

Article

A Low-Latency Mobile Robot Target Following Method Based on Improved YOLO-World

Yanlong Sun ^{1,2} , Kai Miao ^{1,2} , Mingxi Zhang ^{1,2}, Rixing Zhu ^{1,2} and Shougang Huang ^{1,2,*}

¹ School of Traffic and Transportation, Shijiazhuang Tiedao University, Shijiazhuang 050043, China; syl@stdu.edu.cn (Y.S.); 18330012573@163.com (K.M.); 15720332733@163.com (M.Z.); zhurixing@stdu.edu.cn (R.Z.)

² Hebei Key Laboratory of Traffic Safety and Control, Shijiazhuang 050043, China

* Correspondence: huangshg@stdu.edu.cn

Abstract

This paper addresses the challenges of high latency and the lack of an effective recovery strategy in mobile robot target following tasks. In this paper, a low-latency mobile robot target tracking method based on the improved YOLO-World algorithm is proposed. The process primarily consists of three parts: target detection, target tracking, and motion control. First, for target detection, we introduce a tailored lightweight backbone network, GSS, within the YOLO-World framework, which progressively expands the receptive field through cascaded convolutional operations and enhances cross-group feature interaction via a channel mixing mechanism, significantly improving model efficiency with minimal loss in detection accuracy. Additionally, depthwise separable convolution is applied to the detection head to reduce computational redundancy. Secondly, in the target tracking part, a lightweight target tracking algorithm based on improved BoT-SORT is adopted, and the tracking delay is effectively reduced by optimizing the ReID feature extraction backbone network. Then, the motion control part adopts an active search strategy based on visual servo control. When the tracked target is lost, the strategy utilizes a camera motion compensation-based tracker to predict the target motion state and controls the robot to actively search for the target accordingly. Subsequently, feature tracking is resumed through target re-recognition, thus re-establishing target following. Experiments on public datasets and real-world scenarios demonstrate that the proposed method achieves strong robustness and real-time performance.

Keywords: mobile robot; YOLO-world; lightweight backbone; target following method; low latency



Academic Editor: Víctor Casamayor-Pujol

Received: 15 April 2026

Revised: 28 May 2026

Accepted: 18 June 2026

Published: 30 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Mobile robots deployed in security, logistics, and medical-care [1,2] applications rely on real-time target following as a fundamental capability. This task involves three tightly coupled subproblems that together determine system performance: target detection [3] must be fast enough to meet real-time constraints while preserving accuracy in open-vocabulary scenarios; target tracking must maintain low re-identification latency so that the robot can continuously update the target state; and an effective recovery strategy is needed when the target leaves the robot's [4] field of view. Figure 1 illustrates the overall architecture proposed in this paper, which addresses all three subproblems jointly [5].

The YOLO [6–8] (You Only Look Once) family of detectors has evolved through successive versions—YOLOv1 through YOLOv8—progressively improving speed and accuracy

through architectural innovations such as the C2f backbone, decoupled detection heads, and anchor-free regression. YOLO-World [9] extends YOLOv8 with an open-vocabulary detection capability by coupling image features with text embeddings via a RepVL-PAN (Re-parameterizable Vision-Language Path Aggregation Network) neck, enabling category-agnostic detection without retraining. Because of its strong trade-off between accuracy and real-time performance, YOLO-World is selected as the baseline detection model in this work. YOLO-World significantly improves the model's target detection ability in open-vocabulary scenarios; however, such large models usually have a large number of parameters and complex structures, resulting in high computational complexity [10]. When they are deployed in mobile robots to perform real-time target following tasks, they usually exhibit high latency and cannot meet the real-time requirements.

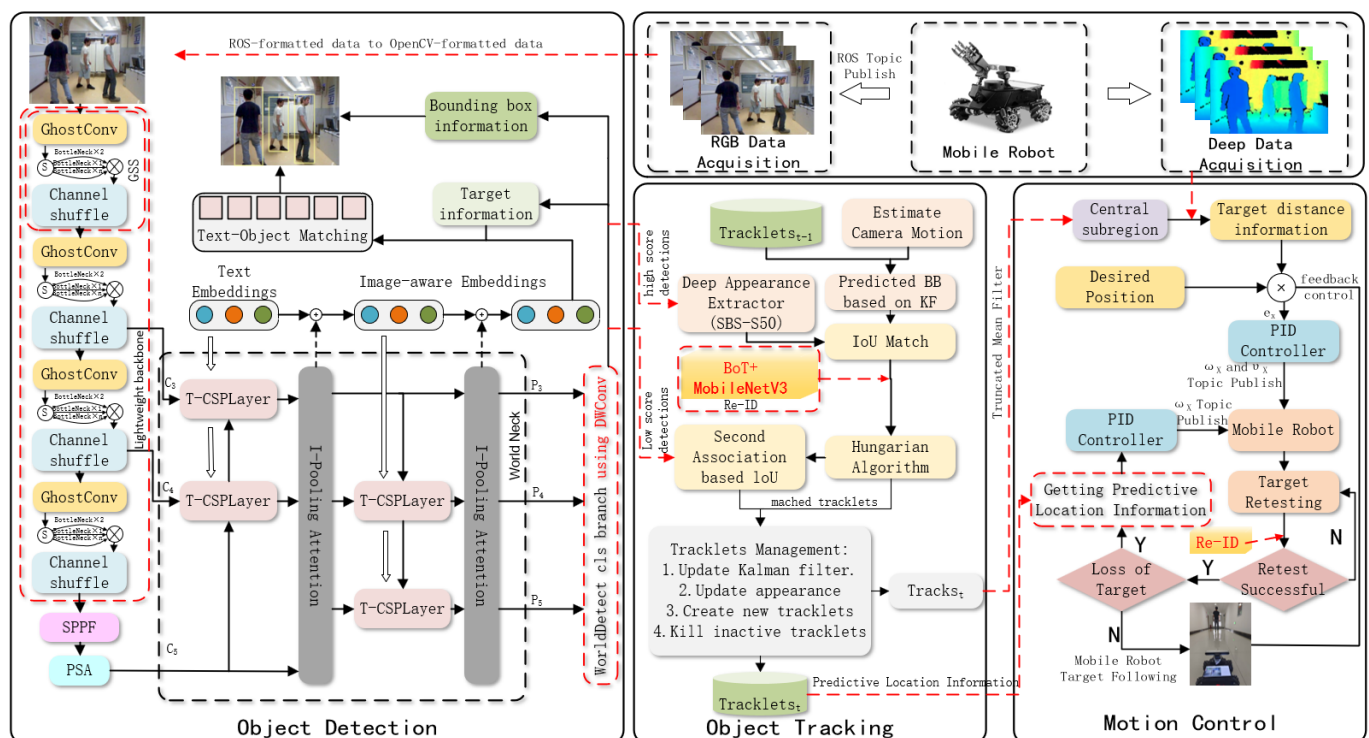


Figure 1. A low-latency mobile robot target following method based on improved YOLO-World.

Therefore, researchers have lightened the network by rational combinations between modules, introducing depth-separable convolution and attention mechanisms. For example, He et al. [11] introduced the ECA attention mechanism into the YOLOv4-Tiny backbone network and optimized the SPP structure by null convolution to achieve lightweighting. Zhou et al. [12], on the other hand, replaced the backbone network of YOLOv5 with the improved ShuffleNetV2 [13] network and embedded the SE attention module with the Inception structure to propose the SA YOLO model to realize the lightweight of the backbone network. Although the above method realizes the lightweight of the deep learning network and reduces the network complexity and the number of exorbitant parameters, the loss of target detection accuracy is large, and the task of performing the target following method for mobile robots requires high target detection accuracy.

In order for a mobile machine to follow a target stably, in addition to the target detector, the target tracker also plays an important role. The target tracker obtains information about the position of the target. Simple online and real-time time tracking (SORT) [14] is an adaptive vision servo controller that updates depth information in an unknown environment. A traditional method, a modified version of which is called DeepSORT [15], is usually used to track multiple targets using detection-by-detection tracking. DeepSORT and related

detection-by-tracking methods have since been extended in numerous ways; several of these extensions, such as the Chain Tracker (CTracker) [16] and FairMOT [17], have demonstrated high accuracy in complex multi-object environments. Recently, transformer-based tracking algorithms—such as TransTrack and MOTR—have demonstrated high accuracy and modeling efficiency by exploiting self-attention to model long-range dependencies between target queries across frames. However, their elevated computational cost and large memory footprint make real-time deployment on resource-constrained mobile robots challenging, which motivates the lightweight design proposed in this paper. Although the above studies have achieved some results in target recognition and tracking, the algorithms used in them require high computer arithmetic power, and the modules occupy relatively large spatial memory and have high latency when performing the task of mobile robot target following methods.

The target following task requires a short response time and stable following of the robot's motion control, and commonly used motion control methods include PID control [18], which is simple to implement and tune but sensitive to sudden target-state changes; sliding mode control, which is robust to disturbances but may produce high-frequency chattering; and fuzzy control, which handles nonlinearities well but requires expert rule design. In this paper, a PID-based visual servo controller is adopted because of its low computational overhead and ease of parameter tuning, which is well suited to the real-time requirements of the mobile robot platform. However, due to the limited field of view of the mobile robot, which leads to the loss of the target, the target tracker can extract the feature information of the lost target of the mobile robot, but there is no further target following search strategy after the loss. Several vision-based approaches have been proposed to improve following robustness, yet none provide an active mechanism to search for and re-acquire a lost target. Motokucho et al. [19] designed a vision-based power-assisted wheelchair, capturing the pedestrian's lower limbs through the camera and utilizing the optic-flow method to track the pedestrian leg information that is moving. The camera is mounted on the wheelchair, the lower limbs of the pedestrians are collected by the camera, and the optical flow method is used to track the information of the pedestrian's legs in motion and determine the relative position between the pedestrian target and the wheelchair, and then real-time following is accomplished by visual feedback. Harish et al. [20] designed an improved recursive neural network and computational vision to track the colored target, predict the linear and angular velocity of the tracked target, and at the same time, use a PID controller to control the following motion of the mobile robot. However, these vision-based mobile robot target following methods only provide motion control of the mobile robot without considering the active search and recovery of the lost target following by the mobile robot after the target is lost.

In summary, two open problems motivate this paper. First, existing mobile robot following systems incur high end-to-end latency in the detection and re-identification stages, which prevents the robot from responding promptly to rapid target movements. Second, no effective strategy is available to recover following after the target leaves the camera's field of view. To address both problems, this paper proposes a low-latency mobile robot target following method based on an improved YOLO-World, designed to minimize pipeline latency while also providing an active search and re-acquisition strategy.

1. In order to enhance the open-vocabulary target detection capability of the target detection network and reduce the target detection latency, a lightweight target detection algorithm based on the improved YOLO-world is proposed, based on the YOLO-world network, and the lightweight network backbone GSS(GhostNet-Split-Shuffle) is proposed under its framework to replace the original model backbone. The model efficiency is achieved by the cross-layer feature reuse mechanism and the multi-scale

feature fusion strategy with 2.5% detection accuracy. The model efficiency is realized with 2.5% detection accuracy through the cross-layer feature reuse mechanism and multi-scale feature fusion strategy. In addition, the detection head is lightweighted by replacing its 3×3 standard convolutions with 3×3 depth-wise separable convolutions—decomposing each convolution into a depth-wise step followed by a 1×1 point-wise step. The depth-separable convolution is used to reduce the significant computational redundancy and realize a more efficient architecture. In the target tracking section, a lightweight target tracking algorithm based on improved BoT-SORT is used to reduce the re-recognition delay in target tracking by improving the ReID feature extraction backbone to MobileNetV3.

2. Aiming at the problem of lack of effective active search mechanisms when the target exceeds the limited field of view (FOV) of the mobile robot leading to the loss of tracking, this paper adopts an active search strategy based on visual servo control. When the target is lost, the strategy utilizes a camera motion compensation-based tracker to predict the target motion state and accordingly controls the robot to actively search the predicted target position; feature tracking is resumed through target re-recognition, thus re-establishing and maintaining stable target following.
3. Finally, qualitative and quantitative experiments are carried out on public datasets, mobile robot open-vocabulary target following experiments are completed in real scenarios, and the experimental results all verify that the proposed method has good robustness and real-time performance.

This paper is organized as follows: Section 1 introduces the low-latency mobile robot target following method with open-vocabulary detection capability, including the target detection part, target tracking part, and motion control part; Section 2 is the experimental validation, briefly describing the mobile robot and the deep learning platform and qualitatively and quantitatively analyzing the low-latency mobile robot target following method based on the improved YOLO-world, and conducting the open-vocabulary target following experiments for mobile robots; Section 3 summarizes the work of this paper and gives an outlook.

2. A Low-Latency Mobile Robot Target Following Method Based on Improved YOLO-World

2.1. Model General Structure and System Overview

The mobile robot target following method has been realized by a deep learning method. The goal of the mobile robot target following method is to maintain a standard distance with a specific target in an unknown and complex environment, to follow the target by obtaining target or environment information through the visual perception system, and then to adopt a motion control strategy, so as to maintain the relative position relationship with the following target. The mobile robot target following method mainly includes three parts, target detection, target tracking and motion control, in which the target detection part is the basis for the mobile robot to acquire external image information and obtain the target detection frame through the target detection algorithm. The YOLO-world target detection algorithm is a regression-based single-stage detection algorithm, which has excellent performance when dealing with the target detection task in some normal scenes but high latency when performing the mobile robot target following task. Secondly, target tracking is the key for mobile robots to track the target stably, but there is a problem of high latency when re-recognizing the target. Finally, mobile robots usually lose the target due to the limited FOV field of view when performing target following and cannot actively search for the lost target and resume following, and therefore, to address the above problems, this paper selects the YOLO-world model as the baseline model and proposes a low-latency

mobile robot target following method based on the improved YOLO-world to meet the practical needs of mobile robot target following methods.

In order to make the mobile robot target following method achieve low latency and open-vocabulary detection, the low-latency mobile robot target following method based on improved YOLO-world is based on the YOLO-world framework in the target detection part, proposes a new type of lightweight backbone GSS to replace the original backbone, improves the efficiency through multi-stage feature optimization to ensure accuracy while improving the efficiency, and adopts depth-separable convolution for the detection head to further lighten the weight. The depth-separable convolution is used for the detection head to further lighten the weight. Then, in the target tracking part, based on the improved BoT-SORT algorithm, the ReID feature extraction backbone is optimized to reduce the re-identification delay. Finally, in the motion control section, when the target is lost, the robot is controlled to actively search for it based on the predicted target motion state and recover the following using the re-identified features. The structure of the low-latency mobile robot target following method based on improved YOLO-world is shown in Figure 1.

2.2. Lightweight Target Detection Algorithm Based on Improved YOLO-World

Target detection is the key problem of mobile robot target following, and the accuracy of target detection largely determines the stability of mobile robot following. In order to enable the target detection network to detect arbitrary object categories specified at inference time without retraining, and to reduce detection latency, a lightweight detection algorithm based on improved YOLO-World is proposed. The backbone network uses the GSS lightweight backbone network proposed in this paper, and the neck network adopts the YOLO-world architecture and then improves the lightweight for the YOLO-world detection head to get the detection results.

2.2.1. Lightweight Network Backbone GSS

With the rapid development of the Internet, many existing target detection models have high computational complexity and a large number of parameters that often lead to high-latency problems when the model performs the target detection task, which cannot meet the actual demand, while the existing lightweight target detection models reduce the computational complexity and the number of parameters to a certain extent but lead to a large loss of model detection accuracy, which cannot achieve a balance between model efficiency and detection accuracy. Detection accuracy cannot be achieved.

The YOLO-world algorithm is extended from the YOLOv8 algorithm, and the main part is the same as the YOLOv8 algorithm. As a representative work of target detection models in recent years, the main part adopts the C2f structure with richer gradient flow, which improves the performance of feature extraction and target detection. However, there are still problems such as large model size and slow inference speed. Therefore, in order to realize real-time target detection for mobile robots, this paper designs the GSS lightweight architecture, and the structure of the GSS lightweight architecture is shown in Figure 2.

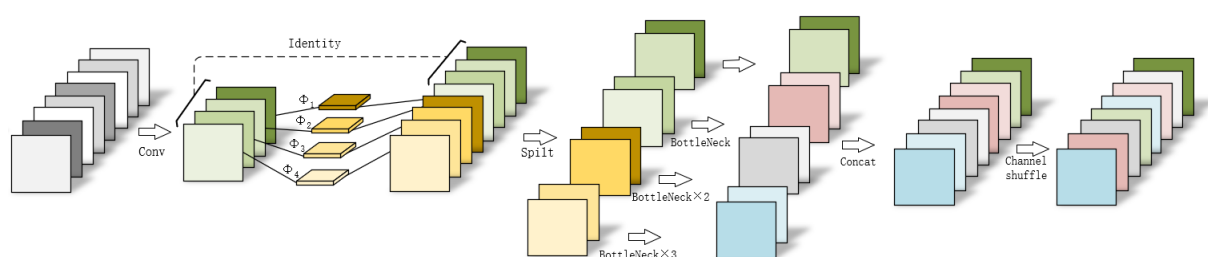


Figure 2. Lightweight network backbone, GSS.

The GSS lightweight architecture first adopts the GhostNet module for primary feature extraction. The Ghost module [21] reduces the model parameters through parameter sharing, adopts the linear operation Φ instead of the original convolution operation, and at the same time, through the sharing of weights, different convolution layers can exchange information with each other so as to improve the representation ability of the model. The output features are then divided into multiple groups of heterogeneous sub-features by a feature grouping strategy, which slices the c channels of the input feature map into n branches with the same number of channels. Subsequently, cascaded progressive convolution kernels are applied to each branch feature for multi-scale feature enhancement, and the progressive expansion of the sensory field is realized through cascaded stacking convolution operations. After the tensor splicing is completed by the feature reorganization layer, a channel blending mechanism is introduced to enhance cross-group feature interaction.

The GSS lightweight architecture processes the input feature map in four steps, assuming that the resolution of the input image is $H \times W$ (H is the height of the image, W is the width of the image). The input feature map is denoted as $F \in R^{H \times W \times C}$, where H is the height, W is the width, and C is the number of channels.

Step 1. Compress the features via 1×1 ordinary convolution in the GhostNet module to generate the feature map $F_{1 \times 1} \in R^{H \times W \times (C/2)}$.

Step 2. Generate the feature map $F_{1 \times 1} \in R^{H \times W \times (C/2)}$ by applying a 3×3 depth-separable convolution to each channel to generate the feature map $F_{3 \times 3} \in R^{H' \times W' \times (C/2)}$ and concatenate it with $F_{1 \times 1} \in R^{H \times W \times (C/2)}$ to generate the feature map $F'_{3 \times 3} \in R^{H' \times W' \times C}$.

Step 3. The feature map $F'_{3 \times 3} \in R^{H' \times W' \times C}$ undergoes a Split operation to equally divide it into n groups of feature maps with the same number of channels $F'_{(3 \times 3)/n} \in R^{H' \times W' \times (C/n)}$.

Step 4. For n groups of feature maps, incrementally insert BottleNeck blocks for different levels of sign extraction to generate n groups of feature maps with different levels of information extraction and concatenate them to generate feature maps $F'_{3 \times 3} \in R^{H'' \times W'' \times C'}$, and let $g = C/n$ be the number of channels in each of the n groups of feature maps.

Step 5. The feature map after shuffling is obtained by expanding the feature map into a four-dimensional matrix of $g \times n \times W \times H$, transposing it along the g - and n -axes of the matrix of dimensions $g \times n \times W \times H$, and tiling the g - and n -axes to obtain the feature map after shuffling $F' \in R^{H'' \times W'' \times C'}$.

The architecture synergistically optimizes shallow detail features and deep semantic features through a cross-layer feature reuse mechanism and multi-scale feature fusion strategy, which effectively enhances the information correlation between feature channels and maintains the detection accuracy comparable to the baseline model while significantly reducing the model complexity.

2.2.2. YOLO-World-Based Neck Network and Lightweight Detection Head

YOLO-world is based on the YOLOv8 model with a greatly improved neck network and detection head. The neck architecture adopts the structure of RepVL-PAN following top-down and bottom-up paths and builds a feature pyramid P3, P4, P5 by multi-scale image features C3, C4, C5. And text-guided CSPLayer (T-CSPLayer) and Image Pooling Attention (I-Pooling Attention) are used to further enhance the interaction between image features and text features, so as to improve the visual semantic representation of open vocabulary. And the detection head part maps multi-scale visual features into high-dimensional semantic space to generate comparable embedding vectors, which enables the model to dynamically define the detection categories through text embedding and realize open-world target detection. However, there is a large computational redundancy

in the detection head network which leads to high detection latency and consumes large computational resources; therefore, this paper addresses this problem by lightweighting and improving the detection head based on the YOLO-world model, and the improved open-vocabulary target detection model is shown in Figure 3:

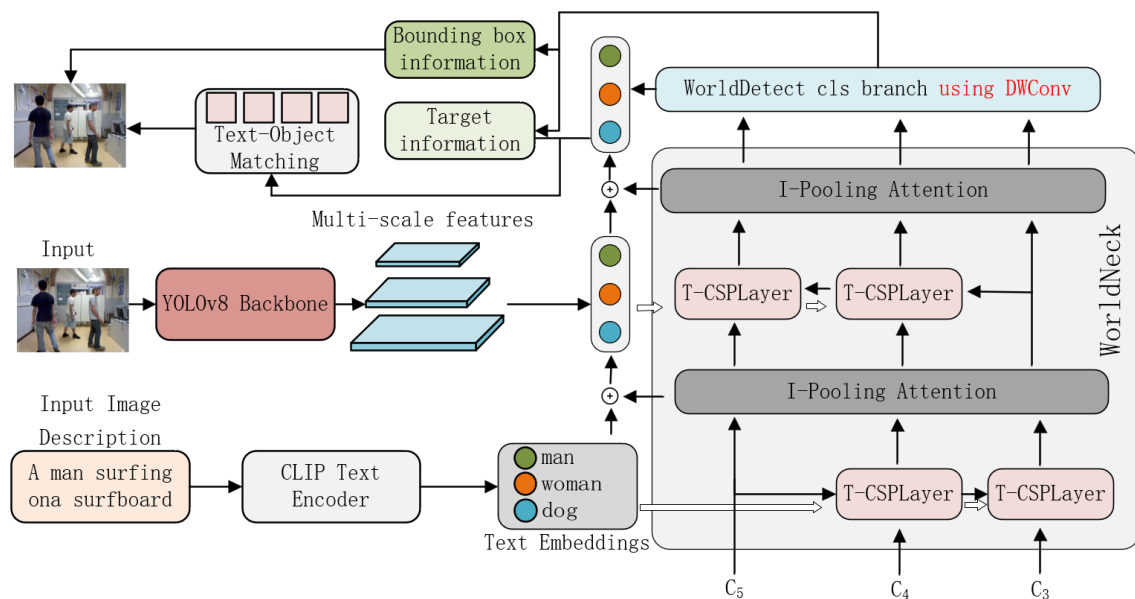


Figure 3. Improved target detection model with lightweight detection head.

The network model of YOLO-World can be roughly divided into a text coding part based on the CLIP model [22] and a detection part based on the YOLOv8 family of network models. Specifically, the text coding part is responsible for collecting information from the input text, extracting the nouns in it, and normalizing them into vectors of uniform length, as shown in the following equation.

$$W = \text{TextEncoder}(T) \in \mathbb{R}^{C \times D}. \quad (1)$$

where C is the number of nouns and D is the length of the expansion into vectors. A RepVL-PAN module for multi-layer cross-modal fusion is also included as the target detection neck network, which can effectively utilize the text vectors and the multi-scale feature information of the image for interaction and fusion to enhance the association between image and text features. In the detection head part, in addition to following the decoupling head regression bounding box and object embedding of YOLO series, a new text comparison head is added. Among them, the convolution module that performs the task of extracting semantic embedding from visual features adopts 3×3 standard convolution, which is used for subsequent comparison learning with text embedding. However, the standard convolution is not separated and optimized when dealing with spatial and channel information, which leads to a large number of parameters and computation, and it stabilizes the region-text training by calculating the object-text similarity s_k , constructing an association-matching mechanism between the object and the nouns in the input text in order to improve the generalization of the model. The following is its calculation formula.

In the original YOLO-World detection head, the standard 3×3 convolution is used to extract semantic embeddings from visual features, which are then compared with text embeddings through the following similarity function. This standard convolution is subsequently replaced in our improved head by a depth-wise separable variant.

$$s_{k,j} = \alpha \cdot L2 - \text{Norm}(e_k) \cdot L2 - \text{Norm}(w_j) + \beta \quad (2)$$

It stabilizes the region–text training by calculating the object–text similarity $s_{k,j}$ between the k -th region embedding e_k and the j -th text embedding w_j , constructing an association-matching mechanism between detected regions and the input text nouns.

Where $L2 - Norm(\cdot)$ is the L2 normalization, $w_j \in W$ is the j th text embedding, and α and β are the learnable scaling factor and offset factor, respectively. The L2 normalization and affine transformation help to stabilize region–text training. The classification part of the detection head shares the same framework with the regression part, which leads to the existence of several-times-higher parametric counts and FLOPs in the classification head than in the regression head, whereas the classification head has a smaller impact on the detection performance; so in order to enable the target detection model to reduce the classification overhead without affecting the detection performance, this paper adopts the 3×3 depth-separable convolution to replace the original convolution by deep convolution (processing only a single channel) and point-by-point convolution (1×1 convolution for channel combination), and spatial feature extraction and channel feature combination are processed separately, which greatly reduces the amount of computation and computational redundancy and improves efficiency.

2.3. Lightweight Target Tracking Algorithm Based on Improved BoT-SORT

Currently, most of the multi-target tracking algorithms use the strategy of high-score bounding box matching to identify the target, and the low-score bounding box is directly discarded, resulting in a large amount of missing target information. BoT-SORT [23] combines the target detection and correlation strategy—the higher the target detection accuracy, the better the target tracking effect—and the open-vocabulary target detector proposed in this paper not only integrates the open-vocabulary detection model with a higher detection accuracy than the original model, but it can also make the mobile robot target following system follow the target better. However, the BoT-SORT tracking algorithm adopts SBS + ResNeSt-50 [24] as the re-recognition network; however, due to the excessive weight of this re-recognition module, it cannot meet the requirements of real-time following of the mobile robot target following system; therefore, in order to address this problem, this paper is based on the improvement in the lightweight target tracking algorithm of the BoT-SORT and selects the more lightweight MobileNetV3 [25] algorithm as the heavy identification network. The lightweight target tracking algorithm based on improved BoT-SORT is shown in Figure 4.

BoT-SORT combines motion camera compensation and Kalman filter direct output for tracking [26]. The targets in each frame are first detected by a target detection model to obtain the bounding box and confidence level of each target. Next, these bounding boxes are correlated with the targets in previous frames using the Hungarian algorithm to ensure that each target matches across frames. Subsequently, feature tracking is performed using sparse optical flow and translation-based local outlier exclusion, followed by calculating the radial transformation matrix using RANSAC [27], predicting and updating the state of each target using the Kalman filter, and outputting the target trajectory and ID for the current frame, transforming the predicted bounding box of the target in the coordinates of the $k - 1$ frame to the coordinates of the k th frame of its next frame. Thus, the tracking accuracy is improved and accurate dynamic tracking is realized.

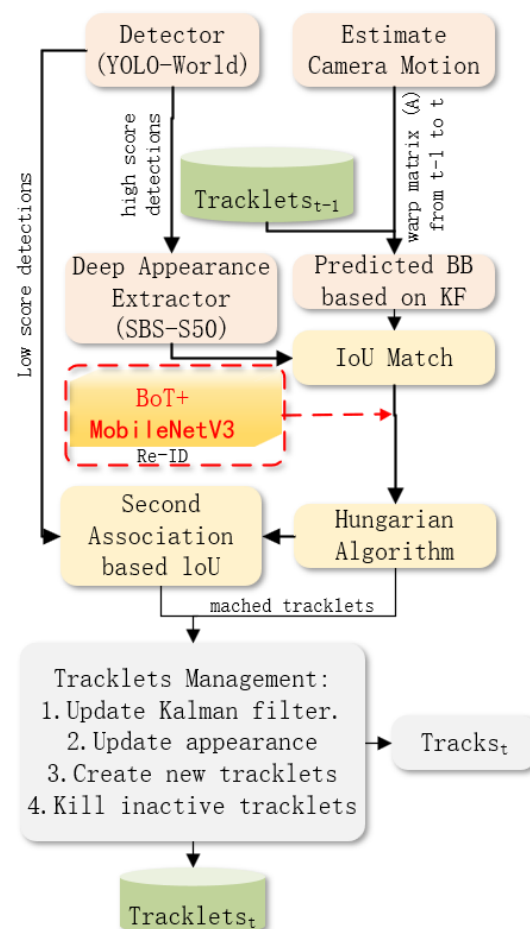


Figure 4. Lightweight target tracking algorithm based on improved BoT-SORT.

The process of re-identification of a mobile robot following a target involves extracting the appearance features of an input image of a specified target, retrieving an image with similar features to the input image in an object image database, and matching the same ID number to ensure that it can be accurately identified when it reappears. MobileNetV3 builds on the experience gained from the previous two generations and delivers excellent performance and speed. MobileNetV3 uses the NetAdapt algorithm to obtain the optimal number of convolutional kernels and channels and carries over the deeply separable convolution of version V1 and the residual structure with linear bottlenecks of version V2. As a result, the MobileNetV3 network extracts image features with satisfactory accuracy in complex tasks at low computational cost.

2.4. Active Search Strategy Based on Visual Servo Control

After obtaining the target tracking results, a target of interest is selected as the following target of the mobile robot. However, when the mobile robot follows a target, the target is often lost because the target exceeds the field of view. In order to enable the mobile robot to achieve stable target following and resume following after the target is lost, this paper proposes an active search strategy based on visual servo control, so that the center point of the following target box is kept as the center point of the field of view and a standard distance is maintained with the target when following. If the target is lost, the mobile robot maintains the motion state in accordance with the predicted target trajectory to conduct active search, re-recognize the target, and make the mobile robot resume following the target. The active search strategy based on visual servo control is shown in Figure 5.

Since the RGB-D camera can directly measure the distance of each pixel in the image from the camera, it can provide richer information than the traditional camera; therefore, in this paper, the RGB-D camera is used as the vision sensor of the mobile robot to obtain the distance between the target and the mobile robot, and its internal reference has been calibrated.

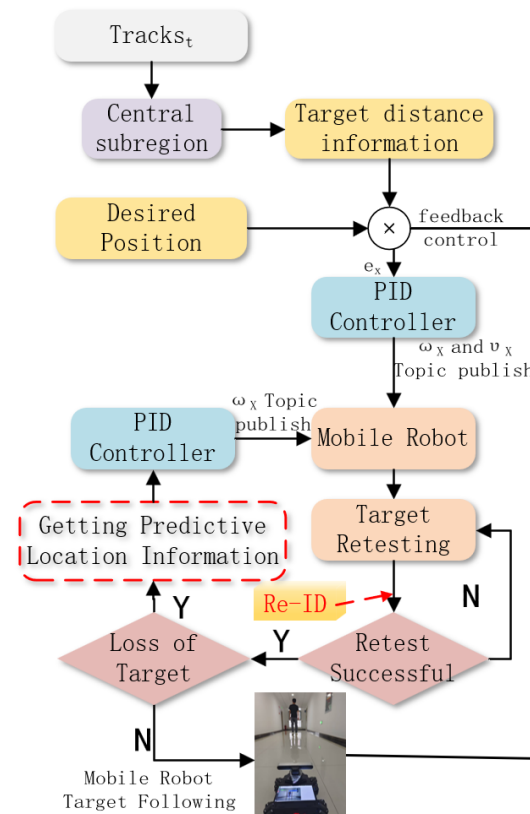


Figure 5. Flowchart of active search strategy based on visual servo control.

The aforementioned tracking algorithm completes the target tracking and obtains the depth information of the target tracking frame. In the process of obtaining the depth, the sudden conversion of a certain point is invalid, which will be considered as the depth value of 0. Measuring only the depth of the center point will have the depth information deviation as too large, resulting in the instability of the target followers, but the direct use of the depth information of the target tracking frame to calculate the distance between the robot and the following target will introduce a large amount of background information. Therefore, in this paper, we extract the depth of the center sub-region and obtain the target depth information by truncated mean filtering to remove the extreme values in the dataset, reduce the interference of the outliers, and reduce the influence of the background noise information on the robot's following accuracy.

The center of the target tracking frame is set as $(x1, x2) \in \beta$. β denotes the coordinates of all pixels in the target tracking frame region. First, the center sub-region β' is extracted by reducing the target tracking frame to one-half of the original one and extracting the depth values within the center sub-region. Then, the depth information average $F(x)$ is obtained by truncated mean filtering, and its value is expressed as the distance between the mobile robot and the target. Let the desired arrival position of the mobile robot be $F'(x)$, and the error $e(x)$ can be defined as

$$e(x) = F(x) - F'(x). \quad (3)$$

The robot control variable is $X_{control} = [U(x) = v_x, W(x) = w_x]$. v_x represents the linear velocity of the mobile robot, w_x represents the angle of the mobile robot, and the PID controller is designed as

$$\begin{bmatrix} U(x) \\ W(x) \end{bmatrix} = -\lambda [k_P e(x) + k_I \sum e(x) + k_D (e(x) - e(x-1))] \quad (4)$$

where k_P , k_I , and k_D are the PID coefficients and λ is the tuning factor.

The error e is the input end of the PID controller, which controls the angular and linear velocities of the mobile robot in real time, and the mobile robot maintains a standard distance from the target and follows the target stably. If the target is out of the field of view of the mobile robot during the following process, the mobile robot will keep the re-recognized features of the tracked target in the last frame before disappearing, predict the next position information of the target according to the tracking frame of the last frame, and output the angular velocity and linear velocity of the mobile robot according to the position information, so as to make the mobile robot continue to move in the direction of the target's disappearance. If the tracking target appears in the field of view again, target detection is performed and the re-identification features of the target are extracted and associated and matched with the re-identification features of the tracking target in the last frame before disappearance; the re-identification of the tracking target is then realized, which is the success of re-checking, and the active search and recovery of the following target can be realized after the target is lost.

3. Experimental Section

To evaluate the performance of the algorithms in this paper, first, ablation experiments and comparison experiments were performed on the target detector proposed in this paper to verify the performance of the grid. Then, comparison experiments were performed using StrongSORT, DeepMOT, ByteTrack, and Deep-SORT algorithms for the target tracker in this paper. Finally, the proposed method is implemented on a mobile robotic platform.

3.1. Experiment Setup and Dataset

The experimental platform consists of two parts: a training platform and an onboard robot platform. The training platform uses an NVIDIA GeForce RTX 4060 Laptop GPU (NVIDIA Corporation, Santa Clara, CA, USA), running Ubuntu 18.04 LTS (Canonical Ltd., London, UK) with ROS Noetic. The deep learning environment is PyTorch 1.12 with CUDA 11.6. The onboard robot platform is equipped with a LeTMC-520 RGB-D camera (LeTV, Beijing, China), which captures images at a resolution of 640×480 pixels with an acquisition rate of 30 fps (raw). The processed frame rate for the following pipeline is approximately 15 fps after resizing to 640×640 . The robot's core processor is an NVIDIA Jetson Nano (4 GB) with an integrated 128-core Maxwell GPU, running Ubuntu 18.04 LTS and ROS Noetic. The software environment on the robot uses PyTorch 1.12 in CPU-only mode (no GPU acceleration on the Jetson Nano due to memory constraints). No ONNX, TensorRT, or other inference acceleration techniques were applied in the current implementation; all inference is performed natively with PyTorch. Future work will explore TensorRT optimization to further reduce onboard inference latency.

The proposed target detector was pre-trained on the PASCAL VOC [28] training set, a publicly available dataset, and evaluated on the PASCAL VOC validation set, which consists of 20 categories with 16,551 images in the training set and 5011 images in the validation set.

3.2. Analysis of the Experimental Results of the Target Detector

The target detector experiments are categorized into ablation, comparison, and open word detection experiments, and the GPU acceleration feature of the PyTorch framework is utilized to train the network. During the training process, the value of batch parameters is set to 8, and the size of the input frame is adjusted to 640×640 . A stochastic gradient descent optimizer (SGD) is used to update and optimize the weights of the model, with momentum configured to 0.937, weight decay configured to 0.0005, and the initial learning rate is 0.01. During the training process, the epoch of all model training is set to 200.

3.2.1. Target Detector Ablation Experiment

To verify the effectiveness of the lightweight backbone network with open-vocabulary detection head lightweight improvement proposed in this paper, we compare this paper's algorithm with four algorithms: (1) the original YOLOv8 network; (2) replacing the original backbone network with the lightweight network backbone GSS proposed in this paper under the framework of YOLOv8; (3) incorporating the YOLO-world network neck and the detection header under the framework of YOLOv8; (4) incorporating the YOLO-world network neck and the detection header under the framework of YOLOv8 and lightweight improvement in the detection header. In order to quantify the performance of target detection, tests were performed on the dataset PASCAL VOC using evaluation metrics such as Parameter (PARAM), Gigafloating Point Operations per Second (GFLOPS), Accuracy (AP), Mean Accuracy (mAP), and Recall. PARAM corresponds to the concept of spatial complexity. GFLOPS denotes one billion floating point operations per second. GFLOPS indicates 1 billion floating point operations per second. AP indicates the average of precision values on the precision and recall (PR) curves. mAP indicates the average AP per category, and recall indicates the accuracy of the prediction in the total positive sample, where the GSS lightweight trunk is denoted by A, the YOLO-world neck and assay head by B, and the lightweight assay head by C. The GSS lightweight trunk is denoted by A, the YOLO-world neck and assay head by B, and the lightweight assay head by C.

Comparing variants YOLOv8n + B (original YOLO-World detection head) and YOLOv8n + C (lightweight detection head with depthwise separable convolution), the lightweight head reduces GFLOPs from 11.7 to 10.5 (a 10.3% reduction) and inference time from 125 ms to 90.9 ms (a 27.3% reduction), while the mAP@0.5 decreases by only 2.0%. This demonstrates that the lightweight detection head achieves meaningful efficiency gains with an acceptable accuracy trade-off for real-time mobile robot applications.

As can be seen in Table 1, the inclusion of the GSS lightweight trunk and the lightweight improvement to the detection header into the original YOLOv8 network reduces the network parameter count and complexity to varying degrees. In particular, after replacing the GSS lightweight backbone, the number of parameters decreases by about 0.5 M, the GFLOPs decreases by about 0.2, and the detection accuracy, recall R, mAP@0.5, and other metrics are slightly reduced from the original network, indicating that the added GSS lightweight backbone reduces the complexity of the network and the number of parameters while maintaining a small loss of precision. The addition of the YOLO-world neck and detection head increases the number of parameters by 0.5 M but gives the network an open detection capability that is not limited to the training of preset categories, and at the same time, improves mAP@0.5 by 0.6%. Adding the YOLO-world neck and detection head and making lightweight improvements to the detection head makes the network reduce the number of parameters by 0.3 M, and the detection precision, recall R, mAP@0.5, and other indexes are slightly reduced, while the final model reduces the number of parameters by 0.8 M and GFLOPs by about 1.6 compared to the addition of the YOLO-world neck and detection head, and the detection precision with mAP@ 0.5 by just 2.5% and 3.2%, which

more than adequately demonstrates that each module used makes the model essential for the target detector proposed in this paper.

Table 1. Comparative experimental results of target detectors on the PASCAL VOC dataset.

Models	A	B	C	PARAM	Precision%	Recall%	mAP@0.5	mAP@0.5:0.95	FPS	GFLOPS
YOLOv8n	×	×	×	3.01	78.8	70.4	78.3	57.7	23	8.2
YOLOv8n + A	✓	×	×	2.53	76.8	65.6	75.3	54.1	26	8.0
YOLOv8n + B	×	✓	×	3.54	78.8	70.4	78.9	58.8	8	11.7
YOLOv8n + C	×	✓	✓	3.22	76.1	67.5	76.9	56.3	11	10.5
Ours	✓	✓	✓	2.74	76.3	65.9	75.7	54.7	13	10.1

3.2.2. Target Detector Comparison Experiment

To verify the effectiveness of the target detectors proposed in this paper, YOLOv8, YOLOv8 replacement backbone for ShuffleNetV2, YOLOv8 replacement backbone for MobileNetV3, and YOLO-world are used to compare with the target detector proposed in this paper and are evaluated by five metrics: accuracy (precision), recall, average precision (mAP@0.5 and mAP@0.5 :0.95), and FPS.

The primary advantage of the proposed detector lies in its combination of open-vocabulary detection capability with a smaller model size and lower inference latency compared to full YOLO-World. While the detection accuracy is approximately 2.5% lower than YOLO-World, the parameter count is reduced by 0.8M and the inference time on the Jetson Nano is reduced from 125 ms to 76.9 ms per frame—a 39.2% reduction in latency. This improvement is critical for real-time mobile robot applications where edge-device constraints limit the use of large models. Models without open-vocabulary capability cannot generalize to user-specified target categories, making them unsuitable for flexible robot following tasks despite their accuracy advantages.

From Table 2, it can be seen that the target detector proposed in this paper significantly reduces the number of parameters and network complexity, proving its superiority. In particular, compared with the original model YOLOv8n, the proposed target detector reduces the number of parameters by about 0.3M, and the detection accuracy and mAP@0.5% are slightly reduced, of which the detection accuracy is reduced by about 2.5%, and mAP@0.5% is reduced by about 2.6%. Compared with replacing the MobileNetV3 backbone model, the number of parameters is reduced by about 0.4M, but the detection precision, recall, and mAP@0.5 are not much different. Compared with the YOLO-world open-vocabulary detection model, the number of parameters is significantly reduced by 0.8M, and the detection precision and mAP@0.5% are slightly reduced, of which the detection precision is reduced by about 2.5% and mAP@0.5% by about 3.2%. Compared with replacing the shuffleNetV2 backbone on the basis of YOLOv8n, the number of parameters is also slightly reduced by about 0.05, and the detection precision is slightly reduced with mAP@0.5%, where the detection precision is reduced by about 1.7%, the recall is reduced by about 2.7%, and mAP@0.5% is reduced by about 1.1%. Although the detection effect is better after replacing the shuffleNetV2 backbone, it does not have the ability of open-vocabulary detection, and the number of open-vocabulary detection model parameters is larger, which proves the efficiency of the target detector proposed in this paper.

Table 2. Comparative experimental results of target detectors on the PASCAL VOC dataset.

Models	PARAM	Precision%	Recall%	mAP@0.5%	mAP@0.5:0.95%	FPS	GFLOPS
YOLOv8n	3.01	78.8	70.4	78.3	57.7	23	8.2
YOLOv8n + MobileNetV3	3.11	75.9	66.0	75.1	54.0	25	6.6
YOLOv8n + shuffleNetV2	2.79	78.0	68.6	76.8	55.2	24	7.5
YOLOv8n + worldv2	3.54	78.8	70.4	78.9	58.8	8	11.7
ours	2.74	76.3	65.9	75.7	54.7	13	10.1

3.2.3. Target Detector Open Lexicon Detection Experiment

In order to verify whether the target detector proposed in this paper has the ability of open-vocabulary detection, the preset categories do not include man, woman, and other categories when the target detector is trained. In this paper, we control the depth camera of the mobile robot to collect the RGB data through the ROS robot operating system, input the target detector proposed in this paper after format conversion, and observe the detection situation of the mobile robot from the viewpoint of mobile robot when man and woman appear in the field of view of the mobile robot. When man and woman appear in the mobile robot's field of view, the mobile robot detects them.

As shown in Figure 6, it can be seen that when the detection category is set to man, and man and woman exist in the field of view at the same time, man can be detected and has a confidence level higher than 0.5, and when the detection category is set to woman, and man and woman exist in the field of view at the same time, woman can be detected, but with a lower level of confidence, which is probably due to the fact that, in order to ensure real-time detection of the target of the mobile robot, the transmitted image data is compressed data, and the image detail information is lost. The possible reason is that, to ensure the real-time detection of mobile robot target, the transmitted image data is compressed data, the image detail information is lost, and the target detector cannot extract enough feature information of man or woman when it carries out feature extraction, which leads to a lower confidence level of detection, whereas when the detection category is set to person, the two people in the field of view can be detected and have a higher confidence level than 0.9, which is due to the training of the target detector being preset. This is due to the fact that the target detector training preset category contains the person category, and the person category will be more easily detected under the premise of enough image data for model training. This shows that the proposed target detector enables the mobile robot to have open-vocabulary target detection capability.

As shown in Table 3, the detector achieves high confidence (0.92) for the seen category 'person', while for unseen categories 'man' and 'woman', the confidence is lower (0.55 and 0.32, respectively). Nevertheless, the model is still able to detect these unseen categories with moderate confidence, demonstrating its open-vocabulary generalization capability. The lower confidence for 'woman' compared to 'man' may be attributed to image compression artifacts in the real-time data stream, which reduce fine-grained feature discriminability.

Table 3. Preset category and non-preset category detection table.

Categories	Train	Confidence
person	✓	0.92
woman	×	0.32
man	×	0.55



Figure 6. Open-vocabulary target detection visualization results.

3.3. Analysis of the Experimental Results of the Target Tracker

The performance evaluation metrics of the target tracking algorithm are selected as multiple object tracking accuracy (MOTA), average ratio (ID F1 score, IDF1), number of lost targets (false negative, FN), number of false detections (false positive, FP), and number of target ID switches (identity switches, IDS) as evaluation metrics for performance assessment.

The public dataset MOT20 [29] test set was selected as the test set for tracking to evaluate the model performance, and five currently popular tracking algorithms (StrongSORT, FairMOT, ByteTrack, Deep-SORT, and BoT-SORT) were selected for experimental comparison; the results are shown in Table 3: BoT-SORT has the highest tracking accuracy and precision. BoT-SORT has the highest tracking accuracy and precision and the lowest number of lost targets and target ID transitions. The BoT-SORT algorithm significantly improves the tracking performance of the algorithm by adopting the motion camera compensation and Kalman filter direct output. Therefore, BoT-SORT was selected as the tracker.

Since the re-recognition part of the BoT-SORT algorithm uses a fast-reid re-recognition model, and the deep learning-based re-recognition model requires large computational resources and has high latency when performing the mobile robot target following task, a lightweight algorithm is required to ensure real-time tracking and efficiency. The re-recognition models ResNeSt-50, MobileNetV3-Large, MobileNetV3-Small, and ShuffleNetV2 are compared experimentally on the dataset Market-1501 [30], and the models are evaluated by mAP and Rank-5 metrics; the results are shown in Table 4. Rank-5 refers to an accuracy metric that measures whether the true label is contained within the top 5 highest-probability categories predicted by the model.

The ResNeSt-50 re-identification model size is 287 MB; the results are shown in Table 5. If the original ResNeSt-50 network of BoT-SORT is used, the tracking speed of the mobile robot to the detected target will be slower, and it cannot meet the demand for low latency of the target following method of the mobile robot; whereas, if MobileNetV3-Small is used as the re-identification network, although the average precision mean value is decreases by 11.82%, its model size is 26.4 MB, and the memory is reduced by about 90.8%, which

greatly reduces the memory occupation and substantially improves the tracking speed, so Mobile-NetV3-Small is used as the backbone of the re-recognition network in the BoT-SORT algorithm.

Table 4. Comparative experimental results of the BoT-SORT target tracking algorithm on the MOT20 dataset.

Models	MOTA	IDF1	FN	FP	IDs
StrongSORT	79.6	79.5	86,205	27,876	1194
FairMOT	73.7	72.3	117,477	27,507	3433
ByteTrack	80.3	77.3	83,721	25,491	2196
Deep-SORT	74.5	76.4	106,526	23,368	1787
BoT-SORT	80.6	79.5	85,398	22,524	1257

Table 5. Comparative experimental results of the fast-reid re-identification model on the Market-1501 dataset.

Models	mAP	Rank-5	Weights
ResNeSt-50	85.63	98.07	287
MobileNetV3-Large	79.22	97.42	59.5
MobileNetV3-Small	73.81	96.70	26.4
ShuffleNetV2	65.52	94.64	37.6

3.4. Moving Machine Target Following Experiment

In order to test the effectiveness of the algorithm in this paper, mobile robot target following experiments were carried out in a school hallway, which has a 90-degree corner in the indoor scene.

As can be seen in Figure 7, firstly, before the target enters the corner, the mobile robot follows the target stably, and it can be seen that the following target id is 1. When the target enters the corner, it is out of the field of view of the mobile robot, resulting in the loss of the following target, and then the mobile robot predicts the target motion state by acquiring the target tracker and releases the corresponding linear and angular speeds through the ROS, so that the mobile robot can actively search along the lost target. The target tracking algorithm adopted in this paper effectively associates the target reappearance frame with the target features of all previous frames, and the id is still 1 after feature re-identification of the following target, so that the mobile robot successfully realizes the restoration of the following of the lost target. Experimentally, it is proved that the algorithm of this paper has high accuracy in target detection and following, has good real-time performance, and is able to realize real-time and stable following of the target.

The results show that the proposed method achieves a recovery success rate of 90% in turn scenarios, 80% under pedestrian occlusion, and 70% in low-illumination conditions. The overall average recovery rate across all scenarios is 80%. The frame rate remains stable at approximately 13 FPS in all scenarios, confirming real-time performance on the mobile robot platform, the results are shown in Table 6. The lower recovery rate under low illumination is attributed to degraded visual feature quality, which will be addressed in future work.

Table 6. Table of recovery and following conditions of mobile robots in different scenarios.

Scenario	Trials	Loss Events	Recovery Rate	FPS
low illumination	20	6	70	13
turns	20	2	90	13
pedestrian obstruction	20	4	80	13

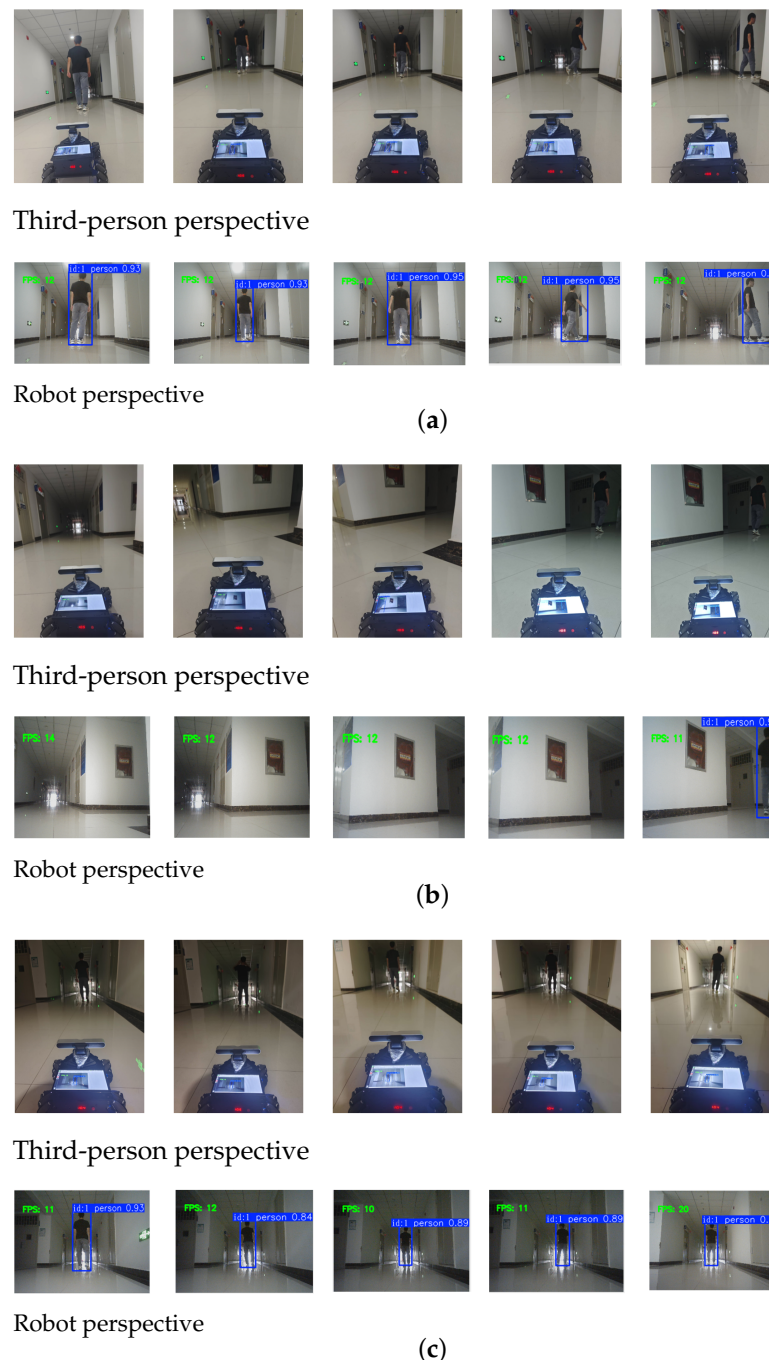


Figure 7. Moving Machine My Target Following Experiment. (a) Target loss. (b) Active search. (c) Resumption of following.

4. Conclusions

This paper proposes a low-latency mobile robot target following method based on improved YOLO-World, consisting of three parts: target detection, target tracking, and motion control. By introducing the GSS lightweight backbone and depthwise separable convolutions in the detection head, the proposed detector achieves 75.7% mAP@0.5 on PASCAL VOC with only 2.74 M parameters while retaining open-vocabulary detection capability. Replacing the ReID backbone with MobileNetV3-Small reduces the model size by 90.8%, and the active search strategy based on visual servo control enables effective target re-acquisition after loss. The system achieves an end-to-end latency of 76.9 ms (13 FPS) on the Jetson Nano platform with an overall recovery success rate of 80% across three

real-world scenarios. However, several limitations remain: the recovery rate degrades to 70% under low-illumination conditions due to reduced feature discriminability; no hardware-specific inference acceleration such as TensorRT has been applied, leaving room for further latency reduction; and the open-vocabulary detection confidence for unseen categories remains lower than for pre-trained ones, particularly under image compression artifacts. Future work will focus on low-light robustness enhancement, TensorRT-based inference optimization, and validation in more complex outdoor environments.

Author Contributions: Conceptualization, Y.S.; methodology, K.M.; software, K.M.; validation, M.Z.; formal analysis, M.Z.; investigation, K.M. and M.Z.; resources, Y.S.; data curation, K.M.; project administration, validation, and writing—review and editing, R.Z.; writing—original draft preparation, K.M.; writing—review and editing, Y.S. and S.H.; visualization, K.M.; supervision, Y.S.; project administration, R.Z.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: Funded by Science Research Project of Hebei Education Department (Grant No. QN2025244).

Data Availability Statement: The data that support the findings of this study are openly available in [PASCAL VOC] at <http://host.robots.ox.ac.uk/pascal/VOC/> (accessed on 16 March 2025).

Conflicts of Interest: The authors declare that they have no known financial or non-financial competing interests that could have influenced the work reported in this paper.

References

1. Wang, L.-J.; Jia, S.-M.; Li, X.-Z.; Wang, S. Person following for mobile robot using improved multiple instance learning. *Acta Autom. Sin.* **2014**, *40*, 2916–2925.
2. Cao, F.; Zhuang, Y.; Yan, F.; Yang, Q.-F.; Wang, W. Long-term autonomous environment adaptation of mobile robots: State-of-the-art methods and prospects. *Acta Autom. Sin.* **2020**, *46*, 205–221.
3. Miao, Z.; Zhong, H.; Lin, J.; Wang, Y.; Chen, Y.; Fierro, R. Vision-based formation control of mobile robots with FOV constraints and unknown feature depth. *IEEE Trans. Control Syst. Technol.* **2020**, *29*, 2231–2238. [[CrossRef](#)]
4. Marvasti-Zadeh, S.M.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S. Deep learning for visual tracking: A comprehensive survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 3943–3968. [[CrossRef](#)]
5. Cao, J.; Pang, Y.; Xie, J.; Khan, F.S.; Shao, L. From handcrafted to deep features for pedestrian detection: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 4913–4934. [[CrossRef](#)]
6. Hsu, W.-Y.; Lin, W.-Y. Ratio-and-scale-aware YOLO for pedestrian detection. *IEEE Trans. Image Process.* **2020**, *30*, 934–947. [[PubMed](#)]
7. Hussain, M.; Al-Aqrabi, H.; Munawar, M.; Hill, R.; Alsabou, T. Domain feature mapping with YOLOv7 for automated edge-based pallet racking inspections. *Sensors* **2022**, *22*, 6927. [[CrossRef](#)] [[PubMed](#)]
8. Varghese, R.; Sambath, M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*; IEEE: New York, NY, USA, 2024; pp. 1–6.
9. Cheng, T.; Song, L.; Ge, Y.; Liu, W.; Wang, X.; Shan, Y. Yolo-world: Real-time open-vocabulary object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16901–16911.
10. Maalouf, A.; Jadhav, N.; Jatavallabhula, K.M.; Chahine, M.; Vogt, D.M.; Wood, R.J.; Torralba, A.; Rus, D. Follow anything: Open-set detection, tracking, and following in real-time. *IEEE Robot. Autom. Lett.* **2024**, *9*, 3283–3290. [[CrossRef](#)]
11. He, X.; Song, X. Improved YOLOv4-Tiny lightweight target detection algorithm. *J. Front. Comput. Sci. Technol.* **2024**, *18*, 138.
12. Zhou, W.; Wu, Z.; Zhang, Z.; Peng, L.; Xie, L. Lightweight small target detection method based on weak feature enhancement. *Control Decis.* **2024**, *39*, 381–390.
13. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
14. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*; IEEE: New York, NY, USA, 2016; pp. 3464–3468.
15. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*; IEEE: New York, NY, USA, 2017; pp. 3645–3649.

16. Peng, J.; Wang, C.; Wan, F.; Wu, Y.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; Fu, Y. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *Computer Vision—ECCV 2020*; Springer: Cham, Switzerland, 2020; pp. 145–161.
17. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087.
18. Normey-Rico, J.E.; Alcalá, I.; Gómez-Ortega, J.; Camacho, E.F. Mobile robot path tracking using a robust PID controller. *Control Eng. Pract.* **2001**, *9*, 1209–1214. [[CrossRef](#)]
19. Motokucho, T.; Oda, N. Vision-based human-following control using optical flow field for power assisted wheelchair. In *2014 IEEE 13th International Workshop on Advanced Motion Control (AMC)*; IEEE: New York, NY, USA, 2014; pp. 266–271.
20. Gona, S.N.R.; Harish, C.H. Intelligent mobility planning for a cost-effective object follower mobile robotic system with obstacle avoidance using robot vision and deep learning. *Evol. Intell.* **2024**, *17*, 1279–1293.
21. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More features from cheap operations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
22. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2021; pp. 8748–8763.
23. Aharon, N.; Orfaig, R.; Bobrovsky, B.-Z. BoT-SORT: Robust associations multi-pedestrian tracking. *arXiv* **2022**, arXiv:2206.14651.
24. Zhang, H.; Wu, C.; Zhang, Z.; Zhu, Y.; Lin, H.; Zhang, Z.; Sun, Y.; He, T.; Mueller, J.; Manmatha, R.; et al. ResNeSt: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, 19–20 June 2022; pp. 2736–2746.
25. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
26. Liu, S.; Li, X.; Lu, H.; He, Y. Multi-object tracking meets moving UAV. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, 18–24 June 2022; pp. 8876–8885.
27. Du, Y.; Zhao, Z.; Song, Y.; Zhao, Y.; Su, F.; Gong, T.; Meng, H. StrongSORT: Make DeepSORT great again. *IEEE Trans. Multimed.* **2023**, *25*, 8725–8737.
28. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136.
29. Dendorfer, P.; Rezatofighi, H.; Milan, A.; Shi, J.; Cremers, D.; Reid, I.; Roth, S.; Schindler, K.; Leal-Taixé, L. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv* **2020**, arXiv:2003.09003.
30. Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; Tian, Q. Scalable Person Re-identification: A Benchmark. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 7–13 December 2015.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.