

Design of Indoor Security Robot based on Robot Operating System

Faxu He, Liye Zhang*

School of Computer Science and Technology, Shandong University of Technology, Zibo, China

Email: *zhangliye@sdut.edu.cn

How to cite this paper: He, F.X. and Zhang, L.Y. (2023) Design of Indoor Security Robot based on Robot Operating System. *Journal of Computer and Communications*, 11, 93-107.

<https://doi.org/10.4236/jcc.2023.115008>

Received: April 23, 2023

Accepted: May 26, 2023

Published: May 29, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution-NonCommercial International License (CC BY-NC 4.0).

<http://creativecommons.org/licenses/by-nc/4.0/>



Open Access

Abstract

The design and implementation of indoor security robot can well integrate the two fields of indoor navigation and object detection, in order to achieve a more powerful robot system, the development of this project has certain theoretical research significance and practical application value. The project development is completed in ROS (Robot Operating System). The main tools or frameworks used include AMCL (Adaptive Monte Carlo Localization) package, SLAM (Simultaneous Localization and Mapping) algorithm, Darknet deep learning framework, YOLOv3 (You Only Look Once) algorithm, etc. The main development methods include odometer information fusion, coordinate transformation, localization and mapping, path planning, YOLOv3 model training, function package configuration and deployment. Indoor security robot has two main functions: first, it can complete real-time localization, mapping and navigation of indoor environment through sensors such as lidar and camera; Second, object detection is accomplished through USB camera. Through the detailed analysis and research of the functional design of the two modules, the expected function is finally realized, which can meet the daily use needs.

Keywords

Indoor Security Robot, Indoor Navigation, SLAM, Object Detection, YOLO

1. Introduction

Using indoor robots to solve problems in real life not only saves labor, but also conforms to the development of The Times. At present, indoor robots have been widely used in patrol security, monitoring, industrial production, smart home and other aspects, and researchers' exploration and design of it is still accelerating [1]. At present, the research of indoor robot mainly focuses on the localiza-

tion and autonomous navigation direction with SLAM algorithm as the core, and some researchers independently explore the object detection direction in computer vision, but there is a lack of work combining the two functions together. Therefore, the purpose of this paper is to realize the combination of robot autonomous navigation and object detection. Specifically, this paper proposes an indoor security robot architecture combining autonomous navigation module and object detection module, and designs and implements it. In this architecture, autonomous navigation module is the central module of the robot. In terms of links, it connects perception and motion control, making decisions based on tasks and perceived environmental information, and planning the robot's motion trajectory. As for the object detection module, in short, it is an extension module of the robot. Through the fine perception of the external environment of the module, it can realize many functions, such as external intrusion detection, employee identification and so on. Finally, the effectiveness of the two modules is verified by systematic testing. Both modules perform well in indoor environment.

In recent years, the exploration of indoor robot is a research hotspot in the field of robot. In terms of robot autonomous navigation, Zhang *et al.* collected video images of indoor environment by using cameras placed at fixed indoor locations, and designed real-time image analysis algorithms to achieve obstacle avoidance, robot position tracking and other functions [2]. Chen *et al.* studied the theory and implementation of a self-positioning algorithm for a single mobile robot based on odometer and laser sensor in a general indoor environment [3]. Yang *et al.* realized wireless communication between upper computer, indoor positioning system and mobile robot, as well as real-time positioning function of mobile robot [4]. Zhong *et al.* proposed to use image recognition technology to obtain the navigation deflection Angle of robot [5]. Guo *et al.* took a two-wheeled differential mobile robot equipped with ROS as the platform and used Lidar as the main sensor to solve the localization, raster map creation and navigation problems of the mobile robot [6]. Zeng *et al.* constructed a visual SLAM system based on point-line feature fusion by adding line feature to point feature [7]. Zheng *et al.* used the TEB (Timed Elastic Band) algorithm as a local path planning algorithm to propose and implement an indoor autonomous navigation system for mobile robots based on obstacle detection with depth camera [8].

In terms of robot indoor environment perception: Sun *et al.* proposed a target autonomous positioning and segmentation extraction algorithm, which independently obtains ROI (Region Of Interest) region and extracts and segments target point cloud according to the voxel-based method [7]. Li *et al.* combined deep learning method to study object detection algorithm based on 3D laser ranging sensor. The effectiveness of the proposed algorithm is proved by experiments [8]. In order to solve the imbalance and instability of dynamic tracking of preschool children's companion robot in indoor environment, Jin *et al.* proposed a human target dynamic tracking method for preschool children's com-

panion robot in indoor environment. The experimental results show that the design method has low error and high accuracy in dynamic tracking of human body target, which has certain application value [9].

Indoor security robot, as a development direction of robotics, is in essence an excellent intelligent system. It not only has good environmental perception ability and can make independent decisions, but also is equipped with robot vision function module to realize object detection related work [10]. This paper takes tracked robots as the research object, which is equipped with an ROS system. The development work of the system is completed through the design and integration of indoor navigation and object detection functions. Among them, how to use the robot sensor to construct the accurate two-dimensional raster map under the condition of low cost and realize the object detection through the high precision and fast neural network is the focus of this paper. This paper makes a detailed analysis and research on the functional design of the two modules, and finally makes the robot system can meet the basic use requirements.

The design and development of the indoor security robot is completed on the ROS platform. ROS, which was developed in 2007, is an open-source robot operating system that has gained popularity among robot developers. ROS adopts distributed architecture as its main design idea [11]. Software and functional modules of robots are regarded as nodes one by one, and the communication between nodes needs to be realized through topics. In this way, these nodes can be deployed in different systems or machines, and the advantages of distributed architecture can be reflected. A common tool used in ROS, 'rviz', is shown in **Figure 1**.

During the development of object detection function, the main development tools used are Darknet deep learning framework and OpenCV (open source

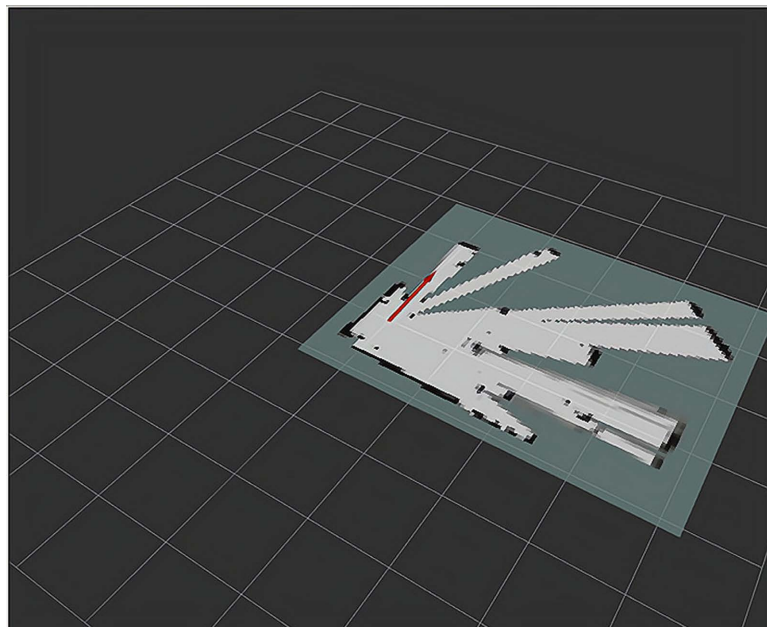


Figure 1. Common tools used in ROS.

computer vision library) [12]. Among them, Darknet is a relatively minority deep learning framework. Compared with TensorFlow and other frameworks, Darknet's functions are not very rich. However, this has become an advantage of Darknet, which is mainly reflected in the following aspects: 1) Easy to install. 2) No other dependencies are required. 3) The structure is clear. 4) Friendly Python interface. 5) The portability is good.

OpenCV uses C and C++ to complete the underlying writing, has good compatibility, can run on Windows, Linux and other operating systems. Light weight and high efficiency are the main advantages of OpenCV. At present, OpenCV has become a powerful and widely used research and processing tool in the field of computer vision [13]. OpenCV has a very wide range of applications, in machine vision, image classification, face recognition, video analysis and many other fields can see its shadow. In addition, OpenCV also has the advantage of real-time application and excellent performance, which provides an effective method to solve the problem in the real-time scenario of computer vision.

2. Design of Localization, Mapping and Navigation Functions

The main content of this chapter is to study the localization, mapping and navigation functions in indoor environment, using a tracked robot based on ROS as the research platform. The ROS version used is Melodic. In this version of ROS system, the software environment is built, and SLAM and other related technologies are completed in the indoor environment for the localization, mapping and indoor navigation of the security robot.

2.1. Laser SLAM Algorithm

SLAM, known in Chinese as synchronous localization and mapping, is designed to solve the localization and mapping problems of a moving robot without knowing the surrounding environment. The localization work of laser SLAM needs to go through several steps. Firstly, it is necessary to obtain the environmental point cloud data information, and then calculate the changes of radar distance and attitude by matching and comparing the information. In this way, the localization work is completed. Among them, the laser SLAM system is mainly composed of sensor data, visual odometer, back-end, map building and loop detection [14].

GMapping is the most widely used algorithm in 2D SLAM algorithm. It is based on particle filtering. If the algorithm wants to achieve a better result, it needs to be realized on the basis of a large number of particles, which will obviously make its calculation very complicated. In order to deal with this bad phenomenon, we need to optimize through some means. GMapping is proposed on the basis of RBPF algorithm, and its main innovation lies in the optimization of proposal distribution and selective resampling. GMapping algorithm has a small amount of computation but good accuracy. Although it is not suitable for the construction of large scene maps, it performs very well in the construction of

indoor small environment maps.

2.2. Coordinate Transformation Theory

A robot system usually has multiple 3D coordinate systems, and the correlation between these coordinate systems will change over time. According to the theory of robot kinematics, the correlative coordinate system is essential for robot localization [15]. The established correlative coordinate system is shown in **Figure 2**.

In **Figure 2**, W is the world coordinate system established according to the robot's surrounding environment; R is the local coordinate system established with the geometric center at the bottom of the robot as the origin; L is a local coordinate system based on the geometric center of lidar. P_{RW} represents the vector of real-time position of frame R in frame W ; P_{LR} is a vector representing the real-time position of frame L in frame R , and is a constant vector; P_{BL} represents the vector of real-time position of point cloud data B in coordinate system L .

Coordinate transformation is a very important link for robot localization and mapping. P_{LR} is known as a constant vector, while P_{BL} can be easily obtained by lidar. In this way, a new vector P_{BR} can be obtained by combining two vectors. Since vector P_{RW} is also easily obtained through inertial measurement units and motor encoders, the desired final coordinate P_{BW} can be obtained by combining vector P_{BR} with vector P_{RW} . The specific process of coordinate transformation is shown in **Figure 3**.

2.3. Detailed Design

For the self-localization and mapping function of the security robot, the acquisition of environmental point cloud information and odometer information is the

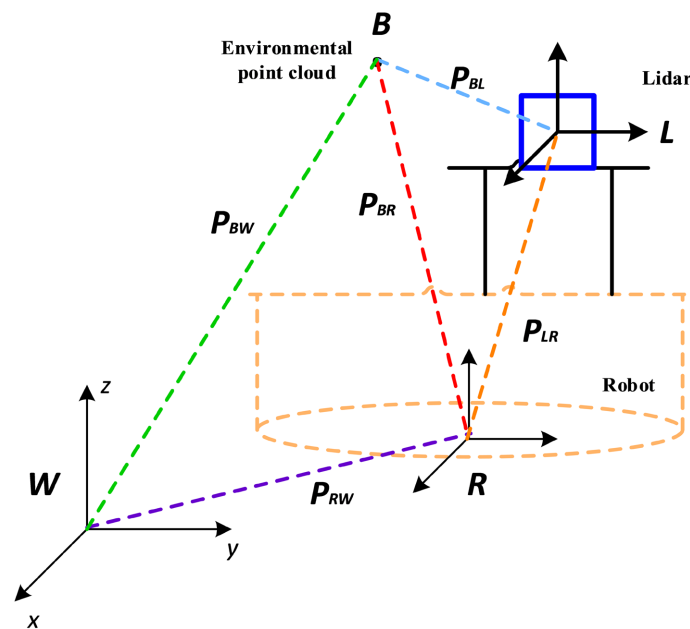


Figure 2. Robot correlative coordinate system.

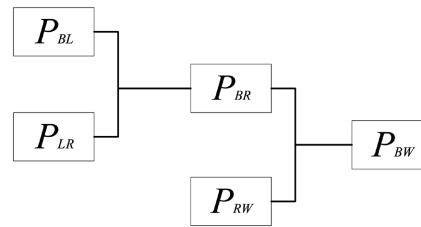


Figure 3. Coordinate transformation process.

most basic. The specific implementation process is as follows:

1) Odometer and imu (Inertial measurement unit) data fusion: The nine-axis gyroscope acquires the original imu data (acceleration, orientation angle, etc.) and transmits it to the drive board. The imu data is obtained through the imu data cleaning operation. At the same time, the motor encoder also returns the rotational speed information of the robot and obtains the original odometer information by integrating the linear velocity and angular velocity. Combining imu data and original odometer information, Kalman filter is used to get more accurate odometer information. Then the AMCL (Adaptive Monte Carlo Localization) is used to estimate and correct the position. The probability of the robot's position in the map is calculated by obtaining radar data and map data through particle filter estimation. The comprehensive odometer information can better obtain the localization information of the robot, which makes the localization of the car in the map more accurate.

2) Coordinate transformation: Since the position of the coordinate system R in the coordinate system W is known (origin or other positions), it is easy to obtain the value of the vector P_{RW} by calculating and transforming the coordinates according to the information such as speed and position during the robot's movement. Moreover, since the position of the coordinate system R is always in the center position at the bottom of the robot, the position of the robot in the coordinate system W is easy to obtain [16].

3) Robot localization and mapping: The robot first obtains the position information of the environmental point cloud in the coordinate system L by using Lidar, and then the information is further converted into the environmental point cloud information in the coordinate system W by coordinate transformation. After obtaining the environmental point cloud information, imu data, coordinate information obtained by coordinate transformation and odometer information obtained by data fusion, the robot releases the information. By subscribing to such information and synthesizing it, GMapping can build a two-dimensional raster map on the basis of such data information and realize self-localization operation at the same time. The specific process is shown in **Figure 4**.

4) Autonomous navigation: The most important navigation algorithm is the move_base path planning algorithm. It first subscribes to lidar, map and positioning information, then plans the global path and local path, and then transforms the path information into speed and other related information through

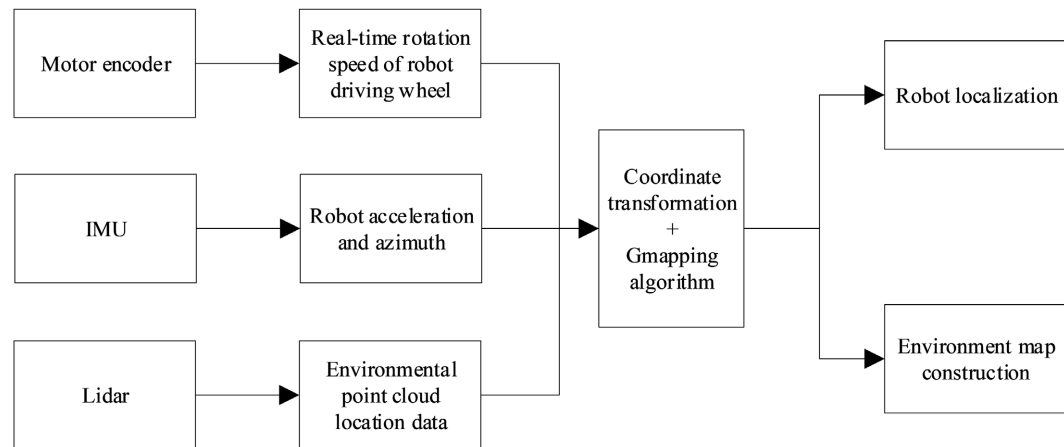


Figure 4. Flow chart of robot localization and mapping.

corresponding processing, and finally realizes robot navigation.

3. Design of Object Detection Function

In addition to autonomous positioning and navigation in the corresponding scene, the security robot also needs to complete the visual perception of the surrounding environment. This chapter describes the implementation process of the object detection function: the YOLOv3 model is trained by self-made data set, and it is applied to the darknet_ros function pack.

3.1. YOLO Object Detection Algorithm

YOLOv1, as the earliest version of YOLO series, was far faster than other algorithms at that time, although its detection accuracy was not very high and there was a limit of detection quantity when detecting small objects. YOLO combines detection problem into a regression problem, and only needs a convolutional neural network to realize end-to-end object detection, which is also the core idea of YOLOv1 [17]. Compared with Faster R-CNN and ResNet detection methods, YOLOv2 detection is faster. By mixing the data of detection data set and classification data set, this new joint training method can complete the expansion of classification and training set, which can greatly accelerate the detection speed [18]. YOLOv2 uses ImageNet classification data set to learn classification information, and COCO detection data set to learn object location detection. With the help of normalization and other technologies, the performance of YOLOv2 model has been greatly improved. The performance comparison between YOLOv2 model and neural networks such as Fast R-CNN is shown in **Table 1**.

YOLOv3 algorithm is the most commonly used object detection algorithm at present. Compared with YOLOv2, the Darknet-53 network is designed based on YOLOv3 algorithm, especially the main network part of the algorithm, and the FPN structure design is realized at the same time. YOLOv3 not only maintains a much faster speed than other algorithms, but also far exceeds other One-stage algorithms in terms of recognition accuracy, especially the detection ability of

Table 1. Performance comparison between YOLOv2 and other neural networks.

Model	Data and Performance		
	Training set	Mean average accuracy/%	Speed of detection/FPS
Fast R-CNN	VOC 2007 + 2012	70.0	0.5
Faster R-CNN VGG-16	VOC 2007 + 2012	73.2	7
Faster R-CNN ResNET	VOC 2007 + 2012	76.4	5
YOLO	VOC 2007 + 2012	63.4	45
SSD300	VOC 2007 + 2012	74.3	46
SSD500	VOC 2007 + 2012	76.8	19
YOLOv2 288 × 288	VOC 2007 + 2012	69.0	91
YOLOv2 352 × 352	VOC 2007 + 2012	73.7	81
YOLOv2 416 × 416	VOC 2007 + 2012	76.8	67
YOLOv2 480 × 480	VOC 2007 + 2012	77.8	59
YOLOv2 544 × 544	VOC 2007 + 2012	78.6	40

small objects has been greatly improved [19]. The YOLOv3 loss function is as follows,

$$\begin{aligned}
L = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{obj} \left[(b_x - \hat{b}_x)^2 + (b_y - \hat{b}_y)^2 + (b_w - \hat{b}_w)^2 + (b_h - \hat{b}_h)^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{obj} \left[-\log(p_c) + \sum_{i=1}^n \text{BCE}(\hat{c}_i, c_i) \right] \\
& + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{i,j}^{noobj} \left[-\log(1 - p_c) \right],
\end{aligned} \tag{1}$$

where: S is the number of grids, that is, S^2 is $13 * 13$, $26 * 26$ and $52 * 52$; B is for box; $1_{i,j}^{obj}$ means that if the box has a target object, its value is 1, otherwise it is 0; $1_{i,j}^{noobj}$ means that if box has no object, its value is 1, otherwise it is 0; BCE (Binary Cross Entropy) is calculated as follows,

$$\text{BCE}(\hat{c}_i, c_i) = -\hat{c}_i * \log(c_i) - (1 - \hat{c}_i) * \log(1 - c_i), \tag{2}$$

3.2. Model Training

To realize the object detection function, the Darknet framework should be installed first. Meanwhile, CUDA and OpenCV should be installed by modifying relevant parameters in the makefile file. Then, download the pre-training model on the official website of YOLO. After the download is complete, run the program, and then observe the entire network structure and the output. Darknet uses two txt files as training data, one is used to record paths and the other is used to record the location of marked targets. However, before generating them, it is necessary to ensure that there are well marked VOC data. The method to generate marked data and VOC2007 data format is relatively simple. After the installation of Anaconda software, LabelImg tool and Qt framework, images can be annotated. After the annotation is completed, VOC2007 format data can be

made and corresponding configuration files can be modified, focusing on the modification of the number of categories under YOLO node and the number of filtering of the convolution layer. Then, the pre-training weight is downloaded and the skeleton network is extracted to carry out the model training. The training process is shown in **Figure 5**.

The change curves of loss (loss function) and Avg_IOU (Average Intersection over Union) during model training are shown in **Figure 6** and **Figure 7**.

3.3. Implementation Details

First, create a new ROS workspace, and then complete the darknet_ros package download from the workspace using git clone command on the terminal. Then the project compilation work will be carried out, during which the weights file in the weights folder will be checked. Since the YOLOv3 model training has been completed previously, it is only necessary to copy the previously trained weights file to this folder. Before darknet_ros carries out the detection work, it first

```
Loaded: 4.533954 seconds
Region Avg IOU: 0.262313, Class: 1.000000, Obj: 0.542580, No Obj: 0.514735, Avg Recall: 0.162162, count: 37
Region Avg IOU: 0.175988, Class: 1.000000, Obj: 0.499655, No Obj: 0.517558, Avg Recall: 0.070423, count: 71
Region Avg IOU: 0.200012, Class: 1.000000, Obj: 0.483404, No Obj: 0.514622, Avg Recall: 0.075758, count: 66
Region Avg IOU: 0.279284, Class: 1.000000, Obj: 0.447059, No Obj: 0.515849, Avg Recall: 0.134615, count: 52
l: 629.763611, 629.763611 avg, 0.001000 rate, 6.098687 seconds, 64 images
Loaded: 2.957771 seconds
Region Avg IOU: 0.145857, Class: 1.000000, Obj: 0.051285, No Obj: 0.031538, Avg Recall: 0.069767, count: 43
Region Avg IOU: 0.257284, Class: 1.000000, Obj: 0.048616, No Obj: 0.027511, Avg Recall: 0.078947, count: 38
Region Avg IOU: 0.174994, Class: 1.000000, Obj: 0.030197, No Obj: 0.029943, Avg Recall: 0.088889, count: 45
Region Avg IOU: 0.196278, Class: 1.000000, Obj: 0.076030, No Obj: 0.030472, Avg Recall: 0.087719, count: 57
z: 84.804230, 575.267700 avg, 0.001000 rate, 5.959159 seconds, 128 images
```

Figure 5. Model training process.

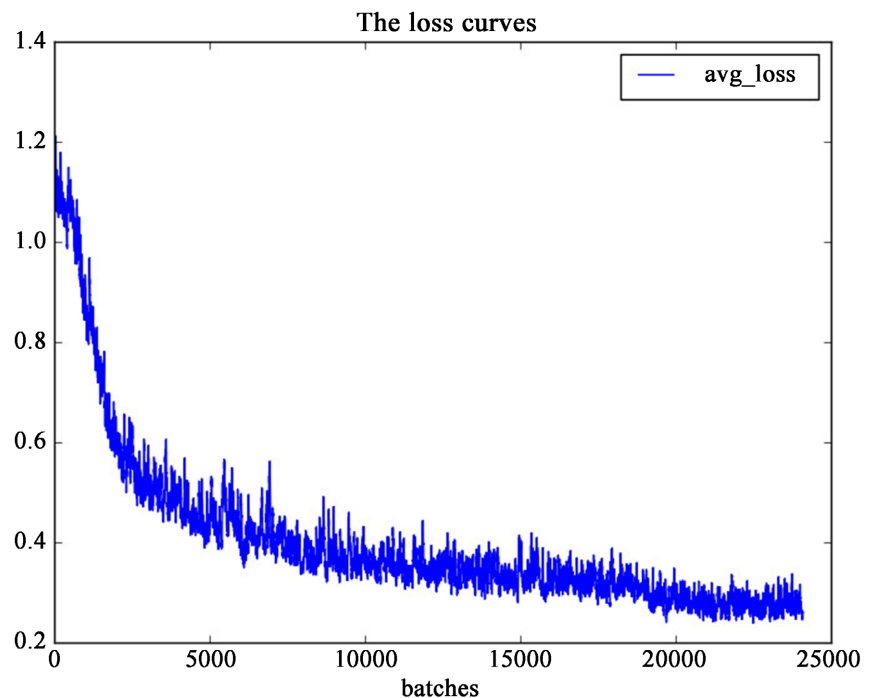


Figure 6. The change curves of loss.

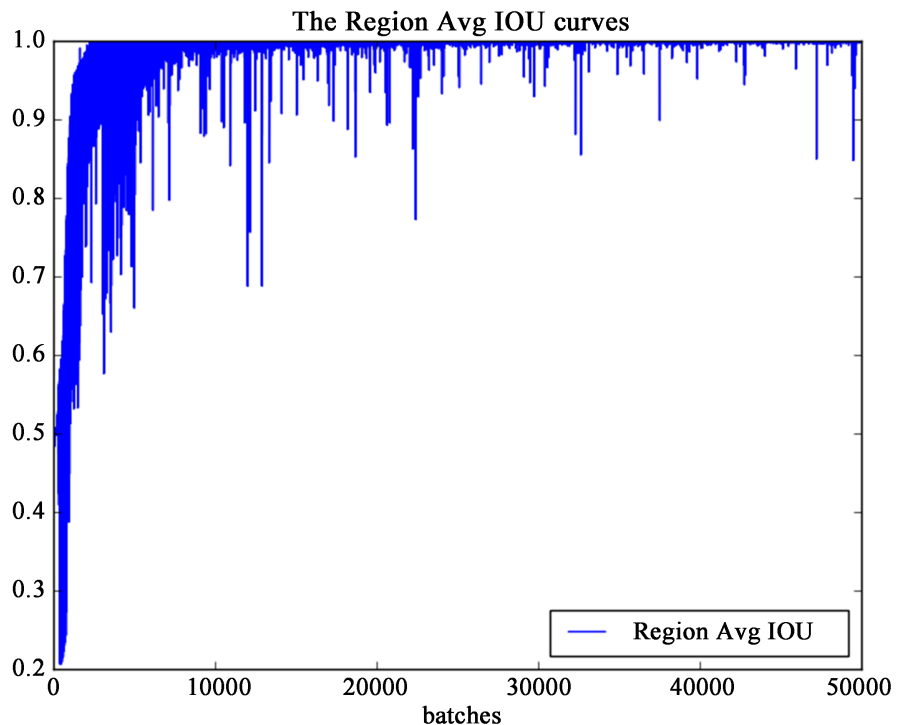


Figure 7. The Region Avg_IOU curves.

installs a function pack that can publish image topics. `usb_cam` package can publish the images read by the camera as image topics. After the camera driver package is downloaded, the image topic can be released by running the launch file, and the camera display interface can be observed on the PC virtual machine. Finally, the topic subscribed by `darknet_ros` corresponds to the topic released by `usb_cam` through modification of the configuration related to the function pack. At this time, the launch file corresponding to the function pack can be executed for object detection.

4. System Test and Results

4.1. Testing of Localization, Mapping and Navigation

In order to test various functions of the indoor security robot, it is necessary to define the test scene, which is selected as the school dormitory. The specific scene environment is shown in **Figure 8**.

After connecting the Ubuntu VM to the robot system using SSH (Secure Shell Protocol), perform imu calibration. For drawing construction, firstly start the `bringup.launch` file, run the necessary hardware components for drawing construction, and obtain the necessary data information; Then launch the `lidar_slam.launch` file to run the slam function pack; Then open the `rviz` visualization tool and keyboard control node, and open the `slam.rviz` file in the `rviz` tool to observe the real-time map construction process. The 2D raster map constructed by keyboard control is shown in **Figure 9**, which is roughly the same as the actual test scenario. The initial position and navigation route of the robot



Figure 8. The test scenario.

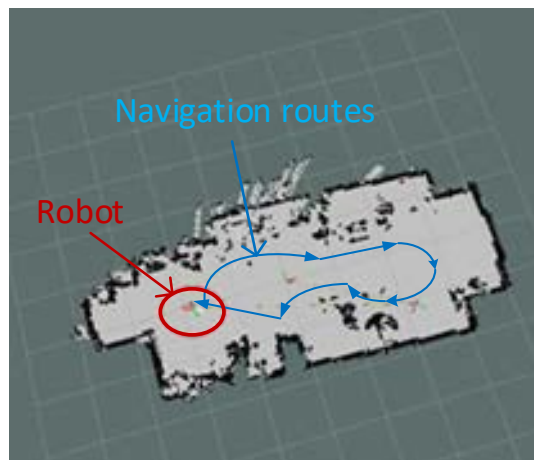


Figure 9. The 2D raster map.

have been marked with red circles and blue arrows respectively, which are consistent with the actual position and movement of the robot in the test scene.

After the map is built and saved, the 2D raster map can be used for navigation. Start file `bringup.launch` first, then start file `navigate.launch` after data information can be obtained, run path planning package `move_base`, and then start the `rviz` tool. Open file `navigate.rviz` inside, and the 2D raster map saved before appears. At this time, navigate through the upper toolbar.

4.2. Testing of Object Detection Function

First, enter the `catkin_workspace`, run the `usb_cam-test.launch` file to release the camera image topic, and check whether the camera can work normally. After the camera works normally, run the `darknet_ros.launch` file to test the target detection function. After the `darknet_ros.launch` file is successfully run, the target detection function will be formally tested. The test results are shown in **Figure 10** and **Figure 11**.

The VOC data test set we produced contained 4952 images with a total of 20 object categories. The predicted number of objects per class are shown in **Figure 12**. AP (Average Precision) And Map (Mean Average Precision) as shown in **Figure 13**.

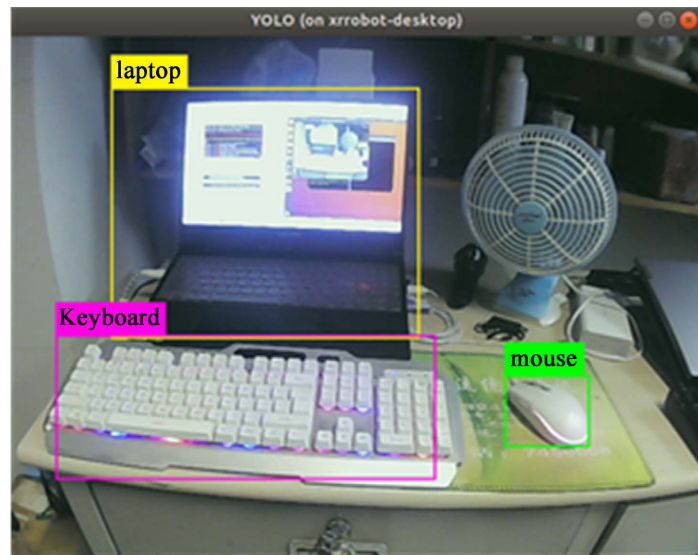


Figure 10. The first test result.

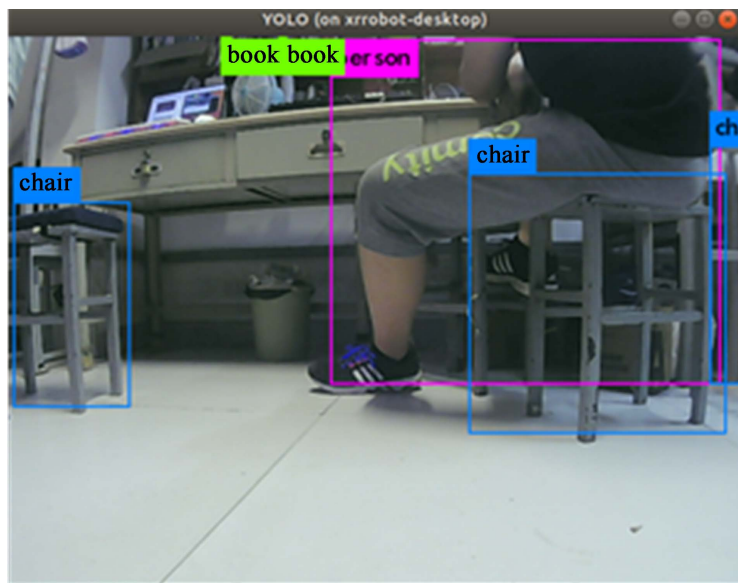


Figure 11. The second test result.

As can be seen from **Figure 12**, the trained YOLOv3 model has a high accuracy and meets the basic usage requirements of indoor security robots.

5. Conclusion

In this paper, a ROS-based indoor security robot system is designed and implemented, and the function and reliability of the system are well guaranteed through complex environment configuration, programming and model training. The system can make full use of the robot's laser radar, imu and other sensors to make the robot's positioning, mapping and navigation more accurate and stable. At the same time, because of the YOLOv3 model with good training effect, the robot's usb camera can be used to complete relatively accurate target recognition.

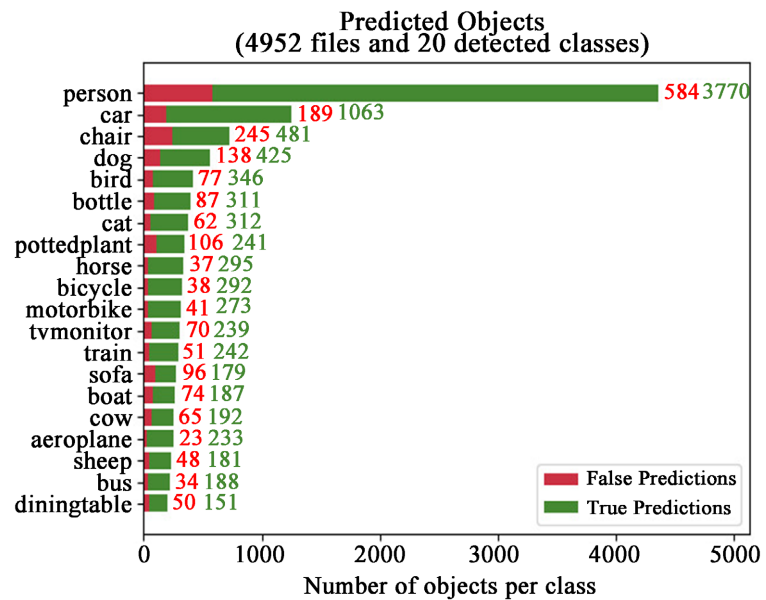


Figure 12. Number of objects per class.

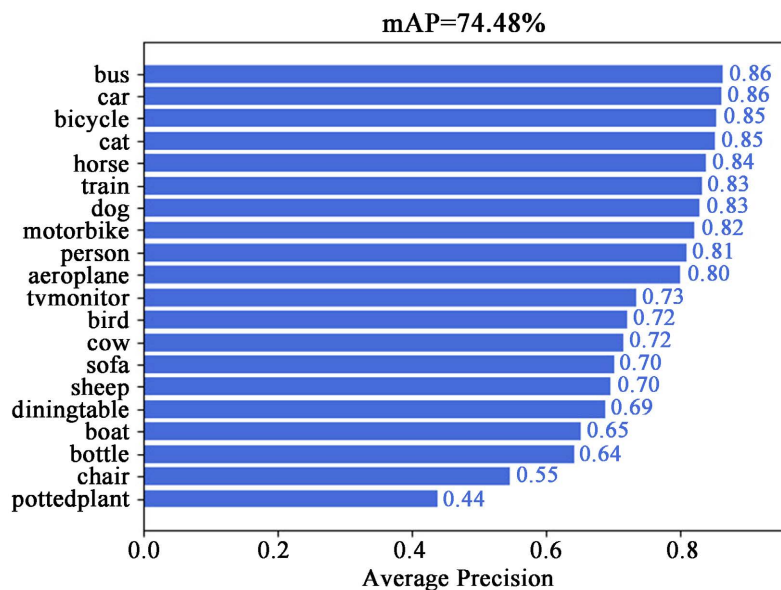


Figure 13. AP and mAP.

Moreover, the positioning, navigation and target detection functions can be combined, so that the robot can complete the target detection task in the navigation process. In the future, we will continue to optimize the structure of the system and the coordination and cooperation between the two functions, and improve the performance of the navigation algorithm and the object detection algorithm. For example, we will integrate the object detection function into the positioning and navigation function, identify obstacles in the process of movement through the object detection, and further transfer the data to the navigation function package, so as to optimize the path planning. The robot system has the navigation function of automatic obstacle avoidance.

Acknowledgements

During my research, many teachers and partners gave me help and support, especially my supervisor, Mr. Zhang Liye. I would like to thank them all here.

Fund

This work was supported by the (National Natural Science Foundation of China, Research on WiFi indoor localization System based on image assistance) under Grant (number 62001272) and the (Natural Science Foundation of Shandong Province) under Grant (number ZR2019BF022).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Ge, J.Q. (2020) Implementation of YOLO-v2 Vision Neural Network under ROS Framework of Mobile Robot Platform. *China Water Transport (Second Half)*, **20**, 95-97.
- [2] Yang, J.L. and Shi, E.X. (2013) Research on Indoor Mobile Robot Localization Method Based on Wireless Network. *Mechanical Science and Technology*, **32**, 457-461+468.
- [3] Zhong, Y.C., Liu, A. and Xie, R.L. (2017) Determination of Deflection Angle of Indoor Mobile Robot Navigation Based on Machine Vision. *Modular Machine Tools and Automatic Processing Technology*, No. 4, 1-4+9.
- [4] Guo, C. (2020) Research on Indoor Mobile Robot Navigation Based on Lidar. Master's Thesis, Zhejiang University of Technology, Hangzhou.
- [5] Zeng, N.W. (2021) Research and Implementation of Visual SLAM for Indoor Mobile Robots Based on Point-and-Line Feature Fusion. Master's Thesis, Chongqing University of Posts and Telecommunications, Chongqing.
- [6] Zheng, W.Q. and Wang, D. (2022) Obstacle Avoidance Research of ROS Indoor Mobile Robot. *Metrology and Measurement Technology*, **49**, 44-47.
- [7] Sun, Z.Y. (2018) Research on Three-Dimensional Modeling and Object Recognition Technology of Mobile Robot Indoor Environment. Master's Thesis, Harbin Institute of Technology, Harbin.
- [8] Li, X.P., Geng, D., Gong, Y.S., *et al.* (2018) Indoor Environment Object Detection for Mobile Robot Based on 3D Laser Sensor. *Proceedings of the 2018 37th Chinese Control Conference*, Wuhan, 25-27 July 2018, 5.
- [9] Jin, F. and Qi, C. (2021) Human Target Dynamic Tracking Method of Preschool Children's Companion Robot in Indoor Environment. *Automation & Instrumentation*, No. 11, 156-159.
- [10] Sun, T.T. (2021) Research on Multi-Sensor Indoor Mobile Robot Autonomous Localization Based on ROS System. *Internet of Things Technology*, **11**, 33-35.
- [11] Nagla, S. (2020) 2D Hector Slam of Indoor Mobile Robot Using 2D Lidar. *Proceedings of 2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, Chennai, 10-11 December 2020, 1-4.
<https://doi.org/10.1109/ICPECTS49113.2020.9336995>

-
- [12] Zhan, R.Z. and Jiang, F. (2018) Object Recognition System of Mobile Robot Based on ROS and Deep Learning. *Electronic Test*, No. 15, 70-71+64.
- [13] Mahendru, M. and Dubey, S.K. (2021) Real Time Object Detection with Audio Feedback Using Yolo vs. Yolo_v3. *Proceedings of 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, 28-29 January 2021, 734-740.
<https://doi.org/10.1109/Confluence51648.2021.9377064>
- [14] Mur-Artal, R., Montiel, J.M.M. and Tardos, J.D. (2015) ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, **31**, 1147-1163. <https://doi.org/10.1109/TRO.2015.2463671>
- [15] Sadruddin, H., Mahmoud, A. and Atia, M.M. (2020) Enhancing Body-Mounted LiDAR SLAM Using an IMU-Based Pedestrian Dead Reckoning (PDR) Model. *Proceedings of 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*, Springfield, 9-12 August 2020, 901-904.
<https://doi.org/10.1109/MWSCAS48704.2020.9184561>
- [16] Fu, G.P., Zhu, L.x. and Zhang, S. (2021) Training Robot Based on ROS and Lidar SLAM. *Information Technology and Informatization*, No. 11, 32-35+42.
- [17] Liu, Y.Q. (2021) Improved Target Detection Algorithm Based on YOLO Series. Master's Thesis, Jilin University, Jilin.
- [18] Li, B. (2020) Localization and Perception of Inspection Robot Based on Vision. Master's Thesis, Shanghai Institute of Electric Engineering, Shanghai.
- [19] Won, J.H., Lee, D.H., Lee, K.M., *et al.* (2019) An Improved YOLOv3-Based Neural Network for De-Identification Technology. *Proceedings of 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, Jeju, 23-26 June 2019, 1-2. <https://doi.org/10.1109/ITC-CSCC.2019.8793382>