Robotics Final Project

Team 119
Bohan Wei (s3600246) , Yuying Xia (s3772233),
Shihang Yu (s3240924), Tian Xia (s3669254), Ziming Guo (s3661237)

June 2023

1 Introduction

In the final project, we will implement a series of tasks using our own robot Picar4wd. Our group name is 119, which is the Chinese fire alarm call. We want to simulate what happens to a fire truck after receiving a fire alarm.

First, after receiving the fire call, the fire truck should go to the fireplace the first time. We simulated this process. We used a tennis ball as the navigation in the map to simulate the arrows in Google Maps.

In the second step, we simulated the situations we would encounter on the road. For example, traffic lights. Although in real life, fire trucks can ignore traffic lights, in our experiments, we used red lights as warnings, and when a fire truck detects a red signal ahead, it beeps as an alert. All codes and demos can be found in our github.

2 Tasks Description

2.1 Object Tracking

Basically, the robot can track any object if we already set the color and shape of the object first. After the robot captures the image, it should identify the object in the image according to our setting. Once the robot locates the object, we can control it to turn right or left based on the location of the object. For example, if we detect an object such as a tennis ball and the object is on the left of the image, the robot will turn left a bit to keep the object in the right front of the robot.

2.2 Color Detection

Here we want to simulate the different operations in different situations. In our case, we simulate the fire. Therefore, when the robot sees the red. It then determines that there is a fire, and it will raise the alarm. It is allowed us to design different operations. If there is a yellow light, The robot moves forward first, then turns left. To avoid noises, other colors will not be identified.

3 Object Tracking

After receiving the fire alarm, the robot needs to follow a target to guide it to the designated place, that is, to achieve target tracking. Next, the method of achieving target tracking will be introduced in detail.

3.1 Find object and HSV

In the demonstration of the actual task, we used an orange as the object to be tracked. In order to enable the robot to recognize better, we used hsv color recognition. First, we used color space conversion to convert the image from RGB to hsv, HSV stands for Hue, Saturation, and Value, a method used to represent color. Compared with RGB color representation, HSV is more in line with human perception and description of color. Just show in Figure 3.

Specifically, the hue mainly indicates the type of color, which is an angle value, the range is 0-360 degrees, 0 degrees represents red, 120 degrees represents green, 240 degrees represents blue, and saturation is used to represent the vividness of the color. Brightness indicates the brightness of the color. HSV can adjust the color intuitively by adjusting the value of hue, saturation and lightness. In the tracking task, we need to track orange objects. Using HSV can also ideally adjust the color to our required color range.

3.2 Mask and contour

Next, we need to create a mask within a predetermined threshold range, locate the largest orange object by finding the contour in the mask, and calculate the coordinates of the center of gravity. If the found contour area exceeds the set threshold, it is judged as the target object, that is, the target object to be tracked is determined.

After starting the robot, we keep calling the camera to obtain images, and use the above algorithm to detect orange objects. If an orange object is found, we adjust the direction of movement of the car according to its position in the image, for example, the captured object The center of gravity is not in the center of the screen, the car will move left and right so that the center of gravity of the object is always in the center of the screen, so as to achieve the effect of target tracking.

4 Color Detection

In this section, we start by simulating fire trucks responding to different situations. In our experiments, we use red color to represent a fire place, yellow color to represent an obstacle, our robot should avoid the obstacle. To reduce noises, the rest of the colors are marked as others.

4.1 Color recognition

We will complete the color detection task in this section. In this task, an extra Bluetooth speaker was required. We first calculated the center of pixels point and start color recognition.

When the color appears in the center of the pixel and is recognized as red, an audio of a fire truck siren is played by mplayer. When the color appears in the center of the pixel and is recognized as yellow, the robot go straight and turn left 90 degrees and then stop. Except red and yellow can be identified, the rest of the colors are represented by 'others' color. When the display recognition color is others, the robot remains stopped. By adjusting the value of hue in HSV, we can define the red and yellow upper and lower ranges to find the color intervals.

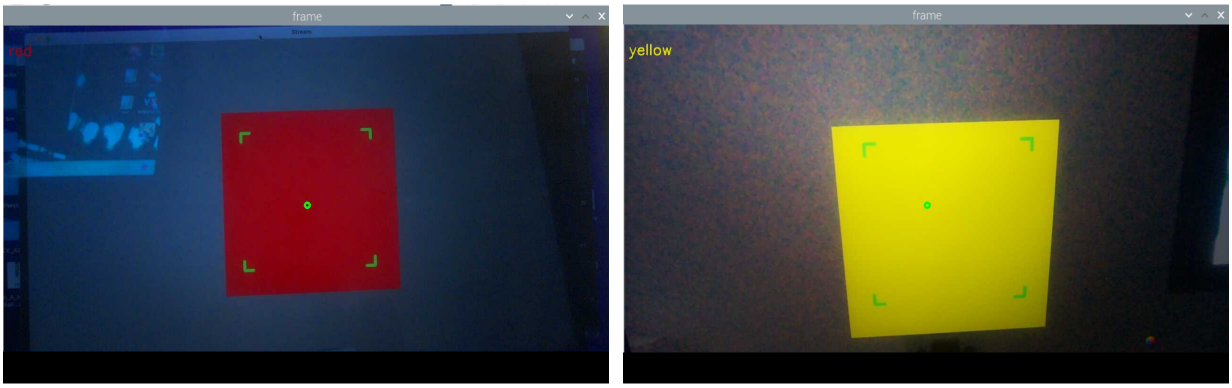

Figure 1: The demo example figures of color detection task.

4.2 Real-time video streaming

Due to the limitation of processing power of Raspberry Pi and the installation of some libraries, we use Transmission Control Protocol (TCP) to enable real-time streams live camera video to a PC, display and processing of the video feed via ZeroMQ library. After starting the camera preview by using libcamera, first it create a ZeroMQ socket() and connect it to the server's IP address and port (5555) using TCP. Then enter an infinite loop to continuously capture frames from the camera. Encode the captured frame as a JPEG image and convert it to a base64 encoded string. These encoded frames are then sent to the server for real-time transmission. In our experiment, the Raspberry Pi is the client and the PC is the server.

4.2.1 Virtual quiz

To simulate the firefighting system more realistically, a three-question quizes of fire location, fire level and the number of fire trucks needed will be asked after performing the previous task. This quiz is like a work log, recorded the fire situation. The example is shown as figure 2. This task is based on computer vision algorithms and requires the HandDetector function in cvzone. To answer the quizes, it just need to use the fingers to click on the rectangle of the answer option boxes. After finish these three questions, a display message according to the answer choices will be shown on the screen[1]. Although we process the stream data on the PC, there is still a high delay of transmission.

[1] https://www.computervision.zone/courses/virtual-quiz-game/

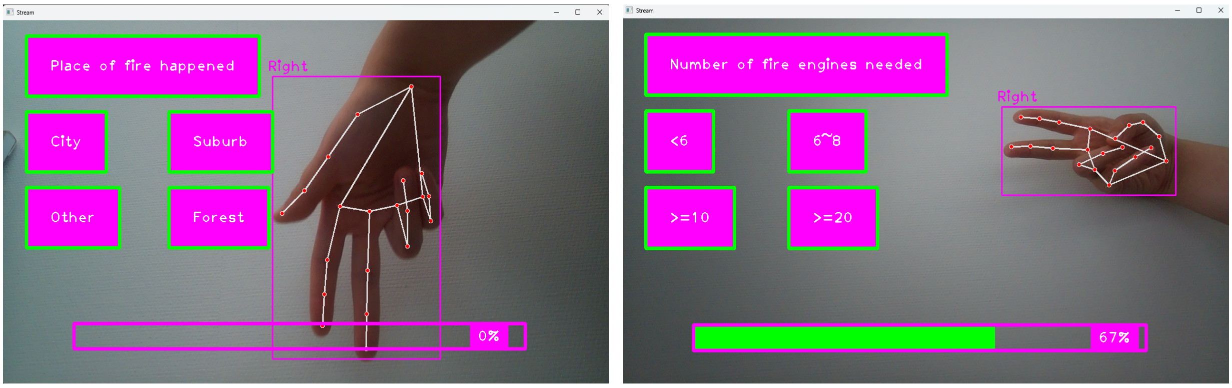

Figure 2: The demo example figures of virtual quiz task.

5 Future work

Although we have trained our own yolov5 data set and model, unfortunately, yolov5 can run normally on the PC, but the camera function cannot be enabled on the Raspberry Pi. We will try to fix this in future work. Besides, regarding the TCP real-time transmission data flow, we can send data from the Raspberry Pi to the PC for processing, but the data processed by the PC cannot be transmitted back to the Raspberry Pi, and there is also a delay of about 10 seconds. Therefore, we can try faster methods like UDP, while continuing to learn and understand the use of TCP.

6 Conclusion

In this project, we have successfully employed a Raspberry Pi cart to simulate a fire rescue mission. The integration of object detection for tennis balls and tracking has enabled the cart to navigate effectively, mimicking the movements of a real fire truck. By incorporating a color detection function, the cart can promptly identify the presence of fire when it recognizes the color red. This triggers an alarm through an external speaker, ensuring immediate response.

Furthermore, we have implemented a refined system for gathering fire-related information. Three specific questions have been devised to extract crucial details about the fire. To streamline the data collection process, we have utilized a HandDetector function, allowing for convenient selection of the answers to these questions.

Overall, this project has successfully demonstrated the capabilities of the Raspberry Pi cart in simulating a fire rescue mission. The combination of object detection, color recognition, and refined information gathering techniques has resulted in an efficient and realistic simulation system.

Appendix

A Figures

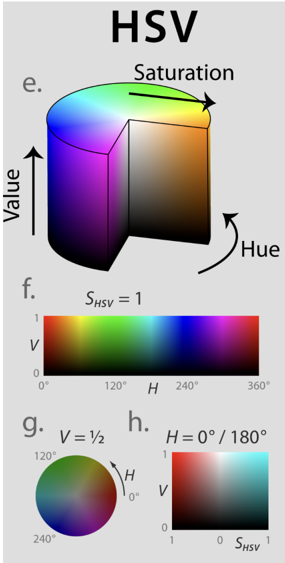

Figure 3: HSV color