HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Journal of King Saud University - Computer and Information Sciences

journal homepage: www.sciencedirect.com

Full length article

Enhancing autonomous driving through intelligent navigation: A comprehensive improvement approach

Zihao Xu [a,1], Yinghao Meng [a,1], Zhen Yin [a], Bowen Liu [b], Youzhi Zhang [a,*], Mengmeng Lin [a]

[a] *College of Mechanical and Electrical Engineering, Wenzhou University, Wenzhou, Zhejiang 325000, China*
[b] *College of Big Data and Intelligent Engineering, Southwest Forestry University, Kunming, Yunnan 650000, China*

ARTICLE INFO

Keywords:
Intelligent navigation
YOLOv7x-CM
Off-center angle
TEB-S
Obstacle avoidance

ABSTRACT

In this paper, an intelligent navigation system is developed to achieve accurate and rapid response to autonomous driving. The system is improved with three modules: target detection, distance measurement, and navigation obstacle avoidance. In the target detection module, the YOLOv7x-CM model is proposed to improve the efficiency and accuracy of target detection by introducing the CBAM attention mechanism and MPDioU loss function. In the obstacle distance measurement module, the concept of an off-center angle is introduced to optimize the traditional monocular distance measurement method. In the obstacle avoidance module, acceleration jump and steering speed constraints are introduced into the local path planning algorithm TEB, and the TEB-S algorithm is proposed. Finally, this paper evaluates the performance of the system modules using the KITTI dataset and the BDD100K dataset. It is demonstrated that YOLOv7x-CM improves the mAP @ 0.5 metrics by 5.3% and 6.8% on the KITTI dataset and the BDD100K dataset, respectively, and the FPS also increases by 35.4%. Secondly, for the optimized monocular detection method, the average relative distance error is reduced by 9 times. In addition, the proposed TEB-S algorithm has a shorter obstacle avoidance path and higher efficiency than the normal TEB algorithm.

1. Introduction

As an important part of autonomous driving, intelligent navigation mainly allows driverless vehicles to perceive the environment in real time, plan the best path, and safely drive to the destination (Pang et al., 2002). However, various obstacles on the road may threaten or interfere with vehicle traveling. Road conditions change rapidly, especially when the vehicle speed is high or the obstacles are obscured. The only way to ensure driving safety is to detect these obstacles quickly and complete the decision-making instantly (Large et al., 2005). However, the current research on intelligent navigation technology regarding the rapidity and accuracy of the perception and decision-making module is still insufficient to meet the requirements of instantaneous response and decision-making of vehicles (Zhang et al., 2017; Lu et al., 2021).

The real-time target detection function is the basic module of the automatic driving system. Deep learning-based target detection algorithms can be categorized into single-stage and two-stage detection algorithms. The two-stage algorithms, from the initial R-CNN (Girshick et al., 2014), gradually developed into SPPNet (He et al., 2015), Fast-RCNN (Girshick, 2015), and Faster-RCNN (Ren et al., 2015), which realize detection through two stages of candidate region generation and classification. Saranya et al. (2022) applied CNN algorithms to an autonomous driving target detection task to recognize road boundaries and obstacles, exhibiting good detection performance. Compared to the former, the previous single-stage algorithms have the advantages of fast detection and small computation, although they have lower accuracy. The development of single-stage algorithms ranges from YOLO (You Only Look Once) (Redmon et al., 2016) to SSD (Single Shot MultiBox Detector) (Liu et al., 2016) to RetinaNet (Lin et al., 2017), YOLOv2 (Redmon and Farhadi, 2017), YOLOv3 (Redmon and Farhadi, 2018), etc., which directly converts the target border localization problems into regression problems, thus enabling thorough

[*] Corresponding author.
E-mail address: zhangyouzhi@wzu.edu.cn (Y. Zhang).

[1] These authors contributed equally to this work and should be considered co-first authors.
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2024.102108
Received 28 March 2024; Received in revised form 13 June 2024; Accepted 17 June 2024
Available online 21 June 2024

end-to-end detection. Nowadays, after a phase of continuous optimization of the algorithms, the YOLO series has also evolved and is used in many fields. Especially in the unmanned vehicle field, YOLO target detection algorithms are widely used as end-user devices in cars. Zaghari et al. (2021) proposed a deep neural network based on driving behavior cloning that combines the YOLO-Fuzzy algorithm and the maximum suppression algorithm for obstacle detection. The method has high accuracy and speed, although there are limitations in dynamic object capture. Dong et al. (2022) proposed a lightweight YOLOv5 model for detecting cars on roads. The model incorporates the Ghost (Han et al., 2020) module and the CBAM attention mechanism, which reduces model computation while improving accuracy by 3.2%. Gu et al. (2023) enhanced the detection of small target vehicles by introducing the SPD-Conv (Sunkara and Luo, 2022), the feature pyramid network structure, as well as the SIoU loss function (Gevorgyan, 2022) in YOLOv5. However, the introduction of the above modules makes the model computationally intensive, and the study does not consider the detection of non-motorized vehicles and pedestrians. The latest YOLOv7 (Wang et al., 2023a) has a faster network structure and more efficient convolutional layers, which is conducive to improving detection speed and accuracy. Wang et al. (2023c) used a modified vehicle target detection algorithm based on YOLOv7 in an intelligent traffic management system, using the MobileNetV3 network with a depth-separable convolutional structure (Howard et al., 2019) in place of the original backbone network and the Focal-EIOU (Zhang et al., 2022) as the loss function. This improved algorithm was shown to be effective in improving vehicle recognition accuracy, but the increase in the number of parameters leads to a decrease in the detection speed. Niu et al. (2023), for the problem of low precision in the vehicle detection task, proposed an approach that adds the SE attention (Squeeze-and-Excitation Attention) mechanism (Hu et al., 2018) and the particular feature fusion module to the YOLOv7 model, which resulted in improved detection capability, but lost the advantage of YOLOv7's original fast detection speed. Therefore, YOLOv7 algorithms with fast detection speed and high accuracy are developed for fast vehicle recognition scenarios.

The visual distance measurement method is obtaining two-dimensional image information of an object through a camera or a camera sensor. Then, it uses computer vision algorithms and geometric derivation to estimate the three-dimensional distance between the object and the camera. The current image-based distance measurement methods can be categorized into monocular detection and binocular detection. In the use of binocular detection, Strbac et al. (2020) used two cameras to obtain stereo vision combined with the YOLOv3 algorithm to estimate the vehicle distance. However, it was experimentally found that the method could only guarantee accuracy within 20 m due to the accuracy defect of the bounding box at long distances. Meanwhile, Liu et al. (2019) pointed out that binocular detection is more challenging in capturing desired features of the image for matching when there is little or no texture in the scene, thus affecting the accuracy. While Brazil and Liu (2019) implemented monocular object detection relying on 3D frame regression, mainly using geometric relations and designing a depth-aware convolutional layer to reduce 2D information into 3D parameters for the distance estimation task, although the method deviates from the aim of high efficiency when using monocular object detection. Vajgl et al. (2022) implemented monocular prediction of objects using the YOLOv3 model distance with high extrapolation speed, but the estimated error value is large, which makes it challenging to meet the actual road detection needs. As mentioned above, there is an urgent need to develop distance measurement models with small measurement biases.

SLAM (Simultaneous Localization and Mapping) (Cadena et al., 2016) technology can support safe obstacle avoidance for self-driving vehicles. ORB-SLAM3 (Campos et al., 2021) is a state-of-the-art visual SLAM system that tracks and maps the environment based on features and uses camera sensors to construct an environment map in real time with high robustness. The instantaneous map construction of ORB-SLAM3 can achieve autonomous localization of the obstacle avoidance module. In addition, realizing autonomous navigation also relies on global path planning algorithms and local path algorithms from universities. The current mainstream local path algorithms include DWA (dynamic window approach) (Saranrittichai et al., 2013), and TEB (Time-ElasticBand). DWA algorithm utilizes the dynamic window strategy to search for the motion state of the mobile robot and the environment-sensing data. TEB algorithm (Rösmann et al., 2012) was initially developed by Rösmann et al. in the context of the Elastic Bands (EB) (Quinlan and Khatib, 1993) method, which locally deforms the path according to the position and shape of obstacles, replans the global path and incorporates time information to improve the motion comfort and efficiency of the mobile robot. Naotunna and Wongratanaphisan (2020) improved the motion comfort and efficiency of the mobile robot by randomly setting up obstacles on a map and comparing the three algorithms, namely DWA, EB, and TEB, with each other. The obstacle avoidance performance of a large differential drive robot under DWA, EB, and TEB proves that TEB outperforms the other two algorithms in terms of the success rate of obstacle avoidance and overall obstacle avoidance performance. Therefore, it is important to consider improving the TEB algorithm to optimize unmanned obstacle avoidance. Kailin et al. (2020) limited the acceleration to a reasonable range by adding acceleration constraints to the TEB algorithm, making the motion smoother. Wu et al. (2021) introduced shortest path constraints for the problem of oversized trajectories during turns in the TEB path algorithm, making the local paths closer to the inner paths and reducing the unnecessary paths. However, this improved strategy increases the probability of hitting obstacles when going around.

It is well known that realizing autonomous driving in places where navigation signals such as GPS are weak or online maps are not covered requires the use of intelligent navigation systems for real-time sensing and decision-making. Many car companies like Tesla and Cruise are actively developing and applying related technologies to realize autonomous driving without GPS (Brenner and Herrmann, 2018). Therefore, the exploration of intelligent navigation in unknown environments has been one of the hotspots of research. Motlagh et al. (2014) first proposed a method of controlling mobile robot navigation autonomously in real dynamic environments using multi-layer feed-forward neural networks, which increases the feasibility of real-time obstacle avoidance through neural network technology. On the other hand, Zhou et al. (2023) investigated the feasibility of wheeled mobile robots in uncertain environments and a hybrid obstacle avoidance method combined with 3D target detection for wheeled mobile robots in uncertain environments. Although this method meets the obstacle avoidance needs of mobile robots, the target detection and obstacle avoidance effects have not yet reached the requirements of real driving.

As summarized above, given the problems with poor accuracy and slow speed of object detection models in vehicle recognition scenarios and the lack of efficient visual distance measurement models with low error and efficient and smooth obstacle avoidance algorithms, this paper develops an intelligent navigation system. Improvements are implemented in three aspects: target detection, distance measurement, and obstacle avoidance, to realize precise and rapid visual detection and efficient automatic obstacle avoidance. The main contributions of this paper are as follows:

- To meet the requirements of intelligent driving for road environment detection with high accuracy and speed, this paper proposes an enhanced YOLOv7x model. The CBAM attention mechanism (Woo et al., 2018) is introduced in the YOLOv7x network to improve detection accuracy by adaptively extracting channel and spatial correlation information of the feature map. Second, the MPDIoU (Siliang and Yong, 2023) loss function is added to improve the detection accuracy and generalization ability.

- Given the low accuracy and inefficiency of the traditional obstacle-ranging model, this paper integrates a single-lens 3D detection module into the target-ranging module. Meanwhile, an optimized monocular ranging algorithm is proposed to eliminate the angular error when the target deviates from the center position and improve the ranging precision.
- A more efficient local path planning algorithm is developed. This algorithm reduces the vehicle's impact force by introducing acceleration jump constraints (Wang et al., 2023b) to eliminate the abnormal changes due to instantaneous speed changes. Additionally, it is combined with steering speed constraints limitations to control the vehicle's deflection speed to ensure smoothness of traveling.

The rest of the paper is organized as follows: Section 2 describes the improved model for target detection and distance measurement in road environments integrated into a visual detection module. Section 3 describes the navigation and obstacle avoidance module for localization and real-time path planning and proposes the improved TEB algorithm. Section 4 presents the methodology and results of the simulation and experiments and gives the experimental conclusions. Finally, the main achievements of the paper are summarized in Section 5.

2. Visual detection module

The visual detection module is divided into two sub-modules: the target detection module and the distance detection module. The target detection module implements an improved YOLOv7 model and is responsible for detecting obstacles in the environment quickly and accurately. The distance detection module calculates the obstacle's distance from the camera based on its position and size information.

2.1. Improved model YOLOv7x-MC

The basic YOLOv7x model suffers from low target detection precision and slow detection speed in the road detection task. Therefore, an improved YOLOv7x algorithm (YOLOv7x-MC) is proposed here to achieve faster and more accurate target detection and recognition. Specifically, based on the original model, the loss function CioU is substituted with the MPDioU loss function with higher convergence speed and accuracy. Secondly, the CBAM attention mechanism is embedded in the YOLOv7x backbone network to improve further the detection model's capability to capture target features. The CBAM module can adaptively extract channel and spatial correlation information from the feature map. Fig. 1 shows the model framework of YOLOv7x-MC, in which the Backbone part replaces one conventional convolutional layer of the original ELAN-1 structure with a CBAM attention layer. The new ELAN structure, ELAN-2, is displayed in the yellow region of Fig. 1.

2.1.1. Basic YOLOv7x model

The model structure diagram of YOLOv7x is shown in Fig. 2. Like YOLOv7, its network structure consists of three parts: Backbone, neck, and head. The structure of each part is described in detail below.

The backbone is made up of CBS, ELAN-1, and MPConv. The CBS comprises a convolutional layer, a batch normalization layer, and a Silu activation function layer for feature extraction (Bai et al., 2023). ELAN-1 is an efficient layer aggregation network that employs a denser residual structure to increase the model depth, improve the accuracy of the feature extraction, and address the gradient vanishing problem. The MPConv module uses both max-pooling and convolution operations in parallel to accomplish downsampling. The neck comprises the spatial pyramid pooling structure SPPCSPC (He et al., 2015) and the PAFPN (Liu et al., 2018) structure. SPPCSPC fuses multiscale features through maximum pooling of different receptive fields. The PAFPN structure consists of the FPN and the PANet. The FPN implements top-down transfer of the semantic information of the deeper feature maps, and the PANet implements bottom-up transfer of the lower feature map position, texture, and other detail information. The final PAFPN structure can output feature maps at three different scales. In the head, the RepConv module applies the reparameterized convolution to the residual module, by fusing multiple feature maps obtained in training into a single feature map. Large, medium, and small targets can be predicted by detecting the head.

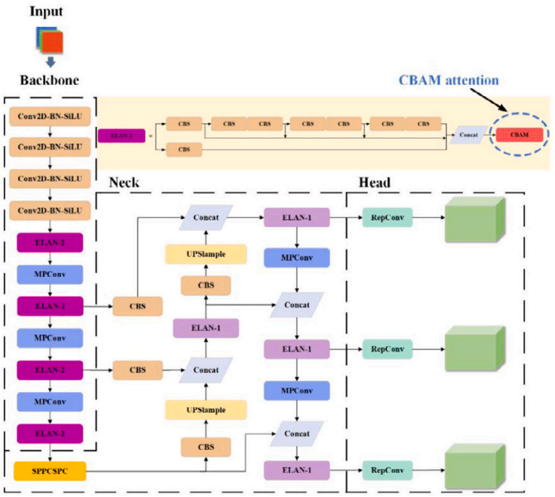

Fig. 1. Structure of the YOLOv7x-MC model.

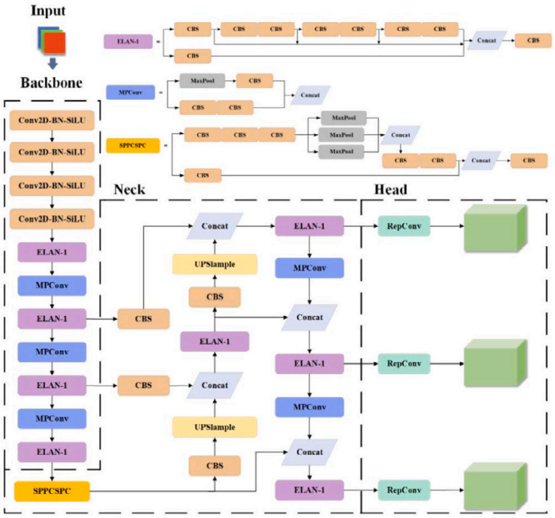

Fig. 2. Structure of the YOLOv7x model.

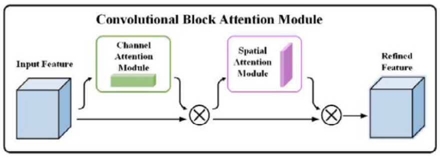

Fig. 3. CBAM structure diagram.

2.1.2. CBAM attention mechanism

As a lightweight module, CBAM contains two basic sub-modules, the Channel Attention Module (CAM) and the Spatial Attention Module (SAM), respectively, to emphasize channel and spatial information. Its module structure is shown in Fig. 3 (Woo et al., 2018)

In CBAM, the input feature map first passes through the channel attention module to generate the channel attention map M_C, then multiplies by F to get the intermediate output result F'. Next, the result is passed through the spatial attention module to produce the spatial attention map M_S. Finally, the final feature mapping result F'' is obtained. Eq. (1) represents the operation of the whole process, in which ⊗ denotes element-by-element multiplication.

$$\begin{aligned} F' &= M_C(F) \otimes F \\ F'' &= M_S\left(F'\right) \otimes F' \end{aligned} \tag{1}$$

The operation principle of the channel attention module is as follows: first, the feature maps are subjected to average-pooling and max-pooling operations based totally on width and height. Then, the two generated results are transferred to the multi-layer perceptron (MLP), where the two outputs of the MLP are added element-by-element and finally processed by the Sigmoid activation function to produce the channel attention map. On the other hand, the spatial attention module operates as follows: first, the feature map is applied with average pooling and maximum pooling down the channel axis, then series concatenation and convolution are performed on the generated results, and eventually, the spatial attention map is processed by the Sigmoid activation function to produce the spatial attention map (Woo et al., 2018). The specific mathematical expressions of the channel and spatial attention modules are presented in Eqs. (2) and (3), respectively.

$$M_C(F) = \sigma(\mathrm{MLP}(\mathrm{Avgpool}(F)) + MLP(\mathrm{Maxpool}(F))) \tag{2}$$

$$M_S = \sigma\left(f^{7\times7}([\mathrm{AvgPool}(F); \mathrm{MaxPool}(F)])\right) \tag{3}$$

where σ denotes the sigmoid function and $f^{7\times7}$ denotes the 7 × 7 convolution operation. In response to the problem of low target detection precision due to dense road vehicles and high occlusion rates, CBAM will be used to replace the three conventional convolutional operations in the backbone network. The introduction of the CBAM attention mechanism allows for a more accurate focus on regions of the channel and space that contain important information and restrains the interference of unrelated information, ultimately improving the overall accuracy of detection.

2.1.3. MPDioU loss function

In the target detection task, the loss function measures the gap between the predicted and expected information of the neural network. And if the gap is smaller, the value of the loss function is smaller. The main loss of the YOLO model contains three kinds: localization loss, confidence loss, and classification loss. In the YOLOv7 model, the CIOU function is utilized to calculate the localization loss. It is based on the original Intersection over Union (IoU), which considers the centroid aspect ratio and distance between the predicted and ground truth boxes (Liu et al., 2019). However, the CIOU function suffers from limited convergence speed and high computational cost. This paper uses the minimum partial distance intersection over the union (MPDioU) loss function (Siliang and Yong, 2023). This loss function is mainly designed to achieve the fast regression of the bounding box by minimizing the distance of the upper-left and lower-right points between the prediction box and the ground truth box, thus simplifying the calculation. The specific mathematical expression is as follows:

$$d_1^2 = \left(x_1^B - x_1^A\right)^2 + \left(y_1^B - y_1^A\right)^2 \tag{4}$$

$$d_2^2 = \left(x_2^B - x_2^A\right)^2 + \left(y_2^B - y_2^A\right)^2 \tag{5}$$

$$\mathrm{MPDioU} \;= \frac{A \cap B}{A \cup B} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2} \tag{6}$$

$$\mathcal{L}_{\mathrm{MOPDioU}} \;= 1 - \;\mathrm{MPDioU} \tag{7}$$

In the formula, A and B refer to the ground truth box and the predicted box, respectively. $\left(x_1^A, y_1^A\right)$ and $\left(x_2^A, y_2^A\right)$ denote the coordinates points in the upper left and lower right corners of the truth box, while $\left(x_1^B, y_1^B\right)$ and $\left(x_2^B, y_2^B\right)$ represent the coordinates of the predicted box. d_1 and d_2 denote the distances of the corresponding points between the two boxes, respectively. $\mathcal{L}_{\mathrm{MOPDioU}}$ denotes the value of the MPDIoU-based loss function. MPDIoU combines the advantages of various loss functions while also increasing the convergence speed and accuracy, which contributes to the overall detection performance of the YOLO model.

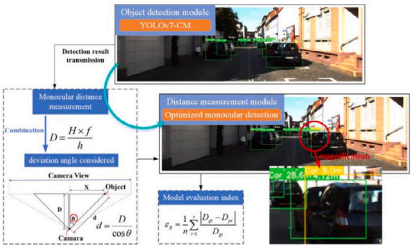
Object detection module
YOLOv7-CM
Detection result transmission
Monocular distance measurement
Combination
$D=\frac{H\times f}{h}$
deviation angle considered
Camera View
X
Object
D
θ
d
$d=\frac{D}{\cos\theta}$
Camara
Distance measurement module
Optimized monocular detection
magnification
Car 28.6
Car 8.0m
Model evaluation index
$\varepsilon_R=\frac{1}{n}\sum_{i=1}^{n}\frac{|D_{gt}-D_{pr}|}{D_{gt}}$

Fig. 4. Ranging module framework.

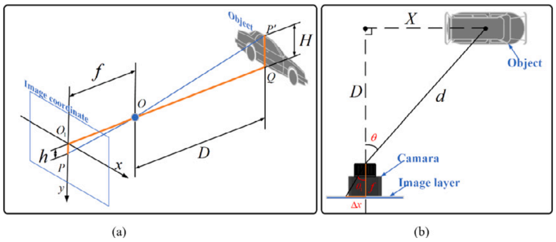

Fig. 5. Schematic diagram of the distance measurement method, including (a) Monocular detection method, (b) Consider deviation angle.

2.2. Ranging module based on improved monocular detection model

The ranging module is an essential part of intelligent driving, and it is responsible for precisely sensing the distance of surrounding vehicles and pedestrians. In this paper, an optimized monocular range detection method is proposed, which considers the object's deviation angle relative to the center axis, aiming to reduce the ranging error to adapt to the complex road environment. The specific flow based on the optimized monocular detection approach is depicted in Fig. 4. Firstly, the target detection module transmits the detection results to the optimized monocular detection model, and then, after the operations of monocular ranging and deviation angle correction, the visual detection image with estimated distance is finally output by the ranging module. Meanwhile, this paper also designs evaluation indexes to assess the accuracy of the improved monocular detection model.

2.2.1. Monocular detection model

The monocular ranging method mainly utilizes the ranging module in combination with the detection boxes to estimate the distance using the size and position information of the target in the image, as shown in Fig. 5(a). The relationship between the camera, target object, and image plane can be established according to the similarity property of triangles. Assuming that the camera-to-object distance is D, the camera focal length is f, the true height of the object is H, and the pixel height of the object within the image is h, the following relation can be derived (Chen et al., 2016):

$$D = \frac{H \times f}{h} \tag{8}$$

2.2.2. Improved ranging model considering deviation angle

In traditional monocular ranging methods, the similarity triangle principle is often applied to estimate the distance of an object. However, when the object deviates from the center of the image, the horizontal deviation of the object from the center of the field of view is not considered, which can lead to ranging errors. To improve the accuracy of range, the angle of deviation between the object and the center axis is considered, and a ranging model that corrects the error using the angle of deviation is proposed. Specifically, as shown in Fig. 5(b), when the target vehicle is to the right of the centerline, the actual camera-to-object distance d will be larger than the estimated distance D due to the existence of a horizontal distance X between the object and the center point.

Eq. (9) can be used to calculate the actual distance, where Δx denotes the horizontal distance between the object and the camera's image layer centerline, and f refers to the camera focal length. However, in practice, the lens shape of the camera and the internal parameters of the camera may cause lens aberration, resulting in a certain deviation between the off-center angle θ in the world coordinates and the off-center angle θ_1 in the image, which can be corrected by introducing the parameter w. In the end, a more accurate result can be obtained with the help of the off-center angle of the object in the image when calculating the actual distance d.

$$\begin{cases} \theta_1 = \arctan\left(\frac{\Delta x}{f}\right) = \arctan\left(\frac{x_c - x_0}{f}\right) \\ d = \frac{D}{\cos\theta} = \frac{D}{\cos\lambda\theta_1} \end{cases} \tag{9}$$

2.2.3. Ranging model evaluation metrics

To assess the level of distance estimation, the mean relative distance error ε_R (Vajgl et al., 2022) can be used, which is defined as follows:

$$\varepsilon_R = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|D_{gt} - D_{pr}\right|}{D_{gt}} \tag{10}$$

where n is the number of detection frames and D_{gt}, D_{pr} denote the true value and predicted distance respectively. The true distance values can be obtained by labeling the dataset, while the predicted distance values are estimated by the monocular ranging method described above.

3. Navigation and obstacle avoidance module

The main tasks of the navigation and obstacle avoidance module are divided into ORB-SLAM3 environment modeling and path planning. The first part is to update the ORB-SLAM3 environment map by combining the detected target objects and their estimated position and distance information with the camera's position information; the second part is to perform local path planning based on the constructed real-time map information.

3.1. ORB-SLAM3 framework

ORB-SLAM3 is a feature point-based visual SLAM system aiming to achieve real-time environment mapping and autonomous localization. Fig. 6 shows the framework of ORB-SLAM3, which mainly includes a tracking phase, a local map-building phase, a closed-loop detection and fusion phase, and a global BA phase. The specific implementation steps are: ORB-SLAM3 receives image frames from the camera and performs ORB feature point extraction, preliminary tracking and local map tracking for each frame to get the initial position estimation; then updates the map points by keyframe insertion and culling, and performs the local BA (Bundle Adjustment) to complete the local map building; then utilizes the position recognition algorithm for detecting closed loops and complete loop correction and map fusion (Mur-Artal and Tardós, 2014); finally, global BA optimization is performed on all keyframes and map points to obtain a map containing the desired information.

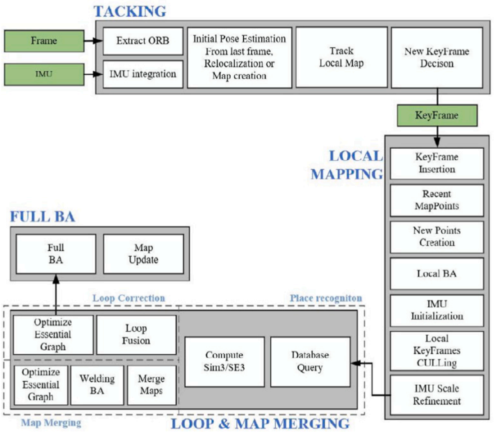

Fig. 6. Schematic diagram of the main structure of ORB-SLAM3.

3.2. Improved TEB local path planning algorithm

The traditional TEB algorithm allows a mobile robot to move according to its dynamics and avoid obstacles through the constraints of various objective functions. However, the TEB algorithm suffers from sudden changes in acceleration and repeated oscillations in velocity during obstacle avoidance and even needs to complete obstacle bypassing through backward operation, which significantly affects the steering efficiency. For this reason, this paper proposes an improved algorithm, TEB-S (TEB-Smooth), which introduces acceleration jump constraints and steering speed constraints to ensure that the acceleration varies stably outside the safe distance and the steering speed is controlled within a reasonable interval, which ultimately achieves the purpose of smooth obstacle avoidance. The TEB algorithm and the improvement strategy are described below.

3.2.1. Traditional TEB algorithm

The timed elastic band (TEB) algorithm transforms a series of positional trajectory information of a mobile robot into a time-invariant elastic band model, which is mainly used to deal with the path planning of local dynamic obstacles. The TEB algorithm is described by n intermediate robot positions $S_i = \left[x_i, y_i, \theta_i\right]^{\mathrm{T}} \in R \times S$, where $x_i, y_i \in R$ denote the planar position, and $\theta_i \in S$ denotes the direction of travel. The TEB is extended by time intervals between neighboring positions, thus generating $n-1$ time differences ΔT_i. In each time difference, the mobile robot changes from the previous position to another, as shown in Fig. 7. Therefore, its motion trajectory model contains position information and time transition information, and the trajectory vector B is represented as follows:

$$B = \left[S_1, \Delta T_1, S_2, \Delta T_2, S_3, \dots, \Delta T_{n-1}, S_n\right] \tag{11}$$

The key idea in the TEB algorithm is to obtain the optimal path through multi-objective optimization of the trajectory, and the specific objective function is expressed as follows:

$$f(B) = \sum_k \gamma_k f_k(B) \tag{12}$$

$$B^* = \arg\min_B f(B) \tag{13}$$

where $f_K(B)$ and γ_K denote each function and the weights of each constraint function, respectively. $f(B)$ represents the objective function considering various constraints, and B^* refers to the optimal trajectory (Rösmann et al., 2012).

The objective function constraints for TEB generally fall into two categories. The first category relates to path analysis, including shortest path constraints and obstacle spacing constraints; the second category

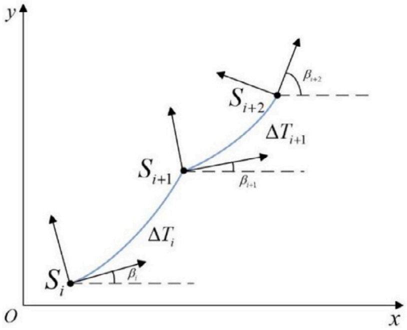

Fig. 7. Relationship between TEB trajectory position and time interval.

considers constraints on the motion characteristics of the mobile robot, such as speed and acceleration. The above constraints can be formulated in the TEB algorithm as a segmented, continuously differentiable penalty function, denoted as follows:

$$e_{\Gamma}\left(x, x_r, \varepsilon, S, n\right) \approx \begin{cases} \left(\frac{x-(x-\varepsilon)}{S}\right)^n & x > x_r - \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where x_r is the critical value, b denotes the small offset allowed in the determination, and c and d are the scaling ratio and order, respectively, used to influence and adjust the function value.

(1) Path point target and obstacle constraints

TEB has similar constraint functions for intermediate path point acquisition and obstacle avoidance for global path planning. The difference is that path points attract elastic bands while obstacles repel them. Combined with the penalty function (Eq. (14)), they are expressed as follows:

$$f_{\text{path}} = e_{\Gamma}\left(d_{\min,j}, r_{p_{\max}}, \varepsilon, S, n\right) \tag{15}$$

$$f_{ob} = e_{\Gamma}\left(-d_{\min,j}, r_{o_{\min}}, \varepsilon, S, n\right) \tag{16}$$

(2) Velocity and acceleration constraints

The velocity and acceleration constraints of the mobile robot are realized by a penalty function similar to the geometric constraints (Wu et al., 2021). Based on the Euclidean distance between neighboring poses, angular changes and time intervals, their linear and angular velocities can be calculated:

$$\begin{cases} v_i \approx \frac{1}{\Delta T_i}\left\|\begin{pmatrix} x_{i+1} & -x_i \\ y_{i+1} & -y_i \end{pmatrix}\right\| \\ \omega_i \approx \frac{\theta_{i+1}-\theta_i}{\Delta T_i} \end{cases} \tag{17}$$

The corresponding acceleration is also derived through the velocity:

$$\begin{cases} a = \frac{2(v_{i+1}-v_i)}{\Delta T_i+\Delta T_{i+1}} \\ \alpha = \frac{2(\omega_{i+1}-\omega_i)}{\Delta T_i+\Delta T_{i+1}} \end{cases} \tag{18}$$

(3) Non-holonomic kinematic constraints

Robots with differential drive configurations have two degrees of freedom of movement and cannot translate along the Y-axis. This motion constraint leads to forming a smooth path segment consisting of circular arcs, thus requiring two neighboring configurations to be on a common circular arc of constant curvature. It is also necessary that the turning angles of the neighboring positions $S_i S_{i+1}$ must be equal, β_i denotes the specific position of the mobile robot in the coordinate system, and $d_{i,i+1}$ denotes the direction vector of the neighboring positions.

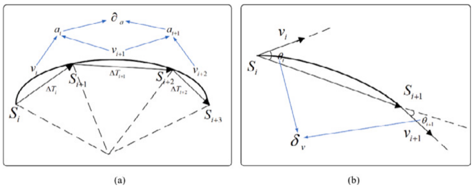

Fig. 8. Schematic diagram of enhanced constraints, including (a) acceleration jump constraint, (b) steering speed constraint.

Accordingly, the corresponding non-holonomic kinematic constraints can be obtained:

$$f\left(S_i, S_{i+1}\right) = \left\|\left[\begin{pmatrix} \cos\beta_i \\ \sin\beta_i \\ 0 \end{pmatrix}\right] + \left[\begin{pmatrix} \cos\beta_{i+1} \\ \sin\beta_{i+1} \\ 0 \end{pmatrix}\right] \times d_{i,i+1}\right\| \tag{19}$$

(4) Fastest path constraints

TEB adds the time interval variable to achieve the fastest path constraint by minimizing the time interval between all robot poses (Wang et al., 2023b), which is described by the mathematical formula:

$$f_k = \left(\sum_{i=1}^{n} \Delta T_i\right)^2 \tag{20}$$

3.2.2. *Acceleration jump constraints*

During the moving process, the traditional TEB algorithm has a problem in that its acceleration change rate is too large when decelerating or accelerating, which may lead to the occurrence of undesirable conditions such as sudden stops or sudden acceleration. Such undesirable conditions greatly impact the robot and cause a waste of time and resources. To overcome this problem, this paper introduces the acceleration jump factor ∂_a to improve the TEB algorithm, which is called TEB-a algorithm. The specific implementation method is as follows: when the distance to the obstacles is detected to be greater than the safe distance, the acceleration value is adjusted according to the current acceleration and the given acceleration change factor. When the distance is equal to or smaller than the safety distance, the TEB-a algorithm sets the current acceleration value to the desired acceleration value to realize the effect of instantaneous braking. The acceleration jump control factor is related to the kinematic characteristics of each of the robot's bit poses, and Fig. 8(a) presents the schematic diagram of the implementation of the algorithm. By selecting the position $S_k(k = i, i+1, i+2, i+3)$ in a period of time and combining Eqs. (17), (18), and (21), the acceleration jump constraint factor ∂_a in this time period can be calculated.

$$\partial_a = \frac{\rho \times \left|a_{i+1} - a_i\right|}{0.25\Delta T_i + 0.5\Delta T_{i+1} + 0.25\Delta T_{i+2}} \tag{21}$$

where ρ is a deterministic parameter in the range between $(0.6, 0.8)$. Within the set safety distance, the acceleration jump constraint factor will control the acceleration to change stably within a small range, thereby suppressing the phenomenon of unnecessary sudden acceleration changes and improving the smoothness and safety of the robot's motion.

3.2.3. *Steering speed constraints*

In the traditional TEB algorithm, the mobile robot often has unstable steering during obstacle avoidance due to the excessive steering speed. It has a long turning path with low turning efficiency. For this reason, the steering speed constraint factor δ_v can be introduced, and the improved algorithm TEBv can be realized by setting appropriate weights to trade off with the original speed penalty function. The TEB-v algorithm can adjust the turning speed adaptively to avoid

Table 1
Summary of experimental parameters.

Experiment one	Model	Batch size	Interval	Epoch	Interval
	YOLOv7x	8	$(1, 64)$	300	/
	YOLOv7x+MPDioU	8	$(1, 64)$	300	/
	YOLOv7x+CBAM	8	$(1, 64)$	300	/
	YOLOv7x-CM	8	$(1, 64)$	300	/
Experiment two	Model	λ	Interval	f (in pixels)	Interval
	General model	/	/	721.538	/
	Optimized model	1.45	$(1, 2)$	721.538	/
Experiment three	Algorithm	Deterministic parameter ρ	Interval	Preset start angle	Interval
	TEB	/	/	/	/
	TEB-a	0.65	$(0.6, 0.8)$	/	/
	TEB-v	/	/	$\pi/14$	$[0, \pi]$
	TEB-S	0.65	$(0.6, 0.8)$	$\pi/14$	$[0, \pi]$

obstacles smoothly. The mathematical expression for the steering speed constraint factor δ_v is as follows:

$$\delta_v = \begin{cases} 1 - \frac{\Delta\theta}{\pi} & \Delta\theta \geq \theta_{\text{set}} \\ 1 & \text{otherwise} \end{cases} \tag{22}$$

The principle of steering speed constraint is shown in Fig. 8(b), $\Delta\theta$ is the difference between the velocity offset angle of the $(i+1)$th pose S_{i+1} and the velocity offset angle θ_{i+1} of the ith pose S_i of the mobile robot, and its value can vary from $[0, \pi]$. Where θ_{set} is the preset start angle, only when $\Delta\theta$ is greater than the preset start angle the steering speed constraint factor δ_v will work to ensure the steering efficiency. The actual speed is calculated as follows:

$$v'_{i+1} = v_{i+1} \times \delta_v \tag{23}$$

According to Eqs. (22) and (23), when the path curvature of the mobile robot increases, $\Delta\theta$ increases and δ_v decreases, resulting in a lower actual speed. This means that the more the robot deviates from the straight line when it turns, the more the steering speed constraint factor δ_v can control the speed decrease, thus ensuring the smoothness and efficiency of the turn. At the same time, this can also, to some extent, reduce the risk of lateral sliding and robot loss of control.

The entire algorithm adjusts the acceleration and velocity based on a_jump _factor and steer_v_factor. The Update process first obtains the distance to the obstacle detected by the sensor. If the distance is greater than the safe distance, the Adjust process is called iteratively to modify the acceleration and velocity until the distance is less than or equal to the safe distance. During the Adjust process, the desired acceleration and velocity are computed, and adjustments to these values are made based on the input parameters a_jump_factor and steer_v_factor, resulting in the return of the adjusted outcomes. Subsequently, the AdjustAcceleration process determines the final acceleration value by comparing the desired acceleration with the current acceleration, while the AdjustVelocity process adjusts the final velocity based on the desired velocity, current velocity, and the timestep. In addition, the AdjustVelocityByDeviation process modifies the velocity value according to the deviation angle and the input parameter steer_v_factor.

Algorithm 1 Improved TEB Algorithm

```
procedure UPDATE($a\_jump\_factor, steer\_v\_factor$)
    $detected\_distance \leftarrow$ GetDetectedDistance
    while $detected\_distance > safe\_distance$ do
        $(a, v) \leftarrow$ Adjust($a\_jump\_factor, steer\_v\_factor, v$)
        $detected\_distance \leftarrow$ GetDetectedDistance
    end while
end procedure
function ADJUST($a\_jump\_factor, steer\_v\_factor, v$)
    $desired\_a \leftarrow$ CalculateDesiredAcceleration()
    $a \leftarrow$ AdjustAcceleration($a\_jump\_factor, desired\_a$)
    $desired\_v \leftarrow$ CalculateDesiredVelocity()
    $v \leftarrow$ AdjustVelocity($a, desired\_v, v$)
    $v \leftarrow$ AdjustVelocityByDeviation($v, steer\_v\_factor$)
    return $(a, v)$
end function
function ADJUSTACCELERATION($a\_jump\_factor, desired\_a$)
    if $desired\_a > a + a\_jump\_factor$ then
        return $a + a\_jump\_factor$
    else if $desired\_a < a - a\_jump\_factor$ then
        return $a - a\_jump\_factor$
    else
        return $desired\_a$
    end if
end function
function ADJUSTVELOCITY($a, desired\_v, v$)
    if $desired\_v > v + timestep \times a$ then
        return $v + timestep \times a$
    else if $desired\_v < v - timestep \times a$ then
        return $v - timestep \times a$
    else
        return $desired\_v$
    end if
end function
function ADJUSTVELOCITYBYDEVIATION($v, steer\_v\_factor$)
    $v\_deviation\_angle \leftarrow$ CalculateDeviationAngle()
    if $|v\_deviation\_angle| > preset\_start\_angle$ then
        return $v \times steer\_v\_factor$
    else
        return $v$
    end if
end function
function GETDETECTEDDISTANCE
    sensor.measureDistance()
    return min_distance
end function
```

4. Simulation and experimentation

4.1. Parameter setting

The experimental section contains three experiments: target detection experiment, distance measurement experiment, and obstacle avoidance experiment. Various algorithms or models are tested in the experiments, and the related parameters and their respective values or ranges are presented in Table 1.

4.2. Experimental environment

The KITTI 3D Object Detection Evaluation 2017 dataset is used in this paper, which includes 7481 training images and 7518 test images. These images mainly contain road scenes in urban areas, villages, and highways, and the dataset is labeled for three types of objects in the scene: cars, pedestrians, and bicycles. In this experiment, 2700 images were selected, including as many vehicles and pedestrians in urban areas and highways as possible, to approximate the real scene of vehicle driving. The image pixel size is 1242 × 375. Meanwhile, the training and validation were carried out using an 8:2 ratio, and the AdamW optimizer (Yao et al., 2021) was used in each model. The software and hardware environments used for model training are listed in Table 2.

Table 2
Experiment environment.

Configuration	Models and versions
Operating system	Window 11
CPU	16 GB RAM, Intel i9-10900K processor
GPU	NVIDIA GeForce GTX 3080

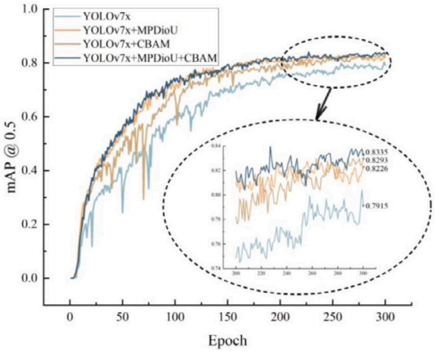

Fig. 9. mAP @ 0.5-Epoch diagram.

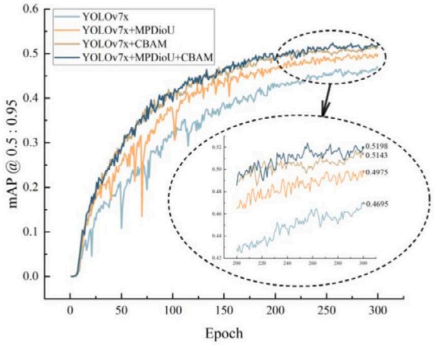

Fig. 10. mAP @ 0.5:0.95-Epoch diagram.

4.3. Target detection experiment

The improved yolov7x algorithm proposed by this paper mainly introduces the MPDIoU loss function and the CBAM attention mechanism. In this paper, ablation experiments are conducted on the KITTI dataset to verify the positive impact of the proposed improved strategy on YOLOv7x. Figs. 9 and 10 show the comparison results of each model under mAP @ 0.5 and mAP @ 0.5:0.95 indicators, respectively. Specifically, mAP @ 0.5 indicates the average AP accuracy when the IoU threshold is 0.5 and mAP @ 0.5:0.95 indicates the average AP accuracy at different thresholds (the IoU threshold ranges from 0.5 to 0.95, and the step is 0.05). As can be seen from the figure, these two indicators gradually improve as the number of training rounds increases.

The results of the ablation experiments are summarized in Table 3. Compared with the original YOLOv7 model, the YOLOv7-CM model, which changes the loss function to MPDioU and adds CBAM, significantly improves different indicators. Regarding accuracy indicators, mAP @ 0.5 and mAP @ 0.5:0.95 improved 5.3% and 10.7%, respectively. This demonstrates that by using the MPDIoU loss function and the CBAM attention mechanism, the improved model can recognize road obstacles more accurately. As for detection speed, the FPS (frames per second) of the YOLOv7-CM model is improved by 35.4%, which illustrates that the improved method also outperforms YOLOv7x in terms of detection speed. It is worth noting that these performance enhancements do not cause a significant increase in the number of parameters and FLOPs.

Table 3
Results of the ablation experiment.

Model	mAP @ 0.5	mAP @ 0.5 : 0.95	Parameters (M)	FLOPs (G)	FPS
YOLOv7x	0.7915	0.4695	70.8	188.0	46.3
YOLOv7x+MPDioU	0.8226	0.4975	70.8	188.0	47.1
YOLOv7x+CBAM	0.8293	0.5143	71.2	195.1	62.6
YOLOv7x+MPDioU+CBAM	**0.8335**	**0.5198**	**71.2**	**209.6**	**64.3**

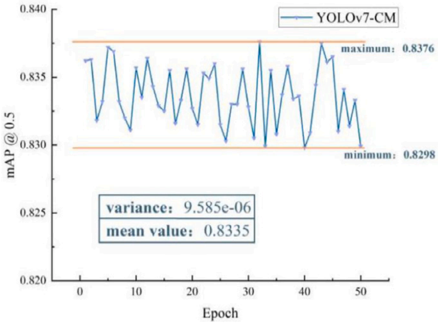

Fig. 11. Statistical plot of mAP @ 0.5.

Next, the improved model was trained and tested 50 times to demonstrate the performance under statistical significance. As shown in Fig. 11, the results of mAP @ 0.5 after each training session were recorded. As can be seen from the figure, the variance of the mAP @ 0.5 metrics of YOLOv7x-CM is minimal in a large number of training, the maximum difference is also small, and the results are all able to maintain a high accuracy, which demonstrates the high stability and reliability of the model.

In addition to experiments on the KITTI dataset, we also used the BDD100K test set to evaluate the generalization ability of the YOLOv7x-CM algorithm in complex traffic scenarios. The BDD100K dataset, as one of the largest and most diverse autopilot datasets available today, contains a total of 100,000 images with different weather, scene, time of day, as well as high-definition and blurred images. For increased transparency and reproducibility of the study, we have uploaded our full dataset to the Zenodo platform https://zenodo.org/. This dataset contains all the specific data used for this study. Interested readers can access and view the dataset using the following digital object identifier (DOI): [10.5281/zenodo.11522908]. To provide further validation regarding the detection performance of the YOLOv7-CM model, other state-of-the-art target detection algorithms were also similarly evaluated on the KITTI dataset and BDD100K dataset, including SSD, YOLOv5s, and YOLOv8s algorithms, as shown in Table 4. The validation results on both datasets indicate that YOLOv7x-CM outperforms the other algorithms in terms of detection accuracy and FPS. Meanwhile, experiments on the BDD100K dataset exhibit a 6.8% improvement in mAP @ 0.5 for YOLOv7x-CM compared to YOLOv7x.

To investigate the detection ability of the model in more complex driving scenarios, the experiments classify typical scenes based on weather (e.g., snowy, clear, and rainy days) and time attributes (e.g., daytime and night). Fig. 12 illustrates the experimental results of advanced target detection algorithms such as YOLOv7x-CM under both attribute classifications. The experimental data show that the target detection capability of YOLOv7x-CM is optimal in all specific attributes. Especially under some bad weather conditions, such as cloudy, snowy, rainy, and foggy days, YOLOv7x-CM is able to maintain strong detection accuracy, while other algorithms have a relatively large decrease in detection ability. Also at night with poor visual conditions, our proposed model achieves high detection accuracy. This is due to the fact that the CBAM attention mechanism in YOLOv7x-CM is able to effectively extract road information and exclude the

Table 4
Comparison of experimental results of detection algorithms.

Model	mAP @ 0.5 (KITTI)	mAP @ 0.5 (BDD)	Parameters (M)	FPS
SSD (Liu et al., 2016)	0.7471	0.7626	193.8	37.5
YOLOv5s	0.7656	0.7819	7.9	58.4
YOLOv5s+CBAM	0.7722	0.7898	11.3	59.0
YOLOv5s+CBAM+EioU	0.7726	0.7883	11.3	61.1
YOLOv8s	0.7944	0.8114	12.1	63.1
YOLOv8s+CBAM	0.8172	0.8205	15.4	62.2
YOLOv8s+CBAM+EioU	0.8179	0.8242	15.4	62.3
YOLOv7x	0.7915	0.8049	70.8	46.3
SE-Lightweight YOLO (Niu et al., 2023)	0.8056	0.8183	36.9	61.9
YOLOv7x-CM	**0.8335**	**0.8594**	**71.2**	**64.3**

Table 5
Comparison of experimental results of detection algorithms.

Model	mAP @ 0.5 (KITTI)	mAP @ 0.5 (BDD)	Parameters (M)	FPS
SSD (Liu et al., 2016)	0.7471	0.7626	193.8	37.5
YOLOv5s	0.7656	0.7819	7.9	58.4
YOLOv5s+CBAM	0.7722	0.7898	11.3	59.0
YOLOv5s+CBAM+EioU	0.7726	0.7883	11.3	61.1
YOLOv8s	0.7944	0.8114	12.1	63.1
YOLOv8s+CBAM	0.8172	0.8205	15.4	62.2
YOLOv8s+CBAM+EioU	0.8179	0.8242	15.4	62.3
YOLOv7x	0.7915	0.8049	70.8	46.3
SE-Lightweight YOLO (Niu et al., 2023)	0.8056	0.8183	36.9	61.9
YOLOv7x-CM	**0.8335**	**0.8594**	**71.2**	**64.3**

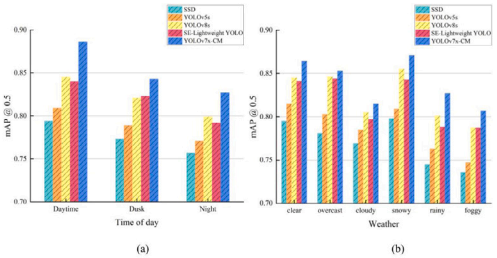

Fig. 12. Comparison of experimental results in different attribute scenarios (BDD100K dataset), including (a) Experimental data in different times, (b) Experimental data in different weather.

interference of irrelevant factors, thus satisfying the need for accurate obstacle recognition in complex scenes.

4.4. Distance measurement experiment

The KITTI 3D Object Detection Evaluation 2017 dataset is also used in the distance measurement experiments, from which 50 images are selected. The corresponding labels contain information such as their position in the image, distance, and relative rotation angle, from which real distance value can be obtained for model quality assessment. The designed ranging module is deployed in the target detection module, and the horizontal focal length $f = 721.538$ (in pixels) is set. The ranging module is then detected on the dataset. Fig. 13 shows a comparison of some of the detections of the general model for monocular detection and the off-center angle adjustment model.

From Fig. 13, it can be seen that the predicted values of the optimized model are closer to the real values than the general model. The model still maintains a high accuracy estimation at long distances. The statistical calculation results of the average relative distance error for all the distance data are shown in Table 5.

As can be seen from Table 5, the average relative distance error of the general model is between 7–9 times compared with the optimized model. The maximum estimation distance of the optimized model is more than 90 m. In addition, in the presence of long-distance targets, the optimized model can control the maximum error within 4.5 m, which indicates that the distance measurement module proposed in this paper is highly robust. On top of that, the optimized model has no significant changes in the model parameters and processing image

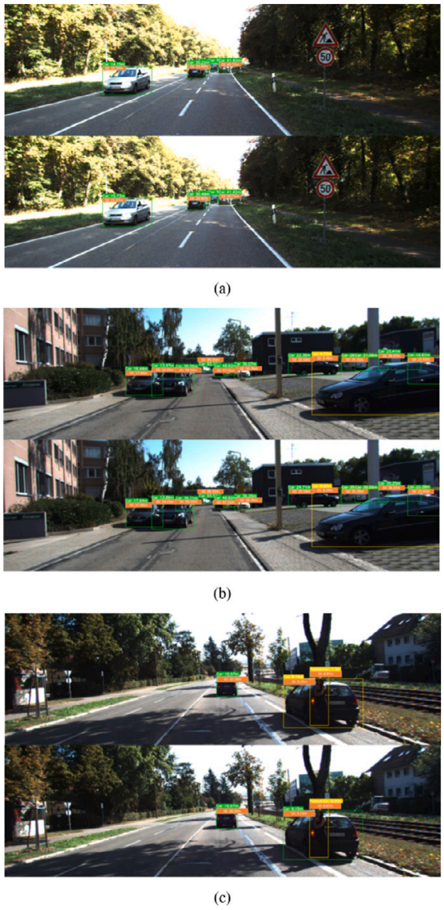

Fig. 13. Comparison of distance detection results,including (a) Detection Figure I, (b) Detection comparison Figure II, (c) Detection comparison Figure III.

Table 6
Parameter configuration of local planners.

Parameter	Value
Maximum linear velocity (m/s)	0.4
Maximum angular velocity (rad/s)	0.5
Maximum X linear acceleration $(\mathrm{m/s^2})$	1.0
Maximum angular acceleration $(\mathrm{rad/s^2})$	0.5
Minimum distance to the obstacle (m)	0.25

speed, which means that it improves the ranging accuracy without increasing the complexity of the model.

In this paper, it is difficult to compare the results directly with other advanced monocular ranging models due to different experimental standards. Therefore the results of other studies are presented below for reference. Zhu and Fang (2019) achieved the metric ε_R of 0.25 by validating the model on the KITTI dataset.

Vajgl et al. (2022) tested it on the KITTI dataset using Dist-YOLO. The results showed that the mean relative distance error for all three types of targets, car, cyclist and pedestrian, was 0.11. Yu and Choi (2021) designed the distance measurement model based on YOLO v4 and evaluated it on the KITTI dataset, and finally achieved about 0.037 relative distance error.

4.5. Obstacle avoidance experiments

The obstacle avoidance experiment mainly uses gazebo and RVIZ software to build and test the environment in the ORB-SLAM3 framework. The other configurations are the same as in Table 2 (Experiment environment) except that the operating system is the Ubuntu18.04 system. The algorithm simulation is carried out with the help of a two-wheeled differential mobile robot, and the initial parameter configurations for the simulation are shown in Table 6.

The obstacle avoidance experiment builds maps based on four specific environments, and multiple local path planning algorithms (DWA, TEB, TEB-a, TEB-v, TEB-S) combined with global path planning algorithms are sequentially simulated in the map scenarios to test their obstacle avoidance performance.

Fig. 14 records the simulation in 4 scenarios in order. Taking Fig. 14(a) as an example, SLAM first builds a real-time environment map based on the image information provided by visual detection, as shown in the bottom left picture. Then, the motion endpoint is issued to the mobile robot on the map, and it will first become the path generated by the global planning algorithm (green solid line in the map). During traveling, the robot also has local path planning for obstacles within range to adjust the global planning path. To visualize the path under the local planning algorithm, horizontal and vertical coordinates (X, Y) were collected every 20 ms during the motion of the mobile robot, from which the paths from the start point to the endpoint are plotted. Different algorithms were tested 15 times in each scenario, and the optimal path was selected to generate the final comparison graph, as shown in the figure below.

It can be seen from the path comparison diagram that the length of the moving path under the DWA algorithm is larger than that under the TEB algorithm, and the bending radius during obstacle avoidance is too large, resulting in more resource waste. Comparing each TEB algorithm in the four scenarios, it can be concluded that the path lengths of TEB-v and TEB-S are shorter and more efficient, especially in obstacle avoidance, the path of TEB-a is the next best, and the original TEB is relatively the worst.

The traveling time of each algorithm in the four scenarios is recorded while keeping the moving target point constant. The algorithms in each scenario were tested ten times for averaging, and a time comparison graph was made, as shown in Fig. 15.

As can be seen from the time comparison graphs, the running times of the DWA algorithms are all close to twice the time of the unimproved TEB algorithm. In addition, according to the size of time consumed, the improved TEB algorithms are ranked from largest to smallest, such as TEB, TEB-a, TEBv, and TEB-S. Combining the results of the path length and travel time, it can be concluded that in the obstacle avoidance experiments, the TEB-S algorithm performs the best in terms of resource wastage as well as time efficiency, followed by the TEB-v algorithm and then the TEB-a algorithm, whereas the traditional TEB algorithm comes last.

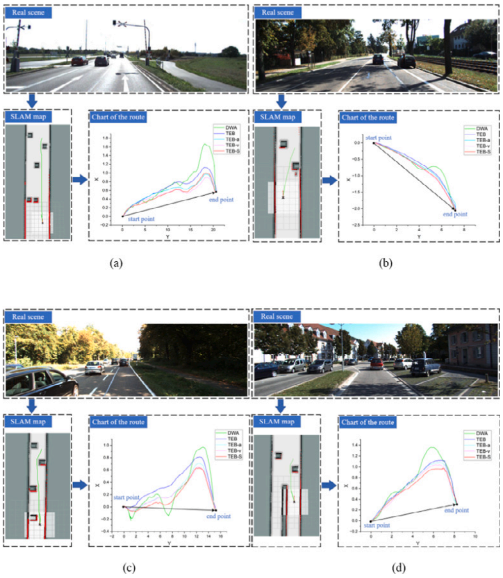

Fig. 14. Comparison of algorithms in simulation scenes, including (a) Scene I, (b) Scene II, (c) Scene III, (d) Scene IV.

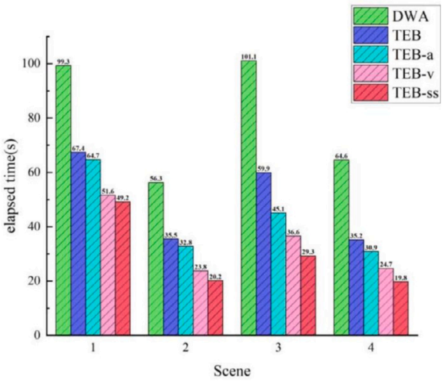

Fig. 15. Elapsed time comparison chart.

To study the sensitivity of the optimized obstacle avoidance algorithm to parameter changes in the enhancement constraint parameters, several experiments were conducted for each of the four scenarios described above. Specifically, keeping other conditions constant, individual parameters in the acceleration jump and steering speed constraints were varied multiple times, respectively, and changes in elapsed time were observed. First, the deterministic parameter ρ in the acceleration jump constraint was varied from 0.5 to 0.8 in steps of 0.05, and Ten tests were performed, with the mean value of elapsed time recorded and calculated. Secondly, the preset start angle in the turn speed constraint then varied regularly, and the average value of elapsed time was obtained for the same various parameters. Through the statistics and analysis, a graph of the average elapsed time versus the parameter values can be obtained, which is presented in Fig. 16. From the figure, it can be seen that in each scenario, the variation of the two parameters

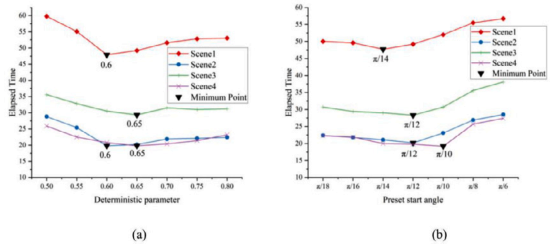

Fig. 16. Single parameter sensitivity analysis diagram, including (a) Deterministic parameters analysis diagram, (b) Preset start angle analysis diagram.

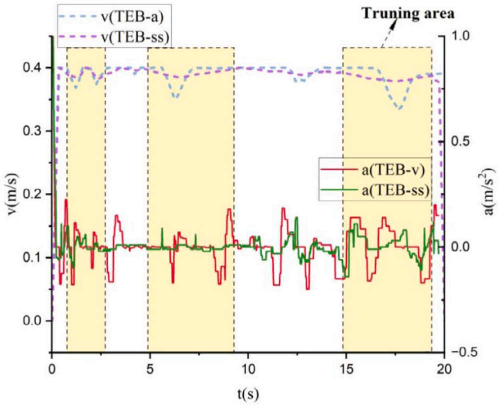

Fig. 17. v-t and a-t diagram.

in some range of values causes fluctuations in the values of elapsed time, indicating that the parameter values have high importance for the output results. However, it can be found that the metrics of the deterministic parameters ρ and preset start angle can remain stable when they are within

the value range of $(0.6, 0.8)$ and $(\pi/18, \pi/10)$, respectively. Therefore, the results of this sensitivity analysis can be used to optimize the algorithm and facilitate the search for optimal input variable values.

Next, the improvement of the improved algorithm on the motion characteristics and its ability to react to the dynamically changing road environment will be investigated. For this purpose, Scene IV is chosen, where two obstacles are dynamically added to the original robot's motion path. In this process, the velocity data of the mobile robot under the TEB-a and TEB-S algorithms over time were recorded. The velocity–time relationship curve (dashed line) was plotted out as in Fig. 17. Subsequently, the change in the speed of the mobile robot motion guided by the TEB-v and TEB-S algorithms was similarly recorded. The acceleration-time curve (solid line) in Fig. 17 was plotted by extracting the curvature of the velocity–time plot.

From the velocity–time curves, it can be concluded that the velocity changes of the TEB-S algorithm are smoother compared to the TEB-a algorithm, especially in the turning area. It can be verified that the steering speed constraint has a positive effect on the turning performance of the vehicle. Meanwhile, the analysis of the acceleration-time curves shows that the overall acceleration jump of the TEB-S algorithm is significantly lower than that of the TEB-v algorithm during multiple steering processes. This result indicates that the acceleration jump constraint has improved in reducing unnecessary acceleration jumps, allowing the vehicle to drive more smoothly. In addition, from the overall motion characteristics, the improved algorithm TEB-S is able to avoid obstacles that suddenly appear smoothly and quickly, which is in line with the obstacle avoidance response to the real-time dynamic environment.

5. Conclusion

To meet the demand for speed and accuracy of intelligent navigation systems, this paper develops an intelligent navigation system that implements improvements in three aspects: target detection, distance measurement, and obstacle avoidance, to ensure that the vehicle can sense and keep away from obstacles in the road environment quickly and accurately, thus guaranteeing the safety of driving. The main achievements are summarized as follows: for the target detection module, the YOLOv7x-CM model is proposed based on the YOLOv7x model, while the CBAM attention mechanism and MPDioU loss function are introduced, aiming to increase the accuracy and speed of target detection. Through the validation of the KITTI dataset, the outcomes show that compared to YOLOv7x, YOLOv7x-CM improves the accuracy evaluation metrics mAP @ 0.5 and mAP @ 0.5:0.95 by 5.9% and 9.5% respectively, while the FPS by 35.4%. Meanwhile, the validation on the BDD100K dataset indicates that YOLOv7x-CM's mAP @ 0.5 has increased by 6.8% compared to the pre-improvement period. Furthermore, classification experiments based on weather and time attributes on a BDD100K dataset demonstrate the excellent detection capability of YOLOv7x-CM in complex driving scenarios. In the distance measurement module, the monocular vision ranging model improves the ranging accuracy by eliminating the off-angle error relative to the center axis. Compared to the traditional monocular detection method, this method significantly reduces the average relative distance error without an increase in model complexity. This paper adopts the ORB-SLAM3 framework for the obstacle avoidance module to construct the environment map in real-time and proposes an improved TEB algorithm named TEB-S algorithm. TEB-S algorithm considers the acceleration jump constraints and steering speed constraints, which can reduce the impacts due to the sudden changes in the acceleration, improve the driving stability and efficiency, and make the steering smoother. The simulation experiments compare the time, path, acceleration, and speed changes of local path planner algorithms in the road scenario, which proves that TEB-S has stronger computational performance. In addition, it is experimentally found that the efficiency of the obstacle avoidance algorithm is stabilized at a high level when the specific controllable parameters in the enhanced constraint are kept within appropriate intervals, respectively. Implementing the above improvement strategies can significantly improve the vehicle's reaction speed to obstacles, and efficient and safe obstacle avoidance and navigation can be 0realized. Although the current improvements have been able to meet the system's needs for accuracy and speed, there is still improvement yet to be made. For example, future work could consider optimizing obstacle avoidance for different kinds of vehicles or exploring the overall performance of the range module when combining monocular ranging with sensors such as radar.

CRediT authorship contribution statement

Zihao Xu: Writing – original draft, Software, Methodology, Conceptualization. **Yinghao Meng:** Writing – original draft, Visualization, Methodology, Conceptualization. **Zhen Yin:** Visualization, Methodology, Formal analysis. **Bowen Liu:** Visualization, Software, Resources. **Youzhi Zhang:** Writing – review & editing, Funding acquisition. **Mengmeng Lin:** Investigation, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by China Postdoctoral Science Foundation (2022M722681), Natural Science Foundation of Zhejiang Provincial, China (LY24E050005), Wenzhou Major Technology Innovation Project, China (ZG2023014), Wenzhou Industrial Science and Technology Project, China (G20210004), Wenzhou Association for Science and Technology, China (jczc32).

References

Bai, W., Zhao, J., Dai, C., Zhang, H., Zhao, L., Ji, Z., Ganchev, I., 2023. Two novel models for traffic sign detection based on YOLOv5s. J. Axioms 12 (2), 160.

Brazil, G., Liu, X., 2019. M3d-rpn: Monocular 3d region proposal network for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9287–9296.

Brenner, W., Herrmann, A., 2018. An overview of technology, benefits and impact of automated and autonomous driving on the automotive industry. J. Digit. Marketpl. Unleashed 427–442.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J., 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. J. IEEE Trans. Robot. 32 (6), 1309–1332.

Campos, C., Elvira, R., Rodríguez, J., Montiel, J., Tardós, J., 2021. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Trans. Robot. 37 (6), 1874–1890.

Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R., 2016. Monocular 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2147–2156.

Dong, X., Yan, S., Duan, C., 2022. A lightweight vehicles detection network model based on YOLOv5. J. Eng. Appl. Artif. Intell. 113, 104914.

Gevorgyan, Z., 2022. SIoU loss: More powerful learning for bounding box regression. J. arXiv preprint arXiv:2205.12740.

Girshick, R., 2015. Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587.

Gu, Z., Zhu, K., You, S., 2023. YOLO-SSFS: A method combining SPD-conv/STDL/IM-FPN/SIoU for outdoor small target vehicle detection. J. Electron. 12 (18), 3744.

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C., 2020. Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1580–1589.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37 (9), 1904–1916.

Howard, A., Sandler, M., Chu, G., Chen, L., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q., 2019. Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1314–1324.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7132–7141.

Kailin, Z., Baoling, H., Xinda, W., 2020. Ackerman robot motion planning system based on improved TEB algorithm. J. Sci. Technol. Eng. 20 (10), 3997–4003.

Large, F., Laugier, C., Shiller, Z., 2005. Navigation among moving obstacles using the NLVO: Principles and applications to intelligent vehicles. J. Auton. Robot 19, 159–171.

Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A., 2016. Ssd: Single shot multibox detector. In: Proceedings of the Computer Vision–ECCV 2016: The 14th European Conference. pp. 21–37.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8759–8768.

Liu, F., Zhou, S., Wang, Y., Hou, G., Sun, Z., Tan, T., 2019. Binocular light-field: Imaging theory and occlusion-robust depth perception application. J. IEEE Trans. Image Process. 29, 1628–1640.

Lu, Y., Ma, H., Smart, E., Yu, H., 2021. Real-time performance-focused localization techniques for autonomous vehicle: A review. J. IEEE Trans. Intell. Transp. Syst. 23 (7), 6082–6100.

Motlagh, O., Nakhaeinia, D., Tang, S., Karasfi, B., Khaksar, W., 2014. Automatic navigation of mobile robots in unknown environments. J. Neural Comput. Appl. 24, 1569–1581.

Mur-Artal, R., Tardós, J., 2014. Fast relocalisation and loop closing in keyframe-based SLAM. In: Proceedings of the 2014 IEEE International Conference on Robotics and Automation. pp. 846–853.

Naotunna, I., Wongratanaphisan, T., 2020. Comparison of ros local planners with differential drive heavy robotic system. In: Proceedings of the 2020 International Conference on Advanced Mechatronic Systems. pp. 1–6.

Niu, C., Song, Y., Zhao, X., 2023. SE-lightweight YOLO: Higher accuracy in YOLO detection for vehicle inspection. J. Appl. Sci. 13 (24), 13052.

Pang, G., Takahashi, K., Yokota, T., Takenaga, H., 2002. Intelligent route selection for in-vehicle navigation systems. J. Transp. Plan. Technol. 25 (3), 175–213.

Quinlan, S., Khatib, O., 1993. Elastic bands: Connecting path planning and control. In: Proceedings of the IEEE International Conference on Robotics and Automation. pp. 802–807.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.

Redmon, J., Farhadi, A., 2017. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7263–7271.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. J. arXiv preprint arXiv:1804.02767.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39 (6), 1137–1149.

Rösmann, C., Feiten, W., Wösch, T., Hoffmann, F., Bertram, T., 2012. Trajectory modification considering dynamic constraints of autonomous robots. In: Proceedings of the ROBOTIK 2012; 7th German Conference on Robotics. pp. 1–6.

Saranrittichai, P., Niparnan, N., Sudsang, A., 2013. Robust local obstacle avoidance for mobile robot based on dynamic window approach. In: Proceedings of the 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. pp. 1–4.

Saranya, M., Archana, N., Reshma, J., Sangeetha, S., Varalakshmi, M., 2022. Object detection and lane changing for self driving car using cnn. In: 2022 International Conference on Communication, Computing and Internet of Things. pp. 1–7.

Siliang, M., Yong, X., 2023. Mpdiou: A loss for efficient and accurate bounding box regression. J. arXiv preprint arXiv:2307.07662.

Strbac, B., Gostovic, M., Lukac, Z., Samardzija, D., 2020. YOLO multi-camera object detection and distance estimation. In: Proceedings of the 2020 Zooming Innovation in Consumer Technologies Conference. pp. 26–30.

Sunkara, R., Luo, T., 2022. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 443–459.

Vajgl, M., Hurtik, P., Nejezchleba, T., 2022. Dist-YOLO: Fast object detection with distance estimation. J. Appl. Sci. 12 (3), 1354.

Wang, C., Bochkovskiy, A., Liao, H., 2023a. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7464–7475.

Wang, Z., Li, P., Li, Q., Wang, Z., Li, Z., 2023b. Motion planning method for car-like autonomous mobile robots in dynamic obstacle environments. J. IEEE Access 11, 137387–137400.

Wang, Y., Tian, Y., Cheng, J., Meng, X., Xie, Z., 2023c. An improved YOLOv7 method for vehicle detection in traffic scenes. In: Proceedings of the 2023 35th Chinese Control and Decision Conference. pp. 766–771.

Woo, S., Park, J., Lee, J., Kweon, I., 2018. Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision. pp. 3–19.

Wu, J., Ma, X., Peng, T., Wang, H., 2021. An improved timed elastic band (TEB) algorithm of autonomous ground vehicle (AGV) in complex environment. J. Sens. 21 (24), 8312.

Yao, Z., Gholami, A., Shen, S., Mustafa, M., Keutzer, K., Mahoney, M., 2021. Adahessian: An adaptive second order optimizer for machine learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 10665–10673.

Yu, J., Choi, H., 2021. YOLO MDE: Object detection with monocular depth estimation. Electronics 11 (1), 76.

Zaghari, N., Fathy, M., Jameii, S., Shahverdy, M., 2021. The improvement in obstacle detection in autonomous vehicles using yolo non-maximum suppression fuzzy algorithm. J. J. Supercomput. 77 (11), 13421–13446.

Zhang, X., Gao, H., Xie, G., Gao, B., Li, D., 2017. Technology and application of intelligent driving based on visual perception. J. CAAI Trans. Intell. Technol. 2 (3), 126–132.

Zhang, Y., Ren, W., Zhang, Z., Jia, Z., Wang, L., Tan, T., 2022. Focal and efficient IOU loss for accurate bounding box regression. J. Neurocomput. 506, 146–157.

Zhou, H., Feng, P., Chou, W., 2023. A hybrid obstacle avoidance method for mobile robot navigation in unstructured environment. J. Ind. Robot 50 (1), 94–106.

Zhu, J., Fang, Y., 2019. Learning object-specific distance from a monocular image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3839–3848.