ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

Research on Lightweight Adaptive Object Tracking Algorithm based on Improved YOLOv11n and DeepSORT

Mingzhou Liu[1,2], Sanpeng Deng[1,2,*], Yuming Qi[1,2], and Xiumin Shi[1,2]

[1] Tianjin University of Technology and Education, Tianjin 300222, China

[2] Tianjin Key Laboratory of Intelligent Robot Technology and Application, Tianjin 300222, China

Abstract

To address the issues of complex model structure, poor real-time performance, and susceptibility to environmental interference in two-stage object tracking algorithms, we propose a lightweight adaptive object tracking algorithm based on an improved YOLOv11n and DeepSORT. The YOLOv11 algorithm is enhanced by incorporating the RepNCSPELAN4-low module to reduce model parameters; the C3k2 module is used to replace the RepC3 module in the CCFM network structure to improve feature expression and further reduce both the parameter and computational loads. Additionally, the SENet attention mechanism is introduced to enhance the network's ability to extract global contextual features. For the tracking part, the DeepSORT algorithm is combined with the ByteTrack framework to improve trajectory matching accuracy; adaptive Kalman filtering is applied to optimize the motion model; and the CIOU matching mechanism is introduced to enhance the stability and accuracy of target association. Experimental results show that the improved algorithm, compared to the original YOLOv11, increases FPS by 26.1%, reduces computational load by 9.4%, decreases the number of parameters by 35.3%, and lowers the model size by 32.7%. Furthermore, mAP50 and mAP95 reach 86.8% and 62.5%, respectively. The enhanced object tracking algorithm achieves a 1.9% improvement in MOTA, a 5.5% increase in HOTA, and a 5.3% rise in IDF1, with an FPS of 27, meeting the requirements of practical applications.

Keywords

Pedestrian Detection and Tracking; Lightweight; Adaptive Kalman Filtering; YOLOv11; DeepSORT.

1. Introduction

Object tracking is an important research direction in the field of computer vision, widely applied in scenarios such as autonomous driving, video surveillance, and pedestrian monitoring. With the rapid development of deep learning technologies, deep learning-based object tracking methods have gradually become mainstream. Object tracking algorithms can be divided into two categories based on whether the detection and tracking phases are independent[1]: DBT (Detection-Based Tracking) and JDT (Joint Detection and Tracking). The DBT algorithm[2] divides detection and tracking into two independent stages. In thctor marks the object, and in the second stage, a tracker utilizes the detection results to extract features for object association and matching. The JDT algorithm[3], on the other hand, optimizes both detection and tracking stages simultaneously through an end-to-end joint optimization network to improve overall performance. Given that object detection algorithms already offer high accuracy and speed, meeting the needs of practical applications, this paper selects the DBT algorithm as the main research focus.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

Wu Jiawen et al.[4] based their work on YOLOv8, incorporating a coordinate attention mechanism into the backbone network, replacing the SPPF module with a SimSPPF module, using GSConv to replace standard convolutions, and optimizing the neck network with Slim Neck to improve detection accuracy and reduce computational cost. Huang Kaiwen et al.[5] introduced depthwise separable convolutions in YOLOv4-Tiny to reduce computation, added detection branches, and used multi-scale feature fusion to reduce small object missed detections, while enhancing feature extraction capability through an improved GC attention mechanism. In DeepSort tracking, uniform acceleration Kalman filtering is used, and a shallow classification network is employed to reconstruct the appearance model. Liu Zhaojin et al.[6] utilized an ADown convolution module to reduce algorithm complexity, introduced the EMA attention mechanism in the C2PSA module to preserve channel information, and employed a dynamic object detection head (Dynamic) to improve pedestrian detection accuracy. Sheng Wenshun et al.[7] combined YOLOv8 with DeepSORT, introducing the OSNet feature extraction network, adaptive forgetting Kalman filtering algorithm, and replacing the IOU association matching mechanism with CIOU. However, these methods generally require high computational resources, making it difficult to balance real-time performance and efficiency, and they are vulnerable to background interference and object occlusion. As a result, lightweight adaptive object tracking algorithms have become a research hotspot. Their goal is to reduce computation and storage requirements while dynamically adjusting according to environmental changes to enhance robustness in complex scenarios.

To address these issues, this paper proposes a lightweight adaptive object tracking algorithm based on an improved YOLOv11 detector combined with DeepSort. The algorithm optimizes tracking accuracy and real-time performance while reducing the number of parameters and computational complexity. Experimental results show that the proposed algorithm performs excellently on the MOT17 dataset, demonstrating high robustness and real-time performance, and meeting the demands of practical applications.

2. Improved YOLOv11 Algorithm

To address the issues of large parameter size and slow inference speed in the YOLOv11 algorithm[8] during object detection, improvements have been made to the YOLOv11 network structure. Figure 1 shows the schematic diagram of the improved network structure.

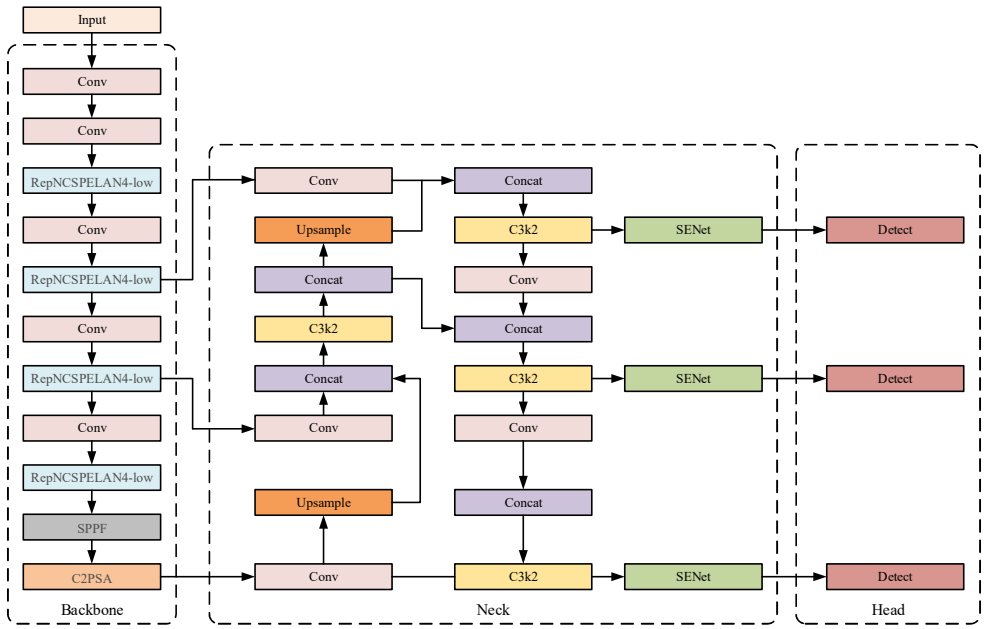

Figure 1. Improved YOLOv11 network structure

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

First, the RepNCSPELAN-low module is introduced to replace the C3k2 module in the YOLOv11 backbone network to reduce the model's parameter size. Then, the CCFM_C3k2 module is used to replace the original neck network, further reducing the model's parameters and computational complexity. Finally, the SENet adaptive channel-wise attention mechanism is applied to the output part of the neck network to mitigate the accuracy loss caused by the lightweight model.

2.1 RepNCSPELAN4-low Module

The Generalized Efficient Layer Aggregation Network (GELAN) is a lightweight feature fusion module in YOLOv9. It combines the CSPNet (Cross Stage Partial Network) and ELAN (Efficient Layer Aggregation Network) architectures, allowing the network to adapt to different application requirements and computational resources. The framework of GELAN is shown in Figure 2.

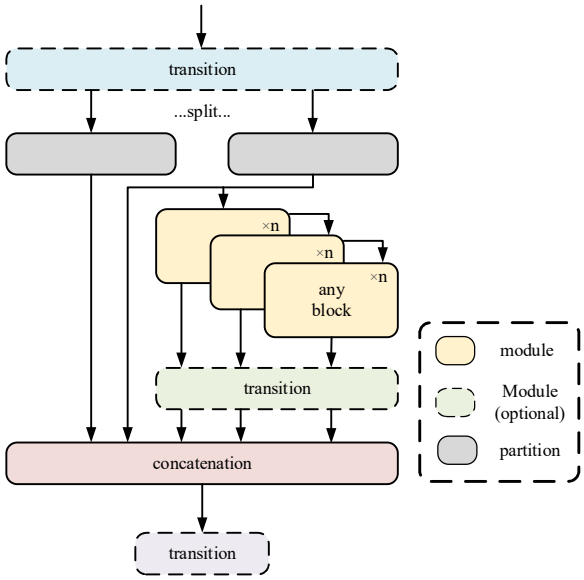

Figure 2. GELAN framework diagram

RepNCSPELAN4 (Rep-Net with CSP and ELAN) is a feature fusion module based on the GELAN architecture[9]. This module divides the input data into two parts and processes each part separately using multiple RepNCSP modules. The output results are then merged and passed to the final convolutional layer. By combining efficient feature extraction with an attention mechanism, it enhances the model's feature representation ability, allowing the network to better capture and emphasize key information related to object detection, thus improving the model's object localization and recognition performance[10].

By increasing the depth of the RepNCSP module, the accuracy of the RepNCSPELAN4 module can be improved, but this also increases the model's parameters and computational complexity. To address this issue, an optimized and more lightweight version of the RepNCSPELAN4 module, called RepNCSPELAN4-low, is designed. Its network structure is shown in Figure 3. By reducing the channel count in the concatenation operation and the RepNCSP module to half of the original, the scale of the weight matrix in convolution operations is reduced, thereby decreasing the number of parameters. Additionally, only one RepNCSP module is retained in RepNCSPELAN4-low, which preserves sufficient expressive power while effectively avoiding unnecessary computational redundancy.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

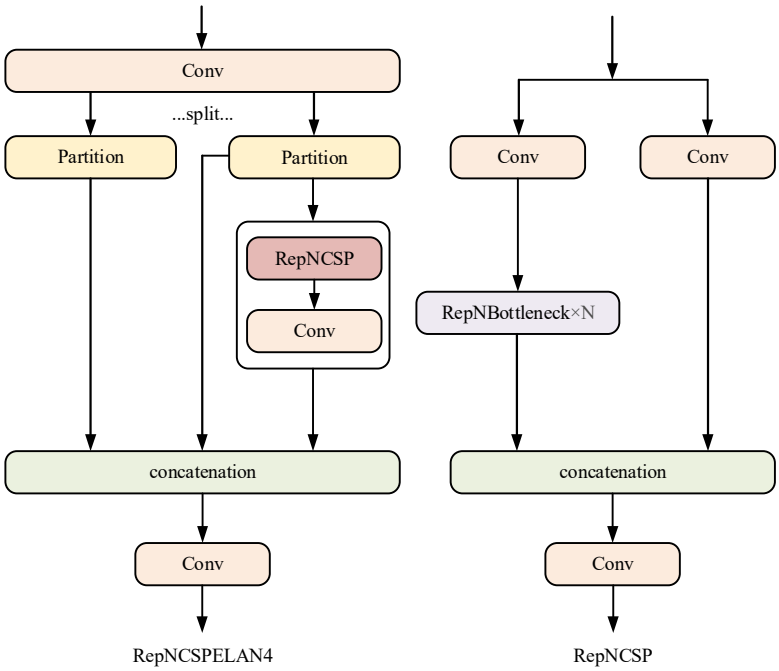

Figure 3. RepNCSPELAN4-low network structure

2.2 CCFM_C3k2 Module

The CCFMC3k2 module is an improvement based on the CCFM (Cross-Scale Feature Fusion Module). It replaces the RepC3 module in the CCFM network structure[11] with the C3k2 module from YOLOv11 to enhance the feature adaptability of the neck network, improve feature representation capabilities, and reduce the model's parameters and computational complexity. Figure 4 shows the network structure of the CCFMC3k2 module.

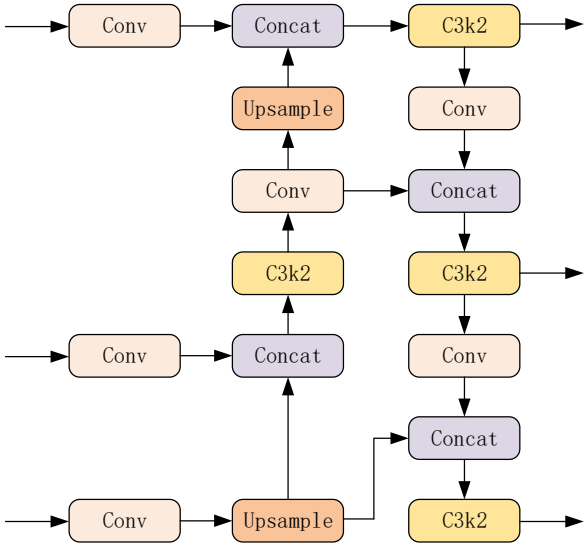

Figure 4. CCFM_C3k2 module network structure diagram

In the YOLOv11 network, three different scales (S1, S2, S3) of features extracted by the backbone network are input into the CCFM module for fusion. These multi-scale features are then converted into image feature sequences and fed into other modules. The structure is shown in Figure 5. The CCFM module enhances the feature map's adaptability to changes in target size by fusing features from different scales, effectively integrating detailed features and contextual information. This improves the model's accuracy in detecting objects of various sizes[12].

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

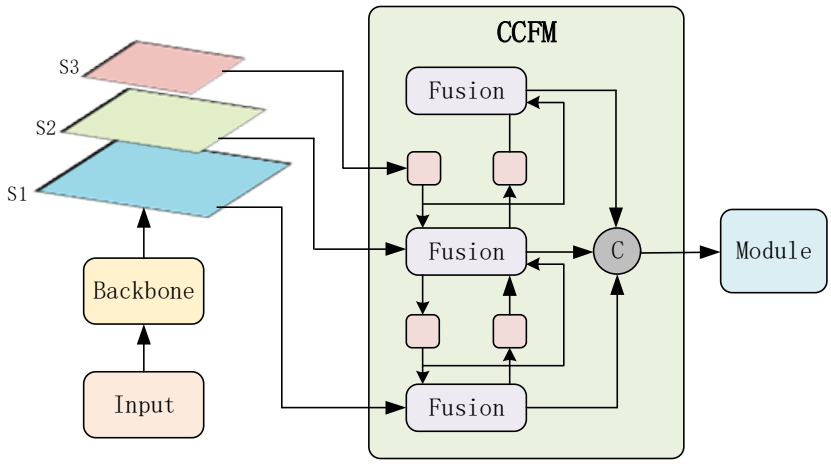

Figure 5. CCFM module

The CCFM is composed of multiple sequential Fusion modules, which are responsible for fusing features from adjacent scales to form new feature representations. This module includes two 1x1 convolutional layers and N RepC3 modules. The fusion of feature maps from different layers or scales is performed via concatenation along the dimension, preserving the information from the original feature maps. The convolutional layers adjust the number of feature channels to ensure that features from different scales have the same dimensionality before fusion. The RepC3 module is used for multi-scale feature extraction and adaptively adjusts the spatial distribution of features, thereby improving the efficiency of feature fusion.In this study, the RepC3 module in the Fusion module is replaced with the C3k2 module from the YOLOv11 network, and the network structure is shown in Figure 6. Tests have shown that this method still effectively integrates features from different scales while significantly reducing the model's computational load and parameter count.

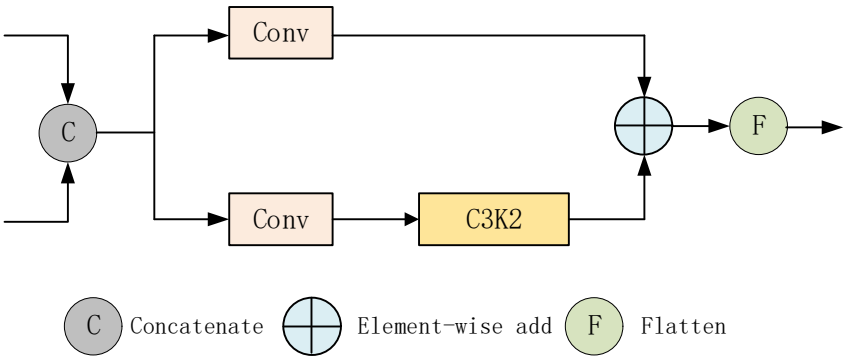

Figure 6. Fusion module

2.3 SENet Attention Mechanism

SENet is a channel-wise attention mechanism[13] that primarily focuses on the relationships between different channels of an input feature map. By adaptively recalibrating the feature responses of each channel, it significantly enhances the network's representational capability while adding minimal computational cost. The structure of the SENet module is shown in Figure 7. To mitigate the accuracy loss in lightweight models, the SENet attention mechanism is introduced at the output part of the neck network. SENet strengthens the representation of effective features and suppresses irrelevant ones through adaptive channel attention, thereby improving the overall performance and detection accuracy of the model.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

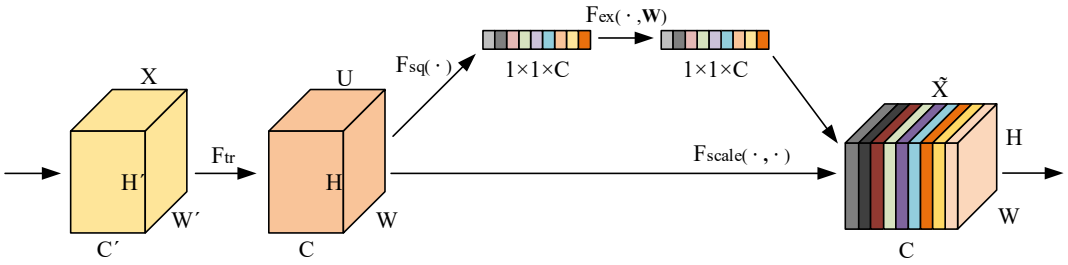

Figure 7. SENet module

First, the input feature map X is transformed through the Ftr operation to produce the feature map U.

$$U=F_{tr}(X)$$

Next, the input feature map U undergoes global average pooling F_{sq}, which compresses the spatial information of each channel into a single value. This results in a feature vector Z of size 1×1×c, which contains the global information of each channel.

$$Z_c=F_{sq}(U_c)=\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}U_c(i,j)$$

In the formula, Zc is the spatial information of the compressed c-th channel of the feature map U, Uc is the c-th channel of the feature map U, H and W are the width and height of the feature map respectively, and $U_c(i,j)$ is the value of the c-th channel of the feature map U at position (i,j).

Then, two fully connected layers F_{ex} are used to perform nonlinear transformation on the channel global information vector Z to generate the weight coefficients for each channel, further determining the importance of each channel. The first fully connected layer compresses C channels to C/r channels through dimensionality reduction to reduce the computational load, and uses the ReLU activation function. The second fully connected layer restores the number of channels to C through dimensionality expansion, and finally obtains the channel attention weight S with a dimension of 1×1×C through the Sigmoid activation function.

$$S=F_{ex}(Z,W)=\sigma(g(Z,W))=\sigma(W_2\delta(W_1Z))$$

In the formula, W_1 and W_2 are the weight matrices of the fully connected layer, δ is the ReLU activation function, σ is the Sigmoid activation function, and r is the compression ratio.

Finally, the Scale operation is applied to multiply the original feature map U by the channel weights, resulting in a recalibrated feature map $\tilde{X}c$.

$$\tilde{X}_c=F_{scale}(U_c,S_c)=S_cU_c$$

In the formula, S_c represents the attention weight of the c-th channel in the feature map U.

3. Improved DeepSORT Algorithm

In view of the limitations of Kalman filtering in predicting motion trajectories in complex scenes, as well as the limitations of using IoU (Intersection over Union) matching in matching strategies, this paper draws on the ByteTrack[14] algorithm framework, combines adaptive Kalman filtering[15] and

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

CIoU (Complete-Intersection over Union) matching[16], and proposes a multi-target tracking algorithm based on DeepSort. The improved DeepSort algorithm process is shown in Figure 8.

Detection section: Firstly, the improved YOLOv11 object detector is used to detect the current frame, obtaining the detection boxes and confidence scores of the tracked targets. The detection boxes are divided into high score detection boxes and low score detection boxes based on the confidence threshold, and different processing methods are adopted for the two types of detection boxes to improve the accuracy of trajectory matching. After the detection section is completed, execute the tracking section.

Tracking part: Firstly, the position of all tracking boxes in the previous frame in the current frame is predicted through adaptive Kalman filtering, and cascaded matching is performed with high score detection boxes to obtain matched trajectories, unmatched trajectories, and unmatched detection boxes. Then, the unmatched trajectories are associated with the low score detection boxes through CIoU matching, thereby achieving matching of the remaining target trajectories. For unmatched detection boxes, initialize them as new trajectories and start tracking in the next frame. For trajectories that have not yet been matched, if they are in an uncertain state or if there are frames greater than the time threshold in a certain state, they will be directly deleted to save computing resources. If the trajectory is less than the time threshold, continue tracking in the next frame. Finally, the matched trajectories after the second matching are updated through adaptive Kalman filtering to further improve tracking accuracy.

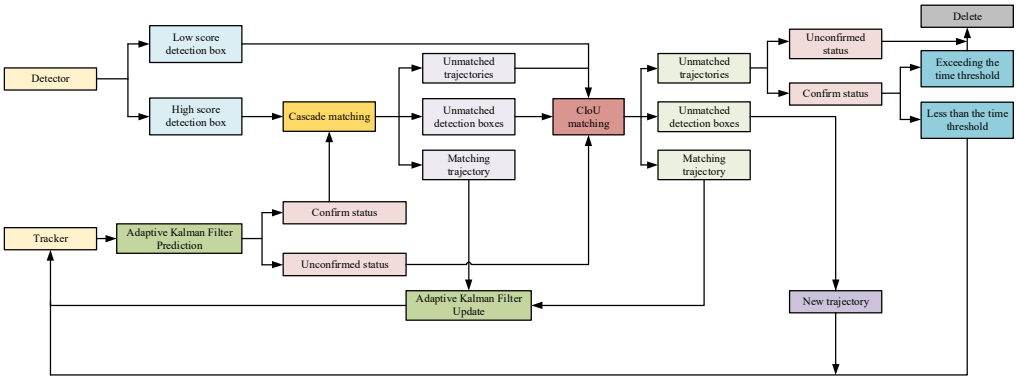

Figure 8. The improved DeepSort algorithm process

3.1 Adaptive Kalman Filtering

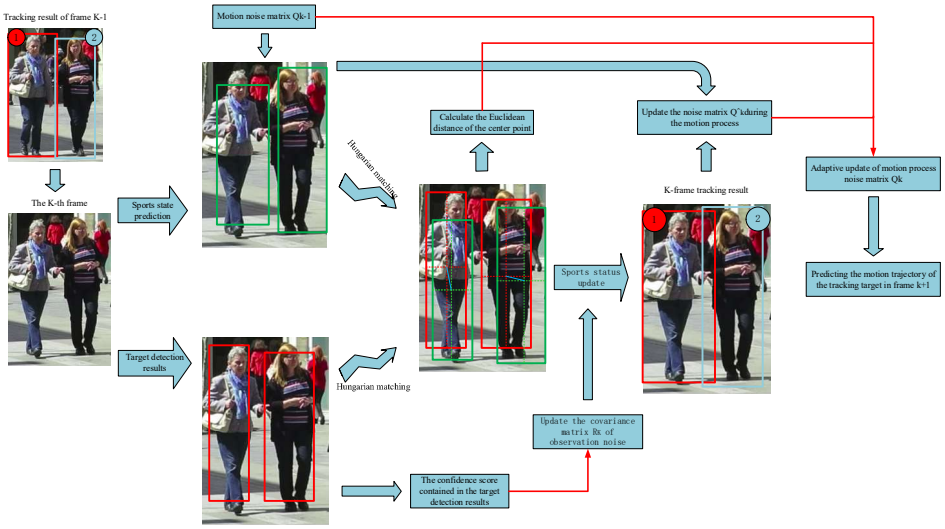

Figure 9. Adaptive kalman filter process.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

The Kalman filter estimates the system state by combining predicted values and actual measured values, and is widely used for state estimation and prediction of dynamic systems. However, it performs poorly in handling state estimation of nonlinear motion and does not fully utilize visual information. To address this issue, this paper proposes an adaptive Kalman filter. This method effectively addresses the tracking problem of irregular moving targets by utilizing the implicit information provided by the target detector to adaptively adjust the noise matrix. The adaptive Kalman filter process is shown in Figure 9, and the process is shown in Table 1.

Table 1. Process of adaptive kalman filter algorithm

Adapt to the Kalman Filter Algorithm Process
Input: The k-1st frame's target motion state X_{k-1}, state transition matrix F, k-1st frame's target motion process covariance matrix P_{k-1}, k-1st frame's target motion process noise covariance matrix, detection box confidence score C_k ($0 \le C_k \le 1$), initial observation noise covariance matrix R_0, observation matrix H, observation value Z_k in the k-th frame, prediction box center point coordinates (X_k, Y_k), detection box center point coordinates (X_z, Y_z), input image resolution W * H.
Output: k-th frame target motion state X_{k-1}, k-th frame target motion process covariance matrix P_k, k-th frame target motion process noise covariance matrix Q_k.
1. Target motion state prediction: $\widehat{X}_{k\|k-1}=FX_{k-1}$
2. Prediction of covariance matrix for target motion process: $P_{k\|1}=FP_{k-1}F^TQ_{k-1}$
3. Adaptively update the covariance matrix of observation noise: $R_k=(1-C_k)R_0$
4. Calculate the Kalman filter gain: $K_k=P_{k\|k-1}+H^TS_k^{-1}(HP_{k\|k-1}H^T+R_k)^{-1}$
5. Update the target motion status: $X_k=\widehat{X}_{k\|k-1}+K_k(Z_k-H\widehat{X}_{k\|k-1})$
6. Update the covariance matrix of the target motion process: $P_k=(1-K_kH)P_{k\|k-1}$
7. Calculate the target motion state error[17]: $\widehat{W}_k=X_k-\widehat{X}_{k\|k-1}$
8. Update the covariance matrix of the error [18]: $\widehat{Q}_k=\widehat{W}_k\widehat{W}_k^T$
9. Calculate the distance between the normalized predicted box and the center point of the detection box: $I=\sqrt{\left(\frac{X_k-X_z}{W}\right)^2+\left(\frac{Y_k-Y_z}{W}\right)^2}$
10. Adaptive update of motion process noise covariance matrix: $Q_k=(1-I)*Q_{k-1}+I*\widehat{Q}_k$

3.2 CIoU

The DeepSort algorithm uses IoU for secondary matching between prediction boxes and detection boxes that have not been successfully matched after cascading matching, in order to improve tracking performance and reduce false positives and false negatives. The calculation formula for IoU is shown in the formula.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

In the formula, A∩B represents the area of overlap between the two bounding boxes of the prediction box and the detection box, and A∪B represents the area of union between the two bounding boxes of the prediction box and the detection box.

However, IoU matching only considers the overlap area and union area between two bounding boxes. When the detected tracking target has mismatched appearance features due to changes in appearance, and there is no overlap area between the predicted box and the detection box, i.e. IoU is 0, the ID number of the same tracking target will change. Therefore, in response to this issue, this article replaces the matching mechanism in the DeepSort algorithm with CIOU matching, as shown in the formula. CIoU can more accurately measure the matching degree of target boxes, enhance the stability and accuracy of data association, and perform better in situations where the target position changes significantly, by introducing a comprehensive consideration of center point distance, aspect ratio, and overlapping areas. The schematic diagram of CIoU matching is shown in Figure 10.

$$CIoU=1-IoU+\frac{\rho^2(b,b^{gt})}{c^2}+\alpha v$$

$$\alpha=\frac{v}{(1-IoU)+v}$$

$$v=\frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}}-\arctan\frac{w}{h}\right)^2$$

In the formula, b and b^{gt} represent the center points of the prediction box and the detection box, w and w^{gt} represent the widths of the prediction box and the detection box, h and h^{gt} represent the heights of the prediction box and the detection box, ρ represents the Euclidean distance between the two center points, c represents the diagonal length of the smallest bounding rectangle that can contain both the prediction box and the detection box, α represents the weight function, and v is used to measure the consistency of aspect ratio.

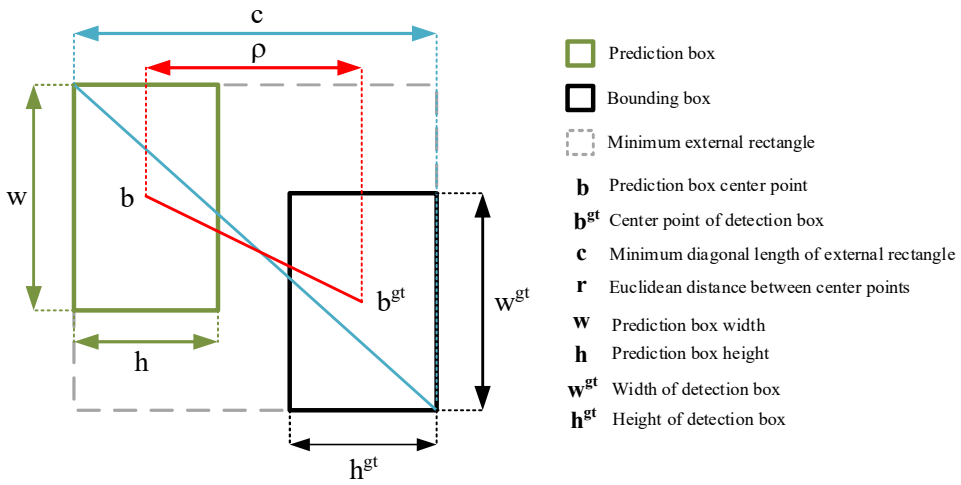

Figure 10. Schematic diagram of CIoU matching

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

4. Experimental Setup

4.1 Dataset

The dataset used in this article is the official multi-target tracking datasets MOT20 and MOT16 provided by MOTchallenge, which are widely used for evaluating pedestrian tracking algorithms. The MOT dataset covers a variety of different environments, such as city streets, shopping malls, etc., and involves various complex situations, including pedestrian occlusion, dense crowds, and dynamic tracking under different lighting, occlusion, and motion speeds. This article uses the MOT20 dataset for training and the MOT16 dataset for validation.

4.2 Experimental Platform and Training Parameters

The experimental environment configuration for this study is shown in Table 2.

Table 2. Experimental environment configuration

Configuration	Configuration Name	Details
Hardware Configuration	GPU	NVIDIA GeForce RTX 3090
	CPU	Intel Xeon Platinum 8260C CPU@2.30GHz
	RAM	24G
Software Configuration	System	Ubuntu 18.04
	CUDA	11.3
	Python	3.8
	Pytorch	1.10

The hyperparameter settings during the training process are shown in Table 3.

Table 3. Hyperparameter settings

Hyperparameters	Configuration
Image Size	640×640×3
Epoch	150
Batch Size	14
Optimizer	SGD
Momentum	0.937
Learning Rate	0.01
Learning Final	0.1

4.3 Object Detection Performance Evaluation Metrics

This article uses Mean Average Precision, FPS (Frames Per Second), computational complexity, parameter complexity, and model volume as evaluation metrics to assess the object detection capability of the model. The mAP calculation formula is shown below.

$$\text{Precision}=\frac{TP}{TP+FP}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

$$AP=\int_0^1 P(R)dR$$

$$mAP=\frac{\sum_{i=0}^{n} AP(i)}{n}$$

In the formula, TP represents the number of correctly predicted positive samples, FP represents the number of negative samples incorrectly predicted as positive samples, and FN represents the number of undetected positive samples.

4.4 Object Tracking Performance Evaluation Metrics

This article uses MOTA, HOTA, and IDF1 as evaluation indicators to assess target tracking capability. Among them, MOTA reflects the accuracy of multi-target tracking, and the calculation method is shown in the formula. HOTA is a comprehensive evaluation of the effectiveness of model detection, correlation, and localization, calculated as shown in the formula. IDF1 represents the F1 score for tracking IDs, mainly used to evaluate identity retention ability, with a focus on data association performance. The calculation method is shown in the formula.

$$MOTA=1-\frac{\sum(FN+FP+IDs)}{\sum GT}$$

$$HOTA=\sqrt{\frac{\sum_{c\in TP} A(c)}{TP+FN+FP}}$$

$$IDF1=\frac{2\times IDTP}{2\times IDTP+IDFP+IDFN}$$

In the formula, FN represents the number of missed targets, FP represents the number of false detected targets, IDs represents the total number of target ID switches, GT represents the number of real targets, TP represents the number of correctly matched targets, A (c) represents the accuracy of the association, IDTP represents the number of correctly matched IDs, IDFP represents the number of incorrectly matched IDs, and IDFN represents the number of unmatched IDs.

5. Experimental Results and Analysis

5.1 Ablation Study on Object Detection Algorithm

In this study, ablation experiments were conducted on different improved modules to verify the effectiveness of the improved object detection algorithm. The experimental results are shown in Table 4.

Table 4. Comparison of ablation experiments

YOLOv11n	CCFM_C3k2	RepNCSPELAN4-low	SENet	mAP50	mAP95	FPS	Computational Load /G	Parameter Count /M	Size/mb
√				86.5%	61.9%	417	6.4	2.58	5.2
√	√			84.6%	61.2%	588	5.4	1.81	3.7
√	√	√		85.2%	61.5%	556	5.7	1.65	3.5
√	√	√	√	86.8%	62.5%	526	5.8	1.67	3.5

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

According to Table 3, after improving the YOLOv11n network structure, the improved algorithm shows a clear optimization trend in multiple dimensions. By replacing the CCFM-C3k2 module, although the average accuracy mAP50 and mAP95 have decreased, the FPS, parameter count, and model volume have significantly increased, and the computational complexity has also been effectively optimized. Specifically, mAP50 and mAP95 decreased by 1.9% and 0.7% respectively, but FPS increased by 41% and computational load decreased by 15.6%. In addition, the parameter count has been reduced by 29.8% and the model size has been reduced by 28.8%, making the model more lightweight and providing better computational efficiency for embedded deployment and practical applications. While optimizing the model structure to achieve lightweighting, it also improves the ability to capture pedestrian contours, controls computational costs, and reduces the number of channels and RepNCSP modules in the RepNCSELAN4 module. By using the optimized RepNCSSPELAN4 low module and sacrificing some FPS and computational resources, the parameter count was further reduced by 6.2%, the model volume was reduced by 3.8%, and mAP50 and mAP95 were increased by 0.6% and 0.3%, respectively. To alleviate the accuracy loss caused by lightweight models, the SENet attention mechanism is introduced. On the basis of maintaining the current model performance, the average accuracy has been significantly improved, with mAP50 increasing by 1.6% and mAP95 increasing by 1%. Significantly enhanced the detection ability of the model in complex backgrounds, occlusions, and multi-target scenes, thereby further improving the robustness and generalization ability of the model. The experimental results show that by optimizing the YOLOv11 network structure, compared to the original YOLOv11 model, FPS has increased by 26.1%, computational complexity has decreased by 9.4%, parameter count has decreased by 35.3%, and model size has decreased by 32.7%. We have achieved lightweight deployment for pedestrian target detection tasks, with mAP50 and mAP95 reaching 86.8% and 62.5%, respectively.

5.2 Comparison Experiments on Object Detection Algorithm

In order to verify the effectiveness of the improved object detection algorithm, this paper compared it with YOLOv5, YOLOv6, YOLOv8, YOLOv10, YOLOv12, YOLOv13 and other algorithms. The experimental results are shown in Table 5. The experimental results show that the improved algorithm exhibits significant advantages in both lightweight indicators and detection accuracy.

Table 5. Experimental comparison table

Algorithm	mAP50/%	mAP95/%	FPS	Computational Load/G	Parameter Count /M	Size/mb
YOLOv5n	83.7	59.4	667	5.9	2.18	4.4
YOLOv6n	86.3	60.2	658	11.6	4.15	8.2
YOLOv8n	84.9	59.9	588	6.9	2.68	5.4
YOLOv10n	83.9	59.5	667	8.4	2.69	5.5
YOLOv11n	86.8	62.5	526	5.8	1.67	3.5
YOLOv12n	85.7	60.7	385	6	2.52	5.2
YOLOv13n	85.8	60.1	192	6.4	2.45	5.1

5.3 Ablation Study on Object Tracking Algorithm

In this study, ablation experiments were conducted on different improved modules to verify the effectiveness of the improved target tracking algorithm. The experimental results are shown in Table 6.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

Table 6. Comparison of ablation experiments

DeepSort	High and low score detection box	Adaptive kalman filter	CIoU	MOTA	HOTA	IDF1
√				78.3	57.5	71.8
√	√			78.5	59.0	72.3
√	√	√		79.2	61.7	73.7
√	√	√	√	80.2	62.8	77.1

According to Table 5, the experiment showed significant improvements in MOTA, HOTA, and IDF1 metrics by sequentially adding high and low score detection boxes, adaptive Kalman filtering, and CIoU modules. After introducing high-low score detection boxes, the model performance slightly improved, with MOTA increasing by 0.2%, HOTA increasing by 0.5%, and IDF1 increasing by 0.5%. After introducing adaptive Kalman filtering, MOTA, HOTA, and IDF1 improved by 0.7%, 2.7%, and 1.4% respectively, significantly enhancing the stability and accuracy of tracking, especially in predicting and updating targets in dynamic environments. Finally, by introducing CIoU, the model accuracy was further improved, with MOTA, HOTA, and IDF1 increasing by 1%, 1.1%, and 0.2% respectively. The regression of the target box was optimized, the position accuracy was improved, and the accuracy and consistency of target tracking were further enhanced. Finally, the experimental results show that the improved target tracking algorithm MOTA improves by 1.9%, HOTA improves by 5.5%, and IDF1 improves by 5.3%.

5.4 Comparison Experiments on Object Tracking Algorithm

In order to verify the effectiveness of the improved target tracking algorithm, this paper compared it with FairMOT, OC-SORT, SORT, ByteTrack and other algorithms. The experimental results are shown in Table 7. The experimental results show that the algorithm studied in this paper exhibits significant advantages in tracking accuracy and FPS.

Table 7. Experimental comparison table

Algorithm	MOTA	HOTA	IDF1	FPS
FairMOT	73.6	58.3	72.5	20
OC-SORT	78.2	62.1	77.4	22
SORT	80.0	61.2	78.2	23
ByteTrack	79.3	62.1	77.4	22
Article's algorithm	80.2	62.8	77.1	27

5.5 Visual Results Analysis

The visualization results of the test set in Figure 11 indicate that the algorithm proposed in this paper can accurately identify and label pedestrian targets in complex environments such as busy city streets and shopping centers, and successfully handle issues such as occlusion and target overlap. Even under dynamic conditions such as changes in lighting or camera translation, it maintains high accuracy and robustness, accurately tracking moving targets, fully demonstrating its wide adaptability and efficiency, and is suitable for fields such as intelligent monitoring and urban safety.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

Figure 11. Visualization results of MOT16 test set

6. Conclusion

This article proposes a target tracking algorithm that can run in real-time on edge devices based on YOLOv11n and DeepSORT algorithms. In response to the complex structure, large parameter count, and poor real-time performance of the YOLOv11n model, the RepNCSSPELAN low module is adopted to reduce the parameter count; Using CCFM-C3k2 module to further reduce computational complexity in the neck network; Introducing SENet adaptive channel attention mechanism to reduce the accuracy loss caused by lightweighting. In the tracking section, the ByteTrack framework is combined to improve the accuracy of trajectory matching; Applying adaptive Kalman filtering to enhance motion trajectory prediction; And use CIoU for secondary matching to reduce false positives and missed detections. The final test on the MOT16 dataset showed that the algorithm significantly reduced the number of parameters and computation compared to the original version, significantly improved the accuracy of multi-target tracking, and achieved a good balance between speed and accuracy, making it suitable for tracking tasks in complex scenarios.

Acknowledgments

Funding: This work was supported by the Tianjin Key Research and Development Program Institute-City Cooperation Project (No.23YFYSHZ00280) and Key Natural Science Project of Tianjin Municipal Education Commission's Scientific Research Program (No. 2022ZD032, 2022ZD026).

References

[1] Jiang Laiwei, Wang Ce, Yang Hongyu. A Review of Research Progress on Multi-Object Tracking Based on Deep Learning . Journal of Jilin University (Engineering Edition), 2025, 20(9): 1-17.

[2] Zhang Y, Sun P, Jiang Y, et al. Bytetrack: Multi-object tracking by associating every detection box(European Conference on Computer Vision), 2022: 1-21.

[3] Sun P, Cao J, Jiang Y, et al. Transtrack: Multiple object tracking with transformer , 2021, 5(04): 35-31.

[4] Wu Jiawen, Ji Wei, Zhai Kelong, et al. An Underwater River Crab Detection and Counting Method Based on YOLO-Crab and Improved DeepSORT. Electronic Measurement Technology, 2025, 9(20):1-10.

[5] Huang Kaiwen, Ling Liuyi, Wang Chengjun, et al. Real-time Multi-Object Tracking Algorithm Based on Improved YOLO and DeepSORT. Electronic Measurement Technology, 2022, 45(06): 7-13.

[6] Liu Zhaojin, Tan Qinhong, Zhu Jiahao, et al. Research on Pedestrian Tracking Algorithm Based on ByteTrack and Improved YOLOv11 Algorithm. Journal of Laser Magazine, 2025, 9(20), 41-47.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202512_11(12).0004

[7] Sheng Wenshun, Shen Jiahui, Chen Qi. Novel Multi-Target Pedestrian Tracking Method in Complex Traffic Scenarios. Liquid Crystals and Displays, 2025, 40(05): 785-795.

[8] Ren Anhu, Yuan Yang, Zhang Chenxi. Road Traffic Object Detection Based on YOLOv11. Journal of Laser Magazine, 2025, 9(25), 1-10.

[9] Zhou Jianxin, Li Zhongze, Hao Yingjie. Method for Detecting Surface Defects on Steel Plates Based on Improved YOLOv9. Electronic Measurement Technology, 2024, 47(22): 181-188.

[10] Gu Yingkui, Ye Biaobiao, Guo Mingjian, et al. Rapid Detection of Appearance Defects in Biscuit Packaging Based on Improved RT-DETR. Food and Machinery, 2025, 41(02): 234-241.

[11] Miao Shujiang, Hui Zhuofan, Shen Lie, et al. Detection Method for Weld Defects in Thermoplastic Materials of Fishing Vessels Based on Improved YOLOv8. Modern Fisheries, 2025, 52(04): 142-152.

[12] Zhang Bide, Wang Zelin, Liao Qilong, et al. Surface Defect Detection of Photovoltaic Modules Based on Lightweight Improved YOLOv8. Journal of Electronic Measurement and Instrumentation, 2025, 39(06): 100-111.

[13] Ma Xubang, Wu Xuanyu, Hu Bingtao, et al. Lightweight Real-time Detection Model for Defects in Aerospace Carbon Fiber Components Based on Multidimensional Collaborative Attention Mechanism . Computer Integrated Manufacturing Systems, 2025, 9(25), 1-20.

[14] Chen Meilong, Zhao Xinhua, Ye Xiufen. A Multi-target Tracking Algorithm for Forward-looking Sonar Based on ByteTrack. Journal of Instruments, 2025, 9(25), 1-13.

[15] Xu Huajie, Zheng Liwen. Visual Multi-target Tracking Based on Adaptive Kalman Filtering. Computer Engineering and Applications, 2025, 61(05): 200-210.

[16] Zhao Xiaoxia, Yuan Hongbo, Cheng Man. Multi-target Tracking of Sheep Using Deep Learning. Journal of Hebei Agricultural University, 2024, 47(06): 24-31.

[17] DU Y, WAN J, ZHAO Y, et al. GIAOTracker: A comprehensive framework for MCMOT with global information and optimizing strategies in VisDrone 2021[C]//Proceedings of the IEEE/CVF International conference on computer vision. IEEE, 2021: 2809-2819 AKHLAGHI S, ZHOU N, HUANG Z. Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation[C]//2017 IEEE power & energy society general meeting. IEEE, 2017: 1-5.

[18] Sun Mingfang, Lü Xu, Zhao Renjie, et al. Radar Target Tracking Algorithm Based on Innovation Adaptive Extended Kalman Filtering. Science Technology and Engineering, 2023, 23(9):3738-3743.