Research on items sorting robot based on SSD target detection

Manming Shu*, *Chuan* Li, *Yongjian* Xiao, *Jie* Deng, and *Jinxin* Liu

Southwest Petroleum University, School of Information, 637001 Nanchong, China

Abstract. To achieve the automatic classification of items and improve the sorting efficiency of the sorting robot. A design method of a robot based on color classification is proposed. The system uses an all-around Mcnamu wheeled mobile platform equipped with a Raspberry Pi 4B, an STM32 MCU, a USB monocular camera, and a 6-DOF robotic arm. The SSD target detection model is used to identify sorted items, and color navigation is used to locate the sorting area and complete operations such as finding, identifying, grabbing, and putting back items. The actual test results show that the robot has an average item recognition rate of 99% and a classification rate of 97.25%, which verifies the feasibility of the robot and provides a reference for research on intelligent collaboration, item classification, fast logistics, and machine vision.

Keywords: SSD target detection, Items sorting, Color navigation, Mechanical control.

1 Introduction

Under the background of the "Internet +" intelligent manufacturing trend, automatic item sorting technology is increasingly used. According to research, most of the sorting robots are stationary. The fixed 6-DOF robotic arm proposed by Iqbal J et al. [1] can only complete sorting operations within a limited range and has relative limitations. Reference [2] proposed an image matching and precise positioning method based on SIFT algorithm. Peng Yuqing et al. [3] proposed a distributed control-based vision system for mobile robot target recognition in unknown environments. Z. Pan et al. [4] proposed a robotic express sorting system based on machine vision. Xu Z et al. [5] improved network feature extraction, reduced memory resource usage and accelerated the detection process. Sun C et al. [6] introduced a segmentation mask to eliminate background regions, which improved the performance of SSD in detecting small-sized objects. Accordingly, a design method of a movable item sorting robot combined with the SSD target detection algorithm is proposed to improve the sorting efficiency of the item sorting robot.

* Corresponding author: 201931772110@stu.swpu.edu.cn

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

2. Overall system design

2.1 Structural design

The item sorting robot consists of a USB monocular camera, a Raspberry PI 4B, an STM32 MCU, and a 6-DOF robotic arm, as shown in Figure 1. As the core of the sorting system, the camera is used to collect image information of the surrounding environment and is secured to the robotic arm, which can rotate 360° to obtain images. Raspberry PI 4B is the primary computing unit, processing the image information captured by the camera. The STM32 MCU is the control center of the whole system and controls the operation of the robotic arm steering unit, sensors, and Mcnamu wheeled chassis. The overall system is shown in Figure 2.

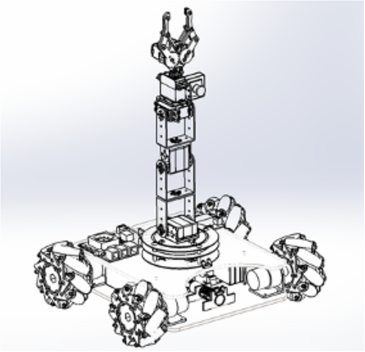

Fig. 1. Overall structure of the robot.

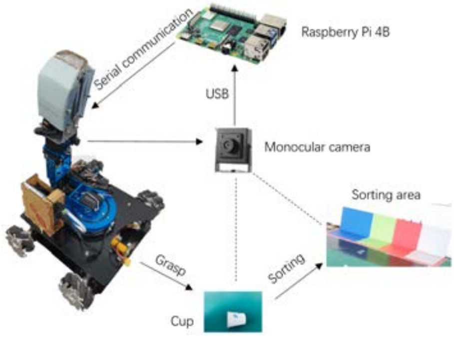

Fig. 2. Overall system diagram.

2.2 Sorting process design

The model captures and identifies the item to be measured by adaptive threshold segmentation, obtains the geometric center coordinates (X_1, Y_1, Z_1) of the item, compares the relative positions of the camera and the item, and drives the sorting robot closer. The distance between the item and the sorting system is acquired by a laser rangefinder and continuously adjusted until it reaches an appropriate grabbing point. The pressure sensor of the mechanical jaw detects whether grab the item or not. If the claw grab successfully, the robot converts into color navigation to find the corresponding sorting area and complete the grabbing task. The system sorting flow chart is shown in Figure 3.

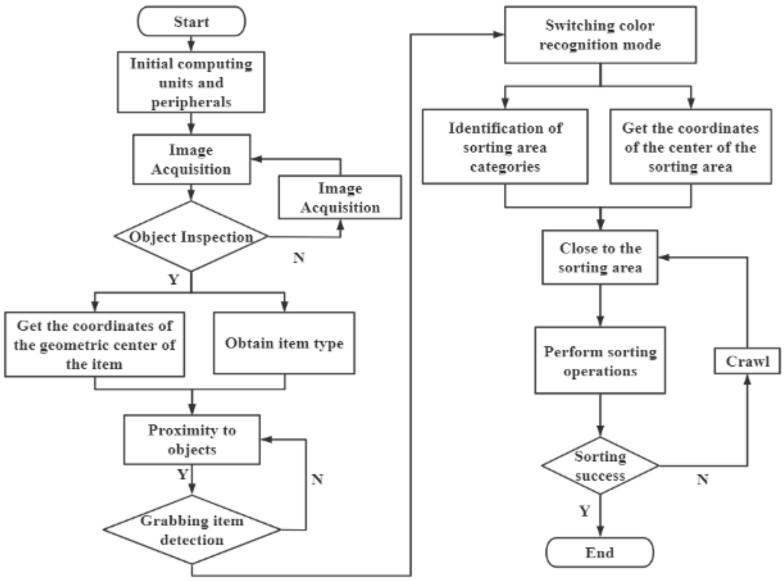

Fig. 3. System flow chart.

3 Neural network model design

The sorting robot builds the TensorFlow1.14 environment on the Raspberry Pi to run the SSD-Mobilenet-v3 target detection model. The SSD model is derived from VGG16, but the VGG16 model contains three fully connected layers, which leads to a large number of input parameters and consumes computer resources. The SSD target detection model replaces FC6 and FC7 with 3×3 the convolution layer Conv6 and 1×1 the convolution layer Conv7 based on VGG16. In the changed network structure, the fully connected layer is replaced by the convolution layer, which reduces the input parameters and releases the computer resources. And the SSD target detection model is a first-order target detection, which can quickly provide the coordinates and type information of the detected object, which is faster and more effective than the second-order target detection.

4 Training and optimization of SSD target detection model

To improve the accuracy of the training model, it is necessary to pre-process the collected images. Label 2000 images with Label Image software, generate executable XML files, select 20% of the item images as the neural network test, and train the neural network in batches using Colab Cloud Server. The model learning rate, the model loss rate, and recognition accuracy are used to judge whether the trained models are qualified. For the unqualified models, adjust the parameters and continue training until a qualified neural network model is optimized, and the model training process is shown in Figure 4.

To solve the problem that the low robustness of the SSD target detection model for small target object detection, the SSD network model is optimized and extended by deconvolution operation from improving the quality of the data set and improving the correlation between different image information features. For the resource problem, the model data is converted from float 32 bits to int 8 bits, which reduces the resource consumption of the model and makes it easier to deploy the mobile SSD model. After several model iterations, the model gradually converges, the learning rate decreases and the loss rate stabilizes, as shown in Figures 5-6.

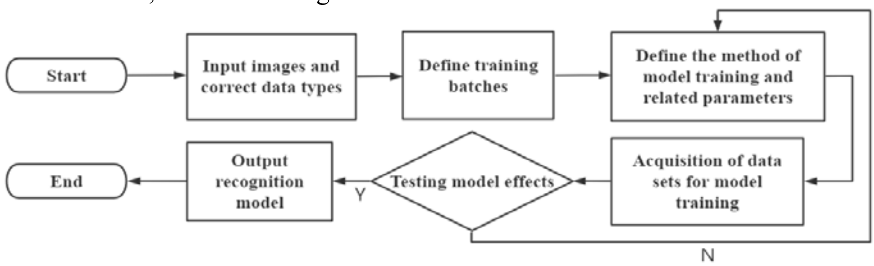

Fig. 4. Model training process.

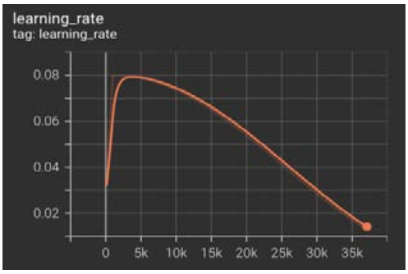

Fig. 5. SSD network model adaptive learning rate.

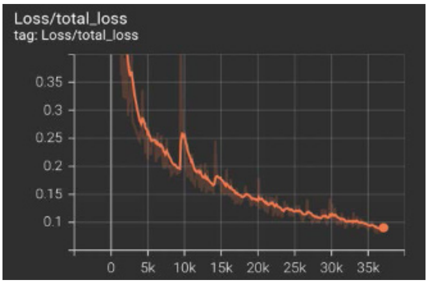

Fig. 6. The total loss rate of the SSD network model.

5 Items search and navigation

5.1 Items location search

To find the target object and consider the efficiency of the sorting system, an adaptive threshold segmentation algorithm [8] is used to highlight the item features. Using Gaussian convolution as the threshold, by adapting the proximity of each pixel to most scenes to make the navigation more accurate and increase the fault tolerance. Local thresholds are by calculating the luminance distribution of different regions through an adaptive threshold algorithm. Highlight the target image using the open operation of image morphology, as shown in Figure 7, where the red dot is the center of the region, to derive the relative position condition of the target and the center of the camera and make the robot close to the object under test.

Fig. 7. Segmentation effect of adaptive threshold algorithm.

5.2 Color location navigation

The essential feature of delineating different color spaces [9] is to find accurate and small shaded color sorting areas so that the sorting robot can better find color sorting areas. By studying four different color spaces, establishing a color channel evaluation system, separating color sorting areas using a mask, and creating a color threshold to separate red sorting areas, the effect graph is shown in Figure 8. To quantitatively evaluate the effect of different color space color separation [10], a function is established

$$PRA = \frac{CO}{MP} \tag{2}$$

PRA is the color separation effect, CO is the color separation profile, and MP is the morphological color profile. The color sorting area separation effects were detected using Matlab software as shown in Table 1, and the captured images were finally converted to HSV color space after a comparison of the effect figures. Calculate the geometric center of the contour by separating the colors, and the relative position of the center coordinates of the camera sorting area is compared with the camera to make the sorting robot approach the sorting area accurately.

Table 1. Color separation values in different color spaces.

Color sorting area	RGB	YCbCr	HSV	LAB
Blue	95.32%	97.58%	98.93%	94.65%
Green	96.43%	95.43%	98.79%	93.29%
Red	94.48%	96.32%	99.13%	93.37%
Grey	93.89%	96.19%	99.65%	93.34%
Average	95.03%	96.38%	99.13%	93.66%

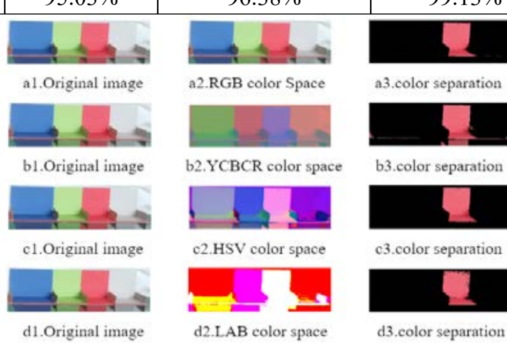

Fig. 8. Color separation renderings of different color spaces.

6 Experimental results and analysis

To verify the synergy among the related functions of item search, SSD object detection, and color navigation, the system is functionally tested. The operational status of the robot is analyzed to improve the performance parameters of the item sorting robot.

After several tests, the results show that the robot can independently search for items, correctly identify them, and correctly put items back into the color sorting area. During the tests, the system operates normally, with an average model recognition accuracy of 99% and an average sorting success rate of 97.25%, as shown in Table 2, making the item sorting robot truly end-to-end autonomous without external intervention.

Fig. 9. Item recognition renderings.

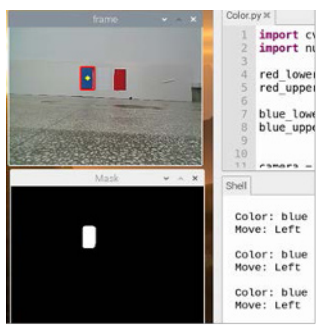

Fig.10. Color navigation interface.

Table 2. Item recognition average accuracy and sorting rate.

Test type	Average precision (%)	Classification success rate (%)
Paper cup	99.5	99.50
Bottles	99.0	96.00
Battery	98.0	94.00
Paper ball	99.5	99.50
Average	99.0	97.25

7 Conclusion

With the growing demand for item sorting and rising labor cost in recent years, this paper designs and develops an end-to-end intelligent item sorting robot. It uses the USB camera, Raspberry Pi, robotic arm, and STM32 MCU to collaborate at all four ends to achieve the operations of finding, identifying, grabbing, and putting back items autonomously. Histogram equalization, image grayscale, and image filtering are used to enhance the image, and after model training, the average item recognition rate reaches 99% and the average sorting success rate reaches 97.25%. The experimental results show that the intelligent

https://doi.org/10.1051/itmconf/20224701031

sorting robot has strong feasibility, stability, and practicality, which provides a practical reference for the research of intelligent collaboration, item sorting, fast logistics, machine vision, and so on.

This research was supported by the key project of Southwest Petroleum University's school-level open experiment (Grant No.NKSZ21013)

References

1. Iqbal J, Islam R U, Khan H. Modeling and analysis of a 6 DOF robotic arm manipulator[J]. Canadian Journal on Electrical and Electronics Engineering, 2012, 3(6): 300-306.
2. FU W P, QIN CH, LIU J, et al． Matching and Location of Image Object Based on SIFT Algorithm[J]. Chinese Journa of Scientific Instrument, 2011, 32 (1): 163-169.
3. Peng Yuqing, Li Mu, Gao Qingqing, Zhang Yuanyuan. Mobile robot target recognition based on dynamic template matching [J]. Journal of Transducer Technology, 2016, 29(01): 58-63.
4. Z. Pan, Z. Jia, K. Jing, Y. Ding and Q. Liang, "Manipulator Package Sorting and Placing System Based on Computer Vision," 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 2020, pp. 409-414, doi: 10.1109/CCDC49329.2020.9164071.
5. Xu Z, Fan M, Zhang Y. Visual detection of eggs based on deep learning for egg picking robot[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1976(1): 012013.
6. Sun C, Ai Y, Wang S, et al. Mask-guided SSD for small-object detection [J]. Applied Intelligence, 2021, 51(6): 3311-3322.
7. Ji X, Yan Q, Huang D, et al. Filtered selective search and evenly distributed convolutional neural networks for casting defects recognition [J]. Journal of Materials Processing Technology, 2021, 292: 117064.
8. Liu T, Tian Y, Zhao S, et al. Automatic whole heart segmentation using a two-stage u-net framework and an adaptive threshold window[J]. IEEE Access, 2019, 7: 83628-83636.
9. Bi Z, Cao P. Color space conversion algorithm and comparison study[C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1976(1): 012008.
10. CEYLAN H, ÇOKAKLI Ö. COLOR SEPARATION WITH COMPUTER VISION [J]. International Journal of Engineering and Innovative Research, 3(3): 201-208.