Computing and Informatics, Vol. 44, 2025, 385–407, doi: 10.31577/cai_2025_2_385

YOLO-DTO: AUTOMOTIVE DOOR PANEL FASTENER DETECTION ALGORITHM BASED ON DEEP LEARNING

Xiaohui Wang

School of Control and Computer Engineering
North China Electric Power University
Baoding 071003, China
&
Engineering Research Center of Intelligent Computing for Complex
Energy Systems, Ministry of Education
Baoding, China
e-mail: wangxiaohui@ncepu.edu.cn

Yunshuo Jia

School of Control and Computer Engineering
North China Electric Power University
Baoding 071003, China
e-mail: 954787934@qq.com

Fengjuan Guo*

School of Control and Computer Engineering
North China Electric Power University
Baoding 071003, China
&
Hebei Key Laboratory of Knowledge Computing for Energy and Power
Baoding, China
e-mail: gfj@ncepu.edu.cn

* Corresponding author

Abstract. The common detection of fasteners of automobile door panels is based on the method of template matching, which has the problems of low detection accuracy and poor real-time performance under the influence of different lighting and different placement positions. To improve the detection speed and accuracy of fasteners in complex scenes, a small object detection algorithm, YOLO-DTO (Detect Tiny Object), was proposed based on the YOLOv8 algorithm. Firstly, considering that the algorithm uses strided convolution to compress the input image prematurely, resulting in the loss of fine-grained information in the early stage of the image, which makes it difficult to recover the complete detail information in the subsequent feature fusion process, this paper modifies the convolution module in the early stage of the algorithm and introduces the SPD (SPace-to-Depth) module to reconstruct the early stage of the original algorithm. Secondly, a selective attention module is embedded in the Neck output position of the algorithm to enhance the algorithm's ability to pay attention to the context information of fasteners. Finally, to optimize the regression efficiency of the bounding box, the MPDIoU loss function replaced the CIoU loss function. Experimental results show that the average detection accuracy of the YOLO-DTO algorithm is 98.8 %, which is 9.1 % and 1.7 % higher than that of the template matching method and YOLOv8 algorithm, respectively, which meets the detection standards of factory production lines and has the practical value.

Keywords: Automotive door panel fastener detection, selective attention, loss function, deep learning, context information, YOLO algorithm

Mathematics Subject Classification 2010: 68T45

1 INTRODUCTION

The automobile manufacturing industry is highly automated, necessitating the use of advanced automation technologies across various production stages. To enable unmanned operations in many aspects of manufacturing, reliable detection technology is essential to verify the accuracy of each assembly process [1]. In the production process of automobile door panels, there are higher requirements for installing and detecting fasteners. Automotive door panel fasteners include ultrasonic welding joints, plastic screws, metal screws, and more. The production workshop judges that there are assembly defects in automobile door panels by testing these fasteners. Therefore, this article will provide a thorough investigation into fastener detection.

Currently, the detection of fasteners in the production workshop mainly adopts the template matching method. First, it is necessary to take a standard image of the automobile door panel and preprocess it with grayscale, denoising, sharpening, and more. Then, the edge detection algorithm is used to extract the edge information, the similarity detection algorithm is used for template matching by the edge information, and the Hough transform is used to detect the number and position of the fasteners. Finally, the same method must be used to process the new image on the production line, and the number and position of fasteners in the new image are detected. The fastener detection task is accomplished by comparing two sets of images [2, 3]. To a certain extent, this method solves the problem of early manual visual detection. However, it is limited to the fact that it takes about 8 seconds to detect each picture, which cannot reach the standard of the factory production line, and when the product needs to be replaced, the template needs to be redesigned for the new product, and the work efficiency is reduced. Therefore, the template matching method needs to realize the detection task of fasteners under specific conditions in specific scenes, and the detection quality depends mainly on the quality of the template, and the design of the new template is also time-consuming. To solve the above problems, there is an urgent need for a detection algorithm that significantly improves the detection rate of fasteners while ensuring accuracy.

In recent years, deep neural networks have stood out in the research of many object detection algorithms, which have strong expression ability and learning abilities and can be trained by large-scale data, automatically learning and extracting features to achieve effective task-solving. There are two main types of deep neural network algorithms in the field of object detection: the first type is the two-stage object detection algorithm, including R-CNN [4], Fast R-CNN [5], and Faster R-CNN [6], which first generates candidate boxes through a Region Proposal Network (RPN), and then classifies and regresses the candidate boxes. This algorithm has good accuracy, but the detection speed is low, and the number of parameters is large, so it is unsuitable for edge terminal equipment deployed in automobile production plants. The second type are single-stage object detection algorithms, such as SSD [7] and YOLO [8], which directly complete the generation of candidate boxes and target classification and localization through the network simultaneously. This type of algorithm has its advantages in speed and is suitable for scenarios with high real-time requirements. The YOLO algorithm has been continuously updated, and the accuracy of the YOLOv8 [9] algorithm has been comparable to that of the two-stage object detection algorithm. Due to the small number of parameters and fast detection speed, it is more suitable for deployment on the edge terminal equipment in automobile production plants.

The image taken by the industrial camera in the automobile production workshop is 5 472 × 3 648. However, the approximate pixel of each fastener is 57 × 66, which is 1/5 000 of the pixel size of the whole picture, which will cause the YOLOv8 algorithm to easily ignore the feature information of small fastener targets in the detection process, resulting in missed detection or false detection. For the problem of small object detection [10], Chen et al. [11] introduced CAM [12] (Channel Attention Module) and FPN [13] (Feature Pyramid Network) to improve the SSD algorithm, Ji et al. [14] introduced Swin Transformer [15] to improve the YOLOv5 [16] algorithm, Pan et al. [17] improved the YOLOv5 algorithm by introducing CBAM [18] (Convolutional Block Attention Module) and ASFF [19]

(Adaptively Spatial Feature Fusion) to improve the detection accuracy of small targets.

The above analysis shows two critical problems in the current automotive door panel fastener detection. First of all, the versatility of the current template matching method is not strong, and it is not suitable for assembly line detection in the production workshop. Secondly, because the YOLOv8 algorithm has small target features and substantial similarity in the process of detecting fasteners, it is easy to cause missed detection and false detection. This paper proposes an improved YOLO-DTO (Detect Tiny Object) real-time detection algorithm based on the above discussion. The contributions and benefits of our research are as follows:

1. The backbone structure of the network is designed by introducing the SPD [20] (space-to-depth) module into the network, and more feature information of fasteners is retained in the convolution process of the input image.
2. The SK [21] (selective kernel) module is introduced into the network's neck structure to obtain a wide range of fastener context information and enhance the algorithm's ability to distinguish similar fasteners.
3. To optimize the algorithm's bounding box regression positioning ability, the CIoU [22] loss function was replaced with the MPDIoU [23] loss function.

2 YOLO-DTO DETECTION ALGORITHM

In 2023, Ultralytics proposed the YOLOv8 algorithm. Since the detection task of automobile door panel fasteners requires the deployment of detection terminals on the edge side, a smaller model is suitable, so the YOLOv8n algorithm is used as the basic framework. Regarding network structure, YOLOv8n is divided into four parts: Input, Backbone, Neck, and Head.

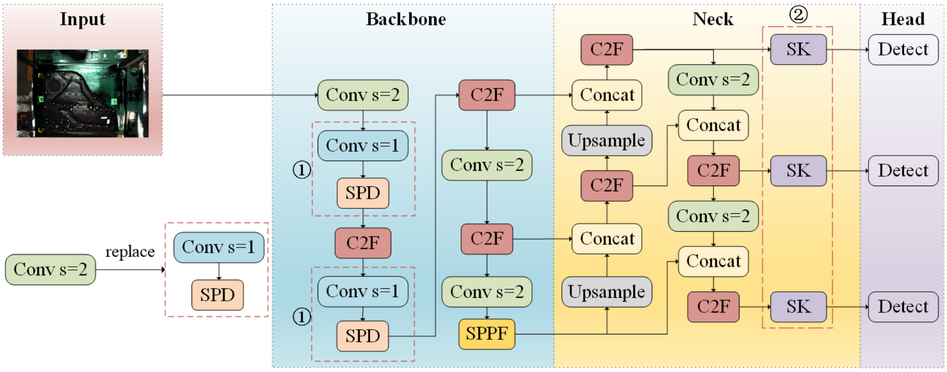


Figure 1. YOLO-DTO network structure

The structure of the YOLO-DTO algorithm proposed by us is presented in Figure 1, and the main improvements are as follows:

1. The step sizes of the second and third strided convolutions of the Backbone were modified to 1, and then the SPD module was introduced after the two convolution blocks, as shown by the mark ① in Figure 1.
2. The SK module has been added to the output of Neck, as shown in the ② marker in Figure 1.
3. Replace the CIoU loss function with the MPDIoU loss function.

2.1 SPD Module

The convolution module in the YOLOv8 algorithm adopts a strided convolution structure with a step size of 2. For the feature map of input $X \in \mathbb{R}^{W \times H \times C}$, after C_2 convolutional kernels with a step size of 2, a feature map with an output of $X' \in \mathbb{R}^{\frac{W}{2} \times \frac{H}{2} \times \frac{C}{2}}$ is obtained, as shown in Figure 2.

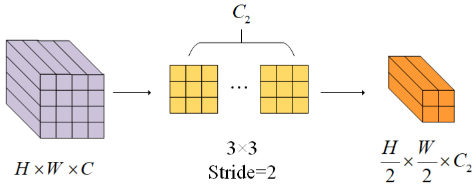


Figure 2. Strided convolution structure

Stridden convolution reduces the size of the input feature map by increasing the stride length, which decreases the number of spatial sampling positions between adjacent pixels in the convolution operation. As a result, small target information may become less noticeable or even completely disappear from the feature map, leading to missed detections. Therefore, to retain more feature information of fasteners in the convolution process of the input image, this paper introduces the SPD module to improve the information loss problem of stridden convolution in the YOLOv8 algorithm.

The SPD module is an algorithm that converts image spatial dimension information into channel dimension information. The feature map is sliced and reorganized in the spatial dimension to obtain a plurality of sub-feature maps with different spatial information. The sub-feature map is then stitched together in the channel dimension. Considering the arbitrary-sized feature map $X' \in \mathbb{R}^{S \times S \times C_1}$, the sub-feature map sequence is sliced as shown in Equation (1).

$$\begin{aligned}
f_{0,0} &= X[0:S:\text{scale}, 0:S:\text{scale}],\\
f_{1,0} &= X[1:S:\text{scale}, 0:S:\text{scale}],\\
&\vdots\\
f_{scle-1,0} &= X[\text{scale}-1:S:\text{scale}, 0:S:\text{scale}];\\
f_{0,1} &= X[0:S:\text{scale}, 1:S:\text{scale}],\\
&\vdots\\
f_{\text{scale}-1,1} &= X[\text{scale}-1:S:\text{scale}, 1:S:\text{scale}];\\
&\vdots\\
f_{0,\text{scale}-1} &= X[0:S:\text{scale}, \text{scale}-1:S:\text{scale}],\\
&\vdots\\
f_{\text{scale}-1,\text{scale}-1} &= X[\text{scale}-1:S:\text{scale}, \text{scale}-1:S:\text{scale}].
\end{aligned} \tag{1}$$

In this work, scale represents the size of the feature map when slicing, scale $= 2$, and f represents a pixel of the image.

When scale $= 2$, input the feature map $X \in \mathbb{R}^{W\times H\times C_1}$, split the feature map into four feature maps, and then stitch the four feature maps in the channel dimension to transform them into feature map X, and realize the down-sampling operation of the image $Y \in \mathbb{R}^{\frac{W}{\text{scale}}\times\frac{H}{\text{scale}}\times\frac{\text{scale}}{2}\times C_1}$, as shown in Figure 3.

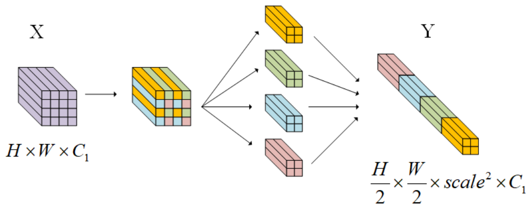


Figure 3. SPD structure

During the down-sampling process, the SPD module retains more spatial information about the feature map. By setting it in the early stage of the algorithm backbone network, the low-level high-resolution feature map can retain more fine-grained information in the feature extraction process, thereby reducing the loss of feature information of small fastener objects.

2.2 SK Module

In fastener detection tasks, automotive door panel images are typically taken with a high-resolution top-down view, and most fasteners are small-dimensional, as shown in Figure 4.

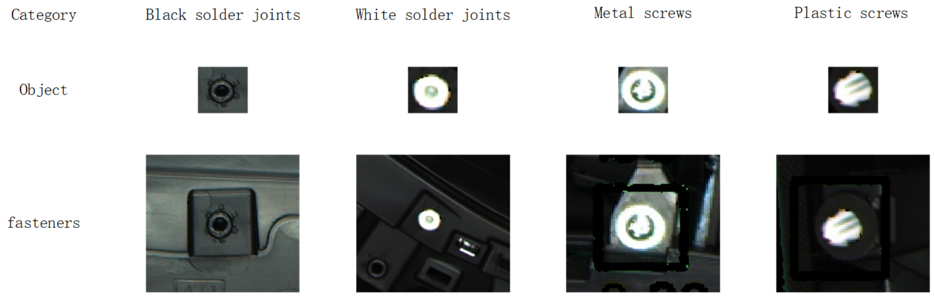


Figure 4. Partial fastener image

White solder joints, metal screws, and plastic screws are very similar in separate scenarios, and the successful detection of these targets is more dependent on their surroundings. The SK module applies the idea of attention mechanism to the convolution kernel, allowing the network to dynamically select the appropriate convolution kernel to help the algorithm better understand different fastener categories in the image.

The SK module consists of three structures: split, fuse, and select, as shown in Figure 5.

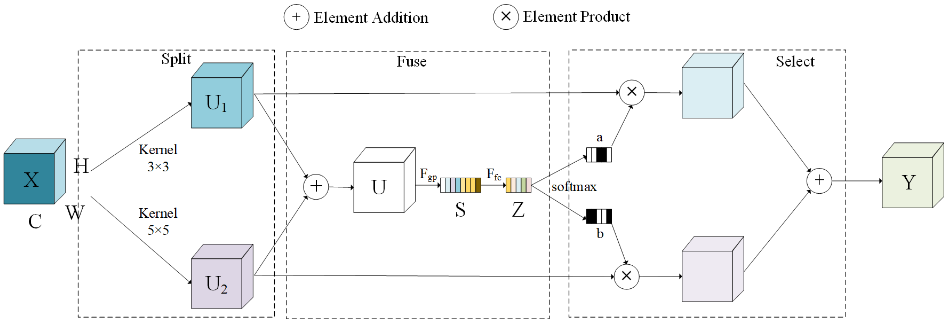


Figure 5. SK module

1) **Split:** Perform two deep convolution operations [24] with different convolution kernels on the input feature map $\mathbf{X}$, and use the convolution kernels of 3×3 and 5×5 to process at the same time to obtain two different feature maps $\mathbf{U_1}$ and $\mathbf{U_2}$.

2) **Fuse:** Add the features of the different feature maps obtained by Split and fuse the information of all branches to obtain the feature map $\mathbf{U}$, then use the global average pooling method to compress all the spatial information into vector $\mathbf{S}$, and finally linearly map the vector $\mathbf{S}$ to vector $\mathbf{Z}$ through full connection, as shown in Equations (2), (3), (4), and (5). This step reduces the information

dimension and better represents the importance of different feature information.

$$\mathbf{U} = \mathbf{U}_1 + \mathbf{U}_2, \tag{2}$$

$$\mathbf{Z} = \mathrm{F}_{\mathrm{fc}}\left(\mathrm{F}_{\mathrm{gp}}(\mathbf{U})\right), \tag{3}$$

$$\mathrm{F}_{\mathrm{gp}}(\mathbf{U}) = \frac{1}{\mathrm{H} \times \mathrm{W}} \sum_{i=1}^{\mathrm{H}} \sum_{j=1}^{\mathrm{W}} \mathbf{U}(i, j), \tag{4}$$

$$\mathrm{F}_{\mathrm{fc}}(\mathrm{g}) = \delta\left(\mathfrak{J}\left(\mathrm{W}_{\mathrm{s}}\right)\right), \tag{5}$$

where F_{fc} is the fully connected layer function, F_{gp} is the global average pooling function, W is the width of the feature graph, H is the height of the feature graph, δ is the ReLU activation function, and $\mathfrak{J}$ is the batch normalization operation. $\mathbf{Z}$ is obtained after $\mathbf{S}$ passing through the fully connected layer, and the information dimensions of $\mathbf{Z}$ are $d \times 1$, $d = \max\left(\frac{C}{r}, L\right)$, r and L are two hyperparameters, r represents the conversion ratio and L represents the minimum value of d. Set $r = 16$, $L = 32$ in this work.

3) **Select:** Perform softmax operation on vector $\mathbf{Z}$ to obtain $\mathbf{a}$ and $\mathbf{b}$, then multiply the feature map $\mathbf{U_1}$ and $\mathbf{U_2}$ obtained by Split with $\mathbf{a}$ and $\mathbf{b}$ to obtain two feature maps $\mathbf{V_1}$ and $\mathbf{V_2}$ with different points of interest, and finally obtain the feature map $\mathbf{Y}$ by adding the features, as shown in Equation (6).

$$\mathbf{Y} = \mathbf{a} \times \mathbf{U_1} + \mathbf{b} \times \mathbf{U_2}. \tag{6}$$

Based on the unique prior knowledge in the fastener detection scenario, the SK module obtains a wider range of fastener context information by dynamically adjusting the spatial receptive field of the model, providing valuable clues about fasteners to the algorithm, thereby effectively improving the algorithm's ability to distinguish similar fasteners.

2.3 Loss Function Design

The bounding box regression loss function measures the accuracy of the algorithm's predictions about the target location and scale. The YOLOv8 algorithm uses the CIoU loss function as the bounding box regression loss function, which calculates the distance and aspect ratio of the center point between the predicted bounding box and the real bounding box based on the IoU [25] (Intersection over Union) loss function.

$$\mathrm{IoU} = \frac{\mathrm{B}_{\mathrm{gt}} \cap \mathrm{B}_{\mathrm{prd}}}{\mathrm{B}_{\mathrm{gt}} \cup \mathrm{B}_{\mathrm{prd}}}, \tag{7}$$

where B_{gt} is the real bounding box, $\mathrm{B}_{\mathrm{prd}}$ is the predicted bounding box, the IoU is the ratio of the intersection and union between the predicted bounding box and the real bounding box.

$$V = \frac{4}{\pi^2}\left(\arctan\frac{w^{\mathrm{gt}}}{h^{\mathrm{gt}}} - \arctan\frac{w^{\mathrm{prd}}}{h^{\mathrm{prd}}}\right)^2, \tag{8}$$

$$\alpha = \frac{V}{1 - \mathrm{IoU} + V}, \tag{9}$$

where V is used to calculate the consistency of the aspect ratios of the prediction and target boxes, α is the parameter for measuring the aspect ratio, w^{gt} and h^{prd} are the width and height of the real bounding box, w^{gt} and h^{prd} are the width and height of the predicted bounding box.

$$\mathrm{CIoU} = \mathrm{IoU} - \frac{\rho^2\left(\mathrm{B_{gt}}, \mathrm{B_{prd}}\right)}{C^2} - \alpha \times V. \tag{10}$$

The complete CIoU loss calculation method is obtained, as shown in Equation (10), where ρ^2 is the Euclidean distance between the center point of the predicted bounding box and the real bounding box, and C^2 is the diagonal length of the smallest closed rectangle that can contain both the predicted bounding box and the real bounding box. The CIoU must calculate the diagonal length of the complete intersection and the complete union, which is highly computationally complex.

The fasteners tested in this paper are mostly square, and the predicted bounding box and the real bounding box may have the same aspect ratio and different heights and widths at this time, as shown in Figure 6 a) and 6 b). In Figure 6 a), the outer frame is the predicted bounding box, the width and height are 4, and the inner frame is the real bounding box, and the width and height are 2; in Figure 6 b), the outer frame is the real bounding box with a width and height of 2, and the inner frame is the predicted bounding box with a width and height of 1, and in both cases, α, V, and ρ^2 are all 0, the CIoU loss function will be degraded into the IoU loss function, which may lead to the reduction of the convergence speed and detection accuracy of the algorithm.

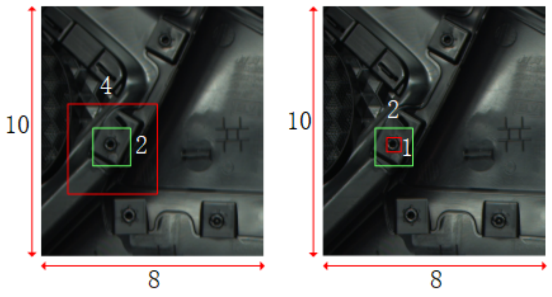
MPDIoU is a bounding box similarity comparison measure based on the minimum point distance, which can compare any two bounding boxes.

$$d_1^2 = \left(x_1^{\mathrm{prd}} - x_1^{\mathrm{gt}}\right)^2 + \left(y_1^{\mathrm{prd}} - y_1^{\mathrm{gt}}\right)^2, \tag{11}$$

$$d_2^2 = \left(x_2^{\mathrm{prd}} - x_2^{\mathrm{gt}}\right)^2 + \left(y_2^{\mathrm{prd}} - y_2^{\mathrm{gt}}\right)^2, \tag{12}$$

$$\mathrm{MPDIoU} = \mathrm{IoU} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2}. \tag{13}$$

where w and h are the widths and heights of the input feature map, $(x_1^{\mathrm{prd}}, y_1^{\mathrm{prd}}, x_2^{\mathrm{prd}}, y_2^{\mathrm{prd}})$ are the coordinates of the upper-left and lower-right corners of the predicted



a) b)

Figure 6. The real bounding box and the predicted bounding box of the fastener

bounding box, $(x_1^{\mathrm{gt}}, y_1^{\mathrm{gt}}, x_2^{\mathrm{gt}}, y_2^{\mathrm{gt}})$ are the coordinates of the upper-left and lower-right corners of the real bounding box, d_1^2 and d_2^2 are the upper-left and lower-right distances between the predicted bounding box and the real bounding box, as shown in Figure 7.

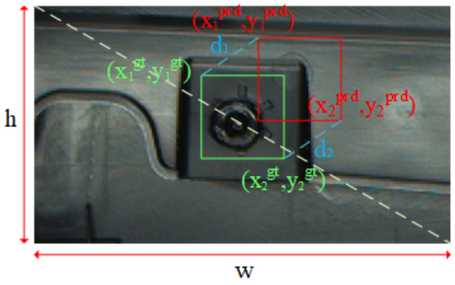


Figure 7. MPDIoU factor of fastener

The points in the upper-left and lower-right corners can uniquely identify a bounding box. The prediction is correct only when the predicted bounding box and the real bounding box exactly coincide, when $d_1^2 = 0$ and $d_2^2 = 0$ appear in Equations (11) and (12). Except that the two bounding boxes coincide precisely, there will be no case where d_1^2 and d_2^2 are 0. Therefore, the MPDIoU loss function does not degenerate to the IoU loss function when the prediction bounding box is regressively located.

The MPDIoU loss function contains all the relevant factors considered in the existing loss functions, such as overlapping or non-overlapping regions, center point distances, width, and height deviations. It simplifies the calculation process and reduces the computational complexity.

3 EXPERIMENTS AND ANALYSIS OF RESULTS

3.1 Experimental Environment

3.1.1 Environment Configuration

This experiment is trained in the Windows 11 environment using the PyTorch deep learning framework, as shown in Table 1.

Configuration	Version
Operating System	Windows 11 x64
CPU	Intel i7-12700H
Running Memory	16 GB
GPU	NVIDIA RTX 3060
Graphics Memory	6 GB
Pytorch	1.12.0
CUDA	11.6
YOLOv8	8.0.125
Python	3.8.18

Table 1. Configuration of the experimental environment

3.1.2 Parameter Optimization

In this work, the model's learning rate is dynamically generated by Equation (14). The model uses a larger learning rate to accelerate convergence at the beginning and gradually decreases it to refine the weight adjustment. The initial learning rate is $lr0 = 0.01$, the learning rate decay is $lrf = 0.125$, and the final learning rate is $lr = 0.00125$.

$$lr = lr0 \times \left[\left(1 - \frac{x}{\text{epochs}} \right) \times (1 - lrf) + lrf \right], \tag{14}$$

where x is the number of rounds currently being trained, and epochs is the total iteration round. This strategy provides flexibility and robustness in the training process, allowing the model to approximate the global optimal solution more effectively.

This paper adopts the weight decay regularization strategy in the optimizer to alleviate the problem of overfitting in the case of less data. We set the hyperparameter momentum to accelerate the algorithm's convergence speed, which reduces the oscillation and accelerates through the flat region by considering the historical gradient in the parameter update to improve the learning efficiency. Three sets of parameters were selected in the preselection range for the experiment, and as shown in Figure 8, although the change is not obvious, the algorithm's convergence is more stable when momentum (m) = 0.9 and weight decay (wd) = 0.0005.

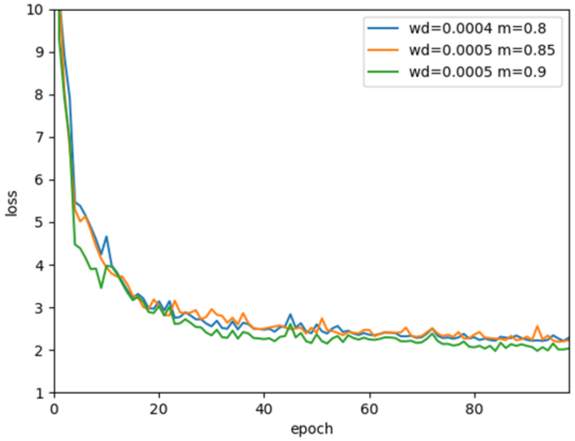


Figure 8. Parameter comparison experiment

3.2 Dataset Preprocessing

The dataset images used in this paper are provided by an automobile company in Hebei Province and contain 421 images. Random flipping, cropping, and tonal transformation of the dataset images enhanced the diversity of the dataset image features. Finally, the dataset was expanded to 2 065 images, with a total of about 90 000 fastener instances. All images were annotated by labeling in Python, including four target categories, namely plastic screws, metal screws, black solder joints, and white solder joints. The dataset images are divided into training set, validation set, and test set at a ratio of 7 : 2 : 1 for training, and the dataset images and annotated images are shown in Figure 9.

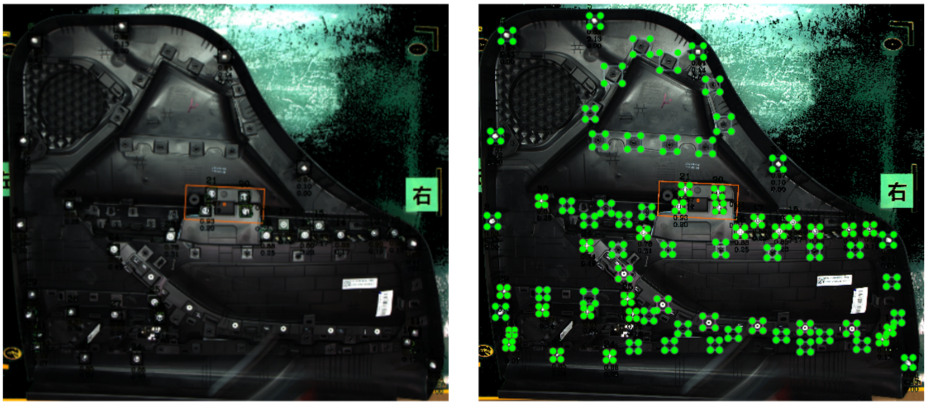


Figure 9. Dataset image (left) and labeled image (right)

3.3 Evaluation Indicators

In this experiment, precision, recall, average detection accuracy mAP50 and mAP50 : 95, and params were used as evaluation indicators to evaluate the model's performance. The specific formula is shown in Equations (15), (16), (17), and (18).

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{15}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{16}$$

$$\mathrm{AP} = \int_0^1 \mathrm{P}(r)\,\mathrm{d}r, \tag{17}$$

$$\mathrm{mAP} = \frac{1}{n}\sum_{i=0}^{n} \mathrm{AP}_{(i)}. \tag{18}$$

Among them, TP is the number of positive samples predicted as correct, FP is the number of negative samples predicted as correct, FN is the number of positive samples predicted as wrong, AP is the lower area of the P-R curve of a certain category, n is the number of detection categories, and $n = 4$, mAP is the average detection accuracy of all categories in this experiment. mAP50 represents the average detection accuracy at the IoU threshold of 50 %, and mAP50:95 is the average detection accuracy of the IoU threshold from 50 % to 95 % (in steps of 5 %).

3.4 Comparative Experiments

3.4.1 Ablation Experiments

This paper introduces the SPD module, attention mechanism, and loss function. Five groups of N0 to N4 ablation experiments were carried out to verify the effectiveness of the improvement points. "✓" indicates the improvement points applied in this group of experiments, "–" indicates the improvement points that were not applied in this group of experiments, and the ablation experiments are shown in Table 2.

Number	SPD	SK	MPDIoU	mAP50	mAP50:95	Params
N0	–	–	–	97.1 %	74.0 %	3 006 428
N1	✓	–	–	98.3 %	76.5 %	3 021 788
N2	–	✓	–	98.1 %	76.1 %	3 073 532
N3	–	–	✓	98.1 %	76.2 %	3 006 428
N4	✓	✓	✓	98.8 %	78.4 %	3 088 908

Table 2. Ablation experiments

N0 is the experimental result of YOLOv8n. The mAP50 value is 97.1 %, the mAP50:95 value is 74.0 %, and the parameter quantity is 3 006 428, which is used as the algorithm's evaluation standard. N1 is the experimental result after the introduction of the SPD module. The mAP50 is increased by 1.2 %, the mAP50:95 is increased by 2.5 %, and the computation and parameter amount is slightly increased. N2 is the experimental result after the introduction of the SK module. The mAP50 is increased by 1 %, mAP50:95 is increased by 2.1 %, and the amount of computation and parameters is slightly increased. N3 is the experimental result of replacing the CIoU loss function with the MPDIoU loss function, with a 1 % increase in mAP50 and a 2.2 % increase in mAP50:95. N4 is the experimental result of the YOLO-DTO algorithm proposed in this paper, which increases the mAP50 by 1.7 % and mAP50:95 by 4.4 % by increasing a small amount of computation and parameters.

3.4.2 Visual Analytics

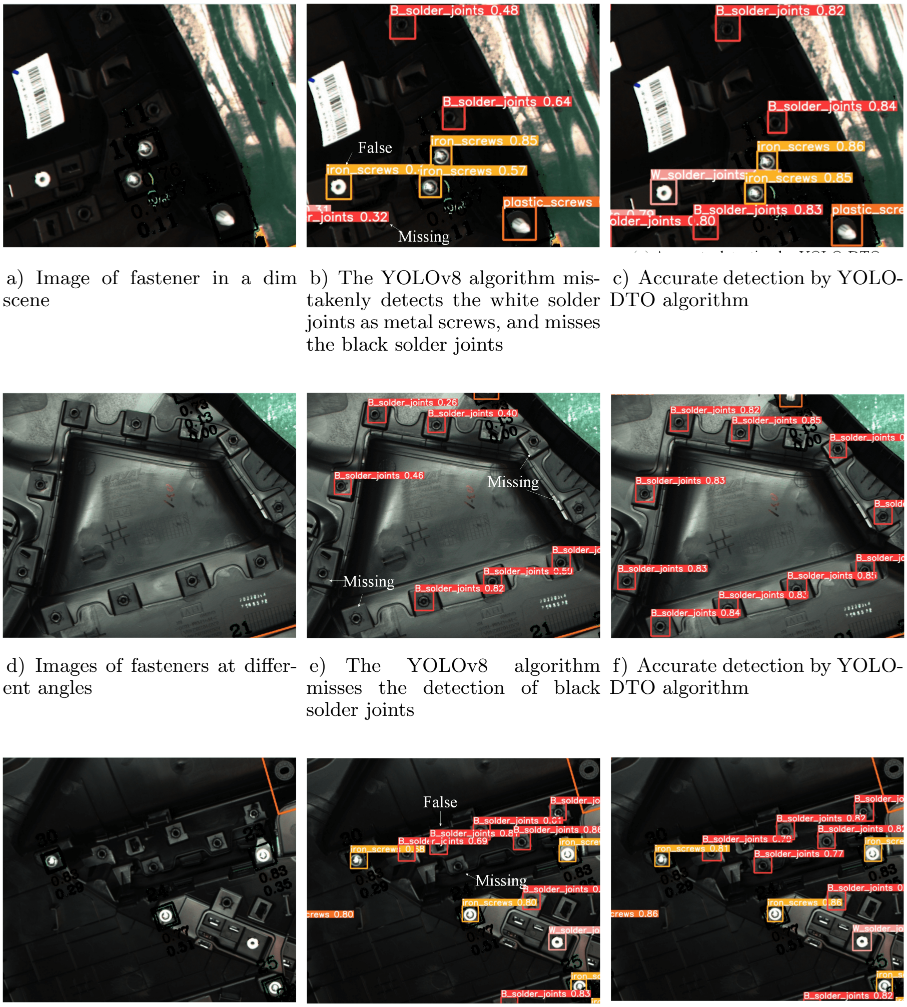
Figures 10 a), 10 d), and 10 g) are fastener images under different lighting and different angles, Figures 10 b), 10 e), and 10 h) are the results of fastener detection by YOLOv8 algorithm, and Figures 10 c), 10 f) and 10 i) are the results of fastener detection by YOLO-DTO algorithm under the same scenario.

The fasteners in the image are classified using different identification frames, and a target confidence score is annotated on each identification frame. As can be seen from the detection result diagram in Figure 10 b), the YOLOv8 algorithm misses the detection of black solder joints and mistakenly detects white solder joints as metal screws for fasteners in dim conditions. As can be seen from the detection result in Figure 10 e), the YOLOv8 algorithm will have more missed detections when the fastener angle in the image is different. When the fasteners are dense, and the surrounding local images are complex, there will be slight false detections and missed detections of YOLOv8 in Figure 10 h). In the same scenario, the YOLO-DTO algorithm accurately detects the fastener, and the confidence score is significantly improved, as shown in the detection results in Figures 10 c), 10 f), and 10 i).

To deeply understand the main reasons for the YOLOv8 algorithm's detection error, the neural network heat map visualization tool was used to represent the algorithm's attention distribution to the image area in the prediction process. The performance impact of the algorithm on fastener detection after adjusting the YOLOv8 algorithm at different stages was compared and analyzed.

In Figure 11, the blue area represents the area the algorithm does not notice. In contrast, the green, yellow, and red regions represent the gradual increase in the algorithm's attention to the image area. The red areas expressly point out the areas where the algorithm gives the highest level of attention.

Figure 11 a) shows the YOLOv8 algorithm's attention distribution of fasteners in automotive door panels. The fastener area is less weighted, reflected in the lighter color on the heat map, which is dominated by green. Figure 11 b) shows the reduction of the convolution step size from 2 to 1 in the Backbone and the introduction of the SPD module. This adjustment significantly increases the algorithm's attention



a) Image of fastener in a dim scene

b) The YOLOv8 algorithm mistakenly detects the white solder joints as metal screws, and misses the black solder joints

c) Accurate detection by YOLO-DTO algorithm

d) Images of fasteners at different angles

e) The YOLOv8 algorithm misses the detection of black solder joints

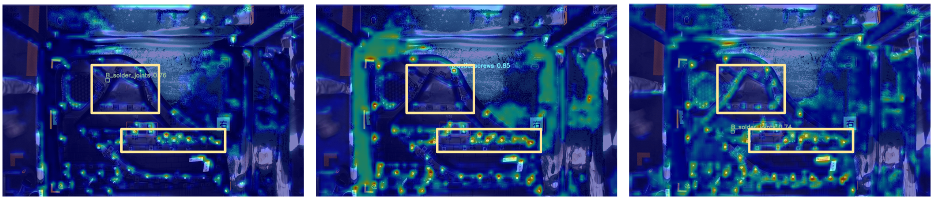
f) Accurate detection by YOLO-DTO algorithm

g) Fastener image in dense scenes

h) The YOLOv8 algorithm mistakenly detects the next parts as black solder joints, and there is a missed detection

i) Accurate detection by YOLO-DTO algorithm

Figure 10. Comparison chart of test results (marked in the false detection and missed detection charts)



a) YOLOv8 b) YOLOv8 + SPD c) YOLO-DTO

Figure 11. The visualization of different methods

weight to the fastener, and the color of the fastener area on the heat map changes to red, indicating an increase in weight. However, we observed that some of the fasteners in the upper area of the car door panels did not receive enough attention. Figure 11 c) shows the integration of the SK module in the network's neck structure. This improvement significantly improves the algorithm's overall focus on automotive door panels, reducing attention to extraneous context and enhancing the ability to identify fasteners and their surrounding areas.

Based on the above analysis, we find that in the fastener detection scenario, the error detection generated by the original algorithm is mainly reflected in two aspects. Firstly, the use of stridden convolution in the early stage of the algorithm will cause the feature information of the fastener to be lost prematurely, resulting in the fastener information becoming weaker or completely disappearing in the feature map, resulting in missed detection. Secondly, due to the imbalance in the distribution of target information and background information of fasteners, the algorithm cannot obtain enough information to distinguish the category of fasteners, resulting in false detection. With the introduction of SPD and SK modules, the algorithm can capture the feature information of fasteners more comprehensively, which helps to improve the accuracy of discriminating different fastener categories.

3.4.3 Comparative Experiment with Loss Function

The MPDIoU loss function selected in this work is compared with the loss functions of CIoU, Focal CIoU, GIoU [26], EIoU [27], SIoU [28], Focal SIoU, and WIoU [29]. The experimental results are shown in Figure 12.

The CIoU loss function in the original algorithm was used as the evaluation criterion. Compared with the CIoU loss function, the mAP50 of Focal CIoU and GIoU did not increase significantly, and the mAP50 : 90 decreased slightly, indicating that the detection accuracy of Focal CIoU and GIoU decreased with the increase of the IoU threshold. Compared with the CIoU loss function, there is no significant improvement in mAP50 and mAP50 : 90 in EIoU. Compared with the CIoU loss function, the mAP50 : 90 of WIoU and SIoU increased by about one percentage point, but there was no significant improvement in mAP50. The mAP50 and mAP50 : 90 of Focal SIoU significantly improve compared with the CIoU loss

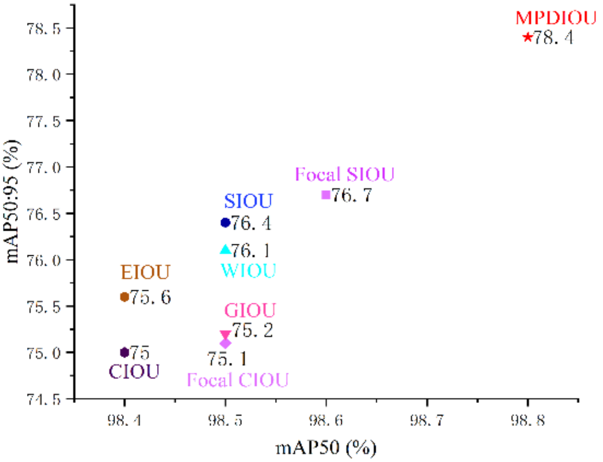


Figure 12. Comparison of different loss functions

function, but compared with the loss function in the algorithm in this paper, the MPDIoU has a higher average detection accuracy than the Focal SIoU loss function. Therefore, the effectiveness of the MPDIoU loss function in predicting the bounding box regression localization in the fastener detection task is verified.

3.4.4 Comparative Experiments with Different Algorithms

To verify the effectiveness of the YOLO-DTO algorithm, the same dataset was trained with SSD, Faster R-CNN, YOLOv5, YOLOv6 [30], and RetinaNet [31] algorithms in the same experimental environment, in which Resnet50 [31] was used as the backbone network for feature extraction for SSD and Faster R-CNN. In addition, the main methods used in the current small target detection research, CAM, CBAM, and the latest feature fusion ASPN [32] (Asymptotic Feature Pyramid Network) method were tested, and the experimental results are shown in Table 3.

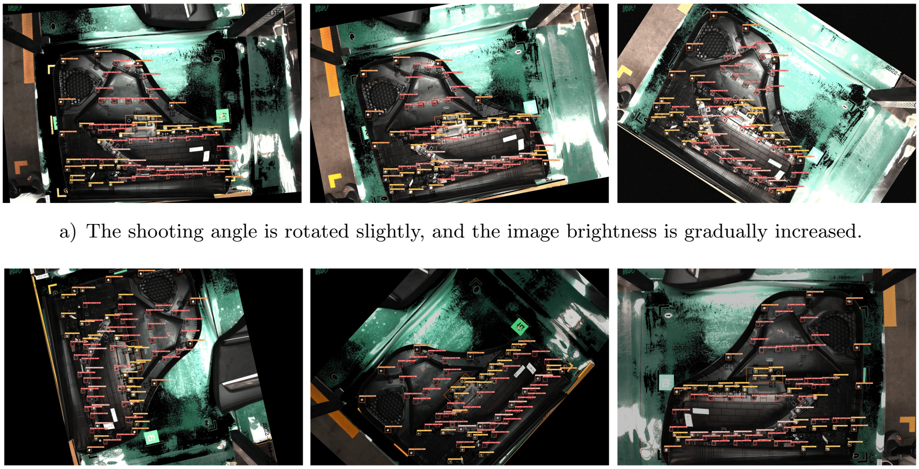
Algorithm	Param/KB	ACC
Template matching	–	89.7 %
Faster R-CNN	5170.95	96.4 %
SSD	3818.62	54.5 %
YOLOv8	375.80	97.1 %
YOLOv8 + CAM	376.02	97.4 %
YOLOv8 + CBAM	376.07	97.7 %
YOLOv8 + AFPN	392.31	98.5 %
YOLO-DTO	377.06	98.8 %

Table 3. Test results

Experimental results show that the proposed algorithm has a small increase in the number of parameters compared with the original algorithm. The average accuracy reaches 98.8 %, which is 9.1 %, 2.4 %, 44.3 %, and 1.7 % compared with the template matching method, Faster R-CNN algorithm, SSD algorithm, and the YOLOv8 algorithm, respectively. Compared with the YOLOv8 algorithm with CAM attention, CBAM attention, and ASPN feature fusion, they are improved by 1.4 %, 1.1 %, and 0.3 %, respectively. The average detection time of the proposed algorithm is 63.8 ms, which is much lower than the commonly used template matching methods and comparable to the average detection time of the original algorithm. These results verify that the YOLO-DTO algorithm has high accuracy and real-time performance in fastener detection tasks.

3.5 Generalization Analysis

The model's generalization ability is a measure of the algorithm's ability to identify samples that are not involved in training. In our work, we improve the generalization ability of the model in two aspects: firstly, the data augmentation of the training dataset is carried out to simulate more data samples that may appear in different forms in practical applications. Then, we use dynamic adjustment of the learning rate and add regularization weight attenuation coefficients to the optimizer to reduce overfitting during model training, thereby enhancing the model's generalization ability in different image data.



a) The shooting angle is rotated slightly, and the image brightness is gradually increased.

b) The shooting angle is rotated drastically, and the image brightness is gradually reduced.

Figure 13. Detection results of YOLO-DTO under different conditions

In the fastener detection scenario, the angle at which workers place the car door panels on the production line may differ each time, and the lighting in the production hall will be slightly different at different times. By rotating the angle of the automobile door panel image to simulate different placement angles and changing the HSV color space of the image to simulate the illumination of different periods, the generalization ability of the YOLO-DTO algorithm for different fastener detection scenarios was tested. We selected several representative prediction images in the experimental results, as shown in Figure 13.

In Figure 13 a), the door panel rotates slightly and gradually increases the brightness. In Figure 13 b), the door panel rotates significantly and decreases brightness. For the above different input conditions and environmental changes, the YOLO-DTO algorithm can accurately predict the position target of the fasteners in the automotive door panel, which verifies the stability and reliability of the algorithm in practical applications.

4 CONCLUSIONS

To improve the accuracy and real-time detection of automotive door panel fasteners, we propose an improved algorithm YOLO-DTO, which fuses the SPD module in the backbone network, introduces the SK module in the neck network, and replaces the bounding box regression positioning ability of the MPDIoU loss function optimization algorithm. Experiments show that the average detection accuracy of the proposed algorithm is 98.8 %, which is 9.1 % higher than that of the template matching method and 1.7 % higher than that of the original YOLOv8 algorithm, and the average detection time is 63.8 ms. Moreover, the average detection accuracy of the proposed algorithm on the public remote sensing dataset RSOD [33] is 1.6 % higher than that of the YOLOv8 algorithm. In the future, we will continue to expand the data set for the detection of automotive door panel parts and study the detection algorithm that is more suitable for small targets on the basis of YOLO-DTO to further improve the detection accuracy of different parts. Our research mainly aims to detect parts of automotive door panels. However, it can be extended after the corresponding optimization to a broader range of industrial component detection fields, which has great development prospects.

REFERENCES

[1] Zhu, Y.—Yin, D.—Zou, S.—Wang, H.—Zhou, W.: The Development and Application of Machine Vision in the Automotive Industry. Automobile Applied Technology, Vol. 42, 2017, No. 22, pp. 8–11, doi: 10.16638/j.cnki.1671-7988.2017.22.004 (in Chinese).

[2] Liu, J.: The Research on Algorithm of Vehicle Door Solder Joint Recognition Based on Machine Vision. Master Thesis. South China University of Technology, Guangzhou, China, 2018 (in Chinese).

[3] Pei, Z.: The Research on Welding Components and Solder Joint Recognition Algorithm of Automobile Door Panel. Master Thesis. South China University of Technology, Guangzhou, China, 2019 (in Chinese).

[4] Girshick, R.—Donahue, J.—Darrell, T.—Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[5] Girshick, R.: Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[6] Ren, S.—He, K.—Girshick, R.—Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, 2017, No. 6, pp. 1137–1149, doi: 10.1109/TPAMI.2016.2577031.

[7] Liu, W.—Anguelov, D.—Erhan, D.—Szegedy, C.—Reed, S.—Fu, C. Y.—Berg, A. C.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9905, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[8] Li, S.—Liu, Y.—Wu, S.—Zhang, S.: MDM-YOLO: Research on Object Detection Algorithm Based on Improved YOLOv4 for Marine Organisms. Computing and Informatics, Vol. 42, 2023, No. 1, pp. 210–233, doi: 10.31577/cai_2023_1_210.

[9] Reis, D.—Kupec, J.—Hong, J.—Daoudi, A.: Real-Time Flying Object Detection with YOLOv8. CoRR, 2023, doi: 10.48550/arXiv.2305.09972.

[10] Pan, X.—Jia, N.—Mu, Y.—Gao, X.: Survey of Small Object Detection. Journal of Image and Graphics, Vol. 28, 2023, No. 9, pp. 2587–2615, doi: 10.11834/jig.220455 (in Chinese).

[11] Chen, X.—Wan, M.—Ma, C.—Chen, Q.—Gu, G.: Recognition of Small Targets in Remote Sensing Image Using Multi-Scale Feature Fusion-Based Shot Multi-Box Detector. Optics and Precision Engineering, Vol. 63, 2021, No. 11, pp. 2672–2682 (in Chinese).

[12] Zhou, B.—Khosla, A.—Lapedriza, A.—Oliva, A.—Torralba, A.: Learning Deep Features for Discriminative Localization. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2921–2929, doi: 10.1109/CVPR.2016.319.

[13] Lin, T. Y.—Dollár, P.—Girshick, R.—He, K.—Hariharan, B.—Belongie, S.: Feature Pyramid Networks for Object Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[14] Ji, C.—Zhang, F.—Huang, X.—Song, Z.—Hou, W.—Wang, B.—Chen, G.: STAE-YOLO: Intelligent Detection Algorithm for Risk Management of Construction Machinery Intrusion on Transmission Lines Based on Visual Perception. IET Generation, Transmission & Distribution, Vol. 18, 2024, No. 3, pp. 542–567, doi: 10.1049/gtd2.13093.

[15] Liu, Z.—Lin, Y.—Cao, Y.—Hu, H.—Wei, Y.—Zhang, Z.—Lin, S.—Guo, B.: Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows.

2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 9992–10002, doi: 10.1109/ICCV48922.2021.00986.

[16] Pan, H.—Guan, S.—Zhao, X.—Xue, Y.: Edge Computing-Based Vehicle Detection in Intelligent Transportation Systems. Computing and Informatics, Vol. 42, 2023, No. 6, pp. 1339–1359, doi: 10.31577/cai_2023_6_1339.

[17] Pan, R.—Lin, T.—Li, C.—Hu, B.: Research on Multi Size Automobile Rim Weld Detection and Positioning System Based on Depth Learning. Optics and Precision Engineering, Vol. 65, 2023, No. 8, pp. 1174–1187 (in Chinese).

[18] Woo, S.—Park, J.—Lee, J. Y.—Kweon, I. S.: CBAM: Convolutional Block Attention Module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11211, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[19] Liu, S.—Huang, D.—Wang, Y.: Learning Spatial Fusion for Single-Shot Object Detection. CoRR, 2019, doi: 10.48550/arXiv.1911.09516.

[20] Sunkara, R.—Luo, T.: No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. In: Amini, M. R., Canu, S., Fischer, A., Guns, T., Kralj Novak, P., Tsoumakas, G. (Eds.): Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2022). Springer, Cham, Lecture Notes in Computer Science, Vol. 13715, 2023, pp. 443–459, doi: 10.1007/978-3-031-26409-2_27.

[21] Li, Y.—Hou, Q.—Zheng, Z.—Cheng, M. M.—Yang, J.—Li, X.: Large Selective Kernel Network for Remote Sensing Object Detection. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 16748–16759, doi: 10.1109/ICCV51070.2023.01540.

[22] Zheng, Z.—Wang, P.—Ren, D.—Liu, W.—Ye, R.—Hu, Q.—Zuo, W.: Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. IEEE Transactions on Cybernetics, Vol. 52, 2022, No. 8, pp. 8574–8586, doi: 10.1109/TCYB.2021.3095305.

[23] Ma, S.—Xu, Y.: MPDIoU: A Loss for Efficient and Accurate Bounding Box Regression. CoRR, 2023, doi: 10.48550/arXiv.2307.07662.

[24] Howard, A.—Sandler, M.—Chen, B.—Wang, W.—Chen, L. C.—Tan, M.—Chu, G.—Vasudevan, V.—Zhu, Y.—Pang, R.—Adam, H.—Le, Q.: Searching for MobileNetV3. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[25] Zou, Z.—Chen, K.—Shi, Z.—Guo, Y.—Ye, J.: Object Detection in 20 Years: A Survey. Proceedings of the IEEE, Vol. 111, 2023, No. 3, pp. 257–276, doi: 10.1109/JPROC.2023.3238524.

[26] Rezatofighi, H.—Tsoi, N.—Gwak, J.—Sadeghian, A.—Reid, I.—Savarese, S.: Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 658–666, doi: 10.1109/CVPR.2019.00075.

[27] Zhang, Y. F.—Ren, W.—Zhang, Z.—Jia, Z.—Wang, L.—Tan, T.: Focal and Efficient IOU Loss for Accurate Bounding Box Regression. Neurocomputing, Vol. 506, 2021, pp. 146–157, doi: 10.1016/j.neucom.2022.07.042.

[28] Gevorgyan, Z.: SIoU Loss: More Powerful Learning for Bounding Box Regression. CoRR, 2022, doi: 10.48550/arxiv.2205.12740.

[29] Tong, Z.—Chen, Y.—Xu, Z.—Yu, R.: Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. CoRR, 2023, doi: 10.48550/arXiv.2301.10051.

[30] Li, C.—Li, L.—Jiang, H.—Weng, K.—Geng, Y. et al.: YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. CoRR, 2022, doi: 10.48550/arXiv.2209.02976.

[31] Chen, Y.—Li, Y.—Ma, Z.—Wang, H.—Zhang, L.: Method for Wind Direction Self-Graph Recognition Based on Residual Network. Computer Engineering and Design, Vol. 42, 2021, No. 8, pp. 2373–2380, doi: 10.16208/j.issn1000-7024.2021.08.037 (in Chinese).

[32] Yang, G.—Lei, J.—Tian, H.—Feng, Z.—Liang, R.: Asymptotic Feature Pyramid Network for Labeling Pixels and Regions. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 34, 2024, No. 9, pp. 7820–7829, doi: 10.1109/TCSVT.2024.3376773.

[33] Li, K.—Wan, G.—Cheng, G.—Meng, L.—Han, J.: Object Detection in Optical Remote Sensing Images: A Survey and a New Benchmark. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 159, 2020, pp. 296–307, doi: 10.1016/j.isprsjprs.2019.11.023.

Xiaohui Wang is Associate Professor at the North China Electric Power University, Ph.D., he currently teaches at the School of Computer Science of the North China Electric Power University. He has been actively researching and applying intelligent monitoring of power systems.

Yunshuo Jia is currently a Master's student in computer science and technology at the School of Control and Engineering, North China Electric Power University, with research interests in deep learning and software engineering.

Fengjuan Guo is Lecturer at the North China Electric Power University, Master's degree. She has been engaged in computer vision related scientific research and education for many years, and is currently working at the North China Electric Power University and the Hebei Provincial Key Laboratory of Energy and Power Knowledge Computing.