Object Searching Robot Controlled by Edge-AI

Ryosuke Miyata¹, Osamu Fukuda¹, Nobuhiko Yamaguchi¹ and Hiroshi Okumura¹

Saga University, Graduate School of Science and Engineering Japan¹
Email: fukudao@cc.saga-u.ac.jp

Abstract: This study proposes and develops an edge-AI-based autonomous mobile robot based on open-source software. The robot is capable of voice and object recognition; it can detect and approach an object specified by a user's voice. Because the robot is controlled by voice commands, the user can control the robot intuitively. In the present study, we used a robot operating system to facilitate the development. All functions, including voice recognition, object recognition, and motor control, were implemented in the edge AI computer based on open-source software. We conducted preliminary experiments to verify the performance of the developed system.

Keywords: Mobile robot, Edge-AI, Open-source software, Image recognition

I. INTRODUCTION

With the development of information technology, mobile robot studies have garnered significant attention. Primal studies can be categorized into three subfields: motion path planning, robot localization, and environment map creation [1]. Path planning is a fundamental problem that various researchers have been working on since the mid 1960s. These techniques enable mobile robots to autonomously move and explore diverse environments.

Several path-planning methods have been developed and verified using various sensors, such as infrared and ultrasonic sensors. Visual navigation has also been proposed for mobile robots [2]. The development of the path-planning technique significantly reduced the travel time and machinery failures, and also simplified the system structure [3]. From a mechanical perspective, hardware specifications and capabilities were limited in the early days of mobile robotics. Moreover, it was difficult for the system to handle visual images; therefore, the mobile robot could not find its way in an environment with several obstacles [4].

Depth cameras have emerged as alternatives to laser range finders and stereo vision systems, significantly evolving robot vision systems. However, depth cameras were expensive when they were first introduced in the market. Therefore, they could not be used as general-purpose tools [5]. In 2010, a depth camera was adopted as a video game interface, and since then has been widely used as a visual aid for robots. This served as a trigger to start using it as a general-purpose sensor. In addition, deep learning has garnered attention as an innovative image recognition technique. Deep learning technology allows robots to model raw multimodal data in an end-to-end manner using a large-scale network structure [6]. Mobile robots have been developed to avoid obstacles and approach a target location based on object detection. The advantage of this approach is that it does not require a predefined map [7], [8].

In addition, robotics technology has advanced from a software perspective. Presently, robot operating system (ROS) is a common software platform for robot development. Various researchers have applied ROS for efficient robot development. ROS facilitates software management, and users can reuse various software libraries among different robots [9].

The conventional mobile robots have various
drawbacks. Because image processing and motor control are performed using an on-site single computer system, complex and high-load computation in deep learning can cause a calculation-capability problem. Distributed computing can be an effective solution. Edge AI is a cutting-edge technology based on small, embedded computers with a general-purpose graphics processing unit. The edge AI distributes the computational load and can enhance the real-time performance in deep learning calculations [10]. It ensures the quality and rapidity of robot control, thus enabling real-time access to information.

In this study, we propose and develop a mobile robot based on edge AI and design its development platform. We aim to contribute to the distribution of a useful robot tool, which is equipped with light and inexpensive edge AI devices. The developed mobile robot has both voice and image recognition functions for detecting the target objects specified by the user. The robot can be instructed by the voice commands of the user to detect and approach an object. In the developed system, the object names are entered into the voice recognition part in Japanese, and the target object is detected using the deep learning technique. In addition, we integrated all functions, including voice recognition, image recognition, and motor control, on a single-board computer based on ROS middleware.

II. SYSTEM COMPONENTS

Figure 1 illustrates the system components of the autonomous mobile robot. The proposed system is divided into three parts: voice recognition, image recognition, and motor control. The voice recognition part receives the operator's voice command through a USB microphone. The name of the target object is recognized based on a predefined word dictionary. The image recognition part captures the image sequence using a vision sensor mounted on the mobile robot; thereafter, the objects are detected using a deep learning algorithm. The motor control part searches for the target object while comparing the voice command with the image recognition result. When the robot detects the target object, the system estimates its position and subsequently controls two motors to approach it.

Jetson Nano, a single-board computer developed by NVIDIA Corp., was adopted to implement the aforementioned three functions. Jetson Nano is an embedded hardware platform that is suitable for the
research and development of the edge AI devices. The board is equipped with a CUDA-compatible 128-core GPU; thus, it can perform higher speed processing compared to other similar-size embedded devices.

In this study, we developed a handy AI mobile robot known as JetBot (Fig. 2). JetBot is based on Jetson Nano, and is equipped with a vision sensor, two motors, motor driver, mobile battery, Wi-Fi module, and PiOLED display. Two wheels are installed symmetrically, and one ball caster is attached to the rear part. A USB microphone (MM-MCU02BK, Sanwa Supply Co., Ltd.) and Julius 4.4.2 were used to perform voice recognition.

The object detection algorithm, you only look once (YOLO) [11], was introduced for image recognition. ROS is responsible for overall system management, including voice and image recognition, and motor control.

A. Voice Recognition Part

The voice recognition part uses a phoneme model that represents phoneme frequency patterns in word speech. The voice command is explored using the phoneme model in a predefined word dictionary. Given the input voice, $X$, the system calculates the posterior probability $p(W|X)$ for word $W$. According to Bayes’ theorem, the posterior probability, $p(W|X)$, is calculated as follows:

$$ p(W|X) = \frac{p(W) \cdot p(X|W)}{p(X)} \quad (1) $$

The denominator can be considered a normalization coefficient. $p(W)$ is a prior probability for word $W$. This value is unrelated to voice $X$, and all the prior probabilities are set to similar values. $p(X|W)$ is the probability that voice $X$ generates for word $W$, and it is calculated based on the phoneme model.

B. Image Recognition Part

The image recognition part recognizes the object and its two-dimensional position on the image. The YOLO algorithm is designed based on convolutional neural network architecture, and can recognize multiple general objects in an image with high accuracy in real time. The advantage of the YOLO algorithm is that it performs object recognition and localization simultaneously, that is, the bounding boxes of the objects and their class probabilities are estimated simultaneously.

The concept of image processing is illustrated in Fig. 3. The image entered in the convolutional neural network is divided into $S \times S$ areas, known as grid cells.
Thereafter, rectangular areas are set on them. Each rectangular area has central coordinates \((x, y)\), size \((h, w)\), and a confidence score. The confidence score indicates the confidence of the bounding box containing the object. The confidence score is 0 when the object does not exist in the grid cell.

When the confidence score exceeds a predefined threshold, the object is detected. The horizontal distance \(x\) is calculated between the image center \(C\) and the center of the bounding box of the object. Thus, the system estimates the positional relationship between the object and robot.

\[
x = C - (x_{\text{max}} - x_{\text{min}}) \times 0.5
\]

\(x_{\text{max}}\) and \(x_{\text{min}}\) are the maximum and minimum coordinates of the bounding box of the target object, respectively. \(x\) represents the horizontal distance of the target object from the image center. The area of the bounding box is also calculated.

\[
S = (x_{\text{max}} - x_{\text{min}}) \times (y_{\text{max}} - y_{\text{min}})
\]

Because the size of the bounding box changes depending on the distance between the robot and object, the object can be detected. This information is sent to the motor control part.

C. Motor Control Part

The motor control part is implemented based on the middleware of ROS. ROS is a software platform that promotes the collaborative development of robot systems globally. The advantages of ROS are as follows:
1. The comprehensive function can be realized by combining a small executable process known as Node. 2. Each node can operate independently in parallel, and bidirectional data transmission/reception is available. 3. Various packages are ready to use, share, modify, and redistribute. 4. Various programs developed by users can be easily converted to ROS-compatible programs using the API provided by ROS. ROS contains large packages provided by developers globally, and is supported and further developed by the ROS community.

The basic principle of the robot control is depicted in Fig. 4. First, the target object is determined according to the voice command of the operator. Thereafter, the image recognition part recognizes the objects and explores the target object in the image. When the system detects the target object, its position is sent to the motor control part. The center coordinates of the target object are calculated from the bounding box. The horizontal distance, \(x\), between the object center and image center is calculated. The system constantly monitors the position of the target object as a visual feedback while controlling the robot. The moving direction turns to the center of the object. The speeds of the left and right wheels are controlled as follows:

\[
v_R = v - K_p x
\]

\[
v_L = v + K_p x
\]

Here, \(K_p\) denotes the proportional gain. The speeds of the left and right wheels are controlled proportional to \(x\), and the moving direction of the robot is changed to maintain the object at the center of the image. Term \(v\) denotes the component of the velocity of the mobile robot in a straight direction. A constant velocity is set at a suitable traveling speed.

When the area of the bounding box of the target object exceeds the prespecified threshold, the system determines that the robot has sufficiently reached the target object, and the motors are stopped.

III. EXPERIMENTS

A. Basic Control

First, we conducted a simple experiment. The experiment confirmed that the developed robot could approach the target object appropriately. A teddy bear was placed in front of the left side of the robot as the target object. The robot approached the target object properly and stopped at an appropriate distance.
The experimental results are presented in Fig. 6. From the top, the speech recognition result, distance $x$ between the object and image center, left and right motor speeds, and size of the bounding box of the target object. The horizontal axis denotes time. The voice recognition part recognized the voice command of the operator, which indicated the target object, a teddy bear. The image on the left depicts the view from the initial position of the robot. The teddy bear was recognized using the YOLO program. Thereafter, the distance between the object and image centers was calculated to regulate the speed of the left and right wheels. Because the object was located on the left side of the robot, the speed of the left wheel was low. The robot turned in the direction of the target object according to the difference in speed between the left and right wheels. Thereafter, the left and right wheels were controlled at a constant speed to approach the object. During the robot movement, the target object was captured at the center of the image, as shown in the center and right images. The size of the bounding box increased gradually. When it exceeded a prespecified threshold, the object search was successfully completed (right-hand-side image). The system returned to the standby state to wait for voice commands.

B. Explore the multiple objects

In the next experiment, multiple objects were set in the environment to examine the object exploration performance. A word dictionary that defines five types
of objects was used: cup, bottle, teddy bear, keyboard, and computer mouse. The system recognized these words to explore the objects. Figure 5 depicts an overview of the environment. The cup, mouse, keyboard, teddy bear, and bottle were placed around the robot, and the robot faced the bottle at the initial position. The robot explored the object and approached it, as specified by the voice commands of the operator. The operator instructed the target object in the following order: teddy bear, cup, and then keyboard. The robot continuously explored and approached them.

The experimental results are presented in Fig. 7. As shown in the images, the robot approached the teddy bear, cup, and then keyboard. The robot moved toward the teddy bear (from 1 to 3), cup (from 4 to 5), and then keyboard (from 6 to 8). The images on the left show the top view of the demonstration, whereas the images on the right show the robot's view captured by the vision sensor mounted on the robot.

IV. DISCUSSION

The experiments verified the robot functions in the following three parts: voice recognition, image recognition, and motor control. The voice recognition part performed almost perfectly. The object could be specified by the voice command of the operator. We used a high threshold value for the recognition determination to avoid misrecognition because the motor and/or environmental noise could affect the recognition accuracy. The image recognition part worked satisfactorily. The robot recognized the object and began to move within approximately 0.25[s] after the system recognized the voice command of the operator. The robot stopped at an appropriate distance from the target object. The robot could also track the object when it moved during the robot control. The motor control part navigated the robot to the target object with a smooth trajectory. From the provided description, the robot detected and approached the target object and stopped in front of it. When the system is unable to detect the target object, the robot rotates and explores the surrounding area.

However, the current system still has much room for improvement. The word dictionary was limited to only five words. The system can understand words, but not sentences. Therefore, we need to design a robust voice
recognition method for more words and sentences. Image recognition was executed at a relatively low frame rate of approximately 6 [fps]. The experimental environment was limited to a desktop. A few misrecognitions were observed for distant objects. Therefore, we should conduct experiments in a more practical environment and examine object recognition accuracy for distant objects. The size of the bounding box depends not only on the distance but also on the object size. Therefore, we should carefully assign an appropriate threshold value to detect the object approach with high confidence.

V. CONCLUSION

In this study, an autonomous AI mobile robot was designed and developed using an edge AI device and open-source software. The robot can detect the object specified by the voice command of the user and approach it based on image recognition. ROS middleware was used for the robot development. Preliminary experiments confirmed that voice recognition was performed with high accuracy. The robot successfully recognized the target object in the environment. The robot approached it by estimating its position. The robot control part realizes smooth tracking of the object based on vision-based feedback control.

In the future, we will introduce a depth sensor to measure the distance between the robot and object to improve the control performance of the robot. Additionally, we will attempt to introduce context recognition to control the robot more intuitively.

This work was supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP19K04296.

REFERENCES


