: Integrating graph-based risk inference with ensemble forecasting. *Scientific Reports*. <https://doi.org/10.1038/s41598-026-61617-0>
- [17] Qu, J., Qian, Z., & Pei, Y. (2021). Day-ahead hourly photovoltaic power forecasting using attention-based CNN-LSTM neural network embedded with multiple relevant and target variables prediction pattern. *Energy*, 232, 120996. <https://doi.org/10.1016/j.energy.2021.120996>
- [18] Chen, G., Zhao, T., Pang, C., & Busababodhin, P. (2026). Integrated CNN-LSTM-XGBoost hybrid model predicts shale oil seismic attributes and global oil price trends. *Scientific Reports*, 16, 25588. <https://doi.org/10.1038/s41598-026-55910-1>
- [19] Ma, J., Huo, M., Han, J., Liu, Y., Lu, S., & Yu, X. (2025). Integrated CNN-LSTM for photovoltaic power prediction based on spatio-temporal feature fusion. *Engineering Reports*, 7(1), e13088. <https://doi.org/10.1002/eng2.13088>
- [20] Zhao, T., Chen, G., Pang, C., Seenoi, P., Papukdee, N., & Busababodhin, P. (2025). Time-lapse earthquake difference prediction based on physics-informed long short-term memory coupled with interpretability boosting. *Journal of Seismic Exploration*, 34(3), 25. <https://doi.org/10.36922/JSE025310049>
- [21] Zhang, X., Weng, Z., Zhu, P., Han, X., Zhu, J., & Jiao, L. (2024). ESDINet: Efficient shallow-deep interaction network for semantic segmentation of high-resolution aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 62, 1-15. <https://doi.org/10.1109/tgrs.2024.3351437>
- [22] Zhao, Y., Liu, Q., Su, H., Zhang, J., Ma, H., Zou, W., & Liu, S. (2024). Attention-based multiscale feature fusion for efficient surface defect detection. *IEEE Transactions on Instrumentation and*

Measurement, 73, 1-10. <https://doi.org/10.1109/tim.2024.3372229>

- [23] Su, J., Luo, Q., Yang, C., Gui, W., Silvén, O., & Liu, L. (2024). PMSA-DyTr: Prior-modulated and semantic-aligned dynamic transformer for strip steel defect detection. *IEEE Transactions on Industrial Informatics*, 20(4), 6684-6695. <https://doi.org/10.1109/tii.2023.3347747>
- [24] Hamed, O., Bakkali, S., Blaschko, M., Moens, S., & Van Landeghem, J. (2024, August). Multimodal adaptive inference for document image classification with anytime early exiting. In *International Conference on Document Analysis and Recognition* (pp. 270-286). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-70546-5_16
- [25] Zhong, H., Fu, D., Xiao, L., Zhao, F., Liu, J., Hu, Y., & Wu, B. (2024). STFE-Net: A multi-stage approach to enhance statistical texture feature for defect detection on metal surfaces. *Advanced engineering informatics*, 61, 102437. <https://doi.org/10.1016/j.aei.2024.102437>
- [26] Xiang, Z., Jia, J., Zhou, K., Qian, M., & Wu, W. (2025). Block-wise feature fusion for high-precision industrial surface defect detection. *The Visual Computer*, 41(11), 9277-9295. <https://doi.org/10.1007/s00371-025-03927-4>
- [27] Cai, X., Wang, Y., & Zhang, L. (2022). Optimus: An operator fusion framework for deep neural networks. *ACM Transactions on Embedded Computing Systems*, 22(1), 1-26. <https://doi.org/10.1145/3520142>
- [28] Li, Y., Ren, Y., Wang, M., Zhang, Y., Zhang, X., & Deng, J. (2026). Deployment method of deep object detection models integrating model pruning and data reuse for reconfigurable architectures: Y. Li et al. *The Journal of Supercomputing*, 82(12), 606. <https://doi.org/10.1007/s11227-026-08749-2>
- [29] Yao, G., Zhu, S., Zhang, L., & Qi, M. (2024). HP-YOLOv8: high-precision small object detection algorithm for remote sensing images. *Sensors*, 24(15), 4858. <https://doi.org/10.3390/s24154858>
- [30] Wang, C., Zheng, B., & Li, C. (2025). Efficient traffic sign recognition using YOLO for intelligent transport systems. *Scientific reports*, 15(1), 13657. <https://doi.org/10.1038/s41598-025-98111-y>

APA Citation

Du, J., & Liu, M. (2026). Intelligent Defect Detection of Design Materials Based on Edge AI. *Journal of Digital Frontier*, 1(2), 27-57. <https://doi.org/10.67541/JDF2609>