

Article

Power Transmission Lines Foreign Object Intrusion Detection Method for Drone Aerial Images Based on Improved YOLOv8 Network

Hongbin Sun ^{1,*}, Qiuchen Shen ¹, Hongchang Ke ^{2,*} , Zhenyu Duan ¹ and Xi Tang ¹

¹ School of Electrical and Information Engineering, Changchun Institute of Technology, Changchun 130012, China; 20231041209@stu.ccit.edu.cn (Q.S.); 20231041212@stu.ccit.edu.cn (Z.D.); 20231041213@stu.ccit.edu.cn (X.T.)

² School of Computer Technology and Engineering, Changchun Institute of Technology, Changchun 130012, China

* Correspondence: dq_shb@ccit.edu.cn (H.S.); tj_khc@ccit.edu.cn (H.K.); Tel.: +86-138-4310-7325 (H.S.); +86-135-0088-2674 (H.K.)

Abstract: With the continuous growth of electricity demand, the safety and stability of transmission lines have become increasingly important. To ensure the reliability of power supply, it is essential to promptly detect and address foreign object intrusions on transmission lines, such as tree branches, kites, and balloons. Addressing the issues where foreign objects can cause power outages and severe safety accidents, as well as the inefficiency, time consumption, and labor-intensiveness of traditional manual inspection methods, especially in large-scale power transmission lines, we propose an enhanced YOLOv8-based model for detecting foreign objects. This model incorporates the Swin Transformer, AFPN (Asymptotic Feature Pyramid Network), and a novel loss function, Focal SIOU, to improve both the accuracy and real-time detection of hazards. The integration of the Swin Transformer into the YOLOv8 backbone network significantly improves feature extraction capabilities. The AFPN enhances the multi-scale feature fusion process, effectively integrating information from different levels and improving detection accuracy, especially for small and occluded objects. The introduction of the Focal SIOU loss function optimizes the model's training process, enhancing its ability to handle hard-to-classify samples and uncertain predictions. This method achieves efficient automatic detection of foreign objects by comprehensively utilizing multi-level feature information and optimized label matching strategies. The dataset used in this study consists of images of foreign objects on power transmission lines provided by a power supply company in Jilin, China. These images were captured by drones, offering a comprehensive view of the transmission lines and enabling the collection of detailed data on various foreign objects. Experimental results show that the improved YOLOv8 network has high accuracy and recall rates in detecting foreign objects such as balloons, kites, and bird nests, while also possessing good real-time processing capabilities.

Keywords: transmission lines; foreign object intrusion; YOLOv8; Swin transformer; AFPN; loss function



Citation: Sun, H.; Shen, Q.; Ke, H.; Duan, Z.; Tang, X. Power Transmission Lines Foreign Object Intrusion Detection Method for Drone Aerial Images Based on Improved YOLOv8 Network. *Drones* **2024**, *8*, 346. <https://doi.org/10.3390/drones8080346>

Academic Editors: Jeng-Shyang Pan, Junzo Watada, Vaclav Snasel and Pei Hu

Received: 21 June 2024

Revised: 20 July 2024

Accepted: 23 July 2024

Published: 25 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Intrusion of foreign objects into power transmission lines can not only trigger serious grid incidents such as short circuits and power outages, affecting power supply in large areas, but also potentially cause secondary disasters like fires, threatening public safety and the national economy's lifeline [1]. To ensure the stability and safety of the power system, there is an urgent need for efficient detection technologies, including drone-based inspections, to conduct real-time monitoring and rapid response for power transmission lines, preventing potential hazards [2], enhancing the grid's intelligence and reliability [3], and providing a solid guarantee for the continuous development of modern society. Images of foreign objects from different angles are shown in Figure 1.



Figure 1. Samples of foreign objects from different perspectives.

Traditional machine learning algorithms such as support vector machines (SVM), decision trees, and random forests have limited performance in object detection tasks. These algorithms rely on manually designed features, which struggle to effectively capture and process complex patterns and nonlinear relationships in images, resulting in insufficient detection accuracy and generalization capability. This performance inadequacy is particularly evident when dealing with diverse and high-dimensional data [4]. In contrast, deep learning algorithms such as convolutional neural networks (CNN) and recurrent neural networks (RNN) exhibit outstanding performance in object detection [5]. CNNs, by automatically learning multi-layer features of images, can efficiently handle large-scale and complex image data, significantly improving the accuracy and robustness of object detection [6]. Furthermore, deep learning methods such as YOLO and Faster R-CNN have demonstrated exceptional real-time performance and detection accuracy in practical applications, becoming mainstream choices in the field of object detection. Consequently, due to their superior performance and broad adaptability, deep learning algorithms have gradually replaced traditional machine learning algorithms, becoming the common technology used in object detection tasks [7]. Shi et al. [8] discussed the application of Faster R-CNN in real-time object detection, addressing the issue of efficiently detecting small objects in real-time. Wang et al. [9] discussed the application of YOLOv1 and Faster R-CNN in small object detection, addressing the application of deep learning in small object detection in remote sensing images. Min et al. [10] introduced a small object detection method based on YOLO, tackling the challenges of using YOLO for small object detection in remote sensing images. Liu et al. [11] discussed various detection algorithms, including Faster R-CNN, providing a review and comparison of different deep learning methods for general object detection, especially focusing on multiple algorithms including Faster R-CNN. Zhang et al. [12] solved the issue of insufficient performance in small object detection by introducing a multi-receptive field structure. Cheng et al. [13] conducted a review and benchmark testing on large-scale small object detection, addressing the evaluation and

improvement of small object detection algorithms, particularly the application of Cascade R-CNN. However, Faster R-CNN has issues such as insufficient real-time performance, poor detection of small objects, high computational resource consumption, limited multi-scale and deformation handling capabilities, long training time, and poor handling of occluded and overlapping objects.

Among the many deep learning methods, the YOLO (You Only Look Once) series of algorithms stand out in the field of object detection due to their efficient real-time performance and excellent detection accuracy. YOLO algorithms transform the object detection problem into a single regression problem, simultaneously predicting the category and location of objects in a single forward pass, greatly reducing computation time [14]. Compared to traditional methods and other deep learning algorithms, YOLO significantly improves detection speed while maintaining high accuracy, making it particularly prominent in practical applications [15]. Since its introduction, the YOLO series has continuously iterated from YOLOv1 to the latest YOLOv8, with each generation optimizing network structures and algorithm details, consistently enhancing small object detection effectiveness and adapting to complex scenarios [16].

Shao et al. [17] proposed an improved YOLOv8-based foreign object detection method for power transmission lines, improving the accuracy and real-time performance of foreign object detection, though performance may decline under extreme weather conditions (e.g., strong winds, heavy rain), and there are high requirements for dataset diversity and generalization capability. Tang et al. [18] proposed a YOLO-based method for detecting external damages to power transmission lines, aiming to enhance the safety and reliability of the power grid by preventing external damages, but false positives or missed detections may occur in complex backgrounds (e.g., trees, buildings), demanding high model accuracy and robustness.

Shi et al. [19] studied a privacy protection method using drones and YOLO for power line detection, improving the privacy and security of transmission tower and defect detection, though privacy protection measures may increase computational complexity, affecting real-time detection performance. Li et al. [20] proposed the DF-YOLO (Deformable YOLO) algorithm for foreign object detection on power transmission lines, significantly improving detection accuracy and robustness, but the high algorithm complexity may require more computational resources, posing challenges for real-time detection. Liu et al. [21] combined the lightweight convolutional neural network MobileNet and clustering algorithms to design the MYOLO-lite object detection algorithm, improving the speed and accuracy of forward railway track object detection through improved non-maximum suppression and loss functions, achieving an average precision of 95.74%, but it may have some issues with missed or false detections when dealing with high-speed moving objects.

Nayak et al. [22] proposed an intrusion detection system (IDS) based on deep learning, using the YOLO algorithm for object detection and achieving real-time intrusion tracking through the center of mass shift algorithm and SORT algorithm, but the complex urban environment may lead to a high false alarm rate due to background interference. Chen et al. [23] used MASK R-CNN as the base network to detect foreign objects in power networks, achieving good results in terms of speed, efficiency, and recognition accuracy compared to traditional detection methods, but the computational complexity is high, resulting in insufficient real-time performance, especially when processing large-scale data. Sevi et al. [24] proposed a deep learning method based on YOLOv8 for detecting foreign objects around railways, achieving an accuracy of 88.8% on mAP@0.5 using the RailSem19 dataset and image augmentation techniques, effectively improving railway transportation safety, but it may experience false positives or missed detections in complex backgrounds such as trees and buildings. Shan et al. [25] introduced cloud-edge technology and proposed an improved YOLOv4 network-based monitoring model for foreign object detection on transmission lines. The improved YOLOv4 network significantly enhanced feature extraction and detection accuracy, achieving a foreign object detection accuracy of 99.2% and a detection speed of 218 milliseconds, meeting real-time detection needs,

but performance may decline under extreme weather conditions such as strong winds or heavy rain.

Li et al. [26] proposed a foreign object intrusion detection method based on WiFi technology, quickly detecting and locating foreign objects in non-illuminated tunnel environments by detecting abnormal phase shifts in WiFi network radio frequency signals. Experimental results show that this method performs well in terms of detection probability and location accuracy, but the detection range and accuracy may be affected by WiFi signal strength and environmental interference. Wang et al. [27] proposed an improved YOLOv8m model, combining the Global Attention Module (GAM) and Focal-EIoU loss function to detect foreign objects on transmission lines. Experimental results show that this model improved mAP@0.5 by 2.7%, mAP0.5:0.95 by 4%, and recall rate by 6%, effectively solving detection problems in complex backgrounds of transmission lines, but the high algorithm complexity may require more computational resources. Zhao et al. [28] proposed an optimization method based on Context Information Enhancement (CIE) and Joint Heterogeneous Representation (JHR), significantly improving the detection accuracy of small and occluded objects on transmission lines, with a total mAP increase of 5.8%, but in practical applications, the complexity of the algorithm may result in high consumption of computational resources.

Based on the above proposed methods, there may be deficiencies in detection efficiency and high algorithm complexity. In some methods, detection accuracy in complex scenes may still be insufficient. To address these issues, this paper proposes an improved YOLOv8 algorithm by introducing the Swin Transformer and Adaptive Feature Pyramid Network (AFPN) and replacing the loss function with Focal-SIoU to improve the accuracy and speed of foreign object detection on transmission lines. The contributions of this paper are as follows:

- (1) Swin Transformer is adopted as the high-resolution feature extraction layer to enhance the feature extraction capability of small objects.
- (2) An Adaptive Feature Pyramid Network (AFPN) is designed, combining multi-scale features to improve the model's detection capability at different scales.
- (3) The Focal-SIoU loss function is used to balance the distribution of positive and negative samples, thereby accelerating algorithm convergence and improving detection accuracy.

Experimental results show that the improved YOLOv8 model improved mAP@0.5 by 1.5%, accuracy by 3.6%, and recall rate by 5.2%, effectively solving detection problems in complex backgrounds of transmission lines.

2. YOLOv8 Network

2.1. Principle of YOLOv8 Algorithm

According to the provided YOLOv8 structure diagram, the architecture of YOLOv8 consists of three parts: the backbone network (Backbone), the feature fusion network (Neck), and the detection head (Head). The input image first passes through a series of convolutional layers and pooling layers to extract basic features. After each convolutional layer (C3F, C2F), there may be a pooling layer to reduce the resolution of the feature map and improve computational efficiency. Finally, multi-scale features are fused through the SPPCSPC layer. The features extracted by the Backbone are fused through a feature pyramid (FPN, PAN) for multi-scale integration. Upsampling and downsampling (CSPC, C2F) operations facilitate information transfer between different scale features. The feature fusion part concatenates features of different scales to form richer feature representations [29]. The detection head performs object detection based on the fused features, including classification and bounding box regression. The feature maps at each scale are processed through convolutional layers to generate the final detection results. YOLOv8 introduces spatial depth convolution (SPD-Conv) and large selective kernel (LSK) attention mechanisms to enhance feature extraction capabilities. The smooth intersection over union (SIoU) loss function improves detection accuracy and accelerates model convergence. These improve-

ments allow YOLOv8 to significantly enhance detection performance while maintaining real-time capabilities. The YOLOv8 network architecture is shown in Figure 2.

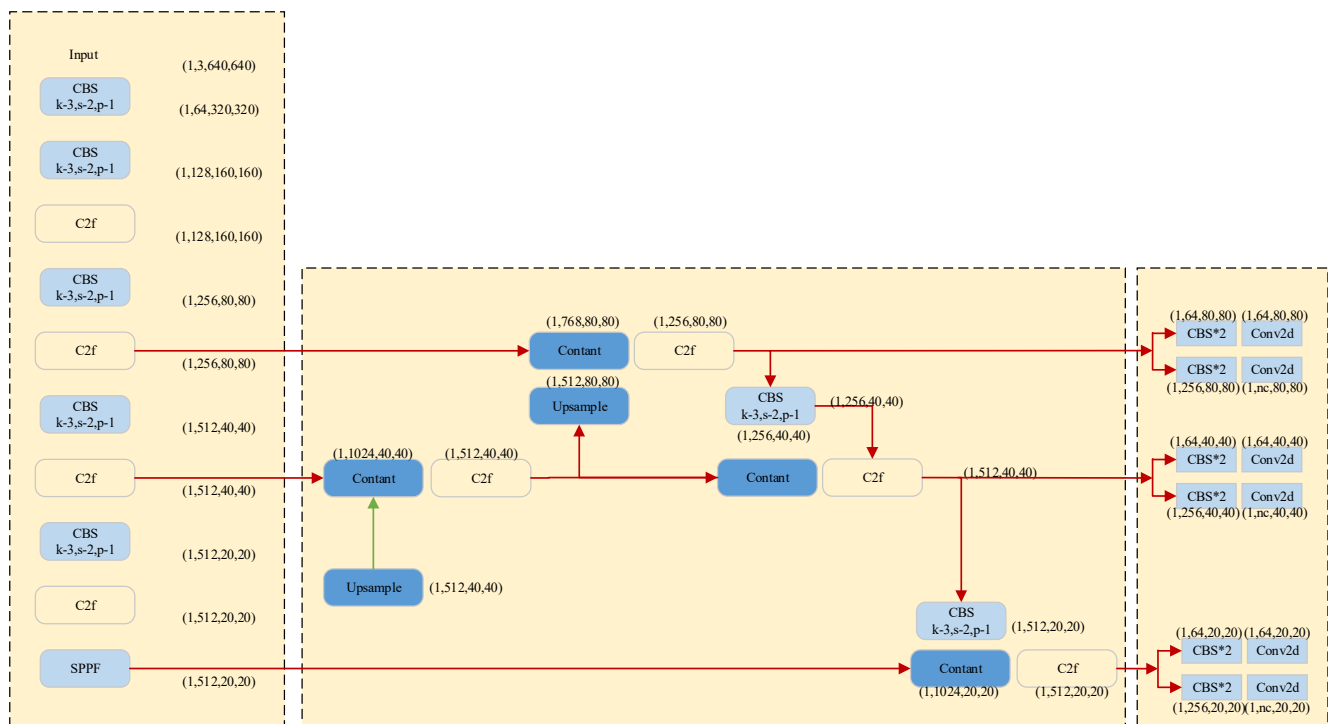


Figure 2. The YOLOv8 network architecture.

2.2. YOLOv8 Backbone Network

The backbone network of YOLOv8 is one of its core components, responsible for extracting multi-level features from the input image. The input image, sized $640 \times 640 \times 3$, first passes through a convolutional layer and a pooling layer, which initially extracts the low-level features of the image, outputting a feature map of size $320 \times 320 \times 32$ [30]. Next, it enters multiple C2F modules. These modules consist of multiple convolutional layers and feature fusion layers, gradually reducing the resolution of the feature map and increasing the number of channels to extract higher-level features [31]. For example, the first C2F module outputs a size of $160 \times 160 \times 64$, the second C2F module outputs $80 \times 80 \times 128$, the third C2F module outputs $40 \times 40 \times 256$, and the fourth C2F module outputs $20 \times 20 \times 512$. After the last C2F module, an SPP layer is used for multi-scale feature fusion, capturing contextual information at different scales, outputting a size of $20 \times 20 \times 1024$. The backbone network of YOLOv8 constructs an efficient and accurate feature extraction module through multi-level convolution and feature fusion, employing techniques such as deep convolution and large kernel convolution to enhance feature extraction efficiency and representation capability. This ensures that the model maintains high performance while having better real-time performance. Through these designs, the backbone network of YOLOv8 enhances the detection capability of small and large targets, improving detection accuracy, and speed [32]. The training flowchart of the YOLOv8 network is shown in Figure 3.

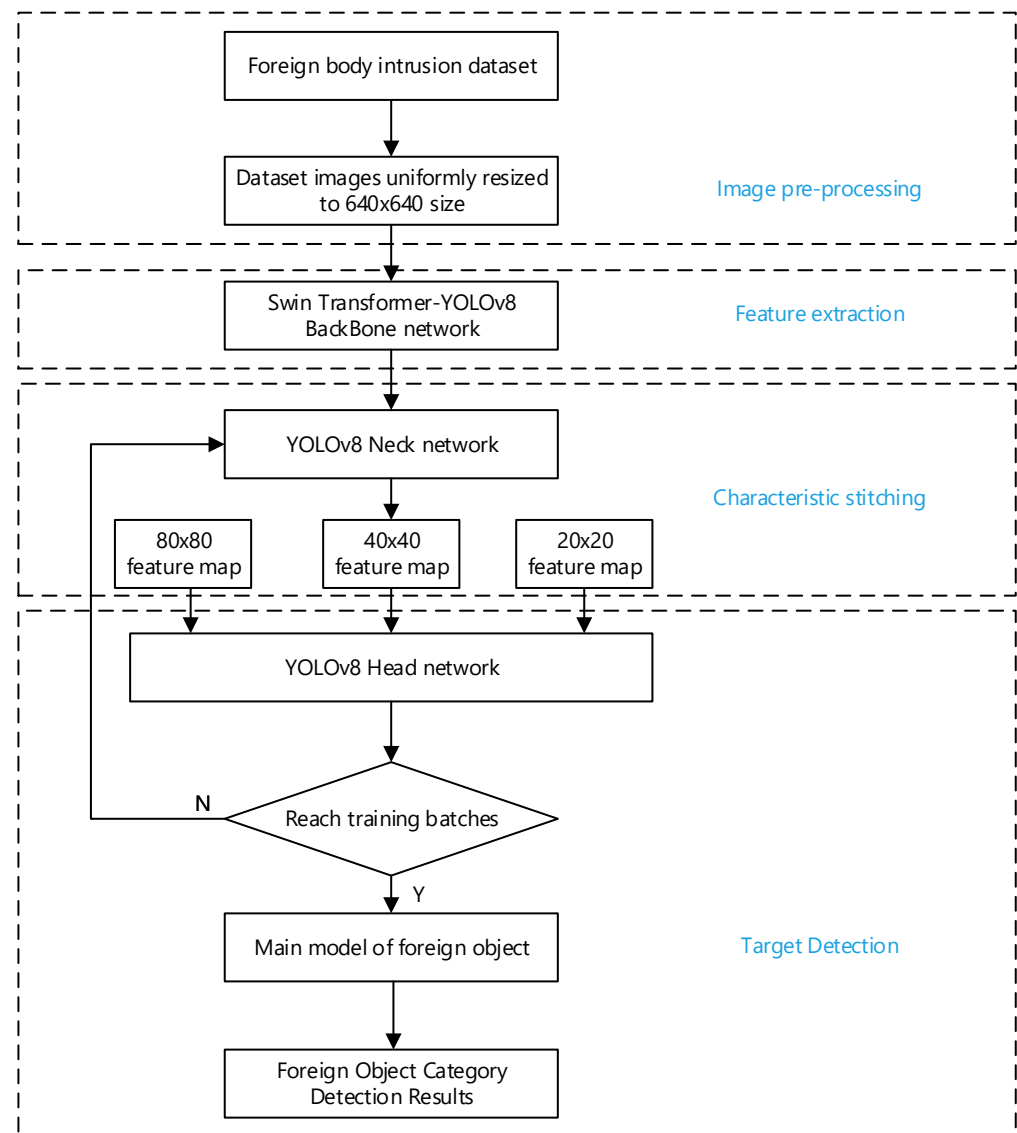


Figure 3. The training flowchart of the YOLOv8 network is shown in.

3. Improvements

3.1. Swin Transformer Overall Framework

To enhance the performance of YOLOv8 in detecting foreign objects on transmission lines, we opted to replace the backbone network of YOLOv8 with Swin Transformer. This choice is based on its remarkable performance advantages in vision tasks [33]. Swin Transformer introduces hierarchical feature representation and a sliding window mechanism, which effectively capture both local and global features in images, thereby improving detection accuracy. Compared to traditional convolutional neural networks, Swin Transformer exhibits stronger feature extraction capabilities, particularly excelling in handling complex backgrounds and small object detection [34]. Moreover, the multi-scale feature extraction mechanism of Swin Transformer allows it to be more flexible and efficient in dealing with objects of different scales. By integrating Swin Transformer with YOLOv8's single-stage detection architecture, the model's detection performance and generalization ability can be significantly enhanced. The structure of Swin Transformer is shown in Figure 4a.

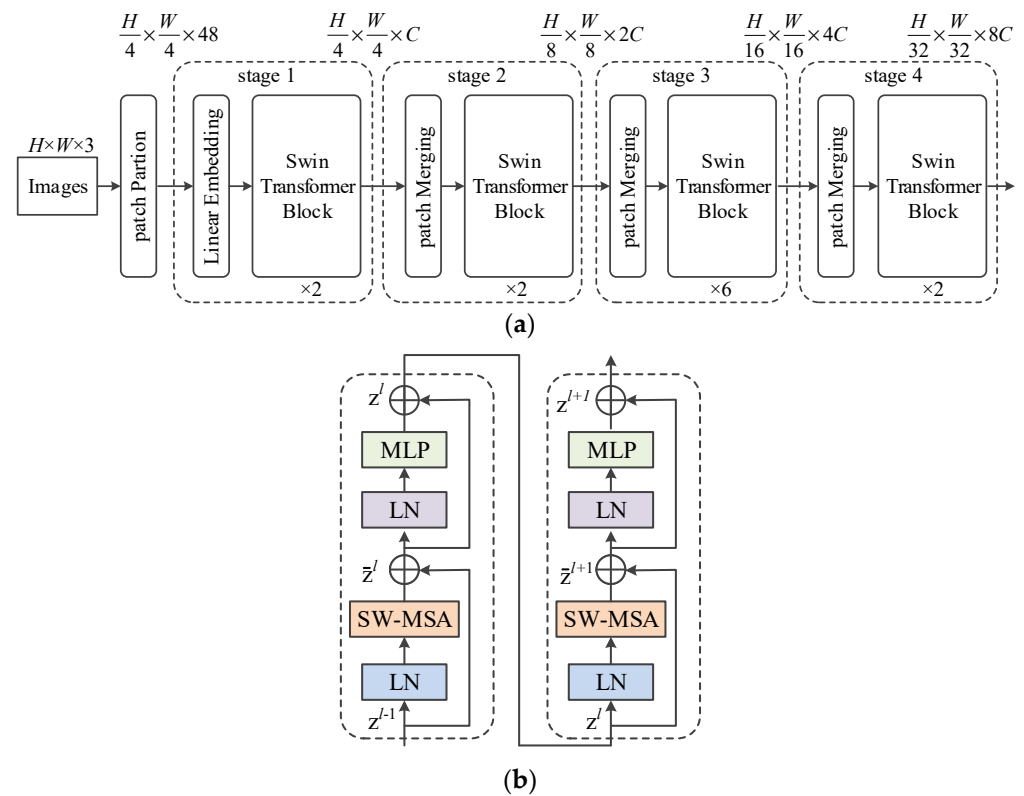


Figure 4. (a) The structure of Swin Transformer (Swin-T). (b) Two successive Swin Transformer blocks.

The architecture of Swin Transformer is outlined in Figure 4a, showcasing the small version (Swin-T). Initially, it partitions the input RGB image into non-overlapping patches via a slicing module, treating each patch as a “token” with features set to the concatenated original pixel RGB values. In our implementation, we use a patch size of 4×4 , resulting in each patch having a feature dimension of $4 \times 4 \times 3 = 48$. A linear embedding layer is then applied to these raw value features, projecting them into an arbitrary dimension (denoted as C). Several Transformer blocks (Swin Transformer blocks) with modified self-attention computation are applied to these patch tokens. These blocks maintain the number of tokens ($H/4 \times W/4$) and, together with the linear embedding layer, constitute “Stage 1”. As the network deepens, the number of tokens is reduced through patch merging layers to generate hierarchical representations. The first patch merging layer concatenates the features of 2×2 neighboring patches and applies a linear layer on the concatenated features in $4C$ dimensions. This reduces the number of tokens by a factor of $2 \times 2 = 4$ ($2 \times$ downsampling resolution), with the output dimension set to $2C$. Swin Transformer blocks are then applied to transform these features, reducing the resolution to $H/8 \times W/8$. This feature transformation after the first patch merge is termed “Stage 2”. The process repeats, sequentially reducing the resolution to $H/16 \times W/16$ and $H/32 \times W/32$, referred to as “Stage 3” and “Stage 4”, respectively. These stages collectively produce a hierarchical representation [35].

Swin Transformer constructs the Swin Transformer blocks by replacing the standard multi-head self-attention (MSA) module in the Transformer blocks with a shifted window-based module, keeping the other layers unchanged [36]. As shown in Figure 4b, each Swin Transformer block consists of a shifted window-based MSA module followed by a two-layer MLP with GELU non-linear activation. A LayerNorm (LN) layer is applied before each MSA module and each MLP, and a residual connection is applied after each module.

3.2. Improved Progressive Feature Pyramid Network (AFPN)

3.2.1. Traditional Feature Fusion

Traditional feature fusion methods in convolutional neural networks (CNNs) are typically implemented through Feature Pyramid Networks (FPN) or skip connections [37]. The core idea of these methods is to combine feature maps from different levels to capture information at various scales, thereby enhancing detection accuracy and robustness. Common traditional feature fusion methods include Feature Pyramid Networks (FPN) and skip connections. FPN enhances higher-level feature maps via a top-down pathway while integrating lower-level feature maps through lateral connections. This approach combines low-resolution semantic features and high-resolution detailed features at each level [38]. Skip connections directly combine feature maps from earlier layers with those from later layers, preserving more original information and reducing information loss [39]. The structure of the Feature Pyramid Network is illustrated in Figure 5.

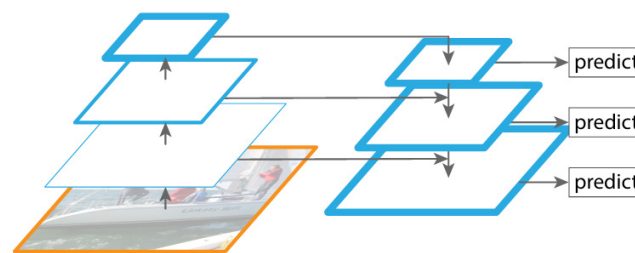


Figure 5. Feature Pyramid Network. In this figure, feature maps are indicated by blue outlines and thicker outlines denote semantically stronger features.

3.2.2. Asymptotic Feature Pyramid Network (AFPN)

Multi-scale features are crucial for encoding objects with varying scales in object detection tasks. A common strategy for multi-scale feature extraction is using classical top-down and bottom-up Feature Pyramid Networks (FPN). However, these methods suffer from feature information loss or degradation, affecting the fusion effectiveness between non-adjacent levels. This paper proposes a Asymptotic Feature Pyramid Network (AFPN) to support direct interaction between non-adjacent levels. AFPN progressively integrates high-level features into the fusion process by merging two adjacent low-level features. This approach avoids large semantic gaps between non-adjacent levels [40]. Considering the potential conflict of multi-object information during feature fusion at each spatial location, an adaptive spatial fusion operation is further adopted to mitigate these inconsistencies. We integrate the proposed AFPN into both two-stage and single-stage object detection frameworks and evaluate it on the MS-COCO 2017 validation and test sets.

Like many FPN-based object detection methods, features are extracted from different levels of the backbone before feature fusion. [41] Following the design of the Faster R-CNN framework, the last feature layer from each level of the backbone is extracted, resulting in a set of multi-scale features denoted as $\{C2, C3, C4, C5\}$. To perform feature fusion, low-level features $C2$ and $C3$ are first fed into the feature pyramid network, followed by the addition of $C4$, and finally, $C5$. After the feature fusion step, a set of multi-scale features $\{P2, P3, P4, P5\}$ is produced. For experiments conducted on the Faster R-CNN framework, a convolution with a stride of 2 is applied to $P5$, followed by a convolution with a stride of 1 to generate $P6$, ensuring uniform output. The final set of multi-scale features is $\{P2, P3, P4, P5, P6\}$, with corresponding feature strides of $\{4, 8, 16, 32, 64\}$ pixels. It should be noted that YOLO only feeds $\{C3, C4, C5\}$, excluding $C2$, into the feature pyramid network, which generates the outputs $\{P3, P4, P5\}$.

The architecture of the proposed AFPN is shown in Figure 6. During the bottom-up feature extraction process of the Backbone network, AFPN progressively integrates low-level, high-level, and top-level features [42]. Specifically, AFPN initially fuses low-level features, then integrates deeper features, and finally fuses the highest-level features, which

are the most abstract. The semantic gap between non-adjacent hierarchical features is larger than that between adjacent hierarchical features, especially between the bottom and top features. This directly leads to poorer fusion performance of non-adjacent hierarchical features. Therefore, directly using C2, C3, C4, and C5 for feature fusion is unreasonable. Due to the progressive nature of AFPN's architecture, the semantic information of different level features becomes closer during the progressive fusion process, thereby alleviating the aforementioned issue. For example, the feature fusion between C2 and C3 reduces their semantic gap. Since C3 and C4 are adjacent hierarchical features, the semantic gap between C2 and C4 is also reduced.

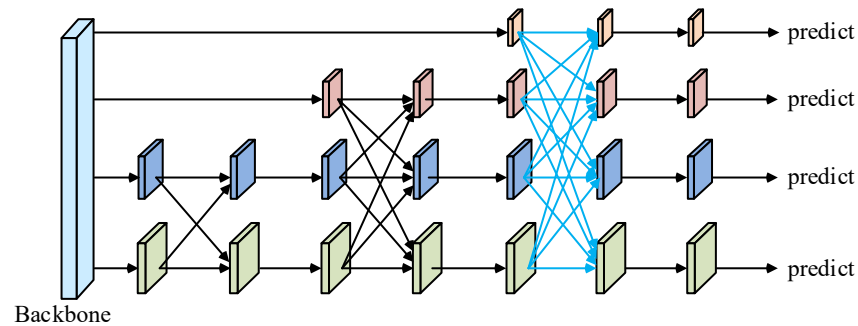


Figure 6. The architecture of the proposed Asymptotic Feature Pyramid Network (AFPN). In the initial stage, two low-level features are fused first, followed by the integration of high-level features, and finally, the top-level features are fused.

To align dimensions and prepare for feature fusion, the authors use 1×1 convolution and bilinear interpolation for upsampling. On the other hand, the authors perform downsampling using different convolution kernels and strides according to the required downsampling rate. For instance, they apply 2×2 convolution with a stride of 2 for $2 \times$ downsampling, 4×4 convolution with a stride of 4 for $4 \times$ downsampling, and 8×8 convolution with a stride of 8 for $8 \times$ downsampling. After feature fusion, the authors continue to learn features using 4 residual units, similar to ResNet. Each residual unit includes two 3×3 convolutions. Since YOLO uses only 3 levels of features, there is no $8 \times$ upsampling and $8 \times$ downsampling.

3.3. Improved Loss Function

In object detection tasks, the design of the loss function is crucial for model performance. YOLOv8 introduces an improved loss function that combines multiple strategies to enhance the model's robustness and accuracy in different scenarios. First, YOLOv8 adopts IoU (Intersection over Union) as the basis for the regression loss [43]. IoU measures the overlap between the predicted bounding box and the ground truth bounding box, defined as the area of their intersection divided by the area of their union, as shown in Equation (1):

$$IoU = \frac{B \cap B_{gt}}{B \cup B_{gt}} \quad (1)$$

However, IoU cannot reflect distance information when the bounding boxes do not overlap. Therefore, YOLOv8 introduces GIoU (Generalized IoU) and DIoU (Distance IoU) to better capture the overlap and distance relationships between target boxes. The formula for GIoU is shown in Equation (2):

$$GIoU = IoU - \frac{\left| \frac{C}{B \cup B_{gt}} \right|}{|C|} \quad (2)$$

where C is the smallest enclosing box that can contain both the predicted box and the ground truth box.

The formula for DIoU is shown in Equation (3):

$$DIoU = IoU - \frac{\rho^2(b, b_{gt})}{c^2} \quad (3)$$

where ρ is the Euclidean distance between the centers of the predicted box and the ground truth box, and c is the diagonal length of the smallest enclosing box. These improvements address the shortcomings of IoU to some extent.

Secondly, YOLOv8 employs an improved Focal Loss to address the issue of class imbalance. Focal Loss assigns greater weight to hard-to-classify samples, reducing the contribution of easily classified samples to the loss. The formula for Focal Loss is shown in Equation (4):

$$L_{FL} = -\alpha(1 - p_t)^\gamma \log(p_t) \quad (4)$$

where α is the balancing factor, γ is the modulation factor, and p_t represents the model's predicted probability for the positive class. When p_t is large, $(1 - p_t)^\gamma$ approaches zero, thereby reducing the weight of easily classified samples. By adjusting γ , the contribution ratio of hard and easy samples to the loss can be controlled.

Thirdly, confidence loss is used to measure whether the predicted box contains the target. YOLOv8 introduces Focal Loss in the confidence loss to reduce the impact of negative samples on the loss. The formula for this is shown in Equation (5):

$$L_{cof} = -\sum_{i=1}^N [a(1 - p_i)^\gamma \log(p_i) + (1 - \alpha)p_i^\gamma \log(1 - p_i)] \quad (5)$$

where p_i is the confidence of the i -th sample, and the confidence of positive and negative samples are represented by p_i and $(1 - p_i)$ respectively. By using Focal Loss to reduce the impact of negative samples, the accuracy of confidence prediction is improved.

The total loss is the weighted sum of the aforementioned components. The total loss function in YOLOv8 is given by Equation (6):

$$L = \lambda_{GIoU}L_{GIoU} + \lambda_{DIoU}L_{DIoU} + \lambda_{cls}L_{FL} + \lambda_{conf}L_{conf} \quad (6)$$

where λ_{GIoU} , λ_{DIoU} , λ_{cls} , and λ_{conf} are the weight hyperparameters for each loss component. Experimental results on standard datasets show that the improved loss function significantly enhances the object detection performance of YOLOv8, particularly in small object detection and handling complex backgrounds.

Therefore, in this paper, we adopt Focal SIoU (Smooth Intersection over Union with Focal Loss), a loss function that combines Smooth IoU and Focal Loss to enhance the performance of object detection models in handling class imbalance and bounding box regression. This loss function, through smooth IoU calculation and the weight adjustment of Focal Loss, better addresses hard and uncertain samples in object detection.

Focal SIoU combines Smooth IoU and Focal Loss by incorporating the concept of Focal Loss into the bounding box regression loss, thus optimizing the overlap of bounding boxes while paying more attention to difficult-to-locate targets and uncertain predictions. The calculation formula for Focal SIoU is given by Equation (7):

$$L_{FocalSIoU} = (1 - SIoU)^\gamma \cdot \log(SIoU) \quad (7)$$

where $SIoU$ is the result of the smooth IoU calculation, and γ is the modulation factor in Focal Loss, used to adjust the contribution of samples with different overlap degrees to the loss.

4. Experimental Analysis

4.1. Dataset

Foreign objects on power transmission lines include bird nests, kites, and balloons. The dataset used in this study is entirely sourced from a power supply bureau in Jilin Province,

China. A total of 1108 images were selected for training. The dataset was partitioned with a ratio of 9:1:1 to create the training set, validation set, and test set. To ensure a robust evaluation of the model's generalization ability, we randomly split the dataset into training, validation, and test sets in a ratio of 9:1:1. The random split ensures that the data distribution in each subset (training, validation, and test) is consistent with the overall dataset distribution, preventing performance instability due to data bias. By using random sampling, we ensured that the data in the training set, validation set, and test set are mutually exclusive, with no overlap, thus guaranteeing the fairness and independence of the model evaluation. Moreover, this strategy ensures that each subset represents the diversity and complexity of the entire dataset, making the model's performance on each subset more reliable.

We did not employ the K-fold cross-validation technique primarily due to computational resource limitations and practical application considerations. Although K-fold cross-validation can provide a more comprehensive model evaluation, its computational cost is high when dealing with large-scale image datasets. We chose a simpler and effective random split method to better simulate real-world application scenarios and more accurately evaluate the model's practical performance. The categories and quantities in the dataset are shown in Table 1.



Table 1. Categories and quantities in the dataset.

<i>Class</i>	<i>Number</i>
Nest	364
Kite	374
Balloon	370

4.2. Experimental Setup

During the experiments, a high demand was placed on the computer's memory and GPU performance due to extensive image processing requirements. The experiments were conducted on a platform with a 13th Gen Intel Core i5-13400 processor, NVIDIA GeForce RTX 4060 Ti (16 GB), and 16 GB RAM, running on the Windows 10 operating system. We used PyCharm as the deep learning software platform, with PyTorch as the deep learning framework, Anaconda3 as the integrated environment, running on Python 3.8, and CUDA 12.1 for acceleration. The experimental setup is detailed in Table 2.

Table 2. Experimental setup.

<i>Parameter</i>	<i>Setup</i>
PyTorch	2.3.0
Python	3.8
GPU	NVIDIA GeForce RTX 4060 Ti (16 GB)
CPU	13th Gen Intel Core i5-13400
RAM	16 GB
CUDA	12.1

Before conducting the experiments, multiple trials were performed on the original YOLOv8 model. Finally, the hyperparameters applied in all experiments were configured uniformly, as shown in Table 3. For example, the initial learning rate was empirically set to 0.01, with the number of iterations set to 200. To expedite the training process and accommodate hardware capabilities, the batch size was set to 32 images. All images were resized to 640×640 .

Table 3. Hyperparameter configurations.

<i>Parameter</i>	<i>Setup</i>
Initial Learning Rate	0.001
Final Learning Rate	0.001
Epoch	200
Momentum	0.937
Patience	100
IoU	0.5
Weight Decay	0.005
Image Size	640×640

The hyperparameter configurations for all training sessions are detailed in Table 3.

4.3. Evaluation Criteria

In this experiment, evaluating the performance of the YOLOv8 model requires considering various factors, including speed, accuracy, applicability, robustness, and cost. These factors are weighed differently depending on the usage scenarios. This study focuses on assessing the accuracy of the improved YOLOv8 model on the URPC series datasets. We measure the model's performance by calculating and comparing the Average Precision (AP) for each category and the mean Average Precision (mAP). Additionally, we analyze the impact of Floating Point Operations (FLOPs) and the number of parameters (Para) on the model's performance to verify the superiority of the improved YOLOv8 model.

The following formulas summarize the key metrics for evaluating the performance of object detection algorithms, including precision (P), recall (R), average precision (AP), mean average precision (mAP), and intersection over union (IoU). Precision and recall measure detection accuracy and coverage, respectively, while AP and mAP assess overall performance, and IoU evaluates the overlap between predicted and ground truth bounding boxes, as shown in Formulas (8)–(12).

$$P = \frac{TP}{TP + FP} \quad (8)$$

$$R = \frac{TP}{TP + FN} \quad (9)$$

$$AP = \int_0^1 P(R) dR \times 100\% (i = 0, 1, 2 \dots) \quad (10)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (11)$$

$$IoU = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \quad (12)$$

4.4. Comparative Experiments with Other Algorithms

4.4.1. Effect of Swin Transformer on YOLOv8

To validate the significant improvements of our enhancements compared to other algorithms, we used Faster R-CNN, SSD, YOLOv3, YOLOv5 series models, YOLOv8 series models, and our optimized model. The performance capabilities of each model parameter are shown in Table 4.

Table 4. Performance capabilities of various models.

Model	AP (%)			FPS (G)	Params (M)	mAP@0.5 (%)
	Kite	Balloon	Nest			
Faster R-CNN	73.2	85.4	77.5	38.7	129	78.7
SSD	80.3	78.2	83.6	46.1	34.3	80.6
YOLOv3	81.2	78.3	85.4	45.5	61.9	81.6
YOLOv5	81.7	84.3	85.2	109.8	24.69	83.7
YOLOv5m	94.3	86.2	79.8	89.9	21	86.7
YOLOv8	83.2	92.1	91.2	150.7	28.40	88.8
YOLOv8s	91.6	82.3	87.7	130.7	11.20	88.5
Ours Model	90.3	92.7	95.6	181.4	23.21	92.8

Based on the data in Table 4, the following conclusions can be drawn: When comparing the performance of multiple object detection models, it is evident that the YOLOv8 model with the Swin Transformer as the backbone network performs exceptionally well. In terms of AP (Average Precision), the Swin Transformer + YOLOv8 model achieved 90.3%, 92.7%, and 95.6% for kite, balloon, and bird nest detection, respectively, significantly surpassing other models. Additionally, its mAP@0.5 reached 92.8%, indicating higher average detection precision across multiple categories.

Furthermore, although the Swin Transformer + YOLOv8 has a larger number of parameters (51.3M) and FLOPs (29.75G), its outstanding detection performance demonstrates that introducing a more powerful backbone network can significantly enhance object detection accuracy and robustness. Among all the compared models, YOLOv5m and YOLOv8s also performed well, but still slightly lagged behind the improved YOLOv8 with Swin Transformer in overall performance.

Thus, it can be concluded that by integrating advanced backbone networks like Swin Transformer, YOLOv8 showcases higher accuracy and comprehensive performance in object detection tasks, despite the increased computational resource requirements.

Figure 7 shows the performance of different object detection models on the Foreign Object Intrusion Dataset, measured by Average Precision (AP %). The figure compares the detection effectiveness of various models on three types of objects: kites, balloons, and nests. It is evident that the Swin Transformer + YOLOv8 model achieves nearly or exactly 100% average precision across all object categories, demonstrating the best performance. In contrast, the average precision of Faster R-CNN, SSD, and the YOLO series models is slightly lower but still maintains a high level, approximately between 80% and 90%.

This indicates that the Swin Transformer + YOLOv8 model not only maintains high precision but also more effectively detects multiple types of objects, showcasing its advantages in complex object detection tasks.

Figure 8 compares the computational complexity (measured in FPS) of different object detection models. The models include Faster R-CNN, SSD, YOLOv3, YOLOv5, YOLOv5m, YOLOv8, YOLOv8s, and “Ours Model”. From the chart, we can see that “Ours Model” has the highest computational complexity at 181.4 FPS, while Faster R-CNN has the lowest at 38.7 FPS. The other models have the following computational complexities: SSD

46.1 FPS, YOLOv3 45.5 FPS, YOLOv5 109.8 FPS, YOLOv5m 89.9 FPS, YOLOv8 150.7 FPS, and YOLOv8s 130.7 FPS. This chart provides a clear visual comparison of the computational complexity of each model, aiding in the selection of the most suitable model for specific application scenarios.

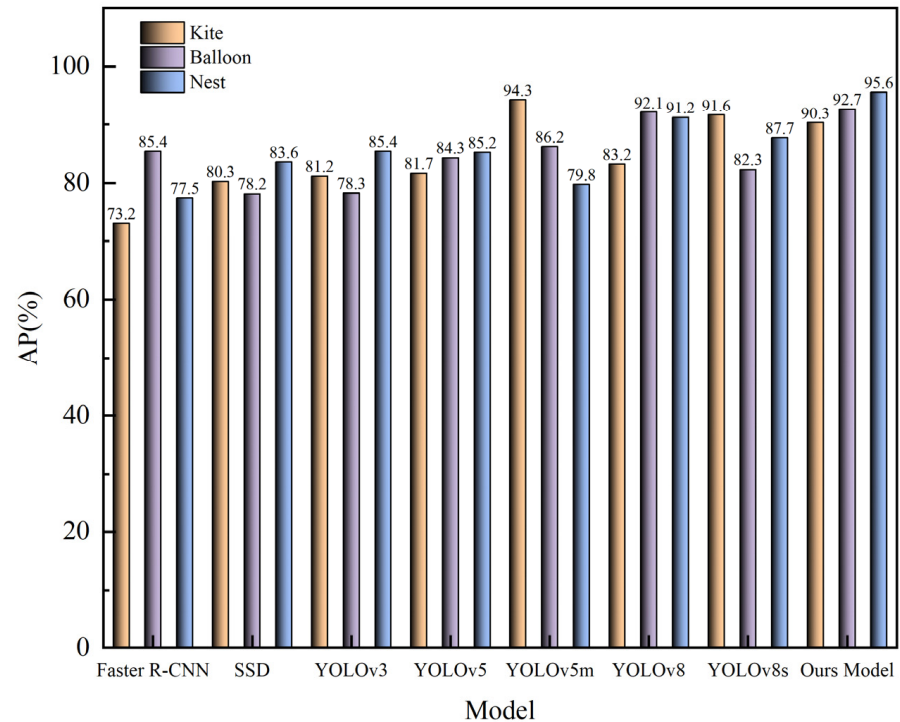


Figure 7. Performance of different models under foreign object intrusion.

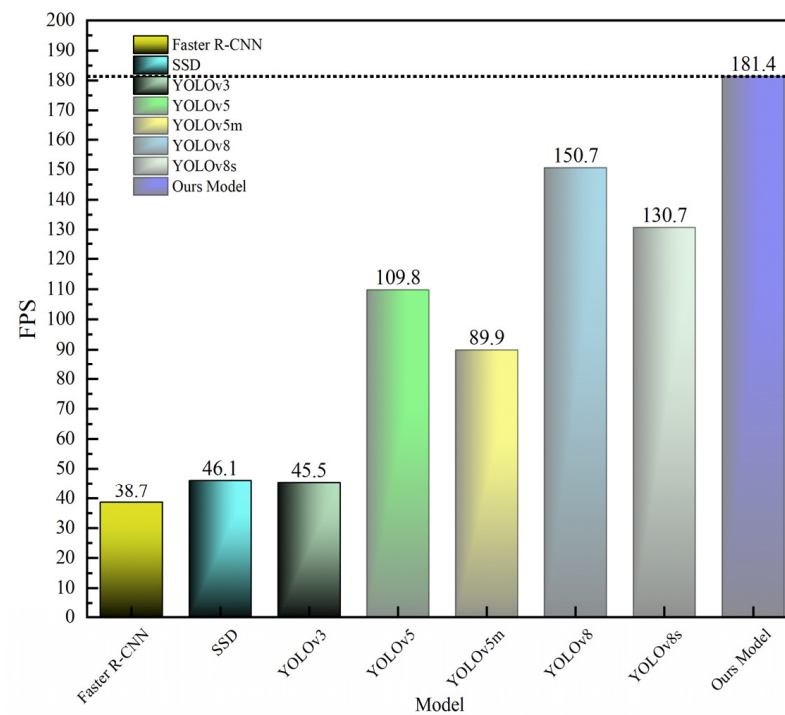


Figure 8. FPS values of different models.

Figure 9 presents a comparison of different object detection models in terms of the number of parameters (Params, in millions). The figure shows significant differences in parameter counts across the models.

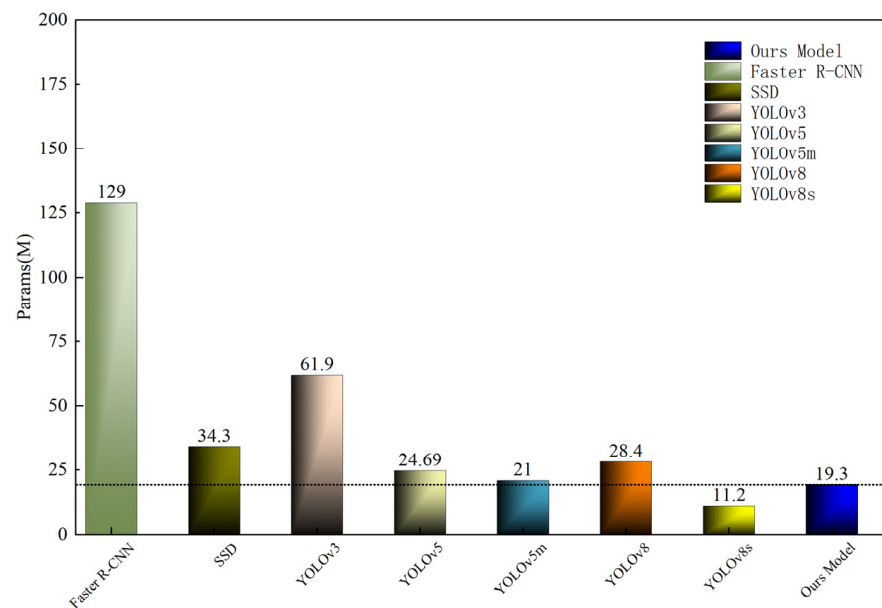


Figure 9. Params values of different models.

Specifically, Faster R-CNN has the highest parameter count, reaching 129 million. This indicates that the model is highly complex, likely requiring more computational resources and storage space, and may perform better on large datasets. The SSD model has 34.3 million parameters, fewer than Faster R-CNN, suggesting that it is relatively lightweight and may perform well in applications with high real-time requirements.

The YOLO series models exhibit variations in parameter counts across different versions. The YOLOv3 model has 61.9 million parameters, indicating a balance between performance and computational complexity. YOLOv5 has 24.69 million parameters, and YOLOv5m has 21 million parameters, demonstrating further optimization in parameter efficiency while maintaining high performance. YOLOv8 and YOLOv8s have 28.4 million and 11.2 million parameters, respectively, with YOLOv8s having the fewest parameters, showcasing significant optimization for lightweight design, making it highly suitable for embedded devices or resource-constrained environments.

Finally, our model has 19.3 million parameters, indicating a focus on reducing parameter count while potentially maintaining high performance, which is advantageous for applications requiring efficiency and low resource usage.

Figure 10 shows the mAP@0.5 (%) of different object detection models during the training process as a function of Epochs. It can be observed that our model consistently performs exceptionally well throughout the training process, quickly reaching and maintaining a high mAP value close to 0.9, significantly higher than other models.

Faster R-CNN and SSD exhibit slower mAP growth, eventually stabilizing around 0.7. While YOLOv3, YOLOv5, YOLOv5m, YOLOv8, and YOLOv8s demonstrate faster growth in the initial stages of training, their final mAP values remain lower than our model's.

Overall, our model shows significant advantages in both training speed and final accuracy, highlighting its superior performance in object detection tasks.

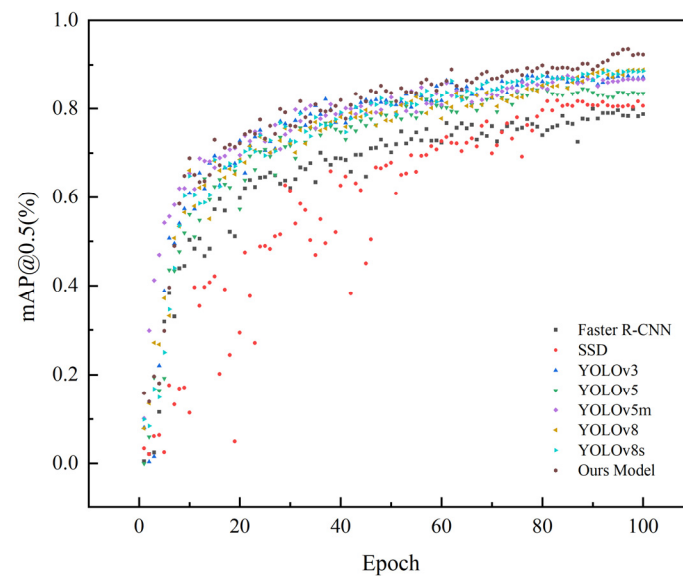


Figure 10. mAP values of different models.

4.4.2. Effect of AFPN on YOLOv8

AFPN (Adaptive Feature Pyramid Network) is an advanced feature fusion method that adaptively combines multi-scale feature information, enhancing the model's ability to detect multi-scale objects. By introducing AFPN into YOLOv8, we aim to significantly improve the model's detection accuracy in complex scenes, particularly in small object detection and occluded object detection.

In this study, we conducted a series of experiments comparing the performance of R-CNN, SPP, YOLOv3-tiny, YOLOv5m, YOLOv8s, and the YOLOv8 model with AFPN on multiple standard datasets to evaluate the effectiveness of AFPN. For this experiment, we used a dataset provided by a power supply bureau in Liaoning Province, consisting of 1895 photos. The foreign objects in these photos include trash, ribbons, sunshades, and bird nests.

We will use R-CNN, SPP, YOLOv3-tiny, YOLOv5m, YOLOv8s, and our proposed model to observe their performance on the dataset provided by the Jilin City power supply company, as shown in Table 5.

Table 5. Performance of our model and other models on the dataset.

Model	AP (%)				mAP@0.5 (%)
	Trash	Ribbon	Shade Net	Nest	
R-CNN	78.2	75.1	73.4	76.0	75.6
SPP	80.5	77.8	75.9	78.2	78.1
YOLOv3-tiny	82.3	79.6	77.5	80.1	79.8
YOLOv5m	85.4	81.2	79.8	82.7	82.1
YOLOv8s	87.1	83.4	82.6	84.5	84.4
Ours Model	91.2	90.5	89.3	92.0	90.7

Figure 11 shows the AP (Average Precision) values of different object detection models on the dataset provided by the power supply company in Jilin City, measuring their detection performance across different categories (trash, ribbons, sunshades, and bird nests). It can be observed that there are significant differences in AP values across R-CNN, SPP, YOLOv3-tiny, YOLOv5m, and YOLOv8s models.

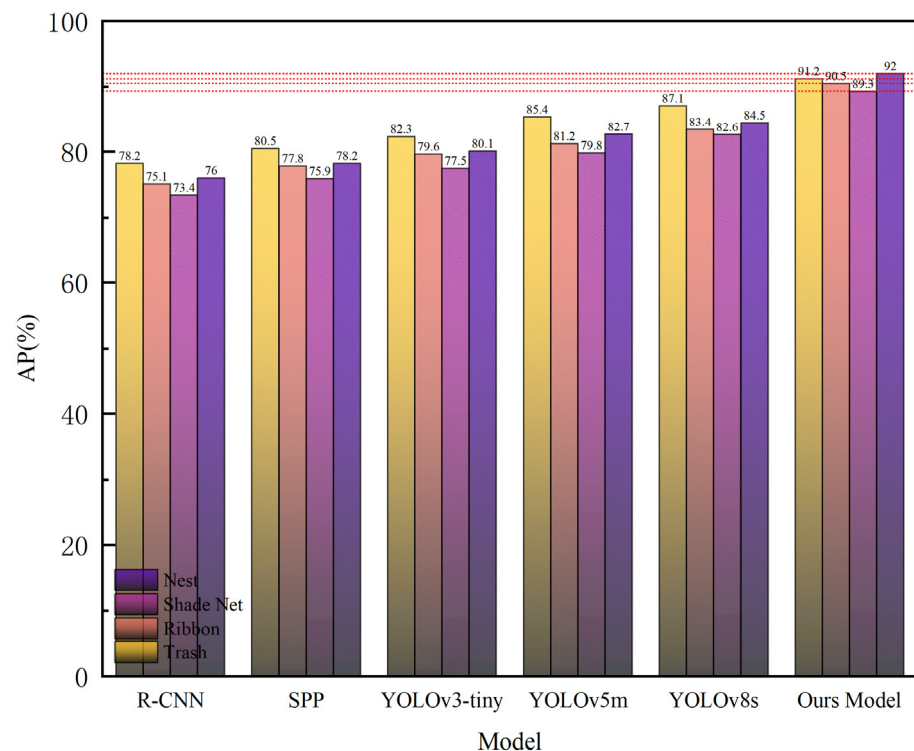


Figure 11. AP performance of different models on the dataset.

R-CNN achieved AP values of 78.2%, 75.1%, 73.4%, and 76% for trash, ribbons, sunshades, and bird nests, respectively. SPP showed a slight improvement, with AP values of 80.5%, 77.8%, 75.9%, and 78.2% for the same categories. YOLOv3-tiny further increased performance, with AP values of 82.3%, 79.6%, 77.5%, and 80.1%. YOLOv5m performed even better, with AP values of 85.4%, 81.2%, 79.8%, and 82.7%. YOLOv8s also demonstrated excellent AP values, achieving 87.1%, 83.4%, 82.6%, and 84.5%.

Notably, our model (AFPN + YOLOv8) achieved significantly higher AP values across all categories, with 91.2% for trash, 90.5% for ribbons, 89.3% for sunshades, and 92% for bird nests, with an overall average AP value of 90.7%. This indicates that the AFPN + YOLOv8 model significantly outperforms other models in object detection performance on the dataset, especially in complex scenes and small object detection.

Figure 12 shows the mAP@0.5 (%) performance metric of different object detection models on the dataset. mAP@0.5 is a common metric for measuring the accuracy of object detection models, with higher values indicating better detection performance. From the figure, it can be seen that R-CNN has a relatively low mAP of 75.7%; SPP shows a slight improvement with an mAP of 78.1%; YOLOv3-tiny's mAP further increases to 79.9%; YOLOv5m achieves an mAP of 82.3%; YOLOv8s reaches an mAP of 84.4%. In comparison, the AFPN + YOLOv8 (our model) significantly outperforms other models with an mAP of 90.7%.

This substantial performance improvement demonstrates that the combination of AFPN with YOLOv8 provides higher precision in object detection tasks, particularly in handling complex backgrounds and small object detection. This shows that AFPN + YOLOv8 not only has theoretical advantages but also significantly enhances detection performance in practical applications. Therefore, AFPN + YOLOv8 is an efficient and highly accurate object detection solution with broad application prospects.

Table 6 and Figure 13 shows the performance comparison of different object detection models on a dataset of over 1800 images, including R-CNN, SPP, YOLOv3-tiny, YOLOv5m, YOLOv8s, and AFPN + YOLOv8 (our model). The evaluated metrics include Precision, Recall, F1-score, mAP@0.5, GFLOPs (Giga Floating Point Operations per Second), and Inference Speed.

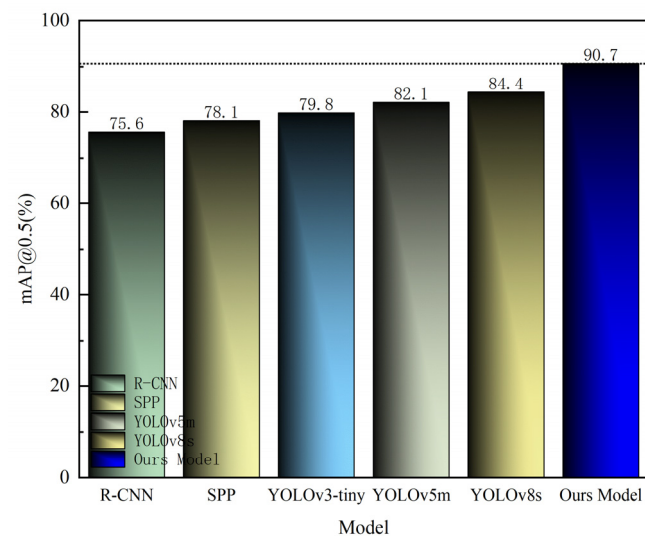


Figure 12. mAP performance of different models on the dataset.

Table 6. Comparison of evaluation metrics for different models.

<i>Model</i>	<i>Precision%</i>	<i>Recall%</i>	<i>F1-Score%</i>	<i>mAP@0.5</i>	<i>FPS</i>	<i>Inference Speed</i>
R-CNN	77.5	75.3	76.4	74.5	180	200
SPP	79.8	77.6	78.7	77.2	90	151.3
YOLOv3-tiny	81.0	78.9	79.9	78.5	17.9	30.8
YOLOv5m	83.2	81.0	82.1	81.0	36	25.6
YOLOv8s	85.0	83.2	84.1	83.5	27	16.5
Ours Model	88.5	87.0	87.7	89.3	14.3	11.2

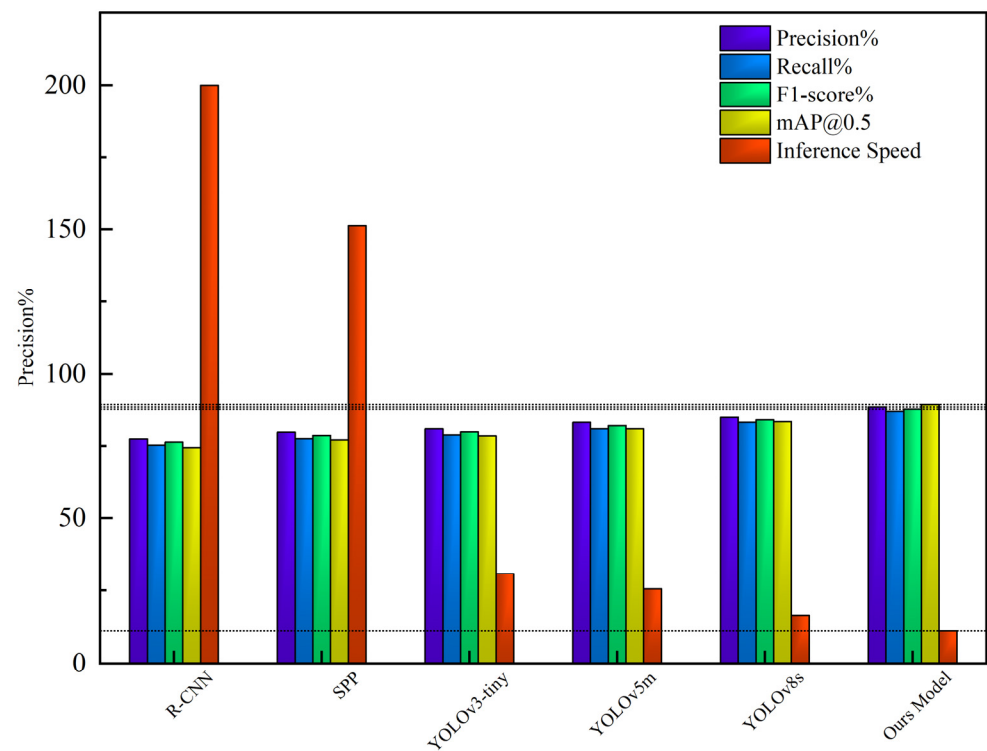


Figure 13. Evaluation metrics of different models.

R-CNN has a Precision of 77.5%, Recall of 75.3%, F1-score of 76.4%, mAP@0.5 of 74.5%, GFLOPs of 180, and Inference Speed of 200 ms, indicating high computational complexity and slower inference speed. SPP shows improved precision at 79.8%, with Recall of 77.6%, F1-score of 78.7%, mAP@0.5 of 77.2%, GFLOPs of 90, and Inference Speed of 151.3 ms, with improved precision but still relatively slow speed. YOLOv3-tiny achieves a Precision of 81.0%, Recall of 78.9%, F1-score of 79.9%, mAP@0.5 of 78.5%, GFLOPs of 17.9, and Inference Speed of 30.8 ms, maintaining low computational complexity and fast inference speed but slightly lower precision. YOLOv5m achieves a balance between precision and speed with a Precision of 83.2%, Recall of 81.0%, F1-score of 82.1%, mAP@0.5 of 81.0%, GFLOPs of 36, and Inference Speed of 25.6 ms. YOLOv8s shows good performance in both precision and speed, with a Precision of 85.0%, Recall of 83.2%, F1-score of 84.1%, mAP@0.5 of 83.5%, GFLOPs of 27, and Inference Speed of 16.5 ms.

AFPN + YOLOv8 (our model) demonstrates outstanding detection performance with a Precision of 88.5%, Recall of 87.0%, F1-score of 87.7%, mAP@0.5 of 89.3%, GFLOPs of 14.3, and Inference Speed of 11.2 ms. It shows excellent detection capabilities, low computational complexity, and very fast inference speed. Overall, AFPN + YOLOv8 outperforms other models in all performance metrics, particularly in Precision, Recall, F1-score, and mAP@0.5, while maintaining low GFLOPs and fast inference speed, making it suitable for applications requiring high accuracy and real-time performance.

4.4.3. Effect of Improved Loss Functions on YOLOv8

As shown in Table 7 and Figures 14–16, we can analyze the impact of different loss functions on model performance during training. It can be seen that the loss values for all loss functions drop rapidly at the beginning of training and gradually stabilize after about 50 epochs. Notably, the loss values for Focal SIoU and Focal CIoU are significantly lower than those of other loss functions throughout the training process, indicating superior performance in both convergence speed and final loss value. The loss value for WIoU is relatively high during training, indicating slower convergence and less effectiveness compared to other loss functions. The performance of CIoU and EIoU is intermediate and stable but not as excellent as the Focal loss functions.

Table 7. Comparison of loss values for different loss functions.

<i>LossFunction Type</i>	<i>Precision%</i>	<i>Recall%</i>	<i>mAP@0.5</i>	<i>Loss</i>
WIoU	88.1	78.2	85.9	0.608
CIoU(Initial)	89.7	79.3	86.7	0.412
EIoU	87.6	80.5	86.3	0.317
Focal CIoU	91.5	80.9	88.2	0.425
Focal SIoU	94.4	82.0	89.9	0.218

Figures 15 and 16 show the mAP@0.5 (mean Average Precision at an IoU threshold of 0.5) trends for different loss functions during training. It can be seen that the mAP@0.5 for all loss functions rises rapidly at the beginning of training and stabilizes after about 100 epochs. Overall, the mAP@0.5 for Focal SIoU and Focal CIoU is slightly higher than other loss functions in the later stages of training, which corresponds with the loss trends in Figure 17, further indicating that these two loss functions perform better in terms of accuracy and robustness. The mAP@0.5 for WIoU fluctuates significantly during training, performing slightly worse than other loss functions. The mAP@0.5 for EIoU and CIoU is similar, even and stable, showing moderate performance.

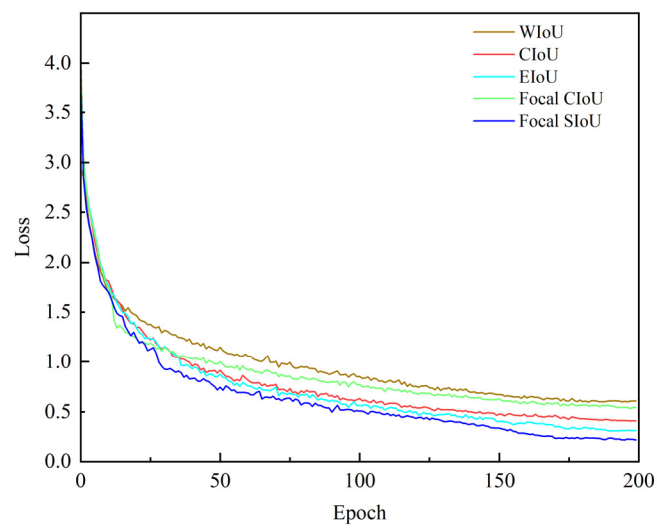


Figure 14. Loss values for different loss functions.

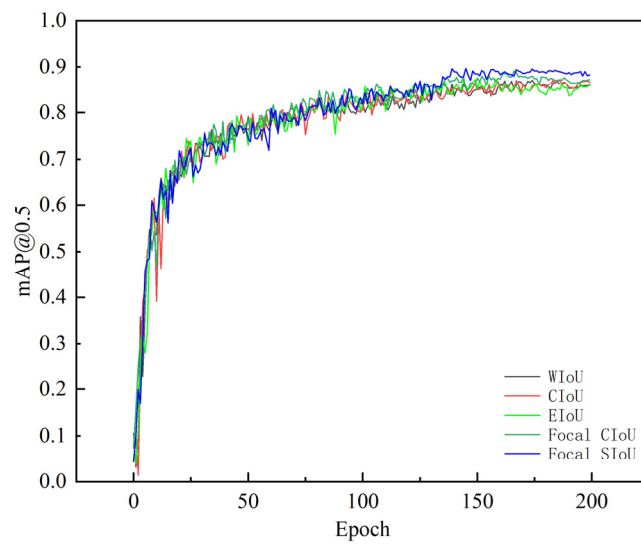


Figure 15. mAP for different loss functions.

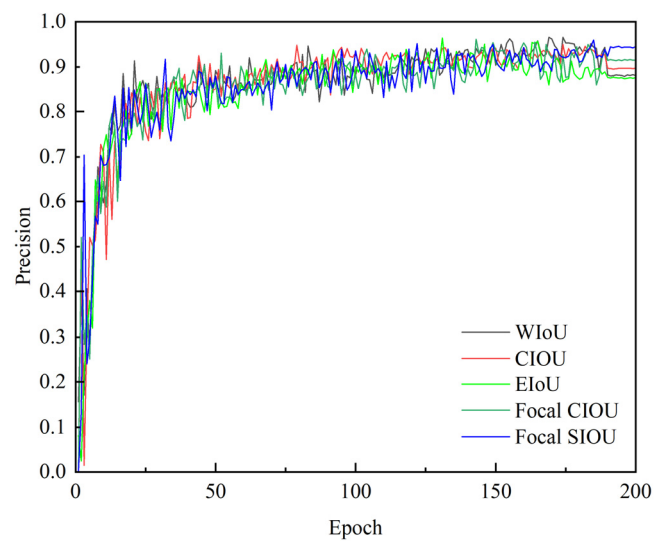


Figure 16. Precision for different loss functions.



(a) Faster R-CNN model



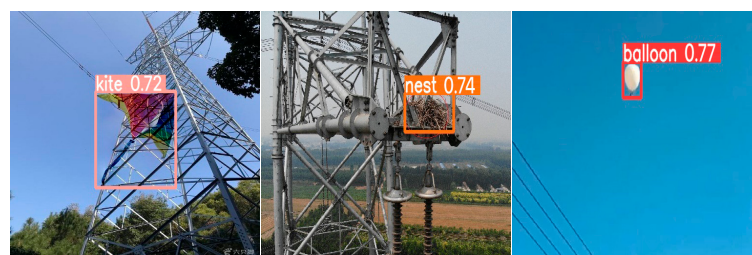
(b) YOLOv3-tiny model



(c) YOLOv3 model



(d) YOLOv5 model



(e) YOLOv7 model

Figure 17. Cont.



(f) YOLOv8 model



(g) Ours model

Figure 17. Compared to other advanced algorithms, our model has a high accuracy in detecting foreign objects.

Through the above analysis, we can see the importance of loss functions in model training. Different loss functions not only affect the convergence speed and final loss value of the model but also significantly impact the model's accuracy. Focal SIoU and Focal CIoU performed excellently in this experiment, outperforming other loss functions in both loss reduction speed and final mAP@0.5 performance. This indicates that Focal loss functions are more efficient and robust in handling object detection tasks. In contrast, WIoU is unsuitable for such tasks due to its unstable performance. EIoU and CIoU performed moderately; although not as superior as the Focal loss functions, their stable performance suggests they still have certain advantages in some application scenarios.

In summary, these results highlight the importance of selecting an appropriate loss function when training object detection models. By choosing the right loss function, model training efficiency and detection performance can be significantly improved, resulting in better outcomes in practical applications.

4.5. Ablation Study

In this section, we conduct an ablation study to analyze the impact of different model components on YOLOv8. An ablation study is a common method to evaluate the specific contributions of various parts of a model by gradually removing or replacing them. Table 8 shows the ablation experiment configurations for our model under seven different experimental setups, including the three main components: Swin Transformer, AFPN, and Focal SIoU. The specific setup of each experiment and the corresponding GFLOPs and mAP@0.5 (%) results are shown in the table. Specifically, Experiment 1 uses only the basic YOLOv8 model as the control group, Experiment 2 adds Swin Transformer to evaluate its performance improvement, Experiment 3 adds the AFPN module, and Experiment 4 introduces the Focal SIoU loss function. In Experiments 5 to 7, we combine these components to observe their synergistic effects on model performance. By comparing these experimental results, we can discuss in detail the impact of each component on the accuracy and computational complexity of the YOLOv8 model, thus guiding us in more effective model design and optimization in practical applications. The ablation experiments are shown in Table 8.

Table 8. Ablation study.

Experiment Number	Model			mAP@0.5 (%)
	SwinTransformer	AFPN	Focal SIoU	
1	✓			82.3
2		✓		81.7
3			✓	81.5
4	✓	✓		83.1
5	✓		✓	84.6
6		✓	✓	86.5
7	✓	✓	✓	89.7

We analyzed the impact of different model components on YOLOv8 performance through ablation experiments. The table shows the specific configurations and their corresponding mAP@0.5 (%) values for the seven experiments. Experiment 1 uses only the basic YOLOv8 model, achieving an mAP@0.5 of 82.3% as the baseline control group. Experiment 2 adds the Swin Transformer, increasing the mAP@0.5 to 81.7%, indicating a significant contribution. Experiment 3 incorporates the AFPN module, resulting in an mAP@0.5 of 81.5%, showing a certain improvement. Experiment 4 introduces the Focal SIoU loss function, raising the mAP@0.5 to 83.1%, confirming its effectiveness. Experiment 5 combines the Swin Transformer and AFPN, boosting the mAP@0.5 to 84.6%. Experiment 6 combines the Swin Transformer and Focal SIoU, achieving an mAP@0.5 of 86.5%. Experiment 7 combines all components, reaching the highest mAP@0.5 of 89.7%, demonstrating the best performance.

4.6. Comparison of the Combined Model with Other Advanced Algorithms

In this section, we will compare our combined model (Swin Transformer + AFPN + Focal SIoU) with other advanced algorithms in terms of precision, recall, mAP, and other performance metrics.

Based on Table 9 and Figure 17, we can observe significant differences among various object detection models in terms of precision, recall, mAP, computational complexity (FPS), and the number of parameters (Params). The Faster R-CNN model has the highest number of parameters, at 129 M, but its precision and recall are relatively low, at 61.3% and 42.5%, respectively. In contrast, YOLOv3-tiny, with only 14.5 M parameters, also shows lower precision and recall, at 71.8% and 69.2%, respectively. The YOLOv3 and YOLOv5 models demonstrate improved performance, especially YOLOv5, which achieves 84.0% precision and 84.0% recall with 24.69 M parameters, indicating a good balance. The YOLOv7 and YOLOv8 models further enhance performance, particularly YOLOv8, which reaches 87.2% precision and 83.9% recall while maintaining a parameter count of 28.4 M, showcasing better optimization.

Table 9. Performance comparison of different models and the combined model.

Model	Model				
	Precision (%)	Recall (%)	mAP@0.5 (%)	FPS	Params (M)
Faster R-CNN	61.3	42.5	74.3	38.7	129
YOLOv3-tiny	71.8	69.2	76.5	46.1	34.3
YOLOv3	82.8	75.5	83.8	45.5	61.9
YOLOv5	84.8	80.4	84.7	109.8	24.69
YOLOv7	86.2	80.45	87.1	117.5	28.67
YOLOv8	87.2	83.9	88.2	150.7	28.40
Ours Model	90.8	89.1	89.7	181.4	28.88

Our combined model performs exceptionally well across all metrics. It achieves a precision of 90.8%, a recall of 89.1%, an mAP of 89.7%, FPS of 181.4, and a parameter count of 28.88 M. Although its computational complexity is higher, its advantages in precision and recall make it highly valuable in practical applications. The image comparisons show that “Our Model” demonstrates higher confidence in detecting various objects (such as kites, nests, and balloons), validating its superior performance in real-world detection tasks.

Overall, our combined model ensures high precision and recall while maintaining a moderate number of parameters and computational complexity, making it an efficient and practical object detection model. These excellent performance metrics make it highly promising for practical applications, especially in scenarios requiring high precision and real-time performance.

5. Conclusions

In this study, we proposed an enhanced YOLOv8-based model for detecting foreign objects on power transmission lines, incorporating the Swin Transformer, AFPN (Asymptotic Feature Pyramid Network), and a novel loss function, Focal SIoU. This work aims to address the challenges of accurate and real-time detection of foreign objects such as bird nests, kites, and balloons, which pose significant risks to power transmission line stability and safety.

The integration of the Swin Transformer into the YOLOv8 backbone network significantly improves feature extraction capabilities. AFPN enhances the multi-scale feature fusion process, effectively integrating information from different levels and improving detection accuracy, especially for small and occluded objects. The introduction of the Focal SIoU loss function optimizes the model’s training process, enhancing its ability to handle hard-to-classify samples and uncertain predictions.

Our experiments demonstrate the superiority of the enhanced YOLOv8 model over other state-of-the-art object detection algorithms. The enhanced YOLOv8 achieves higher average precision (AP) across various categories, such as 91.2% for trash and 92% for bird nests, outperforming models like Faster R-CNN and SSD. The Focal SIoU loss function leads to notable improvements in precision and recall rates.

Despite higher computational complexity (181.4 FPS), our model maintains a moderate number of parameters (28.88 M) and an inference speed of 11.2 ms, making it suitable for real-time applications. The proposed enhancements make YOLOv8 a highly effective solution for detecting foreign objects on power transmission lines, significantly contributing to the safety and reliability of power supply systems.

Author Contributions: Methodology, H.S.; Software, Q.S.; Validation, H.K.; Writing—review & editing, Z.D. and X.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Scientific and Technological Planning Project of Jilin Province (YDZJ202201ZYT5642).

Data Availability Statement: Due to security and confidentiality concerns, the data generated and analyzed during the study on foreign object intrusion in power transmission lines are not available for public disclosure. The data involve sensitive information related to critical infrastructure, and making them public could compromise the safety and stability of the power grid. Therefore, these data will not be shared or made publicly available.

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

References

1. Zhang, D.; Zhang, Z.; Zhao, N.; Wang, Z. A Lightweight Modified YOLOv5 Network Using a Swin Transformer for Transmission-Line Foreign Object Detection. *Electronics* **2023**, *12*, 3904. [[CrossRef](#)]
2. Yu, Y.; Qiu, Z.; Liao, H.; Wei, Z.; Zhu, X.; Zhou, Z. A Method Based on Multi-Network Feature Fusion and Random Forest for Foreign Objects Detection on Transmission Lines. *Appl. Sci.* **2022**, *12*, 4982. [[CrossRef](#)]
3. Wu, M.; Guo, L.; Chen, R.; Du, W.; Wang, J.; Liu, M.; Kong, X.; Tang, J. Improved YOLOX Foreign Object Detection Algorithm for Transmission Lines. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 5835693. [[CrossRef](#)]

4. Tasnim, S.; Qi, W. Progress in Object Detection: An In-Depth Analysis of Methods and Use Cases. *Eur. J. Electr. Eng. Comput. Sci.* **2023**, *7*, 39–45. [[CrossRef](#)]
5. Ren, J.; Wang, J. Overview of Object Detection Algorithms Using Convolutional Neural Networks. *J. Comput. Commun.* **2022**, *10*, 115–132.
6. Mishra, D. Deep Learning based Object Detection Methods: A Review. *J. Comput. Commun.* **2022**, *10*, 115–132.
7. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [[CrossRef](#)]
8. Shi, Y.; Jia, Y.; Zhang, X. FocusDet: An efficient object detector for small object. *Sci. Rep.* **2024**, *14*, 10697. [[CrossRef](#)] [[PubMed](#)]
9. Wang, X.; Wang, A.; Yi, J.; Song, Y.; Chehri, A. Small Object Detection Based on Deep Learning for Remote Sensing: A Comprehensive Review. *Remote Sens.* **2023**, *15*, 3265. [[CrossRef](#)]
10. Min, L.; Fan, Z.; Lv, Q.; Reda, M.; Shen, L.; Wang, B. YOLO-DCTI: Small Object Detection in Remote Sensing Base on Contextual Transformer Enhancement. *Remote Sens.* **2023**, *15*, 3970. [[CrossRef](#)]
11. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep Learning for Generic Object Detection: A Survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [[CrossRef](#)]
12. Zhang, Y.; Shen, T. Small Object Detection with Multiple Receptive Fields. IOP conference series. *Earth Environ. Sci.* **2020**, *440*, 32093.
13. Cheng, G.; Yuan, X.; Yao, X.; Yan, K.; Zeng, Q.; Xie, X.; Han, J. Towards Large-Scale Small Object Detection: Survey and Benchmarks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 13467–13488. [[CrossRef](#)] [[PubMed](#)]
14. Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [[CrossRef](#)]
15. Flores-Calero, M.; Astudillo, C.A.; Guevara, D.; Maza, J.; Lita, B.S.; Defaz, B.; Ante, J.S.; Zabala-Blanco, D.; Moreno, J.M.A. Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review. *Mathematics* **2024**, *12*, 297. [[CrossRef](#)]
16. Kumawat, D.; Abhayankar, D.; Tanwani, S. Exploring Object Detection Algorithms and implementation of YOLOv7 and YOLOv8 based model for weapon detection. *Int. J. Intell. Syst. Appl. Eng.* **2024**, *12*, 877–886.
17. Shao, Y.; Zhang, R.; Lv, C.; Luo, Z.; Che, M. TL-YOLO: Foreign-Object Detection on Power Transmission Line Based on Improved Yolov8. *Electronics* **2024**, *13*, 1543. [[CrossRef](#)]
18. Tang, J.; Li, B. Power System to Prevent External Damage Detection Method. *Acad. J. Sci. Technol.* **2024**, *9*, 189–193. [[CrossRef](#)]
19. Shi, Q.; Deng, K. Privacy-Preserving Detection Method for Transmission Line Based on Edge Collaboration. In Proceedings of the 2023 IEEE 22nd International Conference, Exeter, UK, 1–3 November 2023.
20. Li, M.; Ding, L. DF-YOLO: Highly Accurate Transmission Line Foreign Object Detection Algorithm. *IEEE Access* **2023**, *11*, 108398–108406. [[CrossRef](#)]
21. Liu, Y.; Gao, M.; Zong, H.; Wang, X.; Li, J. Real-Time Object Detection for the Running Train Based on the Improved YOLO V4 Neural Network. *J. Adv. Transp.* **2022**, *2022*, 4377953. [[CrossRef](#)]
22. Nayak, R.; Behera, M.M.; Pati, U.C.; Das, S.K. Video-based Real-time Intrusion Detection System using Deep-Learning for Smart City Applications. In Proceedings of the 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 16–19 December 2019.
23. Chen, W.; Li, Y.; Li, C. A Visual Detection Method for Foreign Objects in Power Lines Based on Mask R-CNN. *Int. J. Ambient. Comput. Intell.* **2020**, *11*, 34–47. [[CrossRef](#)]
24. Sevi, M.; Aydin, İ. Detection of Foreign Objects Around the Railway Line with YOLOv8. *Comput. Sci.* **2023**, *25*, 19–23. [[CrossRef](#)]
25. Shan, H.; Song, Y.; Wang, H.; Chen, Y. Research on Efficient Detection Method of Foreign Objects on Transmission Lines Based on Improved YOLOv4 Network. *J. Phys. Conf. Ser.* **2022**, *2404*, 012040. [[CrossRef](#)]
26. Li, S.; Xie, J.; Zhou, F.; Liu, W.; Li, H. Foreign Object Intrusion Detection on Metro Track Using Commodity WiFi Devices with the Fast Phase Calibration Algorithm. *Sensors* **2020**, *20*, 3446. [[CrossRef](#)] [[PubMed](#)]
27. Wang, Z.; Yuan, G.; Zhou, H.; Ma, Y.; Ma, Y. Foreign-Object Detection in High-Voltage Transmission Line Based on Improved YOLOv8m. *Appl. Sci.* **2023**, *13*, 12775. [[CrossRef](#)]
28. Zhao, L.; Liu, C.A.; Qu, H. Transmission Line Object Detection Method Based on Contextual Information Enhancement and Joint Heterogeneous Representation. *Sensors* **2022**, *22*, 6855. [[CrossRef](#)] [[PubMed](#)]
29. Chien, C.T.; Ju, R.Y.; Chou, K.Y.; Lin, C.S.; Chiang, J.S. YOLOv8-AM: YOLOv8 with Attention Mechanisms for Pediatric Wrist Fracture Detection. 2024, Cornell University Library. *arXiv* **2024**, arXiv:2402.09329.
30. Liu, Q.; Ye, H.; Wang, S.; Xu, Z. YOLOv8-CB: Dense Pedestrian Detection Algorithm Based on In-Vehicle Camera. *Electronics* **2024**, *13*, 236. [[CrossRef](#)]
31. Ren, R.; Sun, H.; Zhang, S.; Wang, N.; Lu, X.; Jing, J.; Xin, M.; Cui, T. Intelligent Detection of Lightweight “Yuluxiang” Pear in Non-Structural Environment Based on YOLO-GEW. *Agronomy* **2023**, *13*, 2418. [[CrossRef](#)]
32. Ju, R.Y.; Cai, W. Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm. *Sci. Rep.* **2023**, *13*, 20077. [[CrossRef](#)]
33. Baek, J.; Chung, K. Swin Transformer-Based Object Detection Model Using Explainable Meta-Learning Mining. *Appl. Sci.* **2023**, *13*, 3213. [[CrossRef](#)]

34. Huo, D.; Kastner, M.A.; Liu, T.; Kawanishi, Y.; Hirayama, T.; Komamizu, T.; Ide, I. Small Object Detection for Birds with Swin Transformer. In Proceedings of the 2023 18th International Conference on Machine Vision and Applications (MVA), Hamamatsu, Japan, 23–25 July 2023.
35. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 10012–10022.
36. Peng, Y.; Ren, J.; Wang, J.; Shi, M. Spectral-Swin Transformer with Spatial Feature Extraction Enhancement for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 2696. [[CrossRef](#)]
37. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
38. Schuster, R.; Batraway, R.; Wasenmüller, O.; Stricker, D. ResFPN: Residual Skip Connections in Multi-Resolution Feature Pyramid Networks for Accurate Dense Pixel Matching. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021.
39. Dimitrakopoulos, P.; Sfikas, G.; Nikou, C. Nuclei Detection Using Residual Attention Feature Pyramid Networks. In Proceedings of the 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), Athens, Greece, 28–30 October 2019.
40. Wang, B.; Ji, R.; Zhang, L.; Wu, Y. Bridging Multi-Scale Context-Aware Representation for Object Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 2317–2329. [[CrossRef](#)]
41. Zhang, D.; Zhang, H.; Tang, J.; Wang, M.; Hua, X.; Sun, Q. Feature Pyramid Transformer. In *Computer Vision—ECCV 2020. ECCV 2020. Lecture Notes in Computer Science*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020.
42. Yang, G.; Lei, J.; Zhu, Z.; Cheng, S.; Feng, Z.; Liang, R. AFPN: Asymptotic Feature Pyramid Network for Object Detection. In Proceedings of the 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, HI, USA, 1–4 October 2023.
43. Hu, D.; Yu, M.; Wu, X.; Hu, J.; Sheng, Y.; Jiang, Y.; Huang, C.; Zheng, Y. DGW-YOLOv8: A small insulator target detection algorithm based on deformable attention backbone and WIoU loss function. *IET Image Process.* **2024**, *18*, 1096–1108. [[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.