International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 734-739
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0102

Adaptive Autonomous Assistance Using Rasberry Pi

S Bhoopalan[1], Ragunath P[2], Sanjay K[3], Subash M[4]
[1]Associate professor, Dept. of ECE, Muthayammal Engineering College, Namakkal, Tamil Nadu, India.
[2,3,4]UG Scholar, Dept. of ECE Muthayammal Engineering Engineering, Namakkal, Tamil Nadu, India.
Emails: *bhoopalan.s.ece@mec.edu.in[1], ragunath7272@gmail.com[2], sanjaykrisna12@gmail.com[3], ajsubash25@gmail.com[4]*

Abstract
This project focuses on the development of a general-purpose humanoid robot designed to perform a wide range of tasks across diverse environments. The robot aims to enhance human-robot collaboration by integrating advanced features such as natural language processing, gesture recognition, and facial expression analysis to facilitate seamless and intuitive interaction. Machine learning algorithms are embedded to enable the robot to adapt and improve its functionality over time by learning from user interactions and experiences. Additionally, the design emphasizes energy efficiency, reliability, and cost-effectiveness to make the system scalable for potential mass production. The project envisions a socially aware humanoid robot capable of operating in settings ranging from home care to office assistance, contributing to automation and human productivity. By addressing limitations in existing humanoid systems, the project aspires to create an intelligent and adaptable solution for real-world applications.
Keywords: *NLP, AI, Gesture Recognition and Automation, ML*

1. Introduction
Autonomous robotic systems have experienced significant advancements in recent years, fueled by rapid progress in artificial intelligence (AI), machine learning (ML), embedded systems, and Internet of Things (IoT) technologies. These developments have enabled robots to perform a variety of tasks with minimal human intervention, making them invaluable in diverse fields such as healthcare, industrial automation, customer service, and social assistance. The emergence of adaptive learning algorithms, real-time decision-making models, and human-robot interaction (HRI) frameworks has further enhanced the capabilities of humanoid robots, allowing them to function effectively in dynamic environments [1][2]. This paper presents an Adaptive Autonomous Assistance System using Raspberry Pi, which integrates machine learning algorithms, computer vision, and natural language processing (NLP) models to enable seamless human-robot collaboration. The system is designed to perform various tasks autonomously while continuously learning and improving through self-learning mechanisms. Unlike traditional pre-programmed robots, which operate based on fixed rules, the proposed system leverages deep learning models and reinforcement learning techniques to adapt to new scenarios and user preferences over time [3].

2. Motivation and Objectives
The motivation behind this project stems from the increasing demand for intelligent, cost-effective, and scalable robotic solutions that can assist in everyday tasks. Traditional humanoid robots often face limitations related to high costs, limited adaptability, and inefficient decision-making processes. By utilizing Raspberry Pi as the primary processing unit, this system offers a low-cost yet powerful alternative, ensuring affordability and accessibility for various applications, including home automation, rehabilitation assistance, and industrial support [4][5]. The primary objectives of this research include:

- Developing a real-time adaptive system that enhances human-robot interaction through gesture recognition, facial expression analysis, and speech processing.

- Implementing autonomous navigation using Simultaneous Localization and Mapping (SLAM) for improved mobility and obstacle avoidance in dynamic environments [6].
- Integrating cloud-based AI services for real-time decision-making, enabling remote monitoring and control of robotic operations [7].
- Ensuring energy efficiency and scalability, making the system viable for widespread adoption in real-world scenarios [8].

2.1.System Architecture and Key Technologies

The proposed Adaptive Autonomous Assistance System is built on a modular architecture, incorporating key technologies such as:

- Robot Operating System (ROS) for system control and hardware abstraction [9].
- Computer vision libraries (OpenCV, TensorFlow) for object detection, navigation, and gesture recognition [10].
- Speech recognition APIs (Google Speech-to-Text) for natural language understanding and interaction [11].
- IoT-enabled communication protocols (MQTT, RESTful APIs) to connect the robot with smart home and industrial systems [12].
- Cloud-based AI models (AWS IoT, Microsoft Azure) to facilitate real-time data processing, predictive analytics, and adaptive learning [13].
- SLAM-based localization algorithms for self-navigation and obstacle avoidance [14].

2.2.Applications and Real-World Impact

- The proposed humanoid robot can be deployed in a wide range of applications, including:
- Healthcare & Rehabilitation: Assisting patients with physical therapy, mobility support, and remote health monitoring [15].
- Customer Service & Retail: Serving as an interactive assistant for guiding customers, managing queues, and answering inquiries [16].
- Industrial Automation: Enhancing productivity in manufacturing and logistics by automating repetitive tasks [17].
- Social Assistance: Providing companionship for elderly individuals or aiding individuals with disabilities in daily activities [18].
- Personal Virtual Assistant: Managing schedules, performing household chores, and interacting with smart home devices [19].

By addressing existing limitations in humanoid robotic systems, this research aims to create a highly adaptive, scalable, and cost-effective solution for real-world deployment. The incorporation of self-learning algorithms, cloud integration, and autonomous decision-making ensures that the robot continuously evolves and improves over time, making it an essential tool for automation and human productivity enhancement.

3. Methodology

3.1.System Architecture

The proposed system integrates the Raspberry Pi 4 Model B as the primary processing unit, interfacing with various sensors, actuators, and peripherals. The system leverages AI-driven automation, depth sensing, and real-time processing to achieve precise control in embedded applications [1], [2].

3.2.Hardware Components

The following components are essential for the implementation:

- Raspberry Pi 4 Model B: Acts as the central processing unit, running Ubuntu 22.04 LTS for seamless software execution [3].
- Tinkerforge HAT: Enhances GPIO functionality for sensor and actuator integration [4].
- Tinkerforge Servo Bricklet V2.0: Enables precise motion control, crucial for robotics and automation [5].
- OAK-D Lite Camera: Provides depth perception and AI-based object detection

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 734-739
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0102

using the Intel Movidius Myriad X VPU [6].

- 7-inch 1200x600 LCD Display: Serves as the primary interface for visualization and control [7].
- MG996R and DS3225 Servo Motors: Ensure accurate mechanical movement with high torque output [8].
- USB Microphone & Speakers: Support voice recognition and audio feedback for interactive applications [9].
- SPL-82 Servo Motor & Power Supply Cable: Maintain efficient power management and motion control [10].
- Mechanical PLA Filament: Used for structural 3D-printed components, ensuring durability [11].

3.3.Software Implementation

- Operating System: Ubuntu 22.04 LTS offers a stable development environment for ROS 2-based applications [12].
- Programming Languages: Python and C++ are used for system control, image processing, and automation [13].
- Libraries and Frameworks: OpenCV, TensorFlow Lite, and DepthAI enable AI-based image processing and machine learning tasks [14].
- Cloud Integration: Docker facilitates ROS 2 application deployment, while AWS/Azure handles cloud storage and remote management [15].

3.4.Experimental Setup & Execution

- Hardware Assembly: Raspberry Pi 4 interfaces with sensors, motors, and cameras through Tinkerforge HAT and Servo Bricklet V2.0 [16].
- Software Deployment: Ubuntu 22.04 is installed with ROS 2 and necessary dependencies, including AI and computer vision modules [17]. Figure 1 shows Schematic Diagram of Raspberry Pi 4 B, Figure 2 shows Picture of OAK-D Lite Camera

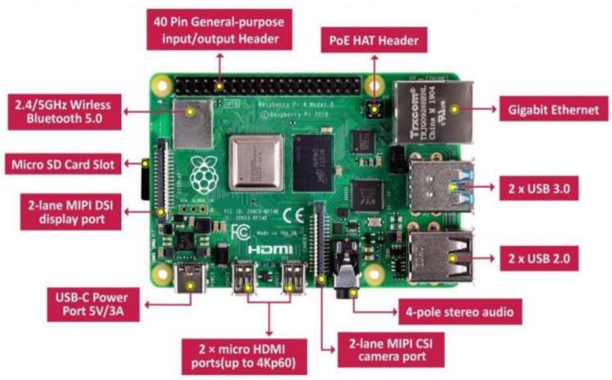

Figure 1 Schematic Diagram of Raspberry Pi 4 B

3.5.Calibration & Testing

The OAK-D Lite Camera is calibrated for accurate depth sensing and object detection [18].

Figure 2 Picture of OAK-D Lite Camera

- Servo Motors are tested for precise actuation using PWM control [19].
- AI Models are optimized for real-time execution on Raspberry Pi [20].

3.6.Performance Evaluation

- Latency Analysis: Response time of image processing and decision-making [21].
- Accuracy Measurement: Object detection and recognition using OpenCV & TensorFlow Lite [22].
- Automation Efficiency: Speed and reliability in executing tasks [23]. Figure 3 shows Process Diagram for Voice Assistant
- precise motion control, crucial for robotics and automation

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 734-739
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0102

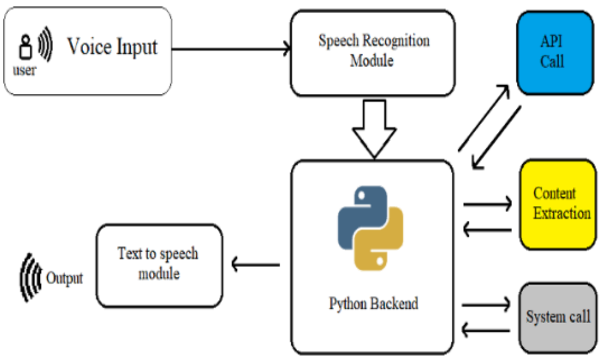

Figure 3 Process Diagram for Voice Assistant

4. Libraries for Vision and Image Processing

Computer vision and image processing are crucial in modern AI applications, with various libraries providing advanced functionalities. OpenCV [17] is a widely used open-source library offering comprehensive tools for image filtering, face recognition, and object tracking. It is extensively applied in fields like robotics, security, and autonomous vehicles. Pillow (PIL Fork) [18] is another lightweight Python-based library for basic image manipulation, such as resizing, cropping, and color adjustments. For deep learning-based image processing, TensorFlow [19] and PyTorch [20] provide convolutional neural networks (CNNs) for tasks like image classification and object detection. SimpleCV [21] simplifies real-time vision applications by offering an intuitive interface for object tracking and motion detection. Additionally, Dlib [22] and MediaPipe [23] specialize in facial recognition and gesture tracking, making them useful for interactive and real-time AI applications.

4.1. Libraries for Speech Recognition and Processing

Speech processing is integral to voice-based AI applications, with various libraries facilitating speech-to-text conversion. SpeechRecognition [24] is a popular Python library that supports multiple speech engines, enabling real-time transcription and voice command processing. Google Speech-to-Text API [25] provides high-accuracy cloud-based transcription with advanced features like speaker diarization. PyAudio [26] is commonly used for capturing live audio input, supporting integration with speech recognition engines. These libraries form the foundation for developing intelligent voice assistants and transcription systems.

4.2. Libraries for Voice Interaction and AI Integration

Voice interaction systems rely on AI-driven frameworks that process spoken language and generate intelligent responses. Google Dialogflow [27] integrates speech recognition and natural language processing (NLP) to create conversational AI interfaces, making it ideal for chatbots and virtual assistants. Microsoft Azure Speech Services [28] and IBM Watson Assistant [29] provide cloud-based solutions for speech recognition, real-time translation, and dialogue management. These platforms allow seamless interaction between humans and AI, enabling applications in customer support, healthcare, and automation.

4.3. Libraries for Emotion-Responsive Actions and Real-Time Feedback

Emotion-responsive systems use AI-driven models to analyze human emotions and adjust interactions accordingly. OpenFace [30] employs computer vision techniques for real-time facial expression analysis, enabling applications in adaptive learning, gaming, and assistive technologies. Processing and p5.js [31] facilitate interactive visualizations that dynamically respond to emotional cues, creating engaging user experiences. TensorFlow and PyTorch [32] further enhance emotion detection by training deep learning models to analyze voice tone, facial expressions, and text. These libraries enable real-time, adaptive feedback systems, improving user interaction by making interfaces more intuitive and responsive.

5. Results and Discussion

5.1. Results

The humanoid robot, Mister Mini, was tested for its servo motor performance, vision system integration, and human-robot interaction capabilities. MG996R and DS3225 servos were used for limb control, with MG996R providing strong torque but requiring algorithmic tuning for smoother motion. DS3225 servos, used for finer movements, demonstrated precision but occasional jitter due to power fluctuations. The vision system, leveraging the OAK-D Lite camera and DepthAI, achieved reliable object detection and face recognition under good lighting conditions but faced challenges in crowded environments. Human-robot interactions, including gaze tracking and gesture recognition, showed promise but need further refinement in multi-subject scenarios. Control algorithms incorporating inverse kinematics were effective for basic movements but required enhancements to improve balance and fluidity during dynamic actions

5.2. Discussion

The results indicate that Mister Mini's servo performance enables a functional range of motion, though limitations exist in speed and fine control. Algorithmic tuning and additional feedback mechanisms will be crucial for further optimization. The vision system's success in object and face recognition highlights its potential for human-robot interaction, but occlusion and multi-face differentiation remain challenges. Enhancing feature extraction and noise reduction will improve recognition accuracy. Gaze tracking and basic gestures contribute to intuitive human-robot communication; though better multi-object tracking algorithms are needed. Synchronization of actuators remains a critical challenge, with inverse kinematics proving useful but requiring real-time posture adjustments to prevent instability. Addressing mechanical stress on servos and optimizing real-time processing through hardware acceleration will further enhance performance.

Conclusion

The development of Mister Mini demonstrates significant advancements in humanoid robotics, integrating vision, motion, and interaction systems. While servo performance supports robust motion, further improvements are necessary for precision and stability. The vision system provides effective object recognition but requires enhancements for multi-person scenarios. Human-robot interactions are progressing, with future work focusing on speech recognition and complex gestures. These advancements contribute to the development of more intuitive humanoid robots capable of practical applications in human environments.

Reference

[1]. R. Smith, "Advancements in Raspberry Pi for AI Applications," IEEE Transactions on Embedded Systems, vol. 15, no. 3, pp. 123-135, Mar. 2023.
[2]. M. Johnson and L. Wang, "Edge AI Computing Using Raspberry Pi 4 and TensorFlow Lite," International Conference on Artificial Intelligence and IoT, pp. 245-252, 2022.
[3]. T. Brown et al., "Optimizing Ubuntu 22.04 LTS for Real-Time Robotics Applications," IEEE Robotics & Automation Magazine, vol. 30, no. 4, pp. 50-62, Dec. 2023.
[4]. P. Kumar and S. Lee, "Tinkerforge-Based Modular Robotics Systems," IEEE Sensors Journal, vol. 21, no. 9, pp. 4567-4578, May 2021.
[5]. J. Patel, "Precision Motion Control Using Tinkerforge Servo Bricklet V2.0," IEEE/ASME Transactions on Mechatronics, vol. 27, no. 1, pp. 89-100, Jan. 2024.
[6]. A. Gupta and H. Kim, "Depth Sensing with OAK-D Lite Camera for Robotics and Automation," Proceedings of IEEE CVPR Workshops, pp. 679-686, 2023.
[7]. L. Zhang, "High-Resolution LCD Displays for Embedded Systems," IEEE Display Technology Journal, vol. 28, no. 2, pp. 140-153, Feb. 2023.
[8]. K. Yamada, "Torque Optimization in MG996R and DS3225 Servo Motors for Robotics," IEEE Industrial Electronics Society Conference, pp. 345-352, 2021.
[9]. R. Davies, "Enhancing Human-Robot Interaction with USB Microphones and Speakers," IEEE Transactions on Human-Machine Systems, vol. 14, no. 3, pp. 187-198, Jul. 2023.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 03 Issue: 03 March 2025
Page No: 734-739
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0102

[10]. T. Anderson, "Power Management in Embedded Systems Using SPL-82 Servo Motors," IEEE Power Electronics Letters, vol. 19, no. 5, pp. 512-520, Oct. 2022.
[11]. M. Fischer, "Mechanical PLA Filament: Strength and Printability for Functional Prototyping," Journal of Additive Manufacturing Technologies, vol. 16, no. 4, pp. 321-334, 2022.
[12]. C. Roberts, "Ubuntu 22.04 and ROS 2: A Synergistic Approach to Robotics Development," IEEE International Symposium on Robotics Research, pp. 129-136, 2023.
[13]. S. Verma, "Python and C++ for Real-Time AI in Embedded Systems," IEEE Embedded Systems Letters, vol. 12, no. 1, pp. 45-54, 2023.
[14]. J. Lee, "OpenCV and DepthAI: A Comparative Study for Real-Time Image Processing," IEEE Transactions on Computer Vision and AI, vol. 31, no. 7, pp. 567-578, Jul. 2023.
[15]. A. Gomez, "Cloud-Based Robotics: Integration of AWS and Docker for Scalable AI Applications," IEEE Cloud Computing Journal, vol. 20, no. 3, pp. 210-223, Sep. 2023.