

CIA-YOLO: an improved steel cable defect detection model based on YOLOv11

Zhoujie Hu, Jiayan Chen , and Changjing Sun*

College of Quality and Standardization, China Jiliang University, Hangzhou 310018, PR China

Received: 16 June 2025 / Accepted: 18 September 2025

Abstract. To tackle small-scale features and blurred boundaries in steel cable defect detection, we propose CIA-YOLO, an enhanced YOLOv11-based model for high-precision industrial inspection. CIA-YOLO integrates three improvements: (1) a Convolutional Block Attention Module (CBAM) combining channel and spatial attention for finer feature extraction; (2) a dynamic-scaling Inner-IoU loss function enhancing robustness and localization accuracy; and (3) an optimized Adaptive Kernel Convolution (AKConv) with a refined C3k2 module for stronger multi-scale modeling. On a dataset of broken wire, corrosion, and wear defects, CIA-YOLO achieved mAP@0.5 of 88.5%, 97.7%, and 99.5%, and Recall of 88.4%, 96.4%, and 99.8%, respectively. Overall, it recorded a mAP@0.5 of 95.2% and Recall of 94.8%, notably with 99.8% Recall on small wear defects. Compared to baseline YOLOv11 variants, CIA-YOLO delivers superior accuracy and faster inference, enabling real-time, in-line quality monitoring and safety assurance in engineering settings.

Keywords: Steel cable / defect detection / YOLOv11 / CIA-YOLO / small-scale defects

1 Introduction

Steel cables are commonly used as important load-bearing components in engineering projects such as hoisting machinery, bridges, and cableways. Their operational condition is directly related to the safety and reliability of the equipment. Over time, under long-term loads and harsh environmental conditions, steel cables are prone to corrosion, wear, and fatigue. These problems often lead to typical defects such as broken wires, surface abrasion, and structural deformation [1,2]. If not detected and addressed in time, such defects may result in cable breakage and serious safety accidents. Therefore, proposing a visual inspection model for wire rope defects holds practical significance for ensuring the operational safety of engineering systems [3].

Currently, common detection methods for steel cables include electromagnetic testing, acoustic emission [4], eddy current detection [5], and vision-based image inspection. Although electromagnetic methods are increasingly adopted in industrial scenarios, their accuracy, cost, and equipment reliability still limit their broader application. Manual visual inspection remains widely used but suffers from low efficiency, strong subjectivity, and poor consistency, which cannot meet the demand for intelligent and real-time detection in modern industrial environments [6].

In recent years, machine vision and deep learning technologies have developed rapidly. Visual inspection methods based on image recognition have demonstrated significant advantages in industrial defect detection [7,8]. Among them, the YOLO (You Only Look Once) series, as a representative single-stage object detection algorithm, has been widely applied to defect detection in industrial products and weld seams due to its end-to-end training, fast inference, and lightweight structure [9–13].

Although the YOLO series has performed well in industrial detection, it still faces limitations in detecting small-scale defects, blurred boundaries, and complex structures of steel cables, especially in UAV images [14]. These issues include insufficient feature extraction, limited detection accuracy, and high sensitivity to low-quality anchor boxes. To address these challenges, this paper proposes an improved visual detection model for steel cable defects, named CIA-YOLO, based on YOLOv11. The model aims to improve accuracy and robustness in detecting small-scale features and blurred boundaries.

To address the limitations of fixed sampling positions in standard convolutions, we introduce an improved Adaptive Kernel Convolution (AKConv) module. This module allows dynamic adjustment of the sampling positions and kernel sizes based on the local content of the image. By learning offsets and multi-point sampling, AKConv adapts to irregular defect structures, enhancing the model's ability to detect small or deformed defects with greater precision. Offering greater flexibility and representation capacity

* Corresponding author: scj@cjlu.edu.cn

compared to traditional and deformable convolutions. Additionally, to meet the requirements of small object localization, a modified Inner-IoU bounding box loss function is developed. It incorporates a dynamic scaling factor to suppress the influence of low-quality predicted boxes during training [15]. Moreover, an enhanced CBAM (Convolutional Block Attention Module) is integrated to further improve the model's feature focusing ability in complex scenes. To fully exploit the capabilities of AKConv, the original C3 module in YOLOv11 is redesigned as the C3k2_AKConv module, enhancing the model's adaptability to multi-scale feature representation [16].

The main contributions of this work are summarized as follows:

- To address the difficulty in capturing features of small defect regions, an enhanced CBAM module is introduced. By incorporating standard deviation pooling and residual connections, the model's attention focusing capability on key areas is effectively improved.
- To solve the challenges posed by complex defect shapes and inaccurate localization in steel cables, a modified Inner-IoU loss function is designed. It integrates a dynamic focusing mechanism and an auxiliary bounding box control factor to improve the model's fitting ability and training stability across diverse targets.
- To overcome the limitations of fixed sampling positions and restricted feature representation in traditional convolution, an improved AKConv module and its integrated C3k2_AKConv structure are proposed. These enhancements break spatial constraints of convolution kernels and significantly enhance the model's structural flexibility and multi-scale feature extraction capability.

2 Related work

In recent years, with the advancement of industrial manufacturing and inspection automation, deep learning-based defect detection methods have achieved significant progress in scenarios such as steel cables and metal surfaces. Artificial intelligence (AI) has played a crucial role in these advancements, significantly improving the accuracy, speed, and robustness of defect detection systems in industrial applications [17]. In particular, object detection algorithms in the YOLO (You Only Look Once) series have become important tools for steel cable defect detection due to their high efficiency and real-time performance. Meanwhile, R-CNN and its variants (e.g., Fast R-CNN, Faster R-CNN and Mask R-CNN) are also widely used in industrial image applications for fine-grained target recognition, owing to their superior detection accuracy [18,19].

AI-based methods, particularly deep learning, have demonstrated exceptional capabilities in processing complex image data, enabling more precise defect detection. Among them, YOLO follows a single-stage detection framework, performing localization and classification in a single forward pass. This makes it suitable for real-time applications. In contrast, R-CNN is a

two-stage method that first generates region proposals and then conducts classification and regression [19]. Although it generally achieves higher accuracy, its slower inference speed makes it more suitable for offline or static detection tasks. Both frameworks have their own strengths, and selecting between them depends on the trade-off between detection speed and accuracy required by specific applications.

Li et al. [20] proposed a steel broken wire detection method based on Faster R-CNN. They constructed a dataset of typical broken-wire images, extracted features, and trained the model to achieve high detection accuracy. Leveraging the advantages of the two-stage structure, the method maintained precision while improving recognition under complex backgrounds [11]. However, due to its relatively slow inference, Faster R-CNN is not ideal for scenarios with real-time requirements. To detect ultra-small defects in braided steel hoses, Ying et al. [21] introduced an improved YOLOv5s model. They first applied the K-means++ algorithm to optimize anchor boxes. Then, Focal Loss was incorporated into the loss function to address sample imbalance. Additionally, an efficient channel attention (ECA) mechanism was embedded in the back-bone and feature fusion layers. A specialized small-object detection head was added at the final stage. The model significantly improved the recall and precision for detecting missing wires, stacked wires, and loose wires. Experiments on a braided steel hose dataset showed that the model achieved 92.2% mAP@0.5 with an inference speed of 23 FPS, balancing accuracy and real-time performance [21]. Furthermore, Fu et al. [22] proposed the CBG-YOLOv5s model for detecting metal surface corrosion in marine environments. Based on YOLOv5s, the model integrated a C3CBAM attention module and a lightweight C3Ghost module. It also included an additional detection layer for small targets. A dataset containing 6,000 corrosion images was constructed for evaluation. Experimental results showed that the model outperformed standard YOLOv5s and other common object detection algorithms in both accuracy and inference speed, particularly in complex scenarios involving small, low-contrast, and irregular corrosion regions [22].

In summary, deep learning-based methods for steel cable and metal surface defect detection are evolving toward higher accuracy, lighter models, and better multi-scale target modeling [18]. Among them, the YOLO series remains a key framework for structure optimization, attention mechanism integration, and multi-scale detection enhancement due to its flexibility and scalability. However, existing YOLO-based models still struggle with robust recognition of small targets and blurred defect regions [23,24].

To address these limitations, this paper proposes CIA-YOLO, an improved model that builds upon YOLOv11. By incorporating an enhanced CBAM attention mechanism, a redesigned Inner-IoU loss function, and an improved AKConv module, CIA-YOLO aims to maintain detection efficiency while significantly enhancing the model's robustness and accuracy in detecting small-scale and ambiguous defects.

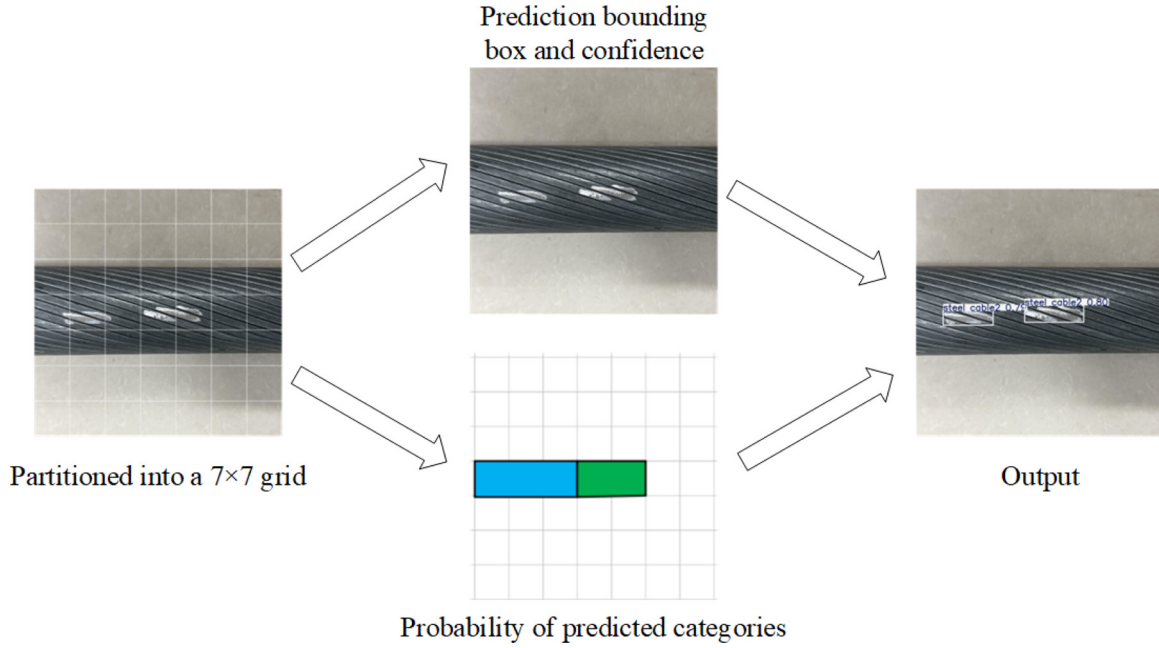


Fig. 1. Defect detection principle of YOLO.

3 Model optimization

3.1 Principle of the YOLO detection network

YOLO treats object detection as a regression problem. By feeding the entire image into the network, it predicts both the locations and the categories of the objects present in the image. After inputting the image [25], the network performs a series of computations through convolutional layers, pooling layers, and fully connected layers. These operations extract features, reduce spatial resolution, and generate predictions for object categories and locations.

The final output of the network is a tensor with dimensions $S \times S \times (B \times 5 + C)$, where S is the number of grid divisions on the image, B is the number of bounding boxes predicted per grid cell, and 5 represents the five parameters of each bounding box. These include the center coordinates (x , y), width (w), height (h), and the confidence score. C denotes the number of object classes to be predicted. The principle of defect detection using YOLO is illustrated in Figure 1.

The input image is divided into a 7×7 grid. Each grid cell is responsible for predicting two bounding boxes. Each bounding box contains five values. As a result, the network outputs a tensor of size $7 \times 7 \times (2 \times 5 + 2)$ [6]. The confidence score reflects whether an object exists in the predicted grid cell and how accurate the predicted bounding box is. It is defined as follows.

$$\text{confidence} = P_r(\text{Object}) * IoU_{truth}^{pred} \quad (1)$$

In the formula, $P_r(\text{Object})$ represents the probability that an object appears in the predicted grid cell, with a value of either 0 or 1. IoU_{pred}^{truth} denotes the Intersection over

Union between the predicted bounding box and the ground truth box. Therefore, the confidence score equals 1 when an object is present, and 0 when there is none. In addition to the bounding box center coordinates, width, height, and confidence score, each grid cell must also predict a vector of length C , where each element represents the conditional probability $P_r(\text{Class}_i/\text{Object})$, which indicate the likelihood of the object being of class C_i given that an object exists in the cell [26]. During the inference phase, the output probability P is calculated as the product of the conditional class probability $P_r(\text{Class}_i/\text{Object})$ and the confidence score $P_r(\text{Object})$, as defined by the following formula:

$$\begin{aligned} P &= P_r(\text{Class}_i/\text{Object}) * P_r(\text{Object}) * IoU_{pred}^{truth} \\ &= P_r(\text{Class}_i) * IoU_{pred}^{truth} \end{aligned} \quad (2)$$

To predict the bounding box, confidence score, and object category, the network loss is composed of three parts. The corresponding formula is as follows:

$$\text{Loss} = \text{Loss}_{bbox} + \text{Loss}_{confidence} + \text{Loss}_{class} \quad (3)$$

The term Loss_{bbox} represents the loss associated with the bounding box, $\text{Loss}_{confidence}$ corresponds to the confidence score loss, and Loss_{class} denotes the classification loss. In the formula, x , y , w , h , c and $P(c)$ represent the ground truth or predicted parameters of the bounding box, confidence level, and class probability for the target object [27]. The indicator variable I is used to determine whether the corresponding bounding box contains an object, taking a value of either 0 or 1. The hyperparameter λ is used to control the contribution of the bounding box prediction loss and to prevent overfitting due to large gradient

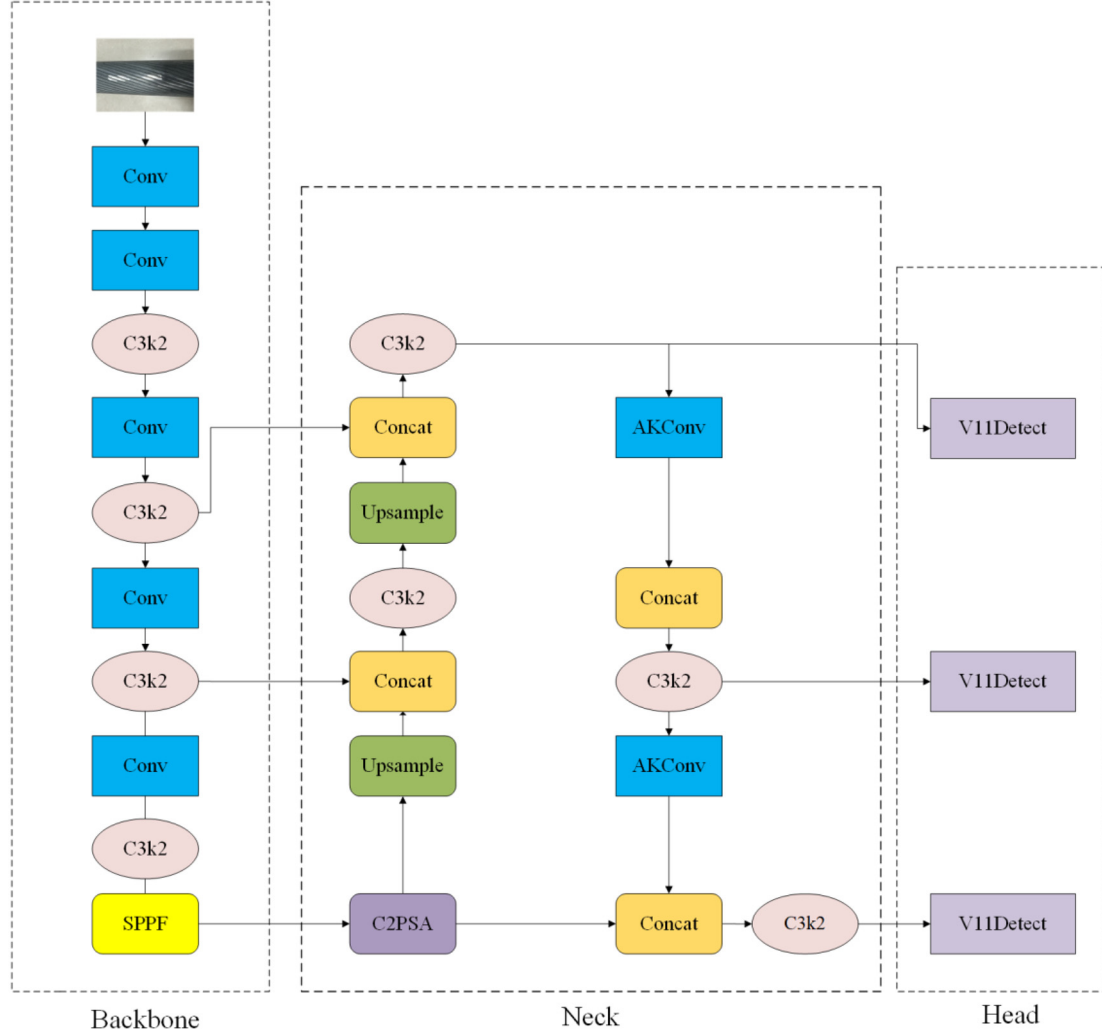


Fig. 2. Network architecture of YOLOv11.

magnitudes [28]. The formula is shown as follows:

$$Loss_{bbox} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[(x_i - \bar{x}_j)^2 + (y_i - \bar{y}_j)^2 \right] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{w_j})^2 + (\sqrt{h_i} - \sqrt{h_j})^2 \right] \quad (4)$$

$$Loss_{confidence} = \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} (c_i - \hat{c}_j)^2 + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{noobj} (c_i - \hat{c}_j)^2 \quad (5)$$

$$Loss_{class} = \sum_{i=0}^{S^2} I_i^{obj} \sum_{c \in class} \left[P_i(c) - \hat{P}_i(c) \right]^2. \quad (6)$$

3.2 Improvements to YOLOv11

To visually underscore our innovations, Figure 2 first depicts the original YOLOv11 architecture, followed by Figure 3, which illustrates the enhanced CIA-YOLO model. Compared with the baseline, CIA-YOLO refines three key components: it integrates an improved CBAM attention mechanism to heighten responsiveness to fine-scale defect regions; replaces the standard IoU loss with an advanced Inner-IoU regression loss to significantly boost bounding-box localization accuracy and accelerate convergence; and employs an adaptive AKConv convolutional block to strengthen multi-scale feature representation. Extensive experiments demonstrate that these enhancements not only markedly increase mAP@0.5 and recall for steel-cable defect detection but also substantially improve the model's generalization across diverse, challenging operating conditions [14,29].

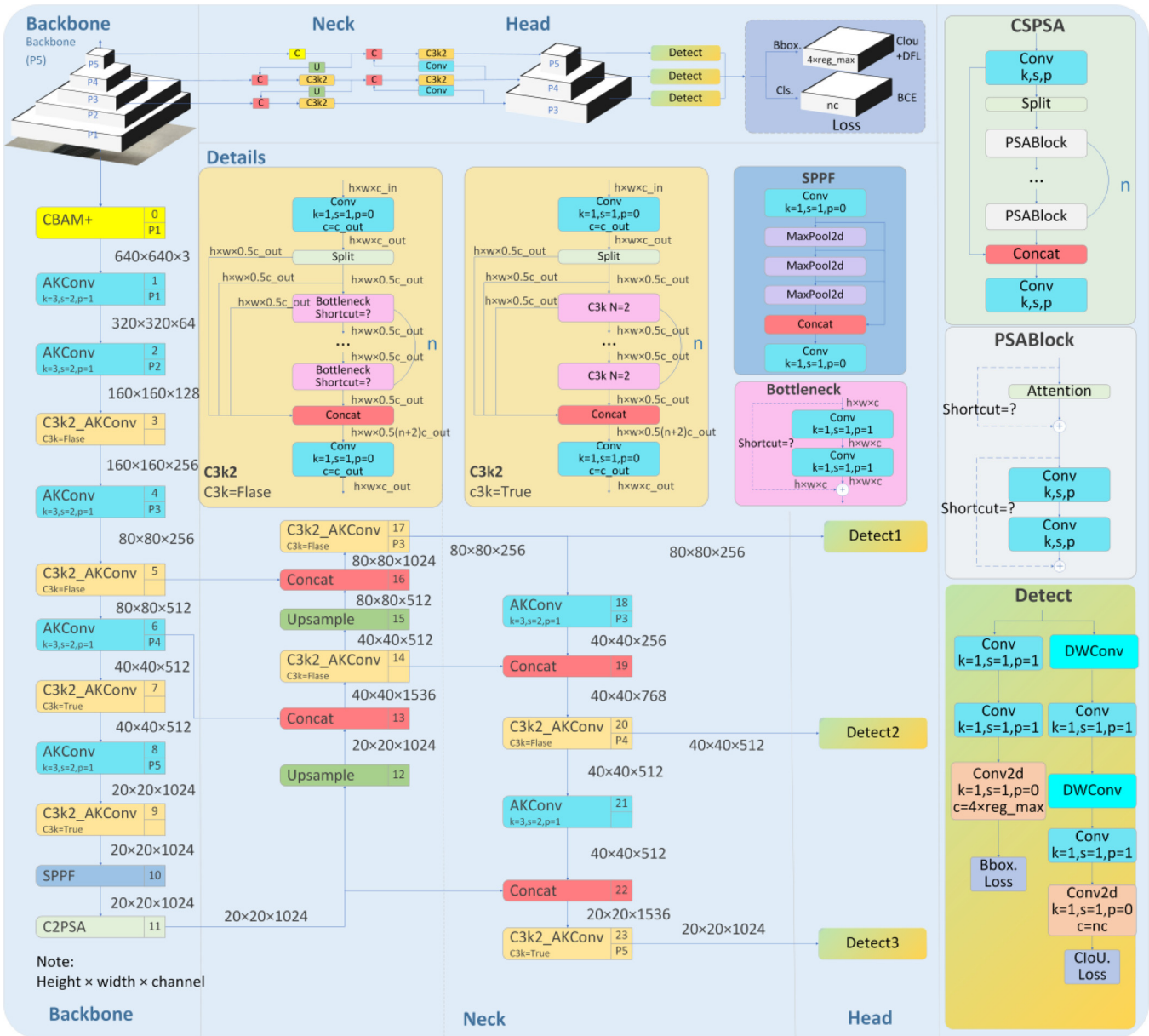


Fig. 3. Network architecture of CIA-YOLO.

3.2.1 Enhanced CBAM attention module

This model incorporates an enhanced CBAM (Convolutional Block Attention Module), designed to more reliably highlight subtle defect patterns and suppress background noise in wire rope inspection. Compared to the original CBAM [30], the enhanced module offers stronger feature representation and greater flexibility, as shown in Figure 4.

To improve the responsiveness of the CBAM attention mechanism to small-scale defect regions, this study introduces an enhanced CBAM module with the following three key improvements:

- In the channel attention branch, standard deviation pooling is incorporated to capture the variance across different channels.

- A three-branch shared MLP (Multi-Layer Perceptron) is employed, and a weighted fusion mechanism is used to adaptively balance the contributions of the three pooling types.
- In the spatial attention branch, a standard deviation mAP@0.5 along the spatial dimension is added to strengthen the response to boundary variation regions. In addition, a residual connection is introduced at the output to preserve essential backbone features.

Experimental results show that the enhanced CBAM better focuses on tiny defects. Figure 5 compares Grad-CAM activations: (a) the original steel cable image; (b) the standard CBAM Grad-CAM, which roughly identifies defect regions but responds diffusely and with low contrast

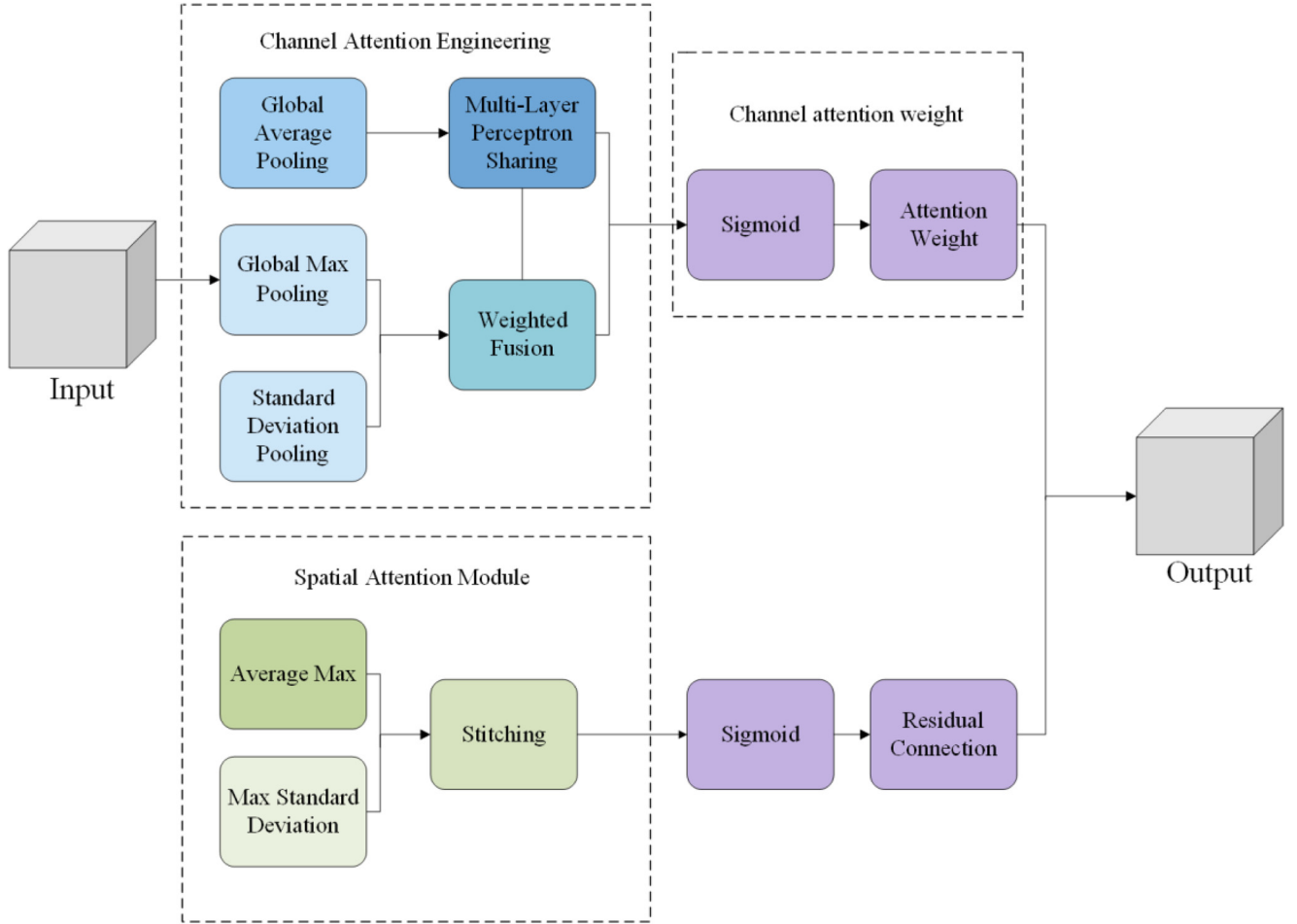


Fig. 4. Architecture of the enhanced CBAM module.

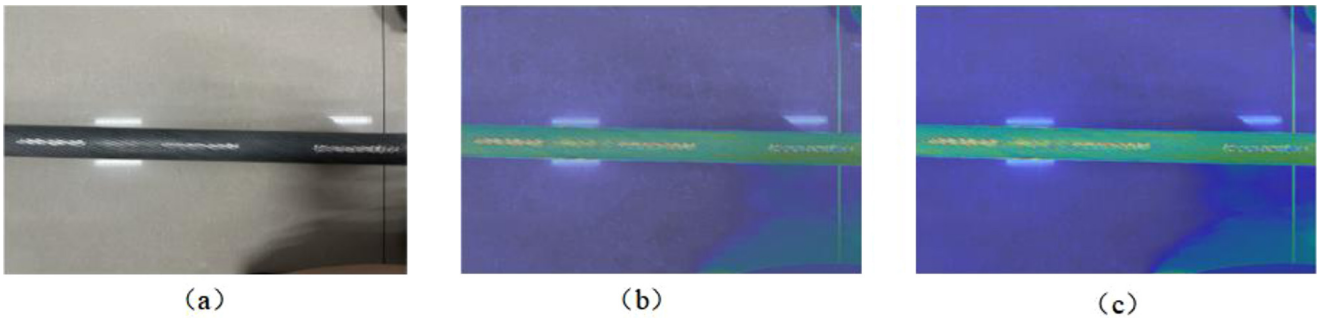


Fig. 5. Comparison of Grad-CAM Activations: (a) Original Steel Cable Image; (b) Standard CBAM Grad-CAM; (c) Enhanced CBAM Grad-CAM.

to fine wire fractures; (c) the enhanced CBAM Grad-CAM, where the heatmAP@0.5 is clearer and tightly focused on small-scale fractures and corrosion spots. These findings confirm that our improvements significantly enhance the precise visualization of small defects.

3.2.2 Improved inner-IoU bounding box loss function

CIA-YOLO incorporates a series of IoU-based loss functions and enhances the existing CIoU and DIoU losses used in YOLOv11.

1. Design of the Inner-IoU Sub-Loss Module

1) EIoU (Efficient IoU): Based on CIoU, EIoU further reinforces consistency in aspect ratios. The corresponding formula is as follows:

$$L_{EIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho_w^2}{c_w^2} + \frac{\rho_h^2}{c_h^2}. \quad (7)$$

In this equation, ρ^2 represents the squared Euclidean distance between the center points of the predicted box and the ground truth box. The term c^2 denotes the squared length of the diagonal of the smallest enclosing box.

Additionally, ρ_w^2 and ρ_h^2 indicate the squared differences in width and height, respectively, while c_w and c_h represent the width and height of the minimum enclosing rectangle.

2) SIOU (Scylla IoU): Introduces three types of constraints angle, shape, and distance to enhance the gradient representation of the loss function and improve its convergence speed.

$$L_{SIOU} = distance_{cost} + shape_{cost} + \varepsilon. \quad (8)$$

The term $distance_cost$ represents the penalty for the distance between the center points of the predicted and ground truth bounding boxes, while $shape_cost$ reflects the cost associated with differences in aspect ratios (width and height). A small constant is introduced to prevent division by zero.

3) WIoU (Wise IoU): This loss introduces a dynamic, non-monotonic focusing factor based on outlier awareness, which helps reduce the influence of low-quality predictions on the overall loss. As a key component of the Inner-IoU framework, the WIoU mechanism assigns smaller weights to poor anchor boxes, enabling the model to focus more on high-quality predictions during training.

$$L_{WIoU} = scaled_{loss} = \omega \cdot (1 - IoU) \quad (9)$$

$$\omega = \frac{1}{1 + e^{-\alpha(\beta-\delta)}}. \quad (10)$$

Here, β represents the anomaly degree of the anchor box, while α and δ are hyperparameters.

4) The total loss function L_{total} of the improved Inner-IoU is defined as follows:

$$L_{total} = \lambda_1 \cdot (1 - EIoU) + \lambda_2 \cdot (1 - SiIoU) + \lambda_3 \cdot (1 - WIoU_{scaled}) \quad (11)$$

$$WIoU_{scaled} = IoU_{inner} \cdot scale_{factor}. \quad (12)$$

Specifically, $\lambda_1 = 0.4$, $\lambda_2 = 0.3$, $\lambda_3 = 0.3$. The $scale_{factor} \in (0, 1)$, is generated by a dynamic focusing function (monotonic or non-monotonic). If the current IoU is lower than the global average IoU, the $scale_{factor}$ will decrease, resulting in a lower $WIoU_{scaled}$ and thus an increase in the overall loss.

2. Improvements to the Original Loss Functions

1) CIOU is further improved by adding a width–height consistency constraint.

2) DIOU minimizes the Euclidean distance between the centers of the predicted and ground-truth boxes. This improves the accuracy of regression.

3 Summary of Advantages

Inner-IoU provides more types of loss functions. It offers better flexibility and task adaptability. The dynamic focusing mechanism reduces the influence of low-quality anchor boxes. As a result, the model achieves faster convergence and more accurate bounding box fitting across different datasets [31].

3.2.3 Improved AKConv

1. Newly added loss functions

1) Overview

AKConv (Adaptive Kernel Convolution) is an adaptive sampling convolution module inspired by the concepts of Deformable Convolution and Dynamic Filter. Its main purpose is to dynamically adjust the sampling positions during the convolution process. This allows the network to better adapt to targets with deformations or irregular structures, such as steel cable defects. Figure 6 illustrates the structural diagram of the AKConv module.

2) Innovation

Compared to standard convolution with fixed sampling positions, AKConv uses dynamic position awareness, learnable offsets, and multi-point sampling fusion. It is suitable for tasks with blurred object boundaries or significant local structural variations. Internally, a global hook is used to dynamically scale down the gradient of preventing gradient explosion.

2. C3k2_AKConv Module

1) Structure

The C3k2_AKConv module inherits the CSP from YOLOv5. It includes two Conv1×1 layers for compression and fusion, followed by n backbone modules, which can either be Bottleneck or C3k. After concatenating the outputs, a Conv1×1 layer is applied for further processing. This design enhances the flexibility and adaptability of the network in processing multi-scale features effectively.

2) Innovation

The C3k2_AKConv module provides flexibility by allowing each sub-module to be replaced with either a Bottleneck (with AKConv), which represents a standard residual structure, or a C3k (with AKConv), which is a deeper CSP-based structure that enhances the fusion of local features. This modular design improves the model's adaptability and feature extraction capability, making it more suitable for handling complex tasks.

3. Summary of Advantages

The improved AKConv enables dynamic sampling through learnable offsets. It fuses multiple sampling points to enhance the model's perception of object boundaries and local structural variations. Compared with traditional convolution, it offers greater adaptability and feature representation capacity. It is especially effective for detecting small or deformable objects.

3.2.4 Summary

Based on the YOLOv11 framework, the proposed CIA-YOLO model introduces three major improvements: an enhanced CBAM attention mechanism, an improved Inner-IoU bounding box loss function, and the AKConv convolution module. These enhancements significantly improve the detection of small-scale defects and the accuracy of bounding box regression. The enhanced CBAM increases the model's ability to focus on fine-grained regions. The improved Inner-IoU loss provides a more flexible and robust loss formulation, reducing the negative influence of low-quality anchor boxes. The refined AKConv module enhances the model's ability to capture local

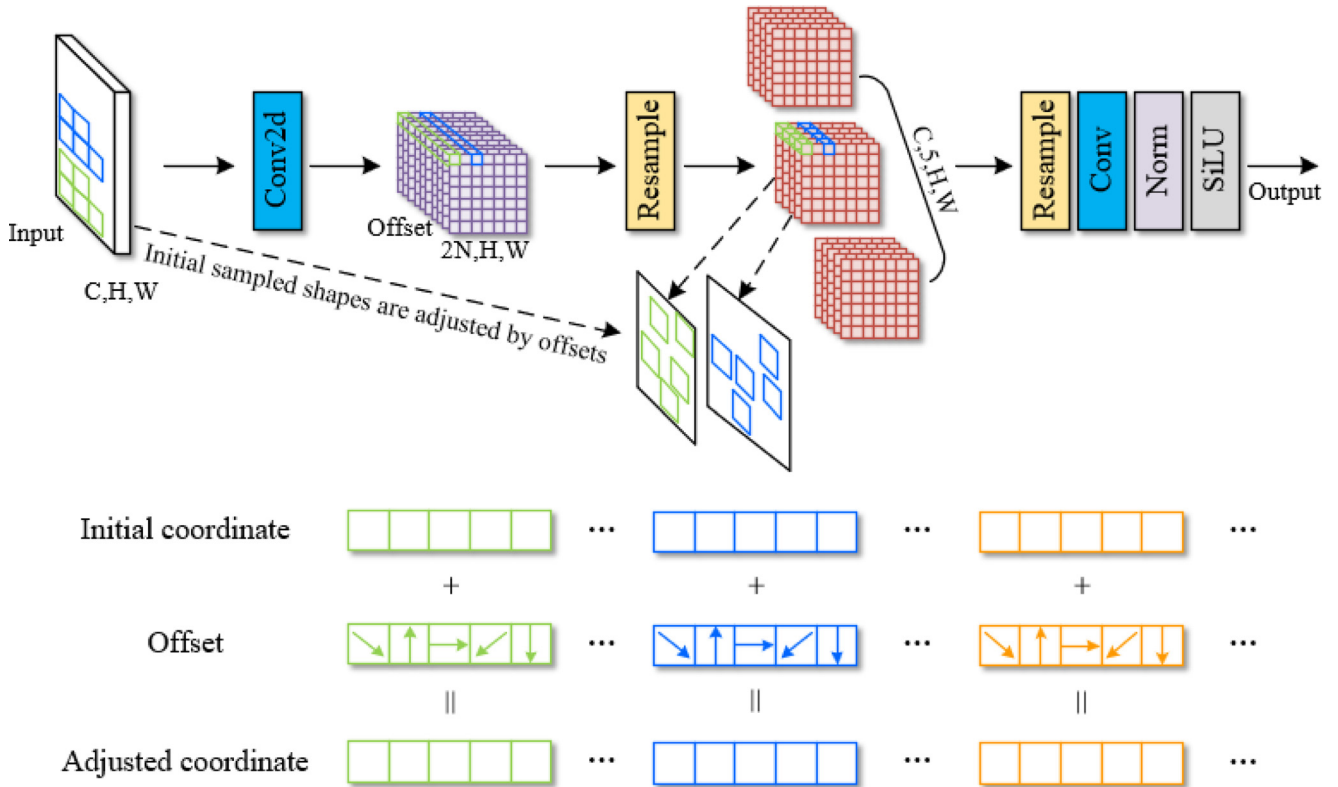


Fig. 6. Structural diagram of the AKConv module.



Fig. 7. On-site experimental images: (a) Data collection site; (b) UAV physical device.

structural variations through dynamic sampling. In summary, the proposed enhancements significantly improve CIA-YOLO in terms of accuracy, robustness, and generalization. The model is particularly well-suited for visual detection tasks involving small-scale and blurred defects in steel cables [32].

4 Experiments and results

4.1 Dataset

4.1.1 Dataset construction and preprocessing

Original steel-cable images were captured in RAW format using a gimbal-stabilized UAV flying pre-programmed flight paths at multiple angles and altitudes, with camera settings fixed at ISO 200 and a $1/200$ s shutter speed to

preserve clarity and detail. All RAW files were transferred via wireless link or storage card to a local server, where a Laplacian variance filter automatically discarded frames with motion blur or low contrast. Remaining images were further screened by viewing angle and lighting conditions to retain only those that clearly depict wear defects. In total, we assembled 2,159 color images at 640×640 resolution covering three defect types: 842 wear images collected on-site by UAV (private dataset) and 1,317 broken-wire and corrosion images sourced from public datasets. This dataset spans diverse lighting, viewing angles, occlusions, and complex backgrounds, thereby closely reflecting real-world scenarios and enhancing the model's generalization capability Figure 7. On-site experimental setup, and Figure 8 presents relevant defect images from the dataset.

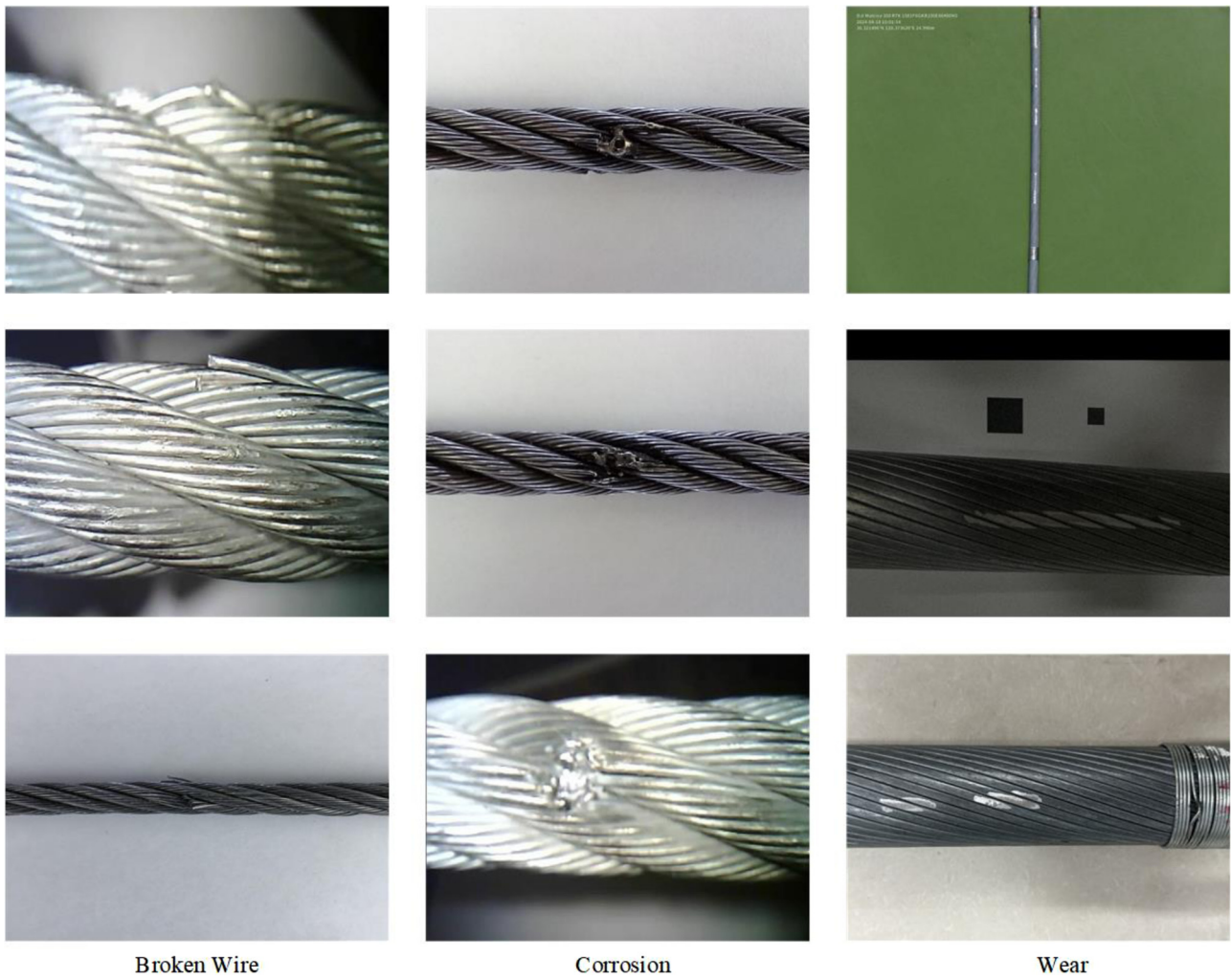


Fig. 8. Relevant defect images from the dataset.



Fig. 9. Schematic of LabelImg annotation.

The image annotation process was completed using the LabelImg tool (LabelImg is an open-source graphical image annotation tool used for labeling object detection datasets, allowing users to efficiently create bounding box annotations in images.). Figure 9 shows the schematic of LabelImg annotation. The annotation results were initially saved in

XML format and later converted to TXT format required by the YOLO framework before training. To facilitate training and classification, the three types of steel cable defects were labeled as follows: broken wires as steel cable0, corrosion as steel cable1, and wear as steel cable2. In total, 2,159 annotated defect images were collected. The dataset

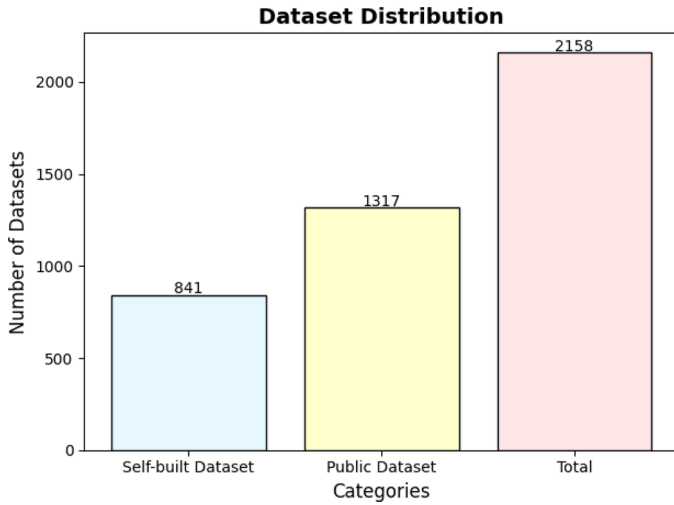


Fig. 10. Distribution of datasets across self-built and public categories.

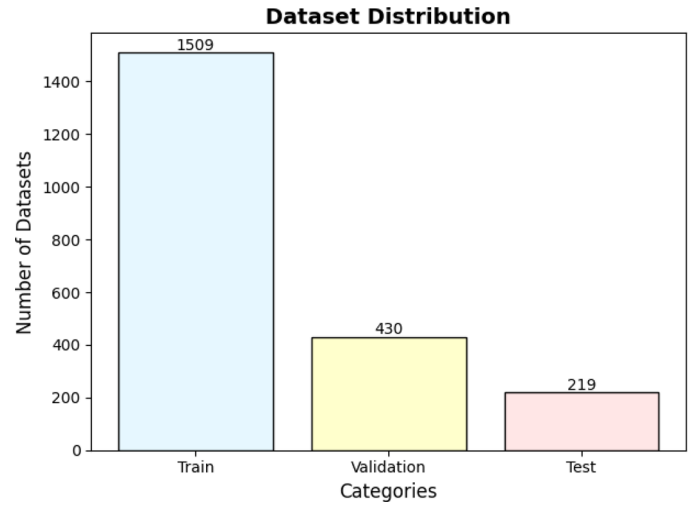
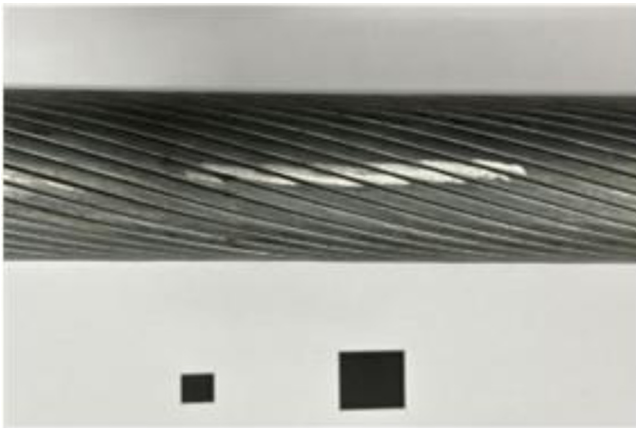
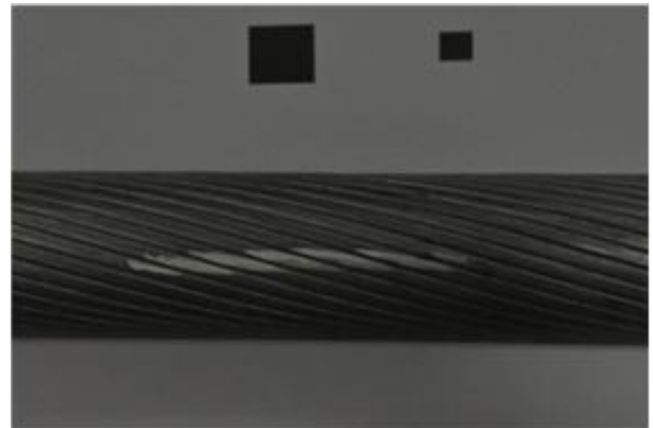


Fig. 11. Dataset distribution across train, validation, and test categories.



(a)



(b)

Fig. 12. Images of steel cable defects before and after transformation: (a) Image before transformation; (b) Image after transformation.

was divided into training, validation, and test sets in a 7:2:1 ratio [33]. Figure 10 shows the distribution of self-built datasets and public datasets, and Figure 11 illustrates the distribution of datasets across the training, validation, and test sets.

Additionally, in the object detection community, the COCO dataset defines “Small” objects as those with a pixel area below 32×32 (i.e. <1024 pixels) [34]. To conform to this standard, we classify any defect in our 640×640 steel cable images whose width or height is less than 32 pixels as a small-scale defect. Specifically, wire fractures span 3–10 pixels in width, corrosion measure 5–15 pixels in diameter, and wear regions range from 4–12 pixels in width—each yielding an area well below 1024 pixels. In our dataset, the average side length of these defects is approximately 8 pixels (area ≈ 64 pixels), and all subsequent performance evaluations for small-scale defects are based on this definition.

4.1.2 Data augmentation strategy

To improve detection accuracy and enhance the model’s robustness in complex environments, various data augmentation techniques were applied to expand the original dataset. These included random horizontal flipping, scaling, brightness adjustment, and color space perturbation [18]. Such operations helped strengthen the model’s ability to adapt to diverse lighting conditions and defect patterns. Figure 12 shows the images of steel cable defects before and after transformation.

In addition, this paper adopts Mosaic data augmentation. This method combines four different training images into one, creating samples with mixed features such as multiple targets, backgrounds, and scales. It significantly improves the model’s ability to detect small defects, such as minor wire breaks and slight corrosion. It also enhances generalization in complex scenes. Figure 13 shows an

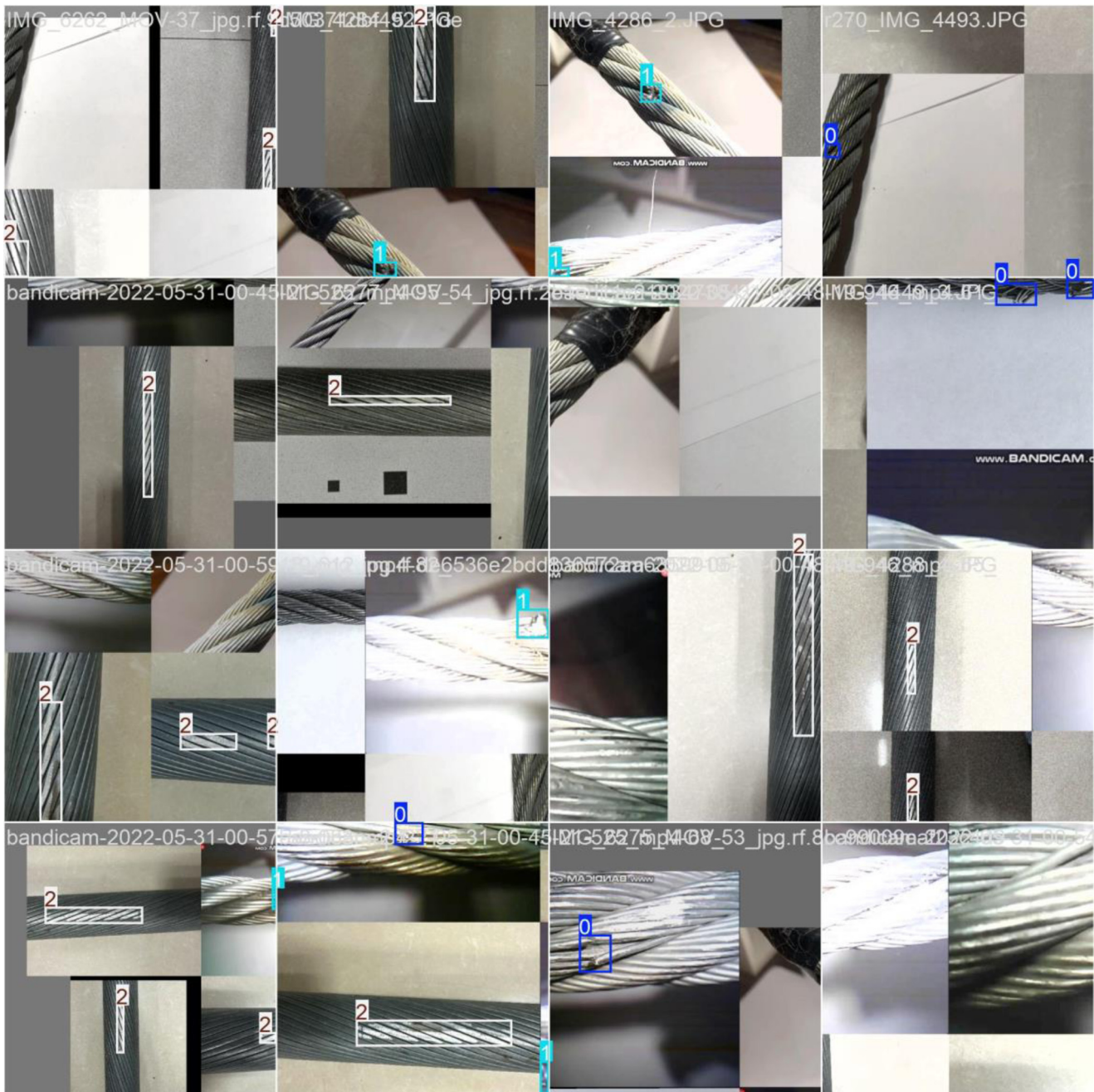


Fig. 13. Example of mosaic-augmented sample image.

example of a Mosaic-augmented image. Different rope images are merged into a single frame, increasing data diversity and object density. As a core preprocessing strategy in the YOLOv11 framework, Mosaic augmentation works together with the proposed CIA-YOLO model to improve detection accuracy and robustness.

4.2 Performance evaluation

In the performance evaluation process, mean Average Precision at IoU=0.5 (mAP@0.5) was adopted as the primary metric to assess detection accuracy. To provide a

more complete picture, we also report Precision and Recall, which are computed from True Positives (TP), False Positives (FP), and False Negatives (FN). Each metric serves a distinct purpose in object detection tasks:

- Precision refers to the proportion of true positive samples among all samples predicted as positive by the model. The formula is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (13)$$

- Recall represents the proportion of actual positive samples that are correctly detected by the model. The formula is as follows:

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (14)$$

- mAP@0.5 is the average of the per-class Average Precision (AP) values, each computed at an IoU threshold of 0.5:

$$\text{mAP@0.5} = \frac{1}{n} \sum_{k=1}^n \text{AP}_k. \quad (15)$$

The variable n represents the number of categories, and denotes the average precision for the k -th category [6]. IoU measures the overlap between the predicted bounding box and the ground truth box, and is defined as the ratio of their intersection over their union. The formula is as follows:

$$\text{IoU} = \frac{A_{\text{pred}} \cap A_{\text{gt}}}{A_{\text{pred}} \cup A_{\text{gt}}}. \quad (16)$$

In practical object detection tasks, IoU is not only used to determine true positives and false positives, but also plays a key role in optimizing the loss function during model training. It serves as a fundamental component in several evaluation metrics, including mAP@0.5.

4.3 Model training and comparative analysis

In terms of experimental setup, the proposed model was trained and tested using the PyTorch deep learning framework. The hardware environment included an 8-core CPU, 16 GB of RAM, a 1 TB hard drive, and an NVIDIA GeForce RTX 3060 Ti GPU with 8 GB of memory.

For dataset construction, images of three defect types—broken wires, corrosion, and wear—were divided into training, validation, and test sets at a ratio of 7:2:1. These subsets were then combined to form a complete dataset, consisting of 1,509 training images, 430 validation images, and 219 test images. To systematically evaluate model performance, mean Average Precision (mAP@0.5) and Recall were selected as the primary evaluation metrics.

4.3.1. Training results analysis

The improved YOLOv11 model, CIA-YOLO, was trained and validated on the constructed dataset. Table 1 shows the parameter settings for CIA-YOLO.

1. Mean Average Precision

The loss function curves and the mAP@0.5 curve during training are shown in Figure 14.

The figure illustrates the changes in loss functions and detection performance metrics during the training process of the CIA-YOLO model. As shown by the training and validation curves for box_loss, cls_loss, and dfl_loss, all loss values drop rapidly within the first 50 epochs and gradually stabilize afterward. This indicates that the model exhibits a stable training process and good convergence. The loss trends for the training and validation sets remain

Table 1. Parameter settings for the CIA-YOLO.

Parameters	Value
Input size	640×640
Initial learning rate	0.01
Batch size	32
Training epochs	300

consistent, suggesting no significant overfitting. Meanwhile, the model's precision and recall improve rapidly during the early stages of training and stabilize around epoch 100, eventually reaching approximately 95% and 94%, respectively. This demonstrates strong object detection capability and effective control over missed detections. The mAP@0.5 continues to increase and reaches 95.2%, while the multi-threshold mAP@0.5 and mAP@0.5:0.95 stabilizes around 85%, indicating robust bounding box regression accuracy and overall detection performance across different IoU thresholds. Overall, CIA-YOLO shows good training stability and comprehensive performance in detecting small targets such as steel cable defects [13].

2. Recall

Figure 15 illustrates the recall trends of different defect categories under varying confidence thresholds. Overall, the model demonstrates strong recall performance in the low to medium confidence range (0.0–0.6), with an average recall of 96% and a final stabilized value of 94.8%, indicating solid detection performance and practical applicability. Notably, for wear-type defects (steel cable2), the recall remains consistently high across all confidence levels. As the confidence threshold increases, the decline in recall is minimal, reflecting strong stability and robustness. This suggests that the predicted bounding boxes for this defect category exhibit high confidence concentration, effectively reducing low-confidence candidates while maintaining high recall. It highlights the model's ability to accurately detect small-scale defects and its reliable discrimination performance in complex backgrounds [35].

4.3.2 Ablation experiments

To thoroughly evaluate the performance improvements of the CIA-YOLO model in steel cable defect detection, a series of comparative experiments were conducted under identical experimental conditions. The baseline models included Mask R-CNN, Faster R-CNN, YOLOv8, YOLOv11, C-YOLO (with enhanced CBAM), I-YOLO (with improved Inner-IoU), A-YOLO (with improved AKConv), and CI-YOLO (with both enhanced CBAM and improved Inner-IoU). As shown in the training loss and mAP@0.5 curves in Figure 8, all models demonstrate good convergence during training, indicating effective feature learning. Among them, CIA-YOLO exhibits faster loss reduction and more stable mAP@0.5 improvement, indicating stronger convergence and generalization ability in feature learning. This suggests that CIA-YOLO not only enhances the model's accuracy but also improves its robustness across different datasets and real-world conditions, making it adaptable to diverse industrial settings.

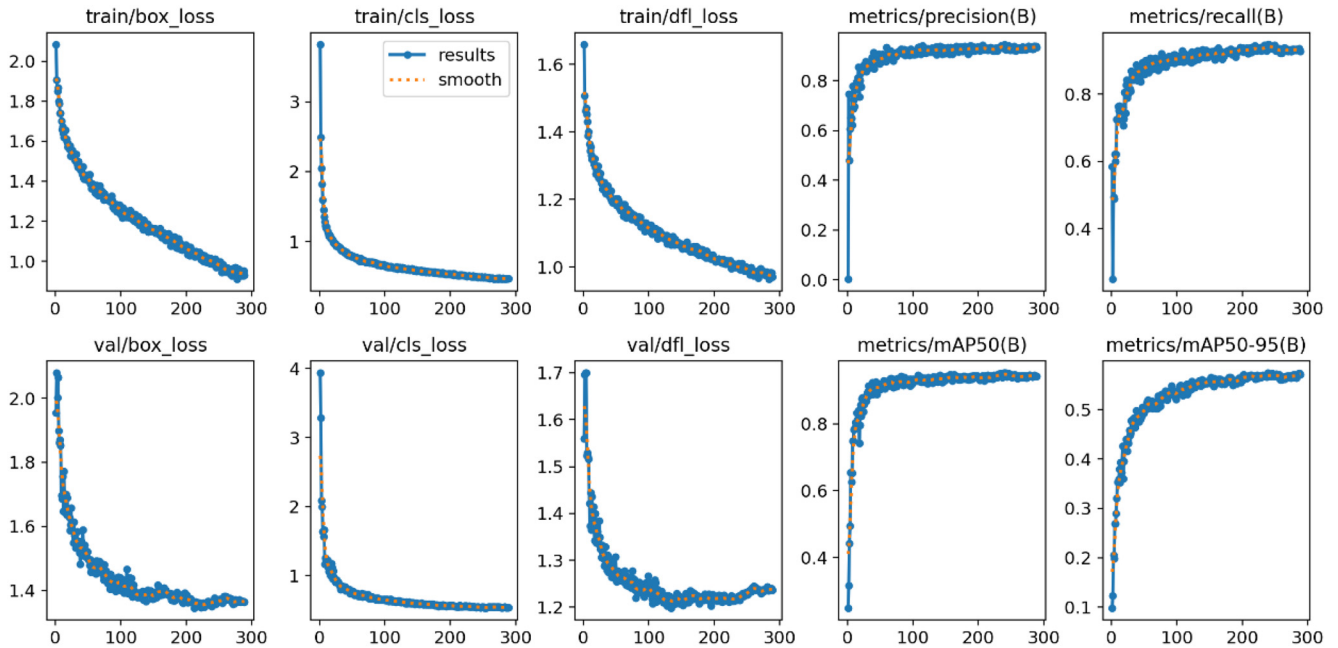


Fig. 14. Loss function curves and mAP@0.5 curve during the training process of CIA-YOLO.

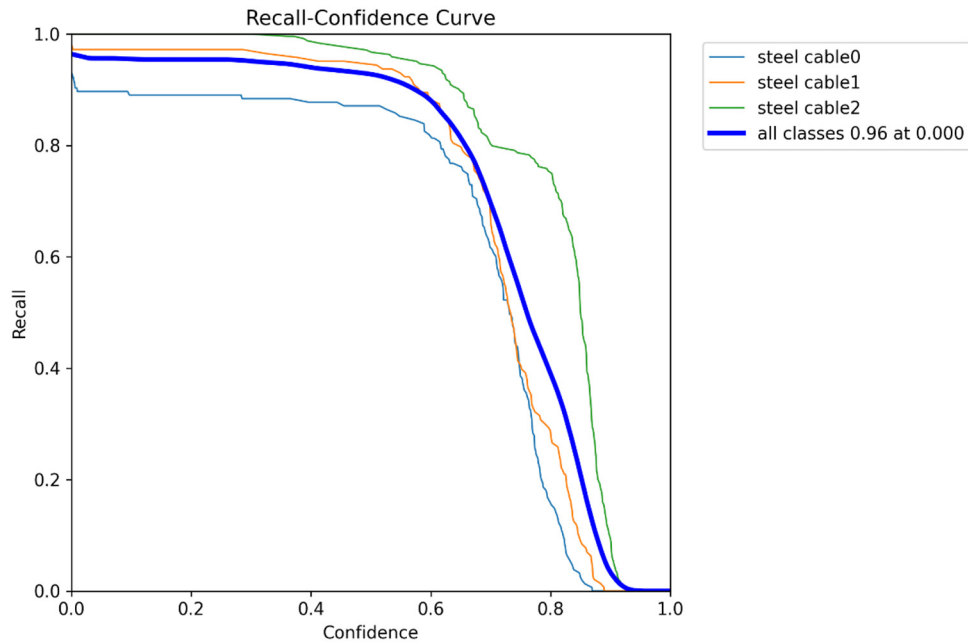


Fig. 15. Recall rates of different defect categories at various confidence thresholds.

The superior performance of CIA-YOLO highlights its potential for real-time industrial applications where both speed and accuracy are critical for defect detection.

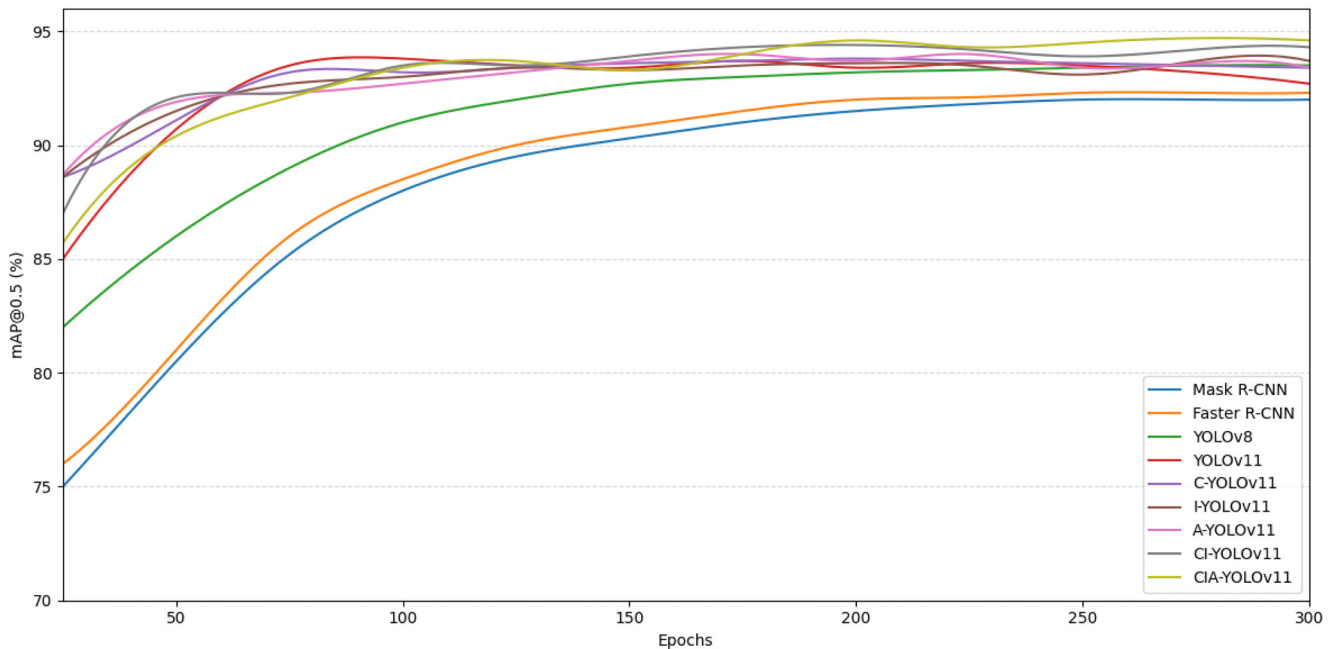
Table 2 presents the overall performance metrics of each model on the validation set, including Recall and mean Average Precision (mAP@0.5).

Table 2 presents the recall and mAP@0.5 of each detector on the same 640×640 validation set. Although the two-stage methods Mask R-CNN and Faster R-CNN converge more slowly, they still deliver reliable localization with recalls of 88.5% and 89.2% and mAP@0.5 of 92.0%

and 92.3%, respectively, at the expense of slower inference. Among single-stage models, YOLOv8 outperforms the YOLOv11 baseline by achieving 91.3% recall and 93.5% mAP@0.5. The YOLOv11-based variants further improve results: C-YOLO raises mAP@0.5 to 93.9% while maintaining 90.5% recall; I-YOLO increases recall to 92.7% with mAP@0.5 of 93.2%; and A-YOLO achieves both 93.2% recall and 93.8% mAP@0.5. Combining the enhanced CBAM attention module with the improved Inner-IoU loss in CI-YOLO yields 91.6% recall and 94.2% mAP@0.5, demonstrating a synergistic benefit. Finally,

Table 2. Overall performance metrics of different models on the validation set.

Method	Resolution	Recall (%)	mAP@0.5 (%)
Mask R-CNN	640 × 640	88.5	92.0
Faster R-CNN	640 × 640	89.2	92.3
YOLOv8	640 × 640	91.3	93.5
YOLOv11	640 × 640	90.2	92.7
C-YOLO	640 × 640	90.5	93.9
I-YOLO	640 × 640	92.7	93.2
A-YOLO	640 × 640	93.2	93.8
CI-YOLO	640 × 640	91.6	94.2
CIA-YOLO	640 × 640	94.8	95.2

**Fig. 16.** mAP@0.5 Curves of All Models over Training Epochs.

CIA-YOLO—integrating all three enhancements—leads the group with 94.8% recall and 95.2% mAP@0.5, confirming its superior robustness and precision for steel cable defect detection [36].

3. Comparison of mAP@0.5 Across Different Epochs

Figure 16 shows the mAP@0.5 Curves of All Models over Training Epochs.

As shown in the figure, the mAP@0.5 of all models steadily increases with training and converges after around 100 epochs, indicating strong stability and convergence. The two-stage detectors Mask R-CNN and Faster R-CNN grow more slowly in early epochs but ultimately stabilize at approximately 92.0% and 92.3% mAP@0.5, respectively, both exceeding the baseline YOLOv11 (92.7%). YOLOv8 catches up quickly during mid-training and settles at 93.5% mAP@0.5. C-YOLO and I-YOLO both surpass 93% mAP@0.5 by around epoch 150, stabilizing at 93.9% and 93.7%. A-YOLO and CI-YOLO reach their peaks near epoch 175, at 93.8% and 94.2% mAP@0.5, respectively. Throughout training, CIA-YOLO remains the top performer and

achieves the highest mAP@0.5 of 95.2%, demonstrating superior modeling of small defects and complex features. Overall, CIA-YOLO offers the best balance of accuracy, convergence speed, and stability, making it highly suitable for real-time, high-precision industrial defect detection.

4 Performance of CIA-YOLO in Steel Cable Defect Detection

Table 3 shows the performance metrics of CIA-YOLO in steel cable defect detection.

Table 3 presents the performance of the CIA-YOLO model in detecting steel cable defects across different types of defects, including broken wires, corrosion, and wear, with two key performance indicators: mAP@0.5 and Recall. For mAP@0.5, the model demonstrates strong performance in detecting wear (99.5%), corrosion (97.7%), and broken wire (88.5%) defects, with an overall average of 95.2%. In terms of Recall, the model achieves impressive results across all defect types, with the highest recall for wear (99.8%), followed by corrosion (96.4%) and broken wire (88.4%), yielding an overall average Recall of 94.8%.

Table 3. Performance of CIA-YOLO in Steel Cable Defect Detection at mAP@0.5 and Recall.

Indicators	Broken Wire	Corrosion	Wear	All
mAP@0.5	88.5%	97.7%	99.5%	95.2%
Recall	88.4%	96.4%	99.8%	94.8%

These results indicate that the CIA-YOLO model is highly effective for detecting various steel cable defects, especially for small and subtle defects like wear, and highlights its potential for real-time defect detection in industrial applications.

5 Prediction Results of Different Models on Various Defect Types

Figure 17 shows the prediction results of each model for different defect types, with the prediction box threshold set to 0.5.

The figure presents the visualized prediction results of YOLOv11 and its improved versions on three typical steel cable defect types: broken wires, corrosion, and wear. Compared to the original YOLOv11, the improved models demonstrate varying degrees of enhancement in bounding box localization accuracy, classification confidence, and small object recognition. I-YOLO successfully detects a broken wire target with a confidence score of 0.64, whereas the original YOLOv11 fails to identify the same defect. A-YOLO and CI-YOLO provide higher confidence and more precise localization for corrosion defects.

Notably, CIA-YOLO achieves the best performance across all three defect types. It detects broken wires with a confidence of 0.83 and generates clear and accurate bounding boxes for both corrosion and wear, significantly reducing missed and false detections. In particular, for wear-type defects, which are typically small targets, CIA-YOLO maintains high confidence and stable detection even in complex backgrounds. This indicates the model's superior capabilities in small target modeling, feature representation, and boundary box regression. Overall, CIA-YOLO, with its integrated improvements, demonstrates enhanced accuracy and robustness in multi-class defect detection tasks.

5 Discussion and analysis

To demonstrate the advancements of the proposed model, we compared and analyzed it against methods based on physics, machine learning, data-driven approaches, as well as improved versions of YOLOv5 and YOLOv8 [37]. Table 4 presents a comparison with relevant published works in this field.

Table 4 compares CIA-YOLO with several representative defect detection methods in terms of method type, components, applicable scenarios, and accuracy. Traditional physics-based methods, such as acoustic emission techniques, can detect cable breaks in bridges but have limited accuracy and only provide rough localization. Machine learning approaches using ECT signal processing offer some recognition capability, but suffer from poor generalization, with accuracy slightly above 90%. Deep learning methods show significantly better performance.

For example, a color segmentation method based on R-CNN can handle multiple defect types with an accuracy of 90.61%. An improved YOLOv5 model incorporating K-means clustering, ECA attention, and Focal Loss achieves 92.2% accuracy in detecting minor damage. Another YOLOv5s-based model combined with C3CBAM and C3Ghost performs exceptionally well in detecting metal corrosion, reaching an accuracy of 95.6%. However, most of these approaches focus on single defect types or specific scenarios.

In contrast, CIA-YOLO, the method proposed in this study, achieves an average accuracy of 95.2% across multiple defect types including broken wires, corrosion, and wear. It approaches the highest performance while offering superior adaptability and generalization. This makes it well-suited for detecting steel cable defects under complex industrial conditions and in small-target scenarios.

6 Conclusion

To address the challenges of visual detection for in-service steel cable defects such as broken wire, corrosion, and wear—particularly issues related to small feature scales and blurred boundaries—this paper proposes an improved steel cable defect detection model based on YOLOv11, named CIA-YOLO. The model integrates an enhanced CBAM attention mechanism, an improved Inner-IoU bounding box loss function, and an adaptive kernel convolution structure. These improvements collectively enhance feature extraction, target localization accuracy, and multi-scale adaptability.

Extensive experiments on a constructed steel cable defect dataset demonstrate that CIA-YOLO achieves outstanding performance, with a mAP@0.5 of 95.2% and a Recall of 94.8%, outperforming the original YOLOv11 and its variants. Compared with physics-based models, traditional image processing methods, and machine learning approaches, CIA-YOLO offers clear advantages in detection accuracy, defect type coverage, and robustness. It performs especially well in identifying multiple defect types, small targets, and defects with blurry boundaries, showing strong stability and practical value [39].

Despite its high performance, CIA-YOLO still faces challenges in extreme scenarios such as low-contrast or weak-texture backgrounds [37,40]. Future work will focus on optimizing the lightweight structure of the model to ensure its real-time performance in industrial settings. Additionally, efforts will be made to enhance the model's generalization ability by testing it in a wider range of environments and integrating multi-modal sensor data, such as thermal and acoustic signals, to improve defect detection accuracy in challenging conditions.

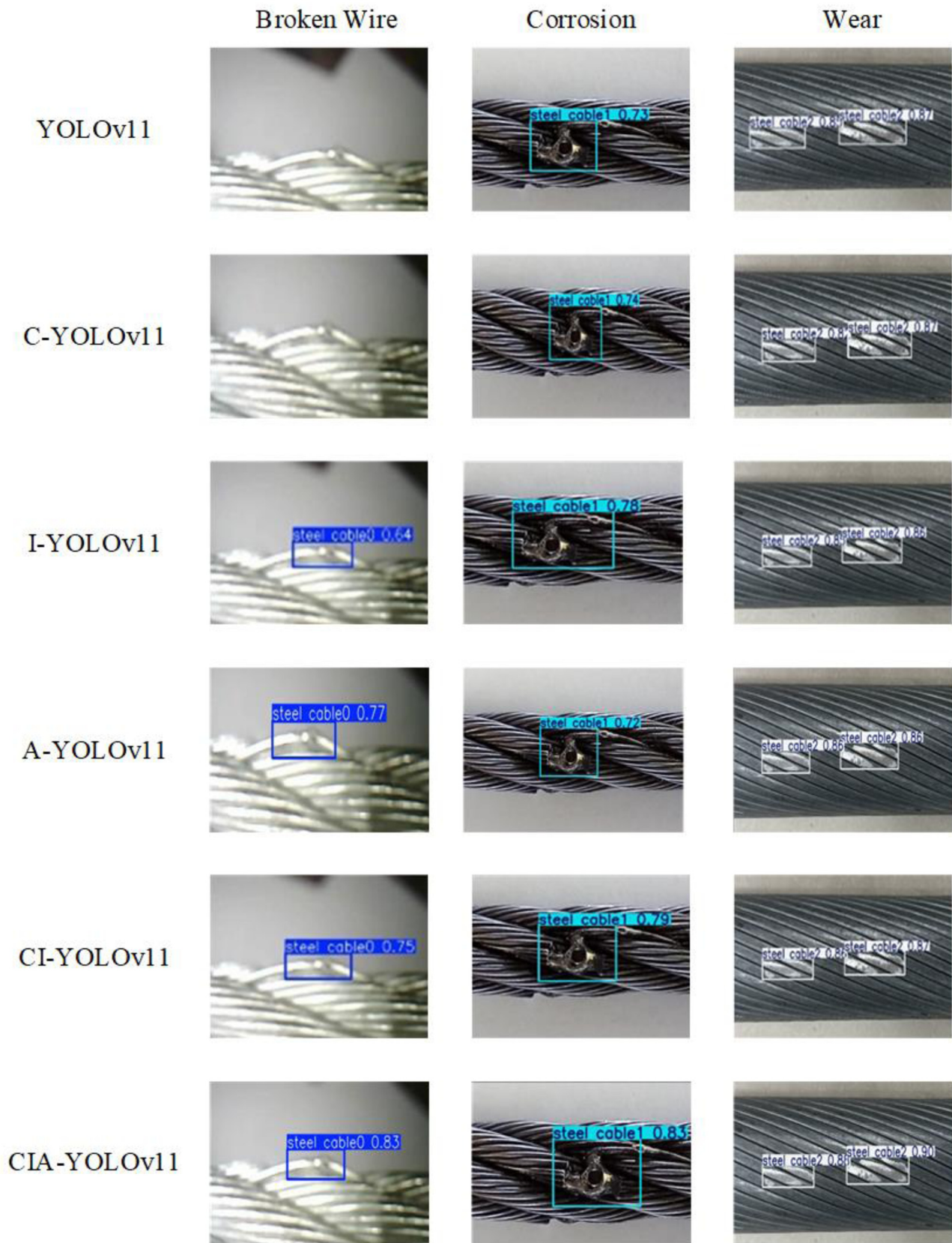


Fig. 17. Prediction results of different models on various defect types.

Table 4. Comparison of CIA-YOLO with methods in related fields.

Type	Method	Function	Accuracy
Physics-based method [4]	AE technique + signal analysis	Detecting bridge cable breaks	The error is approximately 5%.
ML-based method [5]	ECT signals + Machine Learning	classify defect type and shape	Over 90%
DL-based method [11]	Color Segmentation + R-CNN	Adapts to various defect types	90.61%
DL-based method [21]	K-means++ + ECA + Focal Loss	Detecting minor damage and complex structures	92.2%
DL-based method [22]	YOLOv5s + C3CBAM + C3Ghost	Detect and classify corrosion on metal surfaces	95.6%
DL-based method [38]	YOLOv8 + CAFM + DyConv + EfficientViT	Detecting defects in brake joint welds	90.5%
Proposed method	CIA-YOLO	Detecting and classifying wire rope defects	95.2%

Acknowledgments

This research was supported by the National Key Research and Development Program of China (No. 2022YFC13005103). The authors sincerely appreciate the support of the funding agency.

Funding

This research was supported by the National Key Research and Development Program of China (No. 2022YFC13005103).

Conflicts of interest

The authors declare no conflicts of interest.

Data availability statement

The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Author contribution statement

Conceptualization, C.S.; Methodology, C.S., Z.H., and J.C.; Software, Z.H.; Validation, Z.H.; Formal Analysis, C.S. and Z.H.; Investigation, Z.H.; Resources, C.S. and J.C.; Data Curation, Z.H.; Writing—Original Draft Preparation, Z.H.; Writing—Review and Editing, C.S.; Funding Acquisition, C.S. All authors have read and agreed to the final version of the manuscript.

References

1. Y. Chen, Y. Zhang, W. Qin, Mechanical analysis of non-perpendicularly crossed steel wires in frictional wear, *Int. J. Mech. Sci.* **156**, 170–181 (2019)
2. H. Xia, R. Yan, J. Wu, S. He, M. Zhang, Q. Qiu, J. Zhu, J. Wang, Visualization and quantification of broken wires in steel wire ropes based on induction thermography, *IEEE Sensors J.* **21**, 18497–18503 (2021)
3. P. Zhou, G. Zhou, Z. He, C. Tang, Z. Zhu, W. Li, A novel texture-based damage detection method for wire ropes, *Measurement* **148**, 106954 (2019)
4. Z. Zhu, D. Wang, C. Liu, B. Wang, Research on wire-broken monitoring of bridge cable based on acoustic emission technique, *IOP Conf. Ser.: Mater. Sci. Eng.* **652**, 012065 (2019)
5. M.C.C. Monu, J.C. Chekotu, D. Brabazon, Eddy current testing and monitoring in metal additive manufacturing: a review, *J. Manuf. Process.* **134**, 558–588 (2025)
6. P. Zhou, G. Zhou, S. Wang, H. Wang, Z. He, X. Yan, Visual sensing inspection for the surface damage of steel wire ropes with object detection method, *IEEE Sensors J.* **22**, 22985–22993 (2022)
7. F. Feng, X. Yang, R. Yang, H. Yu, F. Liao, Q. Shi, F. Zhu, An insulator defect detection network combining bidirectional feature pyramid network and attention mechanism in unmanned aerial vehicle images 2024. <https://doi.org/10.2139/ssrn.4928072>
8. C.Y. Liew, J.M.-Y. Lim, C.P. Tan, R.M.M. Bin Tun Mohar, Altitude-informed fusion pyramid network for multi-scale waste detection in unmanned aerial vehicle images, *Eng. Appl. Artif. Intell.* **153**, 110814 (2025)
9. S.G. Eladl, A.Y. Haikal, M.M. Saafan, H.Y. ZainEldin, A proposed plant classification framework for smart agricultural applications using UAV images and artificial intelligence techniques, *Alexandria Eng. J.* **109**, 466–481 (2024)
10. J. Huang, X. Zhang, L. Jia, Y. Zhou, An improved you only look once model for the multi-scale steel surface defect detection with multi-level alignment and cross-layer redistribution features, *Eng. Appl. Artif. Intell.* **145**, 110214 (2025)
11. J. Li, H. Wei, S. Yang, L. Fu, Emerging image generation with flexible control of perceived difficulty, *Comput. Vis. Image Understanding* **240**, 103919 (2024)

12. J. Wang, Q.M. Jonathan Wu, N. Zhang, You only look at once for real-time and generic multi-task, *IEEE Trans. Veh. Technol.* **73**, 12625–12637 (2024)
13. H. Zhou, F. Jiang, H. Lu, SSDA-YOLO: Semi-supervised domain adaptive YOLO for cross-domain object detection, *Comput. Vis. Image Understanding* **229**, 103649 (2023)
14. S. Jobaer, X. Tang, Y. Zhang, A deep neural network for small object detection in complex environments with unmanned aerial vehicle imagery, *Eng. Appl. Artif. Intell.* **148**, 110466 (2025)
15. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU Loss: Faster and better learning for bounding box regression, *AAAI* **34**, 12993–13000 (2020)
16. J. Zamora Esquivel, A. Cruz Vargas, P. Lopez Meyer, O. Tickoo, Adaptive convolutional kernels, in: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. Presented at the *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, IEEE, Seoul, Korea (South), 2019, pp. 1998–2005
17. C. Akdogan, T. Özer, Y. Oguz, PP-YOLO: Deep learning based detection model to detect apple and cherry trees in orchard based on Histogram and Wavelet preprocessing techniques, *Comput. Electron. Agric.* **232**, 110052 (2025)
18. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM* **60**, 84–90 (2017)
19. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 1137–1149 (2017)
20. W. Li, T. Dong, H. Shi, L. Ye, Defect detection algorithm of wire rope based on color segmentation and Faster RCNN, in: *2021 International Conference on Control, Automation and Information Sciences (ICCAIS)*. Presented at the *2021 International Conference on Control, Automation and Information Sciences (ICCAIS)*, IEEE, Xi'an, China, 2021, pp. 656–661
21. Z. Ying, Z. Lin, Z. Wu, K. Liang, X. Hu, A modified-YOLOv5s model for detection of wire braided hose defects, *Measurement* **190**, 110683 (2022)
22. M. Fu, Z. Jia, L. Wu, Z. Cui, Detection and recognition of metal surface corrosion based on CBG-YOLOv5s, *PLoS ONE* **19**, e0300440 (2024)
23. T. Talaei Khoei, H. Ould Slimane, N. Kaabouch, Deep learning: systematic review, models, challenges, and research directions, *Neural Comput. Appl.* **35**, 23103–23124 (2023)
24. X. Yang, Y. He, J. Wu, W. Sun, T. Liu, S. Ma, 3DF-FCOS: Small object detection with 3D features based on FCOS, *Comput. Vis. Image Understanding* **235**, 103787 (2023)
25. W.Y. Hsu, W.Y. Lin, Ratio-and-scale-aware YOLO for pedestrian detection, *IEEE Trans. Image Proc.* **30**, 934–947 (2021)
26. J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Presented at the *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, HI, 2017, pp. 6517–6525
27. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, 2018. <https://doi.org/10.48550/arXiv.1804.02767>
28. H. Chen, Z. He, B. Shi, T. Zhong, Research on recognition method of electrical components based on YOLO V3, *IEEE Access* **7**, 157818–157829 (2019)
29. F. Dang, D. Chen, Y. Lu, Z. Li, YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems, *Comput. Electron. Agric.* **205**, 107655 (2023)
30. Q. Ma, YOLOv5-CBAM: A small object detection model based on YOLOv5 and CBAM, in: *2024 6th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*. Presented at the *2024 6th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*, IEEE, Nanjing, China, 2024, pp. 618–623
31. Z. Ge, S. Liu, M. Technology, YOLOX: Exceeding YOLO Series in 202, n.d.
32. Y. Ding, Q. Zhao, T. Li, C. Lu, L. Tao, J. Ma, A rail defect detection framework under class-imbalanced conditions based on improved you only look once network, *Eng. Appl. Artif. Intell.* **138**, 109351 (2024)
33. A. Ahmad, D. Saraswat, V. Aggarwal, A. Etienne, B. Hancock, Performance of deep learning models for classifying and detecting common weeds in corn and soybean production systems, *Comput. Electron. Agric.* **184**, 106081 (2021)
34. Microsoft COCO: Common Objects in Context, in: *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2014, pp. 740–755
35. S. Rezwan, W. Choi, Artificial intelligence approaches for UAV navigation: recent advances and future challenges, *IEEE Access* **10**, 26320–26339 (2022)
36. L.H.R. González, S.L. Flórez, A. González-Briones, F. De La Prieta, Semantic scene understanding through advanced object context analysis in image, *Comput. Vis. Image Understanding*. **252**, 104299 (2025)
37. Z. Ma, Y. Li, M. Huang, N. Deng, Online visual end-to-end detection monitoring on surface defect of aluminum strip under the industrial few-shot condition, *J. Manuf. Syst.* **70**, 31–47 (2023)
38. J. Ren, H. Zhang, M. Yue, YOLOv8-WD: Deep learning-based detection of defects in automotive brake joint laser welds, *Appl. Sci.* **15**, 1184 (2025)
39. S. Yue, Z. Zhang, Y. Shi, Y. Cai, WGS-YOLO: A real-time object detector based on YOLO framework for autonomous driving, *Comput. Vis. Image Understanding*. **249**, 104200 (2024)
40. S. Ye, W. Huang, W. Liu, L. Chen, X. Wang, X. Zhong, YES: You should Examine Suspect cues for low-light object detection, *Comput. Vis. Image Understanding* **251**, 104271 (2025)