ITC 4/54
Information Technology and Control
Vol. 54 / No. 4/ 2025
pp. 1248-1258
DOI 10.5755/j01.itc.54.4.42317

A Novel Method of Object Detection Based on YOLOV8 and PixelAttention in Transmission Lines

Received 2025/07/22 | Accepted after revision 2025/10/15

**HOW TO CITE:** Tian, Q., He, H., Liang, S., Qian, K., Shen, S., Deng, F. (2025). A Novel Method of Object Detection Based on YOLOV8 and PixelAttention in Transmission Lines. *Information Technology and Control*, 54(4), 1248-1258. https://doi.org/10.5755/j01.itc.54.4.42317

# A Novel Method of Object Detection Based on YOLOV8 and PixelAttention in Transmission Lines

**Qingsheng Tian**
Research and Development Center, Yunnan Electric Power Test & Research Institute (Group) Co., Ltd., Kunming, 650011, Yunnan, China

**Houhua He**
College of Information Engineering, Kunming University, Kunming, 650214, Yunnan, China

**Shibin Liang**
Research and Development Center, Yunnan Electric Power Test & Research Institute (Group) Co., Ltd., Kunming, 650011, Yunnan, China

**Kaiguo Qian, Shikai Shen, Fei Deng***
College of Information Engineering, Kunming University, Kunming, 650214, Yunnan, Chinad

**Corresponding author:** feigeoffice@kmu.edu.cn

In recent years, a large amount of debris has appeared on power transmission lines, affecting circuit power supply and endangering human safety. At present, deep learning based object detection algorithms have made continuous progress, but they still cannot meet the practical application requirements of real-time performance. It is necessary to further reduce model complexity and improve detection speed. Therefore, a transmission line debris recognition based on fusion point attention improved YOLOV8 is proposed. The model uses the original YOLOV8 model as the base model and adds PixelAttention in detection to enhance the model's receptive field and multi-scale object perception, thereby improving the model detection accuracy. The experimental results show that the average accuracy of the original model is only 92.3%, while the average accuracy of the improved model has increased to 94%.

KEYWORDS: YOLOV8, target detection, Miscellaneous items on transmission lines, PixelAttention

## 1. Introduction

With the rapid development of the economy and the increasing frequency of human activities, the environment around power transmission lines has become increasingly complex. In addition to natural debris such as tree branches, leaves, and bird nests, there are also a large number of plastic bags, kites, balloons, garbage, and other debris that adhere to power lines [25]. These debris not only affect the appearance of power lines, but also reduce their efficiency, and may even cause a series of serious power accidents, such as short circuits, trips, and other faults. More prominently, icing problems often occur on transmission lines in mountainous or cold regions [14], and debris can seriously exacerbate the shaking of transmission lines, causing them to break and endangering personnel and facilities around them. Therefore, finding a detection method that can quickly and accurately identify impurities in transmission lines has become particularly important for current research.

At present, the YOLO series algorithms [2] are widely used in multiple fields, and many scholars improve the accuracy and recall of models by combining other algorithms [5-7, 11, 13, 18]. Feng Ziwen et al. [4] optimized the backbone structure of YOLOv5 using RepVGG, added convolutional attention mechanism to the network, and introduced a weighted bidirectional feature pyramid network in the feature fusion part. Compared with the original YOLOv5 model, the improved model improved accuracy, recall, and average precision by 3.43%, 1.41% and 3.1%, respectively. Xu Kui et al. [21] proposed the YOLO-ACON Attention algorithm based on the YOLOv5 object detection network, introducing an adaptive judgment activation function and constructing an adaptive attention module in a two round four-way IRNN network to detect the wearing of safety helmets. Compared to the original YOLOv5 model, the algorithm improved accuracy and recall by 7.65% and 5.17% respectively. Zheng Tao et al. [24] used YOLO algorithm combined with Vibe pre-processing foreground algorithm to achieve real-time detection of PCB circuit defects using AOI technology, greatly improving the performance of the model. AodiZhang et al. [22] used the EagleEye pruning method on the YOLOv5s model to search for sub networks and prune channels, fine tune the trimmed model to restore lost accuracy, achieve model compression and solve the problem of multiple parameters and large computational complexity in object detection algorithms.

With the rapid development of artificial intelligence, deep learning has been widely applied in various fields of image recognition and has achieved many accomplishments. Liu Qiang et al. [9] introduced the SENet attention mechanism SE module in generative adversarial networks to repair images. The proposed algorithm achieved good results in image semantics, peak signal-to-noise ratio and structural similarity. Mou Wenqian et al. [16] introduced depthwise separable convolution into capsule networks and added SENet after the depthwise separable convolution layer. The proposed algorithm achieved a recognition accuracy of 94.20%. For the all above networks, SENet is embedded into the substructure of the proposed model, learning the importance of each feature channel through SENet to improve the performance of the model.

Meanwhile, the YOLOV8 model is an open-source object detection algorithm proposed by Ultralytics in 2023, which has wide applications in many fields [8,15,17]. For example, the YOLOV8 model is used to identify the landslide by sensing imagery in mountainous terrain [3]. It is used to detect the surface defects on steel strips successfully [12]. And this model can address the problems of low recognition accuracy and information loss in the progress of small object detection [19]. For avoiding the disadvantage of detecting small targets and handling complex image backgrounds for UAV, an improved version of YOLOv8 is proposed [20]. To challenge the leaf segmentation under the artificial and complex lighting statement, corresponding research of modified model is proposed based on YOLOv8 [23]. However, the YOLOV8 is not kept stable in the progress of usage. More or less, the new algorithm is proposed to address the special problems based on YOLOV8.

In response to the characteristic of transmission line images being in complex environments, this article learns the SENet method and adds a PixelAttention attention mechanism module in the detection part to distinguish between debris on the transmission

line and the surrounding environment using different pixels, giving more attention to debris and suppressing the surrounding environment. Specifically, for the complex transmission line images, the attention mechanism, named as PixelAttention, is used to enhance the feature extraction ability of networks based on YOLOv8. Firstly, the PixelAttention layer uses two two-dimensional convolutions to transform the number of channels for reducing computational complexity and parameter count and maintaining feature richness. Secondly, the PixelAttention is used to put the head part of YOLOV8 as the detect layer, while another layer is convolutional layers. Finally, two-dimensional layers are used again to further compress and extract the features for enhancing the useful features extracted by the network and weakening the interference of useless features. so the model can process images more efficiently, improve image recognition ability, and thus improve the accuracy of model detection. The table 1 shown the definition of variables clearly in Appendix.

## 2. YOLOv8 Model Introduction and Improvement

### 2.1. YOLOV8 Model

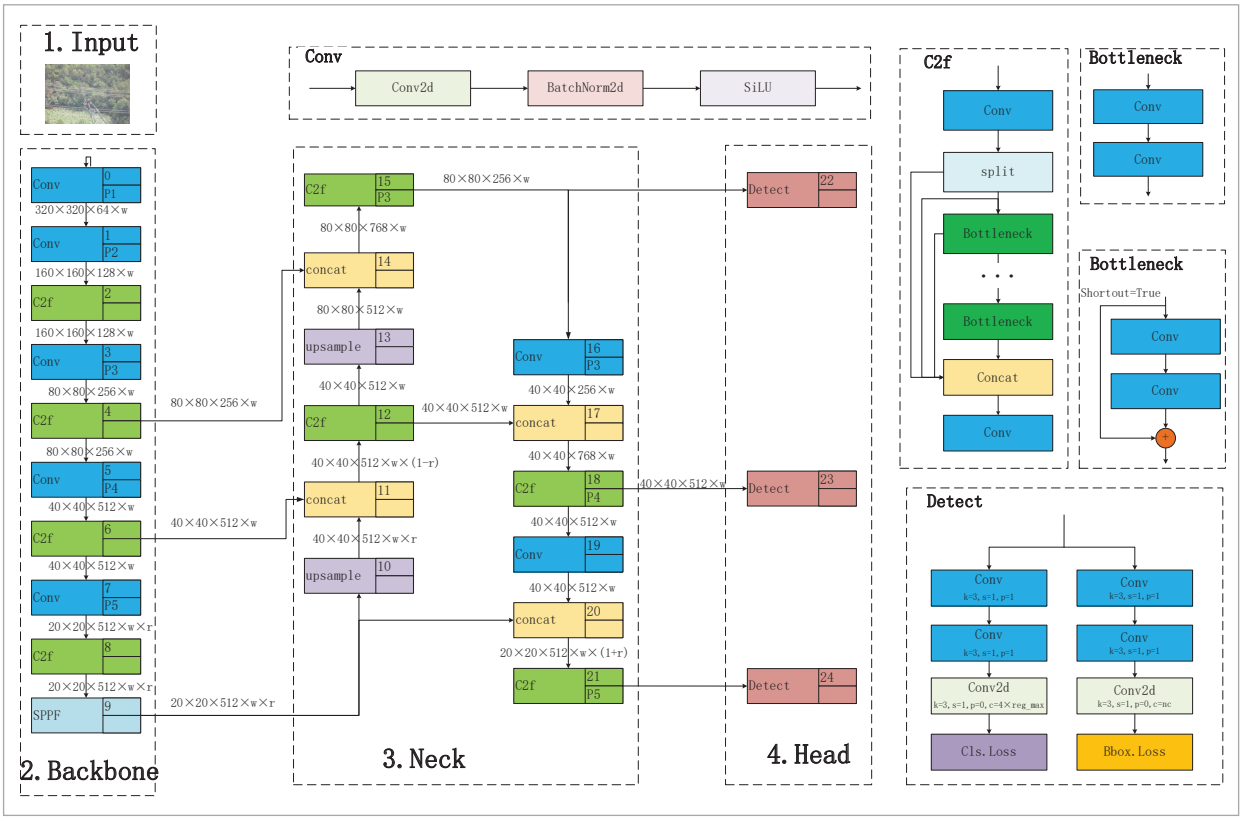
The YOLOV8 model is an open-source object detection algorithm proposed by Ultralytics in 2023, which has wide applications in many fields. The process is divided into four steps: Firstly, the input layer is responsible for inputting the image of the object to be detected into the network; Secondly, the Backbone layer of the network backbone module extracts features from images; Afterwards, the feature enhancement module Neck layer is responsible for pooling and fusing the input feature layers; Finally, it is output by the Head layer at the output end. The structure of the YOLOV8 model is shown in Figure 1. It includes four parts,

#### 2.1.1. The Input Layer

Firstly, fill the short edges of the input image into the network with the same length as the long edges,

**Figure 1**

The structure of the YOLOV8 model.



and then scale it proportionally to a size of 640 × 640. At the same time, in order to enhance the diversity of data and the generalization ability of the model, daaugmentation operations such as random flipping, random cropping, and color transformation are performed on the data. Then normalize the data to eliminate differences between features. Afterwards, convert all channels into 3-channel images, and finally convert them into tensor format.

### 2.1.2. The Backbone Layer

In order to enhance feature extraction capability and reduce computational complexity, YOLOV8 replaced the C3 structure in the Backbone section of YOLOV5 with a C2f structure with richer gradient flow, and adjusted the number of channels for different scale models. Extract features from images through a series of convolutional and pooling layers. At the same time, the Backbone section uses residual connections to alleviate the gradient vanishing problem, allowing the network to learn more deeply. In order to improve the efficiency of the model, the Backbone part not only adopts depthwise separable convolution technology, but also integrates spatial pyramid pooling layer (SPPF). The main operation is as follows:

**a Convolution Operation**

The Backbone layer receives the two-dimensional input feature map I from the Input layer and performs a two-dimensional convolution operation with the convolution kernel K to extract the features of the feature map. Afterwards, batch normalization is performed, which does not change the size of the feature map, but mainly normalizes the data of each small batch. By using the SiLU activation function to process it again, it combines nonlinear and linear characteristics, so that it can alleviate the gradient vanishing problem to some extent. Finally, output the feature map for subsequent operations. The formula for the above process is as follows:

$$(I * K)(x, y) = \sum_m \sum_n I(x-m, y-n) \cdot K(m, n) \tag{1}$$

$$\hat{x} = \frac{x-\mu}{\sqrt{\sigma^2+\varepsilon}} \tag{2}$$

$$SiLU(x) = x \cdot \frac{1}{1+e^{-x}} \tag{3}$$

$$W_{out} = \left\lfloor \frac{w_{in}+2p-k}{s}+1 \right\rfloor \tag{4}$$

$$H_{out} = \left\lfloor \frac{H_{in}+2p-k}{s}+1 \right\rfloor \tag{5}$$

Among them, $(x, y)$ is the position on the output feature map, $(m, n)$ is the position on the convolution kernel, $x$ is the input, $\mu$ is the batch mean, $\sigma^2$ is the batch variance, $\epsilon$ is a small constant, $W_{in}$ and $H_{in}$ are the width and height of the input feature map, $k$ represents the size of the convolution kernel, s represents stride, $p$ represents padding, and $W_{out}$ and $H_{out}$ are the width and height of the output feature map.

**b C2f Module**

First, perform convolution operation on the feature map, then divide the feature map into P parts, and then independently pass each part through a convolution layer before connecting them together. Afterwards, apply a convolution kernel independently to each input channel for deep convolution, and then apply a 1x1 pointwise convolution to the output of all deep convolutions. Perform the final convolution operation on the output after point by point convolution to adjust the number of channels and spatial size. If shortcut is equal to True, the input feature map is added directly to the output for forming a residual block. The formula for the above process is as follows:

$$F_P = Conv2d(X_P, K_P) \tag{6}$$

$$D_{depth} = Conv_{depth}(I, K_{depth}) \tag{7}$$

$$D_{point} = Conv_{point}(D_{depth}, K_{point}) \tag{8}$$

$$O = Conv(D_{point}, K_{out}) \tag{9}$$

$$O_{residual} = O + I \tag{10}$$

Among them, $X_P$ is the segmented P-part feature map, $K_P$ is the corresponding convolution kernel, $F_P$ is the processed feature map, $I$ is the input feature map, $K_{depth}$ is the deep convolution kernel, $D_{depth}$ is the

output after deep convolution, $K_{point}$ is the pointwise convolution kernel, $D_{point}$ is the output after pointwise convolution, $K_{out}$ is the output convolution kernel, $O$ is the final output feature map, and $O_{residual}$ is the result of residual concatenation.

c **SPPF Module**

Firstly, perform convolution operation and perform max pooling operation on the convolved feature map to reduce its size and extract the main features. The SPFF layer will repeat the above steps multiple times, and each repetition will generate a new feature map. Finally, connect all the iteratively generated feature maps to form a richer feature representation. Perform the final convolution operation on the connected feature maps to generate the final output feature map. The formula for the above process is as follows:

$$Y_{pool} = MaxPool(Y) \tag{11}$$

$$Y_1 = Conv(MaxPool(Y_{i-1})) \tag{12}$$

$$Y_{concat} = Concat(Y_1, Y_2, \cdots, Y_n) \tag{13}$$

$$Y_{final} = Conv(Y_{concat}) \tag{14}$$

Among them, $Y_{pool}$ is the output feature map after MaxPool operation, $Y_i$ is the output feature map of the $i$-th iteration, $Y_{i-1}$ is the output feature map of the previous iteration, $Y_{concat}$ is the connected feature map, $Y_1$, $Y_2$,..., $Y_n$ are the feature maps generated by all iterations, and $Y_{final}$ is the final output feature map.

### 2.1.3. The Neck Layer

Use Feature Pyramid Network (FPN) or Path Aggregation Network (PAN) for top-down and bottom-up feature fusion at multiple scales. In addition, a series of convolutional layers, upsampling, and feature map concatenation operations are used to further enhance the expressive power of features.

### 2.1.4. The Head Layer

The Head layer receives multi-scale feature maps from the Neck layer. First, Conv convolution is used to further extract and integrate the features of the image. Then, Conv2d is used to generate the feature map to be predicted. The feature map is then globally averaged and pooled to obtain a channel description vector Z. Then, two fully connected layers are used to transform the channel description vector z to generate channel weights s. Finally, the channel weights are multiplied by the elements of the input feature map to obtain a feature weighted feature map Y.

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_{c,i,j} \tag{15}$$

$$s = \sigma(W_2 \delta(W_1 z)) \tag{16}$$

$$Y = s \cdot F \tag{17}$$

In the formula: $F_{c,i,j}$ represents the pixel values of feature map $F$ in the $c$-th channel, $i$-th row, and $j$-th column, $W_1$ and $W_2$ are learnable parameter matrices, $\sigma$ is the ReLU activation function, and $\delta$ is the Sigmoid activation function.

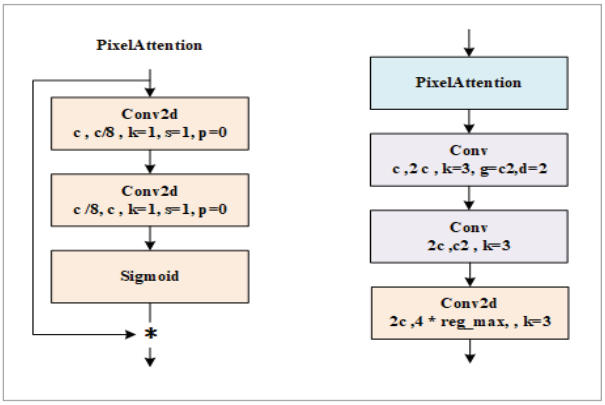
The design adopts a decoupling head structure to separate the classification (Cls) and detection heads (Boxes), and replaces the detection method from Anchor Based to Anchor Free. This approach simplifies the detection process while reducing dependence on anchor boxes, and enables the network to better adapt to target objects of different shapes and sizes.

## 2.2. Model Improvment

In transmission line images, the background is often too complex and the target types are diverse, making it difficult for the network to learn the key features of the target. Attention mechanism is used to enhance the feature extraction ability of networks, which is widely applied in deep learning machine and learning tasks. The attention mechanism can help the model better focus on and process key information in the image, thereby improving the performance of the model.

The PixelAttention class was first proposed by Jie Hu, Li Shen, and Gang Sun in their paper "Squeeze and Excitation Networks" in CVPR2018 as a pixel attention mechanism [23]. In order to enable the network to more efficiently focus on key features,

**Figure 2**
The modified version of PixelAttention.




this paper introduces PixelAttention into the head part of YOLOV8, enhancing the useful features extracted by the network and weakening the interference of useless features. The modified version is shown in Figure 2.

The PixelAttention layer uses two two-dimensional convolutions to transform the number of channels. The first two-dimensional convolution is used to reduce the number of channels, thereby reducing computational complexity and parameter count, while extracting features. Then, two two-dimensional convolution layers are used to restore the number of channels to the original number, in order to maintain feature richness [10,15]. The output from two convolutional layers is passed through the Sigmoid function to obtain a weight between 0 and 1, which is used to weight the original feature map. Multiply the output of the Sigmoid layer with the original feature map to weight the features, suppress unimportant features and highlight important features.

Add PixelAttention module to the Detect layer, then use convolutional layers to increase the number of channels and group convolution to extract richer features, while expanding convolution increases the receptive field. In order to reduce computational complexity and prevent overfitting on training data, while further extracting features, convolutional layers are used to reduce the number of channels. Finally, two-dimensional convolution is used again to further compress and extract the features extracted in the first few layers, and the output of the convolutional layer is converted into the final classification result.

# 3. Experimental Results and Analysis

## 3.1. Dataset Introduction

The dataset used in this article consists of 1300 images, including four common types of debris such as bird nests, kites, balloons, and garbage, as well as other complex and diverse environments, which can be got from the URL (https://pan.baidu.com/s/1SwZMnP5YrQYCVo2P99c_Tg?pwd=G3Wu). Multiple methods such as random scaling, affine transformation, color transformation, and horizontal flipping are used to achieve data augmentation. In order to better fit the network model, the dataset is uniformly scaled into 640 × 640 images and divided into training, validation, and testing sets in a 6:2:2 ratio. Table 2 shows some important settings about training environment parameter , where epoch is the number of training epochs, batch_size represents the number of samples used to calculate gradients and update model weights in each iteration, lr0 represents the initial learning rate, box represents the weight of box loss, and cls represents the weight of classification loss.

**Table 2**
Training Environment Parameter Settings.

| epoch | batch_size | lr0 | box | cls |
|---|---|---|---|---|
| 300 | 24 | 0.01 | 7.5 | 0.5 |

## 3.2. Results

### 1 Parameter Quantity

The size of the parameter count directly determines the complexity of the model. A model with a larger number of parameters can capture more details and complex patterns in the data. The parameter comparison between the original model and the improved model is shown in Table 3.

**Table 3**
Comparison of Parameter Quantities.

| parameter | yolov8s original model | yolov8s improved model | yolov8m original model | yolov8m improved model |
|---|---|---|---|---|
| layers | 183 | 195 | 230 | 218 |
| parameters | 11132956 | 11723852 | 26672620 | 25842076 |
| gradients | 0 | 0 | 0 | 0 |
| GFLOPs | 28.5 | 29.8 | 81.1 | 78.7 |

From the table, it can be seen that the YOLOV8S improved the computational complexity by adding more layers and parameters. The YOLOV8m improved model, on the other hand, reduces computational complexity by decreasing the number of layers and parameters.

## 2 Magnitude of the Loss

During the training process, the YOLOv8 model simultaneously uses multiple loss functions to optimize its performance. Classification loss is the use of binary cross entropy loss to calculate the difference between predicted and true categories. The localization loss uses smoothed L1 loss to calculate the difference between the predicted bounding box position and the true bounding box position. The size loss uses smoothed L1 loss to calculate the difference between the predicted bounding box size and the actual bounding box size. The final loss function is the weighted sum of the aforementioned losses. By minimizing the final loss, the model can be effectively trained to improve the accuracy and efficiency of object detection.

$$L_{cls} = \frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad (18)$$

$$L_{loc} = \frac{1}{N}\sum_{i=1}^{N} SmoothL1(loc_{pred}^{i}, loc_{true}^{i}) \quad (19)$$

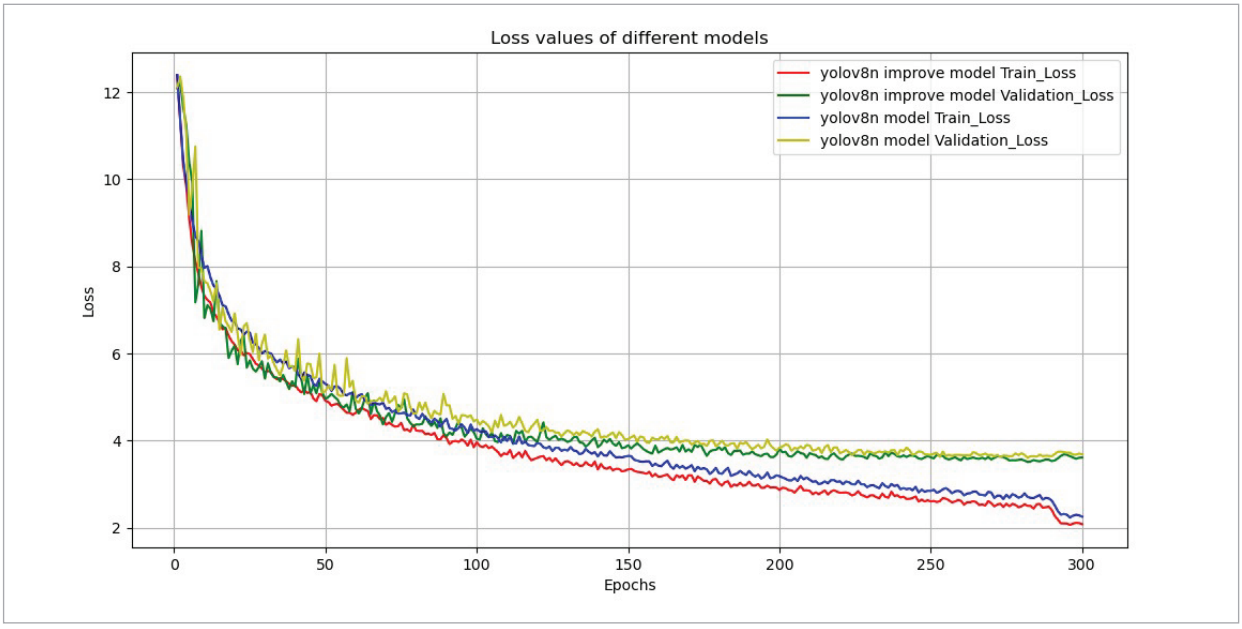
$$L_{size} = \frac{1}{N}\sum_{i=1}^{N} SmoothL1(size_{pred}^{i}, size_{true}^{i}) \quad (20)$$

$$L_{total} = \alpha \cdot L_{cls} + \beta \cdot L_{loc} + \gamma \cdot L_{size} \quad (21)$$

Among them, $N$ is the number of samples, $y_i$ is the true label, $\hat{y}_i$ is the prediction probability, $loc_{pred}^{i}$ is the predicted bounding box position, $loc_{true}^{i}$ is the true bounding box position, $size_{pred}^{i}$ is the predicted bounding box size, $size_{true}^{i}$ is the true bounding box size, $\alpha$, $\beta$ and $\gamma$ are the weights of classification loss, localization loss, and size loss, respectively.

From Figure 3, for improved YOLOv8n, red line and green line are about the curve of train_loss and validation_loss respectively, while the blue line and yellow line are the YOLOv8n models'. It can be seen that the training loss and validation loss of all models gradually decrease with the increase of training epochs, but the improved model exhibits good model performance during both training and validation processes.

**Figure 3**
The Loss of yolov8n model and yolov8n improve model.



That is to say, the red line kept the lower condition of blue line. In a similar way, the green line and yellow line are the same statement. Compared with the original model, the improved model can converge to lower loss values faster and maintain lower volatility throughout the entire training process.

**3 Average Accuracy**

In terms of accuracy evaluation, accuracy P and recall R are selected as evaluation indicators.

$$P = \frac{TP}{TP + FP} \tag{22}$$

$$R = \frac{TP}{TP + FN} \tag{23}$$

In the formula, TP refers to the number of samples that are actually positive and tested as positive, while FP refers to the number of samples that are actually negative and tested as positive. FN refers to the number of negative samples that are actually measured as positive samples.

After improving the models of multiple versions of YOLOV8, their average accuracy is shown in Table 4.

**Table 4**
Comparison of Average Accuracy.

| Model | mAP@0.5 |
|---|---|
| yolov8n original model | 0.923 |
| yolov8n improved model | 0.940 |
| yolov8s original model | 0.928 |
| yolov8s improved model | 0.947 |
| yolov8m original mode | 0.945 |
| yolov8m improved model | 0.945 |

The two smaller scale models YOLOv8n and YOLOv8s have shown significant improvements in average accuracy after improvement. Among them, the average accuracy of YOLOv8n increased from 92.3% to 94%, while the average accuracy of YOLOv8s increased from 92.8% to 94.7%. This indicates that these improvements are particularly effective in enhancing the detection performance of small-sized models, as the addition of PixelAttention layer helps improve the model's feature extraction capability. However, for the medium-sized model YOLOv8m, its average accuracy still remains at 94.5% after improvement, which may be because the model has reached a relatively stable state.

## 4. Conclusion

For identifying impurities in transmission lines, this article proposed an improved YOLOV8 based on fusion point attention. Compared to the traditional YOLOV8, the PixelAttention is used as two-dimensional convolutions to transform the number of channels for reducing computational complexity and parameter count and maintaining feature richness. So that it can assign different weights to different pixels, allow the model to focus on the environment around power lines, enhance sensitivity to target features, and improve the recognition effect of debris on power lines. Compared with the traditional YOLOV8, the improved model has an average accuracy 94%, while the adverse accuracy is 92.3%. The whole accuracy rate is improved 2.3%. In future research, the new direction is to adjust the attention calculation method or parameter settings to adapt to the diversity of impurities in transmission lines in different scenarios, and continuously improve the accuracy of the model.

### Acknowledgement

This work was supported in part by the Special Basic Cooperative Research Programs of Yunnan Provincial Undergraduate Universities Association under Grant No. 202101BA070001-149, and in part by the Academician Weiming Shen Workstation under Grant No. 202505AF350084.

### Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

### Data Sharing Agreement

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

# Appendix

**Table 1**
Notation

| Variables | Definition |
|---|---|
| (x,y) | Position on the output feature map |
| x | Input value |
| y | Output value |
| $\sigma^2$ | Batch variance |
| $\epsilon$ | A small constant |
| $W_{in}$ | Width of input feature map |
| $H_{in}$ | Height of input feature map |
| k | The size of the convolution kernel |
| s | Step length |
| p | Fill |
| $W_{out}$ | Width of output feature map |
| $H_{out}$ | Height of output feature map |
| P | Number of cutting feature map |
| $X_P$ | P-th part of cutting feature map |
| $K_P$ | Convolution kernel of $X_P$ |
| $F_P$ | Addressed feature map |
| I | Input feature map |
| $K_{depth}$ | Deep convolution kernel |
| $D_{depth}$ | Output of deep convolution kernel |
| $K_{point}$ | Pointwise convolution kernel |
| $D_{point}$ | Output of pointwise convolution kernel |
| $K_{out}$ | Output convolution kernel |
| O | Output feature map |
| $O_{residual}$ | Result after residual connection |
| $Y_{pool}$ | Output feature map after MaxPool |
| $Y_i$ | Output feature map of i-th iteration |
| $Y_{i-1}$ | Output feature map of (i-1)-th iteration |
| $Y_{concat}$ | Connected feature map |
| $Y_1,Y_2,...,Y_n$ | Output feature map of all iterations |
| $Y_{final}$ | Final output feature map |
| Z | Channel description vector |

| Variables | Definition |
|---|---|
| s | Channel weights |
| Y | Feature map |
| $F_{c,i,j}$ | The pixel values of the feature map F in c-th channel, i-th row, and j-th column |
| $W_1$ | parameter matrix No.1 |
| $W_2$ | parameter matrix No.2 |
| $\sigma$ | Activation function ReLU |
| $\delta$ | Activation function Sigmoid |
| epoch | Training rounds |
| batch_size | Number of samples used to calculate gradients and update model weights in each iteration |
| lr0 | Initial learning rate |
| box | Weight of box loss |
| cls | Weight of classification loss |
| N | Number of sample |
| $y_i$ | True label |
| $\hat{y}_i$ | Predicted probability |
| $loc^i_{pred}$ | Position of predicted bounding box |
| $loc^i_{true}$ | Position of true bounding box |
| $size^i_{pred}$ | Size of predicted bounding box |
| $size^i_{true}$ | Size of true bounding box |
| $\alpha$ | Weights of classification loss |
| $\beta$ | Weights of localization loss |
| $\gamma$ | Weights of size loss |
| P | Accuracy rate |
| R | Recall rate |
| TP | Number of samples that are actually positive and tested as positive |
| FP | Number of samples that are actually negative and tested as positive |
| FN | Nmber of samples that are actually negative and tested as positive samples |

## References


1. A'La, F. Y., Firdaus, N., Hartatik, A., Setiawan, D. Precision in Safety: YOLOv9 vs. YOLOv10 for Helmet Image Detection. 2024 International Visualization, Informatics and Technology Conference (IVIT), 2024, 159-164. https://doi.org/10.1109/IVIT62102.2024.10692595
2. Birogul, S., Temür, G., Kose, U. YOLO Object Recognition Algorithm and "Buy-Sell Decision" Model Over 2D Candlestick Charts. IEEE Access, 2020, 8, 91894-91915. https://doi.org/10.1109/ACCESS.2020.2994282
3. Chen, X., Liu, C., Wang, S., Deng, X. LSI-YOLOv8: An Improved Rapid and High Accuracy Landslide Identification Model Based on YOLOv8 From Remote Sensing Images. IEEE Access, 2024, 12, 97739-97751. https://doi.org/10.1109/ACCESS.2024.3426040
4. Feng, Z. W., Feng, Y. X. Human Fall Detection Algorithm Based on Improved YOLOv5. Electronic Design Engineering, 2025, 33(1), 1-6. DOI: 10.14022/j.issn1674-6236.2025.01.001.
5. Gašparović, B., Mauša, G., Rukavina, J., Lerga, J. Evaluating YOLOv5, YOLOv6, YOLOv7, and YOLOv8 in Underwater Environment: Is There Real Improvement. 2023 8th International Conference on Smart and Sustainable Technologies (SpliTech), 2023, 1-4. https://doi.org/10.23919/SpliTech58164.2023.10193505
6. He, Y., Li, J., Chen, S., Xu, Y., Liang, J. Waste Collection and Transportation Supervision Based on Improved YOLOv3 Model. IEEE Access, 2022, 10, 81836-81845. https://doi.org/10.1109/ACCESS.2022.3195995
7. He, Z., Wang, K., Fang, T., Su, L., Chen, R., Fei, X. Comprehensive Performance Evaluation of YOLOv11, YOLOv10, YOLOv9, YOLOv8 and YOLOv5 on Object Detection of Power Equipment. 2025 37th Chinese Control and Decision Conference (CCDC), 2025, 1281-1286. https://doi.org/10.1109/CCDC65474.2025.11090973
8. Hu, J., Shen, L., Sun, G. Squeeze-and-Excitation Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745
9. Jin, L. Few-Shot Learning on Edge Devices Using CLIP: A Resource-Efficient Approach for Image Classification. Information Technology and Control, 2024, 53(3), 833-845. https://doi.org/10.5755/j01.itc.53.3.36943
10. Li, X., Nie, T., Zhang, K., Wang, W. Multi-Object Recognition Method Based on Improved YOLOv2 Model. Information Technology and Control, 2021, 50(1), 13-27. https://doi.org/10.5755/j01.itc.50.1.25094
11. Lin, Q., Li, J., Lu, C., Chen, H., Zhou, C. MobileNet-YOLOv2 Mask Wearing Detection Method Fused With Attention Mechanism. 2024 2nd International Conference on Intelligent Perception and Computer Vision (CIPCV), 2024, 82-86. https://doi.org/10.1109/CIPCV61763.2024.00024
12. Liu, H., Hu, R., Dong, H., Liu, Z. SFC-YOLOv8: Enhanced Strip Steel Surface Defect Detection Using Spatial-Frequency Domain-Optimized YOLOv8. IEEE Transactions on Instrumentation and Measurement, 2025, 74, 1-11, Art. no. 9700111. https://doi.org/10.1109/TIM.2025.3548193
13. Liu, J., Qiu, Y., Ni, X., Shi, B., Liu, H. Fast Detection of Railway Fastener Using a New Lightweight Network Op-YOLOv4-Tiny. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(1), 133-143. https://doi.org/10.1109/TITS.2023.3305300
14. Ma, X. W. Research on Icing Faults and Prevention of Transmission Lines in Mountainous Areas. Proceedings of the 3rd Academic Exchange Conference on Power Engineering and Technology, 2023, 235-241. DOI: 10.26914/c.cnkihy.2023.012787.
15. Ma, X., Dai, X., Bai, Y., Wang, L. Rewrite the Stars. GitHub Repository, 2024. URL: https://github.com/ma-xu/Rewrite-the-Stars https://doi.org/10.1109/CVPR52733.2024.00544
16. Mu, W. Q., Dong, M. P., Sun, W. J., Yang, X. X., Wang, X. M. Image Recognition for Tea Leaf Disease Based on Improved Capsule Network and SENet. Journal of Shandong Agricultural University (Natural Science Edition), 2021, 52(1), 23-28.
17. Pandey, S., Chen, K.-F., Dam, E. B. Comprehensive Multimodal Segmentation in Medical Imaging: Combining YOLOv8 with SAM and HQ-SAM Models. 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2023, 2584-2590. https://doi.org/10.1109/ICCVW60793.2023.00273
18. Simic, N., Gavrovska, A. Comparative Analysis of YOLOv11 and YOLOv12 for AI-Powered Aerial People Detection. 2025 12th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN), 2025, 1-4. https://doi.org/10.1109/IcETRAN66854.2025.11114306

19. Tao, S., Shengqi, Y., Haiying, L., Jason, G., Lixia, D., Lida, L. MIS-YOLOv8: An Improved Algorithm for Detecting Small Objects in UAV Aerial Photography Based on YOLOv8. IEEE Transactions on Instrumentation and Measurement, 2025, 74, 1-12, Art. no. 5020212. https://doi.org/10.1109/TIM.2025.3551917

20. Wang, J., Li, X., Zhang, H., Zhou, Y., Fang, M. DPH-YOLOv8: Improved YOLOv8 Based on Double Prediction Heads for the UAV Image Object Detection. IEEE Transactions on Geoscience and Remote Sensing, 2024, 62, 1-15, Art. no. 5647715. https://doi.org/10.1109/TGRS.2024.3487191

21. Xu, K., Li, X. Z., Zhang, L., Zhang, J. J., Yang, N. Safety Helmet Wearing Detection Algorithm for Distribution Network Construction in Natural Scenarios. Computer Engineering and Applications, 2024, 60(8), 320-328.

22. Zhang, A., Guo, R., Liu, K. A Model Compression Method Based on YOLOv5s Target Detection. 2023 2nd International Conference on Cloud Computing, Big Data Application and Software Engineering (CBASE), 2023, 1-5. https://doi.org/10.1109/CBASE60015.2023.10439075

23. Zhang, D., Li, N., Zhao, M., Xu, K. Improved YOLOv8 Algorithm Was Used to Segment Cucumber Seedlings Under Complex Artificial Light Conditions. IEEE Access, 2025, 13, 83944-83955. https://doi.org/10.1109/ACCESS.2025.3568847

24. Zheng, T., Yuan, X. Y., Zhang, L. Research on AOI Detection Based on YOLO Fused with Vibe Algorithm. Modern Transmission, 2024, (6), 42-46.

25. Zhu, Y. Y. Research on Intelligent Detection Algorithm of Abnormal Bodies in Transmission Lines in Complex Environment. Master's Thesis, Hubei Minzu University, 2024. DOI: 10.27764/d.cnki.ghbmz.2024.000280.