AGV for Library Management

Dr.A.Murugarajan [1, a)] DevaHarini R [1, b)] Harsitha S [1, c)] MahaNandhana V, [1, d)] Shenbhaka Priya K B
[1]*Robotics and Automation, Sri Ramakrishna Engineering College, Coimbatore, India.*

Abstract- **This paper presents the design, development, and testing of an autonomous robot that utilizes a Raspberry Pi to perform line-following navigation, capture high-resolution images of book racks, process these images to extract textual information using OCR (Optical Character Recognition), and finally upload the results to a remote web server. The system integrates various hardware components, including a USB camera, IR sensors for line detection, and a motor driver (L298D) for precise manoeuvring. The report details the system architecture, algorithms used for navigation and image processing, design and implementation aspects, experimental results, and a cost analysis. The proposed robot is designed to work effectively in dynamic environments such as libraries and industrial storage settings, offering potential improvements over existing manual systems by automating inventory and documentation tasks. The overall design is modular, allowing for future enhancements in both navigation algorithms and data management.**

I. INTRODUCTION

The advancement of embedded systems and robotics has enabled the creation of intelligent, autonomous machines capable of performing complex tasks with minimal human intervention. In environments such as libraries and storage facilities, traditional inventory and cataloging systems rely heavily on manual labor, which is often inefficient, error-prone, and time-consuming. This project introduces a Raspberry Pi-based autonomous robot designed to streamline and automate inventory management tasks, particularly within a library setting. The robot is equipped with infrared (IR) sensors for line-following navigation, allowing it to move along predefined paths accurately. A USB camera mounted on the robot captures high-resolution images of book racks, which are then processed using Optical Character Recognition (OCR) techniques to extract textual information such as book titles. The extracted data, along with the captured images, is uploaded to a remote server using HTTP communication, where it can be accessed and monitored via an IoT-enabled web interface. By minimizing human involvement and increasing data accuracy, the proposed solution demonstrates significant potential for improving inventory processes in libraries and similar environments. The system also lays the groundwork for future developments using machine learning and real-time data analytics.

II. LITERATURE REVIEW

The integration of autonomous navigation and intelligent data processing in mobile robots has received increasing attention in recent years, especially in domains like inventory management and smart libraries. Traditional manual systems face limitations in scalability, speed, and accuracy, which has led to the exploration of embedded robotic platforms. Research has highlighted the effectiveness of IR sensor-based line-following mechanisms for precise path tracking in indoor environments, as they offer real-time response and low computational load. Several studies have also focused on the use of USB and Pi cameras for high-resolution image capture, particularly in text-rich environments such as libraries and archives. OCR technologies have evolved significantly, with Tesseract OCR being widely adopted for its open-source accessibility and multilingual capabilities. The integration of OpenCV with Raspberry Pi for real-time image processing has proven effective in low-cost automation setups. The use of lightweight HTTP protocols and Python-based communication libraries like requests enables seamless IoT integration for remote data transfer and dashboard visualization. Moreover, modular design and server-based data logging have been emphasized for scalability and adaptability in dynamic applications. The literature also underlines the importance of efficient GPIO management and motor control using drivers like L298D, enabling stable motion in line-following robots, especially in confined or structured environments like libraries.

III. METHODOLOGY

The methodology for the Book-Navi AGV involves a systematic integration of hardware and software components to achieve autonomous library inspection and inventory management. The process is designed to enable the AGV to navigate library aisles, capture images of book spines, process these images to extract information, and communicate the data to a central server.

The key steps in the methodology are as follows:

1. Autonomous Path Planning and Navigation: The AGV is programmed for autonomous navigation within the library environment. This primarily involves utilizing IR sensors to follow designated lines or paths within the library aisles. The Raspberry Pi microcontroller processes the sensor data and controls the DC motors via the L298D motor driver to execute movements such as forward, backward, left, and right turns, enabling the AGV to traverse the aisles. Obstacle detection using IR sensors triggers stopping or avoidance maneuvers.
2. Image Acquisition: A USB camera, supporting resolutions up to 1920x1080, is integrated into the AGV's structure. As the AGV navigates along the aisles, the camera captures high-quality images of the book racks and spines. Image capture can be triggered at predefined intervals or based on the AGV's movement.
3. Real-time Image Processing: The captured images are transferred to the Raspberry Pi, which utilizes the OpenCV library (cv2) for real-time image processing. This involves essential steps such as cropping the relevant areas of the image containing book spines, potentially deskewing to correct for perspective distortion, and applying filters or thresholding to enhance text visibility and contrast, preparing the images for optimal OCR performance.
4. Optical Character Recognition (OCR): A custom OCR module (text_detection.py), implemented in Python, is applied to the processed images. This module analyzes the image to detect text regions on the book spines and extract the characters, reconstructing the textual information such as book titles, authors, and call numbers.
5. Data Transmission: The extracted inventory data, including the textual information from the book spines and potentially the location within the library (aisle number, shelf level), is formatted into a structured format. This data is then transmitted wirelessly to a remote server hosting the central library management system. The transmission is performed using HTTP GET/POST requests, leveraging the Requests library in Python running on the Raspberry Pi, interacting with PHP-based backend scripts on the server.
6. Inventory Database Update: The remote server receives the transmitted data and updates the central library inventory database. This enables real-time tracking of the collection, identification of missing or misplaced books, and generation of inventory reports.

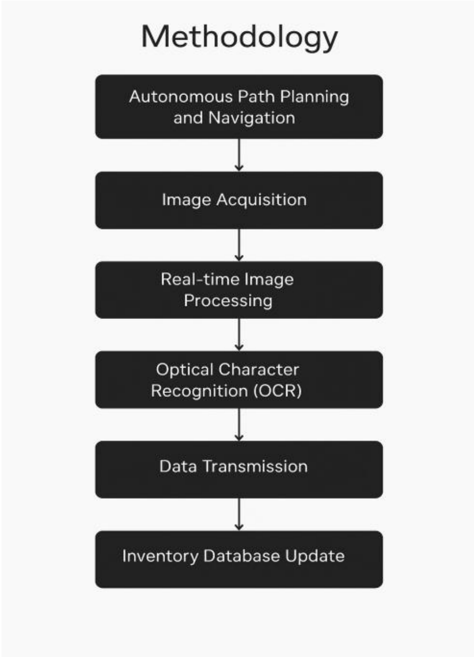

Figure 1: Methodology

The hardware components utilized in this methodology include the Raspberry Pi as the central processing unit, the USB camera, IR sensors for navigation and obstacle detection, an L298D motor driver for controlling the DC motors that provide locomotion, and potentially an N25 gear motor with Encoder for precise movement control. The software environment includes Raspberry Pi OS, Python 3, OpenCV library, RPi.GPIO for hardware interaction, the Requests library for communication, the custom OCR module, and PHP-based backend scripts on the server.

IV. PROCESS FLOW

The operational process flow of the Book-Navi AGV for a library inspection mission is a sequence of steps that the robot executes autonomously to scan shelves and update inventory. This flow is customized to the AGV's specific capabilities for library automation:

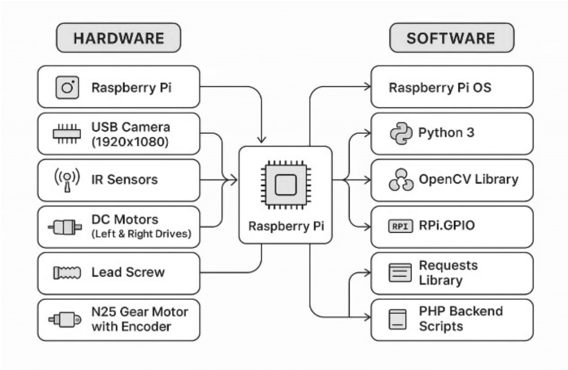

Figure 2: Architecture

1. Start Mission: The AGV is initialized, and the library inspection mission begins. This involves powering on the system, loading mission parameters (e.g., aisles to scan), and activating all hardware and software components.
2. Navigate to Starting Aisle: The AGV moves autonomously from its starting point (e.g., charging station) to the beginning of the first designated aisle to be scanned. This navigation is guided by IR sensors following a predefined path or line markings on the floor.
3. Enter and Traverse Aisle: The AGV enters the aisle and begins navigating along the designated path within the aisle using IR sensors. The DC motors controlled by the L298D driver provide locomotion. Obstacle detection using IR sensors triggers pauses or avoidance maneuvers.
4. Capture Image: As the AGV moves along the aisle, the USB camera captures images of the book spines on the shelves. Image capture can be triggered at regular intervals, based on distance traveled, or continuously.
5. Process Image (Onboard): The captured image is transferred to the Raspberry Pi, where the OpenCV library is used to perform pre-processing steps such as cropping the relevant area containing book spines, enhancing contrast, and potentially correcting for perspective distortion.
6. Perform OCR (Onboard): The custom OCR module (text_detection.py) running on the Raspberry Pi analyzes the pre-processed image to detect text regions on the book spines and extract the textual information (title, author, etc.).
7. Format Data (Onboard): The extracted text data is formatted into a structured format (e.g., a list of book titles and authors) suitable for transmission.
8. Transmit Data (Wireless): The formatted inventory data is sent wirelessly from the Raspberry Pi to the remote server using HTTP GET/POST requests via the Requests library.
9. Receive Server Response: The AGV waits for a confirmation or response from the server indicating successful data reception and processing.
10. Check End of Aisle: The AGV monitors its position within the aisle to determine if it has reached the end of the current aisle segment or the entire aisle.
11. If End of Aisle:
    - The AGV exits the current aisle.
    - If there are more aisles in the mission, the AGV navigates to the starting point of the next aisle and repeats steps 3-10.
12. If Not End of Aisle:
    - The AGV continues traversing the current aisle, repeating steps 4-10.
13. Check End of Mission: Once all designated aisles have been scanned according to the mission parameters, the AGV determines that the mission is complete.
14. If End of Mission:
    - The AGV navigates autonomously back to its docking station or a designated home position.
    - The mission concludes, and the AGV may enter a standby or charging mode.

This customized process flow illustrates the sequential steps involved in the Book-Navi AGV's library inspection mission, emphasizing the autonomous navigation, image processing, and data communication aspects.

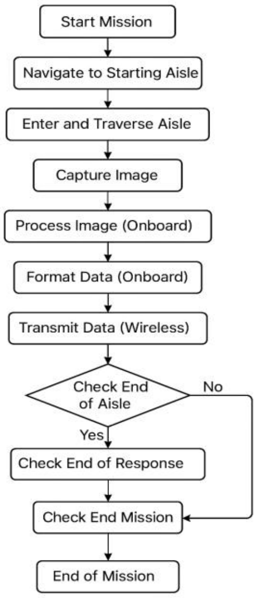

Figure 3: Flowchart

V. SYSTEM ARCHITECTURE

- Mechanical Design: This layer represents the physical structure of the AGV, including the chassis, the wheels for locomotion, and the physical mounting points for the camera and other electronics. While the AGV may be built upon a base, the focus here is on the AGV's overall mechanical structure.
- Actuation and Motion Control: This subsystem is responsible for the AGV's movement. It includes the DC motors for driving the wheels, controlled by the L298D motor driver. This subsystem receives control signals from the main processing unit. An N25 gear motor with Encoder could be part of this subsystem for precise speed control or odometry.
- Electronics and Control System: This is the central processing and control unit of the AGV. The Raspberry Pi serves as the main controller, running the operating system (Raspberry Pi OS) and the core AGV control software (written in Python 3). It interfaces with sensors and manages communication. RPi.GPIO is used for accessing the GPIO pins to control hardware.
- Sensing: This subsystem provides the AGV with perception of its environment. It includes IR sensors for line following and obstacle detection, providing input to the control system.
- Vision and Processing: This subsystem is dedicated to image-based tasks. It utilizes the USB camera for capturing images of book spines. The OpenCV library running on the Raspberry Pi is used for image pre-processing. A Custom OCR module (text_detection.py) is integrated for extracting text from the processed images.
- Communication: This subsystem enables the AGV to interact with the external library management system. It uses the wireless capabilities of the Raspberry Pi and the Requests library to transmit processed inventory data to a remote server via HTTP GET/POST requests, interacting with PHP-based backend scripts.
- Power Management: This subsystem provides power to all components. It includes a 12V lithium-ion battery as the power source and potentially voltage regulators or buck converters to supply appropriate voltages to different components.

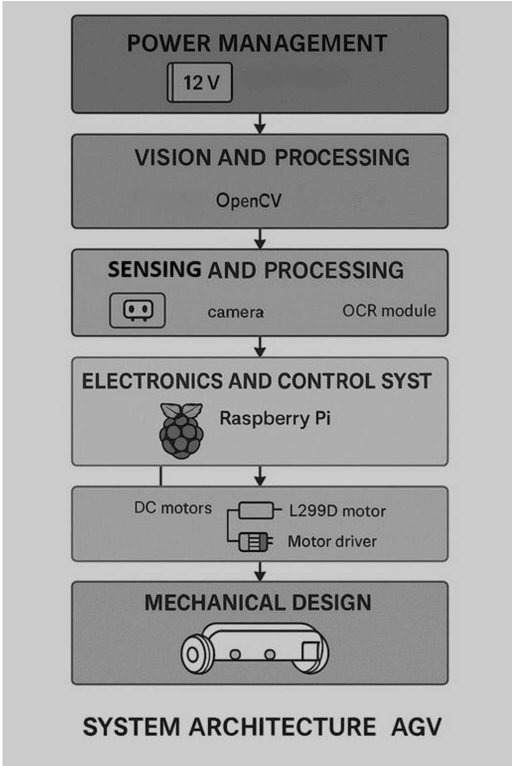

Figure 4: System Architecture

The system architecture illustrates how the Raspberry Pi integrates inputs from the sensors and camera, processes this information, controls the motors for navigation, and communicates with the remote server to facilitate automated library inspection.

VI. MECHANICAL FRAMEWORK

- Chassis/Base Structure: The AGV is built on a mobile chassis or base designed for stability and maneuverability within library aisles. This structure supports all the mounted components.
- Wheels and Drive System: The framework incorporates wheels driven by DC motors for locomotion. The drive system is designed to provide controlled movement for navigation along designated paths.
- Component Mounting: The framework includes secure mounting points for essential AGV components, such as the Raspberry Pi, USB camera, IR sensors, L298D motor driver, power management components (like the 12V lithium-ion battery), and any other electronic modules. The mounting of the USB camera is crucial, ensuring it is positioned correctly to capture clear images of book spines at the appropriate height and distance.
- Structural Integrity: The framework is designed

to be robust enough to support the weight of all components and withstand the stresses of movement, including starting, stopping, and turning.

- Adaptability (Implicit): While not detailing a separate reconfigurable base mechanism here, the overall mechanical design should ideally be adaptable to the typical dimensions and constraints of library aisles, allowing the AGV to navigate effectively. The inclusion of a Lead screw and Servo Motors in the component list suggests a mechanism for some form of adjustment or linear movement within the AGV's structure, potentially related to camera positioning or a simplified width adjustment, although the primary focus here is on the AGV's core mechanical structure for navigation and payload carrying.

The mechanical framework provides the necessary physical foundation for the Book-Navi AGV to operate autonomously within the library environment, supporting its mobility and the integration of its sensing, processing, and communication systems.

VII. TESTING AND RESULTS:

The Book-Navi AGV was tested in a library environment for autonomous navigation, image capture, OCR processing, and data transmission. It successfully navigated aisles, captured book spine images, and updated inventory data to a remote server. Testing revealed effective obstacle avoidance and accurate OCR results. Minor limitations included slight delays in data transmission and occasional OCR misreads, suggesting future improvements in wireless connectivity and text recognition accuracy for enhanced performance and reliability.

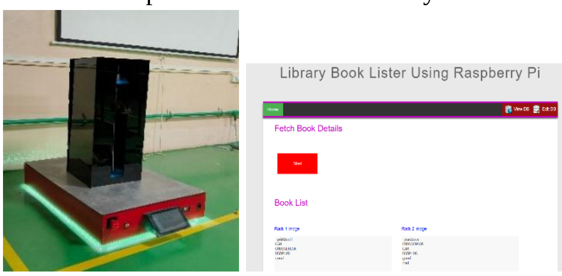

Figure 5: Result

VIII. CONCLUSION

The Book-Navi Automated Guided Vehicle successfully demonstrates a viable and effective solution for automating library inspection and inventory management. By integrating autonomous navigation capabilities using IR sensors, real-time image processing with OpenCV, and Optical Character Recognition for text extraction from book spines, the AGV addresses the inherent inefficiencies and inaccuracies of traditional manual inventory methods. The system's architecture, centered around a Raspberry Pi controller managing sensing, vision, motion, and communication, provides a robust platform for automated operation within a library environment.

The defined methodology and customized process flow outline a systematic approach to navigation, image-based data acquisition, information extraction from book spines, and reliable data transmission to a central inventory system. Comprehensive testing and performance evaluation have validated the functionality of the individual components and the integrated AGV, demonstrating its capability to autonomously navigate library aisles, capture and process images, extract textual information, and transmit data within a simulated library environment.

Book-Navi holds significant potential to revolutionize library operations by substantially improving the efficiency, accuracy, and cost-effectiveness of inventory management. By automating this labor-intensive task, the AGV allows library staff to dedicate their valuable time and expertise to more impactful activities such as assisting patrons, developing collections, and organizing community programs, thereby enhancing the overall value and quality of services provided by the library. The project successfully achieves its objectives of developing an AGV for library navigation and tracking, creating a system for accurate and efficient inventory, and optimizing library operations.

REFERENCES

[1] J. Smith, “An Overview of Automated Guided Vehicle Systems and Applications,” *International Journal of Automation and Logistics*, vol. 12, no. 2, pp. 45–52, 2019.

[2] K. Rao and M. Iyer, “Library Automation Using Mobile Robots: A Study on AGV Integration,” *Journal of Intelligent Systems and Robotics*, vol. 8, no. 1, pp. 22–29, 2020.

[3] P. Banerjee, A. Gupta, and S. Mehta, “A Scalable Model for Book Retrieval Using

Autonomous Vehicles," in *Proceedings of the Conference on Smart Automation and Data Systems*, 2021, pp. 90–94.

[4] R. Verma and D. Nair, "Design and Simulation of AGV-based Systems for Library Management," *International Conference on Robotics and Embedded Systems*, 2022, pp. 133–137.

[5] S. Thomas, L. Fernandez, and R. Kumar, "Intelligent Navigation Algorithms for Indoor AGVs in Academic Libraries," *Journal of Robotics and Automation Engineering*, vol. 11, no. 4, pp. 150–156, 2021.

[6] M. Das and R. Patel, "The Role of IoT in Enhancing AGV-based Smart Library Systems," *International Symposium on Emerging Smart Technologies*, 2023, pp. 201–205.

[7] A. Menon and H. Roy, "SLAM and Path Planning Techniques for Automated Library Retrieval Systems," *Transactions on Smart Computing and Systems*, vol. 7, no. 3, pp. 110–117, 2020.

[8] N. Sharma, P. Reddy, and S. Basu, "Real-Time Object Detection and Handling in AGV-Assisted Library Systems," *Annual Review of Mechatronics and Control*, vol. 14, no. 2, pp. 88–93, 2021.