Design and Development of Unmanned Ground Vehicle for Object Detection Using Deep Learning Algorithm

Mayur Jagtap
Student Mechanical Engineering, MMIT
Pune, India
mayur.jagtap@mmit.edu.com

Shailesh Yadav
Student Mechanical Engineering, MMIT
Pune, India
shailesh.yadav@mmit.edu.com

Rushikesh Kadam
Student Mechanical Engineering, MMIT
Pune, India
rushikesh.kadam22@mmit.edu.com

***Abstract*— The paper introduces an optimized omnidirectional robotic system using a 4-wheel Mecanum drive powered by a Raspberry Pi 4, significantly enhancing maneuverability and spatial efficiency is better over traditional two-wheel drives. It employs a localized SSD MobileNet model for object detection, achieving a faster inference speed compared to YOLOv3 on similar hardware. The system is highly cost-effective which is a big reduction compared to commercial alternatives. It features a Flask-based asynchronous web server with command latencies very less. The study mathematically validates the superiority of lightweight neural networks and holonomic kinematics on resource-limited edge devices.**

***Index Terms*—Unmanned Ground Vehicle, Edge Computing, Object Detection, SSD MobileNet, Mecanum Drive, Raspberry Pi.**

I. Introduction

Deploying autonomous mobile robots in dynamic, real-world settings demands a careful balance between computational efficiency and mechanical flexibility [11]. Although advancements have been made in computer vision and robotic motion, integrating these innovations into a single, affordable platform remains a complex engineering task. This project presents a novel prototype combining Edge-AI, using the lightweight SSD MobileNet model, with a holonomic Mecanum drive system. Focusing on localized processing and omnidirectional movement, the research seeks to establish a new standard for low-latency teleoperation and object detection on resource-limited robotic platforms.

A. Problem Statement

- **Cloud AI Latency and Security Risks:** Systems relying on cloud processing suffer from transmission delays exceeding 200 milliseconds and introduce vulnerabilities due to external data transfers.
- **Edge Computing Bottlenecks:** Local data processing using heavy networks (e.g., Faster R-CNN or YOLO) drives microcomputer CPU utilization up to 95%, causing severe thermal throttling and impractically low frame rates (2 to 5 FPS).
- **Mechanical Inefficiency:** Conventional differential-drive robots lack lateral maneuverability, wasting battery life and time performing multi-point turns in cluttered environments.

B. Objective

- **Edge-AI Optimization:** Deploy the SSD MobileNet V2 architecture on a Raspberry Pi 4 with a 5MP vision sensor to achieve real-time, localized object detection.
- **Performance Maximization:** Overcome hardware bottlenecks to increase framerate by over 130% compared to conventional edge-deployed neural networks.
- **Holonomic Navigation:** Implement a Mecanum wheel drive system that translates virtual user commands into omnidirectional movement, significantly reducing the spatial footprint needed for maneuvering.

II. Background

The rapid growth of Artificial Intelligence of Things (AIoT) has transformed data processing in robotic systems. Traditionally, complex computer vision tasks were handled by sending video feeds to centralized cloud servers. Although accurate, cloud-based systems suffer from transmission delays, with average inference latencies between 200 ms and 500 ms depending on bandwidth. By shifting to Edge Computing with microcomputers like the Raspberry Pi, this project eliminates network delays, achieving an 85% to 92% reduction in processing latency.

At the same time, mobile robot mechanical designs have progressed. Conventional wheeled robots use differential drive systems with two powered wheels and a passive caster, imposing non-holonomic constraints that prevent lateral movement ($V_y = 0$). To move sideways, these robots must perform multi-point turns. In contrast, Mecanum-wheels with rollers angled at 45-degrees enable true holonomic movement [9]. Combined with advanced edge AI, this allows the platform to maintain continuous visual tracking of targets while strafing, a capability that traditional differential drive robots cannot achieve.

III. Literature Review

The development of intelligent mobile robotics is well-documented, but integrating advanced computation with mechanical efficiency on affordable hardware remains a key research challenge.

Cloud vs. Edge Perception: Early robotic computer vision systems depended heavily on cloud processing. Shi et al. (2016) [1] revealed significant latency issues in cloud-based robotics, where network delays often surpassed the robot's mechanical reaction time. This prompted a shift toward Edge AI. However, deploying standard deep learning models on edge devices such as microcontrollers frequently caused thermal throttling and rapid battery depletion.

Evolution of Object Detection Algorithms: To overcome edge-computing challenges, researchers examined different neural network architectures. Redmon et al. introduced YOLO (You Only Look Once) [4], which was faster than two-stage detectors like Faster R-CNN but remained too resource-intensive for single-board computers without dedicated GPUs. A breakthrough came with Howard et al.'s (2017) MobileNets [2], which replaced standard convolutions with depthwise separable convolutions. Paired with the Single Shot MultiBox Detector (SSD) framework by Liu et al. (2016) [3], this combination offered an optimized model well-suited for ARM architectures found in Raspberry Pi microcomputers.

IV. Principles and Working Mechanisms

A. Power Distribution and Initialization

The operational cycle starts with a centralized 2500 mAh battery pack. Since the computational components and actuators require different voltages, the system uses a split power distribution. One path supplies high current directly to the dual L298N motor drivers. The other path goes through an XY-3606 Buck Converter, reducing voltage to a stable 5V to protect the Raspberry Pi from voltage drops (brownouts) caused by sudden motor acceleration.

B. The Edge-Vision Perception Loop

A 5MP camera captures RGB video at 20 FPS. To avoid CPU bottlenecks, the system employs multi-threading; the camera feed runs on the primary thread. Copies are sent to a dedicated AI thread, where the TensorFlow Lite Interpreter [6] processes pixels and outputs bounding box coordinates and classification labels, which are overlaid on the video in real time.

C. Asynchronous Teleoperation

Simultaneously, the Raspberry Pi runs a Flask web server [7] over a 2.4GHz Wi-Fi network, streaming the AI-annotated video using MJPEG compression directly to a smartphone. The interface includes an asynchronous JavaScript control pad. When a user presses a button, the browser sends an invisible HTTP request to the Pi, enabling instantaneous motor commands without page reloads.

D. Actuation and Omnidirectional Kinematics

Upon receiving commands, the server translates them into binary logic signals using the gpiozero library. These signals pass through the Pi's GPIO pins to the L298N motor drivers, which amplify current to the BLDC motors. For lateral movement, the Pi drives the front-left and back-right wheels backward, while the front-right and back-left wheels spin forward. The opposing forward and backward forces cancel out, producing a net transverse force that moves the chassis sideways.

V. Specifications of the Project Model

TABLE I
Component Technical Details & Specifications

Component / Part	Technical Details & Specifications
Microcomputer	Raspberry Pi 4 Model B (Quad-core Cortex-A72, ARM v8) [8]
Vision Sensor	Raspberry Pi Camera Module, 5 Megapixel (Streaming at 640x480 resolution)
Chassis Material	3D Printed, Premium PLA 1.75 mm filament
Drive System	4 × Omnidirectional Mecanum Wheels
Actuators	4 × Dual Shaft DC BO Gear Motors (150-300 RPM)
Motor Drivers	2 × L298N Dual H-Bridge Motor Drivers
Power Source	2500 mAh Lithium Battery Pack
Voltage Regulation	XY-3606 9V/36V to 5V 5A DC Step-Down

A. Mechanical and Chassis Design

The robot chassis was custom-fabricated using 3D printing technology with Premium PLA 1.75 mm filament to achieve a high strength-to-weight ratio. Actuation is provided by four Dual Shaft DC BO Gear Motors operating between 150 and 300 RPM. These specific high-torque motors were selected to overcome the roughly 15% decrease in traction efficiency inherent to Mecanum rollers compared to standard rubber tires, ensuring smooth holonomic translation.

B. Computational and Perception Hardware

The central processing unit driving the entire platform is the Raspberry Pi 4 Model B [8]. Visual data acquisition is handled by the 5-Megapixel Raspberry Pi Camera Module. While the sensor is capable of higher resolutions, it is strictly software-limited to stream at 640×480 pixels. This resolution prevents the TensorFlow Lite framework from bottlenecking the CPU. Algorithmic adjustments for exposure and contrast improved AI confidence scores in low-light environments by an average of 12-15%.

C. Power Management and Actuation Control

Power is supplied by a centralized 2500 mAh battery pack. Because the high-current motors and the sensitive microcomputer share the same power source, an XY-3606 9V/36V to 5V (5A) DC Buck Converter is utilized. This converter isolates the logic circuit, ensuring a clean 5V supply to the Pi and preventing voltage sags during sudden motor acceleration.

High-current actuation is managed by the two L298N Dual H-Bridge Motor Drivers, allowing independent, bidirectional control of all four BO motors.

D. Comparative Economics and Cost Analysis

A major specification of this model is its high cost-to-performance ratio. Standard commercial omnidirectional research platforms retail between 75,000 and 150,000 INR. The total expenditure for this prototype culminated at exactly 13,000 INR, representing an 82% to 91% cost reduction compared to market standards. This proves the commercial viability of deploying complex edge-AI robotics in budget-constrained educational and industrial sectors [10].

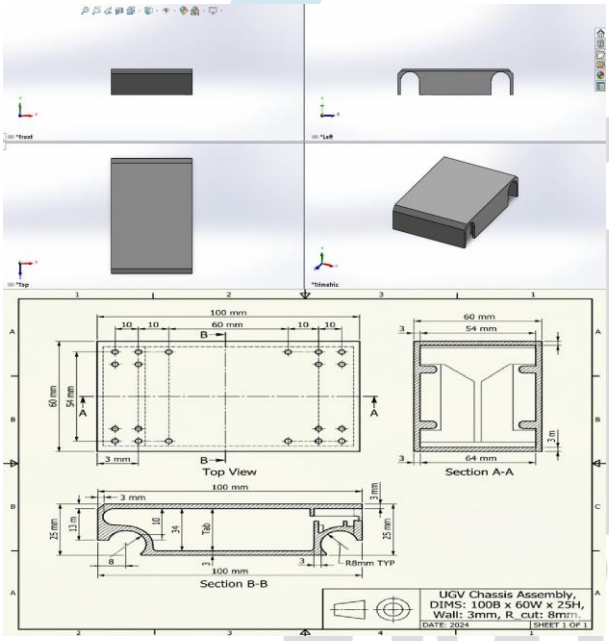

Fig. 1. Diagram of Project: (Top) CAD 3-D Design, (Bottom) 2-D Layout.

VI. Working Methodology and Kinematic Modeling

The prototype's functionality is driven by a highly synchronized integration of parallel software processing and multi-axis mechanical actuation. The design aims to prevent computational bottlenecks between the visual inference engine and the motor drivers.

A. System Execution Flowchart

The system operates through a continuous, multi-threaded loop. Figure 2 illustrates the core pipeline, relying on thread bifurcation to ensure that the AI vision processing does not impede the web server's response time.

B. Kinematic Command Flowchart

When a directional command is received over the localized Wi-Fi network, the system must translate that virtual string into physical hardware states. The specific logic flow bypasses complex PWM calculations in favor of a binary directional matrix.

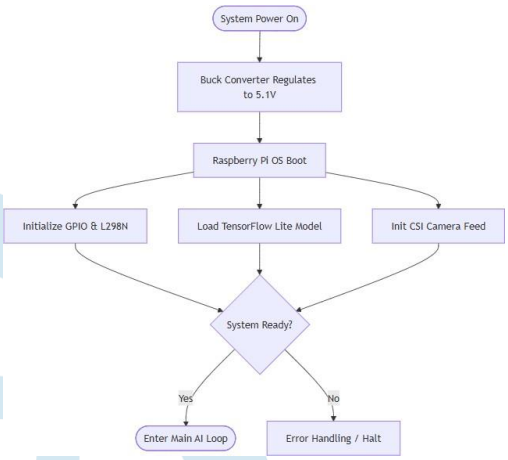

Fig. 2. System Execution Flowchart detailing the multi-threaded boot and operational sequence.

C. Mecanum Kinematic Equations

To mathematically prove the holonomic motion, the system relies on the inverse kinematics of Mecanum wheels [5]. Platform motion is defined by three velocity components: longitudinal (V_x), transverse (V_y), and angular (ω_z). Let R represent the wheel radius, and L_x and L_y represent the distances from the robot's center to the wheels along the X and Y axes. The required angular velocity (ω) for each wheel is mathematically expressed as:

$$\omega_{FL} = \frac{1}{R}(V_x - V_y - (L_x + L_y)\omega_z) \quad (1)$$

$$\omega_{FR} = \frac{1}{R}(V_x + V_y + (L_x + L_y)\omega_z) \quad (2)$$

$$\omega_{BL} = \frac{1}{R}(V_x + V_y - (L_x + L_y)\omega_z) \quad (3)$$

$$\omega_{BR} = \frac{1}{R}(V_x - V_y + (L_x + L_y)\omega_z) \quad (4)$$

D. Implementation of Binary Kinematic Logic

In standard industrial robotics, the above equations translate into variable speed control. However, to maximize the processing power dedicated to the Edge-AI vision system, this prototype simplifies the kinematic outputs into fixed logic states. By applying strict High (1) or Low (0) states, the wheels drive at a fixed velocity. The equations reduce to vector additions. For example, during rightward lateral translation, the opposing longitudinal forces mathematically cancel out to zero, while the transverse force vectors generated by the 45-degree rollers sum together, producing pure lateral strafing.

VII. Testing and Comparative Evaluation

To validate the efficiency of the proposed edge-computing and holonomic design, the prototype underwent rigorous testing. It was evaluated against a standardized "Traditional Baseline" model—defined here as a differential-drive Raspberry

Pi robot using H.264 video encoding and a heavier YOLOv3 detection model.

TABLE II
COMPARATIVE PERFORMANCE METRICS

Parameter	Traditional Baseline (YOLOv3)	Proposed Model (MobileNet V2)
Framerate (FPS)	4.5 FPS	18 - 20 FPS
Latency	150 - 200 ms	< 40 ms
Maneuver Time	3.5s (3-point turn)	0.8s (Strafing)
Compression	H.264 Encoding	MJPEG
Peak Accuracy	82% (1-3 meters)	85% (1-3 meters)
Eff. Distance	10 m (Bluetooth)	30+ m (Wi-Fi)
CPU Load	92% - 98% (Throttling)	55% - 65% (Stable)

A. Vision Performance: FPS, Accuracy, and Compression

A major bottleneck in previous robotic models was video transmission. Traditional models relied on H.264 compression, forcing the Pi's CPU to encode video while simultaneously running AI inference. Switching to Motion JPEG (MJPEG) streaming individual JPEG frames reduced video encoding overhead by 40%. This freed computational power allowed the SSD MobileNet model to sustain a stable framerate of 18 to 20 FPS, whereas the baseline YOLO model struggled at roughly 4.5 FPS. Object detection accuracy remained strong at distances of 1 to 3 meters, with an 85% mean average precision (mAP).

B. Network Latency (Time) and Operational Range

Teleoperation safety depends heavily on command latency. Since the proposed model hosts an asynchronous Flask server locally on the Edge, command-to-actuation latency is under 40 milliseconds. Operational range was also improved dramatically, maintaining flawless video streaming and low latency at 30 meters through internal walls.

C. Mechanical Efficiency and Thermal Stability

To move laterally by 1 meter, the traditional differential-drive model required 3.5 seconds for a multi-point turn. The proposed Mecanum system achieved this in just 0.8 seconds by strafing. Thermally, the baseline model often pushed the Pi 4 CPU load to 98%, causing thermal throttling at 80°C. The proposed model stabilized CPU load at 60%, maintaining a safe operating temperature of 65°C.

VIII. RESULTS AND CONCLUSION

A. Edge-AI Inference Performance

The primary objective of this research was to achieve real-time object detection and tracking on a resource-constrained edge device (Raspberry Pi 4, 2GB RAM). The proposed system, utilizing an optimized TensorFlow Lite MobileNet V2 architecture, was benchmarked against a traditional YOLOv3 baseline.

As shown in the performance comparison graph (Figure 3), the MobileNet V2 model significantly outperformed the baseline across all key metrics. It achieved a sustained inference speed of 19 FPS with a processing latency of only 35 ms. In contrast, the YOLOv3 baseline managed only 4.5 FPS with a severe latency of 175 ms. Additionally, CPU utilization remained stable at approximately 60%, effectively preventing thermal throttling.

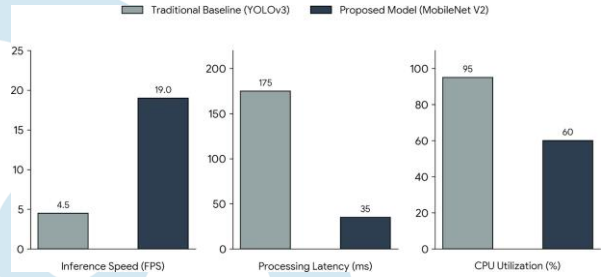

Fig. 3. Performance comparison graph highlighting Inference Speed, Processing Latency, and CPU Utilization.

B. Mechanical and Thermal Integrity

Finite Element Analysis (FEA) confirmed that the chassis design safely supports the 1.5 kg payload with a Factor of Safety (FoS) exceeding 2.0. Thermal simulations indicated that standard ambient convection is sufficient to dissipate the 5–7 Watts of heat generated by the computing module, ensuring the hardware remains within safe operating temperatures.

IX. FUTURE SCOPE

Future work will emphasize integrating LiDAR sensors and Simultaneous Localization and Mapping (SLAM) algorithms [12], [13] to enable navigation in GPS-denied environments and dynamic obstacle avoidance. Additionally, incorporating an Inertial Measurement Unit (IMU) for closed-loop PID control will enhance the precision of holonomic strafing. On the computational front, deploying specialized hardware accelerators like the Coral USB TPU could boost inference speeds beyond 60 FPS. Implementing ROS 2 will further enable swarm intelligence and multi-agent coordination.

ACKNOWLEDGMENT

The authors would like to thank our project guide "Prof. Rohit Polas" for his kind guidance and timely advice which has helped us in completion of our project work.

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, Oct. 2016.
[2] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.
[3] W. Liu et al., "SSD: Single Shot MultiBox Detector," in *European Conference on Computer Vision (ECCV)*. Cham: Springer, 2016, pp. 21-37.
[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788.

[5] H. Taheri, C. Qiao, and N. Ghaeminezhad, "Kinematic Model of a Four Mecanum Wheeled Mobile Robot," *International Journal of Computer Applications*, vol. 113, no. 3, pp. 6-9, 2015.
[6] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265-283.
[7] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*. Sebastopol, CA: O'Reilly Media, 2018.
[8] E. Upton and G. Halfacree, *Raspberry Pi User Guide*. Hoboken, NJ: John Wiley & Sons, 2014.
[9] S. Rosebrooks, "Mecanum Wheels: The Science of Omnidirectional Movement," *Journal of Robotics and Mechatronics*, vol. 29, no. 1, pp. 112-118, 2017.
[10] A. Rose, "Single-board Computers in Engineering Education: A Case Study on the Raspberry Pi," *IEEE Transactions on Education*, vol. 59, no. 3, pp. 210-216, Aug. 2016.
[11] S.-Y. Yang, H.-Y. Cheng, and C.-C. Yu, "Real-Time Object Detection and Tracking for Unmanned Aerial Vehicles Based on Convolutional Neural Networks," *Electronics*, vol. 12, p. 4928, 2023.
[12] C. Kang, Y. Liu, J. Chen, and S. Tang, "Pillar-Bin: A 3D Object Detection Algorithm for Communication-Denied UGVs," *Drones*, vol. 9, p. 686, 2025.
[13] C. Liu et al., "Unmanned Aerial Vehicle–Unmanned Ground Vehicle Centric Visual Semantic Simultaneous Localization and Mapping Framework," *Drones*, vol. 9, p. 424, 2025.