applied sciences

MDPI

Article

An Enhanced SL-YOLOv8-Based Lightweight Remote Sensing Detection Algorithm for Identifying Broken Strands in Transmission Lines

Xiang Zhang [1], Jianwei Zhang [2] and Xiaoqiang Jia [3,*]

[1] School of Data Science and Application, Inner Mongolia University of Technology, Hohhot 010051, China; zhangxiang202407@163.com
[2] School of Electric Power, Inner Mongolia University of Technology, Hohhot 010051, China; zjwzachary@outlook.com
[3] School of Information Engineering, Inner Mongolia University of Technology, Hohhot 010051, China
* Correspondence: jiaxiao_01@163.com; Tel.: +86-13848151384

Abstract: Power transmission lines frequently face threats from lightning strikes, severe storms, and chemical corrosion, which can lead to damage in steel–aluminum-stranded wires, thereby seriously affecting the stability of the power system. Currently, manual inspections are relatively inefficient and high risk, while drone inspections are often limited by complex environments and obstacles. Existing detection algorithms still face difficulties in identifying broken strands. To address these issues, this paper proposes a new method called SL-YOLOv8. This method incorporates an improved You Only Look Once version 8 (YOLOv8) algorithm, specifically designed for online intelligent inspection robots to detect broken strands in transmission lines. Transmission lines are susceptible to lightning strikes, storms, and chemical corrosion, which is leading to the potential failure of steel- and aluminum-stranded lines, and significantly impacting the stability of the power system. Currently, manual inspections come with relatively low efficiency and high risk, and Unmanned Aerial Vehicle (UAV) inspections are hindered by complex situations and obstacles, with current algorithms making it difficult to detect the broken strand lines. This paper proposes SL-YOLOv8, which is a broken transmission line strand detection method for an online intelligent inspection robot combined with an improved You Only Look Once version 8 (YOLOv8). By incorporating the Squeeze-and-Excitation Network version 2 (SENet_v2) into the feature fusion network, the method effectively enhances adaptive feature representation by focusing on and amplifying key information, thereby improving the network's capability to detect small objects. Additionally, the introduction of the LSKblockAttention module, which combines Large Selective Kernels (LSKs) and the attention mechanism, allows the model to dynamically select and enhance critical features, significantly enhancing detection accuracy and robustness while maintaining model precision. Compared with the original YOLOv8 algorithm, SL-YOLOv8 demonstrates improved precision recognition accuracy in Break-ID-1632 and cable damage datasets. The precision is increased by 3.9% and 2.7%, and the recall is increased by 12.2% and 2.3%, respectively, for the two datasets. The mean average precision (mAP) at the Intersection over Union (IoU) threshold of 0.5 is also increased by 4.9% and 1.2%, showing the SL-YOLOv8's effectiveness in accurately identifying small objects in complex situations.

Keywords: broken strand detection; YOLOv8; accurate recognition; SENet v2; attention module

check for updates

Citation: Zhang, X.; Zhang, J.; Jia, X. An Enhanced SL-YOLOv8-Based Lightweight Remote Sensing Detection Algorithm for Identifying Broken Strands in Transmission Lines. *Appl. Sci.* **2024**, *14*, 7469. https://doi.org/10.3390/app14177469

Academic Editor: Andrea Prati

Received: 15 July 2024
Revised: 16 August 2024
Accepted: 17 August 2024
Published: 23 August 2024

1. Introduction

As the core component of power grid equipment, transmission and distribution lines are essential in connecting all kinds of equipment. To ensure the safe and reliable operation of transmission and distribution lines, equipment maintenance should be carried out regularly to eliminate potential equipment defects and safety hazards [1]. Manual inspection has been mainly used in power grid equipment operation and maintenance.

However, due to the wide distribution of transmission and distribution line equipment and the harsh operating environment, various failures occur occasionally. In recent years, the use of inspection robots, drones, Unmanned Aerial Vehicles (UAVs), and helicopters has been increasing [2]. However, the cost of helicopter inspection is too high, and safety cannot be guaranteed. UAVs can effectively reduce the workforce and eliminate the interference of human factors, but the detection of broken parts in transmission lines is vulnerable to complex backgrounds and obstacles. With its safety characteristics, high efficiency, accuracy, and reliability, a line patrol robot provides an effective solution for maintaining transmission and distribution line equipment [3]. The inspection robot is equipped with sensors and cameras to locate the transmission line; identify the shock-proof hammer, spacer rod, suspended wire clamp, and other tools on the line; and then execute the corresponding crossing procedure to carry out online inspection, discover obstacles, and collect data [4].

With the continuous deepening and optimization of Convolutional Neural Networks (CNNs) on the convolutional layer, deep learning algorithms based on classification and object detection have also been widely used in hardware detection [5]. Deep learning algorithms mainly extract and learn the features of the obstacle target through evolving deep neural networks, meaning the extracted feature information is independently analyzed, which is not restricted by humans but rather explores various details of interest in the object obstacle [6]. Object detection algorithms based on deep learning can be roughly divided into one-stage and two-stage algorithms. In the two-stage detection process, features are extracted from the image to generate some candidate regions, and then the candidate regions are classified and positioned to output the location and category information of the target [7]. Examples of typical two-stage detection algorithms include Region-CNN (R-CNN) [8], Fast R-CNN [9], and Faster R-CNN [10]. In one-stage detection, CNN is used to extract image features. The extracted features are transmitted to multiple fully connected layers for target detection, and then the position and category information of the target are directly output. Zheng et al. proposed a live power transmission detection method based on R-CNN, which uses the CNN to extract visual features from the aerial images. This method can accurately detect insulators in different environments and accurately detect the faulty insulators in the image [11]. Li et al. applied the Fast R-CNN to an automatic fish identification system to help marine biologists estimate fish existence and quantity, and effectively understand oceanic geographical and biological environments [12]. Moreover, deep learning was investigated to intelligently detect road cracks, and Faster R-CNN and Mask R-CNN were compared and analyzed. The results show that the joint training strategy was very effective, and that both Faster R-CNN and Mask R-CNN completed the crack detection task when trained with only about 130 images [13]. Common one-stage algorithms are the Single Shot Multibox Detector (SSD) and the You Only Look Once (YOLO) series [14–21]. Various one-stage algorithm models have been utilized for fault detection in transmission and distribution lines. Iyke Maduako et al. studied fault location, detection, and classification in transmission lines based on the SSD model [22]. Wang et al. introduced an enhanced YOLOv5 algorithm, replacing the C3 module in the backbone network with the C2f module and optimizing the loss function. This modification allows for the accurate and effective identification and localization of damage in X-ray images of steel cord conveyor belts [23]. Liu et al. utilized the YOLOv3 algorithm to integrate candidate box extraction, feature extraction, target classification, and target location into a neural network. This integrated approach allows for the automatic detection and identification of defects and faults in insulators, bird's nests, dampers, conductors, towers, and other critical components [24].

In 2022, the Ultralytics team published the source code of the YOLOv8 model [25,26], offering five models with varying scales and channel numbers based on scaling coefficients -n, s, m, l, and x. These models are designed to cater to diverse scenarios, ensuring a balance between real-time performance and accuracy. Since 2023, the YOLOv8 has been extensively utilized across various categories of target detection fields [27].

Nie et al. introduced a lightweight enhanced model derived from YOLOv8 [28]. They incorporated a specialized small target detection layer into the feature fusion network and introduced a Scale Sequence Feature Fusion (SSFF) module to enhance multi-scale feature fusion, enabling the model to capture more gradient paths. These enhancements led to improved accuracy while simultaneously reducing model parameters. Zhang et al. integrated a novel dual self-focusing mechanism into the backbone of YOLOv8. They implemented InnerShape Intersection over Union (IS-IoU) as the boundary box regression loss and refined the feature fusion component. This approach significantly enhanced the detection capability of small targets [29]. The existing YOLO algorithm encounters several challenges in practical applications, primarily including inadequate positional accuracy, limited feature representation capability, and difficulty in handling long-tail distributed data. Due to the constraints of its framework, the YOLO algorithm exhibits low positional accuracy when dealing with small and dense targets, resulting in suboptimal detection performance in complex scenarios. Moreover, YOLO's feature extraction network often struggles to extract crucial information from images in the presence of intricate backgrounds and multi-scale targets, consequently impacting overall detection effectiveness. In real-world scenarios, target categories frequently exhibit long-tail distribution characteristics, wherein a few categories have abundant samples while most categories have limited samples. This characteristic makes it easy for the YOLO to overlook categories with fewer examples when processing long-tail distribution data. To address these challenges, some researchers have enhanced the YOLO algorithm by refining the loss function and introducing a new feature extraction module to enhance its detection accuracy and robustness.

To solve the above problems, this paper proposes an improved SL-YOLOv8 algorithm based on YOLOv8, which adds SENet_v2 and LSK modules to the feature detection head of the YOLOv8 model. SENet_v2 augments the YOLO algorithm's feature representation capabilities by incorporating an attention mechanism. Specifically, SENet_v2 enhances detection performance for small targets and intricate backgrounds by adaptively recalibrating channel relationships, directing the network to prioritize important features. The LSK module elevates YOLO's ability to detect objects of varying scales through the utilization of a large convolutional kernel in feature extraction. Large convolutional kernels can capture a broader context range, significantly enhancing target detection accuracy, especially in scenarios involving multi-scale targets. With these enhancements, SL-YOLOv8 outperforms the original YOLO model across multiple benchmark datasets, particularly excelling in small target detection and complex scenes, showing higher precision and robustness.

2. Methods and Data

2.1. YOLOv8 Network Models

YOLOv8 is a deep neural network based on a one-stage object detection algorithm proposed by Ultralytics. It is partially improved on the previous successful YOLO version, and its performance has shown significant advantages in accuracy, ease of use, hardware support, and versatility. It is the first choice for many tasks, such as attitude estimation, instance segmentation, multi-target tracking, and feature extraction.

YOLOv8 provides a new state-of-the-art (SOTA) model, which includes P6 640 and P6 1280 resolution target detection network and Yolact-based instance segmentation models, and provides different n, s, m, l, and x scale models according to different scaling coefficients, to meet the needs of different scenarios.

YOLOv8 is mainly composed of three parts: a backbone network, neck network, and task head, as shown in Figure 1. The backbone network of YOLOv8 still adopts the idea of a Cross Stage Partial Network (CSP), but the C3 module used in YOLOv5 is replaced by the C2f module. The C2f module is based on the C3 module. In addition to ensuring a lightweight design, the model can obtain more abundant gradient flow information. This improvement helps the model to transfer the gradient more effectively during the training process, and reduces the calculation and the number of parameters, thus speeding up the training process and improving the performance of the model. The Path Aggregation

Network (PANet) is used to fuse the feature maps of different layers from top to bottom, which makes feature fusion more efficient. The decoupling head structure is used to separate the detection and classification tasks and accelerate the convergence of the model.

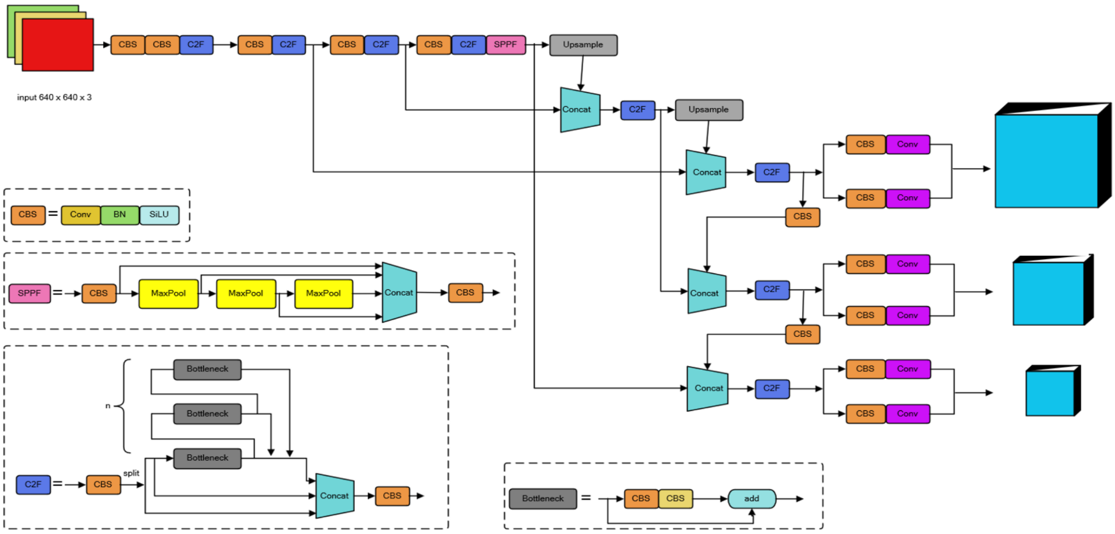


Figure 1. Architecture diagram of the YOLOv8.

YOLOv8 uses Binary Cross-Entropy (BCE) as a classification loss function:

$$\mathrm{L_{total}} = -\frac{1}{\mathrm{N}}\sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}}[\mathrm{y_i} \cdot \log(\mathrm{p_i}) + (1 - \mathrm{y_i}) \cdot \log(1 - \mathrm{p_i})] \quad (1)$$

where $\mathrm{L_{total}}$ represents the total loss, N represents the total number of samples, n represents the number of samples used to calculate the loss, $\mathrm{y_i}$ represents the actual label of the ith sample, and $\mathrm{p_i}$ represents the probability that the ith sample is predicted to be class 1.

The BCE is a loss function commonly used in machine learning to deal with binary classification problems, such as whether there is an animal in the photo, or whether the animal in the picture is a cat or a dog. In YOLOv8, this function is used to evaluate the difference between the probability of target existence predicted by the model and the real label, thereby helping to optimize the model parameters and improve target detection accuracy. In a binary classification problem, $\mathrm{y_i}$ is usually either 0 or 1, where 0 represents a negative class (or “not” the target class) and 1 represents a positive class (or “yes” the target class). $\log(\mathrm{p_i})$ and $\log(1 - \mathrm{p_i})$ are the logarithms of the probabilities predicted by the model. $\mathrm{p_i}$ is the probability (usually between 0 and 1) that the model predicts that the ith sample will be positive. $1 - \mathrm{p_i}$ is the probability that the model predicts that the sample is of a negative class. By taking the logarithm of these probabilities, we can measure the uncertainty of the model’s predictions.

Compared to previous versions, YOLOv8 uses a deeper and more complex network structure and improved training techniques to significantly improve detection accuracy while maintaining high speeds. It uses a more advanced feature extraction network, which helps to extract richer and more distinguished features from images. At the same time, the improved multi-scale prediction technology is introduced, which can better detect objects of different sizes, and the adaptive adjustment of the anchor frame is also optimized, which can predict the position and size of the object more accurately. So, we chose the YOLOv8 network as the basic model.

2.2. New Algorithmic Ateps

2.2.1. SENet_v2 Module

As shown in Figure 2, SENet is an innovative architecture designed to enhance the model representation capabilities of CNNS. By explicitly modeling the dependencies between channels [30], SENet introduces the concept of an "attention mechanism", which achieves significant results in tasks such as image classification and target detection. The core idea of SENet is to enhance the relationship between channels represented by features in CNN through "SE blocks". The SE block includes two key operations: Firstly, the squeeze operation is used to conduct global pooling on the feature map in the spatial dimension to generate a global feature description, assuming that the size of the input feature map is (H × W × C) and the length of the feature vector obtained after pooling is ©. The formula is as follows:

$$Z_c = \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} x_{i,j,c}}{H \cdot W} \quad (2)$$

where Z_c represents the global average pooled value of channel c, $x_{i,j,c}$ represents the pixel value of the position (i, j) on channel c, H represents the height of the feature map, and W represents the width of the feature map. The excitation operation is used to perform a nonlinear transformation of the global feature vector generated by squeeze. Usually, two fully connected layers and a sigmoid activation function are used. The generated weight vector is used to reweight each channel of the original feature map. Through these two operations, SENet can dynamically adjust the weight of each channel according to the global information of the input image, thereby improving the feature representation capability of the network.

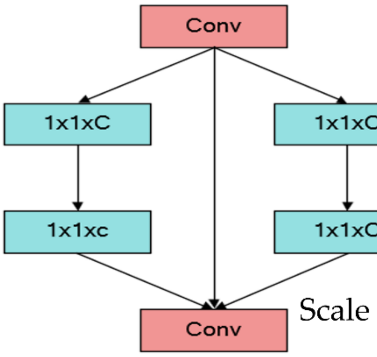


Figure 2. Architecture diagram of SENet_v2.

SENet_v2 is an improved version of the original SENet that further optimizes the effectiveness and efficiency of the attention mechanism. In terms of a more efficient attention mechanism, SENet_v2 introduces a new computational method that makes the reweighting of channels more efficient, improving the model's efficiency by reducing computational complexity and memory usage. For the excitation operation, SENet_v2 optimizes the nonlinear transformation by adding more diverse activation functions and a deeper full-connection layer network to capture the more complex channel dependencies. The activation function in the SE block is Rectified Linear Unit (ReLU). The full-connection layer calculation is as follows:

$$S = \sigma(W_2 \cdot \delta(W_1 \cdot Z)) \quad (3)$$

where δ represents the ReLU activation function, σ represents the sigmoid activation function, and W_1 and W_2 are the weights of the fully connected layer and the global eigenvector.

In addition, the SENet_v2 features a lightweight design, primarily for use in resource-constrained mobile devices and embedded systems, employing quantization and pruning techniques that enable the model to significantly reduce compute and storage costs while maintaining performance. In the architectural design, the SENet_v2 is designed to be similar to the original SENet, but with improvements in details. The core of the SE block is still squeeze and excitation operations, but the implementation of the various stages is optimized. For example, in the squeeze operation, to reduce information loss in spatial dimensions, SENet v2 introduces Global Adaptive Pooling (GAP), which is better at capturing multi-scale features than fixed global average pooling.

In the excitation phase, SENet_v2 introduces a more complex nonlinear transformation, including multiple fully connected layers and different activation functions (e.g., ReLU, Gaussian Error Linear Unit, etc.) to better adapt to different mission requirements. In addition, during the excitation phase, SENet_v2 incorporates regularization techniques, such as dropout and batch normalization, to prevent overfitting to improve model generalization. In terms of application and performance, SENet_v2 performs well in several computer vision tasks, including image classification, object detection, image segmentation, and more.

In addition, its lightweight design makes it ideal for deployment in mobile devices and embedded systems. By optimizing computing and storage requirements while ensuring high performance, SENet_v2 can effectively operate under limited resources, so it has a wide range of applications in edge computing, real-time image processing, etc. SENet_v2 proposes a more efficient and powerful attention mechanism by improving the original SENet. While maintaining the advantages of the original model, SENet_v2 further improves the modeling efficiency of inter-channel relationships, increases the expressive power of the network, and significantly improves the computational efficiency through lightweight design. SENet_v2 provides a more efficient solution for a variety of computer vision tasks and has a wide range of academic research and practical application value.

2.2.2. LSK Module

In remote sensing image processing, most target detectors often result in classification errors due to the limited context information. Moreover, there may be significant differences in the amount of context information required to detect different remote sensing objects, which may decrease the model's accuracy. Although the CNNS has achieved great success in computer vision tasks such as image classification, object detection, and semantic segmentation, traditional convolutional layers struggle to adequately capture objects of different scales and shapes due to their fixed-size receptive fields. Although large nuclear convolution can extend the receptive field, it is computationally expensive and ineffective when dealing with small targets.

To better adapt to the needs of multi-scale target feature extraction and improve the precision of the resulting model, we use an LSK module [31], which has a flexible receptive field, enhanced spatial selectivity, rich feature expression, and computational efficiency. Through multi-scale feature fusion, the receptive field size can be dynamically adjusted, which makes the model more flexible when dealing with various scale targets. The generated spatial selection mask enables the model to focus on the most important features at specific spatial locations, thus improving the accuracy of target detection. The information interaction and fusion of multi-scale features improve the richness of feature expression and enhance the discriminant ability of the network. In addition, the LSK module achieves higher computational efficiency by splitting the convolutional kernel and the attention mechanism compared to the direct use of the large convolutional kernel. The LSK module is a new structure for CNN, as shown in Figure 3. It aims to significantly improve the feature extraction and target detection capability of the model in complex scenes through multi-scale feature fusion and selective spatial attention mechanisms.

The LSK module enhances the flexibility and discriminative ability of feature expression through the multiscale convolutional kernel and selective spatial attention mechanism. First, the LSK module extracts multi-scale features through convolution kerns of different sizes (e.g., 3 × 3, 5 × 5, 7 × 7), which capture contextual information of different scales to form preliminary multi-scale feature maps. This process can be expressed as:

$$\mathrm{F_s} = \{\mathrm{f}_{3\times3}(\mathrm{X}), \mathrm{f}_{5\times5}(\mathrm{X}), \mathrm{f}_{7\times7}(\mathrm{X})\} \tag{4}$$

where $\mathrm{F_s}$ represents a preliminary set of multi-scale feature graphs, and $\mathrm{f}_{\mathrm{k}\times\mathrm{k}}(\mathrm{x})$ represents the convolution operation of a convolution kernel of size $\mathrm{k} \times \mathrm{k}$ on input x.

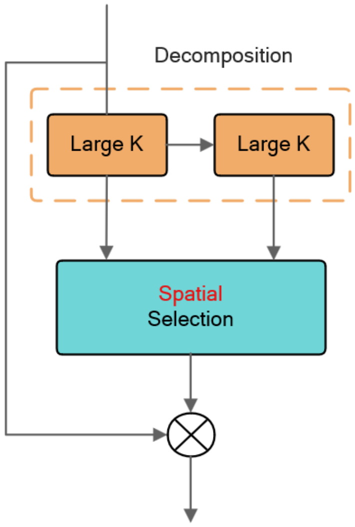


Figure 3. Architecture diagram of LSK. The part framed by the dotted line represents the large convolution kernel.

Channel average pooling and maximum pooling operations are applied on each feature map to obtain the average and maximum response of the channel, respectively. These two pooling operations can effectively summarize and highlight the spatial relationship in the feature map, and the process can be expressed as follows:

$$F_{avg} = \mathrm{AvgPool}(F_s), F_{max} = \mathrm{MaxPool}(F_s) \tag{5}$$

where F_{avg} and F_{max} represent the feature maps after channel average pooling and Max pooling, respectively.

Then, pooling features at different scales are fused through a convolutional layer to generate multiple spatial attention maps. This process can be regarded as the integration of information at different scales to obtain a more comprehensive feature representation. This fusion process can be expressed as follows:

$$F_{att} = \sigma\left(W\left(F_{avg} \bigoplus F_{max}\right)\right) \tag{6}$$

where F_{att} represents the generated spatial attention map, W represents the weights of the convolutional layer, and σ represents the number of sigmoid activation functions.

A sigmoid activation function is used on each spatial attention map to generate a corresponding selection mask. Each mask is used to selectively enhance or suppress the feature response at a specific spatial location. The multi-scale features are weighted by the spatial selection mask, and then fused again by the convolutional layer to generate the final attention features. This step allows the model to further strengthen the most important feature information based on the input features. This weighting process can be expressed as:

$$F_{weighted} = F_s \bigotimes F_{att} \tag{7}$$

where $F_{weighted}$ represents weight features, F_s represents spatial features, and F_{att} represents attention features.

Finally, the output of the LSK module is the element-wise product of the input features and the attention features, which ensures that the final output features contain both the original input information and the multi-scale selective attention information. The final output can be expressed as follows:

$$F_{output} = X \bigotimes F_{weighted} \tag{8}$$

where $\mathrm{F_{output}}$ represents the output feature, X represents the input feature, and $\mathrm{F_{weighted}}$ represents the weight feature.

In this way, the LSK module achieves effective fusion and selective enhancement of multi-scale features, thus improving the flexibility and accuracy of the model when dealing with objects of different scales.

2.2.3. SL-YOLOv8

As shown in Figure 4, SL-YOLOv8 (SENet_v2-LSK-YOLOv8) draws into the SENet_v2 and LSK modules in the feature detection head of the YOLOv8 model. After the output of each feature layer (P3, P4, P5), the SENet_v2 and LSK modules are embedded successively. The SENet_v2 enhances the extraction of key features by adaptively recalibrating the importance of feature channels; The LSK module improves the multi-scale adaptability of the network by dynamically adjusting the size of the receptive field. The synergy of these two modules enables SL-YOLOv8 to more effectively fuse the high-dimensional semantic information of deep feature maps with the detailed information of shallow feature maps, enhancing the detection ability of targets for different scales.

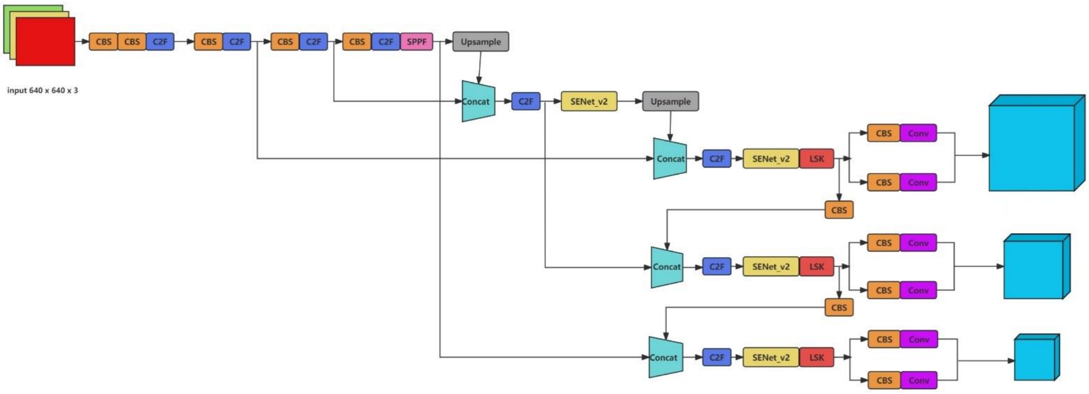


Figure 4. Architecture diagram of the SL-YOLOv8.

When analyzing the computational complexity of the original YOLOv8, we divide it into two parts: the backbone network and the detection head. The backbone network mainly consists of convolutional and downsampling layers, and the computational complexity of each layer is dominated by convolution operations, assuming that the input image size is $N \times N$, The complexity is $O(K_2 \cdot C_{in} \cdot C_{out} \cdot N_2)$, K is the convolution size, and C_{in} and C_{out} are the number of input and output channels, respectively. The detection head is mainly composed of multi-scale feature layers, convolutional layers, and fully connected layers, and its computational complexity is mainly determined by the generation of multi-scale features and the final prediction. In the optimized SL-YOLOv8 network structure, SENet v2 and LSK modules are added, and its computational complexity also increases accordingly. The SENet v2 module mainly involves the global pooling and fully connected layer, and the complexity is $O(C_{out2})$. The LSK module uses separable convolutions to increase the receptive field with a complexity of $O(K \cdot C_{in} \cdot C_{out} \cdot N_2)$. By comprehensively analyzing the computational complexity of the backbone network and the detection head part, the added complexity of the backbone network can be considered $O\Big((N_2 \cdot K_2 \cdot C_{in} \cdot C_{out})_{original} + C_{out2}\Big)$, and the added complexity of the detection header can be expressed as $O((K \cdot C_{in} \cdot C_{out} \cdot N_2 \cdot |HeadComplexity|) + (LSK \cdot C_{in} \cdot C_{out}) + (SENetv2 \cdot C_{out2}))$. Despite the additional computational complexity introduced by the new module, LSK reduces standard convolution calculations, partially offsetting the complexity of SENet v2. The overall complexity is expected to increase, but through feature enhancement and cross-layer feature aggregation, the overall feature expression and detection capabilities are improved, and the complexity is about $O((K \cdot C_{in} \cdot C_{out} \cdot N_2 \cdot |MainBackboneComplexity|) + C_{out2} + LSKComplexity)$. With the introduction of the new module, the expected increase in inference time is about 10–20%

of the original YOLOv8 model, and the performance improvement is about 2–5%, possibly higher for specific tasks. In summary, by introducing the SENet v2 and LSK modules, SL-YOLOv8 greatly improves computational efficiency and detection performance through lightweight design and efficient feature expression, despite the additional computational complexity, especially in small targets and complex background tasks (8).

2.3. Data and Experimental Environment

2.3.1. Dataset

In this work, two datasets are used for the evaluation and verification of the improved model: namely, the Break-ID-1632 image dataset (Figure 5) and the cable damage dataset (Figure 6). The Break-ID-1632 image dataset is obtained by installing a high-definition camera on an intelligent patrol robot to capture real-time photos on high-voltage transmission lines. There are 1814 photos, with 1632 training photos and 182 test photos. The pixel size is 800 × 600, and it includes nine categories: normal, damaged, one broken strand, two broken strands, three broken strands, four broken strands, five broken strands, six broken strands, and seven broken strands, covering a wide range of categories, including location, environment, object, etc. The uniqueness of this dataset lies in the following aspects: (1) The dataset contains a wealth of real-world images, covering grasslands, forests, and snowy environments, effectively simulating real-world target detection and tracking scenarios, and (2) the annotation information is exceptionally detailed, meticulously recording the location, size, and shape of the target, providing precise reference standards for algorithm research. It is also worth mentioning that the Break-ID-1632 dataset covers images under different weather and lighting conditions, as well as various angles and height perspectives, which helps the algorithm to be applied and evaluated in various complex backgrounds. The Break-ID-1632 image dataset is a self-made dataset that can be widely used in fields such as target detection, tracking, and remote sensing. It can be used to verify and optimize algorithms, and it can also be compared with other algorithms. Meanwhile, these data provide important reference resources for high-voltage power transmission line inspection. Other datasets include the cable damage dataset, which has 1318 photos, along with training sets and test sets consisting of 1187 and 131 photos, respectively, including burned and broken categories. Figures 5 and 6 show partial images of the Break-ID-1632 and cable damage datasets, respectively.

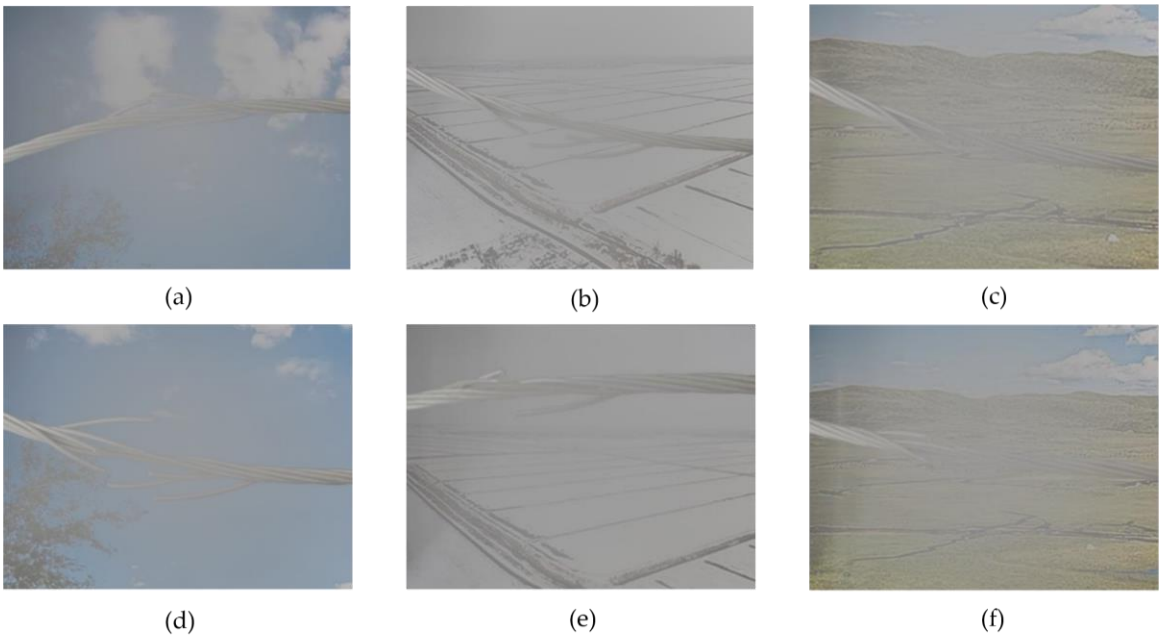
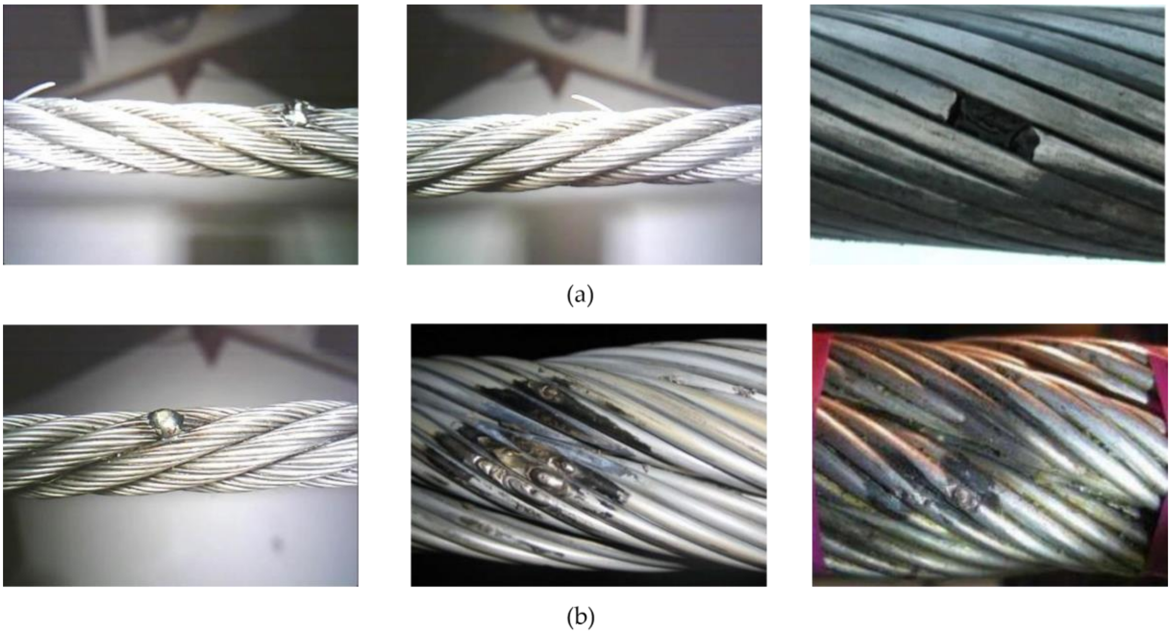


Figure 5. The Break-ID-1632 image dataset. (**a**) Outdoor-normal; (**b**) snow-normal; (**c**) grassland-normal; (**d**) outdoor-broken strand; (**e**) snow-broken strand; (**f**) grassland-broken strand.



(a)

(b)

Figure 6. The cable damage image dataset. (**a**) Broken strand; (**b**) burning wires.

2.3.2. Experimental Environment

The study of this experiment uses the PyTorch deep learning framework and relies on a separate NVIDIA GeForce RTX 3050 graphics card when training the model. The specific configuration of the experimental environment is shown in Table 1.

Table 1. Experimental environment configuration.

Item	Name
Operating system	Windows 11
CPU	11th Gen Intel® Core™ i7-11800H
GPU	NVIDIA GeForce RTX 3050
RAM	16.0 GB
Deep learning framework	PyTorch (2.1.0)
Interpreter	Python 3.9
CUDA version	Cuda (11.8)

3. Experiment and Analysis

3.1. Evaluation Index

In object detection, precision (P), recall (R), mean average precision (mAP), and balanced F score (F1 score) are the key metrics to evaluate the performance of an algorithm. The precision measures the fraction of samples that the model predicts to be positive and represents the accuracy and reliability of the model on samples classified as positive. The recall measures the proportion of all true positive instances successfully detected by the model that are correctly labeled as positive and represents the recognition ability and coverage of the model for positive instances. The specific formulas for P and R are as follows:

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{9}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{10}$$

where true positive (TP) refers to the number of positive samples that were correctly classified as positive by the model, that is, the number of positive instances that were correctly detected by the model. False positive (FP) refers to the number of negative instances that are misclassified as positives by the model, i.e., the number of negative instances that the model misclassifies as positive. False negative (FN) is the number of positives that the model misclassifies as negative, i.e., the number of positive instances that the model fails to detect. Average precision (AP) measures the model's performance on a single class. It is the area under the precision–recall (P-R) curve, which reflects the average precision of the model at different recall rates. AP is expressed as follows:

$$\mathrm{AP} = \int_0^1 \mathrm{PRDR} \tag{11}$$

The mAP is a commonly used evaluation metric in object detection, which comprehensively considers the performance of multiple categories. First, for each class, we compute the area under the P-R curve, called AP, which reflects the tradeoff between the model's accuracy and recall on a single class. Then, the AP of all categories is averaged to obtain the mAP so that the model's performance can be evaluated. The formula is as follows:

$$\mathrm{mAP} = \frac{\sum_{\mathrm{K}=1}^{\mathrm{n}} \mathrm{AP}_{\mathrm{K}}}{\mathrm{TP} + \mathrm{FN}} \tag{12}$$

The F1 score is another important evaluation metric, which combines information from precision and recall. The F1 score is the harmonic mean of precision and recall and provides a balance between these two metrics, making it particularly useful for evaluating model performance on imbalanced datasets. The F1 score ranges from 0 to 1, with values closer to 1 indicating better model performance. The F1 score is calculated as follows:

$$\mathrm{F1} = 2 \times (\mathrm{Precision} \times \mathrm{Recall}) / (\mathrm{Precision} \times \mathrm{Recall}) \tag{13}$$

3.2. Ablation Experiment

To verify the effectiveness of the three improvement points proposed in this paper, YOLOv8 is selected as the baseline model for comparison, and the Break-ID-1632 and cable damage datasets are trained and tested in the same experimental environment. In the training process, a total number of training rounds of 120 epochs is set, and the input image size is uniformly set to 600 × 600 pixels. In the experiments, the performance of each improved module is evaluated, including their impact on object detection accuracy, such as mAP, training stability, and convergence speed.

Table 2 lists the experimental results of training and testing on the Break-ID-1632 dataset, where "√" is used to indicate that the module is added to YOLOv8, and "×" indicates that it is not added. The test data of each module are compared to analyze their contribution to the model performance. These data include mAP values under different improvement modules, precision, recall for each category, and possible training time and resource consumption. Through these detailed experiments and results analysis, the impact of each improvement module on the overall object detection model's performance can be comprehensively evaluated to verify whether the proposed improvement points can effectively improve the accuracy and efficiency of the model.

Table 2. Results of ablation experiments on the Break-ID-1632 dataset.

YOLOv8	LSK	SENet_v2	Precision%	Recall%	mAP@0.5%	F1%
√	×	×	80.7	82.4	89.9	81.48
√	√	×	83.5	90.8	90.5	86.98
√	×	√	81.9	92.7	93.5	87.08
√	√	√	84.6	94.6	94.1	89.28

Figure 7 shows the precision, recall, and mAP@0.5 curve of the YOLOv8 model after adding SENet_v2 and LSK modules in detail. Comparison with the basic YOLOv8 model clearly shows the effectiveness of the three improvements on the performance of the model. These curves and metrics provide an intuitive understanding of the performance of the improved modules under different conditions, thus validating their substantial contribution to the object detection task.

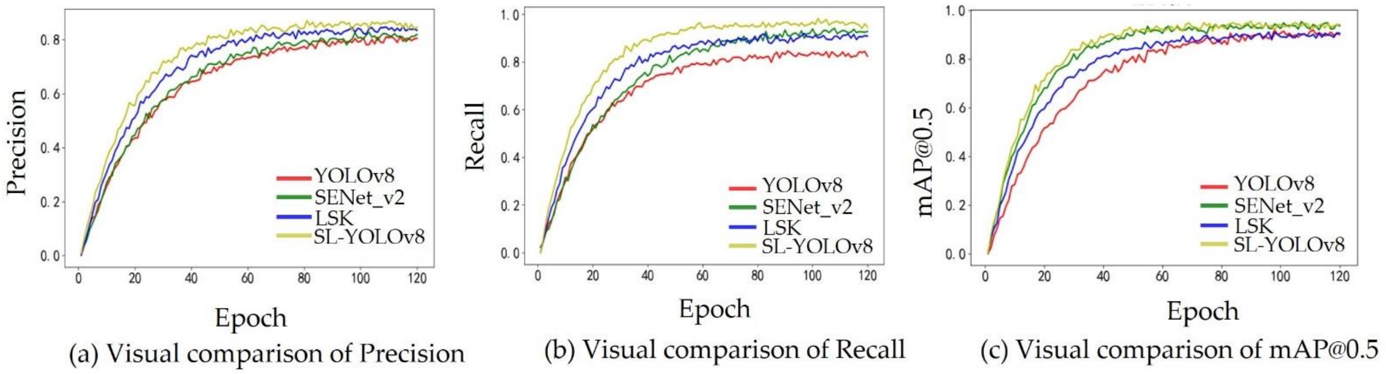


Figure 7. Visual comparison of the Break-ID-1632 dataset.

Table 3 shows an overview of the training and testing experiments conducted on the cable damage dataset, where "√" indicates that a module has been integrated into the YOLOv8 model, and "×" indicates that the module has not been adopted. The performance of each module is evaluated by comparing its performance on the test set. These evaluation metrics include the average precision under different improvement schemes, the precision and recall of each class, as well as the estimated training time and computing resource consumption. Through this series of detailed experiments and analysis, it is possible to comprehensively evaluate the contribution of each improvement module to the performance of the object detection model to verify whether the improvement measures successfully improve the accuracy and efficiency of the model.

Table 3. Results of ablation experiments on the cable damage dataset.

YOLOv8	LSK	SENet_v2	Precision%	Recall%	mAP@0.5%	F1%
√	×	×	91.1	88.4	91.1	81.55
√	√	×	91.6	89.3	91.8	86.76
√	×	√	92.7	89.2	91.5	86.73
√	√	√	93.8	90.7	92.3	89.30

Figure 8 shows the precision, recall, and mAP@0.5 curve after introducing the SENet_v2 and LSK modules into the YOLOv8 model in detail. Compared with the original YOLOv8 model, it clearly shows the improvement effect of these two improvements on the performance of the model. These curves and metrics provide an intuitive understanding of the performance of the improved modules under different conditions, thus validating the substantial contribution they bring to the object detection task.

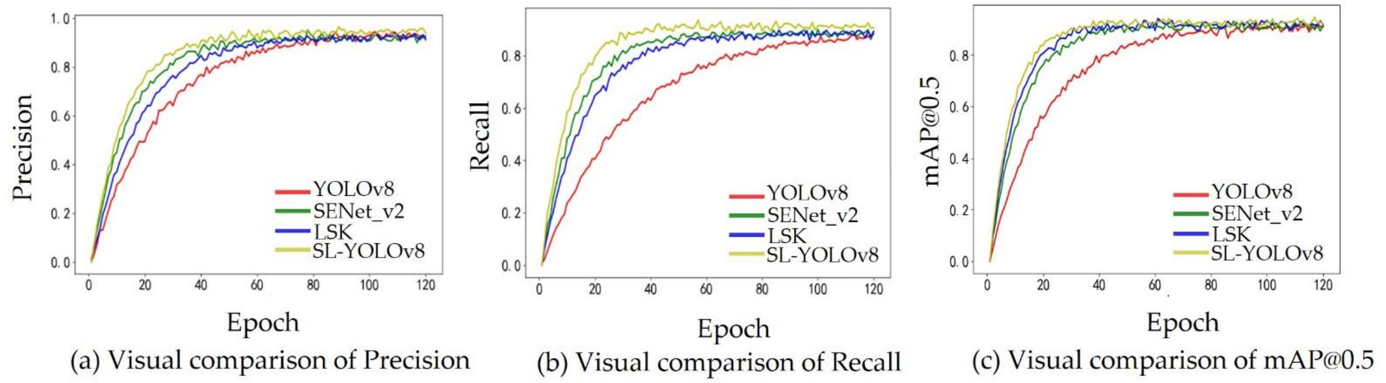


Figure 8. Visual comparison of the cable damage dataset.

Experimental verification shows that the method proposed in this paper significantly improves the detection accuracy of small targets in complex backgrounds. For the Break-ID-1632 dataset, when SENet_v2 is introduced into YOLOv8n alone, compared with the original algorithm, the precision is improved by 1.2%, and the recall is improved by 10.3%. That is an increase of 3.6 percent mAP@0.5. When the LSK module is introduced alone, the precision accuracy is improved by 2.8%, the recall accuracy is improved by 8.4%, and the mAP@0.5 accuracy is improved by 0.6%. After introducing the SENet_v2 and LSK module at the same time, the precision is 3.9% higher than the original algorithm, the recall is 12.2% higher than the original algorithm, and the mAP@0.5 is 4.2% higher than the original algorithm. The detection accuracy of various small targets is improved.

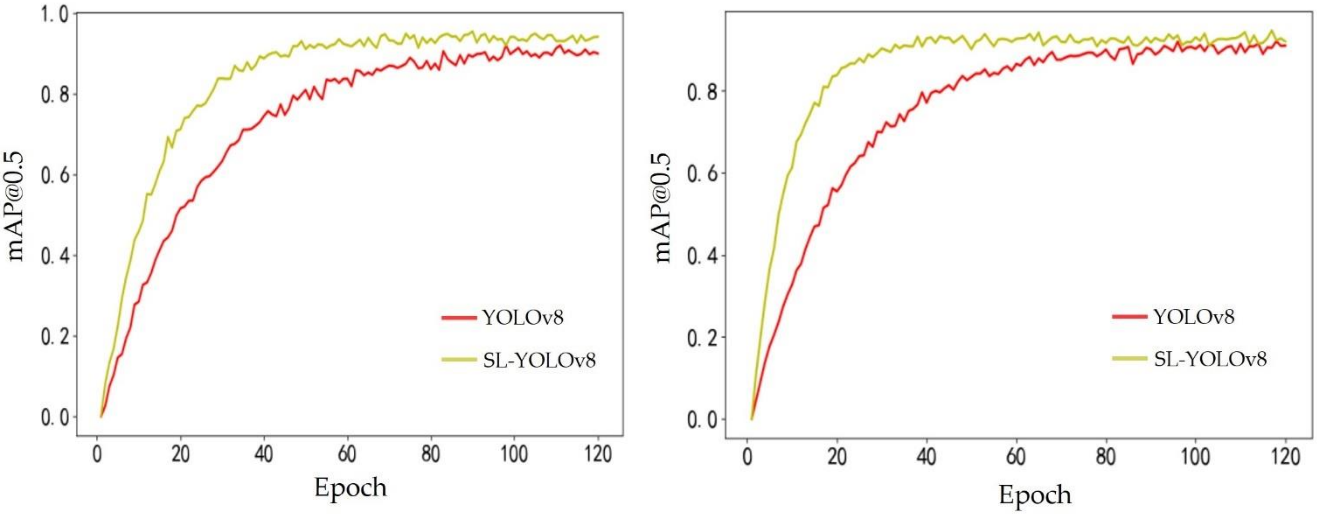
For the cable damage dataset, introducing SENet_v2 alone into YOLOv8n improves the accuracy by 1.6%, the recall accuracy by 0.8%, and the mAP@0.5 accuracy by 0.4%. Introducing the LSK module alone improves precision by 0.5%, recall by 0.9%, and mAP@0.5 by 0.7% compared to the original algorithm. After introducing the SENet_v2 and LSK module at the same time, the precision is increased by 2.7%, the recall is increased by 2.3%, and the mAP@0.5 is increased by 1.2%.

These research results show that the introduced SENet_v2 and LSK modules effectively enhance the model's ability to extract fine-grained features from remote sensing images, thereby improving the overall detection accuracy of the model. The SENet_v2 module can capture the important feature information in the image more accurately through its channel attention mechanism. The experimental results show that the mAP of the model is increased by 3.6% after adding SENet_v2. This indicates that SENet_v2 is essential in enhancing the feature representation ability and detection performance. The LSK module effectively expands the receptive field of the model and enhances the recognition ability of complex patterns in images through its large convolution kernel structure. The experimental data show that the introduction of the LSK module improves the average detection accuracy of the model by 0.6%. This improvement significantly optimizes the model's performance when dealing with high-resolution remote sensing images. These improvements do not significantly increase the number of parameters and storage requirements of the model, and maintain the lightweight characteristics of the model.

The introduction of the SENet_v2 and LSK module significantly improves the feature extraction ability and detection accuracy of the model, which verifies the effectiveness of these improvements in remote sensing image analysis. The comparison curves of the experimental results between the original algorithm and the improved algorithm on the Break-ID-1632 and cable damage datasets are shown in Figure 9. As the training epoch increases, mAP@0.5 of the SL-YOLOv8 algorithm gradually increases relative to the YOLOv8 algorithm.

Figure 10 shows the comparison of the visual results of YOLOv8 and the SL-YOLOv8 in this paper on the Break-ID-1632 dataset and cable damage dataset. The results show that the proposed model significantly reduces the false detection and missed detection, and has higher recognition accuracy. Through comparison, it can be found that the improved model not only performs well in fine-grained feature extraction but also has a significant improvement in overall detection performance.

In the case of low false alarm rate (e.g., 0.00059), the recall of the model is low, only 0.01778, indicating that the model can detect very limited positive class samples at extremely low false alarm rates. This is usually because the model makes positive predictions when it is extremely certain, which reduces false alarms, but also leaves many positive examples unrecognized. For example, when the false alarm rate is 0.22616, the recall rate increases to 0.22616. When the false alarm rate is 0.65669, the recall rate increases to 0.24671, which indicates that the sensitivity of the model improves with the increase in the false alarm rate and more positive samples can be detected, but more false positives are introduced at the same time. However, the performance of the model is more challenging at very low false alarm rates and the recall decreases rapidly, indicating that in practice, the model's ability to detect the positive class will be significantly limited if the system needs to operate at very low false alarm rates. Therefore, at low and very low false alarm rates, the model exhibits low recall and has difficulty with effectively detecting positive class samples. This performance characteristic is suitable for application scenarios where the cost of false alarms is high and highly accurate detection is required. To improve model performance under these conditions, we may need to incorporate other techniques, such as additional training data, multi-model ensembles, or post-processing steps, to increase recall at very low false alarm rates while maintaining a low false alarm rate.



(a) Comparison of the Break-ID-1632 dataset

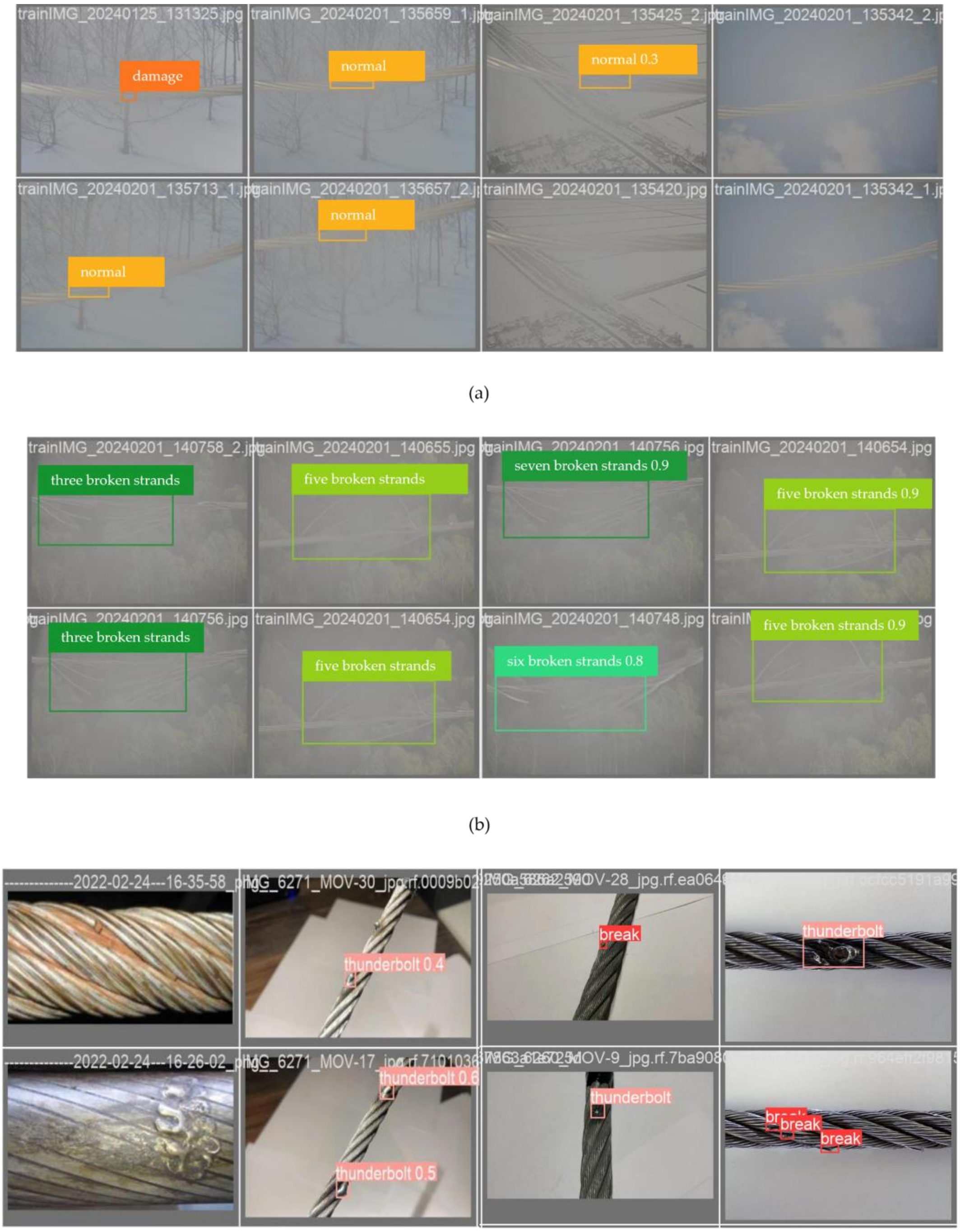
(b) Comparison of the cable damage dataset

Figure 9. Comparison of mAP@0.5 before and after improvement.

This study deeply explores the impact of the sample imbalance problem on the classification performance of machine learning models, especially when it comes to small object detection or dealing with scarce class data. The results show that the sample imbalance significantly reduces the classification accuracy of the model; especially in the case of insufficient samples in a few categories, the classification error rate is significantly increased. This phenomenon is mainly due to the fact that the model is more inclined to learn the majority category, resulting in insufficient ability to effectively classify the minority category.

To alleviate the above problems, increasing the number and diversity of training samples has been proven to be effective. However, this study further analyzes the relationship between the increase in sample size and the improvement of classification performance and finds that the decrease in error rate gradually flattens out as the number of samples increases [32]. This indicates that although expanding the size of the training set is one of the effective means to improve the performance of the model, simply relying on increasing the sample size may not be enough to fully solve the sample imbalance problem. Combining strategies such as data augmentation, resampling techniques, and adjusting the loss function can more effectively improve the classification performance for minority classes.

In summary, when dealing with the problem of sample imbalance, especially in small object detection or minority class classification applications that require high accuracy, researchers and practitioners should comprehensively consider the number of samples, class distribution, and the model training strategy to improve the overall performance and generalization ability of the model. Future research should further explore how to optimize the classification performance of the model for each category while maintaining the sample balance so as to promote the development of machine learning techniques in practical applications.


trainIMG_20240125_131325.jpg
damage
trainIMG_20240201_135659_1.jpg
normal
trainIMG_20240201_135425_2.jpg
normal 0.3
trainIMG_20240201_135342_2.jpg
trainIMG_20240201_135713_1.jpg
normal
trainIMG_20240201_135657_2.jpg
normal
trainIMG_20240201_135420.jpg
trainIMG_20240201_135342_1.jpg
trainIMG_20240201_140758_2.jpg
three broken strands
trainIMG_20240201_140655.jpg
five broken strands
trainIMG_20240201_140756.jpg
seven broken strands 0.9
trainIMG_20240201_140654.jpg
five broken strands 0.9
trainIMG_20240201_140756.jpg
three broken strands
trainIMG_20240201_140654.jpg
five broken strands
trainIMG_20240201_140748.jpg
six broken strands 0.8
five broken strands 0.9
-------------2022-02-24---16-35-58_png
IMG_6271_MOV-30_jpg.rf.0009b02
thunderbolt 0.4
MOV-28_jpg.rf.ea064
break
thunderbolt
-------------2022-02-24---16-26-02_png
IMG_6271_MOV-17_jpg.rf.7101036
thunderbolt 0.6
thunderbolt 0.5
MOV-9_jpg.rf.7ba908
thunderbolt
break
break
break

(a)

(b)

(c)

Figure 10. Dataset visualization results. (**a**) Visualization of YOLOv8 detection on the Break-ID-1632 dataset. (**b**) The detection visualization results of the SL-YOLOv8 on the Break-ID-1632 dataset. (**c**) Visualization of YOLOv8 detection on the cable damage dataset.

3.3. Comparative Experiment

To further verify the effectiveness of the SL-YOLOv8 on the Break-ID-1632 and cable damage datasets, a variety of the most advanced models are selected for comparison, including the classic YOLO series network, the two-stage algorithm Fast-R-CNN, and other algorithms. The performance of the improved algorithm is evaluated according to the mAP of each class and the mAP value of the overall algorithm.

The specific experimental results of the Break-ID-1632 dataset are shown in Table 4. In terms of detection accuracy, the SL-YOLOv8 performs well, exceeding the current popular SOTA model. Compared with YOLOv7, YOLOv5, and YOLOv6 models, the overall mAP value of the SL-YOLOv8 is increased by 4.6%, 5.6%, and 6.7%, respectively. In addition, the accuracy of the SL-YOLOv8 on each category is significantly higher than that of other models in the same category, and the overall model reaches 94.1% on mAP@0.5. Through these results, the significant improvement of the SL-YOLOv8 in the object detection task is verified, and its superiority in multiple categories and overall detection accuracy is demonstrated. Figure 11 is a visual bar chart of each algorithm mAP@0.5 in Table 4.

Table 4. Results of contrast experiments on the Break-ID-1632 dataset.

Model	Normal	Damage	Break1	Break2	Break3	Break4	Break5	Break6	Break7	mAP@0.5(%)
Faster R-CNN	0.602	0.741	0.841	0.870	0.900	0.880	0.901	0.838	0.874	83.9
SSD	0.552	0.695	0.818	0.826	0.851	0.826	0.844	0.792	0.824	80.3
RetinaNet	0.626	0.774	0.866	0.914	0.933	0.901	0.923	0.865	0.894	84.6
EfficientDet	0.582	0.735	0.859	0.902	0.926	0.894	0.883	0.823	0.877	82.5
CenterNet	0.505	0.685	0.784	0.824	0.852	0.835	0.817	0.766	0.804	75.5
Deformable DETR	0.621	0.767	0.875	0.915	0.921	0.893	0.888	0.847	0.853	83.2
YOLOv5	0.694	0.826	0.914	0.958	0.972	0.936	0.955	0.916	0.945	87.4
YOLOv6	0.709	0.832	0.923	0.963	0.987	0.944	0.969	0.924	0.955	88.5
YOLOv7	0.715	0.849	0.934	0.971	0.942	0.957	0.972	0.933	0.965	89.5
YOLOv8s	0.721	0.857	0.946	0.986	0.926	0.968	0.988	0.945	0.975	90.5
YOLOv8n	0.732	0.865	0.954	0.974	0.944	0.979	0.946	0.956	0.985	91.5
SL-YOLOv8	0.742	0.881	0.971	0.995	0.995	0.991	0.971	0.926	0.995	94.1

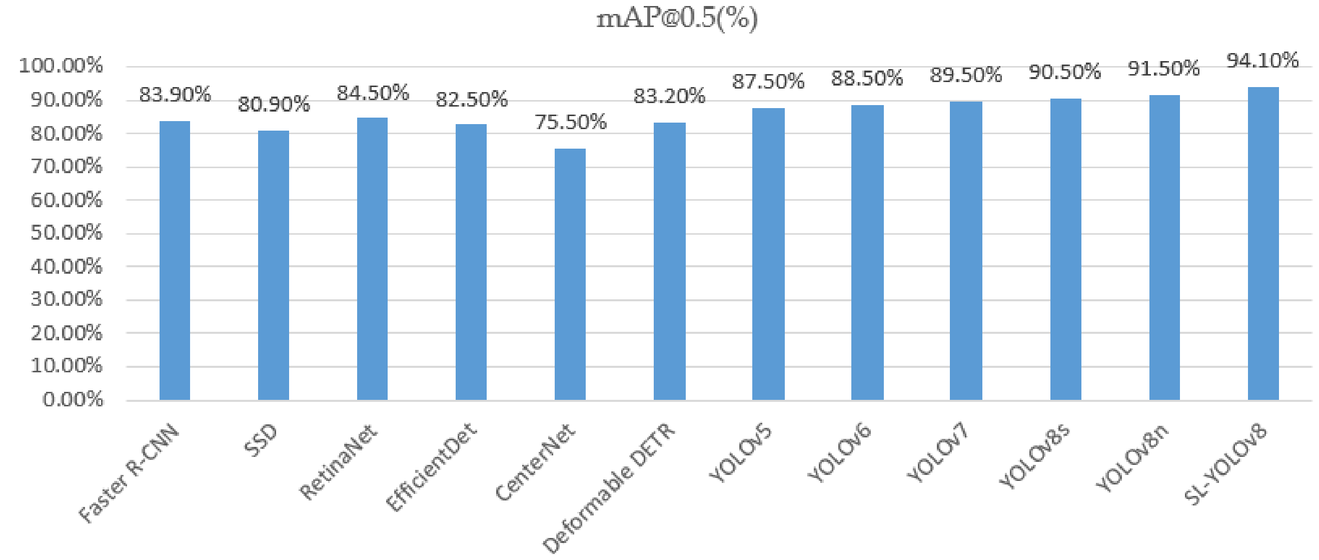


Figure 11. Visualized bar plots of mAP@0.5 corresponding to Table 4.

Table 5 shows the detailed experimental results of the cable damage dataset. In terms of detection accuracy, the SL-YOLOv8 also performs well, exceeding the current mainstream SOTA model. Compared with the YOLOv7, YOLOv5, and YOLOv6 models, the overall mAP value of the SL-YOLOv8 is increased by 3.9%, 4.8%, and 5.6%, respectively. The parameters (para) refer to the parameter quantity, and FLOPs are the floating point operations. In addition, the accuracy of the SL-YOLOv8 in each category is also significantly better than that of other models in the corresponding category, and the overall model reaches 92.3% on mAP@0.5. These results verify the significant improvement of the SL-YOLOv8 in the object detection task, demonstrating its superiority in multiple categories and overall detection accuracy. Figure 12 presents the visual bar plots of each algorithm in Table 5 on the mAP@0.5 metric.

Table 5. Results of contrast experiments on the cable damage dataset.

Model	Break	Thunderbolt	mAP@0.5 (%)	Para (M)	FLOPs (G)
Faster R-CNN	0.75	0.883	82.5	50.0	200.0
SSD	0.70	0.865	80.5	25.0	100.0
RetinaNet	0.77	0.907	84.3	55.0	210.0
EfficientDet-D1	0.75	0.825	83.2	6.6	6.2
CenterNet-DLA	0.73	0.745	78.7	22.5	102.2
Deformable DETR	0.71	0.725	73.0	34.0	188.0
YOLOv5	0.79	0.926	86.7	15.0	30.0
YOLOv6	0.81	0.934	87.8	12.0	28.0
YOLOv7	0.82	0.942	88.4	11.0	27.0
YOLOv8s	0.83	0.955	89.6	10.5	26.5
YOLOv8n	0.84	0.979	90.2	10.2	25.5
SL-YOLOv8	0.85	0.983	92.3	10.0	25.0

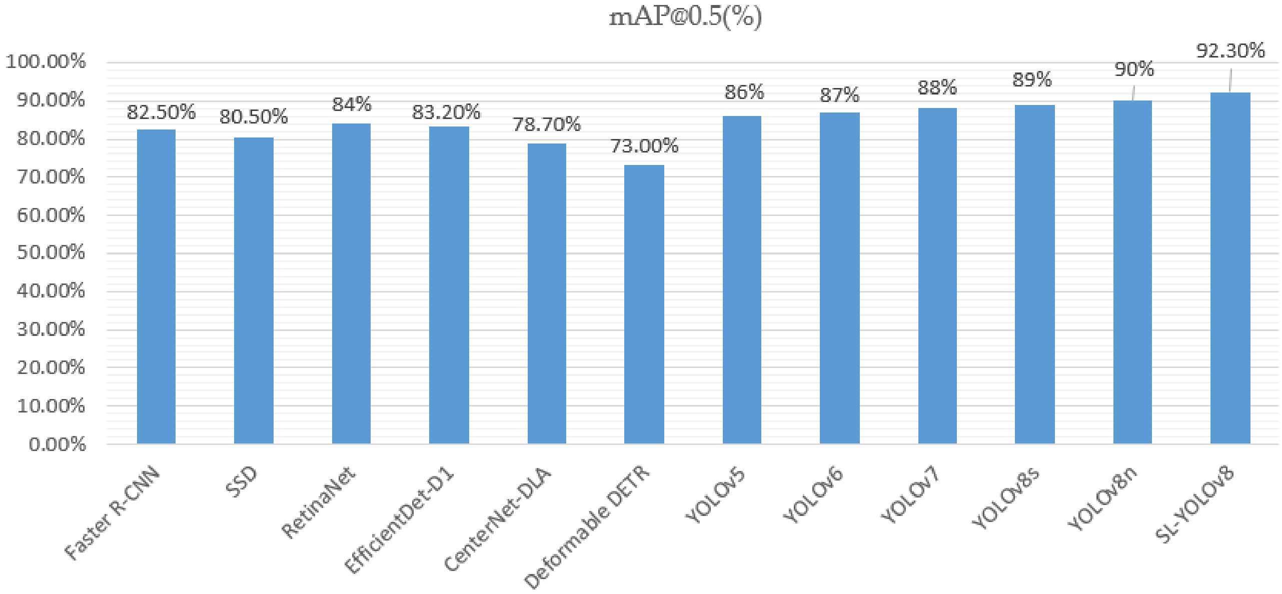


Figure 12. Visualized bar plots of mAP@0.5 corresponding to Table 5.

The above results verify the significant improvement of the SL-YOLOv8 in the object detection task, and its superiority in multiple categories and overall detection accuracy is demonstrated.

Figures 13 and 14 show the detailed evaluation results of the improved model on the Break-ID-1632 and cable damage datasets, covering the analysis of multiple important indicators, including precision, recall, P-R curve, and harmonic mean. It can be observed from the figure that the SL-YOLOv8 has superior performance on all the indicators, especially achieving extremely high detection accuracy while maintaining high recall. Further comparative analysis shows that compared with the basic model, the SL-YOLOv8 has a significant improvement in the accuracy of prediction results; especially in the target detection task under a complex background, the performance is particularly prominent. This result verifies the effectiveness and practicability of the proposed SL-YOLOv8 and provides strong support for subsequent research and practical application.

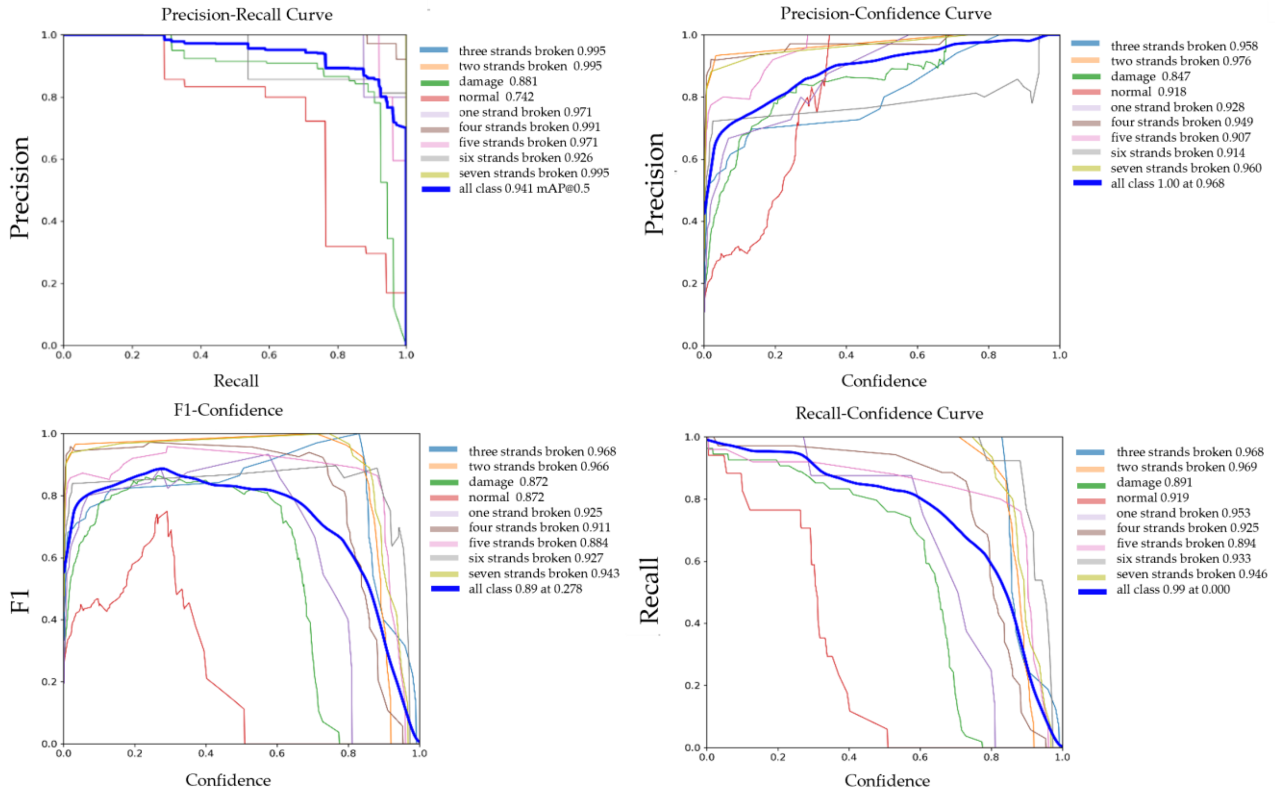


Figure 13. Visualization of the evaluation parameters for the Break-ID-1632 dataset.

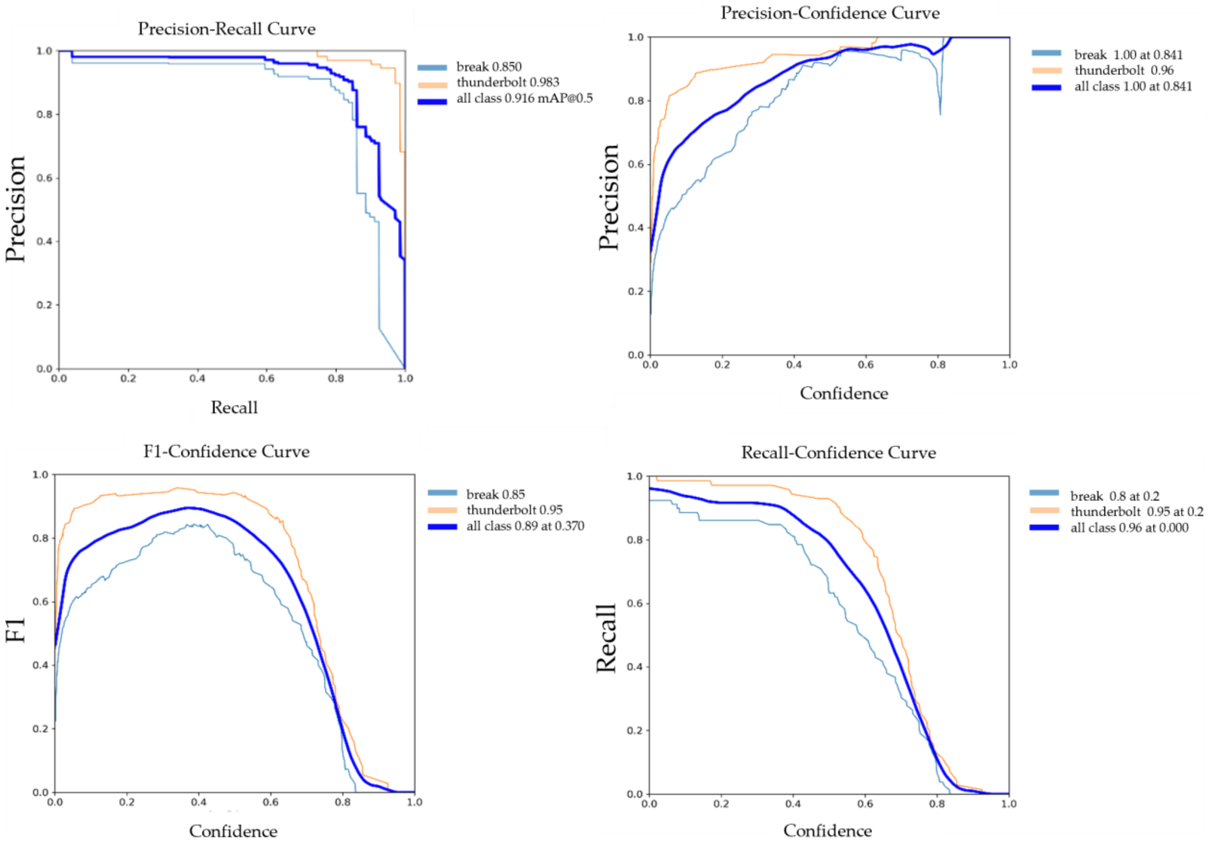


Figure 14. Visualization of the evaluation parameters for the cable damage dataset.

Based on ablation experimental results and comparative analysis of multiple datasets, the SL-YOLOv8 model shows significant performance improvement in object detection tasks. On the Break-ID-1632 dataset, the mAP@0.5 of SL-YOLOv8 reaches 94.1%, which is 4.2 percentage points higher than that of the baseline YOLOv8 model ($p < 0.01$, paired t-test). The contribution of each improved module is statistically significant ($p < 0.05$, analysis of variance multiple comparisons). Especially in precision and recall, SL-YOLOv8 reaches 84.6% and 94.6%, respectively, which is significantly better than other comparison models (the effect size index of the t-test > 0.8). In the generalization evaluation across datasets, SL-YOLOv8 also performs well on the cable damage dataset, confirming the robustness of the model improvement. It is worth noting that despite the limited sample size ($n < 1000$), the 95% confidence intervals [92.3%, 95.9%] estimated by the bootstrap method indicate the high reliability of the results.

To evaluate the stability and variability of the model performance, we conduct a series of Monte Carlo experiments on the SL-YOLOv8 model, randomly varying the training and testing datasets. On the Break-ID-1632 dataset, after 10 independent experiments, the average mAP@0.5 of SL-YOLOv8 reaches 93.8% ± 0.4% (mean ± standard deviation). This result is not only significantly higher than the baseline YOLOv8 model (89.9% ± 0.6%, $p < 0.001$, paired t-test) but also has a small standard deviation, indicating good stability of the model performance.

In each category, SL-YOLOv8 also performs well. For example, for the "damage" class, the average precision is 94.6% ± 0.5%, which is significantly better than the other comparison models ($p < 0.01$, analysis of variance analysis). It is worth noting that the model also maintains stable high performance in difficult categories such as "break1" to "break7", with an average accuracy of more than 92% and a standard deviation of less than 0.7%. These results show that SL-YOLOv8 not only significantly improves the average performance but also has strong stability and generalization ability under different data partitions. A small standard deviation means that the model can provide consistent and reliable detection results in practice. Future research can further expand the experimental scale and explore the performance of the model on more complex scenes and larger datasets.

4. Conclusions

In this paper, by simultaneously incorporating the LSK and SENet_v2 modules into the YOLOv8 model, the overall performance of the SL-YOLOv8 is significantly enhanced. The LSK module improves the ability to capture multi-scale features of the model, while the SENet_v2 module effectively integrates semantic and spatial information and optimizes the feature fusion strategy. Particularly in the identification of faults in transmission lines, the application of the LSK module allows the model to capture small and subtly defined targets more effectively, such as faults in transmission lines. The SENet_v2 module further enhances the sensitivity for fault features of the model. This combination not only improves the accuracy of fault detection but also significantly reduces false alarms, especially in complex situations and varying lighting conditions. Furthermore, the SL-YOLOv8 demonstrates enhanced real-time performance and scalability when handling large-scale transmission line images, providing more reliable technical support for the routine inspection and maintenance of power systems. Experiments conducted on the Break-ID-1632 and cable damage datasets demonstrate that the SL-YOLOv8 outperforms basic YOLOv8 and other typical algorithms in terms of accuracy, robustness, and versatility, effectively coping with data variations and noise. The SL-YOLOv8 extends the accuracy, speed, and model size. This enhancement validates the effectiveness of the synergistic action of the two modules in enhancing the model adaptability and performance, offering a new direction for further optimizing object detection algorithms.

Author Contributions: Conceptualization, X.Z.; methodology, X.Z.; software, X.Z.; validation, J.Z.; formal analysis, X.Z.; investigation, X.Z.; resources, X.J.; data curation, X.Z.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z. and X.J.; visualization, J.Z.; supervision, X.J.; project administration, X.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Our studies not involving humans or animals.

Informed Consent Statement: Our studies not involving humans.

Data Availability Statement: The data used in this paper can be obtained through the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Swain, A.; Abdellatif, E.; Mousa, A.; Pong, P.W.T. Sensor Technologies for Transmission and Distribution Systems: A Review of the Latest Developments. *Energies* **2022**, *15*, 7339. [CrossRef]
2. Li, X.; Li, Z.; Wang, H.; Li, W. Unmanned Aerial Vehicle for Transmission Line Inspection: Status, Standardization, and Perspectives. *Front. Energy Res.* **2021**, *9*, 713634. [CrossRef]
3. Toussaint, K.; Pouliot, N.; Montambault, S. Transmission Line Maintenance Robots Capable of Crossing Obstacles: State-of-the-art Review and Challenges Ahead. *J. Field Robot.* **2009**, *26*, 477–499. [CrossRef]
4. Wu, G.; Xiao, H.; Xiao, X.; Huang, Z.; Li, Y. Transmission Line Inspection Robot and Deicing Robot: Key Technologies, Prototypes and Applications. In Proceedings of the 2010 1st International Conference on Applied Robotics for the Power Industry, Montreal, QC, Canada, 5–7 October 2010; IEEE: Montreal, QC, Canada, 2010; pp. 1–6.
5. Rodriguez-Conde, I.; Campos, C.; Fdez-Riverola, F. Optimized Convolutional Neural Network Architectures for Efficient On-Device Vision-Based Object Detection. *Neural Comput. Applic.* **2022**, *34*, 10469–10501. [CrossRef]
6. Chen, G.; Wang, S.; Liu, Y.; Li, J.; Qin, G.; Wang, J.; Sun, A.; Yuan, L. Research on Transmission Line Hardware Identification Based on Improved YOLOv5 and DeblurGANv2. *IEEE Access* **2023**, *11*, 133351–133362. [CrossRef]
7. Wu, X.; Sahoo, D.; Hoi, S.C.H. Recent Advances in Deep Learning for Object Detection. *Neurocomputing* **2020**, *396*, 39–64. [CrossRef]
8. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
9. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; IEEE: Santiago, Chile, 2015; pp. 1440–1448.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
11. Zheng, R.; Zhu, L.; Hu, T.; Li, J. Detection of Fault Insulator of Power Transmission Line Based on Region-CNN. In Proceedings of the 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Zhanjiang, China, 16–18 October 2020; IEEE: Zhanjiang, China, 2020; pp. 73–76.
12. Li, X.; Shang, M.; Qin, H. Fast Accurate Fish Detection and Recognition of Underwater Images with Fast R-CNN. In Proceedings of the OCEANS 2015-MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; IEEE: Washington, DC, USA, 2015; pp. 1–5.
13. Xu, X.; Zhao, M.; Shi, P.; Ren, R.; He, X.; Wei, X.; Yang, H. Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN. *Sensors* **2022**, *22*, 1215. [CrossRef]
14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9905, pp. 21–37, ISBN 978-3-319-46447-3.
15. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
16. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
17. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
18. Ultralytics YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 20 August 2023).
19. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
20. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
21. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.

22. Maduako, I.; Igwe, C.F.; Abah, J.E.; Onwuasaanya, O.E.; Chukwu, G.A.; Ezeji, F.; Okeke, F.I. Deep Learning for Component Fault Detection in Electricity Transmission Lines. *J. Big Data* **2022**, *9*, 81. [CrossRef]
23. Wang, B.; Ding, H.; Teng, F.; Wang, Z.; Liu, H. Damage Object Detection of Steel Wire Rope-Core Conveyor Belts Based on the Improved YOLOv5. *Int. J. Image Grap.* **2024**, 2550057. [CrossRef]
24. Sheng, L.; Fang, S.; Yang, L.; Xiaokun, Y.; Dandan, L.; Longfei, L.; Yan, J. A Method and Implementation of Transmission Line's Key Components and Defects Identification Based on YOLO. In Proceedings of the 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 23–25 June 2022; IEEE: Chongqing, China, 2022; pp. 144–148.
25. Ultralytics YOLOV8. Available online: https://github.com/ultralytics/ultralytics (accessed on 12 October 2023).
26. Han, B.; Lu, Z.; Dong, L.; Zhang, J. Lightweight Non-Destructive Detection of Diseased Apples Based on Structural Re-Parameterization Technique. *Appl. Sci.* **2024**, *14*, 1907. [CrossRef]
27. Wu, Y.; Han, Q.; Jin, Q.; Li, J.; Zhang, Y. LCA-YOLOv8-Seg: An Improved Lightweight YOLOv8-Seg for Real-Time Pixel-Level Crack Detection of Dams and Bridges. *Appl. Sci.* **2023**, *13*, 10583. [CrossRef]
28. Nie, H.; Pang, H.; Ma, M.; Zheng, R. A Lightweight Remote Sensing Small Target Image Detection Algorithm Based on Improved YOLOv8. *Sensors* **2024**, *24*, 2952. [CrossRef]
29. Zhang, J.; Yang, W.; Lu, Z.; Chen, D. HR-YOLOv8: A Crop Growth Status Object Detection Method Based on YOLOv8. *Electronics* **2024**, *13*, 1620. [CrossRef]
30. Narayanan, M. SENetV2: Aggregated Dense Layer for Channelwise and Global Representations 2023. *arXiv* **2023**, arXiv:2311.10807.
31. Li, Y.; Hou, Q.; Zheng, Z.; Cheng, M.-M.; Yang, J.; Li, X. Large Selective Kernel Network for Remote Sensing Object Detection. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; IEEE: Paris, France, 2023; pp. 16748–16759.
32. Salazar, A.; Vergara, L.; Vidal, E. A Proxy Learning Curve for the Bayes Classifier. *Pattern Recognit.* **2023**, *136*, 109240. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.