EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

MC-YOLO: A Lightweight Insulator Defect Detection Model Based on an Improved YOLOv8

Junlian Wang[1], Zhixiong Li[2], Jinquan Yang[3], Hongbing Ren[2], Wenchao Pan[2], Yang Feng[3], Lei Xiong[1], Chune Li[1], Yigong Zhang[1,*]

[1]School of Information Engineering, Kunming University, Kunming, Yunnan 650214, China
[2]Yunnan Transportation Engineering Quality Testing Company Limited, Kunming, Yunnan 650011, China
[3]Yunnan Transportation Research Institute Company Limited, Kunming, Yunnan 650011, China

Abstract

Due to the fast growth of China's electric power industry, the total length of high-voltage transmission lines has been continuously increasing. As a key component of high-voltage transmission systems, insulators play a critical role, and achieving efficient and accurate defect detection for insulators is of great significance. To address the practical challenges of limited size, limited computational resources, restricted energy supply, and complex environmental conditions on Unmanned Aerial Vehicle (UAV) platforms, this paper proposes a lightweight insulator defect detection model, MC-YOLO, based on an improved YOLOv8 architecture. Specifically, the original backbone network is replaced with the lighter MobileNetV3 module, reducing the model parameters and GFLOPs to 21.3% and 20% of those of the original model, respectively. In addition, a Convolutional Block Attention Module (CBAM) is integrated into the network neck structure to effectively improve the extraction of key features, resulting in a 1.2 percentage points improvement in detection accuracy. Finally, the loss function is changed to Wise-IoU (WIoU) v3, which increases the localization capacity of the model and further increases the accuracy by 1.4 percentage points. Experimental results demonstrate that the proposed MC-YOLO model achieves a lightweight design while maintaining high detection performance, providing a viable technical solution for edge deployment in real-world engineering applications.

Received on 06 January 2026; accepted on 20 March 2026; published on 11 August 2026

Keywords: Insulator Defect Detection, Lightweight Model, YOLOv8, MobileNetV3

doi:10.4108/ew.13841

1. Introduction

As China's electric power industry continues to develop, stable power equipment operation has become a crucial prerequisite to ensure normal production of various sectors. As of 2023, the national power grid had approximately 920,000 kilometers of transmission lines at 220 kV and higher, which marks a 70% increase from 2013[1].

Insulators constitute a critical component in high-voltage power systems, providing mechanical support to conductors while ensuring electrical insulation between the conductors and other components of the system[2]. However, harsh outdoor weather conditions can easily cause damage to insulators. Detecting insulator defects on time and with precision, followed by prompt maintenance, is essential to ensure normal transmission line operation and prevent power safety incidents[3]. Traditional manual inspections require traversing complex terrains, leading to low efficiency and significant safety risks. With the ongoing development of deep learning[4], object detection algorithms have gradually been applied to insulator defect detection.

Current object detection methods mainly consist of single-stage and two-stage approaches. Single-stage algorithms primarily include the YOLO series and SSD algorithms, while two-stage algorithms mainly consist of the R-CNN series and Mask R-CNN series[5]. Single-stage object detection algorithms, unlike two-stage methods, provide end-to-end processing with

*Corresponding author. Email: kmuzhyg@163.com

faster computation. Numerous insulator defect detection studies have been conducted based on these single-stage algorithms. Wang et al. first employed a Fully Convolutional Network (FCN) to segment insulator targets, followed by YOLOv3 for defect detection, which improved the model's mean Average Precision (mAP) by 4.65%[6]. Chen et al. introduced Insu-YOLO, an enhanced method for insulator defect detection derived from the YOLOv8 architecture. While the enhanced model achieved higher accuracy, its GFLOPs increased by 70.4%[7]. Song et al. proposed an insulator defect detection method based on Flexible YOLOv7, with a floating-point computation complexity of 164.7 GFLOPs[8]. To address the issue that insulators are prone to missed detections under partial occlusion, Zhang et al. integrated the Convolutional Block Attention Module (CBAM) attention mechanism into YOLOv7, thereby significantly reducing the missed detection rate[9]. Existing studies still struggle to achieve an optimal balance between model lightweightness and detection accuracy. YOLOv8[10], by incorporating the C2f module and a dynamic label assignment strategy, significantly enhances the trade-off between detection speed and precision. Accordingly, this study aims to implement lightweight optimization based on the YOLOv8 framework. The main contributions of this study are as follows:

(1)MobileNetV3 replaces the backbone of YOLOv8, significantly reducing the model's computational cost and number of parameters. This modification results in a lightweight architecture suitable for edge-device deployment.

(2)The CBAM is added to the network neck, adaptively assigning attention along channel and spatial dimensions. This strengthens multi-scale feature representation and mitigates detection accuracy loss caused by feature compression.

(3)This study adopts the Wise-IoU (WIoU) v3 as the model's bounding box regression loss function, effectively enhancing the model's localization ability and generalization performance.

2. Methods

2.1. YOLOv8

In early 2023, Ultralytics released YOLOv8, an object detection algorithm that includes comprehensive optimizations in both its architectural design and training strategies. Compared with YOLOv5 and YOLOv7, YOLOv8 further improves inference speed and model generalization while maintaining high detection accuracy. It offers five versions—n, s, m, l, and x—by adjusting the network's depth and width coefficients accordingly. As the model scale increases, the number of parameters and resource consumption also grow, accompanied by improvements in detection performance. For the various model scales, their width, depth, and maximum channel numbers are listed in Table 1. YOLOv8s achieves favorable detection performance while consuming relatively low computational resources. To better balance detection accuracy and computational efficiency, this study adopts YOLOv8s as the baseline model.

Table 1. Configuration Parameters of YOLOv8 Models with Different Scales

Model	Depth	Width	Max Channels
YOLOv8n	0.33	0.25	1024
YOLOv8s	0.33	0.50	1024
YOLOv8m	0.67	0.75	768
YOLOv8l	1.00	1.00	512
YOLOv8x	1.00	1.25	512

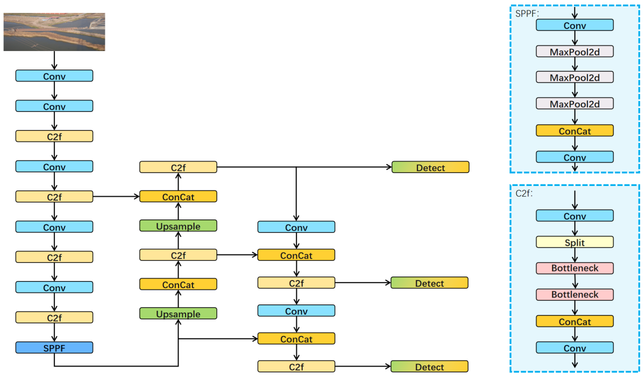

Figure 1. Overall architecture of YOLOv8

YOLOv8 includes three principal components: a backbone network, a neck network, and a prediction head, as illustrated in Figure 1. The input image is first fed into the backbone for feature extraction, and the resulting features are passed to the neck for multi-scale feature combination. Finally, three layers of features from the neck are forwarded to the prediction head to perform object detection and classification at different scales.

In UAV images featuring complex backgrounds, detecting small insulator defects is challenging; this study proposes methods to address this issue. The improvements based on YOLOv8s mainly consist of three parts: first, the backbone of the original model is substituted with the more lightweight MobileNetV3[11] module, significantly reducing the model's computational cost and number of parameters. Secondly, the CBAM[12] is inserted into the neck to recover the accuracy lost due to model lightweighting. Finally, the original bounding box regression loss function is replaced with WIoU v3[13], further improving the detection accuracy. Figure 2 depicts the architecture of the improved model.

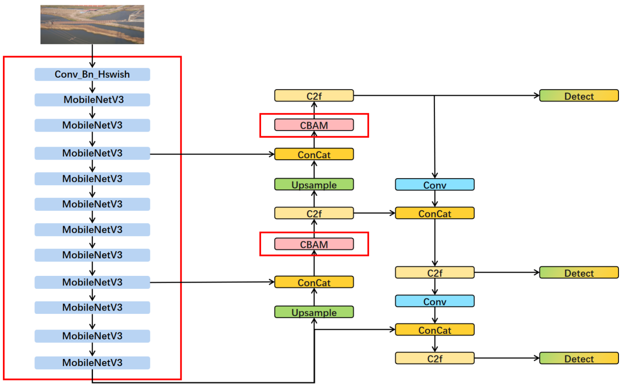

Figure 2. Overall Architecture of MC-YOLO

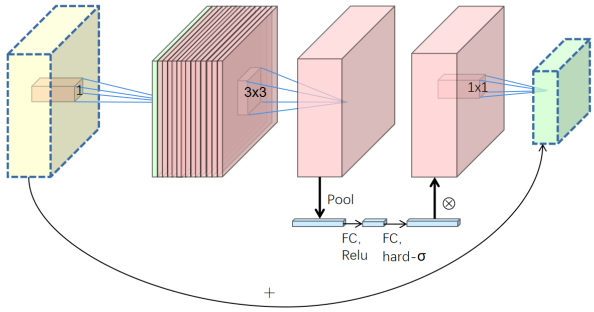

Figure 3. Principal architectural structure of MobileNetv3

2.2. MobileNetV3 Module

MobileNetV3, proposed by Google in 2019, is a convolutional neural network that balances lightweight design with high accuracy. The network uses depthwise separable convolution and linear bottlenecks with residual connections from MobileNetV1 and V2. In addition, structural optimizations have been made based on these foundations. Specifically, the overall architecture was adjusted through Neural Architecture Search (NAS), achieving a better balance between accuracy and computational efficiency. Moreover, to enhance feature representation, the Squeeze-and-Excitation (SE) attention mechanism was introduced. On the other hand, considering the computational constraints of mobile devices, the network replaces the traditional swish activation function with h-swish. The h-swish function not only maintains nonlinear representation capabilities but also significantly reduces computational complexity, enabling the model to run more efficiently on resource-limited edge devices, particularly suitable for applications that require quick responses and low latency. The formula for the swish function is shown in Equation (1), and the formula for the h-swish function is shown in Equation (2)[14].

$$\text{swish}\, x = x \cdot \delta(x) \tag{1}$$

In which x denotes the input. $\delta(x)$ represents the application of the sigmoid function to x.

$$h - \text{swish}(x) = x \cdot \frac{\text{Re}\,LU6(x+3)}{6} \tag{2}$$

Here, x denotes the function's input. $\text{Re}\,LU6(x)$ refers to a rectified linear unit whose output is limited to a maximum of 6, ensuring it remains within a specific range.

To meet varying requirements for accuracy and speed across different application scenarios, two variants of MobileNetV3 are available: MobileNetV3-Large and MobileNetV3-Small. Among them, MobileNetV3-Small is more suitable for resource-constrained devices. Therefore, this study adopts MobileNetV3-Small as the backbone network. The structural schematic of the MobileNetV3 network is illustrated in Figure 3. The layer-wise configuration details of the MobileNetV3 model adopted in this study are presented in Table 2.

Figure 3 shows the network architecture. The network first employs a 1×1 convolution to expand the number of channels. Subsequently, a 3×3 depthwise separable convolution is introduced to extract spatial features. Then, a 1×1 convolution is applied to compress the channels. In some modules, the SE attention mechanism is further integrated. This mechanism dynamically adjusts the feature responses across channels, thereby improving the overall feature representation capability.

Table 2 provides a detailed summary of the MobileNetV3 architecture adopted in this study. It outlines key design elements of each stage of the network, including input resolution, expansion ratio of the bottleneck layers, kernel size, stride, the presence of the SE module, and the type of activation function employed. This comprehensive presentation facilitates a clear understanding of the model's structural configuration.

2.3. CBAM Module

CBAM is a lightweight attention mechanism designed to improve the feature representation of Convolutional Neural Networks (CNNs) for various tasks[12]. It refines intermediate feature maps through two submodules: Channel Attention and Spatial Attention, enhancing sensitivity to key regions. Channel attention is computed by extracting global channel information via global average and max pooling, followed by a shared multi-layer perceptron (MLP) to generate the attention map. Spatial attention is then applied to the channel-weighted feature map by combining average and max pooling results, followed by a convolution to generate the spatial attention map. Finally, the channel

Table 2. Layer-wise Configuration of MobileNetV3

Layer	Operator	out_channels	expand_channels	stride	SE	Activation Function
0	Conv+BN+h-swish,3×3	16	-	2	-	H-Swish
1	MobileNetV3_InvertedResidual,3×3	16	16	2	✓	ReLU
2	MobileNetV3_InvertedResidual,3×3	24	72	2	-	ReLU
3	MobileNetV3_InvertedResidual,3×3	24	88	1	-	ReLU
4	MobileNetV3_InvertedResidual,5×5	40	96	2	✓	H-Swish
5	MobileNetV3_InvertedResidual,5×5	40	240	1	✓	H-Swish
6	MobileNetV3_InvertedResidual,5×5	40	240	1	✓	H-Swish
7	MobileNetV3_InvertedResidual,5×5	48	120	1	✓	H-Swish
8	MobileNetV3_InvertedResidual,5×5	48	144	1	✓	H-Swish
9	MobileNetV3_InvertedResidual,5×5	96	288	2	✓	H-Swish
10	MobileNetV3_InvertedResidual,5×5	96	576	1	✓	H-Swish
11	MobileNetV3_InvertedResidual,5×5	96	576	1	✓	H-Swish

and spatial attention maps are multiplied element-wise with the input feature map for refinement. The structure is shown in Figure 4.

The CBAM module adaptively assigns attention weights to each feature map, thereby strengthening informative features and suppressing redundant information during multi-scale feature fusion. Therefore, this paper introduces this module into the Neck part of YOLOv8. The aim is to further improve the model's attention to the target region.

2.4. Wiou Loss

In deep neural networks, the loss function measures the difference between predicted outputs and ground truth. It directs the model to iteratively update its parameters using optimization algorithms to enhance prediction accuracy. YOLOv8 employs Complete Intersection over Union(CIoU) for loss calculation. Its formulation is presented in Equations (3)–(6)[15].

$$L_{\text{CIOU}} = 1 - \text{IOU} + \frac{\rho(b, b_t)}{(w_c)^2 + (h_c)^2} + \alpha\nu \quad (3)$$

$$R_{\text{CIOU}} = \frac{\rho^2(b, b_t)}{c^2} + \alpha\nu \quad (4)$$

$$\alpha = \frac{\nu}{(1 - \text{IOU}) + \nu} \quad (5)$$

$$\nu = \frac{4}{\pi^2}\left(\arctan\frac{w_t}{h_t} - \arctan\frac{w}{h}\right)^2 \quad (6)$$

w_c and h_c in the above formulas correspond to the width and height of the smallest box covering both the predicted and ground truth boxes, while ρ refers to the Euclidean distance between the center points b and b_t; b and b_t represent the center coordinates of the predicted and ground truth boxes; IoU represents the Intersection over Union between the predicted and ground truth boxes; b_t is used to balance the corresponding penalty term; ν helps maintain consistency in the aspect ratio. In formula(6), w and h represent the width and height of the predicted bounding box, while w_t and h_t denote the width and height of the ground truth box.

For complex insulator defect features, particularly in the case of small objects, the CIoU loss function tends to exhibit significant fluctuations and sample quality imbalance, which may degrade the robustness and accuracy of the model. To resolve this issue, the CIoU loss is replaced with the WIoUv3 loss function in this study. According to Tong et al., the WIoU loss has three distinct versions: WIoU v1 designs the bounding box regression loss based on an attention mechanism[13]. Its formulas are shown in Equations (7)–(9)[15].

$$L_{IoU} = 1 - IoU \quad (7)$$

$$L_{WIoUv1} = R_{WIoU} \cdot L_{IoU} \quad (8)$$

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)}\right) \quad (9)$$

In formula(9), (x,y) represents the center coordinates of the predicted box, and (x_{gt},y_{gt}) represents the center coordinates of the corresponding ground truth box. W_g is the width of the minimum enclosing box, and H_g is the height of the minimum enclosing box. R_{WIoU} represents the penalty weight of the predicted box, which is used to measure the degree of deviation of the predicted box from the center of the target box.

While WIoU v2 and v3 further incorporate a focus factor into the formulation. In particular, WIoU v3 introduces a dynamic focusing mechanism to

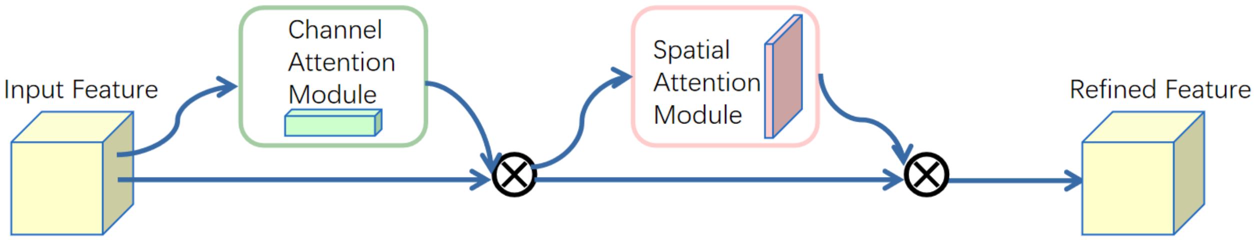

Figure 4. Schematic diagram of the CBAM attention mechanism structure
Note: The symbol ⊗ denotes element-wise multiplication.

replace the monotonic focusing strategy for more effective assessment of the predicted bounding box quality[14]. Unlike CIoU, WIoU v3 evaluates the quality of predicted bounding boxes based on the degree of outlierness rather than the IoU metric. In addition, the WIoU v3 loss function incorporates a gradient gain allocation strategy. Specifically, for small-scale features, WIoU v3 reduces the dominance of high-quality predicted boxes, thereby mitigating the negative gradient influence from low-quality predictions. This mechanism improves the model's capacity to detect small target features, thereby boosting overall performance. The formulation of the WIoU v3 loss function is presented in Equations (10)–(12)[13][15].

$$L_{WIoUv3} = r \cdot L_{WIoUv1} \tag{10}$$

$$r = \frac{\beta}{\delta \alpha^{\beta-\delta}} \tag{11}$$

$$\beta = \frac{L^*_{IoU}}{L_{IoU}} \in [0, +\infty) \tag{12}$$

α and β represent the constants and outliers, respectively, while r is a ratio term that adjusts dynamically, with its value varying according to the number of training iterations.

3. Experimental Preparation

3.1. Construction of the Insulator Dataset

In this study, we utilized a publicly available open-source insulator dataset released by Heitor Felix and collaborators on GitHub[16]. This dataset includes the China Power Line Insulator Dataset (CPLID), which was proposed in 2018, as well as additional insulator images generated through rotation and transformation based on the original CPLID.Based on the original dataset, this study also performed random deletion, and the remaining images were screened and divided, resulting in a final total of 5,210 images being retained. The images were split into training, validation, and test sets in a 7:2:1 ratio. The training set contained 3,647 images, among which 673 images were identified as having invalid labels. The validation set initially contained 1,042 images, of which 184 were found to have invalid labels. As a result, the training set contained 2,974 valid images, and the validation set contained 858 valid images. The number of images in each dataset is shown in Table 3. Examples of the dataset images are presented in Figure 5.

Table 3. Summary Of Dataset Annotations

Class	Train	Val	Test
Image Count	2974	858	413
Normal Insulators	3569	1069	513
Defective Insulators	968	245	119
Insulator Defect	968	245	119

3.2. Experimental Setup

The training was performed on a PC equipped with an NVIDIA GeForce RTX 4050 laptop GPU (6 GB VRAM). All experiments were conducted under identical hardware and software environments, including Python 3.9.20, PyTorch 2.5.1, and CUDA 11.8. The input image resolution was set to 640 × 640 pixels. Training lasted approximately 3 hours, with 100 epochs and a batch size of 16.

3.3. Model Evaluation Metrics

In order to objectively assess the performance advantages of the improved YOLOv8 algorithm, this study adopts mAP50, GFLOPs, and parameter count (Params) as unified evaluation metrics. The corresponding formulas are presented below[17].

$$AP = \int_0^1 P(R)\, dR \times 100\% \tag{13}$$

Figure 5. Sample images from the insulator dataset

$$mAP50 = \frac{1}{n}\sum_{i=1}^{n} APi \tag{14}$$

In the formulas(13), AP represents the average precision of the network model for a single detection object; In the formulas(14), AP(i) denotes the average precision of the i-th category; and mAP50 refers to the mean average precision of all detected objects at an IoU threshold of 0.50. The number of parameters and GFLOPs, as metrics of spatial complexity and computational complexity, respectively, effectively reflect the model's performance and efficiency.

4. Experimental Results and Analysis

4.1. Comparative Experiments on Lightweight Improvements

In this study, four lightweight improvement methods—FastNet, MobileNetV3, ShuffleNetV2, and GhostNet—were selected to replace the Backbone for experimentation. The comparison of each model's mAP50, GFLOPs, and parameter count is presented in Table 4.

As shown in the comparison results in Table 4, the model using MobileNetV3 demonstrates the best overall performance. While its mAP50 is comparable to other lightweight improvements, it achieves lower computational complexity (GFLOPs) and fewer parameters than the other models. Compared to YOLOv8s, although the mAP50 decreases by 7.9 percentage points, the computation reduces by 22.7 GFLOPs and the number of parameters decreases by 8.79 MB. Therefore, subsequent experiments comparing attention mechanisms are based on this model.

4.2. Comparative Experiments on Attention Mechanisms

Due to the accuracy degradation caused by the lightweight modifications, this study attempted to incorporate attention mechanisms into the neck part of the network to improve performance. The selected attention mechanisms include EMA, GAM, CA, and CBAM. All experiments were conducted based on the MobileNetV3-improved model, with the attention modules inserted at the same location within the neck. The comparative experimental results are shown in Table 5.

Experimental results demonstrate that incorporating the CBAM attention mechanism increases mAP50 by 1.2 percentage points, while maintaining unchanged computational complexity and only a slight parameter increase of 0.03 MB. Compared to other attention mechanisms, CBAM achieves the best performance by significantly improving accuracy with minimal computational overhead.

4.3. Ablation Study

In continuation of the two prior modifications, this study delves into the performance implications of employing WIoUv3 as a replacement for the original loss function. The results of the ablation experiments

Table 4. Comparison of Experimental Results for Backbone Improvements

Model	mAP50	GFLOPs	Parameters (Millions)
YOLOv8s	0.899	28.4	11.13
YOLOv8s-MobileNetV3	0.820	5.7	2.34
YOLOv8s-GhostNet	0.857	23.0	8.31
YOLOv8s-ShuffleNetV2	0.818	16.4	6.38
YOLOv8s-FastNet	0.823	16.1	6.08

Table 5. Comparison of Experimental Results with Different Attention Mechanisms

Model	mAP50	GFLOPs	Parameters(Millions)
MobileNetV3	0.820	5.7	2.34
EMA*	0.823	6.1	2.34
GAM*	0.834	12.8	3.0
CA*	0.825	5.7	2.35
CBAM*	0.832	5.7	2.37

*Indicates integration into the MobileNetV3-based model.

for the proposed improvements are presented in Table 6.

In Table 6, a "✓" symbol indicates that the corresponding improvement was applied to the YOLOv8s model. The results show that replacing the backbone with MobileNetV3 significantly reduced the model's GFLOPs and parameter size, but also led to a 7.9 percentage points decrease in mAP50. After introducing the CBAM attention mechanism, the mAP50 increased by 1.2 percentage points with negligible changes in computational cost and parameter size. Furthermore, employing the WIoU v3 loss function resulted in an additional 1.4 percentage points improvement in mAP50. In summary, the improved model MC-YOLO achieved an mAP50 approximately 5.3 percentage points lower than that of YOLOv8s, while its computational cost and parameter size were only 20% and 21.3% of those of YOLOv8s, respectively. Table 7 presents a comparison of the per-class mAP values between MC-YOLO and YOLOv8s.

Table 7 indicates that MC-YOLO exhibits a slight decrease in overall mAP and per-class performance compared with YOLOv8s. Specifically, the detection accuracy for the defect category—the primary focus in practical applications—decreases by only 2 percentage points. This demonstrates that the proposed model effectively maintains its detection capability for key targets while achieving significant model lightweighting. Figure 6 further illustrates a visual comparison of the detection results between MC-YOLO and YOLOv8s.

Figure 6 illustrates the comparison between the original annotations and the detection results of YOLOv8s and MC-YOLO. It is observed that although MC-YOLO yields slightly lower confidence scores than YOLOv8s, the model accurately identifies insulators and their defects in most test samples without significant missed detections. In the fourth example, MC-YOLO detects an additional defect with a confidence score of 0.26, which can be categorized as a low-confidence false positive. Notably, this defect corresponds to a black block present in the original dataset; therefore, the model's identification of it as a defect target is reasonable to some extent. Overall, MC-YOLO enables rapid and accurate detection of insulator defects, providing robust technical support for the safe and stable operation of power systems.

4.4. Comparative Experiments with Lightweight Models

To further evaluate the effectiveness of the proposed MC-YOLO, other representative lightweight object detection models—YOLOv5n, YOLOv8n, and YOLOv10n—were trained under the same experimental conditions and datasets. Table 8 presents the comparative results and per-class mAP values for each model's detection performance.

The comparative results in Table 8 indicate that, relative to mainstream lightweight models, MC-YOLO offers a certain advantage in maintaining model compactness. Compared with YOLOv5n, MC-YOLO's mAP for the defect category decreases by only 1.1 percentage points, while its GFLOPs are reduced by 19.7%. Compared with YOLOv8n, the defect category mAP decreases by only 1.3 percentage points, with GFLOPs reduced by 29.6%. Compared with YOLOv10n, MC-YOLO even achieves a 0.4 percentage point increase in defect category detection, while GFLOPs are still reduced by 12.3%. In power inspection scenarios, the accuracy of insulator defect detection directly affects system safety, and MC-YOLO exhibits stable performance in this critical category, demonstrating high practical value. MC-YOLO reduces computational complexity while maintaining high detection accuracy. As a result, it achieves a reasonable balance between model lightweighting and detection performance, thereby further indicating its potential for real-world engineering applications.

Table 6. Ablation Study Results

MobileNetV3	CBAM	WIoU-Loss	mAP50	GFLOPs	Parameters (Millions)
			0.899	28.4	11.13
✓			0.820	5.7	2.34
✓	✓		0.832	5.7	2.37
✓	✓	✓	0.846	5.7	2.37

Table 7. Comparison of Per-Class Detection Performance Between YOLOv8s and MC-YOLO

Model	mAP50_all	mAP50_normal_insulator	mAP50_defective_insulators	mAP50_insulators_fault
YOLOv8s	0.899	0.892	0.969	0.836
MC-YOLO	0.846	0.797	0.949	0.793

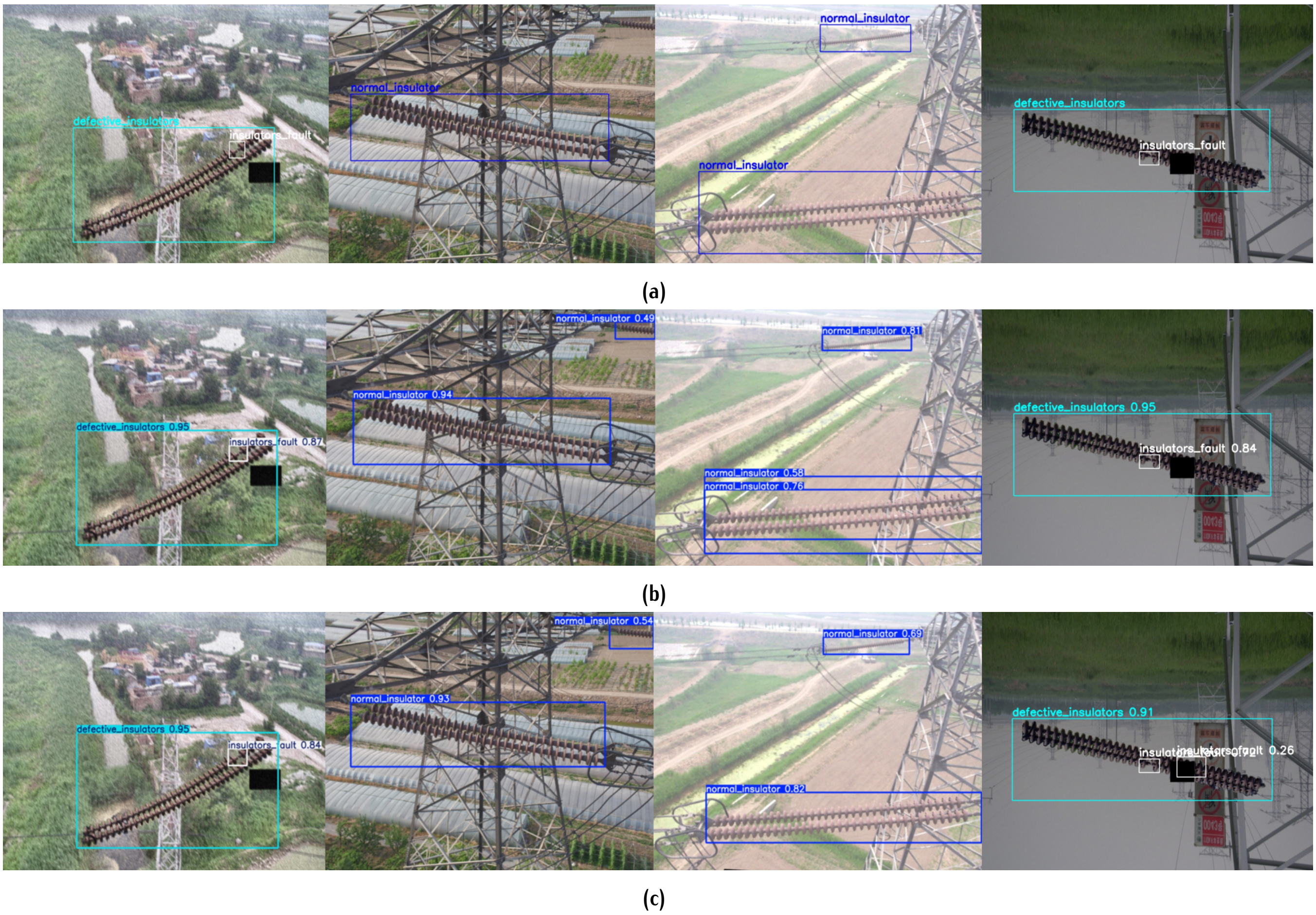

Figure 6. Comparison of Detection Results (a) Dataset Annotations (b) Predictions by YOLOv8s(c) Predictions by MC-YOLO

Table 8. Performance Comparison of Lightweight Models

Model	mAP_all	mAP_normal_insulator	mAP_defective_insulators	mAP_insulators_fault	GFLOPs	Parameters (Millions)
YOLOv5n	0.882	0.881	0.960	0.805	7.1	2.50
YOLOv8n	0.882	0.882	0.962	0.801	8.1	3.01
YOLOv10n	0.857	0.864	0.945	0.762	6.5	2.27
MC-YOLO	0.846	0.797	0.949	0.793	5.7	2.37

5. Conclusion

This paper presents MC-YOLO, a lightweight algorithm for detecting insulator defects in UAV inspection tasks, built on the YOLOv8 model.

To address the issues of high computational cost, large parameter size, and stringent hardware requirements for deployment, this study replaces the YOLOv8 backbone with the MobileNetV3 module. By making this replacement, the model's computational load and number of parameters are notably decreased, thereby reducing hardware reliance and enhancing runtime efficiency. In order to compensate for the accuracy decline caused by model lightweighting, the CBAM is added to the model's neck, which successfully restores its detection accuracy. In addition, replacing the default loss function with WIoU v3 further improves the model's accuracy and robustness.

Experimental results demonstrate that compared to YOLOv8s, the proposed MC-YOLO model reduces computational cost and parameter size to 20% and 21.3% of the original model, respectively, while the mAP50 decreases by only 5.3 percentage points. Specifically, MC-YOLO reduces computational resource consumption while maintaining high accuracy, enabling fast and accurate insulator defect detection on hardware platforms with limited resources. The proposed MC-YOLO model can effectively m eet r outine inspection requirements without relying on high-cost hardware, demonstrating significant p ractical v alue a nd strong potential for widespread application.

Subsequent research will aim to refine the MC-YOLO architecture, enhance detection precision, and achieve seamless integration with UAV platforms to improve the system's real-world applicability and scalability.

Acknowledgement Financial support for this work was provided by the National Natural Science Foundation of China (Grant Nos. 12173085), the Yunnan Province Technological Innovation Talent Training Project (No. 202305AD160004), and the CCTA Science and Technology Project (No. CCTA-TI&A-2024058CT).

References

[1] National Energy Administration of China. Length of Transmission Lines of 220 kV and Above Reached About 920,000 Kilometers; 2024. Accessed: 2025-09-21. https://www.nea.gov.cn/2024-07/19/c_1310782066.htm.

[2] Park KC, Motai Y, Yoon JR. Acoustic Fault Detection Technique for High-Power Insulators. IEEE Transactions on Industrial Electronics. 2017;(64-12).

[3] Qiu Z, Zhu X, Liao C, Shi D, Qu W. Detection of transmission line insulator defects based on an improved lightweight YOLOv4 model. Applied Sciences. 2022;12(3):1207.

[4] LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015;521(7553):436-44.

[5] Liu C, Wu Y. Research Progress on Visual Inspection Methods for Transmission Lines Based on Deep Learning. Proceedings of the Chinese Society for Electrical Engineering. 2023;43(19):7423-46.

[6] Wang Z, Wang Y, Wang Q, Kang S, Mikulovich VI. Two-Stage Insulator Fault Detection Method Based on Cooperative Deep Learning. Transactions of China Electrotechnical Society. 2021;36(17):3594-604.

[7] Chen Y, Liu H, Chen J, Hu J, Zheng E. Insu-YOLO: an insulator defect detection algorithm based on multiscale feature fusion. Electronics. 2023;12(15):3210.

[8] Song Z, Huang X, Ji C, Zhang Y. Transmission Line Insulator Defect Detection and Fault Early Warning Method Based on Flexible YOLOv7. High Voltage Engineering. 2023;49(12):5084-94.

[9] Zhang J, Wei X, Zhang L, Chen Y, Lu J. Insulator Detection and Localization Using an Improved YOLOv7. Computer Engineering and Applications. 2024;60(4):183-91.

[10] Ultralytics. YOLOv8 Documentation; 2023. Accessed: 2025-09-05. https://docs.ultralytics.com/models/yolov8/.

[11] Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, et al. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision; 2019. p. 1314-24.

[12] Woo S, Park J, Lee JY, Kweon IS. Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 3-19.

[13] Tong Z, Chen Y, Xu Z, Yu R. Wise-IoU: bounding box regression loss with dynamic focusing mechanism. arXiv preprint arXiv:230110051. 2023.

[14] Ennaama S, Silkan H, Bentajer A, Tahiri A. Enhanced real-time object detection using YOLOv7 and MobileNetv3. Engineering, Technology & Applied Science Research. 2025;15(1):19181-7.

[15] Huang Y, Feng X, Han T, Song H, Liu Y, Bao M. GDS-YOLO: A Rice Diseases Identification Model With Enhanced Feature Extraction Capability. IET Image Processing. 2025;19(1):e70034.

[16] Vieira-e Silva AL, Chaves T, Felix H, Macêdo D, Simões F, Gama-Neto M, et al.. Unifying Public Datasets for Insulator Detection and Fault Classification in Electrical Power Lines; 2020. https://github.com/heitorcfelix/public-insulator-datasets.

[17] Zhang S, Bei Z, Ling T, Chen Q, Zhang L. Research on high-precision recognition model for multi-scene asphalt pavement distresses based on deep learning. Scientific Reports. 2024;14(1):25416.