iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72173

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Multimodal LLM-Based Robotic System for Dynamic Task Execution and Human Interaction

P. Srujan Reddy[1], S. Shekar[2], P. Pranav Chandra[3], M. Sanjeev Kumar[4], Ch. Raja[5]
Department of Electronics and Communication Engineering, Mahatma Gandhi Institute of Technology-Hyderabad, India

Abstract: ***This paper presents an integrated multimodal robotic system that effectively combines state-of-the-art Large Language Models with advanced perception and control mechanisms, enabling sophisticated task execution and natural human-robot interaction. Current robotic implementations, predominantly reliant on rigid programming paradigms, demonstrate significant limitations in adaptability when confronted with complex, real-world scenarios. Our proposed architecture addresses these constraints through a comprehensive framework leveraging the contextual reasoning capabilities of multimodal LLMs. The system architecture incorporates cutting-edge models including GPT-4o and Gemini 2.0 Flash for nuanced linguistic interpretation and environmental understanding, working in conjunction with object detection systems such as YOLO and Grounding DINO to achieve robust situational awareness. Following rigorous validation in PyBullet simulations, we successfully deployed the framework on a physical platform utilizing Raspberry Pi 5 hardware with ROS 2 integration. Experimental evaluations confirm the system's exceptional performance in processing complex directives, navigating challenging environments, and executing precise manipulation tasks according to user specifications, demonstrating significant advantages over conventional approaches. This research establishes a promising foundation for next-generation autonomous systems with applications spanning industrial automation, healthcare assistance, and adaptive support technologies.***

Keywords: Human-Robot Interaction (HRI), Large Language Models (LLMs), Multimodal Robotics, Dynamic Task Execution, ROS 2, Object Detection, SLAM, Vision-Language Models (VLMs).

I. INTRODUCTION

The development of robotic systems capable of complex interaction and autonomous operation in human inhabited environments is a key goal in AI and robotics. Conventional robots, reliant on predefined programming, struggle with adaptability in dynamic settings such as homes or warehouses, where unforeseen circumstances are common [8]. Recent advances in Large Language Models (LLMs) and multimodal perception offer promising paths to more versatile and intuitive systems. However, current platforms often fail to interpret nuanced natural language commands, effectively integrate diverse sensory inputs (vision, depth, proprioception), or adapt their behavior to unforeseen situations and environmental changes [8], significantly limiting their practical utility across a wide range of collaborative tasks. This work aims to develop and validate an intelligent robotic system that overcomes these prevalent issues, enabling sophisticated, autonomous tasks guided by intuitive human interaction. This is achieved by leveraging the reasoning capabilities of LLMs, advanced computer vision, and a rich suite of sensor modalities. The core research challenge addressed is the effective, real-time integration of these heterogeneous components into a cohesive system that can operate robustly and reliably on resource-constrained hardware within complex and dynamic real-world environments. Early robotic systems established foundational capabilities in navigation (using LIDAR, cameras) and basic object manipulation (e.g., with YOLO for object detection [18]), but these systems typically operated in highly structured environments and lacked deep semantic understanding or natural interaction capabilities. The advent of LLMs (e.g., GPT-4 [17], Gemini) has introduced transformative potential, enabling robots to process natural language and perform complex reasoning for task planning and execution [1], [8]. Numerous studies explore LLM applications: [2] surveys their use in multi-robot task allocation, [3] demonstrates LLM-enhanced manipulation via human collaboration, and [4] integrates vision with LLMs for improved interaction. However, existing LLM integrations often exhibit limitations; some focus primarily on safety verification aspects [1] or specific NLP/LLM deployment techniques [5], rather than holistic, end-to-end dynamic task execution in complex and unpredictable physical settings. Multimodal approaches seek to connect language with perception to create a more holistic understanding. Wang et al. [6] present LaMI, an LLM-based system for multimodal Human-Robot Interaction (HRI), showing adaptability but potentially limited by its reliance on predefined ”atomic actions” for novel tasks. Conceptual works [7] often provide high-level overviews without presenting specific implemented systems or their empirical evaluation.

While Vision-Language Models (VLMs) [11], [12] and modular system proposals [14] show significant promise for enhancing robot perception and interaction, the efficient integration of these models for dynamic planning and control on low-power edge devices for real-time interaction remains a largely unaddressed challenge. Despite the availability of robust frameworks like ROS 2 [15] and established navigation tools such as Nav2 [16], the engineering challenge of effectively integrating these with cutting edge LLM/VLM components, diverse sensors, and custom controllers on a resource-constrained platform (e.g., Raspberry Pi 5) for complete end-to-end dynamic task execution is not adequately addressed in the existing literature. Many studies remain in simulation environments [7], focus on specific subproblems like safety [1] or navigation [16], or lack thorough hardware validation for comprehensive dynamic task execution [8], [12].

The primary contributions are:

1) *Integrated Multimodal Robotic Architecture:* A system coupling LLMs, advanced vision (YOLO, OWL-ViT, Grounding DINO), and sensors in ROS 2 for dynamic tasks.
2) *LLM-Driven Dynamic Task Execution:* LLMs interpret complex commands and generate executable robot actions, validated for dynamic tasks beyond simulation.
3) *Comprehensive Sim-to-Real Validation:* Rigorous testing of navigation, object detection, and task execution in PyBullet, validated on a Raspberry Pi 5 robot.
4) *Natural Language Interaction Modality:* A userfriendly multimodal interface (Streamlit, pyttsx3) for intuitive command and feedback.

II. METHODOLOGY

This section outlines the methodology for our Multimodal LLM-Based Robotic System, detailing its architecture, simulation environment, hardware platform, software implementation, and the operational workflow for task execution. The system is designed to integrate advanced AI with robotics to enable dynamic task handling from natural language commands and sensory input.

A. *System Architecture*

The system employs a modular architecture for efficient integration and scalability shown in Fig. 1.

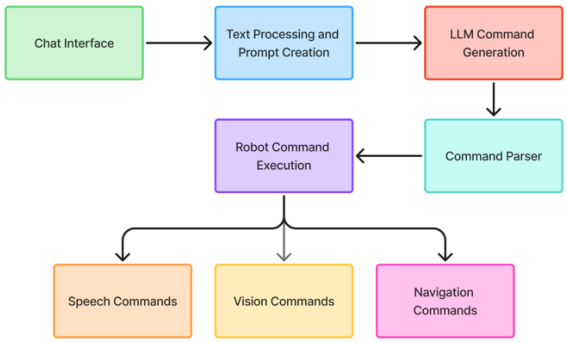


Fig 1: System Achitecture Block Diagram

1) *Input Modules:* These capture diverse data streams: textual commands via a web interface, rich visual data from a camera (for object detection, recognition), and environmental data from a 2D LIDAR (for SLAM and navigation), multiple ultrasonic sensors (for close-proximity obstacle detection), an IMU (for orientation and motion tracking), and wheel encoders (for odometry).
2) *Processing Unit (Raspberry Pi 5):* This central unit orchestrates the system's intelligence.

a) *LLM-Based Decision Making:* LLMs (e.g., GPT-4V, Gemini Pro accessed via API) interpret natural language commands. Carefully crafted prompt engineering guides the LLMs to decompose high-level instructions into a structured sequence of JSON-formatted executable task plans (e.g., ""function": "navigate", "parameters": "target_location": "kitchen""), considering the robot's capabilities and current context.

b) *Multimodal Perception:* This subsystem fuses data from multiple sensors. It uses YOLO for fast detection of known objects and VLMs (OWL-ViT, Grounding DINO) for zero-shot detection of novel objects based on textual descriptions. Camera data is crucial for object identification, pose estimation, and visual servoing, while LIDAR data is the primary input for SLAM algorithms.

c) *Task Management:* A dedicated task manager serially executes the LLM-generated action sequence, monitors the status of each sub-task, provides feedback, and implements basic error recovery strategies.

3) *Output Modules:* These facilitate interaction with the user and the physical world. They include Text-to-Speech (TTS) synthesis (using pyttsx3) for providing audible feedback and status updates, and motor/actuator control for physical mobility (via DC motors) and object manipulation (using a servo-driven gripper).

B. *Simulation Environment*

Core functionalities are extensively validated in a highfidelity PyBullet simulation environment before hardware deployment to de-risk and accelerate development.

1) *Simulator:* PyBullet is used for its accurate simulation of rigid body dynamics, contact forces, collisions, and robot kinematics, enabling rapid prototyping and iterative testing of algorithmsas shown in Fig. 2.

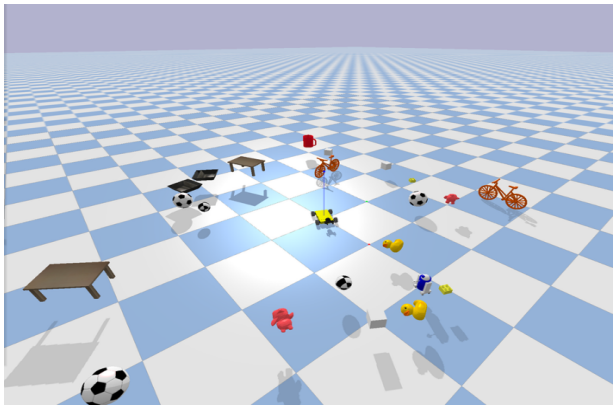


Fig 2: PyBullet Simulation Environment with the virtual robot model.

2) *Virtual Environment:* The simulation environment is designed to replicate diverse and complex real-world scenarios, populated with various objects, clutter, and obstacles to test the robot's adaptability.

3) *Robot Model:* A detailed URDF (Unified Robot Description Format) model of the robot is created, including simulated versions of all key sensors with realistic noise models to approximate real-world sensor imperfections and behavior.

4) *Validation Scope:* Key functionalities validated include navigation algorithms (path planning, obstacle avoidance), scene understanding (object recognition, spatial relationships), object detection performance shown in Fig. 3, the correctness and feasibility of LLM-generated task plans, and basic manipulation sequences.

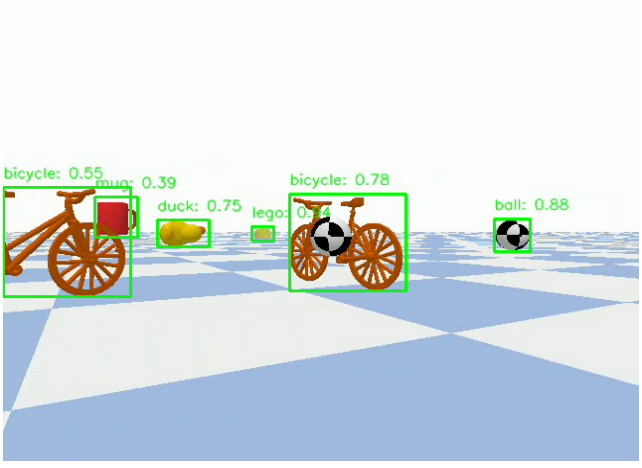


Fig 3: Object Detection using YOLO within the Simulation.

C. Hardware Platform

The physical robot is built on a robust two-wheeled differential drive chassis shown in Fig. 5, with a Raspberry Pi 5 as the central onboard computer. The circuit diagram is shown in Fig. 4.

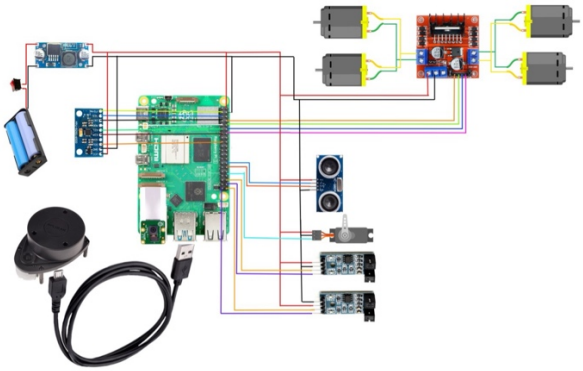


Fig 4: Circuit Diagram of the robot

1) *Compute Unit*: Raspberry Pi 5 (BCM2712 quad-core Arm Cortex-A76, 4GB RAM) running a 64-bit Ubuntu OS and ROS 2 Humble, providing sufficient processing power for onboard tasks.
2) *Sensor Suite:* Includes a Raspberry Pi Camera Module V3 (Sony IMX708, 12MP, autofocus via CSI) for vision, a 2D LIDAR (360° scan, 12m range via USB) for SLAM and navigation, multiple HC-SR04 Ultrasonic sensors for close-range obstacle detection, an MPU6050 IMU (6-axis via I2C) for orientation, and IR Encoder sensors on motor shafts for wheel odometry.
3) *Actuators:* Comprises two 12V DC geared motors controlled by L298N H-bridge drivers via GPIO PWM for mobility, and a standard servo motor controlling a custom gripper mechanism via GPIO PWM for manipulation.
4) *Power System:* A 12V 3000mAh LiPo battery powers the system, with a buck converter providing a stable 5V supply for the Raspberry Pi and other sensitive electronics.
5) *User Interface (Remote):* A Streamlit-based web interface accessible from any browser-enabled device for text commands and feedback.

Fig 5: Assembled physical Robot.

D. Software Implementation

The software stack leverages ROS 2 Humble on Ubuntu (64-bit) for its modularity, robust communication, and extensive robotics libraries.

1) *OS & Middleware:* Ubuntu OS (64-bit for Raspberry Pi), ROS 2 Humble Hawksbill.
2) *LLM Integration Layer:* API calls are made to external LLM services (e.g., GPT-4, Gemini). A custom ROS 2 node manages these API communications, formats requests, parses LLM responses, and translates them into ROS 2 actions/goals.
3) *Perception Pipeline:* OpenCV is used for basic image processing. TensorFlow Lite enables efficient inference
4) for object detectors like YOLOv5 on the Raspberry Pi. ROS 2 nodes publish detection results. VLMs (OWL-ViT, Grounding DINO) are integrated for zero-shot detection, invoked based on LLM queries.

5) *User Interaction Interface:* A Streamlit web GUI (Fig. 6) for text commands and feedback. pyttsx3 is used for text-to-speech output.
6) *Low-Level Control Drivers:* Custom ROS 2 nodes (robot_base_controller for motors, manipulation_controller for gripper) interface with hardware drivers.

E. Workflow and Task Execution

The system follows a structured operational workflow from command input to physical action and feedback, ensuring coherent response.

1) *Command Input and Ingestion:* The user provides a natural language command via the Streamlit web interface, which is then relayed to the robot's processing unit.
2) *LLM-Powered Interpretation and Decomposition:* The command is sent to the LLM API. The LLM, guided by prompt engineering, interprets the command's intent and decomposes it into a JSON array of executable functions and parameters.
3) *Task Planning and Perception Invocation:* The task manager receives the JSON actions. Perception modules (YOLO, VLMs) are invoked for relevant actions (e.g., search_object). Nav2 handles navigation actions.
4) *Sequential Action Execution and Real-time Adaptation:* Actions are executed sequentially by ROS 2 nodes. The robot uses continuous sensor feedback (LIDAR, ultrasonics, IMU) for real-time adjustments (e.g., Nav2 local planner for obstacle avoidance). Action outcomes are monitored.
5) *User Feedback and Status Reporting:* The robot provides textual feedback via the Streamlit interface and synthesized speech (pyttsx3) on its status, actions taken, and task completion.

III. EXPERIMENTAL RESULTS

System performance was rigorously assessed in the PyBullet simulation environment. This section details the evaluation of LLM capabilities in robot control and the efficacy of different object identification strategies. A suite of 50 distinct tasks was designed, categorized into 15 navigation-based tasks (e.g., "go to the area near the large table," "find the shortest path to the red block and stop 1 meter in front of it"), 15 visual-based tasks (e.g., "describe what you see in front of you," "is there a blue cup currently on the wooden table?"), and 20 general tasks combining navigation and visual understanding (e.g.,"find the book in the living room and tell me its predominant color"). Each task comprised approximately 15 to 20 sub-commands or checkpoints to assess granular performance and robustness.

A. LLM Performance Comparison

Four multimodal LLMs GPT-4o, Gemini 2.0 Flash, Llama 3.2 Vision, and LLaVA were evaluated for their ability to generate executable sub-commands, navigate environments, and interpret visual scenes. As shown in Table I, Gemini 2.0 Flash led in command generation (88%) and navigation (86%), indicating strong planning and task execution. GPT-4o followed closely with 87% in command generation, 84% in navigation, and the highest scene understanding at 82%, reflecting strong contextual reasoning. Llama 3.2 Vision performed reliably (82% command, 79% navigation, 74% scene understanding), while LLaVA trailed with lower scores across all tasks (73%, 70%, 69%), facing difficulties with complex, multi-step scenarios. Differences stem from model architecture, training diversity, and reasoning capabilities.

TABLE I
COMPARATIVE PERFORMANCE OF LLMS IN ROBOTIC TASKS

LLM Model	Command Generation	Navigation	Scene Undersatnding
Gemini 2.0 flash	88	86	77
GPT-4o	87	84	82
Llama 3.2 Vision	82	79	74
LLaVA	73	70	69

B. Object Identification Strategies with Gemini

Effective object identification is vital for robotic interaction and reasoning. This study evaluates the integration of object detection models with Gemini 2.0 Flash, a language-vision reasoning LLM.

We compare YOLO (e.g., YOLOv8), known for real-time speed with fixed object classes, against two zero-shot detectors Grounding DINO and OWL-ViT , which identify objects from arbitrary text prompts. The models were assessed on accuracy for in-distribution (ID) versus out-of-distribution (OOD) objects, and inference speed on a simulated robotic testbed. As shown in Table II, YOLO (YOLOv8n) achieved the highest ID accuracy (92%) and real-time performance (15-20 FPS), making it ideal for fast-paced tasks with known object categories. However, it performed poorly on novel objects (35% OOD accuracy), limiting adaptability.

Zero-shot models demonstrated better generalization. OWL-ViT achieved 70% accuracy on OOD objects, outperforming Grounding DINO (65%), with slightly lower ID performance (OWL-ViT: 88%, DINO: 85%). Their key limitation lies in slower inference speeds (OWL-ViT: 3-6 FPS, DINO: 4-8 FPS), which can hinder real-time operation. Both required descriptive prompts (e.g., "a red apple"), which were generated contextually by Gemini 2.0 Flash based on user commands or task reasoning. This dynamic prompting enabled flexible object detection in unfamiliar scenarios. Ultimately, model choice hinges on task demands: YOLO for known, time-sensitive environments; OWL-ViT or Grounding DINO for open-world generalization, with Gemini acting as the reasoning bridge across modalities.

TABLE II
PERFORMANCE OF OBJECT DETECTION MODELS INTERGRATED WITH GEMINI 2.0 FLASH

Model	Accuracy (ID)	Accuracy (OOD)	Speed
YOLO	92%	35%	15-20 FPS
Grounding DINO	85%	65%	4-8 FPS
OWL-ViT	88%	70%	3-6 FPS

IV. CONCLUSIONS

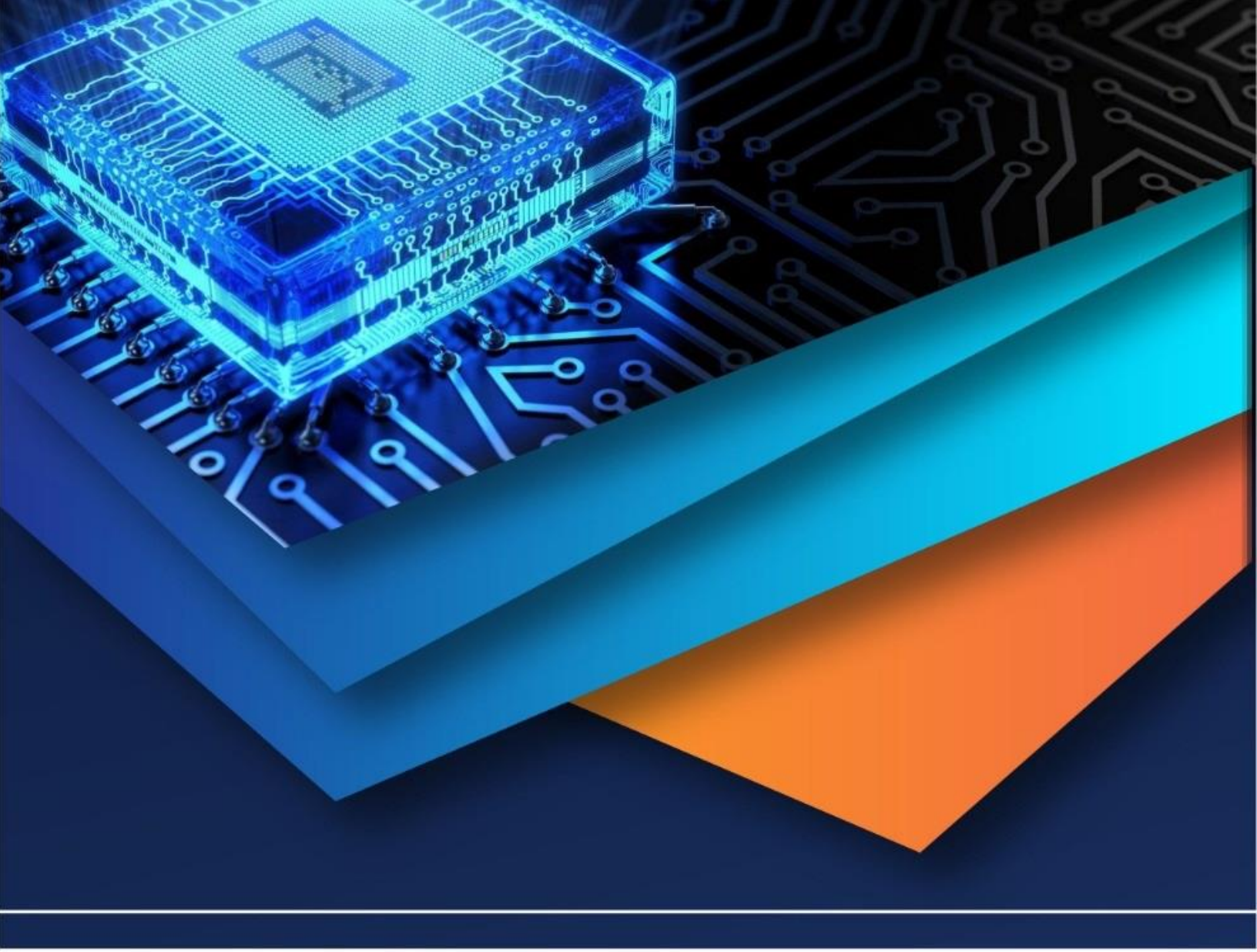
Simulated evaluations in PyBullet demonstrate the potential of integrating multimodal LLMs with robotic systems for complex tasks. Models like Gemini 2.0 flash show promise in interpreting commands, generating actions, and demonstrating proficient navigation and scene understanding. This highlights LLM advancements and their applicability to robotics. Object identification investigation reveals a trade-off: YOLO excels in speed for known objects, while zero-shot models like OWL-ViT and Grounding DINO, guided by an LLM, offer superior flexibility for novel objects, crucial for dynamic human environments.

This work supports creating more intuitive and adaptable robots. LLM-driven task decomposition, coupled with robust perception, enables robots to handle a wider range of requests. While simulation results are foundational, future work includes sim-to-real transfer and hardware validation. Challenges in real-time performance on constrained hardware, safety, and seamless integration of perception, reasoning, and action remain key research areas.

REFERENCES

[1] R. Zhang, A. Gupta, J. Zhu, K. Gopalakrishnan, and A. Faust, ”Safety Aware Task Planning via Large Language Models in Robotics,” arXiv preprint arXiv:2503.15707, 2025. [Online]. Available: https://arxiv.org/abs/2503.15707
[2] Y. Wu, Z. Chen, H. Zhang, M. Chen, J. M. Alonso, and C. Chen, ”Large Language Models for Multi-Robot Systems: A Survey,” arXiv preprint arXiv:2502.03814, 2025. [Online]. Available: https://arxiv.org/abs/2502.03814
[3] H. Liu, Y. Zhu, K. Kato, A. Tsukahara, I. Kondo, T. Aoyama, and Y.Hasegawa, ”Enhancing the LLM-Based Robot Manipulation Through Human-Robot Collaboration,” IEEE Robotics and Automation Letters, vol. 9, no. 8, pp. 7165-7172, Aug. 2024.
[4] H. K. Omeed, A. O. Alani, I. H. Rasul, A. M. Ashir and S. A. Mohammed, ”Integrating Computer Vision and language model for interactive AI - Robot,” in 2024 21st International Multi-Conference on Systems, Signals & Devices (SSD), Erbil, Iraq, 2024, pp. 124-131.
[5] P. Sikorski, L. Schrader, K. Yu, L. Billadeau, J. Meenakshi, N. Mutharasan, F. Esposito, H. AliAkbarpour, and M. Babaiasl, ”Deployment of NLP and LLM Techniques to Control Mobile Robots at the Edge: A Case Study Using GPT-4-Turbo and LLaMA 2,” arXiv preprint arXiv:2403.05381, 2024.
[6] C. Wang, S. Hasler, D. Tanneberg, F. Ocker, F. Joublin, A. Ceravola, J. Deigmoeller, and M. Gienger, ”LaMI: Large Language Models for Multi-Modal Human-Robot Interaction,” arXiv preprint arXiv:2401.15174, 2024.
[7] R. K. Thaker, ”Generative AI and Robotics: From Large Language Models to Intelligent Human-Robot Interaction and Task Planning,”International Journal of Innovative Research in Management, Programming, and Sliding Shapes (IJIRMPS), vol. 12, no. 4, pp. 1-10, Jul.-Aug.2024.
[8] J. Wang et al., ”Large Language Models for Robotics: Opportunities, Challenges, and Perspectives,” arXiv preprint arXiv:2401.04334, 2024.
[9] S. Alzahrani, N. Aldoahman, F. Bou Nassif and A. Bou Nassif, ”LLM-Based Edge Intelligence: A Comprehensive Survey on Architectures, Applications, Security and Trustworthiness,” arXiv preprintarXiv:2409.05217, 2024.
[10] M. Waseem and A. Orouskhani, ”Object Recognition for NAO Robot in Webots Simulation Environment: A Comparative Study between YOLO (Doubao Vision Understanding Model),” International Journal of Innovative Research in Computer and Communication Engineering, 2024.
[11] S. Minaee et al., ”A Review of 3D Object Detection with Vision-Language Models,” arXiv preprint arXiv:2504.18738, 2025.
[12] X. Han et al., ”Multimodal Fusion and Vision-Language Models: A Survey for Robot Vision,” arXiv preprint arXiv:2504.02477, 2025.

[13] H. Su, W. Qi, J. Chen, C. Yang, J. Sandoval, and M. A. Laribi, ”Recent advancements in multimodal human–robot interaction,” Frontiers in Neurorobotics, vol. 17, p. 1084000, 2023.
[14] G. Babu and E. Sathiyanarayanan, ”Revolutionizing Human-Robot Interaction (HRI): Multimodal Intelligent Robotic System for Responsive Collaboration,” International Journal of Intelligent Systems and Applications in Engineering (IJISAE), vol. 12, no. 17s, pp. 464-473, 2024.
[15] A. Alharbi, B. Alahmadi, M. Alharthi, A. Alsubaie, R. Babli, B. Alsolai, M. Baljoon, N. Mohammed, E. Serrano, H. Vega, T. Alhmiedat and J. M. Alonso, ”ROS 2 Key Challenges and Advances: A Survey of ROS 2 Research, Libraries, and Applications,” Preprints.org, 2024101204, 2024. DOI: 10.20944/preprints202410.1204.v2..
[16] J. Stewart, ”ROS 2 Robot With SLAM,” University of Cape Town, Dept. of Electrical Engineering, B.Sc. Mechatronics Report, Oct. 2024.
[17] T. B. Brown et al., ”Language Models are Few-Shot Learners,” in Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020.
[18] J. Redmon and A. Farhadi, ”YOLOv3: An Incremental Improvement,” arXiv preprint arXiv:1804.02767, 2018.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY