J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 8427--8439

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Lightweight Improved YOLOv8-based Multi-Vehicle Steering Trajectory Tracking Algorithm for Tunnel Scenes

Zhixian Li[1,✉], Nianfeng Shi[1,2], Guoqiang Wang[1,2], Liguo Zhao[1]

[1] *School of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang, Henan, 471023, China*
[2] *Henan Key Laboratory of Green Building Materials Manufacturing and Intelligent Equipment, Luoyang Institute of Science and Technology, Luoyang, Henan, 471023, China*

ABSTRACT
The study adopts a detection followed by tracking paradigm. In the detection stage, the BiFormer dynamic sparse attention module is embedded in the YOLOv8 network model, while the original nearest neighbor interpolation upsampling is improved by replacing it with the lightweight upsampling operator CARAFE. In the target tracking stage, a multi-vehicle steering trajectory tracking algorithm based on particle filtering is proposed, and the particle filtering algorithm is improved by combining the target motion direction weighted resampling algorithm. The two improved algorithms are combined for multi-vehicle detection and tracking in tunnel scenarios, and the average tracking accuracy can reach 97.3%. Compared with the traditional YOLOv8 combined with particle filtering algorithm for tracking, the method in this paper is more advantageous.

Keywords: YOLOv8, BiFormer, Lightweight, Particle Filtering, Vehicle Detection and Tracking

1. Introduction

As the cornerstone to support the development of economic undertakings, the construction of China's transportation has been growing rapidly, and highways have become the lifeblood of transportation by connecting cities to each other. Tunnels account for a large proportion of highways and are the throat section of the highway network, which are characterized by narrow internal space, relative closure, and high traffic flow, which lead to frequent tunnel safety problems [1-3]. Therefore, in order to ensure the safe operation and intelligent monitoring of tunnels, vehicle detection and tracking in tunnel scenarios is an important means.

The detection and tracking of moving vehicle targets is done by processing the underlying video pixels, using background modeling or modeling of the target to segment the moving vehicle from the

✉ Corresponding author.
E-mail address: lzx@lit.edu.cn (Z. Li).
Received 25 February 2024; Revised 10 April 2024; Accepted 15 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-465

scene, and after extracting the relevant features, these features are used to track the moving target using certain matching algorithms to search for the region that is closest to the vehicle's target template to predict where the target will be located in the next frame [4-8]. Single target tracking is better handled and the implementation of multi-target tracking faces many complexities such as mutual occlusion occurring between targets and the number of targets changing in real time [9-10]. The motion state of multi-targets in the scene is a nonlinear system, which also puts higher requirements on multi-target tracking algorithms [11-12]. Considering the real-time and accuracy of vehicle detection and tracking algorithms, the lightweight improved YOLOv8 multi-target detection algorithm is able to realize high-precision tunnel vehicle detection and car model recognition.

With the rapid development of artificial intelligence technology in recent years, vehicle detection is mainly represented by deep learning algorithms based on deep convolutional neural networks [13]. Abdallah, M. S. et al. introduced a multi-vehicle tracking model based on heterogeneous deep learning by utilizing the appearance and motion features of the target vehicle in consecutive frames, which was trained by a ternary loss minimization method to extract the appearance features and accurately predicts the probability distribution of the next appearance and motion features of the tracked object through a long and short-term memory network [14]. Zhang, H. et al. proposed a solution to the challenges associated within highway tunnel scenarios by designing a multi-camera multi-vehicle tracking system for tunnel roads that integrates cascaded multi-level multi-objective tracking strategies, trajectory refinement methods, and spatio-temporal constraints modules, and verified the system's effectiveness and robustness [15]. Zhang, X. et al. proposed a practical appearance resolving spatio-temporal trajectory matching network (ASTM-Net), which utilizes the multi-appearance attribute parsing (MAP) module to obtain global appearance attribute features through video sequences, and then employs the spatio-temporal matching (ASTM) module to achieve multi-camera trajectory-to-target assignment [16]. Zeng, Y. et al. designed the Wasserstein correlation metric multi-vehicle tracking method (MTWAM), which uses the 1-Wasserstein distance to measure the similarity between the tracking block and the detection block in order to intercept the partial occlusion problem, in addition to improving the feature representation of the vehicle using sparse coding of target-specific features to reduce the influence of interference [17]. Min, H. et al. for the tunnel scenario with anomalous lighting conditions by proposing an adaptive vehicle detection model that can take into account changes in the intensity of the illumination off, using a brightness adaptive adjustment module to adjust the brightness of the image to a normal level, and then utilizing YOLOv7 for vehicle detection, which significantly improves the vehicle detection accuracy [18]. Lee, K. B. et al. showed that there are serious perspective aberrations in the images from the tunnel CCTVs, which, combined with spatial constraints in the tunnels, make the target detection task more arduous, a scheme based on region of interest setting and inverse perspective transformation technique is proposed to transform the image, which maintains the target detection performance and appearance speed across distance by enlarging the distant targets [19]. You, L. et al. proposed a multi-target vehicle detection and tracking algorithm based on CDS-YOLOv8 and improved ByteTrack, and experiments showed that the proposed algorithm still has superior performance in complex, diverse and occluded scenes [20].

The study first improves the YOLOv8 detection algorithm by introducing the BiFormer sparse dynamic attention module in the network structure for extracting small target feature information. Meanwhile, the lightweight upsampling operator CARAFE is used to replace the original nearest-neighbor interpolation upsampling, which reduces the problem of small target feature loss in the upsampling process. Subsequently, a multi-vehicle steering trajectory tracking algorithm based on particle filtering is proposed, and the particle filtering algorithm is improved by combining the target motion direction weighted resampling algorithm. Finally, the two improved algorithms are combined for multi-vehicle detection and tracking in tunnel scenes.

2. Research on multi-vehicle steering trajectory tracking in tunnel scenarios

2.1. Vehicle detection algorithm based on improved YOLOv8

2.1.1. Principles of YOLOv8. YOLO series has the advantages of fast speed, high detection accuracy and wide applicability in target detection, which is widely used in various scenarios of actual production and life. YOLOv8 is the most advanced one-stage target detection algorithm in YOLO series at present, and the model is based on YOLOv5 with new improvement methods added, which makes the detection accuracy effectively improved [21]. The framework of this algorithm is mainly composed of four parts: input, backbone module, neck module and head module. Compared with YOLOv5, the main improvement ideas are as follows:

1) Backbone module: the backbone network still adopts the backbone network structure of YOLOv5, but the original C3 module is replaced by the C2f module with richer gradient flow, which not only makes the module lighter, but also enhances the fusion ability of the features.

2) Neck Module: The neck module structure also adopts the neck network structure of YOLOv5, while also replacing the C3 module with the C2f module, and deleting the convolution module in the upsampling stage of PAN-FPN.

3) Loss function: YOLOv8 adopts the Anchor-Free idea, therefore, only the classification loss BCE Loss and the regression loss CIOU Loss+DFL are used.

4) Sample allocation strategy: YOLOv8 adopts Task-AlignedAssigner positive and negative sample dynamic matching method to replace the original allocation method with IOU matching or one-sided proportion.

2.1.2. YOLOv8 model improvement strategy:

1) BiFormer module. In order to improve the feature extraction capability for small targets, this paper introduces the BiFormer dynamic sparse attention module based on the Transformer variant in the YOLOv8 network model, which proposes a two-layer routing attention mechanism (BRA) to alleviate the scalability problem of multiple self-attention (MHSA).The BiFormer module is designed in accordance with Vision BiFormer is composed of several BiFormer Block modules, and the BRA module is the core module of BiFormer Block.The core idea of BRA is to filter out irrelevant key-value pairs in the coarse region of the feature graph. Regions to filter out irrelevant key-value pairs, i.e., by constructing and pruning region-level directed graphs, and then applying fine-grained token-to-token attention in the union of regions to realize the filtering operation [22]. The specific implementation of the BRA module is given below (Figure 1).

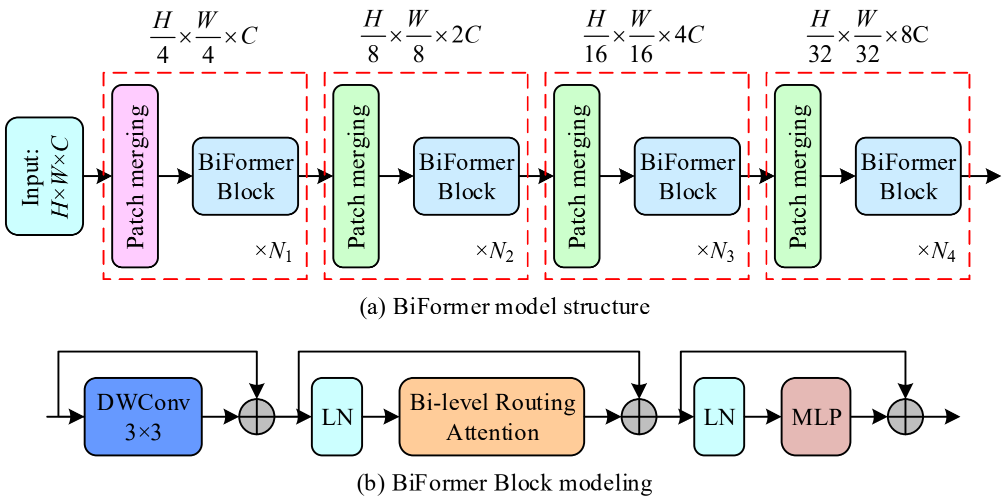

Fig. 1. BiFormer, BiFormer Block model structure diagram

Assuming an input image with width W, height H, and number of channels C, represented by $X \in R^{H\times W\times C}$, the input image is divided into $S\times S$ regions, each region is characterized by HW/S^2, then X is represented by $X^r \in R^{S^2\times\frac{HW}{S^2}\times C}$, which is then obtained by linear projection:

$$Q = X^r W^q, K = X^r W^k, V = X^r W^v \tag{1}$$

In Eq. (1), W^q, W^k, $W^v \in R^{C\times C}$, are the projection weights of $Q(query)$, $K(key)$, $V(value)$, Q, K, $V \in R^{S^2\times\frac{HW}{S^2}\times C}$, respectively. then it is necessary to find the portion of each region of interest, i.e., first calculate the mean Q^r, K^r of Q, K in the region, and then calculate the inter-region correlation matrices of both:

$$A^r = Q^r\left(K^r\right)^T \tag{2}$$

Then prune the correlation map by keeping only the first $k(top-k)$ connection in each region. The index matrix is obtained using the rank $top-k$ operator:

$$I^r = topkInder\left(A^r\right) \tag{3}$$

$I^r \in N^{S^2\times K}$, where row i of I^r contains the indexes of the first k most relevant regions of the i th region. The tensor of Key and Value is then aggregated to get:

$$K^g = gather\left(K, I^r\right), V^g = gather\left(V, I^r\right) \tag{4}$$

Is then obtained by using the attention operation on the aggregated K^g, V^g:

$$\begin{aligned} O &= Attention\left(Q, K^g, V^g\right) + LCE(V) \\ &= soft\max\left(\frac{Q\left(K^g\right)^T}{\sqrt{C}}\right)V^g + LCE(V) \end{aligned} \tag{5}$$

where $LCE((V))$ is the local context enhancement term. Therefore, the BRA module not only captures the image contextual feature information, but also greatly reduces the computational complexity.

In this paper, the BRA module in BiFormer is introduced into the YOLOv8 network model. When the input image passes through the SPPF module, the image at this time has been downsampled by 32 times, and the smallest size of the feature map is obtained, therefore, adding a convolution module and a BRA module at this position can capture the fine information of the feature map, enhance the correlation of the image contextual information, and increase the capability of feature extraction of small targets in the feature map. The ability of feature extraction for small targets in the feature map can be improved.

2) CARAFE module. In YOLOv8 algorithm, the up-sampling operation is mainly based on the commonly used nearest-neighbor linear interpolation, whose function is to enlarge the resolution of the feature map, so that the network is able to integrate feature information from different scales. However, when dealing with dense small target scenes, only the values of the nearest neighbor pixels in the image are used to weight with the up-sampling kernel without using the semantic information of the feature map, which leads to the loss of details and texture information in the image, making the interpolation results appear too smooth and unable to accurately restore the details of the original data.

To address this problem, this paper replaces the nearest-neighbor linear interpolation method in YOLOv8 with the lightweight up-sampling operator CARAFE, which is a content-aware feature reorganization up-sampling method, compared to linear interpolation up-sampling, by chunking the input feature map and reorganizing the features of each chunk by using convolution operation. This approach enhances the perceptual range, enabling the up-sampled feature map to incorporate more contextual information and maintain the details and edge information of the original feature map, thus improving the quality and feature expression of the up-sampling [23]. The CARAFE operator implementation flow is shown in Fig. 2. It is mainly accomplished through the two modules of up-sampling kernel prediction and feature reorganization. After an image passes through the BRA module in the improved YOLOv8 model in this paper, a feature map of shape $H \times W \times C$ is obtained, which is then input into the up-sampling operator CARAFE.The specific implementation steps of up-sampling using the CARAFE operator are as follows:

1) Firstly, the number of channels C of the input feature map is compressed to C_m, in order to reduce the amount of computation in the subsequent steps.

2) Then use the convolutional layer of $k_{encoder} \times k_{encoder} \times \sigma^2 k_{up}^2$ to predict the up-sampling kernel for the feature map after the channel number compression, and get an up-sampling kernel with the shape of $H \times W \times \sigma^2 k_{up}^2$, and then unfold its channel dimensions into the width and height dimensions, at this time, the up-sampling kernel with the shape of $\sigma H \times \sigma W \times k_{up}^2$ is obtained, where σ is the up-sampling multiplicity, and k_{up}^2 is the size of the up-sampling. Finally, the upsampling kernel is softmax normalized so that its convolution kernel weights sum to 1.

3) In the feature reorganization module, each position of the output feature map is mapped back to X, and then take out the $k_{up} \times k_{up}$ region with this as the center point, and make dot product of the obtained region with the upsampling kernel at this point, and different channels at the same position share the same upsampling kernel, and finally output a feature map with the shape of $\sigma H \times \sigma W \times C$.

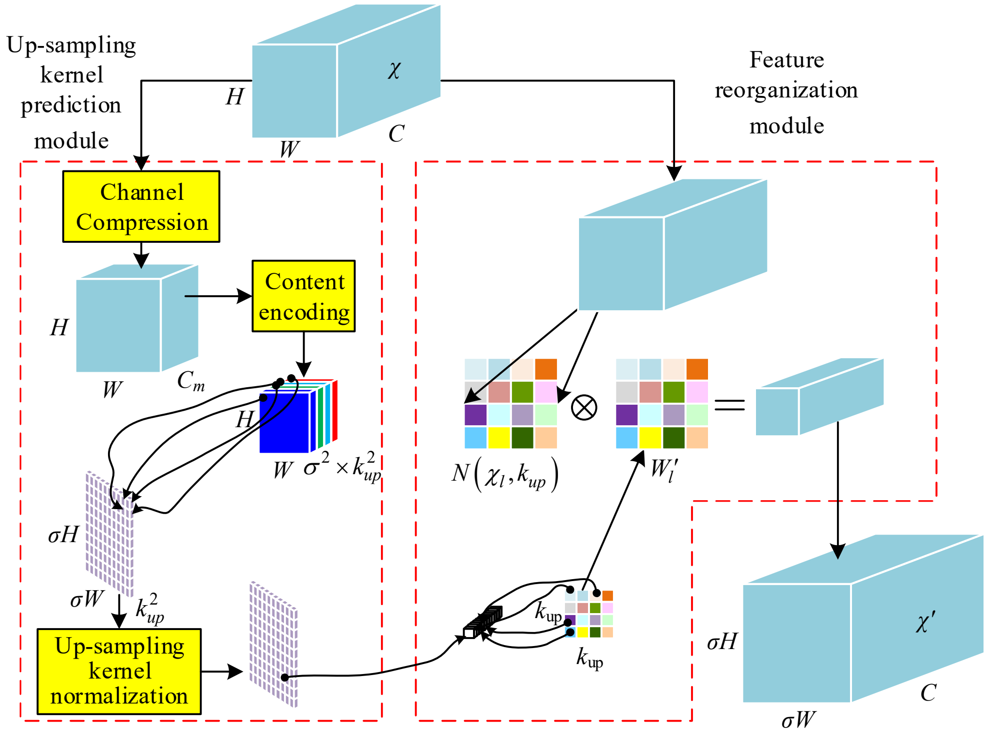

Fig. 2. CARAFE model structure

2.2. Multi-target tracking algorithm under particle filtering algorithm

The particle filtering method is based on the Bayesian filter estimation framework, and adopts a nonparametric Monte Carlo integration method to reduce the high-dimensional integration operation in the recursive process, which has no limitation on the system noise, and is applicable to any nonlinear system that can be represented by a state space model. And the particle filtering algorithm has a parallel structure, which itself enables the tracking of multiple targets [24].

2.2.1. Steps of multi-target tracking algorithm under particle filtering algorithm:

1) Initializing particles. At moment t_0, the set of particles with weights $\{\omega_k^i, i=0,\ldots,N_s\}$ corresponding to particle swarm $\{x_0^i\}_{i=0}^{N_s}$ is generated by a priori conditional probability $p(x_k, n)$.

2) Weights update. The weights of the particles are updated after initialization to obtain the weights of each particle $\{x_0^i\}_{i=0}^{N_i}$ at moment k:

$$\omega_k^i = \omega_{k-1}^i p\left(y_k \middle| x_k^i\right) = w_{k-1}^i p\left(y_k\right) - h\left(x_k^i\right), i = 1, 2, \ldots, N_s \tag{6}$$

where $p\left(y_k \middle| x_k^i\right)$ is the posterior probability distribution of the target state x_k at moment k, $h\left(x_k^i\right)$ is a bounded nonlinear mapping, and x_k and y_k are the state and measurement vectors of the system at moment k, respectively.

3) Normalized particle weights:

$$\hat{\omega}_k^i = \omega_k^i \Big/ \sum_{i=1}^{N_s} \omega_k^i \tag{7}$$

4) Resampling. If $Neff < N_{th}$, resampling is performed, otherwise proceed to the next step. N_{th} is the resampling threshold. If the particles have been degraded very badly, resample the particle set to get a new particle set as $\left\{x_{0:k}^{i*}, i = 0,1,2\ldots, N_S\right\}$.

5) Output. p_k is the variance of the output state. τ is a scale factor which can be taken as 0. The state is estimated as follows:

$$x_k = \sum_{i=1}^{N} \omega_k^i x_k^i \tag{8}$$

2.2.2. Improved particle filter tracking algorithm. The tunnel environment is unidirectional motion, so in general the vehicle is moving in one direction, i.e., the target motion state is a continuous process, and the target's motion direction in the current frame does not change much and is correlated with the direction in the previous n frame. This paper proposes a target motion direction weighted resampling approach, biased resampling, the relevant process is as follows:

1) Target motion direction prediction Set the current frame as the k nd frame, divide the first m frames into two parts, in which the n th frame as the center point, respectively, to calculate the center of the two parts of the target is located in the average position of the center of the two centers of the connecting vector Vp that is, the direction of the two centers $\tan\theta$ is the next frame of the target's possible direction of motion.

$$\tan\theta = \frac{\frac{1}{m-n}\left(\sum_{i=k-m+n}^{k} y_i\right) - \frac{1}{n}\left(\sum_{i=k-m}^{k-m+n-1} y_i\right)}{\frac{1}{m-n}\left(\sum_{i=k-m+n}^{k} x_i\right) - \frac{1}{n}\left(\sum_{i=k-m}^{k-m+n-1} x_i\right)}, n < m \tag{9}$$

2) Motion direction weight calculation is based on the center point of the current frame, and the direction within $\frac{\pi}{2}$ range of the left and right of the vector direction (i.e., $0-\pi$) is divided into 8 equal arcs of space and assigned a weight of $w_p^{(i)}$. The closer the direction is to V_p, the larger the weight is.

3) Calculate the weights of the particles. Multiply the weights based on vector Vp with the particle similarity to get new particle weights $w_n^{(i)} = w_p^{(i)} - w_k^{(i)}$ and normalize them as particle weights.

4) Perform resampling.

5) Read the next frame and redo the particle initialization.

2.3. Lightweight Improved YOLOv8 and Particle Filter Tracking Algorithm Based on

The standard particle filtering algorithm in the sampling stage, taken is undifferentiated to the entire search space random sampling method, the tunnel environment background by the light, sand, dust, exhaust and other factors of interference, will lead to tracking the target in the process, the particles received noise interference, the wrong transfer tracking target, which led to the particles are more and more dispersed, and ultimately lead to the particle depletion, tracking failure. In order to reduce the meaningless sampling of a wide range of background areas, this paper reduces the dispersion of particles and reduces the computational amount by sampling particle samples from the target area obtained in the target detection stage [25]. At the same time, for the unidirectional driving environment in the tunnel, combined with the target movement direction weighted sampling algorithm to improve the particle filtering algorithm, the relevant algorithm steps are as follows:

Step 1: Foreground Detection Firstly by the improved YOLOv8 algorithm, we can get an accurate target motion region centered at (x_0, y_0), and then particle sampling is performed on the target region.

Step 2: Feature extraction and particle initialization. Build a model with the weighted HSV color histogram of the kernel function of the foreground target region $\{q_u\}$ to extract features, and use the sequential importance sampling method to get an initial sample set $\left\{X_0^{(i)}, \frac{1}{N}\right\}_{i=1}^{N}$.

Step 3: Eigenvalue matching calculates the $\{p\}$ of particle $X_k^{(i)}$ and the target template $\{q\}$ of the two kernel functions HSV color histogram similarity, and calculates the Barcol coefficient as shown in equation (10):

$$\rho(p,q) = \sum_{u=1}^{m} \sqrt{p_u q_u} \tag{10}$$

The baroclinic distance between the two particle features and the target feature can be obtained as shown in Eq:

$$d_{HSV} = \sqrt{1 - \rho(p,q)} \tag{11}$$

Step 4: Weight normalization defines the weight $\omega_k^{(i)}$ of the particle as shown in Eq:

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-d^2/2\sigma^2\right) \tag{12}$$

When the value of the barometer distance d is smaller, the particle weights are larger. The weights of the particles are normalized so that all weights sum to 1 as shown in Eq:

$$\omega_k^{(i)} = \omega_k^{(i)} \Big/ \sum_{i=1}^{N} \omega_k^{(i)} \tag{13}$$

Step 5: Position prediction weights indicate the size of the likelihood that the target is located, and the particle with the largest weight indicates the location of the target.

Step 6: Particle Resampling As the particles predict the state of the target, the number of particles with larger weights decreases, and resampling is required to calculate the number of particles that can effectively describe the state of the system $Ne\hat{f}f$, as shown in Eq:

$$Ne\hat{f}f = \frac{1}{\sum_{i=1}^{N} \left(\omega_k^{(i)}\right)^2} \tag{14}$$

The effective number of particles $Neff$ is determined and resampling is performed when $Neff \le N_{th}$. The threshold of this algorithm is $N_{th} = \frac{2}{3}N$.

3. Experimental results and application effect analysis

3.1. Experimentation and Analysis

3.1.1. Experimental data set. The target detection dataset established in this paper uses images taken by multiple highway tunnel cameras, camera shooting height 5~8m, image size 1920×1080, a total of 29332 images, of which 23548 images in the training set, 2600 images in the validation set, and 3184 images in the test set, and the vehicles are labeled as Car, Truck, and Bus categories, and the data covers Various common tunnel scenarios, including high and low shooting angles, high and low lighting conditions, 42,633 labeled frames in Car category, 22,251 labeled frames in Truck category, and 2,766 labeled frames in Bus category.

3.1.2. Evaluation indicators. The target detection experiments use accuracy (P), recall (R), multiple categories average precision (mAP), and frames per second (FPS) to judge the performance of the algorithm.

The commonly used evaluation metrics for multi-target tracking are multi-target tracking accuracy (MOTA) and multi-target tracking precision (MOTP), which are calculated as shown in equations (15) and (16):

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + e_t)}{\sum_t g_t} \tag{15}$$

$$MOTP = 1 - \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{16}$$

Wherein, m_t, fp_t, and e_t denote the number of missed, misdetected, and incorrectly matched targets by the detector at frame t of the video, g_t denotes the number of true targets in frame t. d_t^i denotes the distance between the target in frame t and the detection frame in frame i, and c_t denotes the number of successfully matched targets in frame t.

3.1.3. Vehicle target detection experiment. In this subsection, the images in the test set are detected by using the target detection dataset trained on the lightweight improved YOLOv8 based model, while multiple rounds of training are performed in order to obtain more accurate weights, and the training is terminated when the average cost function stops decreasing, the rounds with the optimal evaluation of the metrics are selected, and the weight files are used as the weights of the model. The loss function training curve is shown in Figure 3, with the increase in the number of network training, the model gradually mastered the characteristics of the dataset, the loss value is decreasing and stabilized after 95 rounds of training.

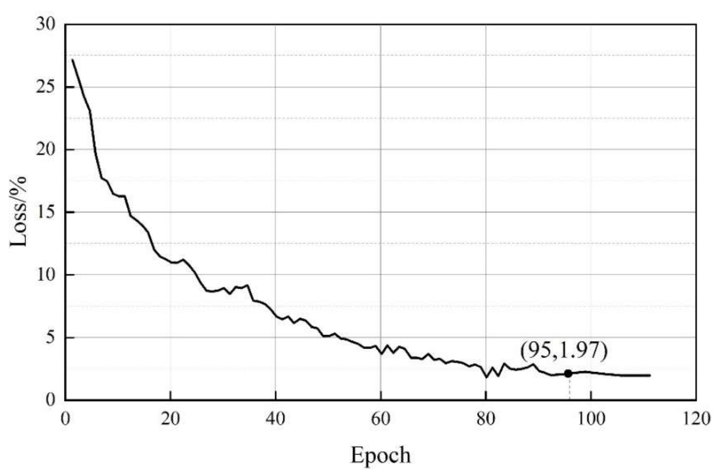

Fig. 3. Loss function training curve

The average detection accuracy of the model after training is shown in Figure 4. From the figure, it can be seen that the vehicle detection model $mAP_{0.5}$, which uses the optimal weights selected after multiple rounds of training, reaches about 82% in the 95th round, but at this time, the average detection accuracy of the model is still in an unstable state, and when the model is trained to the 225th round or so, the detection accuracy of the model reaches a relatively smooth state, which indicates that the model has been trained at this time, and its average detection accuracy $mAP_{0.5}$ reaches up to 82.12%.

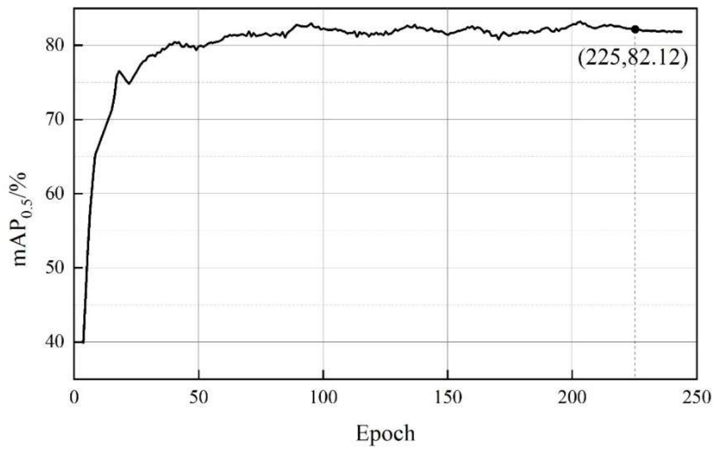

Fig. 4. The average detection accuracy of the post-training model

The vehicle target detection ablation experiments are shown in Table 1. The experimental results show that the detector accuracy is improved by 1.24 percentage points after the lightweight improvement of the YOLOv8 model in this paper, and it has less impact on the detection speed, and the number of detected frames per second can be up to 95, which proves the effectiveness of the improved method.

Table 1. Vehicle detection algorithm ablation experiment

Algorithm	$mAP_{0.5}$/%	P/%	R/%	FPS
YOLOv8	84.63	86.47	83.55	91
Ours	85.87	87.99	85.02	95

3.1.4. Vehicle tracking experiments. In this experiment, target tracking based on lightweight improved YOLOv8 combined with improved particle filter tracking algorithm is used. In order to verify the feasibility of its method, a comparison experiment is set up in this subsection to measure the tracking performance in terms of several aspects of metrics and analyze its actual performance on the dataset. The experimental results and analysis are specified as follows:

The comparison of tracking performance of different models is shown in Fig. 5. Compared with the SORT model, the multi-target tracking accuracy and multi-target tracking precision of the model using the improved particle filter tracking algorithm are improved by 21.45% and 18.61%, respectively. The multi-target tracking performance of the improved model proposed in this paper is better than other models. It can be seen that due to the addition of YOLOv8 as a target detector, the multi-target tracking accuracy and tracking precision of the DeepSORT+YOLOv8 model are improved by 5.32 and 3.05 percentage points, respectively, compared with the combination of DeepSORT, which suggests that YOLOv8 as a target detector can significantly improve the vehicle tracking effect.

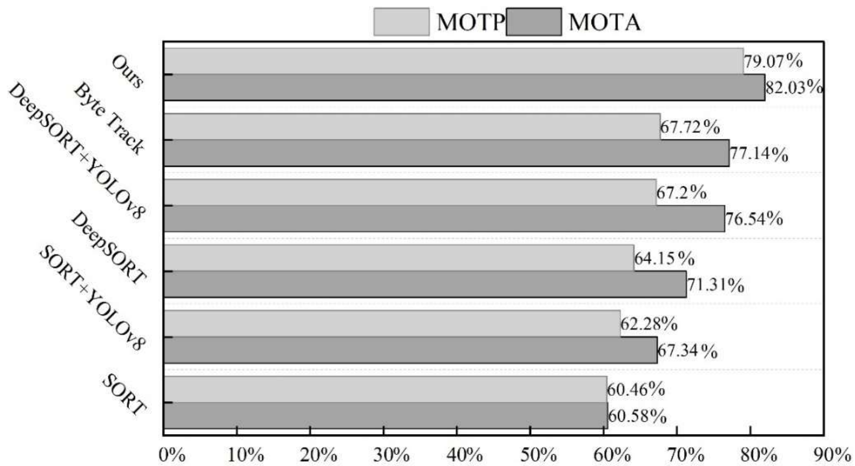

Fig. 5. Different model tracking performance comparison

3.2. *Actual vehicle detection tracking effect analysis*

The experiments use some of the surveillance video pairs in a provincial highway tunnel to conduct vehicle detection and tracking experiments on the trained model.

3.2.1. Analysis of the effectiveness of vehicle detection. The trained lightweight improved YOLOv8 based model is used for vehicle detection experiments and the vehicle detection results are shown in Table 2. The model performance is good and the average accuracy reaches 98.41%.

Table 2. The YOLOv8 model was improved before and after the error detection rate

Vehicle type	Actual quantity	Correct quantity	Accuracy rate	Average accuracy
Car	203	200	98.52%	98.41%
Bus	46	45	97.83%	
Truck	89	88	98.88%	

3.2.2. Analysis of vehicle tracking effectiveness. A lightweight improved YOLOv8 based combined with improved particle filter tracking algorithm is used for target tracking. The vehicles in the tunnel are tracked in the same video scene to evaluate the performance of the tracking method by the accuracy of the traffic count. The tracking results are shown in Table 3. The method performs well in tracking in the highway tunnel scene, and the average accuracy of the traffic count reaches 97.3%, among which the tracking accuracy of cars is the highest, reaching 98.5%. Compared with the traditional YOLOv8 method of detecting and tracking each image frame in combination with particle filtering, the present method has a great advantage in terms of frame rate, with an average frame rate increase of 14 frames. As a whole, the present method can meet the needs of vehicle detection and tracking applications in highway tunnel scenarios.

Table 3. Tracking result

Method	Vehicle type	Actual quantity	Algorithm statistics	Accuracy rate/%	Average accuracy/%	Average frame rate/(Frame $\cdot$ s^{-1}
YOLOv8+Particle filtering	Car	203	187	92.1	89.3	13
	Bus	46	39	84.8		
	Truck	89	81	91		
Ours	Car	203	200	98.5	97.3	27
	Bus	46	44	95.7		

	Truck	89	87	97.8		

4. Conclusion

The study combines the lightweight improved YOLOv8 algorithm with the improved particle filtering algorithm to conduct test experiments for multi-vehicle steering trajectory tracking.

In this paper, the detection accuracy of the lightweight improved YOLOv8-based model is improved by 1.24 percentage points compared with the traditional YOLOv8. It shows that the use of lightweight improved YOLOv8-based model in the detection stage can provide a good foundation for the vehicle tracking stage.

After improving the particle filtering algorithm by combining the target motion direction weighted resampling algorithm, the lightweight improved YOLOv8 algorithm and its combination are used for multi-vehicle detection and tracking in the tunnel scene. The average tracking accuracy can reach 97.3%. Compared with the traditional YOLOv8 combined with particle filtering algorithm for tracking, the tracking method in this paper achieves the expected results, and the frame rate is greatly improved compared with the traditional method, which can fully meet the real-time requirements, and can satisfy the needs of vehicle detection and tracking applications in tunnel scenes.

Funding

This work was sponsored in part by Science and Technology Research Project of Henan Province (232300420157).

References

[1] Yu, F., Xie, Z., & Jia, Y. (2023, May). Highway Tunnel Environment Perception System Based on Internet of Things and Cloud Computing Technology. In 2023 4th International Conference for Emerging Technology (INCET) (pp. 1-6). IEEE.

[2] Abdulrahim, K., & Salam, R. A. (2016). Traffic surveillance: A review of vision based vehicle detection, recognition and tracking. Int. J. Appl. Eng. Res, 11(1), 713-726.

[3] Švorc, D., & Tichý, T. (2023). R užicka, M.; Ivasienko, P. Use of One-Stage Detector and Feature Detector in Infrared Video on Transport Infrastructure and Tunnels. Sustainability 2023, 15, 2122.

[4] Wang, S., Zhuang, X., Gao, H., & Wu, J. (2022). A vehicle trajectory tracking method based on roadside LiDAR data in tunnel. In Advances in Urban Engineering and Management Science Volume 2 (pp. 381-387). CRC Press.

[5] Liu, C., Liu, H., Han, L., & Xiang, C. (2023). New integrated multi-algorithm fusion localization and trajectory tracking framework of autonomous vehicles under extreme conditions with non-Gaussian noises. International journal of automotive technology, 24(1), 259-272.

[6] Rafique, A. A., Al-Rasheed, A., Ksibi, A., Ayadi, M., Jalal, A., Alnowaiser, K., ... & Park, J. (2023). Smart traffic monitoring through pyramid pooling vehicle detection and filter-based tracking on aerial images. IEEE Access, 11, 2993-3007.

[7] Liu, Y., Zhang, X., Zhang, B., Zhang, X., Wang, S., & Xu, J. (2022). Multi-camera vehicle tracking based on occlusion-aware and inter-vehicle information. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3257-3264).

[8] Tang, Z., Naphade, M., Liu, M. Y., Yang, X., Birchfield, S., Wang, S., ... & Hwang, J. N. (2019). Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8797-8806).

[9] Tran, D. N. N., Pham, L. H., Jeon, H. J., Nguyen, H. H., Jeon, H. M., Tran, T. H. P., & Jeon, J. W. (2022). A robust traffic-aware city-scale multi-camera vehicle tracking of vehicles. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3150-3159).

[10] Li, P., Li, G., Yan, Z., Li, Y., Lu, M., Xu, P., ... & Chuxing, D. (2019, June). Spatio-temporal Consistency and Hierarchical Matching for Multi-Target Multi-Camera Vehicle Tracking. In CVPR Workshops (pp. 222-230).

[11] Luo, Z., Bi, Y., Yang, X., Li, Y., Yu, S., Wu, M., & Ye, Q. (2024). Enhanced YOLOv5s+ DeepSORT method for highway vehicle speed detection and multi-sensor verification. Frontiers in Physics, 12, 1371320.

[12] Min, H., Li, Y., Wu, X., Wang, W., Chen, L., & Zhao, X. (2023). A measurement scheduling method for multi-vehicle cooperative localization considering state correlation. Vehicular Communications, 44, 100682.

[13] Li, M., Liu, M., Zhang, W., Guo, W., Chen, E., & Zhang, C. (2024). A Robust Multi-Camera Vehicle Tracking Algorithm in Highway Scenarios Using Deep Learning. Applied Sciences, 14(16), 7071.

[14] Abdallah, M. S., Han, D. S., & Kim, H. (2022). Multi-vehicle tracking using heterogeneous neural networks for appearance and motion features. International Journal of Intelligent Transportation Systems Research, 20(3), 720-733.

[15] Zhang, H., Fang, R., Li, S., Miao, Q., Fan, X., Hu, J., & Chan, S. (2024). Multi-Camera Multi-Vehicle Tracking Guided by Highway Overlapping FoVs. Mathematics, 12(10), 1467.

[16] Zhang, X., Yu, H., Qin, Y., Zhou, X., & Chan, S. (2024). Video-Based Multi-Camera Vehicle Tracking via Appearance-Parsing Spatio-Temporal Trajectory Matching Network. IEEE Transactions on Circuits and Systems for Video Technology.

[17] Zeng, Y., Fu, X., Gao, L., Zhu, J., Li, H., & Li, Y. (2020). Robust multivehicle tracking with wasserstein association metric in surveillance videos. IEEE Access, 8, 47863-47876.

[18] Wei, Y., Zeng, Z., He, T., Yu, S., Du, Y., & Zhao, C. (2024). An Adaptive Vehicle Detection Model for Traffic Surveillance of Highway Tunnels Considering Luminance Intensity. Sensors, 24(18), 5912.

[19] Lee, K. B., Gong, J. H., Ryu, B. H., & Shin, H. S. (2023). Deep Learning-Based Consistent Object Detection in Distance on Highly Perspective Region. Applied Sciences, 13(24), 13169.

[20] You, L., Chen, Y., Xiao, C., Sun, C., & Li, R. (2024). Multi-Object Vehicle Detection and Tracking Algorithm Based on Improved YOLOv8 and ByteTrack. Electronics, 13(15), 3033.

[21] Athulya Sundaresan Geetha,Mujadded Al Rabbani Alif,Muhammad Hussain & Paul Allen. (2024). Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy. Vehicles(3),1364-1382.

[22] Shi Lei,He Yi & Jeffrey S Sarmiento. (2024). Synchronous End-to-End Vehicle Pedestrian Detection Algorithm Based on Improved YOLOv8 in Complex Scenarios. Sensors (Basel, Switzerland)(18),6116-6116.

[23] Zhongmin Huangfu & Shuqing Li. (2023). Lightweight You Only Look Once v8: An Upgraded You Only Look Once v8 Algorithm for Small Object Identification in Unmanned Aerial Vehicle Images. Applied Sciences(22).

[24] Duo Peng,Kun Xie & Mingshuo Liu. (2024). Application of Gray Wolf Particle Filter Algorithm Based on Golden Section in Wireless Sensor Network Mobile Target Tracking. Electronics(13),2440-2440.

[25] Abdulla Abdalgalil Alsagair,Graovac Stevica,Papic Veljko & Kovacevic Branko. (2021). Triple-feature-based Particle Filter Algorithm Used in Vehicle Tracking Applications. ADVANCES IN ELECTRICAL AND COMPUTER ENGINEERING(2),3-14.