J Syst Sci Complex (2025) 38(2): 573–596

A Low-Complexity and High-Accuracy Defect Detection Network*

ZHOU Xunkuai · CHEN Xi · CHEN Jie · CHEN Ben M.

DOI: 10.1007/s11424-025-4425-8
Received: 3 September 2024 / Revised: 23 October 2024

Abstract Visual-based defect detection efficiently monitors the health and quality of construction and industrial products. However, current defect detection methods often improve detection accuracy at the cost of lower inference speeds or more parameters, struggle with complex data representation, emphasize target features while neglecting environmental information importance, and utilize convolutional or max pooling operations for downsampling, leading to more feature loss. To address these issues, this work presents a low complexity, accurate defect detection network augmented by environmental information-assisted and flexible activation functions to enhance the neural network performance on complex data representation. Environmental information-assisted module is designed for defect detection tasks to assist in accurately locating and predicting defects. Moreover, this work restructure features post-downsampling to mitigate feature loss and design a simple feature module called deep-global fusion that integrates deep and global features to enhance detection performance. Extensive experiments validate the superiority of the proposed detection network. The deployment of the network on edge computing devices confirms its competitive advantage in portability and reliability.

Keywords Activation function, defects detection, environmental information-assisted, low complexity.

ZHOU Xunkuai
School of Electronics and Information Engineering, Tongji University, Shanghai 201804*, China; Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong* 999077*, China.* Email: 2010474@tongji.edu.cn; xunkuaizhou@cuhk.edu.
CHEN Xi (Corresponding author)
The Chinese University of Hong Kong, Hong Kong 999077*, China.* Email: xichen002@cuhk.edu.hk.
CHEN Jie
School of Electronics and Information Engineering, Tongji University, Shanghai 201804*, China; Harbin Institute of Technology, Harbin* 150001*, China.* Email: chenjie206@tongji.edu.cn.
CHEN Ben M.
The Chinese University of Hong Kong, Hong Kong 999077*, China.* Email: bmchen@cuhk.edu.hk.
*This study was supported in part by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government via the Hong Kong Centre for Logistics Robotics and in part by the Research Grants Council of Hong Kong SAR under Grant Nos. 14217922 and 14209623.
◇ *This paper was recommended for publication by Guest Editor XIE Lihua.*

1 Introduction

Defect detection in buildings and industrial products is crucial for safety evaluation and quality control. These structures often exhibit various defects, such as cracks, corrosion, and stains, whose progression can lead to significant losses. Traditional inspection methods, which largely depend on human visual assessments, face challenges in inspecting tall structures due to the need for high-altitude operations or specialized imaging equipment[1, 2]. Such methods can compromise safety and suffer from inaccuracies caused by human fatigue and equipment limitations[3]. Consequently, automated defect detection is vital for ensuring safety, maintaining production and environmental standards, and facilitating efficient operations and maintenance in large-scale construction and infrastructure management.

In recent years, significant advancements have been made in vision-based defect detection methodologies. Extensive research has utilized deep learning-based image processing techniques. Xie, et al.[4] introduced an industrial product defect detection dataset, facilitating algorithm design and performance evaluation. FFCNN[5] introduces deep neural networks for surface defect detection in magnetic materials, overcoming efficiency and cost limitations. Hu and Wang[6] incorporated an object-level attention module into their training strategy for casting defect detection method. However, detection speed may deteriorate when deployed on edge-computing devices. Gu, et al.[7] proposed an improved fault diagnosis scheme for sensors, which includes both fault detection and fault identification. Most existing methods primarily detect cracks, though defects in infrastructures and industrial products can manifest as crazing, spalling, and pitted, among others. Addressing this diversity of problems presents significant challenges in defect detection. The prevalent use of convolutional neural network layers for downsampling and feature extraction in many defect detection methods risks feature loss. These operations have limited potential to enhance detection performance.

To improve the efficiency, Yang, et al.[8] introduced a convolutional neural network (CNN) for defect detection that enhances efficiency by incorporating EIoU and modification loss functions into YOLOv3. However, the proposed method's inspection speed of 93.5 ms/image on the NVIDIA GTX1050Ti GPU renders it impractical for edge-computing devices like the Nvidia Orin NX. Similarly, YOLO-M[9] adapts YOLOv3 using an acceleration algorithm and a median flow (MF) algorithm for crack counting. Yet, it suffers from low processing speed and limited defect detection types, restricted to pavement cracks. Convolutional recurrent reconstructive network (CRRN)[10] enhances the performance of defect detection by integrating convolutional spatiotemporal memory (CSTM), and the effectiveness is validated across two public datasets. Despite achieving relatively reliable performance in defect detection, the current algorithms face the following challenges:

- Employing convolutional neural networks or max pooling operation for downsampling and feature extraction inevitably leads to feature loss. Enhancing feature retention during propagation is anticipated to improve detection performance.
- Predominantly focusing on target features often neglects the importance of environmental characteristics. Incorporating environmental features, as certain targets are inherently

linked to specific contexts, can enhance detection accuracy.

- Limited activation capabilities yield insufficient representation of complex defect data, contributing to sub-optimal detection accuracy.
- Enhancing detection accuracy frequently involves adding more parameters; however, balancing detection accuracy with memory efficiency is crucial for the practical application of memory-constrained devices.

Moreover, the high maneuverability of drones enables access to environments unreachable by humans. Drone-mounted defect detection systems can enhance inspection efficiency and mitigate accuracy degradation due to human fatigue. The advancement of highly accurate defect detection methodologies promises to enhance inspection efficacy, while accelerated detection rates contribute to overall inspection efficiency[11, 12].

Consequently, this work aims to develop a defect detection framework that balances parameters, speed, and accuracy. We introduce a novel detection network, Context-aware and Activation Representation Network (CARNet), which offers accurate and fast defect detection with minimal parameters and computational cost. To achieve this objective, 1) this work advocates using space-to-depth downsampling over convolutional layers to ensure complete feature propagation[13]. 2) We propose an environmental interaction module designed to enhance detection performance. 3) We propose the Kilu activation function, which offers flexible nonlinear representation capabilities through adjustable parameters, thereby improving detection performance. As depicted in Figure 1(a), the proposed CARNet achieves more accurate detection than YOLOv9[14]. These visualization results underscore the practical effectiveness of our method in defect detection. Figure 1(b) and Figure 1(c) confirm that our method achieves higher accuracy with fewer parameters and the fastest speed. Specifically, our method attains an accuracy of 52.3% with only 4.9 M parameters and delivers the fastest inference speed. This level of performance is highly competitive for memory-constrained applications.

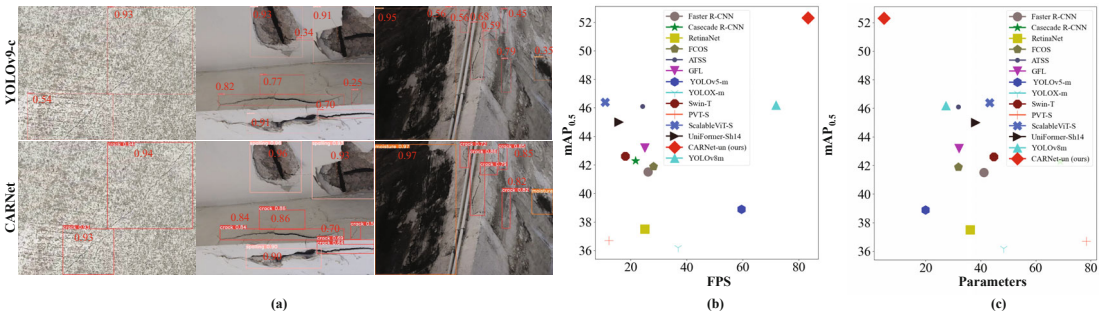

Figure 1 These pictures illustrate the superior performance of our method. (a) The first row illustrates the visualization results from YOLOv9, while the second row presents those from our method, which detect defects with higher confidence scores. (b) Trade-off performance of inference speed versus accuracy. (c) Trade-off performance of parameters versus accuracy. Please zoom in for the best view

In summary, the main contributions of this work are:

1) A defect detection network (CARNet) that balances parameters and accuracy is proposed. Introducing environmental interactive information in defect detection research marks an advancement development that enhances defect localization and assessment, thereby improving detection performance.

2) A novel, adaptable activation function is proposed to augment the nonlinear representation capabilities of neural networks, thereby enhancing detection accuracy without an increase in parameters. The activation performance surpasses the other 20 activation functions by comparative experiments.

3) Comprehensive ablation studies and feature map analysis illustrate the effectiveness of the proposed strategies. The efficacy of CARNet is validated across three challenging datasets. Deploying CARNet on an edge computing device with 1920×1080 resolution videos confirms its real-time detection capabilities in UAV onboard applications.

The rest of this paper is structured as follows: Section 2 discusses related works, Section 3 explains the theoretical basis and establishment process of the model, Section 4 presents the experimental results and performance evaluation, and Section 5 concludes the article and outlines future work.

2 Related Works

2.1 Feature Fusion

Researchers have proposed various techniques using feature fusion to enhance the detection accuracy. Liu, et al.[15] introduced an adaptive parallel feature learning and hybrid feature fusion-based deep learning approach for machining condition monitoring, which incorporates handcrafted features enhanced by domain knowledge. Hu, et al.[16] introduced a hybrid multi-dimensional feature fusion structure for thermography defect detection, which enhances both accuracy and robustness. Gao, et al.[17] proposed an enhanced detection network for small insulator defects that incorporates a batch normalization convolutional block attention module (BN-CBAM) and a feature fusion module to improve detection accuracy. Li, et al.[18] introduced a bidirectional fusion network (BiFNet) that integrates the image and BEV of the point cloud through the dense space transformation (DST) module and the context-based feature fusion module for road detection. However, most methods focus primarily on extracting features related to the detected objects during feature fusion, often overlooking the significance of environmental features. For instance, employing attention mechanisms to focus on target features.

2.2 Activation Function

Activation functions are pivotal in transforming input data into an abstract feature space, enhancing the nonlinear representation capabilities of deep neural networks (DNNs) to process

intricate data sets[19]. In the initial phases of neural network research, traditional activation functions such as Sigmoid and Tanh are widely used, despite their tendency to cause vanishing gradients. The Rectified Linear Unit (ReLU)[20] is introduced to overcome this issue, effectively addressing the vanishing gradient problem by setting negative inputs to zero and maintaining positive values. However, ReLU still faces challenges with dying gradients for non-positive inputs. Gaussian Error Linear Unit (GELU) can address this limitation using a probabilistic approach for input processing[21]. However, we have observed a performance degeneration when applying GELU to defect detection tasks. For a more detailed analysis, see Subsection 4.3.

To tackle the identified challenges, this study introduces a convolutional space-to-depth approach for feature extraction and downsampling that maintains full convolutional feature propagation. Context-aware information enhances defect localization and identification. Improved detection results are obtained by merging depth and global features to clarify feature representation. A novel activation function is proposed to strengthen the network's capability to represent complex data.

3 Methodology

3.1 Framework Overview

As depicted in Figure 2, The CARNet for accurate defect detection consists of two main components: The encoder and the decoder. The encoder features a structured sequence of five Convolutional Spatial-to-Depth (CSD) modules, illustrated by the orange block, and four Combined Ambient Residual modules (CRM), depicted by the light green block. The Space-to-Depth approach facilitates downsampling while ensuring the complete propagation of convolutional features, and CRM integrates contextual interaction capabilities. The encoder sequence culminates with the Depth Global Module (DGM), which orchestrates feature extraction and integration, capturing depth and global feature dynamics.

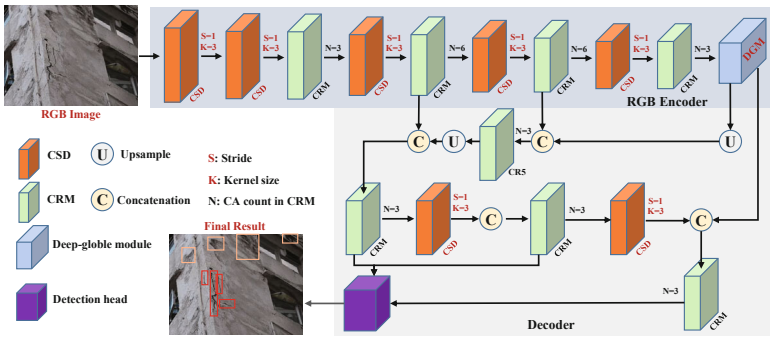

Figure 2 The defects detection framework begins with four rounds of downsampling applied to the original input image. Next, the deep convolutional features are fused with global features using the DGM. A top-down branch is followed by a bottom-up branch to perform feature fusion. Finally, the fused features are passed to the detection head to generate the detection results. Please zoom in for the best view

The decoder is split into two pathways: The bottom-up and the top-down branches. The bottom-up branch consists of two upsampling modules and one CRM module, with upsampling modules increasing the feature scale and the CRM module refining convolutional residual features. In contrast, the top-down branch employs two CRM modules and two CSD modules, focusing on downsampling and extracting convolutional features, where each CSD operation reduces the feature map dimensions by half. This branch selectively integrates features solely from high-level CRM modules during extraction to conserve computational resources. Enhancements in the bottom-up path integrate precise localization cues into lower feature layers, thereby strengthening the hierarchical structure and minimizing the information propagation distance from lower levels to the topmost features. The variable "N" indicates the number of CA network in each CRM module, and the CSD module includes convolutional, normalization, Kilu activation function layers, and space-to-depth operation.

3.2 CSD Module

As illustrated in Figure 3, the CSD module initially obtains convolutional features through a convolutional layer, maintaining the same feature dimensions for both input and output. Subsequently, spatial features are converted into depth features through pixel reorganization manner. Through the CSD operation, the channels of the output features become four times that of the input, while the dimensions of the features are halved. However, this operation preserves more of the features. Let $\boldsymbol{F} \in \mathbb{R}^{C_1 \times H \times W}$ represent the input; the operation of the CSD can be described as follows:

$$\boldsymbol{F_1} = Conv_{3\times3}(\boldsymbol{F}), \quad \boldsymbol{F_2} = Spacedepth(\boldsymbol{F_1}), \tag{1}$$

where $\boldsymbol{F_1} \in \mathbb{R}^{C_1 \times H \times W}$ is the output of convolution layer. $\boldsymbol{F}_2 \in \mathbb{R}^{4C_1 \times H/2 \times W/2}$ represents the final output after the operations of the space-to-depth.

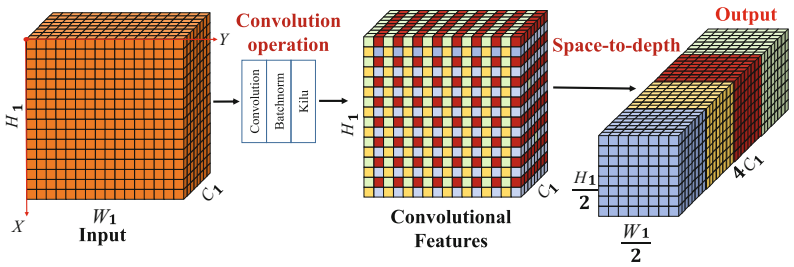

Figure 3 Illustration of CSD. Following the convolution operation and subsequent downsampling, the output dimensions are reduced to half of the input dimensions, while the number of channels increases to four times that of the input

The space-to-depth operation can be formulated in two dimension scenarios as follows:

$$\begin{aligned} f_{0,0} &= \boldsymbol{F_1}[0:W:2, 0:H:2], \quad f_{0,1} = \boldsymbol{F_1}[0:W:2, 1:H:2], \\ f_{1,0} &= \boldsymbol{F_1}[1:W:2, 0:H:2], \quad f_{1,1} = \boldsymbol{F_1}[1:W:2, 1:H:2]. \end{aligned} \tag{2}$$

Generally, for a given feature map $\boldsymbol{F_1}$, a sub-map $f_{x,y}$ comprises all entries $\boldsymbol{F_1}(i,j)$ for which $i+x$ and $j+y$ are given by half, effectively downsampling $\boldsymbol{F_1}$ by this scale.

Compared to employing pooling layers or convolutional layers for downsampling, the CSD operation preserves more feature information, which is beneficial for loss function calculation and overall network weight optimization.

3.3 Proposed CRM Module

Neural networks typically extract features from objects for detection tasks, yet certain objects are inherently associated with specific environments; for instance, penguins are indigenous to Antarctica. While challenging, the differentiation between turtles and tortoises can be facilitated by considering their habitats—sea turtles reside in aquatic environments, whereas tortoises are terrestrial.

The types of building defects considered in this study include cracks, dampness, and spalling. For the first two types of defects, this study suggests that their causes are closely related to environmental conditions. For example, dampness typically occurs in environments with high moisture levels, while cracks are often caused by environmental vibrations. Therefore, if moisture is detected in the surrounding environment of a target, the detection network will be more confident in identifying the target as dampness. Similarly, vibrations can lead to a higher incidence of cracks, increasing the likelihood of finding another crack near an existing one. Inspired by this observation, this study posits that incorporating environmental information could enhance detection accuracy.

To capitalize on this, a module is proposed to leverage environmental context. As depicted in Figure 4, features $\boldsymbol{F} \in \mathbb{R}^{C\times H\times W}$ are first extracted via a convolutional layer and then divided into two halves, $\boldsymbol{F_s1}, \boldsymbol{F_s2} \in \mathbb{R}^{C/2\times H\times W}$. These are input into the Contextual Interaction Module (CA), where feature enhancement occurs through an additional convolutional layer. Contextual environmental features are extracted using a dilated convolution (i.e., DCBK operation) with a dilation rate of 4. The features are subsequently fused, and a residual connection integrates $\boldsymbol{F_s}$ with the enhanced features to produce $\boldsymbol{F_{a1}} \in \mathbb{R}^{C/2\times H\times W}$. This output is combined with $\boldsymbol{F_s}$ and further processed by another convolutional layer to generate the final feature set. Multiple CA operations, if applied, involve a fusion step before the convolutional output. Additionally, the CA module in the decoder does not include any addition operations. As detailed in the operations of the Combined Ambient Residual module (CRM).

$$\begin{aligned} &\boldsymbol{F} = Conv_{1\times1}(\boldsymbol{F_{input}}), \quad \boldsymbol{F_{s1}}, \boldsymbol{F_{s2}} = split(\boldsymbol{F}), \quad \boldsymbol{F_{a1}} = CA(\boldsymbol{F_{s1}}), \\ &\boldsymbol{F_{a2}} = CA(\boldsymbol{F_{a1}}), \quad \boldsymbol{F_{f2}} = Concate(\boldsymbol{F_{a1}}, \boldsymbol{F_{a2}}, \boldsymbol{F_{s2}}), \quad \boldsymbol{F_{out}} = Conv_{1\times1}(\boldsymbol{F_{f2}}), \end{aligned} \tag{3}$$

where $\boldsymbol{F_{a2}} \in \mathbb{R}^{C/2\times H\times W}$ is the output feature map from another CA module. $\boldsymbol{F_{f2}} \in \mathbb{R}^{3C/2\times H\times W}$ denotes the concatenation feature map. $\boldsymbol{F_{out}} \in \mathbb{R}^{C\times H\times W}$ denotes the final output feature map.

The operations of the Contextual Interaction (CA) module proceed as follows:

$$\begin{aligned} &\boldsymbol{F_{c1}} = Conv_{3\times3}(\boldsymbol{F_s}), \quad \boldsymbol{F_{c2}} = Conv_{3\times3}(\boldsymbol{F_{c1}}), \quad \boldsymbol{F_{d1}} = DConv_{3\times3}(\boldsymbol{F_{c1}}), \\ &\boldsymbol{F_{f1}} = Concate(\boldsymbol{F_{c2}}, \boldsymbol{F_{d1}}), \quad \boldsymbol{F_{a1}} = \boldsymbol{F_{f1}} + \boldsymbol{F_{c1}}, \end{aligned} \tag{4}$$

where $\boldsymbol{F_{c2}} \in \mathbb{R}^{C/2\times H\times W}$ denotes the convolutional feature map. $\boldsymbol{F_{d1}} \in \mathbb{R}^{C/2\times H\times W}$ denotes the feature map from dilation convolutional layer. $\boldsymbol{F_{c2}} \in \mathbb{R}^{C/4\times H\times W}$ is feature map convoluted to $\boldsymbol{F_{c1}}$.

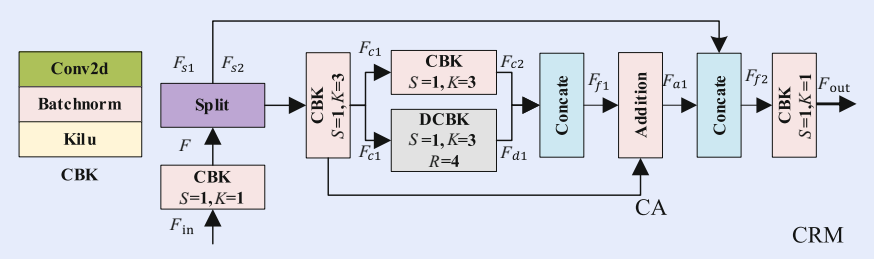

Figure 4 Illustration of CRM. The CA module is capable of multiple serial connections. Here, only a scenario with two serial connections is demonstrated

3.4 DGM Module

Figure 5 depicts the Deep Global Module (DGM) architecture. The input feature $\boldsymbol{F} \in \mathbb{R}^{C_1\times H\times W}$ undergoes an initial convolutional layer, generating the feature $\boldsymbol{F_1} \in \mathbb{R}^{C_0\times H\times W}$. Subsequently, $\boldsymbol{F_1}$ is enhanced through three successive 7×7 global pooling layers, each refining the features further. Concurrently, $\boldsymbol{F_1}$ passes through a separable convolution (indicated by the green block), resulting in a nuanced feature set $\boldsymbol{F_2} \in \mathbb{R}^{C_0\times H\times W}$. This feature set $\boldsymbol{F_2}$ is then merged with the globally pooled features to form a composite feature matrix, which is processed by another convolutional layer to produce the final output feature. This configuration leverages separable convolutions to delve deeper into the feature space efficiently, and the 7×7 pooling size enhances global contextual capture by extending the receptive field.

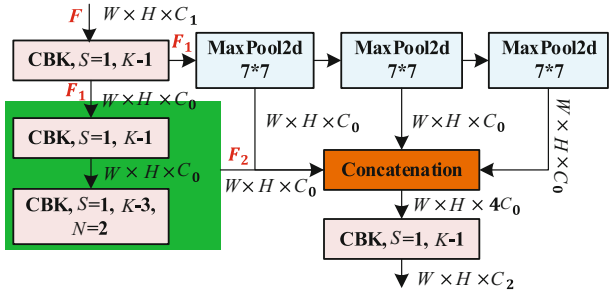

Figure 5 The structure of the proposed DGM. The three-channel global features are fused with the deep convolutional features

3.5 Proposed Kilu Activation Function

Activation functions are nonlinear point-wise functions that introduce nonlinearity into the linearly transformed input within a deep neural networks layer (DNN). The selection of the activation function is crucial for gauging the network's performance. Mathematically, the application of an activation function in a neural network layer is expressed as $z = \phi(y) = \phi(\sum_i w_i x_i + b)$, where z is the output of the activation function $\phi(y)$.

Searching for an effective and robust activation function in DNNs poses considerable challenges, primarily due to the saturation characteristics of conventional functions. Saturation refers to the tendency of the derivative of an activation function, $\delta(x)$, to approach zero in both positive and negative domains, resulting in vanishing gradients. Classic activation functions such as *Sigmoid* and *Tanh* are particularly susceptible to this phenomenon, often leading to diminished gradient propagation during training, especially when the inputs are excessively large or small. The introduction of the Rectified Linear Unit (ReLU), defined as $\delta(x) = \max(0, x)$, marked a significant advancement in activation functions, facilitating more efficient training dynamics. However, ReLU is not without its limitations, notably its susceptibility to the "dying neuron" problem, where neurons become inactive and only output zero if the inputs are negative, thereby impeding the gradient flow across these neurons.

Therefore, we propose a flexible activation function called *Kilu*. As shown in Figure 6(a) and Figure 6(b), like SiLU utilized in YOLOv9[14], the *Kilu* shares the similar unbounded upper limits property on the right side of activation curve. The proposed activation function *Kilu* is computed by multiplying the logarithm of the exponential function of *Tanh* with its input x and defined as:

$$\delta(x) = x \log(1 + \mathrm{e}^{\tanh(\alpha x)}), \tag{5}$$

where α is the scaling parameter.

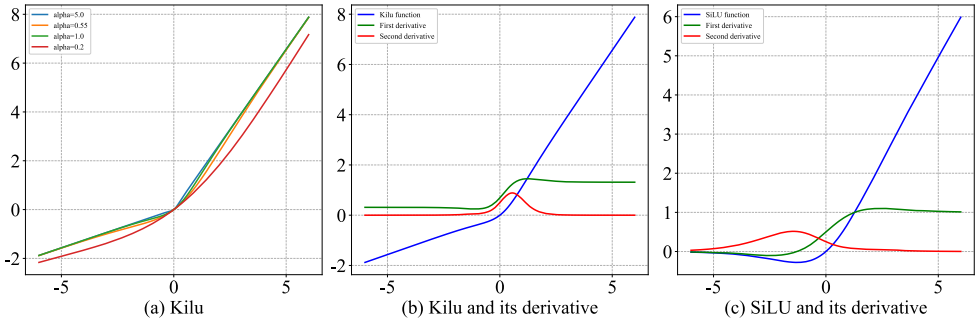

Figure 6 The graph of the function. (a) Kilu function's graph for different values of α. (b) The first derivative plot, and the second derivative plot of the Kilu function. (c) The first derivative plot, and the second derivative plot of the SiLU function. Please zoom in for the best view

The first-row in Figure 6 depicts the graph of *Kilu* function for different values of α. It is observable that the value of α affects the amplitude of the circular arc in the middle. As α increases, the amplitude of the circular arc diminishes.

For substantial positive inputs, the Kilu function exhibits characteristics akin to SiLU, with the output approximating a linear relationship to the input. Distinctively, the Kilu function maintains a linear response even for negative inputs, unlike SiLU and other prevalent activation functions. The first-order derivative of Kilu, denoted $\delta(x)'$, is formulated as follows:

$$\delta(x)' = \log(1 + \mathrm{e}^{\tanh(x)}) + x\left(\frac{\mathrm{e}^{\tanh(x)(1-\tanh^2(x))}}{1 + \mathrm{e}^{\tanh(x)}}\right). \tag{6}$$

Similarly, the 2^{nd} order derivative of Kilu with α=1 is given as follows:

$$\begin{aligned}\delta(x)'' =& \frac{2\mathrm{e}^{\tanh(x)} * (1-\tanh^2(x))}{1+\mathrm{e}^{\tanh(x)}} + \frac{x\mathrm{e}^{2\tanh(x)}(1-\tanh^2(x))^2}{(1+\mathrm{e}^{\tanh(x)})^2} \\ &+ \frac{x\mathrm{e}^{\tanh(x)}(1-\tanh^2(x))^2 - 2x\tanh(x)\mathrm{e}^{\tanh(x)}(1-\tanh^2(x))}{1+\mathrm{e}^{\tanh(x)}}.\end{aligned} \tag{7}$$

The second and third columns of Figure 6 depict the graphs of the first and second derivatives of the Kilu and SiLU functions, respectively. Analyzing the first derivatives, it is evident that the gradients of our activation functions do not approach zero as they extend towards negative or positive infinity. Furthermore, they demonstrate consistent gradient values, which suggests that our activation functions contribute to enhanced stability during network training. Moreover, they facilitate activation across a broader spectrum of values. Notably, the second-order derivative of the proposed Kilu function resembles the negative Laplacian operator, similar to the second-order derivative of the Gaussian operator. This resemblance is advantageous for function maximization.

3.6 Loss Function

Our framework employs two distinct loss functions: (i) Classification loss, which measures the correctness of the class assigned to the detected object; and (ii) Regression loss, which assesses the accuracy of the bounding box coordinates about the actual location of the object, as detailed in [22].

3.6.1 Classification Loss

As detailed in Equation (8), the class-specific loss is calculated utilizing the cross-entropy method:

$$L_{cls} = -\sum_{i}\sum_{c=1}^{M} g_{ic}\log(p_{ic}), \tag{8}$$

where g and p denote the ground truth and predicted values, respectively. M is the number of categories. i denotes the i-th sample.

3.6.2 Regression Loss

Regression loss is consist of two parts: L_{DFL} and L_{iou}, L_{DFL} can enhance the generality capability and is formulated as:

$$L_{DFL} = -((y_{i+1}-y)\log(s_i) + (y-y_i)\log(s_{i+1}), \quad s_i = \frac{y_{i+1}-y}{y_{i+1}-y_i}, \quad s_{i+1} = \frac{y-y_i}{y_{i+1}-y_i}, \tag{9}$$

where the y_i and y_{i+1} are possible float vector. y is the ground-truth class.

L_{iou} has two formations: L_{ciou} and $L_{shape-iou}$[23]. The L_{ciou} formulations are defined:

$$\begin{aligned}&L_{ciou} = 1 - IOU + \frac{Dis^2(b,\widehat{b})}{c^2} + \rho k, \quad IOU = \frac{|B\cap\widehat{B}|}{|B\cup\widehat{B}|}, \quad \rho = \frac{k}{(1-IOU)+k}, \\ &k = \frac{4}{\pi^2}\left(\arctan\frac{\widehat{\omega}}{\widehat{h}} - \arctan\frac{w}{h}\right)^2.\end{aligned} \tag{10}$$

In the specified formulation, $\widehat{B} = (\widehat{x}, \widehat{y}, \widehat{\omega}, \widehat{h})$ denotes the coordinates of the ground-truth bounding box, while $B = (x, y, \omega, h)$ corresponds to the predicted bounding box. The variables b and $\widehat{b}$ represent the central points of B and $\widehat{B}$, respectively. The function $Dis(\cdot)$ is defined as the Euclidean distance between these central points, and c represents the diagonal length of the smallest enclosing box that covers both B and $\widehat{B}$.

The $L_{shape-iou}$ is formulated as:

$$\begin{aligned} &L_{shape-iou} = 1 - IOU + distance^{shape} + 0.5\omega^{shape}, \quad ww = \frac{2(\widehat{w})^{scale}}{(\widehat{w})^{scale} + (\widehat{h})^{scale}}, \\ &hh = \frac{2(\widehat{h}))^{scale}}{(\widehat{w})^{scale} + (\widehat{h})^{scale}}, \quad distance^{shape} = \frac{hh(x_c - \widehat{h_c})^2 + ww(y_c - \widehat{y_c})}{c^2}, \\ &\omega^{shape} = \sum_{t=w,h} (1 - \mathrm{e}^{-wt})^{\theta}, \quad \theta = 4, \quad w_w = \frac{hh|w - \widehat{w}|}{max(w, \widehat{w})}, \quad w_h = \frac{ww|h - \widehat{h}|}{max(h, \widehat{h})}. \end{aligned} \tag{11}$$

We explore the influence of L_{ciou} and $L_{shape-iou}$ on a network in Subsection 4.3. The total loss is expressed as follows:

$$Loss_{total} = 0.5L_{cls} + 1.5L_{DFL} + 7.5L_{iou}. \tag{12}$$

4 Experiment

This section assesses the generalizability of the constructed defect detection framework across three distinct datasets. We also experimentally validate the effectiveness of the proposed module and activation function. Additionally, we evaluate the feasibility of deploying the constructed framework on edge-computing devices.

4.1 Experimental Setup

4.1.1 Implementation Details

To expedite our training experiments, we employ two NVIDIA A40 GPUs, one of NVIDIA RTX 3090 and NVIDIA GTX 4090 for independent training runs. All evaluations are conducted exclusively on the NVIDIA GTX 3090 during the testing phase. The initial and final learning rates are 0.01. The optimizer uses stochastic gradient descent (SGD) with a momentum parameter of 0.937 and weight decay of 0.0005. The value of epochs is 500.

4.1.2 Datasets

To illustrate the generalization performance of CARNet, we benchmark the constructed CARNet on three challenging datasets: A self-collected dataset, SSGD[24] and NEU-Det[25]. SSGD is an open-source smartphone screen glass dataset that includes seven basic common types of defects occurring on glass panels, while NEU-Det is a steel surface defect dataset encompassing six types of defects. SSGD has an input resolution of 1500×1500, which helps verify the computational efficiency of our method under high-resolution input conditions, while its diverse detection targets contribute to validating the detection effectiveness of our approach. NEU-Det is a classical dataset containing various types of defects, and due to its lower resolution

and smaller defect sizes, defect features are difficult to extract. Using this dataset helps evaluate the robustness of our method in detecting small defects and extracting their features.

We remove the images with resolution of 8000×6000 in CUIBIT[26] and capture additional images using smartphone. As depicted in Figure 7, optical images are captured at a resolution of 4624 × 3472 using both cameras and smartphones. This extensive defect dataset includes 5527 high-resolution images collected from various infrastructures such as pavements, roads, buildings, and bridges, cataloging defects like cracks, spalling, and moisture. About 20% of the images constitute the test set, with the remaining 80% split between training (72%) and validation (8%). For training and testing, the input resolution is adjusted to 1024 × 1024.

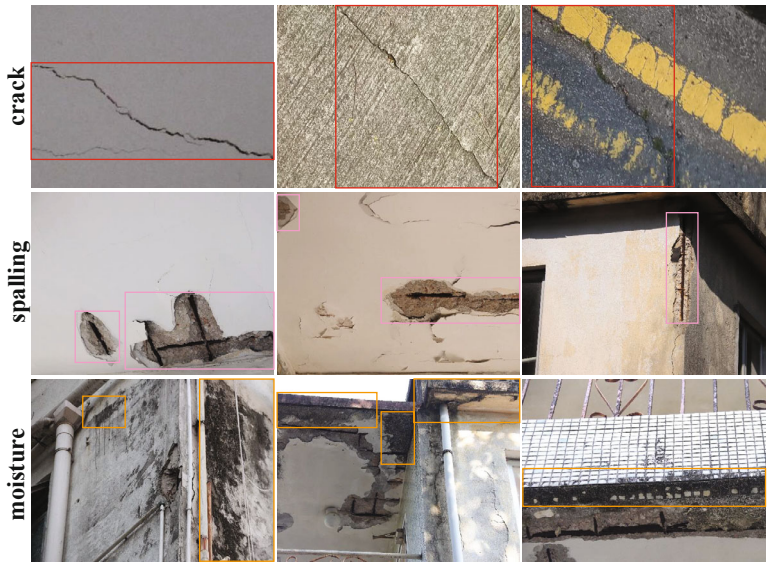

Figure 7 The example images of self-collect datasets. Please zoom in for the best view

Figure 8 presents the distribution of dataset labels and the correlation among the target classes. Figure 8(a) respectively represent the following: The number of instances for each category in the dataset, the distribution of object bounding box sizes, the distribution of bounding box center positions relative to the entire image, and the ratio of bounding box width and height to the overall image dimensions. Figure 8(b) illustrates the modeling of label correlations by the object detection algorithm during the training process. Each matrix cell represents a label used by the model during training, and the shading of the cells reflects the degree of correlation between the corresponding labels. Darker cells indicate that the model has learned a stronger association between the two labels, while lighter cells suggest a weaker correlation. The color along the diagonal represents the self-correlation of each label, which is typically the darkest, as the model finds it easier to learn the relationship of a label with itself. The dataset poses a challenge due to the considerable variation in the physical characteristics of the targets, including size and shape. Figure 8 underscores the imbalanced and biased nature of the dataset, as indicated by the significant disparities between the defect classes.

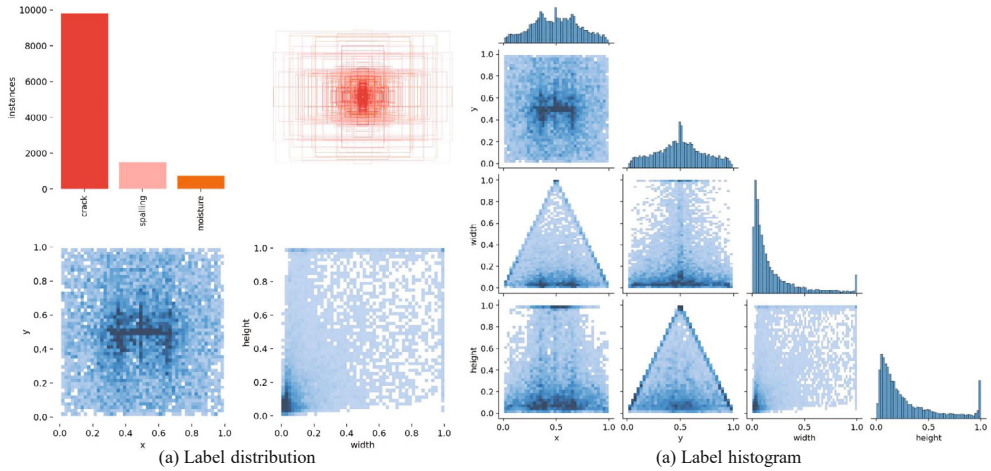

Figure 8 Correlogram of the target classes with their (a) corresponding label distribution and (b) the size and location of the labels in the dataset

4.2 Comparison with Prior Works

Self-collected dataset. From Table 1, our CARNet performs best in $mAP_{0.5}$ and $mAP_{0.5:0.95}$ with fewer parameters. Specifically, our CARNet-n surpasses YOLOv10-x in accuracy by 3%, while achieving nearly 86% reduction in parameters and 93% reduction in computational costs. Additionally, with a model size that is not much less than YOLOv8-l (83.7 vs. 87.7), our CARNet-m exhibits a 2% higher accuracy ($mAP_{0.5:0.95}$) than YOLOv8-l (57.9 vs. 56.9). Our CARNet-n shows an accuracy similar to YOLOv8-m at $mAP_{0.5}$ (81.3 vs. 81.4), whereas surpasses YOLOv8-m by nearly 1% at $mAP_{0.5:0.95}$ (56.3 vs. 55.8), while also reducing the model size by 82% (9.3 vs. 52.1). Similarly, our model achieves higher accuracy with significantly lower computational overhead. Even our most computationally expensive CARNet-m is 56% less than the best-performing YOLOv8-x in computational cost (289.4 vs. 660.8). The visual comparison in Figure 9 demonstrates that our method has a low miss detection rate and high confidence, highlighting the practical utility of the proposed approach. Figure 10 demonstrates that our approach offers a better balance between computational cost and accuracy, as well as between model size (parameters) and accuracy.

SSGD. From Table 2, we can clearly observe that our CARNet achieves the best performance on all five evaluation metrics, where our CARNet-n obtains 4.9, 42.2, 83.3, 25.2, and 51.1 on parameters, FLOPs, FPS, $mAP_{0.5:0.95}$, and $mAP_{0.5}$, respectively. The best performance on the five metrics fully illustrates the advantages of the proposed network.

The proposed CARNet-n only has 4.9 M parameters and runs at 83.3 FPS for the 1500 image resolution input. Our method is 7.6×, 6.8×, 5.3× and 4.6× faster than ScalableViT-S, PVT-S, UniFormer-Sh14$_{h14}$, and Swin-T, respectively. CARNet achieves its best detection performance with lower computational overhead without either sacrificing inference speed or increasing parameters. Figure 1(b) and Figure 1(c) also demonstrate that our method does not increase accuracy by adding parameters or sacrificing speed to enhance accuracy. Instead,

it maintains an excellent balance between the number of parameters and accuracy, as well as between speed and accuracy. The competitive performance fully illustrates the advantages of the proposed network for UAV applications.

Table 1 Quantive benchmarking results on self-collected datasets

Model	mAP_{all} (%)↑		mAP_{crack} (%)↑		$mAP_{spalling}$ (%)↑		$mAP_{moisture}$ (%)↑		Model size (M)↓	FLOPs (B)↓
-	$mAP_{0.5}$	$mAP_{0.5:0.95}$	$mAP_{0.5}$	$mAP_{0.5:0.95}$	$mAP_{0.5}$	$mAP_{0.5:0.95}$	$mAP_{0.5}$	$mAP_{0.5:0.95}$	-	-
YOLOv6-n[27]	76.5	49.8	77.8	48.4	87.3	60.8	57.6	31.7	10.0	29.0
YOLOv6-s[27]	78.5	52.7	80.8	51.3	88.9	63.7	66.0	43.0	38.7	115.6
YOLOv6-m[27]	79.8	54.4	81.3	53.0	91.1	67.6	66.8	42.5	72.5	210.4
YOLOv6-l[27]	82.0	54.9	82.1	53.5	92.3	66.0	71.5	45.2	111.6	368.6
YOLOv7-t[28]	72.6	42.9	73.8	40.0	86.4	56.9	57.6	31.7	12.3	13.0
YOLOv7[28]	77.2	49.7	80.8	49.0	87.6	59.8	63.0	40.3	74.9	264.3
YOLOv8-n[22]	79.5	53.0	81.5	51.9	90.9	63.5	65.9	43.6	6.4	21.0
YOLOv8-s[22]	80.8	55.2	82.5	53.1	88.3	65.3	71.7	47.1	22.6	73.3
YOLOv8-m[22]	81.4	55.8	83.3	54.5	90.4	67.0	70.5	46.0	52.1	202.4
YOLOv8-l[22]	81.8	56.9	84.4	55.7	91.8	68.9	69.2	45.9	87.7	423.4
YOLOv8-x[22]	82.4	57.5	82.7	55.7	92.8	69.9	71.5	46.8	136.8	660.8
RT-DETR-l[29]	78.4	48.9	77.8	45.5	85.7	60.7	71.8	40.6	66.1	-
RT-DETR-x[29]	79.2	49.9	79.2	46.5	86.0	60.6	72.4	42.7	135.4	-
YOLOv9-t[14]	77.4	52.4	77.9	50.6	88.2	64.7	66.2	41.9	6.2	17.2
YOLOv9-c[14]	82.4	57.0	84.8	55.9	92.2	68.9	70.3	46.2	98.3	263.2
YOLOv10-n[30]	77.8	50.6	80.3	49.9	87.3	62.1	65.8	39.8	5.9	17.2
YOLOv10-s[30]	79.1	52.4	81.7	52.9	90.1	63.6	65.7	40.5	16.6	55.3
YOLOv10-m[30]	79.5	53.3	81.4	52.9	90.6	64.8	66.5	42.3	33.6	151.3
YOLOv10-b[30]	81.1	54.4	83.4	54.0	90.5	66.0	69.6	43.2	41.6	235.5
YOLOv10-x[30]	81.7	54.6	83.3	54.4	90.0	64.2	71.6	45.1	64.2	410.6
CARNet-un	80.9	55.4	83.9	54.8	90.9	67.0	67.9	44.4	9.3	29.2
CARNet-us	81.8	56.6	84.2	55.9	91.1	67.7	70.2	46.1	34.7	108.2
CARNet-n	81.3	56.3	83.8	54.9	90.4	66.4	69.6	47.6	9.3	29.2
CARNet-s	81.6	57.2	83.5	56.2	91.8	69.3	69.6	46.2	34.7	108.2
CARNet-m	82.8	57.9	84.6	57.6	91.7	68.9	72.1	47.3	83.7	289.4

Note: CARNet-un indicates that the regression loss function used is shape-IOU. ↑ (↓) indicates that larger (smaller) values lead to better (worse) performance.

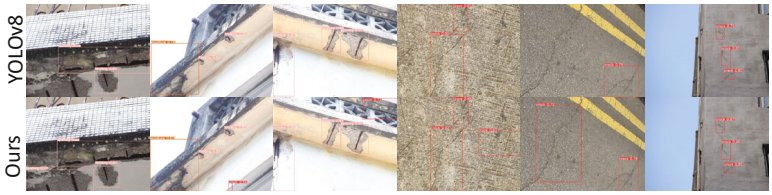

Figure 9 Qualitative visualization. Our method achieves high detection confidence with a low miss-detection rate. Please zoom in for the best view

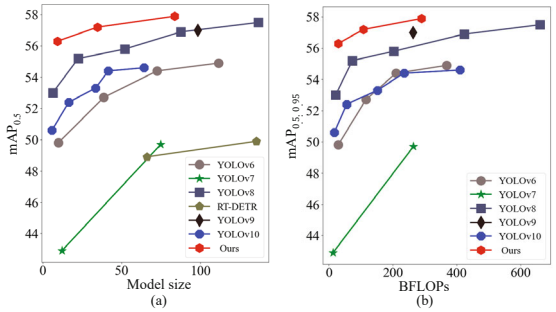

Figure 10 (a) Trade-off performance of model size versus accuracy. (b) Trade-off performance of BFLOPs versus accuracy

Table 2 Quantive benchmarking results on SSGD-part1

Method	Parameters (M)↓	FLOPs (B)↓	FPS↑	$\mathbf{mAP}_{0.5:0.95}$ (%)↑	$\mathbf{mAP}_{0.5}$ (%)↑
Faster R-CNN[31]	41.2	303.8	26.2	19.3	41.5
Casecade R-CNN[32]	68.9	331.6	21.7	20.9	42.3
RetinaNet[33]	36.2	311.2	25.0	16.4	37.5
FCOS[34]	31.9	296.2	28.1	19.4	41.9
ATSS[35]	31.9	303.3	24.2	22.3	46.1
GFL[36]	32.1	307.9	25.0	19.6	43.2
YOLOv5-m[37]	19.9	266.7	59.5	16.2	38.9
YOLOX-m[38]	48.3	405.1	36.9	13.4	36.2
Swin-T[39]	44.8	308.2	18.1	19.2	42.6
PVT-S[40]	78.4	281.3	12.3	16.0	36.7
ScalableViT-S[41]	43.3	297.7	10.9	21.2	46.4
UniFormer-Sh14$_{h14}$[42]	38.2	276.4	15.8	18.9	45.0
YOLOv8-m[22]	27.4	432.3	71.8	21.7	46.2
CARNet-un	4.9	42.2	83.3	24.2	52.3
CARNet-n	4.9	42.2	83.3	25.2	51.1

Note: ↑ (↓) indicates that larger (smaller) values lead to better (worse) performance.

NEU-Det. We opt to compare our approach with the latest method, BDDN[43], and as shown in Table 3, our method surpasses BDDN in $\mathrm{mAP}_{0.5}$. While our method underperforms BDDN in the categories of Crazing and Rolled, it significantly outperforms this method in the remaining three categories regarding detection accuracy. Notably, our method exhibits a 20% higher accuracy on Pitted than BDDN, demonstrating the suitability of our method for detecting industrial defects.

Table 3 Quantive benchmarking results on NEU-Det

Method	Backbone	$\mathbf{mAP}_{0.5}$	Crazing	Inclusion	Rolled	Scratches	Pathes	Pitted
SSD512[44]	VGG16	72.1	39.9	79.6	61.9	84.4	86.7	79.8
RetinaNet[33]	Darknet53	68.0	43.7	76.2	58.1	76.0	74.3	79.6
BDDN[43]	DRN+DA+FPN	76.2	48.3	82.4	74.9	90.4	89.3	71.7
CARNet-um	-	**76.9**	**44.0**	**82.9**	**65.9**	**95.1**	**91.9**	**81.6**
CARNet-m	-	**76.4**	**42.9**	**83.6**	**60.5**	**96.0**	**90.2**	**85.3**

Note: The bold values represent the experimental results from the proposed method.

4.3 Ablation Study

Table 4 presents the results of the ablation study on the self-collected dataset. The advanced method YOLOv8 is chosen as baseline. By conducting ablation experiments on the baseline through the integration of various modules and methods, we aim to validate the contribution and performance improvement of each component relative to the baseline. Here, mAP_1 and mAP_2 respectively represent $\mathrm{mAP}_{0.5:0.95}$ and $\mathrm{mAP}_{0.5}$.

Table 4 Effects of various components on performance

Components	Ablation study									
space-depth				✓	✓	✓	✓	✓	✓	✓
Shape-IOU			✓		✓	✓				✓
DGM						✓	✓	✓	✓	✓
CRM								✓	✓	✓
Kilu									✓	✓
Gelu		✓								
mAP_1/ mAP_2	53.0/79.5	52.5/79.9	53.0/78.7	53.9/80.5	54.1/79.6	54.2/81.4	54.8/81.5	55.5/81.2	56.3/81.3	55.4/80.9
Improvement	-	-0.5/+0.4	NI/-0.8	+0.9/+1.0	+1.1/+0.1	+1.2/+1.9	+1.8/+2.0	+2.5/+1.7	+3.3/+1.8	+2.4/+1.4

Note: The ✓denotes that the module is integrated into this ablation study. NI denotes "no improvement".

4.3.1 Baseline+Shape-IOU

In the baseline configuration using CIoU, substitution with Shape-IoU yields $mAP_{0.5:0.95}$ and $mAP_{0.5}$ of 53.0 and 78.7, respectively, with $mAP_{0.5}$ experiencing a decline of 0.8. Analysis of Tables 1 and 3 reveals Shape-IoU's sensitivity to bounding box shapes, affecting dimensions and leading to performance reductions in $mAP_{0.5:0.95}$ with larger inputs. Conversely, smaller inputs, as demonstrated in Table 2, improve outcomes, highlighting the necessity for rigorous validation of Shape-IoU across different detection scenarios, with Table 1 indicating superior performance of CARNet-us in $mAP_{0.5}$. In summary, while Shape-IoU may impact localization performance, it positively affects detection efficacy in $mAP_{0.5}$; however, its effectiveness requires experimental validation.

4.3.2 Baseline+Space-to-Depth

After integrating the baseline with space-to-depth, the performance $mAP_{0.5:0.95}$ and $mAP_{0.5}$ respectively increased by 0.9 and 1.0. This enhancement is attributed to the space-to-depth operation effectively preserving the complete transfer of features.

4.3.3 Baseline+GeLU

The baseline accuracies are 53.0 and 79.5, respectively, compared to 52.5 and 79.9 after incorporating the GeLU function, with $mAP_{0.5:0.95}$ decreasing by 0.5 and $mAP_{0.5}$ increasing by 0.4. This suggests that GeLU inadequately enhances the data representation capabilities of the neural network, necessitating the exploration of a more effective activation function to improve model performance.

4.3.4 Baseline+Space-to-Depth+Shape-IOU

Although the sole integration of Shape−IoU[23] did not enhance performance, its combination with space-to-depth improves the $mAP_{0.5:0.95}$ by 1.1. However, $mAP_{0.5}$ experiences a degradation of 0.9 compared to the scenario with only space-to-depth. Therefore, further optimization is required.

4.3.5 Baseline+Space-to-Depth+Shape-IOU+DGM

Due to the degradation observed in $mAP_{0.5}$ during the previous ablation study, we design the DGM by integrating global and depth features to enhance performance further. The results show improvements in $mAP_{0.5:0.95}$ and $mAP_{0.5}$ by 1.2 and 1.9, respectively, compared to the baseline. This indicates the effectiveness of the DGM and aligns with our initial intent for designing this module.

4.3.6 Baseline+Space-to-Depth+DGM

Due to the performance degeneration observed with Shape-IoU and the enhancements achieved with space-to-depth and DGM, we are prompted to conduct an ablation study on space-to-depth and DGM. We discover that the combination of space-to-depth and DGM improved $mAP_{0.5:0.95}$ and $mAP_{0.5}$ by 1.8 and 2.0, respectively. This further confirms the effectiveness of the involved DGM and the space-to-depth.

4.3.7 Baseline+Space-to-Depth+DGM+CRM

We observe that moisture detection performance is the poorest by evaluating the existing methods. By incorporating environmental semantics, we design a Contextual Residual Module (CRM), and the results of integrating CRM are shown in Table 5. The accuracy for moisture detection improved by 3.6% and 3.2%. Cracks are often caused by environmental vibrations, so it is common to find additional cracks in the vicinity of an existing one. On the other hand, spalling is typically a result of material quality issues in construction or industrial products, and is less related to environmental factors. Therefore, as shown in Table 5, the detection accuracy for cracks improves, but environmental information introduces redundant data for the spalling category, leading to a decrease in spalling detection accuracy. Additionally, the overall $mAP_{0.5:0.95}$ increased by 1.3%. These results are consistent with our initial intent for designing this module.

Table 5 Ablation study on CRM

Method	**mAP**$_{all}$ **(%)↑**		**mAP**$_{crack}$ (%)↑		**mAP**$_{spalling}$ (%)↑		**mAP**$_{moisture}$ **(%)↑**	
-	$mAP_{0.5}$	$mAP_{0.5:0.95}$	$mAP_{0.5}$	$mAP_{0.5:0.95}$	$mAP_{0.5}$	$mAP_{0.5:0.95}$	$mAP_{0.5}$	$mAP_{0.5:0.95}$
w/o CRM	81.5	54.8	83.7	53.3	91.6	66.3	69.3	44.8
w/ CRM	**81.2**	**55.5**	**82.5**	**53.5**	**89.6**	**65.6**	**71.5**	**47.3**

4.3.8 Baseline+Space-to-Depth+DGM+CRM+Kilu

In the previous ablation study, while $mAP_{0.5}$ show improvement, $mAP_{0.5:0.95}$ experience a slight regression. To enhance the neural network's ability to represent complex data without additional parameters, we propose a flexible activation function. This function allows the adjustment of the parameter α, improving the network's nonlinear representation capabilities at varying depths, as shown in Table 6. Initial experiments with α values ranging from 0.3 to 1.0 indicate optimal accuracy at $\alpha = 0.5$. Further testing focused around $\alpha = 0.5$ reveal that accuracy peaks at $\alpha = 0.55$. The comparative experiments shown in Table 4.3.8 demonstrate that the proposed activation function, Kilu, outperforms 20 other activation functions, including Mish[45], SiLU, and ReLU, in terms of activation performance.

Table 6 Ablation study on Kilu

α	0.3	0.4	0.45	0.5	0.54	0.55	0.56	0.57	0.6	0.7	0.8	0.9	1.0
$mAP_{0.5:0.95}$	55.2	55.6	56.1	55.6	54.9	**56.3**	55.1	55.4	54.4	55.5	54.6	54.7	54.9
$mAP_{0.5}$	80.0	80.6	81.1	80.8	79.7	**81.3**	80.3	80.4	80.6	80.5	80.7	80.7	79.7

Table 7 Activation function comparison

Activation function	$\mathbf{mAP}_{0.5:0.95}$ (%)↑	$\mathbf{mAP}_{0.5}$ (%)↑
RReLU[46]	54.8	80.5
Mish[45]	54.2	80.0
LeakyReLU[47]	55.2	80.9
Tanh	49.6	75.4
Tanhshrink	48.2	73.8
Hardshrink[48]	10.1	22.8
ReLU[20]	53.7	79.7
GeLU[21]	53.8	78.1
SiLU	55.5	81.2
PReLU	54.0	79.3
CELU	54.6	79.7
Hardtanh[49]	49.0	75.7
ReLU6[50]	54.2	79.6
Hardswish[51]	55.2	79.8
ELU[52]	55.5	81.0
SELU[53]	53.5	78.1
Sigmoid[54]	47.2	73.1
Softsign[55]	49.3	74.0
LogSoftmax	40.8	64.5
Softshrink	48.9	74.4
Kilu(ours)	**56.3**	**81.3**

4.3.9 Baseline+Space-to-Depth+DGM+CRM+Kilu+Shape-IOU

Although prior experiments indicate that Shape-IoU might impair performance, we decided to reevaluate it as a regression loss function based on findings from an earlier ablation study. This reevaluation indeed confirms a reduction in accuracy, yet an increase in accuracy ($MAP_0.5$) is observed on the NEU-Det dataset with an input size of 640. This indicates that Shape-IoU's effectiveness is sensitive to variations in bounding box size and shape, which are influenced by changes in input size, underscoring that its efficacy is contingent upon input dimensions.

4.4 Analysis of Activation Feature Maps

The visualization of activation maps across various layers in a deep neural network is a critical technique for elucidating the internal mechanics of model learning processes. To dissect the impact of each component on the extraction and learning of salient features, we present the activation maps associated with different non-linear functions in Figure 11. The set comprises 32 distinct activation maps. An observable phenomenon in Figure 11 is the prevalence of deeper blue hues in certain maps, indicative of the dying neuron issue, which stems from the non-linearities inability to effectively manage negative values during activation.

The space-to-depth downsampling method is engineered to transmute spatial features into depth features while ensuring the integrity of feature transmission is maintained. This methodology results in a distinctly refined feature map in Figure 11(c) compared to Figure 11(b).

Furthermore, this approach adeptly mitigates the influence of non-pertinent features. For example, although the water pipe is conspicuous in both the input image and Figure 11(b), it is markedly subdued in Figure 11(c). Additionally, it can be observed that the feature maps are more refined and smooth, facilitating the computation of loss. The DGM specifically enhances the representation of critical target features.

The CRM is specifically designed to exploit the contextual semantics of the ambient environment to augment the detection capabilities concerning moisture. This is evidenced in Figure 11(e), where the CRM broadens the activation regions pertinent to the environmental context, thereby yielding more pronounced features relative to those seen in Figure 11(d).

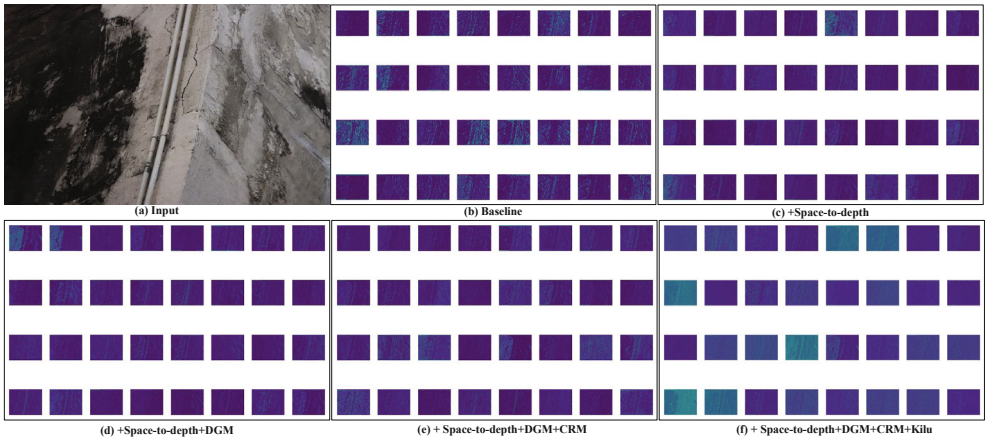

Figure 11 Visualization of feature map for different components

Kilu tends to generate fewer non-learnable filters compared to other activations. This characteristic fosters more expansive activation zones and a more homogeneous feature map distribution. Such uniformity in the output feature maps is beneficial to easier optimization and better generalization[56]. This factor contributes to the superiority of Kilu over SiLU and GeLU.

4.5 Deployment for Edge-Computing: NVIDIA Jetson Orin NX

To demonstrate the deployability of our method on edge-computing devices mounted on unmanned aerial vehicles (UAVs), we conduct experiments to validate its effectiveness in such a setting. In practical applications of UAV-based building or industrial product inspection, it is sufficient to simply mount edge computing devices equipped with the detection system on the drones for use. Consequently, we implement CARNet on an NVIDIA Jetson Orin NX device with 16 GB GPU memory and 8 CPU cores. The Orin NX is equipped with a USB camera. We test the exterior walls, bridge rails, and bridge piers of Cheung Shu Tan Village in Hong Kong on-site. We are using input frames at a resolution of 1024×1024. The CARNet-n model effectively detects defects and achieves an enjoyable **real-time** detection of **15.0 FPS**. Figure 12 displays the on-site detection results on edge-computing device. We conduct detections on four buildings, with each column representing the results of defect detection at different locations of the same building.

Through the simulations above and real-world experiments, we have thoroughly demonstrated that our method can be deployed on edge-computing devices while ensuring high accuracy, confirming the reliability and feasibility of our approach, and aligning with our initial expectations.

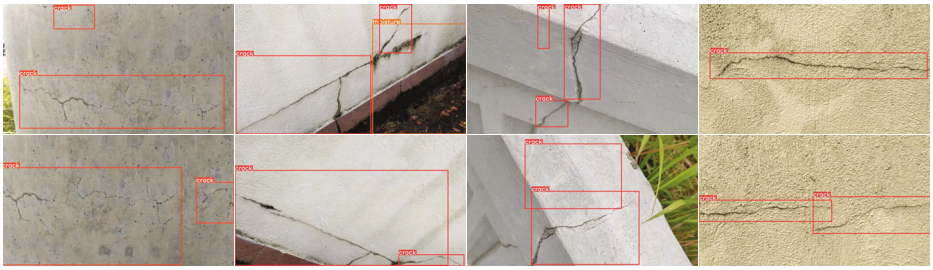

Figure 12 Field test results for edge computing devices. Please zoom in for the best view

5 Conclusion

In this article, we have presented a environmental information-assisted and activation representation network for accurate and fast defect detection. Our network named CARNet employs convolutional space-to-depth for extracting features and downsampling. The proposed DGM and CRM for feature enhancement and fusion. The environmental information can assistant locate the defect target and defect prediction. Novel and flexible activation functions further stimulate the network's capacity for nonlinear representation, thereby enhancing its detection performance. Our approach enhances accuracy while reducing parameters by 82% for CNN methods. Additionally, it surpasses Transformer-based methods in accuracy with an 89% reduction in parameters and an 86% decrease in computational cost, achieving nearly eight times faster inference speed. The real-time detection of 15 FPS deployed in edge computing validates the portability and reliability of our approach.

In the future, we will extend our method to larger datasets to validate its effectiveness and apply a weighted loss function to address the common issue of data imbalance in large-scale datasets. Additionally, we plan to integrate extra sensors to capture more environmental information to aid defect detection, such as using infrared sensors and acoustic sensors to gather additional signals, enhancing the robustness of model training and improving defect detection accuracy.

Acknowledgement

The authors would like to thank GAO Chuanxiang, ZHAO Benyun, LI Guang and DOU Jia for their assistance with dataset annotation and data collection.

Conflict of Interest

CHEN Jie is Editor-in-Chief for Journal of Systems Science & Complexity and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

References

[1] Zhang D J and Zhang Y S, Fault detection for uncertain delta operator systems with two-channel packet dropouts via a switched systems approach, *Journal of Systems Science & Complexity*, 2020, **33**(5): 1446–1468.

[2] Gao Y Q, Yang J F, Qian H J, et al., Multiattribute multitask transformer framework for vision-based structural health monitoring, *Computer-Aided Civil and Infrastructure Engineering*, 2023, **38**(17): 2358–2377.

[3] Shen L J, Ma C, Luo J, et al., An automatic classification pipeline for the complex synaptic structure based on deep learning, *Journal of Systems Science & Complexity*, 2022, **35**(4): 1398–1414.

[4] Xie G Y, Wang J B, Liu J Q, et al., IM-IAD: Industrial image anomaly detection benchmark in manufacturing, *IEEE Transactions on Cybernetics*, 2024, **54**(5): 2720–2733.

[5] Xie L F, Xiang X, Xu H N, et al., FFCNN: A deep neural network for surface defect detection of magnetic tile, *IEEE Transactions on Industrial Electronics*, 2020, **68**(4): 3506–3516.

[6] Hu C F and Wang Y X, An efficient convolutional neural network model based on object-level attention mechanism for casting defect detection on radiography images, *IEEE Transactions on Industrial Electronics*, 2020, **67**(12): 10922–10930.

[7] Gu X D, Deng F, Gao X, et al., An improved sensor fault diagnosis scheme based on TA-LSSVM and ECOC-SVM, *Journal of Systems Science & Complexity*, 2018, **31**(2): 372–384.

[8] Yang Z S, Xu Z, and Wang Y H, Bidirection-Fusion-YOLOv3: An improved method for insulator defect detection using uav image, *IEEE Transactions on Instrumentation and Measurement*, 2022, **71**: 1–8.

[9] Ma D, Fang H Y, Wang N N, et al., Automatic detection and counting system for pavement cracks based on pcgan and yolo-mf, *IEEE Transactions on Intelligent Transportation Systems*, 2022, **23**(11): 22166–22178.

[10] Yoo Y-H, Kim U-H, and Kim J-H, Convolutional recurrent reconstructive network for spatiotemporal anomaly detection in solder paste inspection, *IEEE Transactions on Cybernetics*, 2022, **52**(6): 4688–4700.

[11] Gao C X, Wang X Y, Wang R Y, et al., A UAV-based explore-then-exploit system for autonomous indoor facility inspection and scene reconstruction, *Automation in Construction*, 2023, **148**: 104753.

[12] Zhao B Y, Zhou X K, Yang G D, et al., High-resolution infrastructure defect detection dataset sourced by unmanned systems and validated with deep learning, *Automation in Construction*, 2024, **163**: 105405.

[13] Sunkara R and Luo T, No more strided convolutions or pooling: A new cnn building block for low-resolution images and small objects, *Machine Learning and Knowledge Discovery in Databases*, Springer Nature Switzerland, Cham, 2023, 443–459.

[14] Wang C-Y, Yeh I-H, and Liao H-Y M, YOLOv9: Learning what you want to learn using programmable gradient information, 2024, arXiv: 2402.13616.

[15] Liu B F, Chen C-H, Zheng P, et al., An adaptive parallel feature learning and hybrid feature fusion-based deep learning approach for machining condition monitoring, *IEEE Transactions on Cybernetics*, 2023, **53**(12): 7584–7595.

[16] Hu B Z, Gao B, Woo W L, et al., A lightweight spatial and temporal multi-feature fusion network for defect detection, *IEEE Transactions on Image Processing*, 2020, **30**: 472–486.

[17] Gao Z S, Yang G D, Li E, et al., Novel feature fusion module-based detector for small insulator defect detection, *IEEE Sensors Journal*, 2021, **21**(15): 16807–16814.

[18] Li H R, Chen Y R, Zhang Q C, et al., BiFNet: Bidirectional fusion network for road segmentation, *IEEE Transactions on Cybernetics*, 2022, **52**(9): 8617–8628.

[19] Nie X B and Zheng W X, Dynamical behaviors of multiple equilibria in competitive neural networks with discontinuous nonmonotonic piecewise linear activation functions, *IEEE Transactions on Cybernetics*, 2016, **46**(3): 679–693.

[20] Glorot X, Bordes A, and Bengio Y, Deep sparse rectifier neural networks, *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings*, 2011, 315–323.

[21] Hendrycks D and Gimpel K, Gaussian error linear units (gelus), 2016, arXiv: 1606.08415.

[22] Jocher G, Chaurasia A, and Qiu J, YOLO by Ultralytics, January 2023.

[23] Zhang H and Zhang S J, Shape-IoU: More accurate metric considering bounding box shape and scale, 2023, arXiv: 2312.17663.

[24] Han H N, Yang R, Li S Y, et al., SSGD: A smartphone screen glass dataset for defect detection, *ICASSP* 2023–2023 *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2023, 1–5.

[25] He Y, Song K C, Meng Q G, et al., An end-to-end steel surface defect detection approach via fusing multiple hierarchical features, *IEEE Transactions on Instrumentation and Measurement*, 2019, **69**(4): 1493–1504.

[26] Zhao B Y, Zhou X K, Yang G D, et al., High-resolution infrastructure defect detection dataset sourced by unmanned systems and validated with deep learning, *Automation in Construction*, 2024, **163**: 105405.

[27] Li C Y, Li L L, Jiang H L, et al., YOLOv6: A single-stage object detection framework for industrial applications, 2022, arXiv: 2209.02976.

[28] Wang C-Y, Bochkovskiy A, and Liao H-Y M, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 7464–7475.

[29] Lv W Y, Xu S L, Zhao Y, et al., Detrs beat yolos on real-time object detection, 2023, arXiv: 2304.08069.

[30] Wang A, Chen H, Liu L H, et al., YOLOv10: Real-time end-to-end object detection, 2024.

[31] Ren S Q, He K M, Girshick R, et al., Faster RCNN: Towards real-time object detection with region proposal networks, 2016, arXiv: 1506.01497.

[32] Cai Z W and Vasconcelos N, Cascade R-CNN: Delving into high quality object detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 6154–6162.

[33] Lin T-Y, Goyal P, Girshick R, et al., Focal loss for dense object detection, *Proceedings of the IEEE International Conference on Computer Vision*, 2017, 2980–2988.

[34] Tian Z, Shen C H, Chen H, et al., FCOS: Fully convolutional one-stage object detection, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 9627–9636.

[35] Zhang S F, Chi C, Yao Y, et al., Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 9759–9768.

[36] Li X, Wang W H, Wu L J, et al., Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection, *Advances in Neural Information Processing Systems*, 2020, **33**: 21002–21012.

[37] Jocher G, YOLOv5 by Ultralytics, 2020, https://github.com/ultralytics/yolov5.

[38] Ge Z, Liu S T, Wang F, et al., YOLOX: Exceeding yolo series in 2021, 2021, arXiv: 2107.08430.

[39] Liu Z, Lin Y T, Cao Y, et al., Swin transformer: Hierarchical vision transformer using shifted windows, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 10012–10022.

[40] Wang W H, Xie E, Li X, et al., Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 568–578.

[41] Yang R, Ma H L, Wu J, et al., Scalablevit: Rethinking the context-oriented generalization of vision transformer, *European Conference on Computer Vision*, Springer, 2022, 480–496.

[42] Li K C, Wang Y L, Gao P, et al., Uniformer: Unified transformer for efficient spatiotemporal representation learning, 2022, arXiv: 2201.04676.

[43] Peng Y K, Xia F, Zhang C L, et al., Deformation feature extraction and double attention feature pyramid network for bearing surface defects detection, *IEEE Transactions on Industrial Informatics*, 2024, **20**(6): 9048–9058.

[44] Liu W, Anguelov D, Erhan D, et al., SSD: Single shot multibox detector, *Computer Vision–ECCV* 2016*: 14th European Conference, Amsterdam, The Netherlands, October* 11–14, 2016*, Proceedings, Part I* 14, Springer, 2016, 21–37.

[45] Misra D, Mish: A self regularized non-monotonic activation function, 2019, arXiv: 1908.08681.

[46] Xu B, Empirical evaluation of rectified activations in convolutional network, 2015, arXiv: 1505.00853.

[47] Maas A L, Hannun A Y, Ng A Y, et al., Rectifier nonlinearities improve neural network acoustic models, *Proc. Icml*, Atlanta, GA, 2013, **30**: 3.

[48] Konda K, Memisevic R, and Krueger D, Zero-bias autoencoders and the benefits of co-adapting features, 2014, arXiv: 1402.3337.

[49] Lupu D and Necoara I, Exact representation and efficient approximations of linear model predictive control laws via hardtanh type deep neural networks, *Systems & Control Letters*, 2024, **186**: 105742.

[50] Howard A G, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv: 1704.04861.

[51] Howard A, Sandler M, Chu G, et al., Searching for mobilenetv3, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 1314–1324.

[52] Clevert D-A, Unterthiner T, and Hochreiter S, Fast and accurate deep network learning by exponential linear units (elus), 2015, arXiv: 1511.07289.

[53] Klambauer G, Unterthiner T, Mayr A, et al., Self-normalizing neural networks, *Advances in Neural Information Processing Systems*, 2017, arXiv: 1706.02515.

[54] Elfwing S, Uchibe E, and Doya K, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, *Neural Networks*, 2018, **107**: 3–11.

[55] Pandey G K and Srivastava S, Resnet-18 comparative analysis of various activation functions for image classification, 2023 *International Conference on Inventive Computation Technologies*, IEEE, 2023, 595–601.

[56] Li H, Xu Z, Taylor G, et al., Visualizing the loss landscape of neural nets, *Advances in Neural Information Processing Systems*, 2018, 6391–6401, arXiv: 1712.09913.