Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064191

ARTICLE

RC2DNet: Real-Time Cable Defect Detection Network Based on Small Object Feature Extraction

Zilu Liu[1,#] and Hongjin Zhu[2,*,#]

[1]School of Mechanical Engineering, Jiangsu University of Technology, Changzhou, 213001, China
[2]School of Computer Engineering, Jiangsu University of Technology, Changzhou, 213001, China

*Corresponding Author: Hongjin Zhu. Email: zhuhongjin@jsut.edu.cn

[#]These authors contributed equally to this work

Received: 07 February 2025; Accepted: 17 March 2025; Published: 29 August 2025

ABSTRACT: Real-time detection of surface defects on cables is crucial for ensuring the safe operation of power systems. However, existing methods struggle with small target sizes, complex backgrounds, low-quality image acquisition, and interference from contamination. To address these challenges, this paper proposes the Real-time Cable Defect Detection Network (RC2DNet), which achieves an optimal balance between detection accuracy and computational efficiency. Unlike conventional approaches, RC2DNet introduces a small object feature extraction module that enhances the semantic representation of small targets through feature pyramids, multi-level feature fusion, and an adaptive weighting mechanism. Additionally, a boundary feature enhancement module is designed, incorporating boundary-aware convolution, a novel boundary attention mechanism, and an improved loss function to significantly enhance boundary localization accuracy. Experimental results demonstrate that RC2DNet outperforms state-of-the-art methods in precision, recall, F1-score, mean Intersection over Union (mIoU), and frame rate, enabling real-time and highly accurate cable defect detection in complex backgrounds.

KEYWORDS: Surface defect detection; computer vision; small object feature extraction; boundary feature enhancement

1 Introduction

Cable trenches, as integral infrastructure within power plants, substations, and power grids, primarily serve the function of cable installation. These cables, akin to the arteries of the power grid, interconnect various critical devices, including relay protection systems, automatic control systems, communication equipment, and measurement and metering instruments, forming the neural hub of power generation, transformation, and distribution systems. Within substations, underground cables are typically carefully laid in specially designed cable tunnels to optimize space usage and ensure the safe operation of the cables [1]. These cables consist of conductors, insulation layers, and protective coatings, playing a vital role in information transmission and power distribution. Compared to overhead power lines, underground cables occupy less space, offer more reliable power transmission, and exhibit a greater ability to resist external interference.

However, despite the numerous advantages of underground cables, their inspection and maintenance present significant challenges. In particular, within the complex tunnel environment, cables may suffer damage due to natural aging, chemical corrosion, or rodent gnawing, potentially leading to safety incidents such as fires [2]. At present, cable inspections in cable trenches still largely rely on manual labor, which is not only inefficient and costly but also poses considerable safety risks. Cables generate significant heat during operation, and in the event of smoldering, they can quickly produce large amounts of smoke and toxic gases, presenting a serious threat to the safety of personnel.

In recent years, with the rapid development of deep learning technologies, the use of deep learning for surface defect detection in cables has increased [3]. However, this field still faces several challenges. As cables are often laid in narrow tunnels, and surface defects are typically small in size, detection algorithms struggle to accurately capture and identify these defects. Furthermore, the environment within the cable tunnels is dynamic, with various potential sources of interference, such as brackets, pipes, and other cables, which complicate the background information and make defect detection more difficult [4,5]. In addition to these challenges, factors such as insufficient lighting, poor camera angles, or lens contamination can result in blurry images captured by the equipment, hindering accurate defect detection. Additionally, the cable surface may be covered with dust, oil, or other contaminants, which obscure or blur defect features, leading to misdetections or missed detections by the algorithms [6,7].

To address issues such as small target sizes, complex backgrounds, poor image quality, and potential contamination interference [8,9], this paper designs a Real-Time Cable Defect Detection Network (RC2DNet). The primary contributions are as follows:

- We propose a small-object feature extraction module that not only integrates Feature Pyramid Networks (FPN) for multi-level feature fusion but also introduces an adaptive weighting mechanism to dynamically adjust feature importance. Additionally, an improved Focal Loss is incorporated to enhance the semantic representation and detection accuracy of small objects. Unlike existing approaches, our module employs a global-local dynamic adjustment strategy to further refine small-object feature representation.
- We design a boundary feature enhancement module that leverages boundary-aware convolution and a dedicated boundary-aware branch to capture fine-grained boundary details. Moreover, we introduce a novel boundary attention mechanism to reinforce feature representation in boundary regions, significantly improving localization accuracy. Compared to conventional boundary enhancement methods, our approach is more robust in distinguishing object boundaries from background noise in complex scenes.
- Extensive experiments demonstrate that RC2DNet achieves a superior balance between detection accuracy and speed in cable defect detection tasks. It outperforms existing state-of-the-art methods in key metrics, including precision, recall, F1-score, mean Intersection over Union (mIoU), and frame rate. Our evaluation covers both standard datasets and real-world industrial scenarios, validating the model's generalization capability.

2 Related Works

Surface defect detection for cables faces challenges such as small defect sizes, complex backgrounds, and image quality issues. Chen et al. [10] proposed an Alternating Current Field Measurement (ACFM)-based method, modeling weld defects in COMSOL Multiphysics to analyze induced current density and magnetic field signals. They designed an orthogonal receiving coil for defect characterization and validated its effectiveness experimentally. However, ACFM struggles with small target localization and image quality variations, limiting its practical accuracy.

Deep learning, particularly You Only Look Once (YOLO)-based object detection, has shown promise in defect detection [11]. Chen et al. [12] optimized YOLOv7 by integrating EfficientViT-M4 as the backbone, introducing deformable convolution, and employing weighted intersection over union (WIoU) loss to improve regression. Despite efficiency gains, class imbalance remains a challenge. Zhang et al. [13] enhanced

YOLACT++ with a Convolutional Block Attention Module (CBAM), optimized anchor generation, and introduced the Mish activation function [14], achieving higher mean average precision but still exhibiting errors compared to manual measurements. Wen et al. [15] improved YOLOv3 for bridge cable defect detection by merging batch normalization (BN) layers with convolutions and embedding CBAM, raising precision to 82.3%, though still insufficient for safety-critical applications. Wang et al. [16] refined YOLOv5s by incorporating a TRANS module for feature extraction, yet its generalization across different environments remains unverified.

Beyond YOLO, edge detection techniques have been explored. Xu et al. [17] combined visual inspection with diameter measurement, using local grayscale contrast enhancement and improved threshold segmentation for defect identification. Their method achieved recall rates of 80.4% and 85.2%, though generalization remains a concern. Wu et al. [18] proposed a deep learning-based detection network allowing less precise boundary localization for improved overall performance, but accurate defect localization remains critical. Liu et al. [19] developed a semi-supervised autoencoder-based anomaly detection model (AEAL), leveraging normal samples to refine defect localization, but its effectiveness for regular cable defects is limited. Tong et al. [20] introduced FAST, a defect detection algorithm suitable for medium- and long-distance inspection, yet its robustness in highly complex environments remains uncertain.

Despite advancements, challenges persist. Electromagnetic-based methods suffer from environmental interferences, while deep learning models, though fast, struggle with small defect detection and background complexity. Edge detection techniques and hybrid approaches offer improvements but often lack robustness and generalization, underscoring the need for further research.

3 Proposed Defect Detection System

3.1 Overall Architecture

To improve cable defect detection, particularly for small targets and boundary localization, we propose the Real-time Cable Defect Detection Network (RC2DNet). Unlike standard YOLO-based models, RC2DNet introduces a small object feature extraction module and a boundary feature enhancement module, leveraging multi-scale fusion, adaptive weighting, and boundary-aware convolution.

The overall architecture is shown in Fig. 1. The backbone consists of five stages: an initial convolution layer, three C3 blocks, and a spatial pyramid pooling–fast (SPPF) module, extracting hierarchical features efficiently.

The small object feature extraction module follows a feature pyramid network (FPN) approach. Features from different backbone levels (C3, C4, C5) undergo convolutional transformations and weighted fusion (P3, P4, P5), dynamically adjusting feature importance based on small object presence. The adaptive weighting function is defined as:

$$W_i = \frac{1}{1 + e^{-\alpha(R_i - \theta)}} \tag{1}$$

where W_i is the weight for the i-th feature map, R_i is its response intensity, α controls sensitivity, and θ sets the transition threshold. This mechanism prioritizes small targets without suppressing larger structures.

To refine localization, the boundary feature enhancement module employs a boundary-aware convolutional branch, generating a boundary attention map via deformable convolutions and a sigmoid activation. This map highlights defect edges and is fused with feature maps via residual connections, preserving edge details.

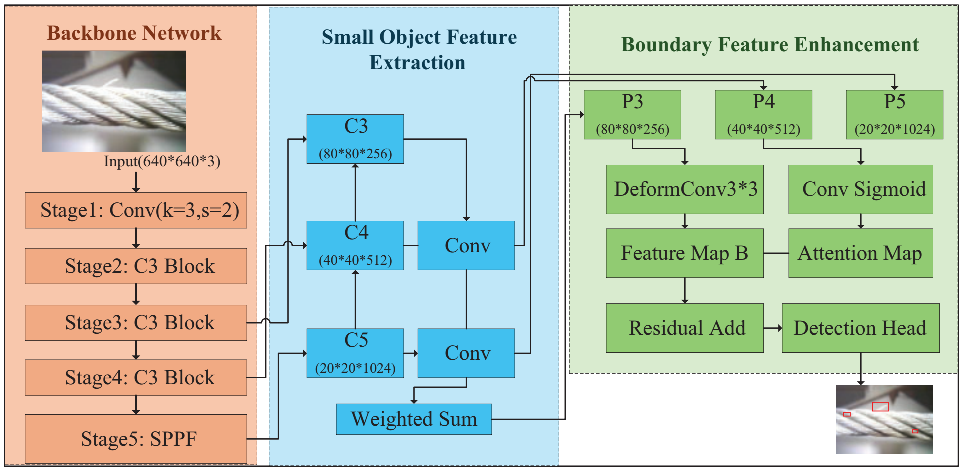


Figure 1: Overall architecture of RC2DNet

Finally, the detection head integrates enhanced features for classification and localization, with a boundary-aware loss improving detection in blurred or irregular defect regions. RC2DNet balances accuracy and efficiency, making it well-suited for real-time industrial defect inspection.

3.2 Data Preprocessing

In the task of cable surface defect detection, data preprocessing is essential to enhance the model's generalization ability for complex backgrounds and diverse defect samples. The main preprocessing operations include data augmentation techniques, such as rotation, cropping, brightness adjustment, and contrast adjustment, as well as filtering operations.

The rotation and cropping operations increase the diversity of training samples by randomly altering the image orientation and scale. Specifically, the original image is rotated by a randomly selected angle, followed by cropping a fixed-sized region from the rotated image. The center of the cropped region is denoted as (x_c, y_c), while its width and height are represented by w_c and h_c, respectively. Brightness and contrast adjustments are achieved through linear transformations of pixel values, as expressed by the following formula:

$$I_{\text{adjusted}} = \alpha \cdot I_{\text{original}} + \beta \tag{2}$$

where I_{original} and I_{adjusted} denote the original and adjusted pixel values, respectively. The parameter α controls contrast, β adjusts brightness, and $\alpha > 0, \beta \in \mathbb{R}$.

During image acquisition, the surface of cables may be affected by contamination, dust, or other external interferences, leading to reduced image quality. To address this, appropriate filtering methods are applied to remove noise while preserving critical features. RC2DNet adopts high-pass filtering, which focuses on retaining high-frequency components to emphasize edges and texture details. The high-pass filtering in RC2DNet is implemented using the Laplacian operator:

$$\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{3}$$

The specific form is given as:

$$\Delta I = I(x+1, y) + I(x-1, y) + I(x, y+1) + I(x, y-1) - 4 \cdot I(x, y) \tag{4}$$

The filtered image can be further enhanced by combining the original image with the high-pass filtering result:

$$I_{\text{enhanced}} = I_{\text{original}} + \lambda \cdot \Delta I \tag{5}$$

Here, λ is the enhancement factor, which controls the prominence of edges in the final image.

3.3 Small Object Feature Extraction

In the task of cable surface defect detection, high-resolution images captured by cameras are often dominated by background information, with small objects occupying only a few pixels. To address this, RC2DNet improves the YOLO feature extraction module and decoding structure, while introducing a small object perception mechanism to enhance the model's sensitivity and detection performance for small objects.

Traditional feature extraction networks tend to prioritize high-level semantic information over low-level spatial information, whereas small objects typically rely on high-resolution spatial features. By adopting a Feature Pyramid Network (FPN) structure to fuse multi-scale feature maps, the network can effectively represent small objects. Let the input image be I, and the multi-layer feature maps extracted by the backbone network be $\{F_1, F_2, \ldots, F_n\}$, with corresponding downsampling ratios of $\{s_1, s_2, \ldots, s_n\}$. The feature fusion formula is as follows:

$$F_i^{\text{fused}} = \text{Conv}(F_i) + \text{Upsample}(F_{i+1}), \quad i = 1, \ldots, n-1 \tag{6}$$

where Upsample represents the upsampling operation implemented via bilinear interpolation, Conv denotes the convolution operation to align feature channels, and F_i^{fused} represents the multi-scale fused feature map.

The small object feature extraction module performs preliminary processing on high-level feature maps to generate an activation probability map, calculated as follows:

$$P_{\text{foreground}} = \sigma(\text{Conv}(F_{\text{high}})) \tag{7}$$

where σ is the Sigmoid activation function that constrains feature values to the range $[0, 1]$, representing the activation probability of the foreground. The activation map is then progressively upsampled and combined with intermediate feature maps to produce finer activation features:

$$F_{\text{foreground}} = P_{\text{foreground}} \odot F_{\text{mid}} \tag{8}$$

where $\odot$ represents element-wise multiplication to incorporate foreground attention. The resulting foreground features are aligned with the size of the input image for further processing.

To refine the activated foreground features and remove redundant background information, a semantic refinement module is introduced to classify foreground objects into multiple categories. Assuming the total number of categories is C, the refined category probability distribution is denoted as P_{class}. For a foreground pixel, the category probability is computed as:

$$P_{\text{class}}(c) = \text{Softmax}(\text{Conv}_c(F_{\text{foreground}})), \quad c = 1, \ldots, C \tag{9}$$

where Conv_c represents the convolution operation associated with class c.

To balance the weights of small objects and the background during classification, a collaborative probability fusion strategy is adopted, combining the foreground activation probability and refinement probability into a joint probability:

$$P_{\text{joint}} = P_{\text{foreground}} \cdot P_{\text{class}} \tag{10}$$

The final loss function is defined as a combination of cross-entropy losses:

$$\mathcal{L} = \sum_{i=1}^{N} \left[-y_i \log(P_{\text{joint},i}) \right] \tag{11}$$

where y_i denotes the true category of pixel i.

To address sample imbalance, an optimization strategy based on small object mining is designed. By computing the loss for each pixel, the top $\gamma \cdot N$ pixels with the highest loss are selected for backpropagation:

$$\mathcal{S} = \text{TopK}(\mathcal{L}, \gamma \cdot N) \tag{12}$$

where γ is the sampling ratio, and N is the total number of pixels.

The specific process of small object feature extraction is shown in Algorithm 1:

Algorithm 1: Small object feature extraction algorithm

```
Require: High-resolution cable surface image $I$, sampling ratio $\gamma$
Ensure: Joint activation probability $P_{\text{joint}}$, total loss $\mathcal{L}$
Step 1: Extract multi-layer feature maps $\{F_1, F_2, \ldots, F_n\}$ using the backbone network.
Step 2: Perform progressive upsampling and fusion to generate multi-scale fused features.
Step 3: Select interpolation method for upsampling. Use bilinear interpolation for resizing feature maps.
Step 4: For each feature map $F_i$ in $\{F_1, F_2, \ldots, F_n\}$:
    a. Perform upsampling using bilinear interpolation:
        $F_{\text{upsampled}} = \text{BilinearUpsample}(F_i, \text{scale factor})$
    b. Fuse the upsampled feature map with other feature maps to obtain multi-scale fused features.
Step 5: Compute foreground activation probability using the sigmoid function:
    $P_{\text{foreground}} = \sigma(\text{Conv}(F_{\text{high}}))$
    where $F_{\text{high}}$ is the high-resolution feature map.
Step 6: Perform upsampling on $P_{\text{foreground}}$ using bilinear interpolation:
    $P_{\text{foreground}}^{\text{upsampled}} = \text{BilinearUpsample}(P_{\text{foreground}}, \text{scale factor})$
Step 7: Refine foreground features by multiplying upsampled $P_{\text{foreground}}$ with intermediate feature maps:
    $F_{\text{foreground}} = P_{\text{foreground}}^{\text{upsampled}} \times F_{\text{intermediate}}$
Step 8: Input the refined features $F_{\text{foreground}}$ into a multi-class classification module:
    Compute category probabilities using the softmax function:
    $P_{\text{class}} = \text{Softmax}(\text{Conv}_c(F_{\text{foreground}}))$
Step 9: Combine foreground and category probabilities to form the joint probability:
    $P_{\text{joint}} = P_{\text{foreground}} \times P_{\text{class}}$
Step 10: Compute pixel-wise loss based on $P_{\text{joint}}$ and ground truth labels:
```

(Continued)

Algorithm 1 (continued)

```
    $\mathcal{L}_{\text{pixel}} = \sum \left| P_{\text{joint}} - \text{Ground Truth} \right|$
**Step 11:** Select the top $\gamma \cdot N$ pixels with the highest loss for backpropagation:
    Top Pixels = $\text{SelectTop}(N, \gamma)$
**Step 12:** Return $P_{\text{joint}}$ and the total loss $\mathcal{L}$.
```

3.4 Boundary Feature Enhancement

In complex scenarios, boundary information plays a critical role in the precise localization and classification of targets. The ambiguity of small object boundaries significantly reduces detection accuracy. To address this, RC2DNet designs a boundary feature enhancement (BFE) module to improve the model's sensitivity to boundary information, thereby optimizing detection performance. This section describes the module in terms of feature fusion, boundary-aware convolution, and loss function design.

To effectively fuse multi-scale features and enhance boundary representation, a Boundary-Guided Attention (BGA) mechanism is introduced. This mechanism learns a boundary feature weight matrix to suppress redundant information in non-boundary regions. The calculation formula is as follows:

$$F_{\text{boundary}} = \sigma(W_b \cdot F) \odot F \tag{13}$$

where σ represents the Sigmoid activation function, W_b denotes the boundary attention weights learned through convolution, $\odot$ is element-wise multiplication, and F_{boundary} represents the enhanced boundary features.

The boundary-aware convolution module enhances the expression of boundary information to improve localization accuracy. This module first extracts the boundary regions in feature maps and uses boundary information to guide the convolution operation. The key addition is a boundary-adaptive kernel weight matrix, which scales convolutional responses based on boundary importance:

$$F' = W_b \odot \text{Conv}(F) + (1 - W_b) \odot F \tag{14}$$

where W_b is the learned boundary importance map, $\text{Conv}(F)$ is the standard convolution, and F' represents the enhanced feature map. This equation ensures that boundary regions are emphasized, while non-boundary areas retain their original feature representations. Next, boundary regions are computed through horizontal and vertical projections:

$$A_x = \text{Conv}(F), \quad A_y = \text{Conv}(F) \tag{15}$$

where A_x and A_y are the horizontal and vertical attention weights, respectively, predicted via convolution. The target boundary region is divided into multiple bins, with each bin representing a candidate region for boundary points. The bin width is defined as:

$$w_{\text{bin}} = \frac{\Delta x}{n_{\text{bins}}} \cdot k \tag{16}$$

where n_{bins} is the number of bins, k is a scaling factor, and Δx represents the boundary box width. The choice of n_{bins} determines the resolution of the binning, balancing the need for sufficient granularity in detecting small target features and computational efficiency. k is a scaling factor that adjusts the sensitivity of the bin width, ensuring that boundary points are localized effectively within the target regions. The parameter Δx corresponds to the width of the boundary box, controlling the overall scale of the binning process.

Within each bin, fine boundary regression predicts the precise position of boundary points. The regression loss function adopts the smooth L_1 loss, defined as:

$$\mathcal{L}_{\text{reg}} = \begin{cases} 0.5(\Delta p)^2, & \text{if } |\Delta p| < 1, \\ |\Delta p| - 0.5, & \text{otherwise,} \end{cases} \tag{17}$$

where Δp is the difference between the predicted and ground truth boundary positions. Boundary confidence is calculated to adjust classification scores:

$$C_{\text{boundary}} = \frac{1}{n_{\text{points}}} \sum_{i=1}^{n_{\text{points}}} c_i \tag{18}$$

where c_i represents the confidence of the i-th boundary point, and n_{points} is the total number of boundary points.

To improve classification and localization performance, we design an improved loss function that incorporates Focal Loss for handling class imbalance and a boundary confidence term to enhance boundary localization accuracy. The overall optimization objective of the BFE module is given by:

$$\mathcal{L}_{\text{BFE}} = \mathcal{L}_{\text{cls}} + \lambda_1 \mathcal{L}_{\text{bin}} + \lambda_2 \mathcal{L}_{\text{reg}} \tag{19}$$

where $\mathcal{L}_{\text{cls}}$ is the improved classification loss, defined as:

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^{N} \alpha_i (1 - \hat{y}_i)^{\gamma} y_i \log(\hat{y}_i) + \lambda_3 (1 - C_{\text{boundary}}) \tag{20}$$

Here, α_i is a weighting factor for class imbalance, γ is the focusing parameter that reduces the weight of well-classified samples, and C_{boundary} is the boundary confidence term given by Eq. (18). The boundary bin estimation loss remains:

$$\mathcal{L}_{\text{bin}} = -\sum_{j=1}^{n_{\text{bins}}} t_j \log(p_j) \tag{21}$$

where t_j is the true label of bin j, and p_j is its predicted probability.

By integrating boundary information extraction and enhancement, the BFE module improves the model's perception of boundaries, thereby optimizing both classification and localization performance.

3.5 Theoretical Analysis

Given an input feature map of size $H \times W \times C$, the computational complexity of a conventional channel attention mechanism (e.g., CBAM) is $O(C^2)$. However, in our task-adaptive weighting approach, we employ a spatially adaptive formulation, reducing the complexity to:

$$O(HW + C) \tag{22}$$

thereby improving efficiency in high-resolution defect detection.

Furthermore, our boundary-aware convolution is optimized through a dynamic receptive field selection process. The convergence property is established by proving that the iterative weight refinement follows a contraction mapping with a Lipschitz constant $L < 1$, ensuring stability.

Definition: A function $f(x)$ is a contraction mapping if there exists a constant $0 \leq L < 1$ such that:

$$\|f(x) - f(y)\| \leq L\|x - y\| \tag{23}$$

for all x, y in the domain. Under this condition, the iterative process of weight refinement is guaranteed to converge to a unique fixed point, ensuring stable optimization.

4 Experimental Analysis

4.1 Experimental Setup

To investigate the speed and accuracy of RC2DNet in real-world cable surface defect detection, this study utilized a real dataset containing 4307 cable defect images (source: https://universe.roboflow.com/jkchen/yqmsb/dataset/2, accessed on 11 January 2025), with defects classified into three categories: "fracture (1500 samples)" "pitting (1200 samples)" and "wear (1607 samples)". To enhance the dataset, we performed several augmentation techniques, including random rotation, horizontal and vertical flipping, scaling, cropping, and color jittering. After data augmentation, 12,921 images were generated, which were then split into training, validation, and test sets in an 8:1:1 ratio.

RC2DNet is implemented using the PyTorch framework, which is known for its flexibility and ease of use in computer vision tasks. The training was conducted on a Windows 11 machine equipped with an NVIDIA GeForce RTX 4060 GPU and an AMD Ryzen 7 CPU. The model was trained using the Adam optimizer with a learning rate of 0.0001. The batch size was set to 16, and the training was carried out for 50 epochs, with early stopping applied based on validation accuracy to prevent overfitting. The original dataset and model detection results are shown in Fig. 2.

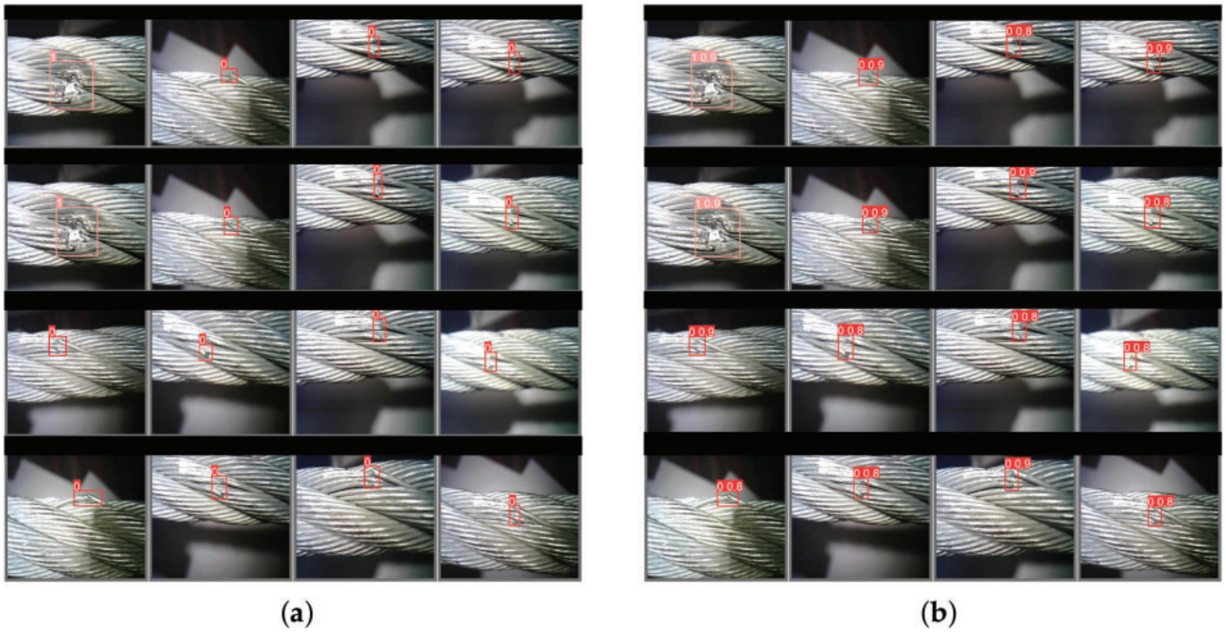


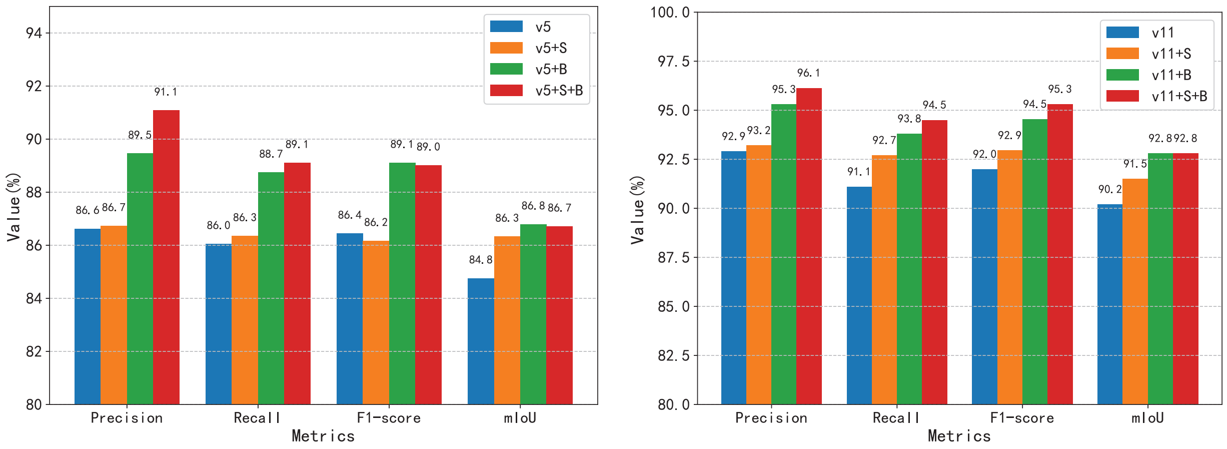
Figure 2: Original dataset and model detection results (a) Original dataset; (b) Model detection results

The evaluation metrics include precision, recall, F1-score, mean Intersection over Union (mIoU), and frame rate. Precision measures the proportion of actual positive samples among those predicted as positive, focusing on the accuracy of the detection results. Recall indicates the proportion of actual positive samples successfully detected, emphasizing the model's detection capability. The F1-score, as the harmonic mean of precision and recall, provides a comprehensive evaluation of the model's balance between accuracy and coverage. mIoU quantifies the model's localization accuracy by calculating the ratio of intersection to union between predicted and ground truth bounding boxes, with higher values indicating better overlap. To assess the model's video detection capability, frame rate was introduced as a metric, representing the number of frames processed per second, reflecting the model's real-time processing ability.

The comparison methods include the Image Preprocessing Scheme and Deep Convolutional Neural Network (IPDCNN) [21], the Optimized YOLOv5s (oYOLOv5s) [16], the Real-Time Detection Transformer (RT-DETR) [22], and the DETR with Improved Denoising Anchor Boxes (DINO) [23]. IPDCNN reduces cable jitter and background noise using image preprocessing and an improved LeNet-5-based CNN [24]. oYOLOv5s enhances feature extraction with a TRANS module and a more lightweight backbone. RT-DETR eliminates the need for NMS with an efficient Transformer-based architecture, improving speed and accuracy. DINO further optimizes DETR models with contrastive denoising training and improved anchor initialization.

4.2 Ablation Experiments

RC2DNet is an improvement on YOLOv11 (referred to as v11), so we need to explore the impact of the small target feature extraction module (denoted as S) and the boundary feature enhancement module (denoted as B) on performance. YOLOv5 (denoted as v5), as a baseline algorithm that has been extensively validated, is also tested to explore the effect of adding the S and B modules. The impact of the small target feature extraction module and boundary feature enhancement module on accuracy is shown in Fig. 3.



(a) (b)

Figure 3: Impact of the small target feature extraction module and boundary feature enhancement module on accuracy (a) YOLOv5 experimental results; (b) YOLOv11 experimental results

For the v5 series, the baseline model achieves a precision of 86.6%, recall of 86.0%, F1-score of 86.44%, and mIoU of 84.75%. Adding the S module slightly improves precision and recall but leads to a minor drop in F1-score (86.16%), suggesting that while feature fusion aids detection, it may introduce redundancy. The B module (v5 + B) significantly enhances localization, boosting precision to 89.4%, recall to 88.7%, and F1-score to 89.11%, with mIoU increasing to 86.79%. Combining both modules (v5 + S + B) further raises precision

(91.08%) and recall (89.10%), but F1-score (89.01%) shows a slight decline compared to v5 + B, indicating a trade-off between feature fusion and boundary refinement.

For the v11 series, improvements are more consistent. The baseline model achieves an F1-score of 91.99%, increasing to 92.95% with the S module and 94.54% with the B module, showing their positive impact. The best performance is observed with both modules (v11 + S + B), reaching an F1-score of 95.31%, confirming that the S module strengthens feature fusion while the B module refines boundary localization.

The ablation experiment on frame rate is shown in Table 1:

Table 1: Ablation experiment frame rate performance

	YOLOv5				**YOLOv11**			
	v5	**v5 + S**	**v5 + B**	**v5 + S + B**	**v11**	**v11 + S**	**v11 + B**	**v11 + S + B**
GPU	38.27	38.51	38.93	38.31	57.41	57.56	57.83	57.71
CPU	16.12	16.37	16.78	16.60	29.15	29.09	29.36	29.78

The frame rate comparison results show that the introduction of the modules has minimal impact on computational overhead, with only small fluctuations in frame rate, maintaining the efficiency of the model. In the GPU environment, the v5 series maintains a frame rate of around 38 fps, while the v11 series maintains around 57 fps, indicating that the addition of the S and B modules does not significantly increase the computational burden. In the CPU environment, the frame rate shows a similar trend, with the v5 series around 16 fps and the v11 series around 29 fps. Notably, the v11 + S + B combination achieves the highest frame rate in both GPU (57.71 fps) and CPU (29.78 fps) environments, reflecting RC2DNet's significant advantage in real-time processing. This shows that the improvements not only enhance detection accuracy and boundary localization but also meet the real-time processing requirements in practical applications.

4.3 Comparison Experiments

Fig. 4 shows the detection accuracy of different methods. On the Cable dataset, RC2DNet outperforms all other methods in terms of precision, recall, F1-score, and mIoU, achieving the highest values across all metrics, with precision at 96.13%, recall at 94.50%, F1-score at 95.31%, and mIoU at 92.80%. This superior performance can be attributed to the effectiveness of its small target feature extraction and boundary feature enhancement modules, which enhance both small target detection and boundary localization, particularly in complex cable defect scenarios. While RT-DETR and DINO also show strong performance, their precision and F1-scores are slightly lower than RC2DNet, particularly in terms of boundary accuracy. On the COCO dataset, RC2DNet still performs well but shows a slight decrease compared to its performance on the Cable dataset, with precision at 94.45% and F1-score at 92.97%. This drop is likely due to the greater diversity and complexity of the COCO dataset, as RT-DETR and DINO, which are designed for more general object detection tasks, perform better on this more varied dataset. Nonetheless, RC2DNet remains competitive, demonstrating its robust performance in small target detection and boundary localization. These results suggest that while RC2DNet excels in specialized tasks like cable defect detection, methods like RT-DETR and DINO are more adaptable and robust across a wider range of object detection challenges, highlighting the generalization gap between specialized and general-purpose models.

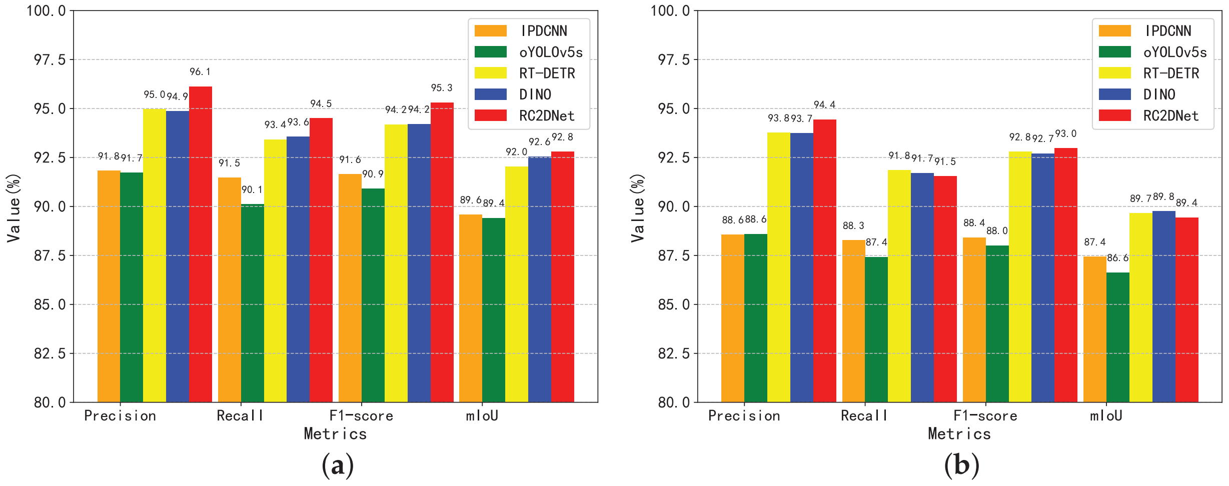


Figure 4: Detection accuracy of different methods (a) Cable Dataset; (b) COCO Dataset (Experiment on the COCO dataset was conducted to evaluate the generalization and robustness of the models)

The frame rate comparison results show that RC2DNet achieves significant real-time advantages in both GPU and CPU environments (see Table 2). On the GPU, IPDCNN achieves a frame rate of 29.72 fps, oYOLOv5s reaches 35.54 fps, RT-DETR achieves 56.12 fps, DINO reaches 57.03 fps, while RC2DNet achieves an impressive 57.71 fps, significantly outperforming the other methods. In the CPU environment, IPDCNN achieves 9.36 fps, oYOLOv5s reaches 14.18 fps, RT-DETR achieves 26.75 fps, DINO reaches 28.48 fps, and RC2DNet maintains a high 29.78 fps, demonstrating its superior efficiency for real-time detection tasks. These results indicate that RC2DNet, with its optimized module design, significantly improves processing speed while maintaining high accuracy, especially for real-time detection in both GPU and CPU environments.

Table 2: Frame rate performance of three methods

	IPDCNN	oYOLOv5s	RT-DETR	DINO	RC2DNet
GPU	29.72	35.54	56.12	57.03	57.71
CPU	9.36	14.18	26.75	28.48	29.78

5 Conclusion and Future Directions

With the continuous development of industrial automation and intelligent manufacturing, real-time cable defect detection plays a critical role in ensuring product quality and production safety. The Real-time Cable Defect Detection Network (RC2DNet), based on YOLOv11 improvements, incorporates a detailed small target feature extraction module and a boundary feature enhancement module, enabling efficient detection of small-sized cable defects in complex backgrounds. The introduction of feature pyramids and adaptive weighting mechanisms enhances the network's robustness across multiple scales, while the combination of boundary-aware convolution and an improved loss function effectively improves detection accuracy, particularly for small target recognition in extreme cases.

Although this study has made significant progress in terms of both accuracy and real-time performance, there is still room for further optimization. Future research may explore more adaptive detection methods for multiple scenarios and defect types to address more diverse industrial environments. Additionally, the diversity and richness of the dataset are crucial for enhancing the model's generalization ability, and future efforts can be made to collect and annotate more industrial field data to further improve the model's performance in practical applications.

Acknowledgement: The authors would like to express their gratitude to the editors and reviewers for their detailed review and insightful advice.

Funding Statement: This work was supported by the National Natural Science Foundation of China under Grant 62306128, the Basic Science Research Project of Jiangsu Provincial Department of Education under Grant 23KJD520003, and the Leading Innovation Project of Changzhou Science and Technology Bureau under Grant CQ20230072.

Author Contributions: Writing—original draft: Zilu Liu; Supervision: Hongjin Zhu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data will be made available on request.

Ethics Approval: Not applicable.

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

References

1. Xie J, Sun T, Zhang J, Ye L, Fan M, Zhu M. Research on cable defect recognition technology based on image contour detection. In: 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE); Zhuhai, China; 2021. p. 387–91.
2. Wu Y, Zhang P, Lu G. Detection and location of aged cable segment in underground power distribution system using deep learning approach. IEEE Trans Ind Inform. 2021 Nov;17(11):7379–89. doi:10.1109/TII.2021.3056993.
3. Chang SJ, Kwon G-Y. Anomaly detection for shielded cable including cable joint using a deep learning approach. IEEE Trans Instrum Meas. 2023;72:1–10. doi:10.1109/TIM.2023.3264025.
4. Said A, Hashima S, Fouda MM, Saad MH. Deep learning-based fault classification and location for underground power cable of nuclear facilities. IEEE Access. 2022;10(6):70126–42. doi:10.1109/ACCESS.2022.3187026.
5. Zhu Y, Zhou Q, Liu N, Xu Z, Ou Z, Mou X, et al. Scalekd: distilling scale-aware knowledge in small object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023 Jun 17–24; Vancouver, BC, Canada.
6. Kumar TS, Meena RS, Mani PK, Ramya S, Khandan KL, Mohammed A, et al. Deep learning based fault detection in power transmission lines. In: 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA); Coimbatore, India; 2022. p. 861–7.
7. Shehzadi T, Hashmi KA, Stricker D, Afzal MZ. Sparse semi-DETR: sparse learnable queries for semi-supervised object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2024 Jun 16–22; Seattle, WA, USA. p. 5840–50.
8. Dong F, Hou B, Liu Y, Wu YD. Research on quality inspection of cable conductor based on machine vision. J Electron Meas Instrum. 2020;34(6):144–53. (In Chinese).
9. Qiao X, Wang H, Qi C, Li XQ. Design and algorithm research of cable surface defect detection system based on machine vision. Mach Tool Hydraul. 2020;48(5):49–53. (In Chinese).
10. Chen T, Dong Y, Zhang S, Lv C, Zhang LH, Liao CH. Research on ACFM detection method and detection system for weld defects of aluminum sheath of high voltage cable. Chin J Eng Des. 2022;29(3):394–400. (In Chinese).
11. Redmon J. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA.
12. Chen D, Jin Y. AWUCD-Net: the armored wire umbilical cable surface defect detection algorithm based on improved YOLOv7. IEEE Access. 2024;12(4):167559–74. doi:10.1109/ACCESS.2024.3492212.

13. Zhang H, He J, Jiang X, Gong Y, Hu T, Jiang T, et al. Quantitative characterization of surface defects on bridge cable based on improved YOLACT++. Case Stud Constr Mater. 2024;21(4):e03953. doi:10.1016/j.cscm.2024.e03953.
14. Misra D. Mish: a self regularized non-monotonic activation function. arXiv:1908.08681. 2019.
15. Wen Y, Chen J, Dong L, He R, Wang X, Tian M. Surface defect detection of cable based on improved YOLO V3. In: 2022 28th International Conference on Mechatronics and Machine Vision in Practice (M2VIP); Nanjing, China; 2022. p. 1–6.
16. Wang PF, Li YT, Huang YY, Zhu WK, Lin J, Wang BR. Defects detection for cable surface of cable-stayed bridge based on improved YOLOv5s network. Opto-Electron Eng. 2024;51(5):1, 9–20. (In Chinese).
17. Xu F, Kalantari M, Li B, Wang X. Nondestructive testing of bridge stay cable surface defects based on computer vision. Comput Mater Contin. 2023;75(1):2209–26. doi:10.32604/cmc.2023.027102.
18. Wu Z, Zhang Z, Xu J, Ling Q. A deep learning-based surface defect detection method with coarse granularity for corrugated fuel tubes. In: 2022 41st Chinese Control Conference (CCC); Hefei, China; 2022. p. 6725–30.
19. Liu Q, He D, Shen Y, Lao Z, Ma R, Li J. Surface defect detection of stay cable sheath based on autoencoder and auxiliary anomaly location. Adv Eng Inform. 2024;62(Part B):102759.
20. Tong X, Zhang X, Liu G, Li C, Zhao J. Research on surface defect detection technology of long-distance and long-span fast cable. In: 2023 9th International Conference on Mechatronics and Robotics Engineering (ICMRE); Shenzhen, China; 2023. p. 225–8.
21. Zhou P, Zhou G, Wang H, Wang D, He Z. Automatic detection of industrial wire rope surface damage using deep learning-based visual perception technology. IEEE Trans Instrum Meas. 2020;70:1–11.
22. Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang QQ, et al. Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and pattern Recognition; 2024 Jun 16–22; Seattle, WA, USA.
23. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, et al. Dino: detr with improved denoising anchor boxes for end-to-end object detection. arXiv:2203.03605. 2022.
24. Wu ML, Zhu Y, Li XM, Zhang JG. Fault diagnosis of rotor system based on attention mechanism and Lenet5 network. Mech Des Res. 2024;40(2):113–6. (In Chinese).