Instrumentation
China Instrument and Control Society

OPEN ACCESS

INSTRUMENTATION, Vol. 12, No. 2, June 2025 https://doi.org/10.15878/j.instr.202500274

Article

Intelligent Factory Vehicle Detection Algorithm Based on Improved YOLOv8

Qiannian Miao, Tianhu Wang*, Rong Wang

School of Electrical and Information Engineering, Jiangsu University of Technology, Changzhou City, Jiangsu Province, People's Republic of China

* Corresponding author email: tianhu2003@jsut.edu.cn

Abstract: Aiming at the problem that the existing algorithms for vehicle detection in smart factories are difficult to detect partial occlusion of vehicles, vulnerable to background interference, lack of global vision, and excessive suppression of real targets, which ultimately cause accuracy degradation. At the same time, to facilitate the subsequent positioning of vehicles in the factory, this paper proposes an improved YOLOv8 algorithm. Firstly, the RFCAConv module is combined to improve the original YOLOv8 backbone. Pay attention to the different features in the receptive field, and give priority to the spatial features of the receptive field to capture more vehicle feature information and solve the problem that the vehicle is partially occluded and difficult to detect. Secondly, the SFE module is added to the neck of v8, which improves the saliency of the target in the reasoning process and reduces the influence of background interference on vehicle detection. Finally, the head of the RT-DETR algorithm is used to replace the head in the original YOLOv8 algorithm, which avoids the excessive suppression of the real target while combining the context information. The experimental results show that compared with the original YOLOv8 algorithm, the detection accuracy of the improved YOLOv8 algorithm is improved by 4.6% on the self-made smart factory data set, and the detection speed also meets the real-time requirements of smart factory vehicle detection and subsequent vehicle positioning.

Keywords: smart factory; vehicle detection; improved YOLOv8; vehicle positioning

Citation: Qiannian Miao, Tianhu Wang, Rong Wang. "Intelligent Factory Vehicle Detection Algorithm Based on Improved YOLOv8. " Instrumentation 12, no.2 (June 2025). https://doi.org/10.15878/j.instr.202500274

1 Introduction

In the era of Industry 4.0, the intelligent transformation of factories has gradually become a trend. Many factories use the integration of advanced science and technology to promote intelligent transformation, and the intelligent management of vehicles in factories is also one of the major manifestations. On the one hand, for the operation and production, the real-time status and position of the vehicle can be obtained by detecting the operation vehicle, which can provide an accurate basis for the production scheduling of the smart factory, effectively mobilize the operation vehicle, optimize the operation efficiency of the production line, and improve the production efficiency. When unloading, the accurate detection and positioning of the vehicle can provide accurate location information for the automatic forklift or unloading vehicle, which is convenient for unloading[1-4]. On the other hand, the safety of the smart factory is very important. Through real-time detection and other technologies, real-time detection of foreign vehicles entering the factory can be realized, and then these vehicles can be further tracked and positioned to detect anomalies in time. Locate the place where the accident occurred and take corresponding safety measures. This helps to prevent accidents and accidents, improve the safety and safety management level within the smart factory, and ensure the safety of employees and equipment. However, at present, many factories have a large number of vehicle target loss, low detection accuracy and other problems in the real-time detection of vehicles, which will lead to large subsequent vehicle positioning errors. Therefore, the research on vehicle detection for smart factories is of great significance[5-7].

Early object detection algorithms based on deep learning are mostly two-stage. As the name implies, candidate regions are first generated through specialized modules and then classified and predicted. In 1989, Professor Yann Le Cun published the first convolutional neural network LeNet-5, which is a milestone in the history of CNN(Convolutional Neural Network). The processing speed of this kind of target detection algorithm is relatively slow. Therefore, in recent years, the single-stage target detection algorithm has developed rapidly. It does not need to be a candidate region but directly performs category prediction and positioning, and the speed is fast. One of the typical algorithms is a real-time target detection method YOLO(You Only Look Once) proposed by Joseph Redmon et al. in 2015. Over the years, it has been widely used in various target detection scenarios after generations of versions[8]. In 2023, the Ultralytics team developed a new YOLO version YOLOv8(You Only Look Once version 8), which has higher detection accuracy and robustness under the premise of meeting real-time requirements.

With the continuous breakthrough of deep learning technology and the rise of target detection algorithms based on deep learning, researchers are no longer limited to the original traditional visual target detection algorithms, which also provide a new solution for vehicle detection and detection for smart factories. Vehicle detection based on a deep learning algorithm extracts vehicle image features through CNN and Generative Pre-trained Transformer to achieve final vehicle detection[9]. Many scholars at home and abroad have carried out a series of related research. Song et al.proposed a detection model called MEB-YOLO. This model uses the BiFPN (Bidirectional Feature Pyramid Network) module to improve the neck, which makes up for the shortcomings of the existing model in feature fusion. The improved model can be used in the vehicle detection process to achieve more complex feature fusion. Many researchers are committed to the research of lightweight models for the problem of difficult deployment and slow detection speed caused by the complex structure and large amount of calculation of the existing vehicle detection model. Guo Yuyang et al. proposed an improved GS-YOLO model, which greatly reduces the number of parameters and calculation of the model under the premise of ensuring detection accuracy[10-13]. In the past two years, Zhao Qing et al. proposed a lightweight YOLOv5(You Only Look Once version 5) algorithm. In order to reduce the number of downsampling in the backbone network, the algorithm added multiple pyramids and multi-scale attention. Finally, it not only realized the lightweight of the model, but also reduced the impact of complex environments on vehicle detection and improved the detection accuracy of the model for vehicles. Aiming at the problem that the original v8 uses NMS(Non--Maximum Suppression) to process the candidate box is difficult, Zhang Lifeng et al. proposed an improved YOLOv8 vehicle target detection algorithm, using SoftNMS(Soft Non Maximum Supression) to replace the original NMS, which enhances the model's ability to detect targets[14].

Although the above research has improved the accuracy and model size, it needs further improvement to be applied to vehicle detection in complex factory environments and ultimately achieve accurate vehicle positioning. Therefore, this paper proposes a smart factory vehicle detection model based on improved YOLOv8. The main work of this paper is as follows:

(1)The RFCAConv(Receptive Field Collaborative Attention Convolution Operation) is introduced to improve the C2f module and reconstruct the trunk. For vehicles that are partially occluded and fail to be detected, the importance of different features in the receptive field slider is emphasized and the spatial features of the receptive field are given priority, so as to capture more useful vehicle feature information.

(2)The SEF (Scattering Feature Enhancement module) is added to the neck of YOLOv8 to highlight the detected vehicle target, thus reducing the influence of complex background interference on vehicle detection in smart factories.

The head of the original YOLOv8 is replaced by the head of RT-DETR(Real-Time Detection Transformer), which not only combines the context information and emphasizes the global vision, but also solves the problem that the original YOLOv8 will excessively suppress the real target, and improves the detection accuracy of the model for smart factory vehicles.

2 Introduction of the Related Algorithms

2.1 *Introduction of the YOLOv8*

YOLOv8 is a real-time target detection algorithm, which is a new target detection model in the YOLO series. It combines the advantages of the previous generation of YOLO, introduces new modules and technologies, and improves the detection accuracy and speed. In addition, YOLOv8 also supports image segmentation, key point detection and skeleton tracking[15].

The overall structure of YOLOv8 is divided into three parts: backbone, neck and detection head. The first is the backbone, which is mainly used for feature extraction. C2f(CSP Bottleneck with 2 convolutions), as a key part of it, is formed by combining C3(CSP

Bottleneck with 3 convolutions) and ELAN(Efficient Layer Aggregation Networks) in YOLOv5 to improve the network performance of the backbone, while SPPF (Spatial Pyramid Pooling-Fast) is used to capture feature information at various scales. The second is the neck. The role of the neck is to process and fuse the features extracted from the trunk. Through the combination of PAN(Path Aggregation Network) and FPN(Feature Pyramid Network), the features from different levels are fused to better detect targets of different scales. The output head part is responsible for extracting the target information from the feature map output by the neck and generating the final detection result[16]. The overall structure diagram of YOLOv8 is shown in Fig.1.

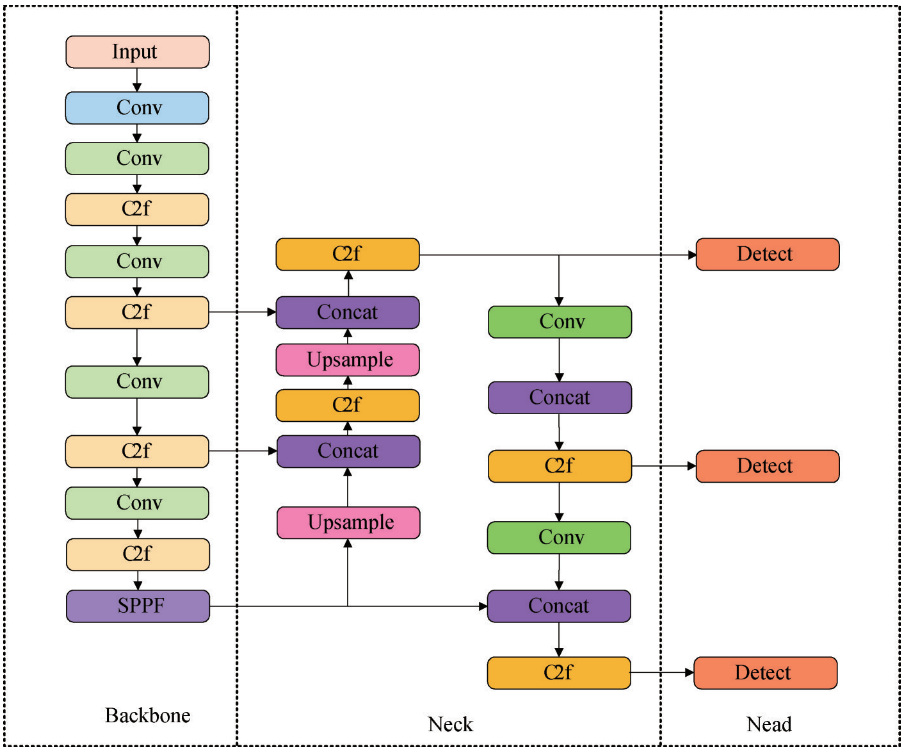


Fig.1 The overall structure diagram of YOLOv8

2.2 Introduction of the RT-DETR

RT-DETR is an end-to-end real-time detection algorithm based on Transformer architecture. Different from the v8 structure, RT-DETR is composed of a trunk, a hybrid encoder and a Transformer decoder with an auxiliary prediction head. As shown in Fig. 2, when the input is extracted through the trunk, the features extracted from the three parts of S3, S4 and S5 are passed to the hybrid encoder. In the encoder, the AIFI(Attention-Based Intra-Scale Feature Interaction) and the CCFF(CNN-Based Cross-Scale Feature Fusion) are used to transform the multi-scale features into a series of image features, and then the minimum uncertainty query is used in the head-to-filter the required encoder features into the decoder. Finally, the Transformer decoder with auxiliary prediction header queries the incoming features as the initial object and iterates to optimize the subsequent object query, and generates categories and prediction boxes[17-19] the attention-based intra-scale feature interaction(AIFI) and the CNN-based cross-scale feature fusion(CCFF) are used to transform the multi-scale features into a series of image features, and then the minimum uncertainty query is used in the head to filter the required encoder features into the decoder. Finally, the Transformer decoder with auxiliary prediction header queries the incoming features as the initial object and iterates to optimize the subsequent object query, and generates categories and prediction boxes[17-19].

3 The Intelligent Factory Vehicle Detection Model Based on Improved YOLOv8

Considering the practical application in the factory environment, this paper uses YOLOv8n with smaller parameters and weights as the improved model. In the actual detection process of smart factory vehicles, YOLOv8 will have some practical problems. First of all, the standard convolution used to extract features in the YOLOv8 backbone is prone to capture insufficient vehicle features and cause detection failure when the factory vehicle is partially occluded. Secondly, when vehicle detection is carried out in the smart factory, the complex background environment of the factory will cause great interference to the detection of v8. Furthermore, YOLOv8 pays too much attention to local information,

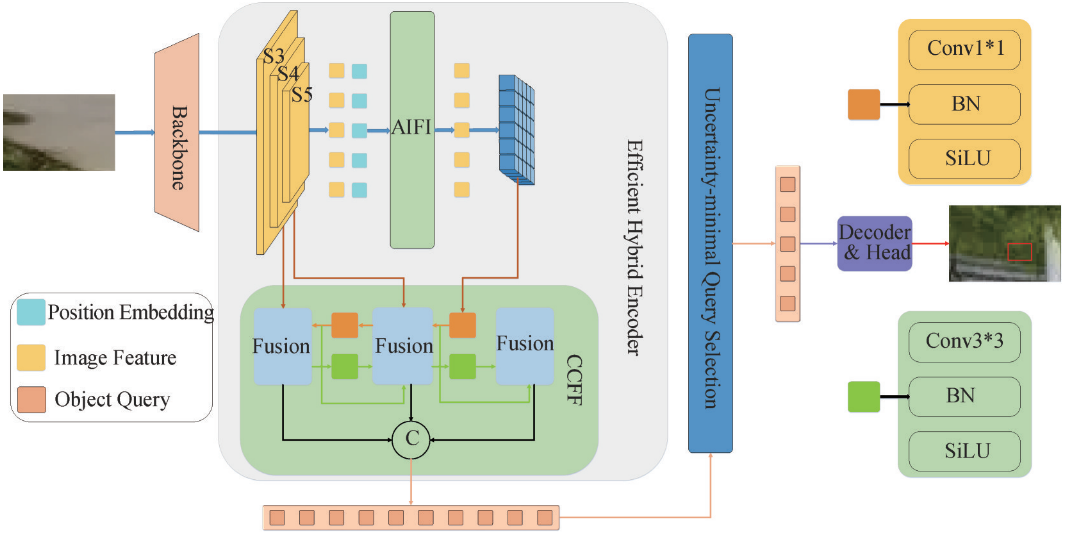


Fig.2 The overall structure diagram of RT-DETR

does not fully consider context information, and lacks global vision. Finally, when more vehicles have overlapping detection frames, the non-maximum suppression NMS of YOLOv8 may cause excessive suppression of real targets, resulting in missed detection and other phenomena. These problems will eventually lead to a decline in vehicle positioning accuracy. In order to solve the above problems, this paper proposes a vehicle detection model based on improved YOLOv8, which is mainly divided into the following three parts: (1) The YOLOv8 backbone is improved by using the receptive field collaborative attention convolution operation RFCAConv; (2) Add a scattering feature enhancement module SFE to the neck of YOLOv8; (3) Using the head of RT-DETR to replace the head of the original YOLOv8 greatly improves the accuracy of vehicle detection in the factory environment[20]. The improved YOLOv8 smart factory vehicle detection model structure is shown in Fig.3.

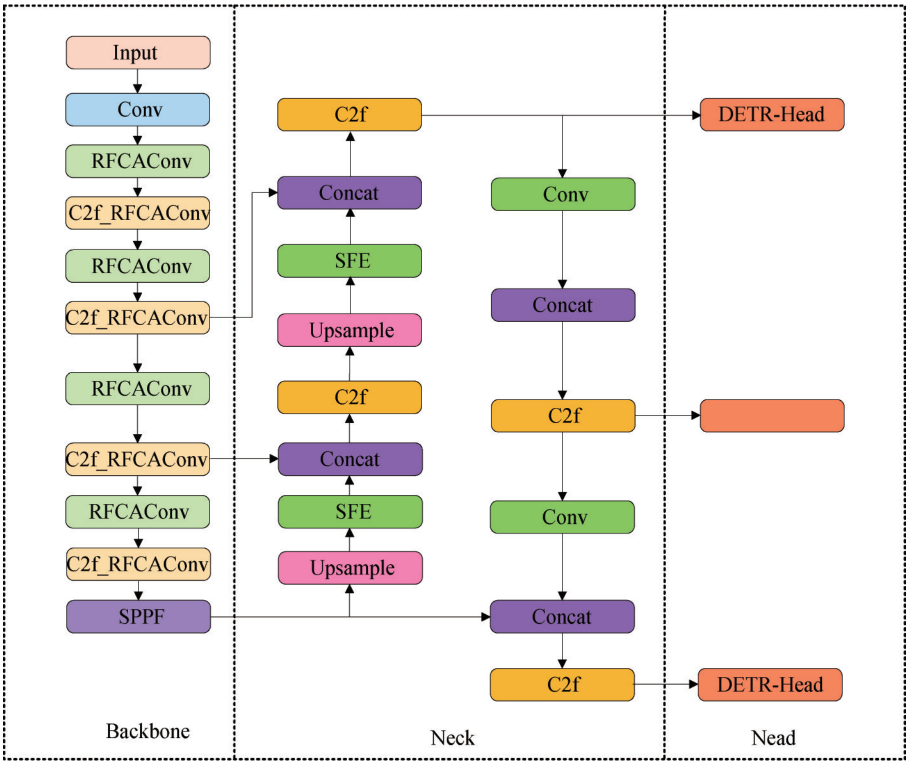


Fig.3 The overall structure diagram of improved YOLOv8

3.1 RFCAConv Module

Since the original YOLOv8 cannot capture enough vehicle feature information and lead to detection failure due to partial occlusion of the vehicle when detecting the vehicle of the smart factory, and finally cannot locate the vehicle, the Receptive-Field Coordinated Attention Convolutional(RFCAConv) is proposed to improve the backbone of the original YOLOv8.

RFA(Receptive-Field Attention) makes up for the limitations of the current spatial attention mechanism and provides a new solution for spatial processing. Since then, a series of related spatial attention mechanisms inspired by RFA have emerged, which has given new vitality to convolutional neural networks. The RFCA (Receptive-Field Coordinated Attention) used in this paper guides the attention of the CA(Coordinate Attention) to the spatial features of the receptive field. It can solve the problem of parameter sharing and remote information modeling in a similar way to self-attention, but it requires much less parameters and computing resources than self-attention. This method not only emphasizes the importance of different features in the receptive field slider, but also gives priority to the spatial features of the receptive field, so that the model can better capture the structure and context information of the input data, so as to obtain more comprehensive and richer intelligent factory vehicle feature information. The feature map composed of the spatial features of the receptive field obtained by RFCA integrates the feature information of each receptive field slider. In other words, the attention feature map is no longer just shared within each receptive field slider. This completely makes up for the shortcomings of the existing collaborative attention mechanism. The receptive field collaborative attention can be used as a plug-and-play lightweight module. The receptive field collaborative attention convolution operation designed by RFCA brings good benefits to convolution, and innovates the standard convolution, thereby improving the performance of the entire convolutional neural network[21]. Fig. 4 shows the overall structure diagram of RFCAConv.

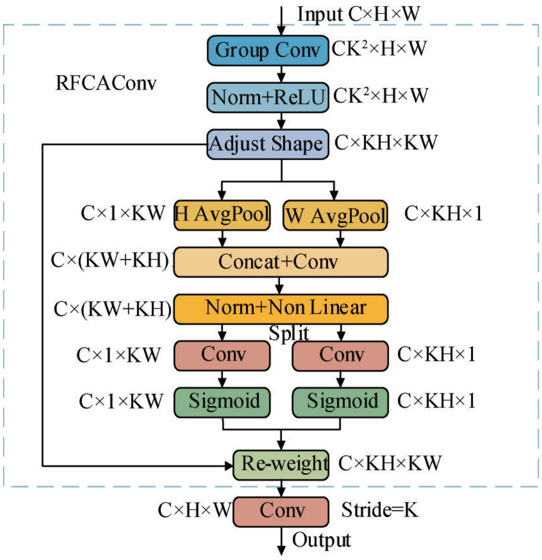


Fig.4 The overall structure diagram of RFCAConv.

First, group convolution is performed on the input, and group convolution is used to extract the spatial features of the receptive field, and the original features are mapped to new features. This method is faster and more efficient than the general expansion method. Then the shape is adjusted, and the height and width of the feature are changed to k times of the original. For RFCAConv, learning attention maps through interactive receptive field feature information can improve network performance. At the same time, in order to reduce the additional computational overhead caused by interacting with each receptive field feature, this paper uses AvgPool (Average Pooling) to aggregate the global information of each receptive field feature from the two dimensions of width and height, and then combine them together. The information is interacted and divided into two parts through a 1×1 convolution operation, and then sigmoid activation is performed on these two parts to generate attention weights, so as to emphasize the importance of each feature in the receptive field feature. Finally, the output result of the sigmoid and the output result after adjusting the shape are reweighted to obtain the feature map[22]. However, the height and width of the feature are k times the original, so a convolution operation with a step of k is used to extract the feature information. The formula of RFCAConv is as follows:

$$X' = \mathrm{Adjust}(\mathrm{ReLU}(\mathrm{Norm}(\mathrm{GroupConv}(X)))) \tag{1}$$

$$\begin{aligned} X_{\mathrm{H}} &= \mathrm{HAvgPool}(X') \\ X_{\mathrm{W}} &= \mathrm{WAvgPool}(X') \end{aligned} \tag{2}$$

$$Z = \mathrm{NonLinear}(\mathrm{Norm}(\mathrm{Conv}(\mathrm{Concat}(X_{\mathrm{H}}, X_{\mathrm{W}})))) \tag{3}$$

$$Z_{\mathrm{H}}, Z_{\mathrm{W}} = \mathrm{Split}(Z) \tag{4}$$

$$\begin{aligned} A_{\mathrm{H}} &= \mathrm{Sigmoid}(\mathrm{Conv}(Z_{\mathrm{H}})) \\ A_{\mathrm{W}} &= \mathrm{Sigmoid}(\mathrm{Conv}(Z_{\mathrm{W}})) \end{aligned} \tag{5}$$

$$Y = \mathrm{Conv}(X' \times A_{\mathrm{H}} + X' \times A_{\mathrm{W}}) \tag{6}$$

3.2 SFE Feature Enhancement Module

Considering the complex background interference that may occur during vehicle detection in the factory environment, this paper adds a Scattering Feature Enhancement(SFE) module to the neck. By capturing texture details and semantic information and effectively integrating and utilizing them, the influence of complex background interference on feature fusion is reduced, and the target of the detected vehicle is highlighted. The overall performance of the model is improved, which is more conducive to subsequent vehicle positioning.

The SFE module in this paper is applied to the high-level semantic feature map. It can be seen from Fig. 5. Specifically, due to the unique anisotropy of the vehicle target and the self-similarity of the environment background, this paper introduces the CPDC (Central Pixel Difference Convolution). By subtracting the weighted environment around the target, the background interference is suppressed and the saliency of the target is enhanced. The calculation process of the central pixel difference convolution CPDC is similar to that of the traditional ordinary convolution, in which the original pixels in the local feature block are covered by the

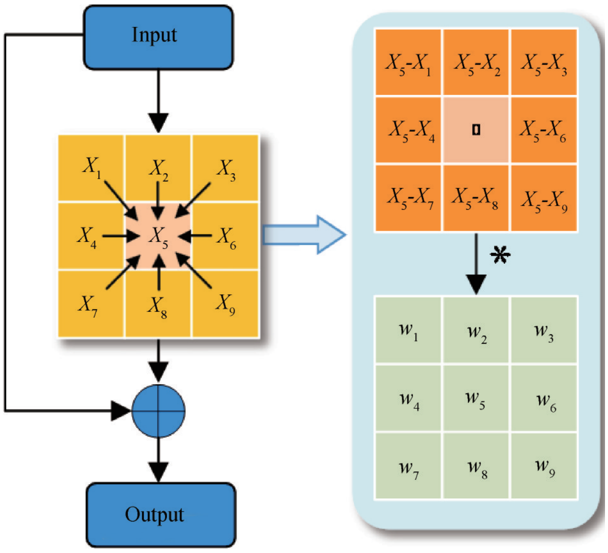


Fig.5 The overall structure diagram of SFE

convolution kernel and replaced by the pixel difference in the convolution operation[23]. The calculation formulas of traditional ordinary convolution and central pixel differential convolution CPDC are as follows:

$$y=f(x,\theta)=\sum_{i=1}^{k\times k} w_i\cdot x_i \tag{7}$$

$$y_{\mathrm{cpdc}}=f(\nabla x,\theta)=\sum_{(x_i,x_i')\in \mathrm{P}} w_i\cdot(x_i-x_i') \tag{8}$$

Among them, x_i is the input of the original pixel, w_i is the size of the convolution kernel of $k\times k$ weights. P is the set of pixel pairs selected from the current local block, $P=(x_1,x_1'), (x_2,x_2'),\cdots, (x_n,x_n')$ is the set of pixel pairs selected from the current local block, and $n\leqslant k\times k$.

Considering that if the SFE module is applied to the shallow feature map, the model may focus too much on capturing texture details and semantic information, resulting in false detection. Therefore, only the SFE module is applied to the high-level semantic feature map in this paper. In addition, in order to enhance the saliency of the related vehicle target, this paper also uses a feature fusion design in the SFE module, as shown in Fig.5, By fusing the original features with the features after clutter suppression, the advantages of clutter suppression and the integrity of the original features can be effectively combined, thereby further improving the signal-to-noise ratio. This fusion helps to enhance the contrast between the target signal and the background clutter, making the target more prominent in the data. At the same time, the fused features focus more on the core features of the target signal, further highlighting the target information.

3.3 DETR-Head Module

Because YOLOv8 has excellent performance in accuracy and speed, it has been applied to various scenarios and has become one of the most popular real-time target detection algorithms. However, on the one hand, it does not fully consider the context information and lacks a global vision, which makes it difficult to optimize the model when performing vehicle detection in different scenarios of the factory, the generalization effect is poor, and the final accuracy is reduced[24]. On the other hand, when more vehicles gather together for detection in the factory, many overlapping detection boxes will be generated. At this time, non-maximum suppression (NMS) is needed for post-processing. This processing will directly filter out the detection boxes below the confidence threshold, and then calculate the IOU (Intersection over Union) of the current highest confidence box with the IOU of all other detection boxes, and remove the detection boxes that exceed the set IOU threshold. In this way, the processing may cause excessive suppression of the real target, and the phenomenon of missed detection may cause a decline in accuracy[10]. In the subsequent case, the vehicle will not be located. Therefore, this paper uses the head of RT-DETR to replace the head of the original YOLOv8

As shown in Fig. 6, the RT-DETR headers include: Uncertainty-Minimal Query Selection, Transformer Decoder, and Prediction Header. Different from the original Transformer decoder, the decoder used in this algorithm decodes N objects in parallel at each decoder layer. The FFN(Prediction Feed-Forward Network) in the prediction header is composed of multiple linear layers and ReLU(Rectified Linear Unit) activation functions. The linear layer is mainly used to map the high-

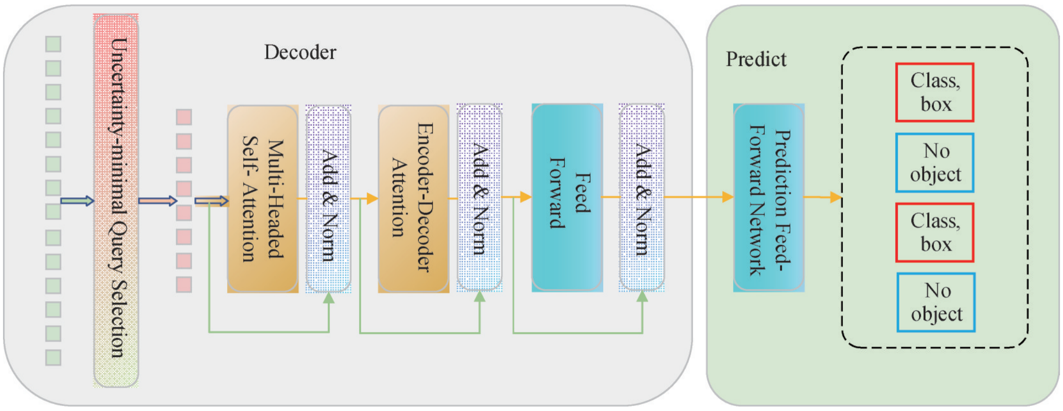

Uncertainty-minimal Query Selection
Decoder
Multi-Headed Self- Attention
Add & Norm
Encoder-Decoder Attention
Add & Norm
Feed Forward
Add & Norm
Predict
Prediction Feed-Forward Network
Class, box
No object
Class, box
No object

Fig.6 The head structure diagram of RT-DET

dimensional embedding vector output by the decoder to the specific parameters of the target detection, that is, the embedding vector is converted into the center coordinates, width, height and category probability distribution of the bounding box, and the softmax function is used to generate the probability of each category[19]. The overall processing flow is as follows : Firstly, after the feature map is processed and fused by the neck, the current query scheme will lead to the optimization of the decoder and affect the detection performance. The minimum uncertainty query is used to filter out the required encoder feature input decoder. The decoder is processed by the Multi-Headed Self-Attention layer and then input into the Encoder-Decoder Attention layer. The pairwise relationship between input embeddings is used to globally reason all objects. At the same time, the whole image is used as the context, and the output embedding is finally obtained. Finally, the output is embedded as the input of the prediction feedforward network and they are independently converted into the coordinates and category labels of the detection box through the feedforward network to obtain the final prediction result[25].

By replacing the head of YOLOv8 with the head of RT-DETR, the model can not only better deal with the relationship between complex scenes and targets through the global context-aware ability of Transformer, improve the generalization ability, and improve the overall performance and ease of use of the model, but also significantly simplify the post-processing steps of the model, avoid over-inhibition of real targets, improve the detection efficiency and accuracy of smart factory vehicles, and facilitate the subsequent tracking and positioning of vehicles in smart factories.

4 Results

4.1 Data Sets and Experimental Platform

By collecting the monitoring data of a factory and collecting it from the Internet, a total of 10,000 vehicle images of the factory environment were collected and manually labeled by Labelme. Finally, they were divided into training sets, verification sets and test sets according to the ratio of 8：1：1, that is, 8000 training sets, 1000 verification sets and 1000 test sets.

The experimental platform is configured as follows : CPU is Intel core i7-12700H, GPU is NVIDIA GeForce RTX 3080Ti, its memory is 12GB, CUDA version is 11.8. In this paper, the training imagesize size is 640×640, the batchsize size is 16, and the total number of iterations is set to 300.

4.2 Evaluation Indicators

In this paper, mAP@0.5, GFLOPS, Parameters and FPS are used as the evaluation indexes of model performance. Among them, GFLOPS reflects the computational complexity and resource requirements of the model, Parameters reflects the storage requirements and complexity of the model, and FPS reflects the inference speed and real-time performance of the model, mAP@0.5 represents the average precision of all pictures in each category when the IoU threshold is 0.5, and the average precision is the area under the Precision and Recall curves. The relevant calculation formula is as follows:

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

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

$$mAP=\frac{1}{N}\sum_{i=1}^{N}AP_i \tag{11}$$

4.3 Ablation Experiment

In order to verify the influence of the improvement of each module on the performance of the whole model, this paper makes a comparative analysis by setting the ablation experiment. The experimental results are shown in Table 1.

Table 1 Results of ablation experiments.

Improvement	mAP@0.5	GFLOPS	Parameters	FPS
YOLOv8n	73.4%	8.1	3007013	256.8
YOLOv8n-RFCAConv	74.7%	8.6	3079253	192.7
YOLOv8n-SEF	74.4%	8.2	3040867	243.8
YOLOv8n-DETR	76.8%	11.7	6098828	138.8
YOLOv8n-RFCAConv-SEF	75.2%	8.7	3752546	176.8
YOLOv8n-RFCAConv-DETR	77.3%	12.2	6171068	116.5
YOLOv8n-SEF-DETR	76.6%	11.7	6020147	128.7
YOLOv8n-RFCAConv-SEF-DETR	78.0%	12.6	6793660	93.4

From the results of ablation experiments, it can be seen that the accuracy mAP has been improved by using RFCAConv to improve the trunk, using SFE to improve the neck, and using RT-DETR head to replace the original head, but the amount of calculation and parameters has increased, resulting in a decrease in FPS, among which the result of using RT-DETR head to replace the original head is the most significant. On this basis, the improvement is combined in pairs, and the accuracy mAP is improved more than any improvement alone, but the amount of calculation and the number of parameters are also increased compared with any improvement alone, and the FPS decreases slightly. Finally, the combination of all improved models is tested, and the accuracy mAP is greatly improved. However, due to the increase of calculation and parameters, FPS decreases greatly. In general, although the number of parameters and the amount of calculation of GFLOPS have increased, resulting in a decrease in FPS, the FPS at this time can still meet the needs of real-time vehicle detection in smart factories. At the same time, these improvements ultimately improve the detection accuracy mAP of the model, which provides convenience for vehicle positioning in subsequent smart factories.

4.4 Comparative Experiments of Detection Performance of Different Models

In order to further verify the detection performance of the intelligent factory vehicle detection model based on improved YOLOv8 for intelligent factory vehicles, this paper compares the improved vehicle detection model with mainstream single-stage target detection algorithms such as SSD, YOLOv3-tiny, YOLOv5n, YOLOv7-tiny, YOLOv8n, YOLOv11n and YOLOv12n.

SSD is an efficient target detection algorithm, which is processed by convolutional neural network (CNN) and can locate and classify targets in a single forward propagation. It can detect objects of different sizes at the same time by predicting on feature maps of multiple scales. The advantage of SSD is that it is fast and suitable for real-time target detection tasks, but it is relatively weak in the detection of small objects and has low accuracy. YOLOv3-tiny is a lightweight version of YOLOv3, which greatly improves the detection speed by reducing the number of network layers and parameters. It is suitable for devices with low requirements for computing resources, such as embedded devices and mobile terminals. Although it performs well in speed, it is not as good as the YOLOv3 standard version due to the sacrifice of accuracy, especially in complex scenes or small object detection. YOLOv5n is the smallest model in the YOLOv5 series and is designed for resource constrained environments. Compared with the standard version of YOLOv5, YOLOv5n significantly improves the inference speed by reducing the number of network layers and parameters. Its advantage lies in its very fast processing capability, which is suitable for edge computing devices and real-time detection tasks, but its performance is slightly insufficient when dealing with complex scenes or tasks with high precision requirements. YOLOv7-tiny is a lightweight version of YOLOv7, which improves the inference speed by reducing the network size, and is suitable for real-time detection tasks that require low latency and high efficiency. Although it sacrifices some accuracy, compared with other lightweight models such as YOLOv3-tiny, YOLOv7-tiny achieves a better balance between accuracy and speed, especially for object detection tasks in complex scenes. YOLOv11n is a lightweight version of the YOLOv11 series, which further optimizes the network structure to improve reasoning speed and efficiency. It is suitable for low-resource environments and embedded devices, especially in real-time detection tasks. Although it sacrifices some accuracy, its ultra-fast reasoning speed and compact model structure make it perform well in low-computing-power devices. YOLOv12n is the smallest version of the YOLOv12 series, which adopts a more streamlined network architecture and further reduces the demand for computing resources. It provides a very efficient solution for real-time target detection and embedded devices, with ultra-low latency and efficient inference speed. Although there is a gap in accuracy with larger versions, it is still an ideal choice for resource-constrained devices. The final results are shown in Table 2.

Table 2 Comparison results of detection performance of different models.

Model	mAP@0.5	GFLOPS	Parameters	FPS
SSD	57.6%	86.3	25569872	110.8
YOLOv3-tiny	60.3%	13.0	8683736	121.6
YOLOv5n	69.9%	4.3	1773388	273.6
YOLOv7-tiny	73.9%	13.1	6023832	143.3
YOLOv8n	73.4%	8.1	3007013	256.8
YOLOv11n	73.2%	6.3	2583517	248.6
YOLOv12n	74.2%	6.3	2558093	168.7
Ours	78.0%	12.3	6793660	93.4

It can be seen from the above table that the improved vehicle detection model proposed in this paper is higher than most of the YOLO series algorithms in computational complexity and parameter quantity, but still lower than some algorithms. On FPS, although the improved vehicle model proposed in this paper is lower than all the algorithms compared, it still meets the requirements of vehicle detection in smart factories ; in terms of accuracy, the improved vehicle detection model proposed in this paper has achieved different degrees of improvement compared with the current mainstream single-stage target detection algorithm.

4.5 Comparison of Heat Map Before and After Improvement

As shown in Figure 7, the heat map of the output of the detection model before and after the improvement. Since the improved heat map is more widely distributed, it can better reflect the various parts of the truck and help detect the details of the body, including the front, rear and side. In contrast, the focus of the pre-improved model is mainly on the upper part of the truck, the cab and part of the body, which may lead to neglect of other areas (such as trailers, wheels, etc.). The heat map of the improved model also has some thermal distribution in the background area, but compared with the first map, the heat is more balanced, which helps to reduce the interference of the background on the detection results and ensure that more focus falls on the truck itself, not just the local area.

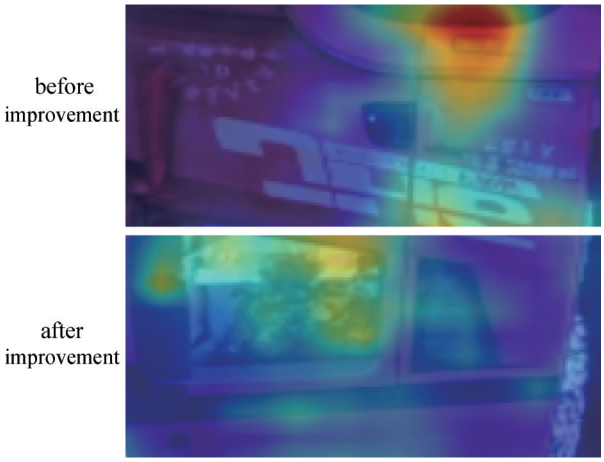
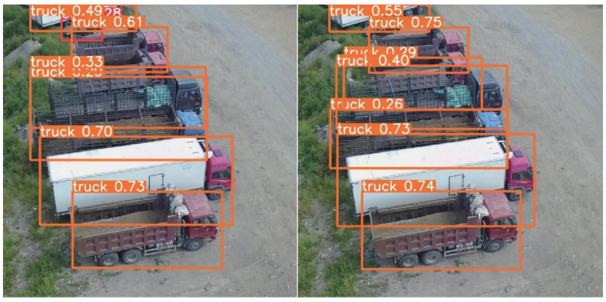


Fig.7 Heat map before and after improvement

4.6 *Comparison of Actual Detection Results*

As shown in Fig.8, the improved model in this paper is compared with the actual detection effect of the original YOLOv8n detection model. It can be seen that when the vehicle is partially occluded, the model in this paper can detect, while the original YOLOv8n detection model has missed detection and false detection. In general, the detection performance of the proposed model for vehicle targets is significantly better than that of the original model, and it is more suitable for subsequent vehicle positioning.



Original YOLOv8n Ours

Fig.8 Comparison of actual detection results

4.7 *Model Performance Bottlenecks and Future Optimization Strategies*

Through the above experiments, it can be concluded that although the proposed method has achieved significant improvement in accuracy, due to the increase of model complexity, the increase of calculation amount and parameter amount leads to the decrease of FPS index. In order to optimize the performance of the model while maintaining the accuracy improvement, we plan to use pruning as a potential solution. The basic idea of pruning technology is to reduce the computational complexity and memory usage of the model by deleting unnecessary parameters or connections in the neural network, thereby improving the inference speed and reducing the delay. Specifically, we can apply different pruning strategies, such as structured pruning and unstructured pruning.

Structured pruning usually refers to removing the entire convolution kernel, neuron or layer, which can significantly reduce the computational complexity of the model. By removing redundant channels or layers, structured pruning can effectively reduce the amount of calculation, improve the processing speed, and maintain the relatively complete calculation structure of the model, so it is very suitable for the application of hardware accelerators. On the other hand, unstructured pruning achieves the purpose of reducing the amount of computation by removing a single weight or connection. Unstructured pruning can control the complexity of the model more finely, but its efficiency in hardware implementation is low, and more post-processing is needed to ensure the optimization of the model.

In order to further improve the effect of pruning, we can combine different pruning strategies and combine the importance of weights to perform pruning operations. By evaluating the importance of each layer or channel, we can determine whether to prune according to their respective contribution to the output results. Low-contribution connections or channels will be preferentially cut off to ensure that the transmission of important information is not affected.

In addition, the precision of the pruned model can be restored by fine-tuning to ensure that the original detection ability is maintained as much as possible while reducing the amount of calculation. This process helps to optimize the pruned model and solve the problem of accuracy degradation.

Future research directions will focus on further optimizing the combination of pruning strategies, including exploring adaptive pruning methods, and dynamically adjusting the strength and method of pruning according to different smart factory scenarios and hardware platforms. In addition, the joint optimization of quantization technology and pruning technology is also a direction worth exploring, which can further reduce the storage requirements and inference delay of the model. Through these methods, we expect to achieve a better balance between accuracy, computational complexity and inference speed, so that the model can achieve the best performance in various smart factory scenarios.

5 Conclusion

In order to realize real-time detection of vehicles in smart factories, this paper proposes a smart factory vehicle detection algorithm based on improved YOLOv8. The algorithm solves the problems existing in the traditional YOLOv8 algorithm in the smart factory vehicle detection task, which greatly facilitates the subsequent positioning of vehicles and realizes the intelligent management of vehicles in smart factories. The algorithm first combines the receptive field collaborative attention convolution operation RFCAConv to improve the YOLOv8 backbone, which solves the problem of insufficient effective features captured by the vehicle due to partial occlusion in the smart factory, and finally fails to detect. Then, the scattering feature enhancement module SFE is added to the neck to highlight the detected vehicle target by capturing texture details and semantic information, which reduces the influence of background interference on factory vehicle detection. Finally, the head of YOLOv8 is replaced by the head of RT-DETR, which realizes the combination of context information and improves the accuracy of vehicle detection in smart factory. At the same time, it also solves the problem of excessive suppression of real targets when there are many vehicle targets in the smart factory and the overlapping detection boxes need to be processed. Experiments show that the improved YOLOv8 smart factory detection algorithm can meet the real-time requirements of smart factory vehicle detection, and the accuracy mAP is improved by 4.6 % compared with the original YOLOv8n. Compared with other mainstream detection models, it has also been improved to varying degrees, laying a foundation for subsequent vehicle positioning in smart factories.

Author Contribution:

Miao Qiannian: As the first author, I am mainly responsible for creating the dataset, improving the model, and training it. Wang Tianhu: Provided guidance for research directions and methods. Wang Rong: Assisted in model training and experimental result analysis.

Foundation Information:

This work was funded by Changzhou Science and Technology Project(No. CZ20230025) and Postgraduate Research&Practice Innovation Program of Jiangsu Province(No.XSJCX23_36).

Data Availability:

The authors declare that the main data supporting the findings of this study are available within the paper and its Supplementary Information files.

Conflicts of Interest:

The authors declare no competing interests.

Dates:

Received 21 February 2025; Accepted 06 May 2025; Published online 30 June 2025

References

[1] Jia Z Y, Ren G Q, Li D W, et al. (**2017**). Vehicle recognition and methods based on laser radar depth information and visual HOG features[J]. *Journal of Academy of Armored Forces Engineering* , 31(6), 88-95.

[2] Fu Y Y, Wang J P, Zhang T S, et al. (**2023**). Real-time vehicle detection and research based on YOLOv5s[J]. *Journal of Anhui University of Engineering* , 38(1), 26-32.

[3] Wu Q Y, Zhao Z P, Wang L F, et al. (**2024**). Vehicle detection and dataset in complex road monitoring scenarios[J]. *Applied Science and Technology* , 51(1), 10-18.

[4] V K K, Sonali D, Priyadarsan P. (**2024**). Vehicle detection in varied weather conditions using enhanced deep YOLO with complex wavelet[J]. *Eng. Res. Express* , 6(2), 025224.

[5] Zhao Y, Wang T H, Miao Q N, et al. (**2024**). Research on indoor and outdoor positioning switching algorithm based on improved PSO-BP[J]. *Meas. Sci. Technol* , 35(8), 086313.

[6] Li Y, Jiang Z, Cai Y, et al. (**2024**). Map-based localization for intelligent vehicles based on fusion of multiple visual features in underground parking[J]. *Engineering Research Express* , 6 (2), 025220.

[7] Zhou H, Yang J, Deng S, et al. (**2024**). VTIL: A multi-layer indoor location algorithm for RSSI images based on vision transformer[J]. *Engineering Research Express* , 6(1), 015069.

[8] Xu L, Huang L B, Bai J. (**2019**). Vehicle detection in traffic scenarios based on Faster RCNN and YOLO[J]. *Journal of Jiamusi University(Natural Science Edition)* , 37(2), 232-235.

[9] Yang X L, Duan M, Yu H N, et al. (**2021**). Research on real-time vehicle detection based on YOLO algorithm[J]. *Instruments and analysis monitoring* , (1), 7-10.

[10] Li Y, Wang J, Huang J, et al. (**2022**). Research on deep learning automatic vehicle recognition algorithm based on RES-YOLO model[J]. *Sensors* , 22(10), 3783.

[11] Jiang C W, Jin L Z. (**2023**). Vehicle-YOLO-a vehicle detection model based on aerial images[J]. *Microcomputer application* , 39(9), 134-137.

[12] Feng J, Fu D D, Liu Q, et al. (**2023**). Research on aerial vehicle detection technology based on YOLOv8[J]. *Computer Science and Application* , 13(12), 2399-2407.

[13] Gong C, Sun Y, Zou C, et al. (**2024**). Real-time visual SLAM based YOLO-Fastest for dynamic scenes[J]. *Measurement Science and Technology* , 35(5), 056305.

[14] Zhang L F, Tian Y. (**2024**). Improved YOLOv8 multi-scale lightweight vehicle target detection algorithm[J]. *Computer Engineering and Application* , 60(3), 129-137.

[15] Wang C M, Du Y C. (**2023**). Vehicle target detection method in complex traffic environment based on YOLO algorithm[J].

Transportation and transportation , 39(2), 20-24.

[16] Zhao N, Wang K, Yang J, et al. (**2024**). CMCA-YOLO: A Study on a Real-Time Object Detection Model for Parking Lot Surveillance Imagery[J]. *Electronics* , 13(8), 1557.

[17] Zhang L L, Hu X X, Hu KZ. (**2024**). Research on vehicle detection of transmission line engineering based on improved DETR model[J]. *Software engineering* , 27(4), 49-53.

[18] Carion N, Massa F, Synnaeve G, et al. (**2020**). End-to-end object detection with transformers[C]. European conference on computer vision. Cham: Springer International Publishing, 213-229.

[19] Zhao Y, Lv W, Xu S, et al. (**2024**). Detrs beat yolos on real-time object detection[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16965-16974.

[20] Wu M, Yang J, Zhang W, et al. (**2022**). Attention feature fusion network for small traffic sign detection[J]. *Engineering Research Express* , 4(3), 035047.

[21] Zhang X, Liu C, Yang D, et al. (**2023**). Rfaconv: Innovating spatital attention and standard convolutional operation[J]. *ArXiv preprint arXiv* , 23(4), 03198.

[22] Guo Y Y, Hu W C, Dai S, et al. (**2022**). Lightweight vehicle detection model for roadside traffic monitoring scene[J]. *Computer Engineering and Application* , 58(6), 192-199.

[23] Zhou J, Xiao C, Peng B, et al. (**2024**). DiffDet4SAR: Diffusion-based Aircraft Target Detection Network for SAR Images[J]. *IEEE Geoscience and Remote Sensing Letters* , (21), 1-5.

[24] Zhai X, Huang M, Wei H. (**2023**). Chip detection algorithm based on lightweight E-YOLOv5 convolutional neural network[J]. *Engineering Research Express* , 5(1), 015083.

[25] SP K M P. (**2024**). DETR-SPP: a fine-tuned vehicle detection with transformer [J]. *Multimedia Tools and Applications* , 83 (9), 25573-25594.