electronics

MDPI

Article

Research on Road Sign Detection and Visual Depth Perception Technology for Mobile Robots

Jianwei Zhao and Yushuo Liu *

School of Mechanical Electronic & Information Engineering, China University of Mining and Technology, Beijing 100083, China; zhaojianwei@cumtb.edu.cn

* Correspondence: sqt2100402037@student.cumtb.edu.cn

Abstract: To accomplish the task of detecting and avoiding road signs by mobile robots for autonomous running, in this paper, we propose a method of road sign detection and visual depth perception based on improved Yolov5 and improved centroid depth value filtering. First, the Yolov5 model has a large number of parameters, a large computational volume, and a large model size, which is difficult to deploy to the CPU side (industrial control computer) of the robot mobile platform. To solve this problem, the study proposes a lightweight Yolov5-SC3FB model. Compared with the original Yolov5n model, the Yolov5-SC3FB model only loses lower detection accuracy, the parameter volume is reduced to 0.19 M, the computational volume is reduced to 0.5 GFLOPS, and the model size is only 0.72 MB, making it easy to deploy on mobile robot platforms. Secondly, the obtained depth value of the center point of the bounding box is 0 due to the influence of noise. To solve this problem, we proposed an improved filtering method for the depth value of the center point in the study, and the relative error of its depth measurement is only 2%. Finally, the improved Yolov5-SC3FB model is fused with the improved filtering method for acquiring centroid depth values and the fused algorithm is deployed to the mobile robot platform. We verified the effectiveness of this fusion algorithm for the detection and avoidance of road signs of the robot. Thus, it can enable the mobile robot to correctly perceive the environment and achieve autonomous running.

Keywords: lightweight Yolov5; visual depth perception; mobile robot

check for updates

Citation: Zhao, J.; Liu, Y. Research on Road Sign Detection and Visual Depth Perception Technology for Mobile Robots. *Electronics* **2023**, *12*, 3202. https://doi.org/10.3390/electronics12143202

Academic Editors: Dah-Jye Lee, Gábor Kertész, Sašo Tomažič, Sara Stančin, Peter Sarcevic and Akos Odry

Received: 29 May 2023
Revised: 5 July 2023
Accepted: 15 July 2023
Published: 24 July 2023

1. Introduction

Mobile robots are an important branch of intelligent robots. The most critical technology for mobile robots is an autonomous navigation system, which includes environment perception, path planning, and motion control. Environment perception of robots refers to the use of various sensors to obtain information about the surrounding environment, and the acquired information is analyzed and processed to realize the perception and understanding of the surrounding environment. Thus, the robot can complete various tasks intelligently and autonomously. Scholars began to study environmental perception technologies for mobile robots as early as the 1960s. In the 1960s, the Stanford International Research Institute developed the Shakey robot [1]. It was the first mobile robot capable of sensing and reasoning about its surroundings. By being equipped with devices such as cameras, rangefinders, and collision sensors to sense the environment, it performs path planning, moving targets and automatic obstacle avoidance in a structured environment. The Sojourner robot is a first-generation Mars exploration rover, used in NASA's Mars Exploration Mission launched in 1996. It senses the environment on the surface of Mars by carrying various sensors in order to perform a series of exploration and scientific experiments [2]. The Nao robot, developed by the French robotics company Aldebaran Robotics in 2006, has multiple sensors and actuators to navigate, move and maneuver autonomously in complex environments. It uses various artificial intelligence technologies, including neural networks and computer vision, and is oriented towards autonomous and interactive social robots [3]. In addition, floor sweeping robots [4], food delivery robots [5], humanoid robots [6], self-driving cars [7], and autonomous flying drones [8] also belong to mobile robots. These mobile robots walk, drive, fly, and perform tasks in a given environment depending on their ability to perceive their surroundings. The important elements of perception include how to obtain depth information (distance between the camera and the object) and how to detect the class of objects in the environment. In recent years, LIDAR (Light Detection and Ranging) or traditional RGB (Red, Green, Blue) binocular cameras have often been used to obtain depth information (distance). LIDAR is more accurate, but often expensive [9]. The traditional RGB stereo camera has low hardware requirements, simple implementation, and low power consumption. However, it can only obtain two-dimensional images of the object scene in the real world and obtaining depth information often requires complex calculations [10]. To better obtain depth information, people are increasingly using depth cameras to capture the depth information of the environment and then further complete tasks such as environment modeling and object recognition. There are generally three solutions for obtaining depth information using depth cameras, which can be roughly divided into the time-of-flight method [11], the binocular stereovision method [12], and the structured light method [13]. The Intel RealSense depth camera is the world's first device that integrates 3D depth and 2D camera modules, giving the device a vision depth similar to that of the human eye [14]. The Intel RealSense depth camera combines the characteristics of both structured light camera and pure stereo camera. Currently, the Intel RealSense depth camera (3D camera) is widely used in robotics due to its low price, light weight, and small size [15–18].

Traditional target detection algorithms often rely on hand-designed features. However, hand-designed features are not robust to changes in the diversity of targets, backgrounds, and lighting conditions, which can affect the accuracy of target detection. The sliding window-based region selection is not targeted, which affects real-time detection. With the continuous development of computer hardware, many excellent deep learning-based target detection models (algorithms) have emerged, including two-stage target detection models and one-stage target detection models. Typical two-stage detection models include R-CNN series, such as: R-CNN [19] model, Fast R-CNN [20] model, and Faster R-CNN [21] model. Typical one-stage detection models include YOLO [22–28] series models and SSD [29] models. The R-CNN series tends to have a significant advantage in detection while detection speed is slow. The SSD models tends to have a significant advantage in the speed of detection, while the accuracy of detection is lower. The Yolov5 model in the one-stage models balances the speed and accuracy of detection very well. However, the application of the Yolov5 model to mobile robots is limited by the computing power and storage space of the mobile platform.

The Yolov5 model is difficult to deploy to the CPU side (IPC: Industrial Personal Computer) of the robot mobile platform due to its large number of parameters, large computation and large model size (large model weight file), and low detection speed. In this paper, we propose a Yolov5-SC3FB model, which is improved based on Yolov5n. The Yolov5-SC3FB model reduces the number of parameters and computation of the network, shrinks the model volume, has less detection accuracy loss, and improves the detection speed in dynamic scenarios. Combined with the experimental scenarios of mobile robots, this paper selects Intel RealSense D455 camera as the vision sensor for mobile robots to endow the mobile robot with visual depth perception capability. For the problem that the center point of the bounding box obtained by Intel RealSense D455 sometimes causes the measured depth value to be unstable (i.e., the measured depth value is 0) due to the influence of noise points, in this paper, we improve the method of obtaining the depth value of the center point by filtering. In addition, the paper integrates the Yolov5-SC3FB model with the improved filtering method to obtain the depth value of the center point and simultaneously obtains the category information (turn left, turn right, backward, and stop) and depth (distance) information of the environmental road signs. Finally, the obtained category information and depth information are extracted and transformed into control commands. This enables the mobile robot to avoid the road signs and achieve autonomous running according to the instructions of the road signs under the set distance threshold.

2. Materials and Methods

2.1. Construction of the Road Signs Sample Dataset

The sample dataset of road signs constructed in this paper is mainly divided into four categories: left, right, ban, and stop, as shown in Figure 1.

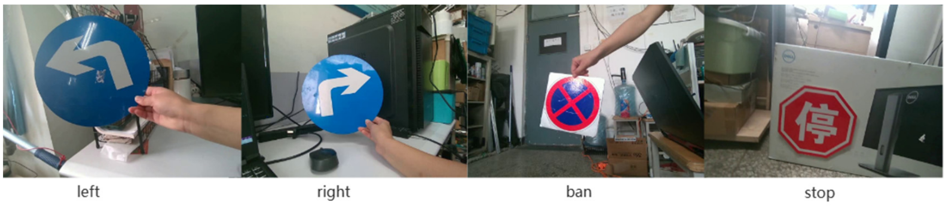


Figure 1. Sample dataset of road signs.

The road sign sample dataset constructed in this paper is derived from two parts. One part of the data comes from the video acquisition of road sign samples by an Intel Realsense D455 camera in a laboratory scene; the other part comes from the Chinese traffic sign recognition database. The video acquisition of road sign samples was performed with the Intel Realsense D455 camera under different illumination, different angles, and different distances, and the resolution of the acquired video was set to 640 × 480. As the camera continuously acquires road sign samples, it will cause the problems of a single scene, low quality, and too many repetitions, which will greatly affect the training effect of the neural network. To avoid the above problems, this paper retains the image data of road sign samples every 50 frames in the captured video, uses a Python script to delete the sample data with similarity greater than 0.8, and then manually rejects the samples with no target or poor quality to obtain 549 road sign sample data points. The 688 road sign sample images in the Chinese traffic sign recognition database were manually screened.

In order to improve the anti-interference ability of the network, the road sign sample data were enhanced by adding Gaussian noise and adding pretzel noise. Considering that in the subsequent robot research, the robots will make the camera move, thus causing the problem that the acquired road sign data are blurred and difficult to identify, in this paper, we performed motion blurring processing on the acquired road sign sample data. The partially enhanced images of the road sign samples are shown in Figure 2.

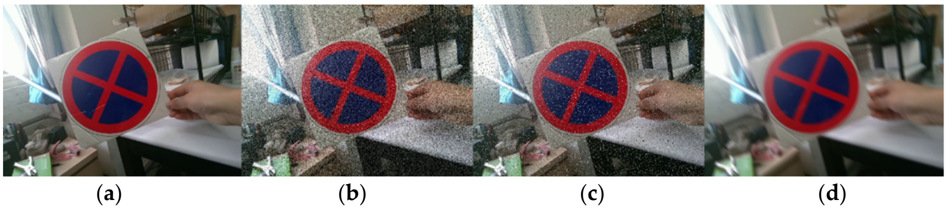


Figure 2. Enhanced partial road sign dataset. (**a**) Original image; (**b**) Sample image with Gaussian noise added; (**c**) Sample image with pretzel noise added; (**d**) Sample image for motion blur processing.

In this paper, the road sign sample data were expanded by data augmentation and we constructed the final dataset of 7012 road sign samples. LabelImg (version: Binary v1.8.1) image annotation tool was used to manually label the road sign sample ldataset. The training set, validation set, and test set were randomly divided according to the ratio of 6:2:2, the training set had 4206 data points, the validation set 1403, and the test set 1403.

2.2. Overall Solution for Mobile Robot Target (Road Sign) Detection and Visual Depth Perception

The overall flowchart of the proposed road sign detection and visual depth perception scheme is shown in Figure 3. First, a trained and improved Yolov5-SC3FB detection model is loaded. Then, the Intel Realsense D455 camera is used for image acquisition and aligning the acquired color map with the depth map. The aligned color map is fed into the Yolov5-SC3FB model for detection until a road sign is detected, otherwise the image is captured all the time. The detection of road signs using the Yolov5-SC3FB model yields the pixel coordinates of the road sign's bounding box, the category information, and the confidence information. The pixel coordinates of the center point of the bounding box of the road sign are used to calculate the pixel coordinates of the center point of the bounding box of the road sign, and the depth value of the center point of the bounding box of the road sign is initially obtained by aligning the pixel coordinates of the center point with the depth image. Due to the noise effect, the depth value of the acquired road sign bounding box centroid is filtered and improved, and the depth information of the road sign is obtained by the improved filtering method of acquiring the depth value of the bounding box centroid. Finally, the category information and the depth information of the road signs are obtained simultaneously.

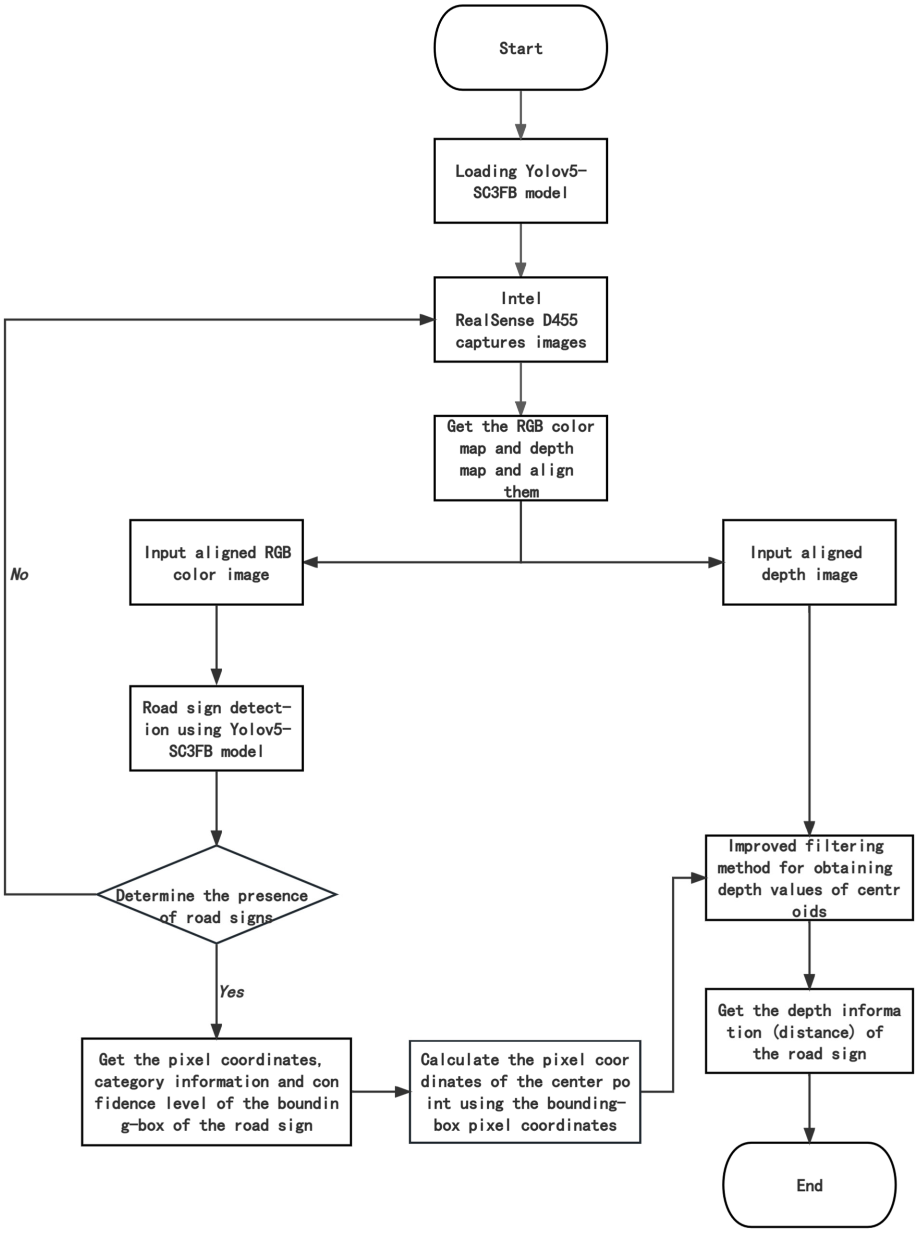

Start
Loading Yolov5-SC3FB model
Intel RealSense D455 captures images
Get the RGB color map and depth map and align them
Input aligned RGB color image
Input aligned depth image
No
Road sign detection using Yolov5-SC3FB model
Determine the presence of road signs
Yes
Get the pixel coordinates, category information and confidence level of the bounding-box of the road sign
Calculate the pixel coordinates of the center point using the bounding-box pixel coordinates
Improved filtering method for obtaining depth values of centroids
Get the depth information (distance) of the road sign
End

Figure 3. Flow chart of the overall scheme of road sign detection and visual depth perception solution.

2.3. Yolov5 Network Model

The network structure of Yolov5 is mainly composed of four parts: Input, Backbone, Neck, and Head. The basic idea of the Yolov5 network is described below. Firstly, on the Input side, the road sign image is pre-processed mainly by using Mosaic data enhancement, image scaling, and adaptive initial anchor frame calculation. Then, the pre-processed road sign images are subjected to feature extraction by convolutional neural network (Backbone), and the extracted features are fused by Neck's feature pyramid structure FPN [30] + PAN [31]. Finally, the output layer (Head) performs classification and regression based on the fused image features to obtain the target bounding box and category confidence. The network structure diagram of Yolov5 and the structure of each module are shown in Figure 4.

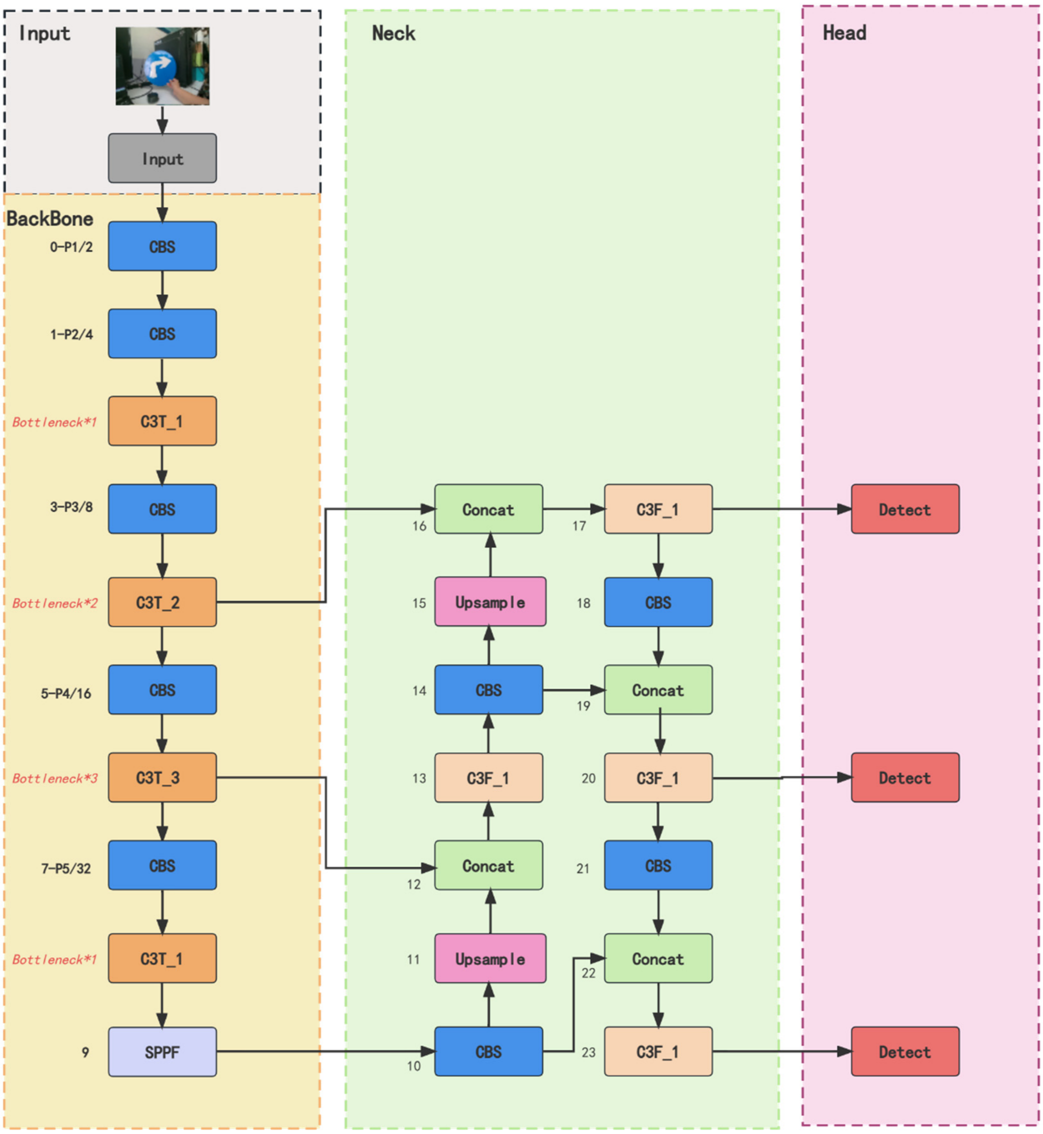


Figure 4. *Cont.*

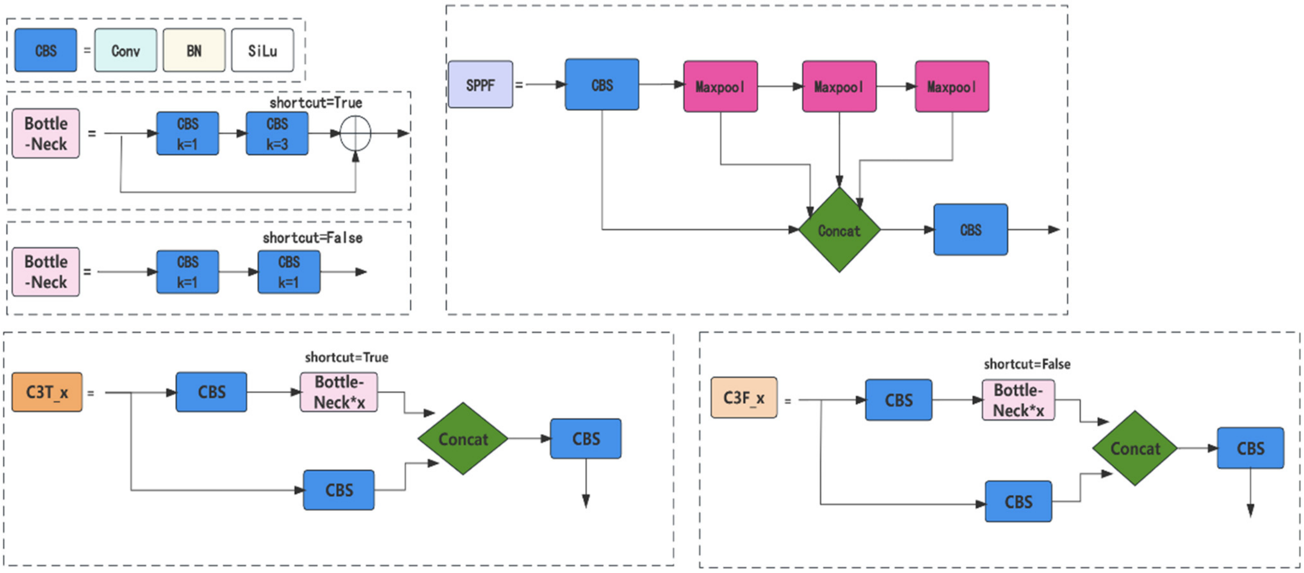


Figure 4. Network structure diagram of Yolov5 and the structure of each module. The * in the figure means multiplication.

2.4. Improved Yolov5 Network Model

In this study, we improve the Yolov5n network model and propose a model applicable to mobile robots for road sign detection in indoor scenarios: Yolov5-SC3FB. The main improvements are as follows. (1) We adopt the structure of the lightweight network ShuffleNet V2. Without significantly reducing the detection capability of the model, the stage2, stage3, and stage4 stacking structure of ShuffleNetV2 are shrunk to reduce the number of parameters. The shrunken ShuffleNet V2 is also used to reconfigure the backbone network of Yolov5. (2) In the feature fusion part of Yolov5, the Faster_Block is embedded into the C3 module of Yolov5 to build the C3Faster module to improve the detection speed of the model. (3) In the feature fusion part of Yolov5, we adopt the idea of cross-scale connectivity and use a simplified version of BiFPN, which can fuse feature information more effectively without introducing too much computation and affecting the detection speed of the network as little as possible.

2.4.1. Reduction of ShuffleNet V2 and Its Stacking Structure

ShuffleNet V2 [32] uses Channel Shuffle operation for channel rearrangement, cross-group information exchange, and enhanced feature extraction. Two basic modules of ShuffleNet are designed according to four lightweight network design guidelines. The two basic modules of ShuffleNetv2 are shown in Figure 5. In this paper, to simplify the names of these two basic modules, they are named as SF_B1 and SF_B2 (stride = 2). The number of input feature channels is divided into two branches in SF_B1 of Figure 5a (the number of channels in both branches is half of the number of input channels). The left branch is left unprocessed, while the right branch performs a series of operations such as Conv, BN, DWConv, etc. Then the number of output channels is made the same as the number of input channels by Concat operation. Finally, the Channel Shuffle operation is performed to exchange channel information between different groups. SF_B2 (stride = 2) in Figure 5b divides the input feature channel number into two branches (the number of channels in both branches is equal to the number of input channels). Among them, the left and right branches perform Conv and DWConv operations, and the feature splicing is performed by Concat. At this time, the number of feature channels is twice the number of input feature channels and the feature map size is halved (downsampling). These two basic units in Figure 5 are stacked with convolution and pooling modules to jointly build the ShuffleNet V2 network structure.

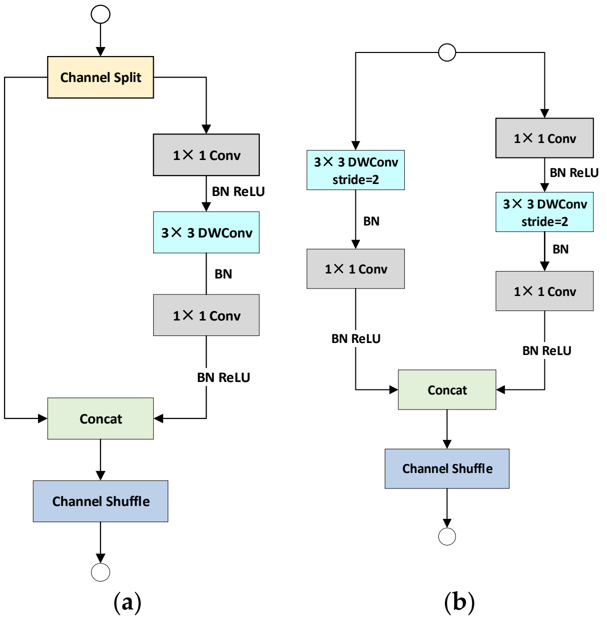


Figure 5. The two basic modules of ShuffleNetV2. (**a**) SF_B1; (**b**) SF_B2 (stride = 2).

To minimize the number of parameters in the ShuffleNetV2 model, the stage2, stage3, and stage4 stacking structures in the ShuffleNetV2 structure are reduced in this paper without significantly reducing the detection capability of the model. The overall network structure of the reduced ShuffleNetV2 is shown in Table 1. The modules of the stage2, stage3, and stage4 stacks and the number of channels are scaled to generate a network of different complexity than the original ShuffleNetV2.

Table 1. General network structure of ShuffleNetV2.

Layer	Output Size	KSize	Stride	Repeat	Output Channels
Image	640 × 640				3
ConvBNReLU	320 × 320	3 × 3	2	1	32
Maxpool	160 × 160	3 × 3	2	1	32
Stage2	80 × 80		2	1	32
	80 × 80		1	1	32
Stage3	40 × 40		2	1	64
	40 × 40		1	3	64
Stage4	20 × 20		2	1	128
	20 × 20		1	1	128

The Channel Shuffle idea of the reduced ShuffleNetV2 is introduced into the Yolov5 network to reconstruct the feature extraction network of Yolov5. The number of parameters of the original Yolov5 network is reduced and the feature extraction capability of the network is affected to a lesser extent.

2.4.2. Building the C3Faster Module

There is usually a discrepancy between the FLOPs of the network and the latency, and to try to eliminate this discrepancy, Chen J et al. [33] proposed a new partial convolution (PConv) to extract spatial features more efficiently by simultaneously reducing redundant computations and memory accesses. PConv is shown in Figure 6a.

In Figure 6a, h and w denote the height and width of the input features (Input) and output features (Output), respectively. c_p denotes that the input features are convolved conventionally on c_p number of channels only. The regular convolution operation uses c_p convolution kernels of size k × k. The remaining channels of the input features are left unprocessed and the Identity operation is applied directly.

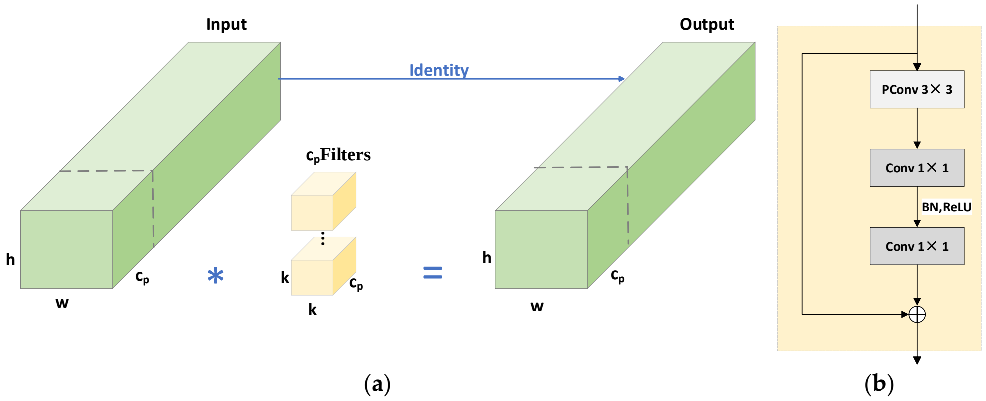


Figure 6. PConv and Faster_Block. (**a**) Partial Convolution (PConv); (**b**) Faster_Block. The * in the figure means multiplication.

To fully and efficiently utilize the information from all channels, further point-by-point convolution (PWConv) is connected behind PConv to form the Faster_Block module. The Faster_Block module is shown in Figure 6b.

In Figure 6b, the Faster_Block has two main branches, the left branch and the right branch. The left branch is a shortcut operation, which can directly transfer the shallow information to the deep layer and solve the degradation problem while reusing the features. The right branch performs operations such as PConv, 2 PWConv (or 1 × 1 Conv). Finally, the output features of the left branch and the right branch are fused together.

Due to the large number of parameters and computation of the C3 module used in the feature fusion part of Yolov5, this paper constructs the C3Faster module by embedding the Faster_Block into the C3 module in Yolov5. The C3Faster module is shown in Figure 7, which replaces BottleNeck in the original Yolov5 model with a large number of parameters.

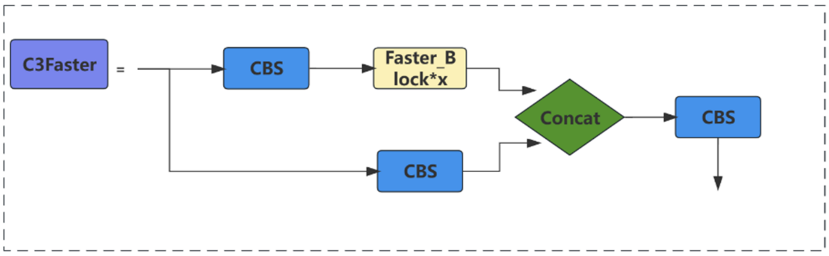


Figure 7. C3Faster module. The * in the figure means multiplication.

2.4.3. BiFPN and Its Simplification

The BiFPN structure evolves on the basis of PANet, which proposes two main ideas: one is cross-scale connectivity and the other is multi-scale feature fusion by weighting [34]. BiFPN is shown in Figure 8a. From Figure 8a, it can be seen that the BiFPN structure fuses the features from layer 3 feature layer P3 to layer 7 feature layer P7 of the original network (Efficientdet) on the basis of PANet (the dashed box part is drawn as the feature fusion part.) BiFPN considers that nodes with only one input do not contribute much to feature fusion, therefore, in order to reduce the computational effort, P3 and P7 nodes corresponding to single-input and top-down paths for feature fusion are removed. P4, P5, and P6 are connected across scales with the nodes of bottom-up paths for feature fusion. In addition, the top-down and bottom-up multi-scale feature fusion process uses weighting to introduce learnable weights to learn the importance of different input features. However, the network tends to affect the speed of the network in the process of learning weights. Therefore, this paper only refers to the idea of cross-scale connectivity in BiFPN without setting learning weights on the PANet of Yolov5. Yolov5 only has three feature layers, P3, P4, and P5, for feature fusion, which does not consume much time. Therefore, in this paper, we choose to retain the nodes on the top-down path of the feature fusion part corresponding to P3 and P5. The idea of BiFPN spanning connectivity is taken into account at the P4 layer. A simplified version of BiFPN applicable to Yolov5 network is formed for feature fusion. The PANet structure in the original feature fusion network of Yolov5 and the simplified BiFPN structure used in this paper are shown in Figure 8b,c.

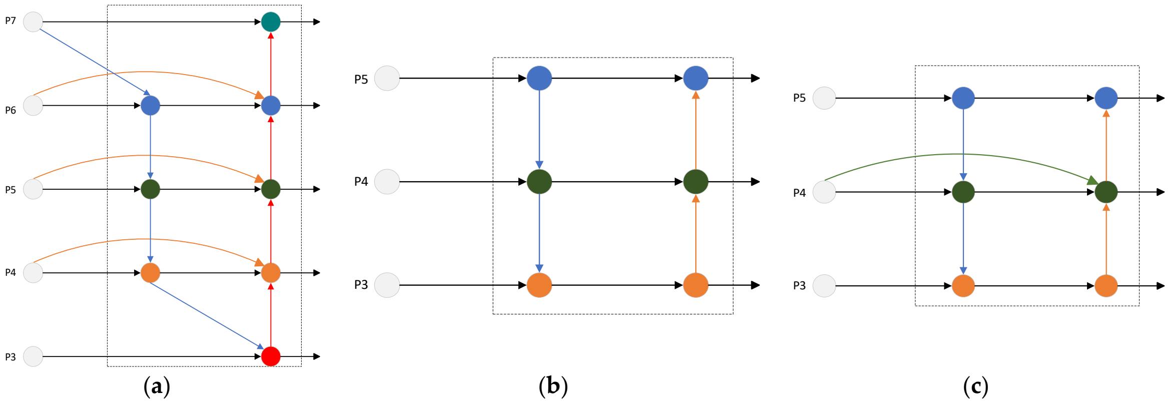


Figure 8. Structure used for feature fusion. (**a**) BiFPN in Efficientdet [34]; (**b**) PANet used in Yolov5; (**c**) The simplified version of BiFPN used in this paper.

2.4.4. Yolov5-SC3FB Network Model

The above reduced ShuffleNet V2 structure, the constructed C3Faster module and the simplified BiFPN are used to improve the Yolov5 network. The improved Yolov5-SC3FB network structure is shown in Figure 9, and all the improved modules are marked by red outer frame lines and red connection lines.

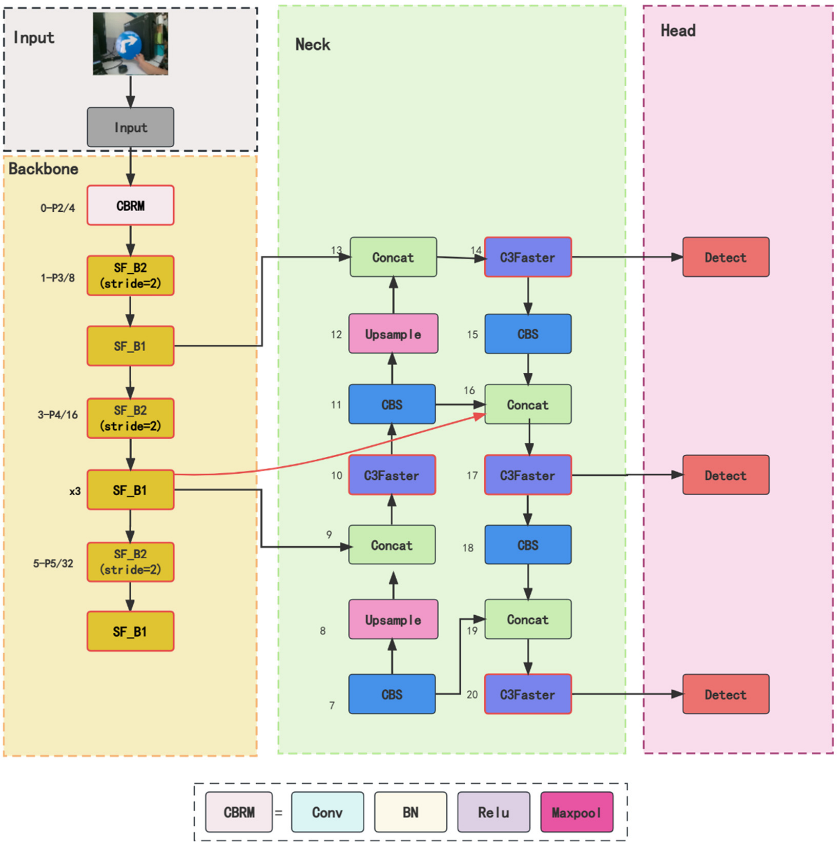


Figure 9. Improved network structure of Yolov5-SC3FB.

2.5. Introduction, Principle, Accuracy Test, and Alignment of Intel RealSense D455

In this paper, Intel Realsense D455, based on the principle of binocular IR structured light technology, is used to realize the detection and visual depth perception of road signs in the environment by the robot. The Intel Realsense D455 consists of an RGB sensor, a pair of stereo infrared sensors (IR Stereo Cameral), and an infrared laser emitter (IR Projector). An image of the Intel Realsense D455 is shown in Figure 10.

Figure 10. D455 camera physical diagram.

The Intel Realsense D455 combines the features of a structured light camera and a pure binocular camera. The infrared laser emits infrared "structured light" [35] to the surface of the object to be measured, and the receiver (left and right infrared cameras) receives different structured light images from different distances of the object to be measured, and then uses the binocular ranging principle to obtain the depth information of the target object. Despite having an infrared laser emitter, it does not use infrared reflection ranging. It serves only to project invisible fixed infrared texture patterns, such as infrared scatter, to improve the accuracy of depth calculation in environments where the texture is not obvious (e.g., white walls) and to assist binocular vision ranging. A schematic diagram of the binocular measurement principle is shown in Figure 11.

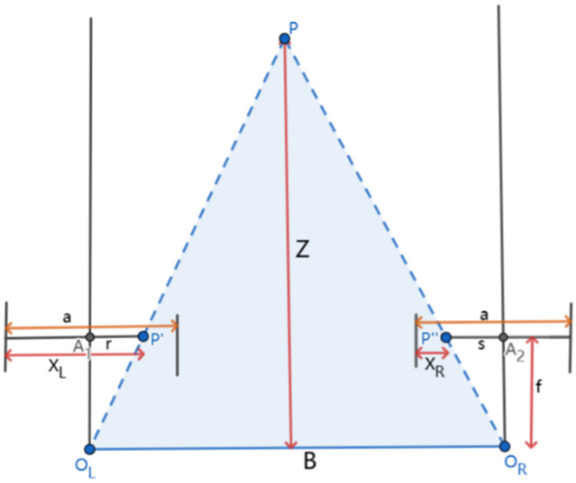


Figure 11. Binocular distance measurement schematic.

In Figure 11, the meaning of the points and lines are shown below.

Point P is the target point.

O_L, O_R is the optical center of the two stereo infrared cameras in Intel Realsense D455.

Between P' and P'' denotes the target point P passing through the left and right infrared cameras on the left and right images, and the corresponding coordinates on the X-axis are X_L and X_R, respectively.

B is the center distance (focal length) of the two stereo infrared cameras.

Z denotes the depth value of the target point P (the distance from point P to the baseline).

f indicates the focal length of the camera.

a indicates the width of the camera lens.

A_1, A_2 denote the lens midpoints A_1, A_2 of the left and right IR cameras, respectively.

r,s denote the distance from A_1, A_2 to P', P'', respectively.

The process of obtaining the depth value (distance) of the target (road sign) based on the binocular ranging principle is as follows. The distance between P' and P'' is calculated, and the distance between these two points is expressed by d, as shown in Equation (1).

$$\mathrm{d} = \mathrm{B} - (\mathrm{r} + \mathrm{s}) \tag{1}$$

$$\mathrm{X_L} = \frac{\mathrm{a}}{2} + \mathrm{r} \tag{2}$$

$$\mathrm{X_R} = \frac{\mathrm{a}}{2} - \mathrm{s} \tag{3}$$

Equations (2) and (3), to obtain Equation (4).

$$\mathrm{X_L} - \mathrm{X_R} = \mathrm{r} + \mathrm{s} \tag{4}$$

Equation (4) is combined with Equation (1) to obtain Equation (5).

$$\mathrm{d} = \mathrm{B} - (\mathrm{X_L} + \mathrm{X_R}) \tag{5}$$

Equation (6) is obtained according to the principle of similar triangles.

$$\frac{\mathrm{Z} - \mathrm{f}}{\mathrm{Z}} = \frac{\mathrm{d}}{\mathrm{B}} \tag{6}$$

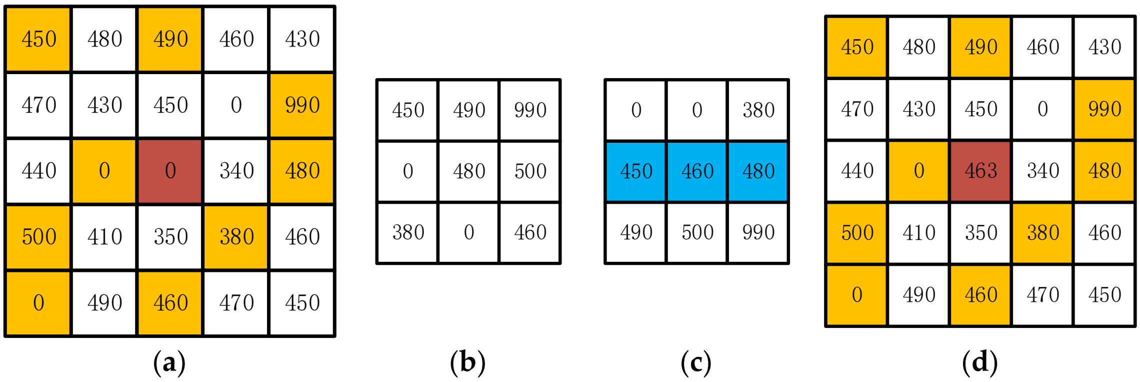
Rectifying Equation (6), the depth value Z of the target point can be obtained as shown in Equation (7).

$$\mathrm{Z} = \frac{\mathrm{Bf}}{\mathrm{d}} \tag{7}$$

Intel Realsense D455 cameras are calibrated at the factory. This ensures that users can get their hands on the device and use it. However, over time and due to factors such as the working environment, the accuracy of the camera may change if the camera is exposed to extreme temperatures or there are frequent shocks and vibrations for a long time. For this reason, Intel Realsense D455 needs to be tested for accuracy first. Since the RGB sensor and the stereo IR sensor of Intel Realsense D455 have different coordinate systems, the depth map and the color map do not correspond to each other and there is a certain error. Therefore, in order to obtain the actual depth values of the pixels in the color map, we need to transform the color map and the depth map to the same coordinate system, so that the pixel points in the color map and the pixel points in the depth map correspond to each other, that is, to complete the camera alignment.

2.6. Improved Filtering Method for Obtaining Depth Values of Centroids

In the process of acquiring the depth value using Intel RealSense D455 depth camera, the depth (distance) between the object under test and the D455 depth camera is usually measured using the center point of the object under test (road sign) as the standard. However, when acquiring the depth value, some interference factors or noise may cause the obtained depth value to be 0. If the depth value of the center point of the target bounding box detected using Intel Realsense D455 is 0 due to these interference factors or noise points in the Yolov5-SC3FB network, the visual depth perception capability of the mobile robot will be affected, thus affecting the decision-making process and the execution of subsequent actions of the mobile robot. To solve this problem, this study proposes to use the idea of filtering to improve the method of acquiring the depth value of the center point of the road sign bounding box. The improved method is shown in Figure 12.



450	480	490	460	430
470	430	450	0	990
440	0	0	340	480
500	410	350	380	460
0	490	460	470	450

(**a**)

450	490	990
0	480	500
380	0	460

(**b**)

0	0	380
450	460	480
490	500	990

(**c**)

450	480	490	460	430
470	430	450	0	990
440	0	463	340	480
500	410	350	380	460
0	490	460	470	450

(**d**)

Figure 12. Improved filtering method for obtaining depth values of centroids. (**a**) Bounding box, centroid and random points; (**b**) Random points taken out; (**c**) Sorting to take the intermediate points; (**d**) Find the average Z_1, Z_1 instead of Z_0.

In Figure 12a, the large square represents the target bounding box, and the small square inside the large square represents some pixel points inside this bounding box. In Figure 12a, the small red square represents the center point of the bounding box, and its depth value is expressed by Z_0, which can be seen as 0. Some random pixel points are selected around the center point of the bounding box (the points cannot be taken beyond the bounding box), and the depth value of these random points is back-checked according to the depth index of these points. The small yellow squares in Figure 12a are the randomly selected pixel points and their corresponding depth values. As shown in Figure 12b, the depth values of these random points are placed in an empty depth list in turn. In Figure 12c, these points are sorted according to the size of the depth value of the random points. Then, the depth value of the random points in the middle of the list (represented by the small blue square) is taken and the average value of their depth values is Z_1, as shown in Figure 12d. Z_1 is used instead of Z_0 to effectively alleviate the problem of the depth value of 0. From Figure 12d, it can be seen that the value of Z_1 is 463 (preserving integers).

In the above improved method, the sorting is to eliminate some points with abnormal depth values, so as to get more accurate depth values. Taking the depth value of a random point in the middle section instead of a single point is to further reduce the effect of outliers and make the depth value more reliable.

2.7. Evaluation Indicators

2.7.1. Evaluation Indicators of the Model

(1) Metrics to measure whether the model is lightweight

In this paper, we will evaluate whether a model is lightweight in terms of the number of parameters (Parameter), the amount of computation FLOPs, and the size of the model volume (size of model weights). The number of parameters refers to the sum of parameters in the network model. It is used to describe the size of the network model; it is similar to the space complexity in algorithms, and affects the memory occupation. Computational quantity refers to the number of floating-point operations per second. It refers to the number of multiplications and additions performed by the network model when performing forward inference. It is used to describe the execution efficiency of the model, similar to the time complexity in an algorithm. In deep learning, the smaller the number of parameters and computation of the model, the less time and resources the model needs to consume when performing inference. The model weight file is the file needed for the final deployment, and the model size (model weight file) is as small as possible, which facilitates deployment in devices with limited space resources.

(2) Evaluation index of model detection effect

In this paper, we use mAP (mean average precision) as an accuracy evaluation metric. AP (average precision) measures how good the trained model is at detecting a certain category, and mAP measures how good the trained model is at detecting all categories. Assuming that there are k categories and k > 1, the formula for mAP is given in Equation (8).

$$\mathrm{mAP} = \frac{\sum_{i=1}^{k} \mathrm{AP_i}}{k} \tag{8}$$

This study will use mAP@0.5 and mAP@0.5:0.95 to evaluate the detection effect of the model: mAP@0.5 denotes the average detection accuracy for all categories at an IOU threshold of 0.5; mAP@0.5:0.95 denotes the average mAP over different thresholds (IOU threshold settings range from 0.5 to 0.95, in steps of 0.05).

(3) Model speed evaluation index

When deploying a model to a mobile device or an embedded device, it is necessary to consider the dynamic detection speed of the model with a camera attached to the mobile device, in addition to the number of parameters, computation, size, and accuracy of the model. In this paper, we use FPS (Frames Per Second) to evaluate the detection speed of the model after deploying the model to the CPU side (IPC) of the upper computer of the mobile robot platform.

2.7.2. Accuracy Evaluation Index of the Measured Depth Values

Depth Error and Depth Relative Error are used to measure the accuracy of the depth values measured by the "improved filtering method for obtaining depth values of centroids" proposed in this paper.

Depth Error: It indicates the difference between the measured depth and the true depth. It can be calculated by Equation (9).

$$\text{Depth Error } = \left|\mathrm{Depth_{Measured}} - \mathrm{Depth_{True}}\right| \tag{9}$$

Depth Relative Error: It indicates the ratio between the difference between the measured depth and the true depth to the true depth. It can be calculated by Equation (10).

$$\text{Depth Relative Error } = \left|\frac{\mathrm{Depth_{Measured}} - \mathrm{Depth_{True}}}{\mathrm{Depth_{True}}}\right| \tag{10}$$

2.8. Mobile Robot Platform

In this paper, the fused Yolov5-SC3FB road sign detection model and the improved filtering method for obtaining centroid depth values are validated on a mobile robot platform developed in our laboratory. The mobile robot includes an actuator, a drive system, a control system, and a vision system. Their specific components are shown in Table 2. Among them, the maxon RE35 DC motor, the DC motor drive unit RMDS-108, and the drive control unit Arduino mega 2560 are packaged inside the mobile robot chassis.

Table 2. Specific components of the mobile robot.

Name of Each Part of the Robot	The Specific Composition of Each Part
actuator	Maxon RE35 DC motor Mecanum wheel
drive system	DC Motor Drive Unit RMDS-108 Driver control unit Arduino mega 2560
control system	IPC (Industrial Personal Computer)
vision system	Intel RealSense D455

The physical diagram of the mobile robot is shown in Figure 13.

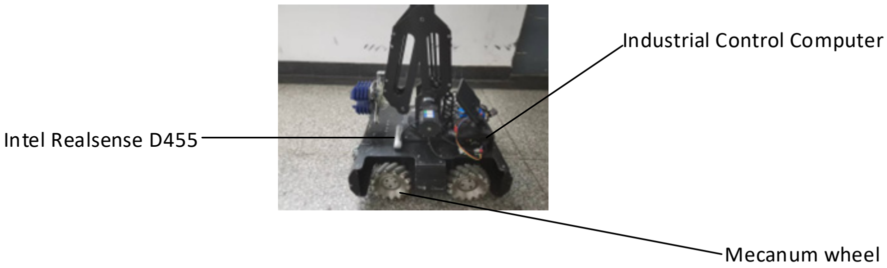


Figure 13. Mobile robotics platform.

The experimental scheme of the mobile robot to achieve autonomous running in this paper is shown in Figure 14. The Yolov5-SC3FB model proposed in this paper is used to train in the constructed road sign sample dataset using igher performance computers. The training results are analyzed, and the trained weight results are deployed in the IPC. The IPC uses the trained weight results to acquire the road signs in real time using Intel Realsense D455 depth camera to complete the road sign detection function. At the same time, Intel Realsense D455 measures the depth (distance) from D455 to the road sign using the binocular ranging principle and the improved filtering method for the center point depth value described above. To prevent collision between the mobile robot and the road sign, a threshold value of 0.4 m is set. When the distance between the mobile robot and the road sign is greater than 0.4 m, the mobile robot travels straight. When the distance between the mobile robot and the road sign reaches 0.4 m, the category information and depth information (distance) of the road sign detected by the IPC (upper computer) is sent to the lower computer Arduino main controller by means of serial communication. According to the sent command, the Arduino main controller controls the motor movement. Then, the motor drives the four McNamee wheels of the mobile robot to rotate, thus controlling the mobile robot to make corresponding movements (turn left, turn right, backward, stop) and realize the autonomous running of the mobile robot.

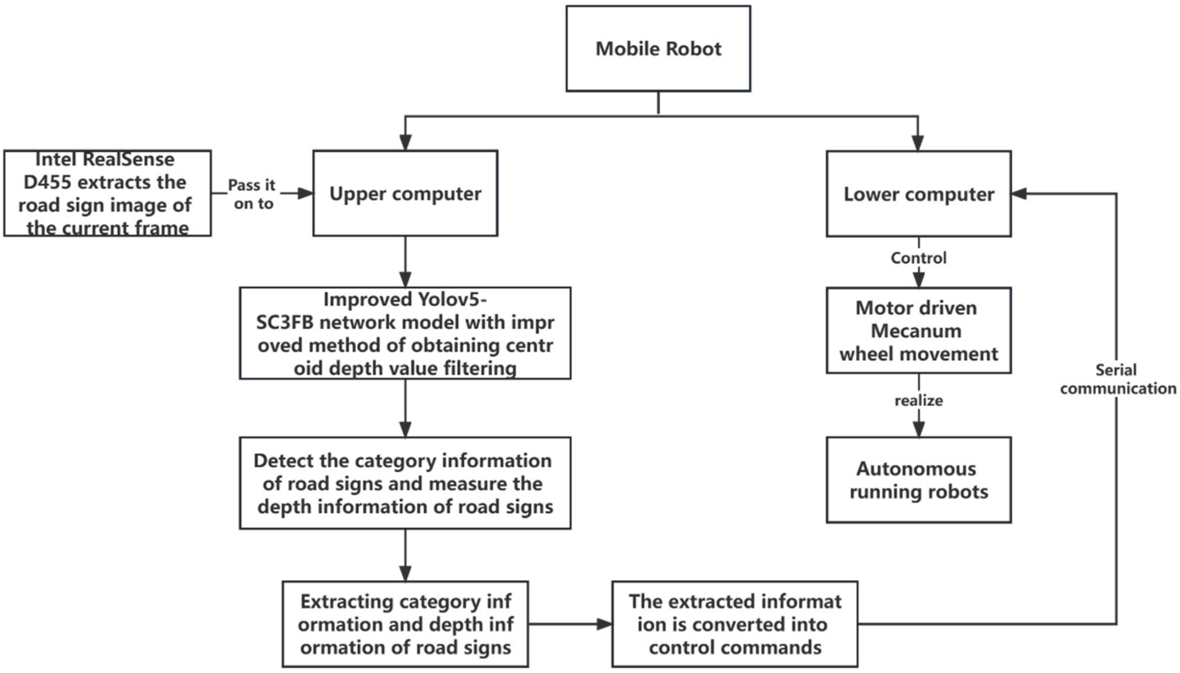

Mobile Robot
Intel RealSense D455 extracts the road sign image of the current frame
Pass it on to
Upper computer
Lower computer
Control
Improved Yolov5-SC3FB network model with impr oved method of obtaining centr oid depth value filtering
Motor driven Mecanum wheel movement
realize
Serial communication
Detect the category information of road signs and measure the depth information of road signs
Autonomous running robots
Extracting category inf ormation and depth inf ormation of road signs
The extracted informat ion is converted into control commands

Figure 14. Experimental scheme of autonomous running of mobile robot.

3. Results and Analysis

3.1. Experiments and Analysis of the Improved Yolov5-SC3FB Model

3.1.1. Model Training Platform

The environment configuration of the model training platform used in this study is shown in Table 3.

Table 3. Training environment configuration.

GPU	NVIDIA GeForce RTX 2080 Super
CPU	IntelI CITM) i9-10900K CPU @ 3.70 GHz
Video Card Memory	8 G
Memory	16 GB
Operation System	Ubuntu 18.04.1
CUDA version	10.2
CUDNN version	8.3.0

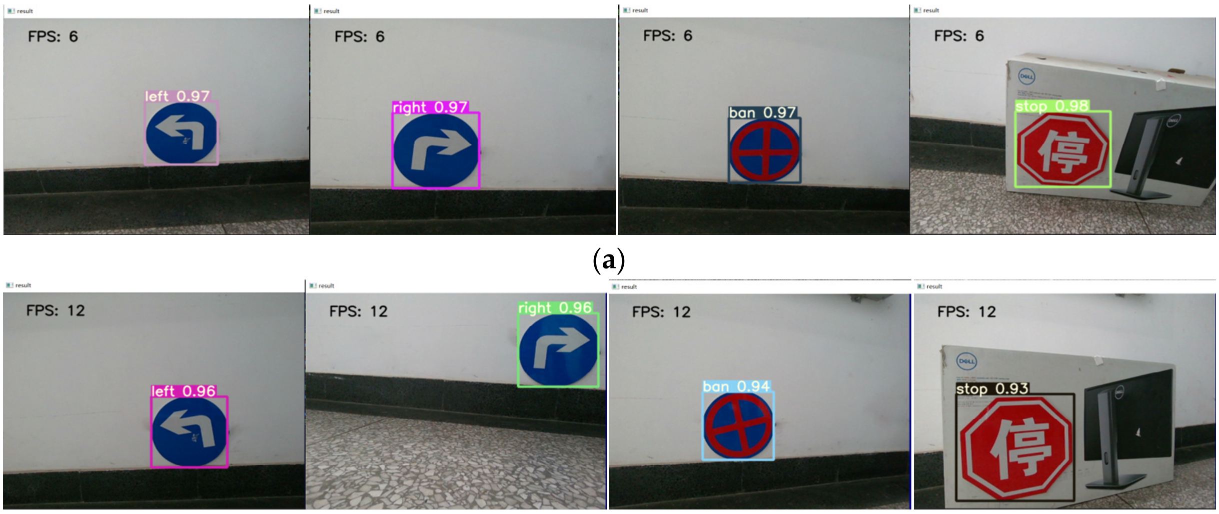
3.1.2. Network Performance Analysis

First, Yolov5 introduces the reduced ShuffleNetV2 to reconstruct the Yolov5 feature extraction network (strategy A). Then, the C3 module is replaced with the C3Faster module constructed in this paper in the feature fusion network of Yolov5 (strategy B). Strategy B reduces the number of parameters and computation of the network. Feature fusion using a simplified BiFPN (strategy C) improves the detection accuracy of the model, and it introduces almost no number of parameters or computations. The results of the ablation experiments on the road sign dataset constructed in this paper are shown in Table 4.

Table 4. Results of ablation experiments.

Experiment	Strategy A	Strategy B	Strategy C	Parameters (M)	GFLOPS	Size (MB)	mAP@0.5	mAP@0.5:0.95
Yolov5n				1.76	4.1	3.9	0.995	0.958
Experiment 1	√			0.22	0.5	0.78	0.994	0.901
Experiment 2		√		1.63	3.9	3.6	0.995	0.966
Experiment 3			√	1.78	4.2	3.9	0.995	0.971
Ours	√	√	√	0.19	0.5	0.72	0.995	0.898

From the data in Table 4, it can be seen that using different strategies for road sign detection is effective, and these methods are added simultaneously to achieve the best balance between model size and detection accuracy. The Yolov5-SC3FB model proposed in this paper reduces Parameters by 89%, GFLOPS by 88%, and Size by 82% compared to the Yolov5n model. The detection speed of the original YOLOv5n model is improved by effectively achieving a lightweight YOLOv5n while keeping the detection accuracy mAP@0.5 almost unchanged and mAP@0.5:0.95 down by only 6%. The YOLOv5n model and the Yolov5-SC3FB model are deployed on the upper computer (IPC) and CPU side of the mobile robot platform, respectively, and the detection speed of the original YOLOv5n model is improved by using Intel Realsense D455 for dynamic detection. The detection speed and detection effect are shown in Figure 15. Figure 15a shows the dynamic detection speed and detection effect of YOLOv5n model. Figure 15b shows the dynamic detection speed and detection effect of the Yolov5-SC3FB model. From Figure 15, it can be seen that the dynamic detection accuracy of the Yolov5-SC3FB model is slightly reduced compared with the original Yolov5n model at the CPU side of the upper computer (IPC) of the mobile robot platform. However, the detection speed is increased from 6 FPS to 12 FPS. Detection speed is increased by 2 times.



(**a**)

(**b**)

Figure 15. Dynamic detection speed and detection effect of the model. (**a**) Dynamic detection speed and detection effect of YOLOv5n model; (**b**) Dynamic detection speed and detection effect of Yolov5-SC3FB model.

The improved Yolov5-SC3FB model training process is visualized in Figure 16.

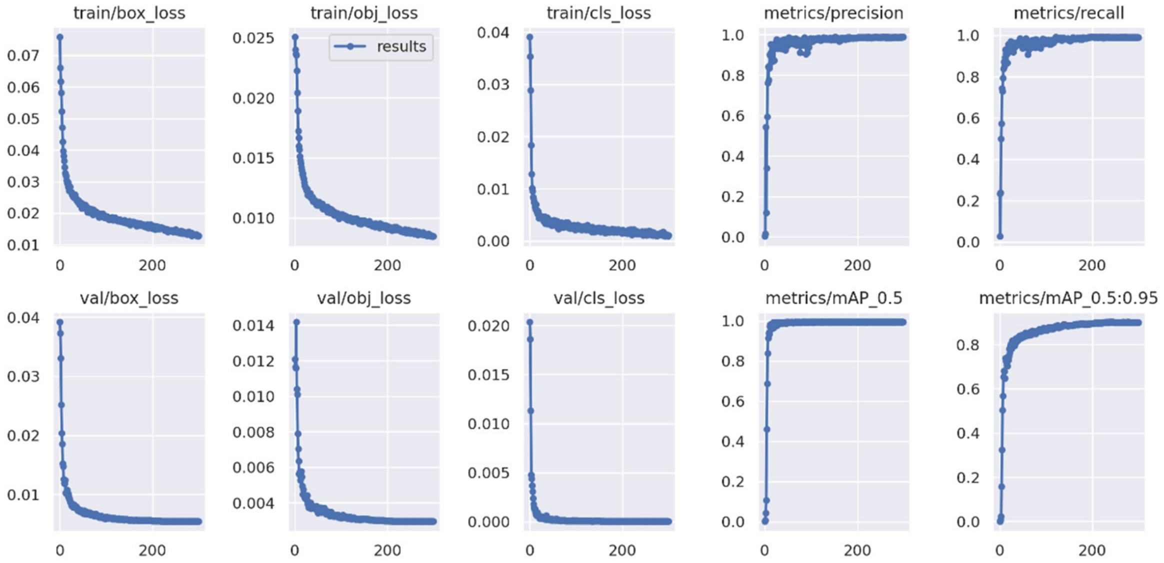


Figure 16. Visualization of the training process of Yolov5-SC3FB model.

3.2. *Intel Realsense D455 Camera Testing and Camera Alignment Experiment*

3.2.1. Camera Accuracy Test Experiment

In this article, we use Intel Realsense D455 camera, USB3.0 cable, tripod, and laptop for the Intel Realsense D455 accuracy test. We use the Depth Quality Tool (version: SDK 2.0) for Intel RealSense Cameras software to check if the camera accuracy is good. The process is as follows. Find a white wall as a target and align the Intel Realsense D455 depth camera horizontally to the wall. The accuracy test procedure of Intel Realsense D455 is shown in Figure 17. Measure the depth (distance) using Intel Realsense D455 as shown in Figure 17a. Adjust the relevant parameters on the left side of the "Depth Quality Tool for Intel RealSense Cameras" software. Check "Ground Truth", fill in the actual true distance measured physically using a meter ruler, manually adjust Angle to stay within 5 degrees, and measure the vertical distance of Intel Realsense D455 from the wall. The actual real distance of the Intel Realsense D455 from the wall is measured by manually adjusting the Angle to within 5 degrees. The process of measuring the depth using a meterstick is shown in Figure 17b.

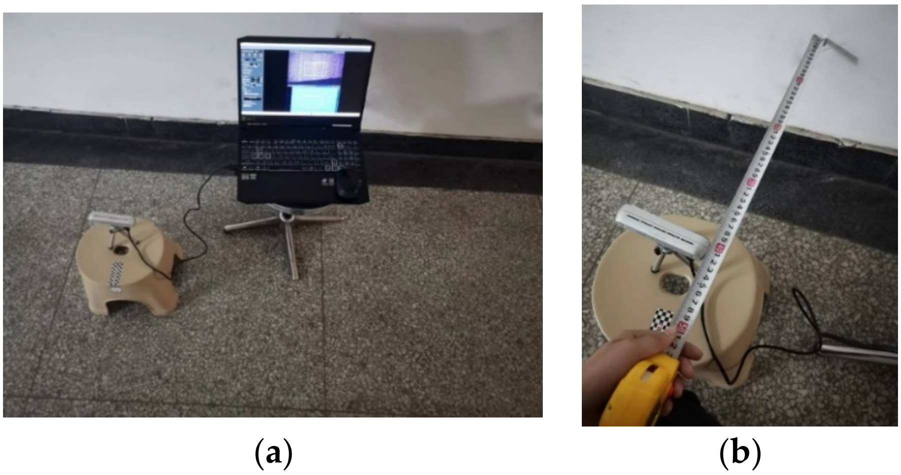


Figure 17. Intel Realsense D455 accuracy testing process. (**a**) Intel Realsense D455 measuring depth; (**b**) Metric ruler to measure depth.

The Depth Quality Tool for Intel RealSense Cameras software returned values for Fill-Rate, Z-Accuracy, and Plane Fit RMS Error, and some of the measured results are shown in Table 5.

Table 5. Selected results of measurements.

Measuring Distance (mm)	Actual Physical Distance (mm)	Z Accuracy (%)	Fill-Rate (%)	Plane Fit RMS Error (%)	Angle (Deg)
391.73	400	−1.77	99.99	0.10	3.51
502.21	500	0.91	100.00	0.10	4.07
600.06	600	0.29	99.99	0.09	1.91
702.74	700	0.4	99.99	0.10	2.23
805.58	800	0.74	100.00	0.13	0.41
906.18	900	0.76	99.99	0.11	0.40
1007.75	1000	0.84	99.99	0.21	0.76
1103.03	1100	1.46	100.00	0.25	3.92
1216.41	1200	1.58	99.99%	0.43	2.96

According to Intel's official instructions, analysis of Table 4 shows that the Z-axis error (distance error) is less than 2%, the fill rate is greater than or equal to 99.99%, and the RMS error is less than or equal to 2%, which meets the factory specifications of the D455. Therefore, the accuracy of the D455 camera used is relatively accurate, and there is no need for self-calibration of the camera.

3.2.2. Camera Alignment Experiments

Use Python to call the camera's API, the environment required to configure the Intel Realsense D455 depth camera. Set the resolution of the color stream and depth stream of the input depth camera to 640 × 480. Perform the alignment operation on the obtained depth image data and color image data and align the depth map with the color map. The aligned color map and depth map as after the alignment operation are shown in Figure 18.

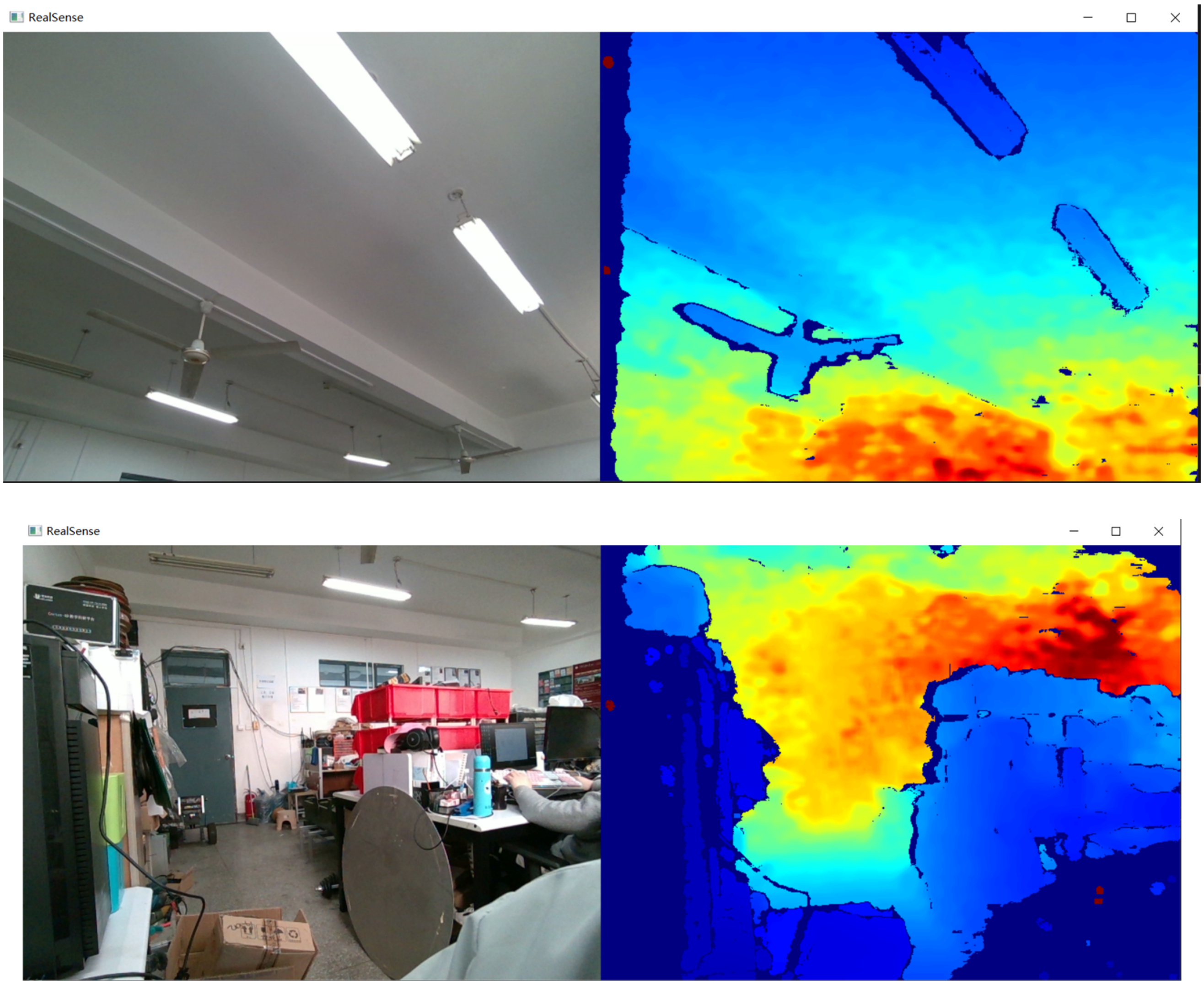


Figure 18. Color map and depth map after alignment.

3.3. Experiment and Error Analysis of the Improved Filtering Method for Obtaining the Depth Value of the Centroid

The depth values of the road signs were measured using an improved filtering method for obtaining the depth values of the centroids. The true depth values of the road signs and the measured depth values, the depth error, and the depth relative error data are shown in Table 6. The depth relative error in Table 6 is retained to two decimal places.

Table 6. Depth measurement and error analysis.

True Depth Value (mm)	Measured Depth Value (mm)	Depth Error (mm)	Depth Relative Error
300	304.23	4.23	1.41%
400	405.28	5.28	1.32%
500	506.36	6.36	1.27%
600	605.72	5.72	0.95%
700	706.43	6.43	0.92%
800	806.37	6.37	0.80%
900	907.82	7.82	0.87%

Analysis of the results in Table 6 shows that the relative error in depth is within 2%. Therefore, the improved filtering method for obtaining the depth value of the center point can measure the depth value of the road signs more accurately.

3.4. Fusion of Yolov5-SC3FB road Sign Detection Model with Improved Filtering Method for Obtaining Centroid Depth Values

The experimental results of fusing the Yolov5-SC3FB road sign detection model with the improved method of obtaining centroid depth value filtering are shown in Figure 19. It can be seen from Figure 19 that the fused algorithm is able to acquire the category information, confidence information, and depth information of road signs simultaneously, which provides an environment-aware solution for the autonomous running of mobile robots.

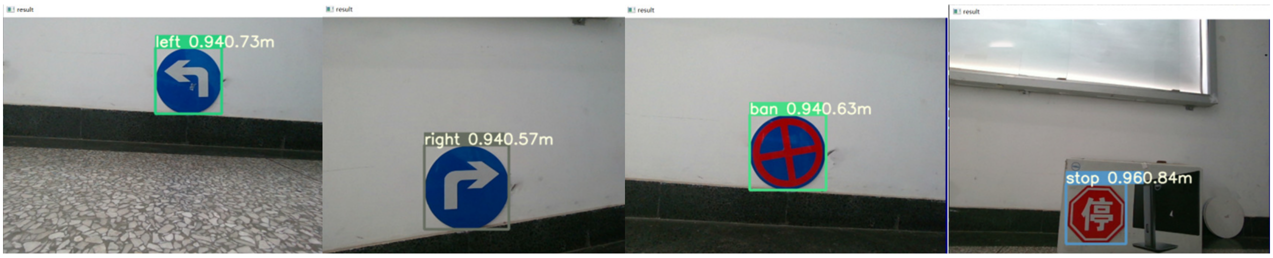


Figure 19. Experimental results after fusion algorithm.

3.5. Mobile Robot Road Sign Detection and Depth Perception Experiment

Extract the category information and depth information of the road signs obtained from the fused algorithm. Implement the communication between the host computer and the Arduino main controller through the serial port. This information is converted into control commands. Thus, the mobile robot completes the function of detection and depth perception of road signs and realizes the autonomous driving, as shown in Table 7.

Table 7. Robot autonomous running state and corresponding control commands.

Extraction of Detected Road Sign Category Information	Extraction of the Acquired Road Sign Depth Information	Control Commands Sent by the Serial Port	Autonomous Running State of the Robot
left	>0.4 m	No command	Go straight
	≤0.4 m	‘l’	Turn left
right	>0.4 m	No command	Go straight
	≤0.4 m	‘r’	Turn right
ban	>0.4 m	No command	Go straight
	≤0.4 m	‘b’	Backward
stop	>0.4 m	No command	Go straight
	≤0.4 m	‘s’	Stop

The autonomous running process of the mobile robot is shown in Figure 20. When the speed of detecting road signs is increased, the results of detection (detection of ‘left’, ‘right’, ‘ban’, ‘stop’) are then translated more quickly into control commands (‘l’, ‘r’, ‘b’, ‘s’). If the control commands can be sent to the Arduino host controller faster, then the Arduino host controller will control the movement of the McNamee wheel by controlling the motor motion faster, which affects the robot’s ability to avoid obstacles (road signs) in time.

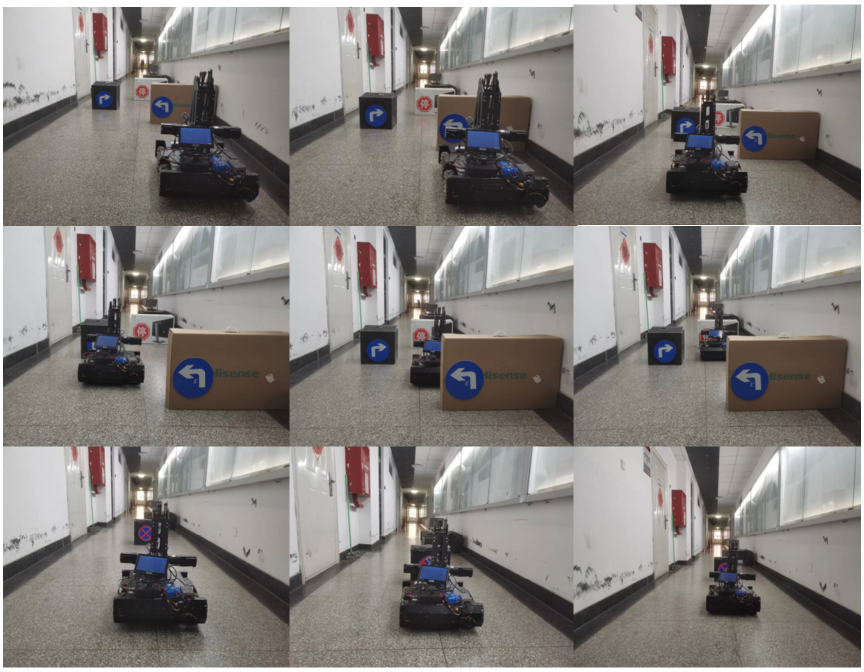


Figure 20. Autonomous running process of mobile robot.

4. Discussion and Conclusions

In this paper, we propose a target (road signs) detection algorithm Yolov5-SC3FB that can be used for indoor mobile robots. The feature extraction network of YOLOv5 is redesigned using the reduced lightweight network ShuffleNetV2. The C3 module in the original Yolov5 feature fusion network is replaced by using the constructed C3Faster module, and the simplified BiFPN is used for feature fusion in the original Yolov5 feature fusion part. In this study, the Yolov5-SC3FB model is experimentally validated in terms of lightweight metrics, detection accuracy, and detection speed of the CPU side deployed on the mobile robot platform. The experimental results show that the proposed Yolov5-SC3FB reduces the amount of model parameters by 89%, the amount of computation by 88%, and the model size by 82% compared with Yolov5n. Deploying the Yolov5-SC3FB model on the CPU side of the upper computer of the mobile robot platform loses less detection accuracy and increases the detection speed by a factor of 2.

In addition, this study proposes an improved filtering method to obtain the depth value of the center point of the bounding box. The experimental results show that this method can effectively alleviate the problem of obtaining the depth value of the center point of the bounding box as 0 due to the interference of noise points, and the relative error of depth is only 2%. Finally, this study fuses the Yolov5-SC3FB model with the improved filtering method for obtaining centroid depth values to obtain both category and depth information of road signs, and deploys the fusion algorithm on a mobile robot platform to empower the mobile robot with environment sensing capability. Finally, the category information and depth information of road signs are extracted and transformed into control commands. The mobile robot can accurately avoid road signs under the set distance threshold and achieve autonomous running according to the instructions of road signs. The upper computer platform for the mobile robot in this paper mainly uses an IPC, which has no GPU but only a CPU. Therefore, the final detection speed still has a gap with the real-time detection speed. Future work will use embedded devices with GPU such as Jetson Nano and Jetson TX2 as the upper computer system for the mobile robot platform and use TensorRT acceleration, enabling the robot to detect road signs in real time and react in a more timely manner based on the detection results.

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

Funding: This research was funded by China Association of Higher Education, grant number 21GCYB04; National Social Science Foundation of China, grant number BIA200191.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nillson, N.J. Shakey the Robot. *SRI Int. Tech. Note* **1984**, 323. Available online: https://www.mendeley.com/catalogue/e16fcde6-d477-3f1c-ab20-8f0505435aa5/ (accessed on 14 July 2023).
2. Matijevic, J. Sojourner: The Mars pathfinder microrover flight experiment. *Space Technol.* **1997**, *3*, 143–149. [CrossRef]
3. Maisonnier, B.; Magnenat, S. NAO: A Humanoid Robot Platform for Autonomy and Interactive Social Robotics. *Adv. Robot.* **2008**, *22*, 1233–1264.
4. Abubakkar, A.; Achuthan, R.; Gowri Shankar, S.; Yeasigan, S.K.; Manoj Kumar, K. Design and fabrication of mobile app-controlled floor sweeper. *Mater. Today Proc.* **2022**, *55*, 365–369.
5. Seo, S.; Jung, H. A robust collision prediction and detection method based on neural network for autonomous delivery robots. *ETRI J.* **2023**, *45*, 329–337. [CrossRef]
6. Li, S.F.; Zheng, P.; Liu, S.C.; Wang, Z.X.; Wang, X.V.; Zhang, L.Y.; Wang, L.H. Proactive human–robot collaboration: Mutual-cognitive, predictable, and self-organising perspectives. *Robot. Comput.-Integr. Manuf.* **2023**, *81*, 102510. [CrossRef]
7. Wang, X.; Guo, J.; Yi, J.L.; Song, Y.C.; Xu, J.D.; Yan, W.Q.; Fu, X. Real-time and efficient multi-scale traffic sign detection method for driverless cars. *Sensors* **2022**, *22*, 6930. [CrossRef]
8. Shimada, T.; Nishikawa, H.; Kong, X.; Tomiyama, H. Fast and High-Quality Monocular Depth Estimation with Optical Flow for Autonomous Drones. *Drones* **2023**, *7*, 134. [CrossRef]
9. Park, H.J.; Kim, K.B. Depth image correction for Intel® realsense depth camera. *Indones. J. Electr. Eng. Comput. Sci.* **2020**, *19*, 1021–1027. [CrossRef]
10. Jiang, J.H.; Jiang, Y.H.; Zhang, L. Workpiece detection and localization system based on neural network and depth camera. *Trans. Sens. Microsyst.* **2020**, *39*, 82–85.
11. Guo, N.B.; Chen, X.Y.; Xue, J.S. Research Overview of Infrared Camera Based on Time of Flight. *Trans. J. Ordnance Equip. Eng.* **2017**, *38*, 152–159.
12. Ma, X.; Shu, B.L.; Li, J.C. Binocular Stereo Vision Ranging Technology. *Trans. Electron. Des. Eng.* **2016**, *24*, 81–83.
13. Chen, Y.J.; Zuo, W.M.; Wang, K.Q.; Wu, Q.F. Survey on Structured Light Pattern Codification Methods. *Trans. Mini-Micro Syst.* **2010**, *31*, 1856–1863.
14. Sai, D. Intel will launch immersive and user-friendly interactive devices in 2014. *Trans. Appl. IC* **2014**, 20–21. (In Chinese) [CrossRef]
15. Tadic, V.; Odry, A.; Kecskes, I.; Burkus, E. Application of Intel realsense cameras for depth image generation in robotics. *WSEAS Transac. Comput* **2019**, *18*, 2224–2872.
16. Bayer, J.; Faigl, J. On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265. In Proceedings of the European Conference on Mobile Robots (ECMR), Prague, Czech Republic, 4–6 September 2019.
17. Ahluwalia, V.; Arents, J.; Oraby, A.; Greitans, M. Construction and benchmark of an autonomous tracked mobile robot system. *Robot. Syst. Appl.* **2022**, *2*, 15–28. [CrossRef]
18. Yoshida, T.; Kawahara, T.; Fukao, T. Fruit recognition method for a harvesting robot with RGB-D cameras. *ROBOMECH J.* **2022**, *9*, 15. [CrossRef]
19. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
20. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 18 April 2015.
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1440–1448. [CrossRef]
22. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
23. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
24. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
25. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.

26. Li, C.Y.; Li, L.L.; Jiang, H.L.; Weng, K.H.; Geng, Y.F.; Li, L.; Ke, Z.D.; Li, Q.Y.; Cheng, M.; Nie, W.Q.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
27. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
28. Li, C.Y.; Li, L.L.; Geng, Y.F.; Jiang, H.L.; Cheng, M.; Zhang, B.; Ke, Z.D.; Xu, X.M.; Chu, X.X. YOLOv6 v3. 0: A Full-Scale Reloading. *arXiv* **2023**, arXiv:2301.05586.
29. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Amsterdam, The Netherlands, 2016.
30. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
31. Mei, Y.Q.; Fan, Y.C.; Zhang, Y.L.; Yu, J.H.; Zhou, Y.Q.; Liu, D.; Fu, Y.; Huang, T.S.; Shi, H. Pyramid attention networks for image restoration. *arXiv* **2020**, arXiv:2004.13824.
32. 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.
33. Chen, J.; Kao, S.H.; He, H.; Zhuo, W.P.; Wen, S.; Lee, C.H.; Gary Chan, S.H. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. *arXiv* **2023**, arXiv:2303.03667.
34. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
35. Dong, C.X.; Si, Z.J.; Liu, J.D. Design of Offset Machine Perception Display System Based on Intel RealSense. *Trans. Packag. Eng.* **2017**, *38*, 204–208.

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.