Journal of Innovations in Engineering Education
JIEE 2025, Vol. 8, Issue 1.

https://doi.org/10.3126/jiee.v8i1.82571

THAPATHALI CAMPUS
Institute of Engineering
Tribhuvan University

Enhancing Humanoid Robot functionality through vision-based navigation with fall recovery and object manipulation

Bal Krishna Shah[a,*], Sabin Acharya[a], Safal Karki[a], Aayush Pathak[a] and Saroj Shakya[a]

[a]*Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, Tribhuvan University, Kathmandu, Nepal*

ARTICLE INFO

Article history:

Received 31 July 2025
Revised in 23 December 2025
Accepted 28 December 2025

Keywords:

Autonomous navigation
Degree of Freedom
Distance estimation
Humanoid Robot

Abstract

The robotics sector struggles to integrate vision-based navigation on a bipedal humanoid robot capable of performing human-like tasks. Although the use of ultrasonic sensors and infrared sensors is a traditional method for object detection, it has significant drawbacks such as low range, high cost and sensitivity to the environment. "Enhancing Humanoid Robot Functionality Through Vision-Based Navigation with Fall Recovery and Object Manipulation" proposes to give vision to the robot, making it capable of transporting objects from one location to another. The two ESP32-CAMs are used as a stereo camera for image capturing, employing the use of YOLOv11 for object detection, and the principle of the stereo camera for depth calculation. With the use of one of the most robust and accurate object detection algorithms available, the project aims to enhance object transportation within the visual range of the robot. The final robot can navigate intelligently and grab objects using image processing. The developed humanoid robot encompasses the feature of automatic fall recovery in simulation and natural human movement patterns through kinematical calculations, showcasing potential applications in hazardous environments, industrial automation and interplanetary exploration.

©JIEE Thapathali Campus, IOE, TU. All rights reserved

1. Introduction

The concept of a humanoid robot has been around for quite a long time and has long captured the imagination of researchers and the public. Humanoid robots are designed to replicate human form and functionalities, with human-like structures that incorporate two arms, two legs, a torso, and a head. With evolving automation systems, the relevance of the integration of human-like machines in the industrial and service sectors has increased, along with expectations of improved task efficiency, accuracy, and autonomy [1]. The popularity of humanoid robots, increased by Sci-Fi movies, has further fueled interest and innovation in this field.

Practical developments of humanoid robots, such as Hanson Robotics' Sophia, capable of facial expressions and conversation, and Boston Dynamics' bipedal systems capable of dynamic movements, including jumping and rolling, show the real-world potential of humanoid robots [2][3]. Similarly, events like Robo-One in Japan demonstrate innovation and agility in bipedal robots through competitions, encouraging the practical and entertainment value of these systems.

A typical humanoid robot is designed to mimic the human structure by implementing major joints such as the neck, shoulder, elbow, hand, pelvis, knee, and ankle, each requiring its own motor and range of motion [4]. A key advantage of humanoid robots lies in their structure, which enables them to operate in human-centric environments without the need to redesign existing tools and infrastructures [1]. As AI, machine learning, and hardware continue to advance, the use of techniques such as reinforcement learning and imitation learning allows robots to learn and replicate complex human behaviors from the data [5][6].

However, significant challenges remain in developing humanoid robots, particularly in the domain of vision-based navigation. Unlike wheeled robots, bipedal robots need to maintain balance while navigating uneven and dynamic environments [7]. Real-time decision mak-

*Corresponding author:
bal.077bei010@tcioe.edu.np (B.K. Shah)

ing, visual perception and object manipulation require accurate sensors and high-performance computing [8], which are often expensive and difficult to integrate efficiently and effectively within the constraints of a humanoid platform.

This work aims to develop a cost-effective humanoid robot addressing the challenge of integrating vision-based navigation for object recognition using a stereo-camera setup. The robot is designed to recognize objects, estimate distance using stereo vision, and perform autonomous navigation, pick and place task utilizing YOLOv11 object detection model [9], and geometric methods for distance estimation, allowing it to navigate visually to the object's location [7][10].

The novelty of this work lies in the integration of low-cost stereo vision using ESP32-CAM modules with a lightweight YOLOv11n-based perception pipeline and a servo-driven humanoid platform, enabling vision-guided navigation and object manipulation using affordable hardware. While the system demonstrates reliable performance for short-range perception and basic locomotion, its capabilities are limited by actuator torque constraints, predefined motion trajectories, and reduced depth accuracy at longer distances, which are addressed as directions for future improvement.

2. Related works

Vision-based perception and object manipulation have been extensively studied in mobile and humanoid robotics. An integrated framework combining SLAM-based navigation and vision-guided grasping using YOLO-GGCNN was proposed in [11], demonstrating effective object detection and grasp prediction in unknown environments. Similarly, object detection and recognition techniques for robotics pick-and-place tasks have been explored using feature extraction and classification methods to localize objects in clustered scenes [12]. These approaches highlight the importance of vision-driven perception for manipulation tasks but often rely on computationally intensive architectures and high-end hardware platforms.

Stereo vision has been widely adopted for distance estimation in robotic systems due to its passive sensing nature. In [13], a stereo camera setup was used to estimate the distance between a camera and a human face using pixel disparity, achieving acceptable accuracy at short ranges while requiring frequent calibration. Although stereo vision provides depth perception without active sensors, its accuracy degrades with increasing distance and limited image resolution, which remains a practical challenge for compact robotic platforms.

Vision-based navigation and path planning for humanoid robots have also been investigated. A vision-based obstacle detection and path-finding system for the NAO humanoid robot was developed using Support Vector Machines for classification and A* algorithms for path planning [14][15]. While effective in structured environments, such systems typically depend on proprietary hardware and predefined vision sensors, limiting their adaptability and accessibility for low-cost platforms.

Kinematic modeling and motion generation form the foundation of humanoid robot locomotion. Geometric inverse kinematics solutions for bipedal robots with varying degrees of freedom have been presented in [16], simplifying analytical computation through Denavit–Hartenberg parameterization. More comprehensive kinematic models for high-DOF humanoids, including the Bioloid Premium platform, have been developed to achieve precise motion control of both upper and lower body segments [17]. Other studies have focused on biologically inspired humanoid designs, emphasizing lower-limb kinematics, distributed control architectures, and sensor integration for improved stability and motion execution [18]. Beyond kinematics, motion planning and balance control strategies have been explored through walking pattern generation, trajectory planning, and stability criteria such as Center of Gravity (COG) and Zero Moment Point (ZMP). Vision-based following systems using depth sensors and PID control have also demonstrated effective relative positioning between robots in dynamic environments [19][20]. More recently, learning-based approaches such as rapid motor adaptation using reinforcement learning have shown promising results in dynamic legged robots, enabling terrain adaptability and payload variation handling, albeit at the cost of significant computational resources and complex training pipelines [21].

While these studies provide valuable insights into humanoid robot perception, locomotion, and manipulation, many rely on simulation-heavy workflows, proprietary platforms, or expensive sensing hardware, limiting real-world accessibility and reproducibility. In contrast, the work presented in this paper focuses on a cost-effective, modular humanoid robot design using off-the-shelf components, integrating stereo vision and deep-learning-based object detection to achieve autonomous navigation and manipulation within practical hardware constraints.

3. System overview

The overview of system is shown in Figure 1.

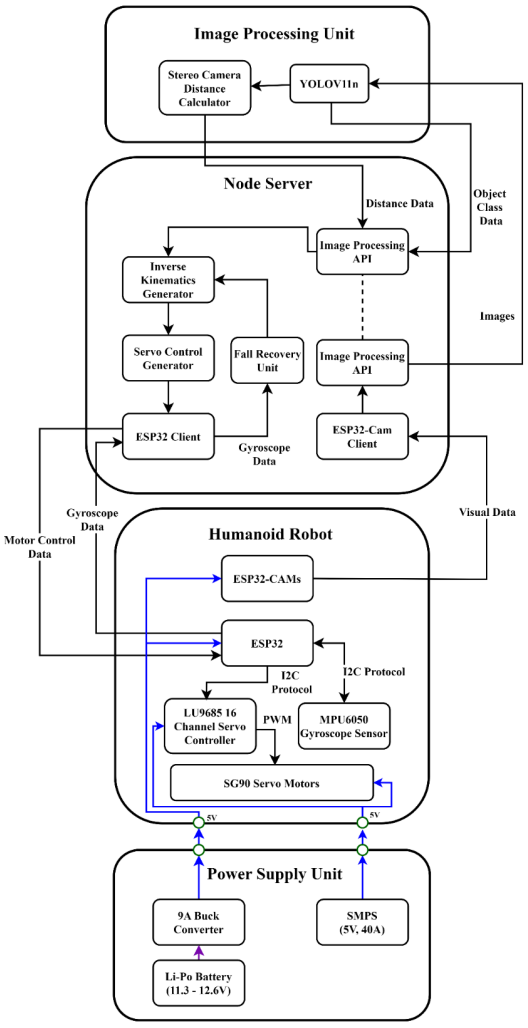


Figure 1: Block Diagram of the System

3.1. Humanoid Robot

The Humanoid Robot is a robotic system made up of a LU9685 16-Channel Servo Controller to control the servo motors, MG996R servo motors to move the robot's joints, and ESP32-CAMs to record images. An ESP32 micro-controller uses the generated commands from the computer's server to control the robot's arms and legs. An MPU6050 gyroscope (IMU) sensor is included to identify the equilibrium state. The mechanical structure of the robot is fabricated using 3D-printed components.

3.2. Node Server

WebSockets facilitate real-time data exchange between the Node.js server and connected devices. The servo control generator, equipped with essential data for the ESP32 to operate servo motors, receives instructions from the Image Processing Unit. Additionally, the Node.js server transmits image data from the ESP32-CAM to the Image Processing Unit for analysis and processing.

3.3. Image Processing Unit

A distance calculator that determines the separation between the item and the robot is part of the image processing unit. This unit determines the distance by using the YOLOv11 to recognize objects in the stream that was received from the stereo camera configuration.

3.4. Power Supply Unit

The power supply unit is responsible for delivering electrical energy to the entire humanoid robot system. It comprises a 3S Li-Po battery, a 9A Buck Converter, and a 5V 40A Switch Mode Power Supply, ensuring reliable and efficient power management.

4. Materials and methods

4.1. Robot design and kinematics

The robot comprises 19 DOFs, each consisting of a servo motor to represent its joints. The division of Degree of Freedom in each part of the robot's body is as below (Figure 2):

- **Head:** 1 DOF for Neck Joint
- **Torso:** 0 DOFs (Circuits and Controllers)
- **Arms:** 3 DOFs in each arm for a total of 6 DOFs with 2-Shoulder, 1-Elbow
- **Legs:** 6 DOFs in each leg for a total of 12 DOFs with 3-Pelvis, 1-Knee, 2-Ankle

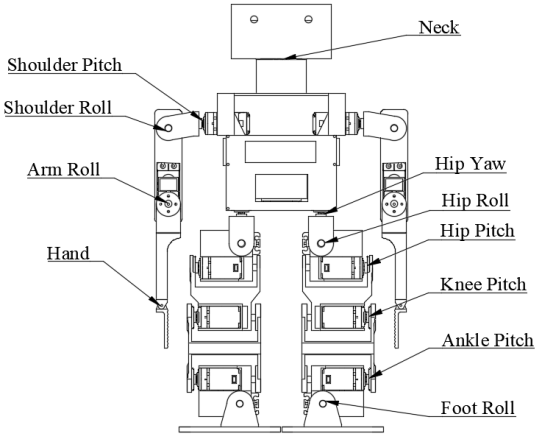


Figure 2: Robot's design with all DOFs

The coordinated actuation of leg joints enables forward, lateral, and rotational locomotion, while arm joints facilitate object grasping and assist in balance during walk-

ing. The neck DOF allows horizontal head rotation to support object detection and distance estimation using the stereo camera system. Robot motion is generated using forward and inverse kinematics derived in simulation and adapted for real-world execution. Joint trajectories are represented as discrete angle frames, with intermediate values obtained through linear interpolation to ensure smooth motion. The kinematic structure of the robot is modeled using the Denavit-Hartenberg (DH) convention, as illustrated in Figure 3.

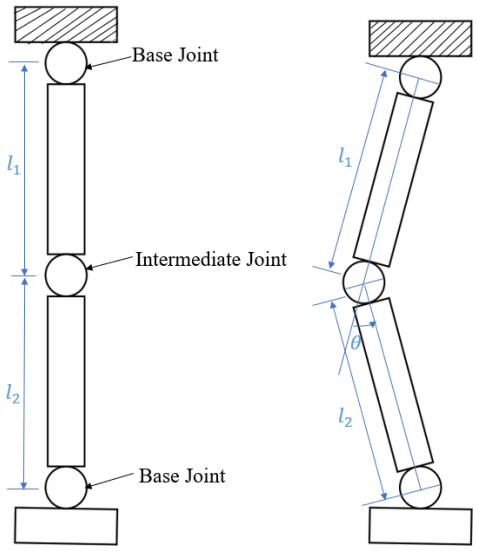


Figure 3: Basic Kinematics Structure of the Robot

For each joint, the homogeneous transformation matrix is defined using Equation 1.

$$T_{n-1}^{n} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n \cos\alpha_n & \sin\theta_n \sin\alpha_n & r_n \cos\theta_n \\ \sin\theta_n & \cos\theta_n \cos\alpha_n & -\cos\theta_n \sin\alpha_n & r_n \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The overall transformation from the base frame to the end effector is obtained by sequential multiplication of individual joint transformations.

$$T_0^4 = T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot T_3^4 \tag{2}$$

Each leg is modeled as an independent kinematic chain, with the left and right legs being mirror configurations. The corresponding DH parameters for a single leg are summarized in Table 1. The computed joint angles are mapped directly to servo actuators to generate walking, rotation, and manipulation motions. Basic gait stability is maintained by lowering the robot's center of gravity and increasing the effective support polygon through foot placement. Symbolic DH parameters are presented to preserve generality across mirrored leg configurations [22].

Table 1: DH-Parameters Configuration for Legs

Joint Pair	d [m]	θ [deg]	r [m]	α [deg]
Hip_Yaw – Hip	d_1	θ_1	0	90
Hip – Thigh	d_2	θ_2	a_2	−90
Thigh – Knee	0	θ_3	a_3	0
Knee – Ankle	0	θ_4	a_4	0
Ankle – Feet	d_5	θ_5	a_5	90
Feet – FootTip	d_6	θ_6	a_6	90

4.2. Electronic Architecture

The robot is controlled by an ESP32-WROOM-32 microcontroller, which manages sensing, communication, and motion commands. Since the ESP32 lacks sufficient PWM outputs to drive all 19 servo motors, a 16-channel servo controller (LU9685-20CU) based on the STC8H microcontroller is used. The servo controller communicates with the ESP32 via the I^{2}C protocol. An MPU6050 IMU is connected on the same I^{2}C bus to provide accelerometer and gyroscope data for motion monitoring. The overall electronic architecture and circuit configuration are shown in Figure 4.

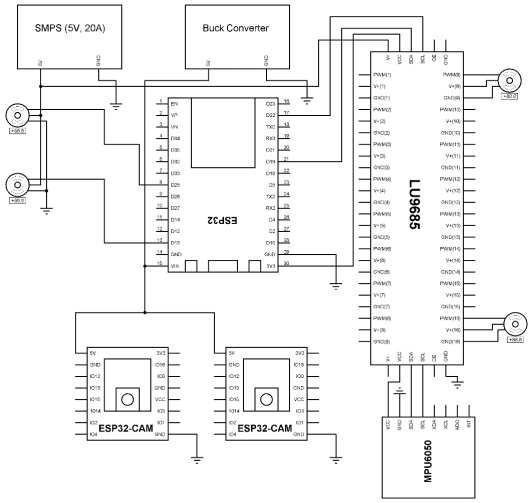


Figure 4: Circuit diagram of the electronic system of the Robot

Servo motors are powered using a 5V, 20A SMPS. To avoid controller resets caused by voltage drops during current transients, the ESP32 and ESP32-CAM mod-

ules are powered separately through a 5V buck converter supplied by a 3-Cell Li-Po battery. The ESP32-CAM modules operate independently of the motor power circuit and transmits images directly to a remote server via Wi-Fi, while the ESP32 receives motion commands wirelessly.

4.3. Object detection and distance estimation

Object detection is performed using YOLOv11n model trained on a custom cube dataset. The trained model is deployed on a remote server using the ONNX runtime, enabling inference within a Node.js environment. Object bounding boxes are detected independently from the stereo camera pair, as shown in Figure 5, and used for distance estimation.

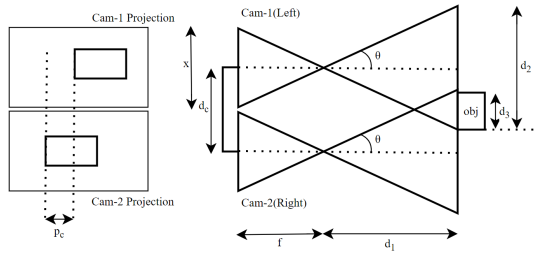


Figure 5: Stereo Camera setup for distance estimation

Distance to the object is computed using the horizontal pixel disparity between the bounding boxes obtained from the two cameras. The distance d is calculated using Equation 3.

$$d = f + \frac{N d_c}{2 p_c \tan\theta} \tag{3}$$

where f is the camera focal length, p_c is the pixel offset between stereo images, N is the image width in pixels, and d_c is the baseline distance between the cameras. The parameters $\tan\theta$ and f are calibrated using two known reference distances d_1 and d_2 with corresponding pixel offsets p_1 and p_2, as expressed in Equation 4 and Equation 5.

$$\tan\theta = \frac{N d_c}{2\left(d_1 - d_2\right)} \left(\frac{1}{p_1} - \frac{1}{p_2}\right) \tag{4}$$

$$f = d_1 - \frac{N d_c}{2 p_1 \tan\theta} \tag{5}$$

4.4. Control logic and fall recovery

The robot employs a hybrid control strategy combining closed-loop and open-loop control. Closed-loop control is used during navigation to compensate for motion inconsistencies arising from actuator variability, surface friction, and mass distribution. After each motion cycle, visual feedback from the stereo vision system is used to re-localize the target object and correct positional drift.

Open-loop control is applied during the final approach and grasping phase due to mechanical constraints of the robot. The head mechanism provides only a single rotational degree of freedom about the vertical axis, limiting vertical field-of-view adjustment. When the object moves outside the camera's view at close range, a predefined forward motion followed by a grasp sequence is executed without further visual feedback. This strategy assumes consistent gait execution over short distances.

Fall detection and recovery are based on data from the MPU6050 IMU. Pitch (θ) and roll (ϕ) angles are computed using accelerometer readings.

$$\theta = \tan^{-1}\left(\frac{A_y}{\sqrt{A_y^2 - A_z^2}}\right) \tag{6}$$

$$\phi = \tan^{-1}\left(\frac{A_x}{\sqrt{A_x^2 - A_z^2}}\right) \tag{7}$$

If either angle exceeds a predefined threshold for a sustained duration, the robot is classified as fallen and the current motion is halted. Due to hardware limitations, the fall recovery motions are implemented and validated in simulation only, while the physical robot performs fall detection and notification through a mobile application.

4.5. Simulation setup

To simulate the robot's motion dynamics, the designed humanoid model was exported from the CAD environment in .OBJ format with non-essential components merged to reduce mesh complexity. The optimized model was imported into CoppeliaSim and scaled to match the real-world dimensions of the physical robot. To improve simulation stability and computational efficiency, the model was convex decomposed using CoppeliaSim's built-in tools, converting complex geometries into primitive convex shapes.

Robot joints were manually defined according to the kinematic structure, with each joint configured in position control mode to emulate servo motor behavior. Physical parameters, including component mass and joint torque limits, were adjusted to approximate real-world dynamics. Motion control and joint behavior were implemented using the simulation environment's scripting interface, while rigid-body dynamics and collision handling were managed using the PyBullet physics engine.

5. Implementation

5.1. Simulation implementation

To accurately emulate real-world behavior, the mass distribution and joint torque values in the simulation were matched to the physical robot. Component masses, including motors and batteries, were assigned such that the total robot mass was approximately 2 kg, while joint torque limits were set to 6 Nm. Joint motions were programmed using Lua scripts, where predefined joint angle sequences were executed frame by frame. Each motion sequence consisted of multiple frames, with the robot transitioning between states based on the implemented control logic.

5.2. Motor initialization

Motor initialization is required to maintain an upright posture before executing any motion. During initialization, each servo motor is set to its predefined reference angle through the servo controller, ensuring stable load distribution across joints. Without this step, the robot cannot support its own weight and collapses due to unbalanced torque. Initialization is performed at system startup by sequentially assigning the starting joint angles to all motors.

5.3. Dynamic environment implementation of Data from walking simulation

Walking motions were executed in a dynamic simulation environment incorporating gravity, friction, and collision effects. Although the robot was stable under static conditions, rapid transitions between motion frames caused excessive momentum and instability. To mitigate this, the execution time for each frame was increased, reducing abrupt joint movements. Additional intermediate frames were introduced between unstable transitions to maintain balance during motion execution. These intermediate configurations were generated through manual motion refinement, resulting in improved stability throughout the walking cycle.

5.4. Physical implementation of pick-drop motion

Motion data obtained from simulation could not be directly applied to the physical robot due to differences in mechanical compliance and surface interaction. The bending angle of the robot was therefore adjusted to prevent forward collapse before executing the pick-and-place sequence. After calibration, the arm motion closely followed the simulated trajectory, enabling successful object pickup.

5.5. Physical implementation of rotation motion

Rotation motions derived from simulation were transferred to the physical robot with minor adjustments. To achieve smooth and stable rotation, the transition time between frames was increased to compensate for actuator limitations and environmental variations. This approach resulted in consistent rotational motion under real-world conditions.

6. Results and discussion

6.1. Object detection model training and performance

The YOLOv11n (nano) object detection model was trained on a custom cube dataset intended for robotic manipulation. Preliminary testing using non-target objects (e.g., coffee bottles and foam cubes) was conducted to assess generalization before final training on the cube dataset. The final dataset comprised 10,719 training images, 602 validation images, and 257 test images, manually annotated using assisted labeling techniques. Data augmentation was applied to improve robustness under varying illumination and viewing conditions. The model was trained for 110 epochs, achieving reliable object detection confidence suitable for real-time distance estimation (Figure 6).

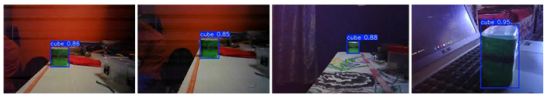
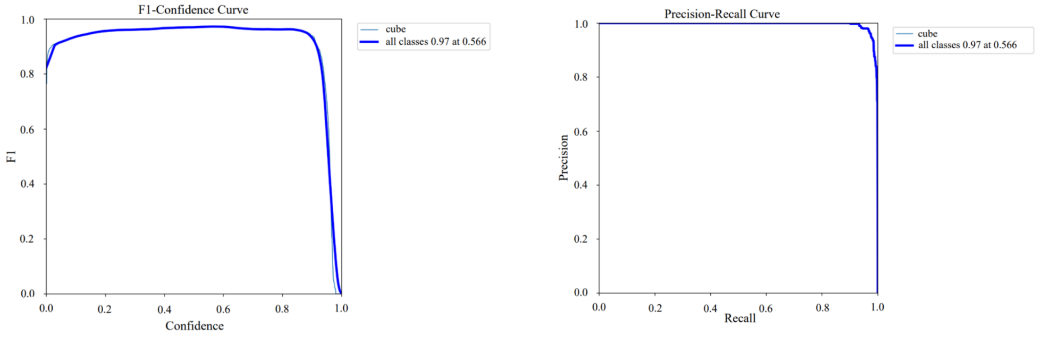


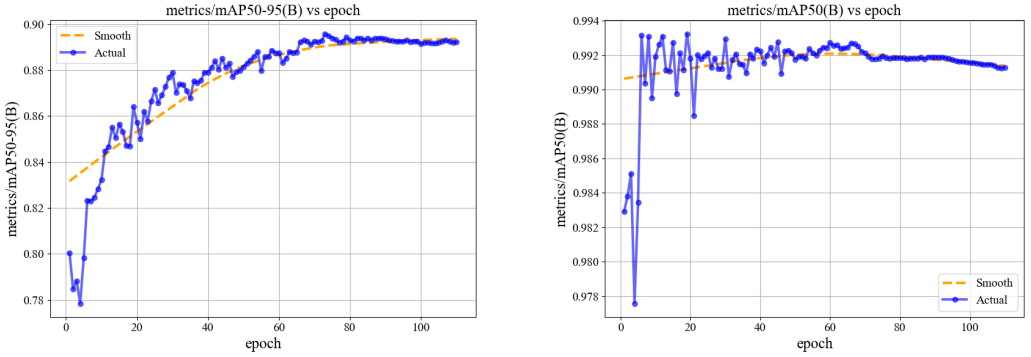
Figure 6: Detection of Cubical object by trained Model

Detection performance was evaluated using standard metrics. The F1-confidence curve (Figure 7a) shows a near-unit area under the curve, indicating balanced precision and recall across confidence thresholds. Similarly, the Precision-Recall curve (Figure 7b) demonstrates consistently high classification accuracy. Bounding box localization performance is illustrated in Figure 8a and Figure 8b, where mAP50-95 reached 88.9%, indicating robust generalization across IoU thresholds, while mAP50 peaked at 99.3%, reflecting excellent detection capability under relaxed overlap constraints.



(a) (b)

Figure 7: F1 (7-a) and PR-Curve (7-b) of trained model



(a) (b)

Figure 8: mAP50-95 (8-a) and mAP50 (8-b) of model for Bounding Box

6.2. Simulation results

All robot motions were validated in simulation prior to physical implementation. Walking, rotation, sideways motion, pick-and-place actions, and fall recovery behaviors were successfully executed using frame-based joint trajectories. Fall recovery sequences performed as intended in simulation for both forward and backward fall scenarios (Figure 9). Minor discrepancies were observed due to physics engine limitations, particularly in friction and contact modeling, highlighting the inherent differences between simulated and real-world environments.

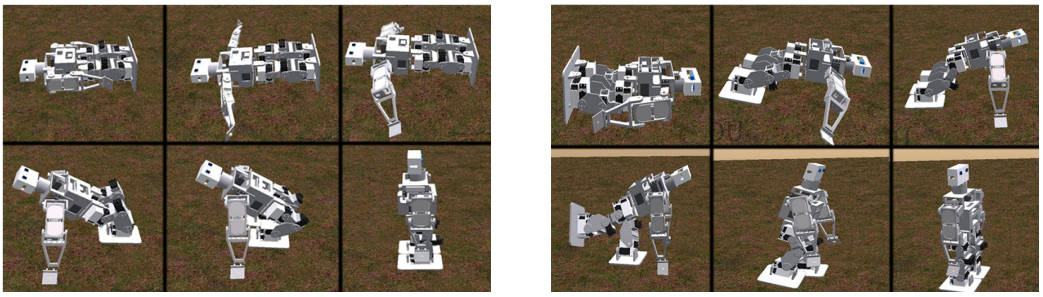
6.3. Distance estimation performance

Distance estimation accuracy was evaluated using the stereo camera setup across multiple trials. The system achieved its highest accuracy within the 10–60 cm range, where calculated distances closely matched ground truth values. For example, an actual distance of 29 cm was estimated as 28 cm. At larger distances (e.g., 75 cm), estimation error increased due to reduced pixel disparity. Excluding the farthest measurement, the average distance estimation accuracy was 93.37%, decreasing to 90.52% when all data points were included (Figure 10).

Accuracy degradation at longer distances was primarily caused by the limited resolution of the ESP32-CAM, which reduced detectable pixel shifts and lowered detection confidence. Lowering the confidence threshold from 80% to 70% and increasing resolution from HVGA to VGA partially mitigated this issue, improving bounding box coverage and distance estimation consistency.

6.4. Physical Robot performance

The assembled robot consists of one SG90 micro servo for the neck and eighteen MG996R metal gear servos for major joints (Figure 11). Motor initialization ensured a stable standing posture before motion execution (Figure 12a). After calibration, the robot achieved stable forward walking, rotation, and sideways motion, with motion sequences adapted from simulation and refined for real-world execution (Figure 12b and Figure 13).



(a) (b)

Figure 9: Fall recovery in simulation

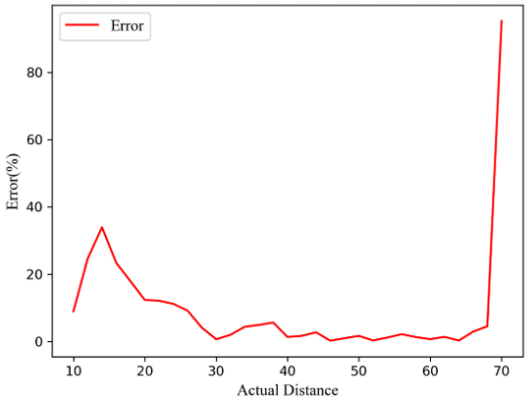


Figure 10: Error in Distance Estimation

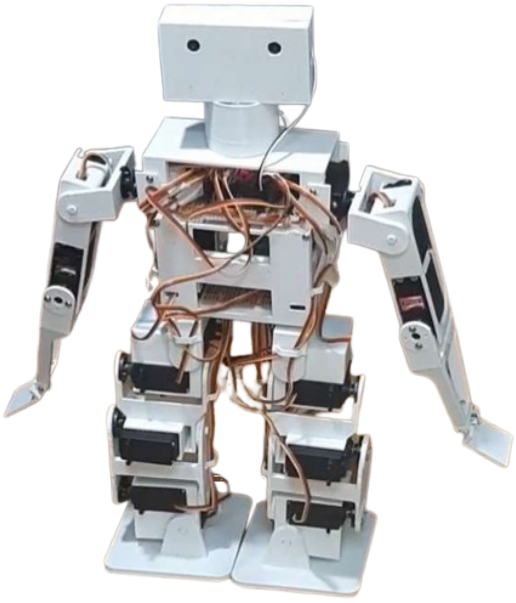


Figure 11: Assembled Robot

The robot covered approximately 30 cm over five walking cycles, with an average displacement of 5.5 cm per cycle and a positional variation below 1.5 cm (Figure 14). Rotational motion achieved an average turning angle of 12° per cycle, enabling effective alignment with target objects. Sideways motion allowed lateral correction when objects were not centered within the camera's field of view.

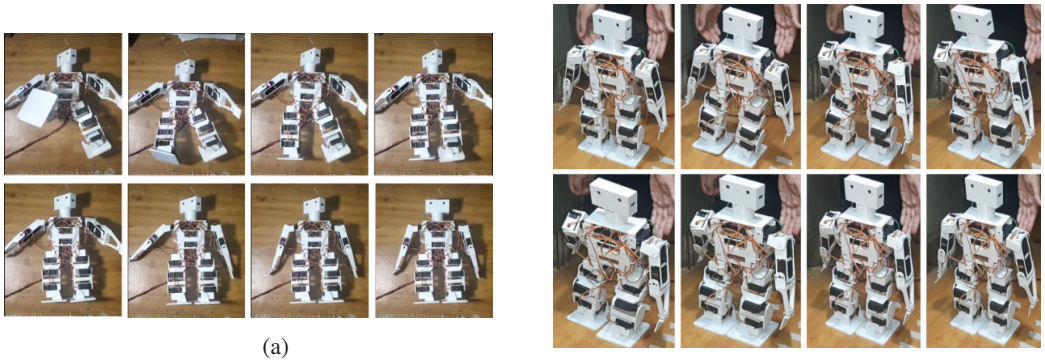
Pick-and-drop actions were successfully executed after modifying the bending component of the motion to maintain stability (Figure 15). Removing excessive torso bending enabled reliable grasping and placement of the target cube while preserving balance.

6.5. Simulation vs real world comparision

Notable differences were observed between simulation and physical execution due to idealized simulation conditions and real-world mechanical constraints. Motion smoothness, fall recovery capability, and distance estimation accuracy varied between environments. A summary comparison is provided in Table 2, highlighting key deviations in torque handling, sensory accuracy, and motion execution.

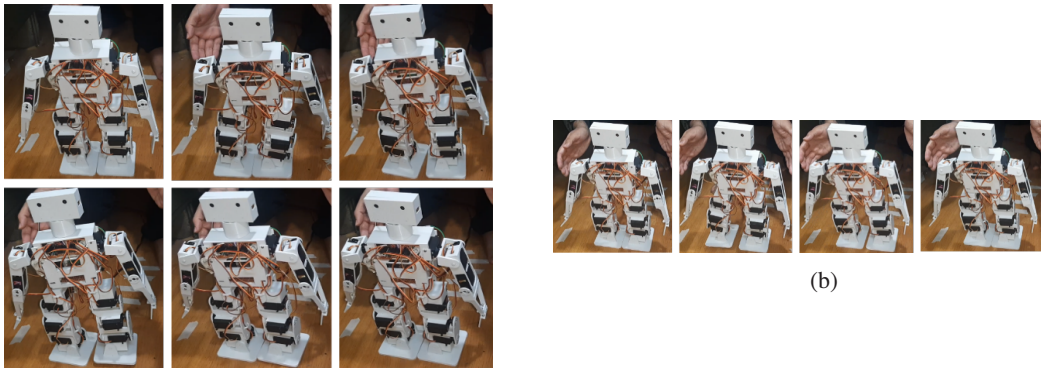
6.6. Technical challenges

Three primary technical challenges were identified. First, a trade-off between distance estimation accuracy and computational performance was observed, as higher image resolutions improved stereo accuracy but increased processing load and thermal stress on the ESP32-CAM modules. Second, servo angle variability introduced cumulative errors during locomotion; with up to $\pm 2°$ deviation per joint, a single leg could experience an effective error of up to 12°, significantly affecting walking stability. Third, the object detection model was trained on a cube-specific dataset to enable reliable proof-of-concept validation under limited re-



(a) (b)

Figure 12: Motor Initialization (12-a) and Forward Walking (12-b) motions



(a) (b)

Figure 13: Rotation (13-a) and Sideways (13-b) motions

Table 2: Simulation vs real world comparison

Parameter	**Physical**	**Simulation**
Motor torque	4.8 kg cm	5 kg cm
Fall recovery	Not implemented	Implemented
Distance estimation	Below 15 cm and above 60 cm, low accuracy	Highly accurate
Per-turn rotation	12 degrees	13 degrees
Pick-up motion	Cannot bend to pick the object	Can bend to pick the object
Walking stability	Transition frames for stability	No transition frames used
Smoothness	4-ms delay between motions	No delay required
Visual sensor	Stereo camera setup	Vision sensor
Real-time streaming	May lag	No lag

sources, which constrained immediate generalization to other object types. Importantly, the perception framework is modular, allowing object generalization through retraining or replacement of model weights without hardware modification.

7. Conclusion

This study presented the design and implementation of a low-cost humanoid robot capable of vision-based navigation and object manipulation using stereo vision and lightweight deep-learning models. A pair of ESP32-CAM modules was employed as a stereo camera system, demonstrating reliable object detection and distance estimation despite hardware constraints. While distance

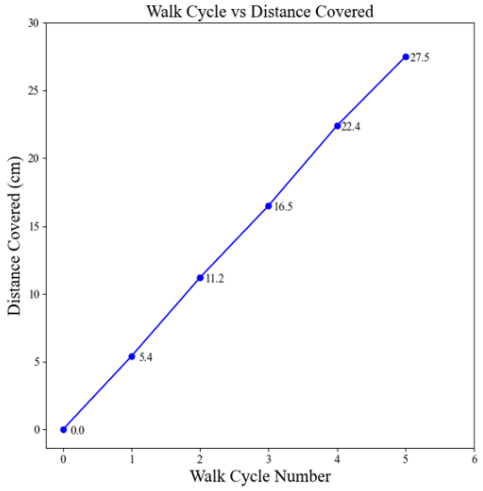


Figure 14: Walk cycle vs Distance covered

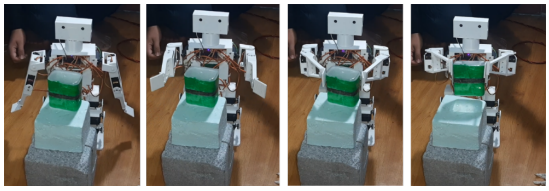


Figure 15: Pick and drop motion

estimation accuracy decreased at larger ranges due to reduced pixel disparity, the approach proved effective within practical operating distances for manipulation tasks. Motion planning and execution were supported through simulation-based kinematic modeling, which provided a structured framework for generating walking, rotation, sideways, and pick-and-place motions. Although direct deployment of simulated motion data to physical hardware was limited by servo torque constraints and actuator variability, iterative refinement enabled stable real-world motion execution. The results highlight both the feasibility and the limitations of servo-driven humanoid platforms built using affordable components.

Overall, the proposed system demonstrates that functional humanoid behavior integrating perception, locomotion, and manipulation can be achieved at low cost, making it suitable for educational and experimental robotics research. The work establishes a practical baseline for future enhancements in adaptive control and sensing.

8. Future work

The current system executes motion using predefined frame-based trajectories, which limits adaptability to surface conditions and environmental variations. Future work will focus on implementing dynamic balance control by incorporating force-sensing resistor (FSR) sensors for real-time Zero Moment Point (ZMP) estimation and closed-loop gait adjustment. Reinforcement learning–based control strategies may further enhance adaptability across varying terrains.

Perception capabilities can be extended beyond cube-specific detection by training the YOLO model on multi-object datasets or fine-tuning pre-trained models through transfer learning. The modular perception pipeline allows model replacement without hardware modification. Additionally, integrating oriented bounding box (OBB) detection would enable object orientation estimation, improving grasp planning and selective manipulation. Higher-resolution camera modules may further improve stereo depth accuracy at longer distances. These enhancements would significantly improve the robot's autonomy, robustness, and applicability in more complex and dynamic environments.

Acknowledgements

The authors would like to express their sincere gratitude to the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, Tribhuvan University, for providing academic supervision, laboratory access, and institutional support throughout the course of this research.

This work did not receive any external funding or specific grant, and no award number is associated with this study. All resources utilized were supported internally by the department.

Declaration of competing interest

The authors declare that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Kajita S, Hirukawa H, Harada K, et al. Introduction to humanoid robotics: volume 101[M/OL]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. DOI: 10.1007/978-3-642-54536-8.

[2] Jewell C. Bringing ai to life[EB/OL]. 2018. https://www.wipo.int/web/wipo-magazine/articles/bringing-ai-to-life-40507.

[3] Atmeh G M, Ranatunga I, Popa D O, et al. Implementation of an adaptive, model free, learning controller on the atlas robot[C/OL]// 2014 American Control Conference. IEEE, 2014: 2887-2892. DOI: 10.1109/ACC.2014.6859431.

[4] Ficht G, Behnke S. Bipedal humanoid hardware design: a technology review[J/OL]. Current Robotics Reports, 2021, 2: 201-210. DOI: 10.1007/s43154-021-00050-9.

[5] Atkeson C G, Hale J G, Pollick F, et al. Using humanoid robots to study human behavior[J/OL]. IEEE Intelligent Systems, 2000, 15: 46-56. DOI: 10.1109/5254.867912.
[6] Duan Y, Chen X, Houthooft R, et al. Benchmarking deep reinforcement learning for continuous control[Z]. 2016.
[7] Guizzo E, Ackerman E. The hard lessons of darpa's robotics challenge[J/OL]. IEEE Spectrum, 2015, 52: 11-13. DOI: 10.1109/MSPEC.2015.7164385.
[8] Kotha S S, Akter N, Abhi S H, et al. Next generation legged robot locomotion: A review on control techniques[J/OL]. Heliyon, 2024, 10: e37237. DOI: 10.1016/j.heliyon.2024.e37237.
[9] He L, Zhou Y, Liu L, et al. Research on object detection and recognition in remote sensing images based on yolov11[J/OL]. Scientific Reports, 2025, 15: 14032. DOI: 10.1038/s41598-025-96314-x.
[10] Jocher G, Qiu J. Ultralytics yolo11[Z]. 2024.
[11] Li Z, Xu B, Wu D, et al. A yolo-ggcnn based grasping framework for mobile robots in unknown environments[J/OL]. Expert Systems with Applications, 2023, 225: 119993. DOI: 10.1016/j.eswa.2023.119993.
[12] Kumar R, Lal S, Kumar S, et al. Object detection and recognition for a pick and place robot[C/OL]// Asia-Pacific World Congress on Computer Science and Engineering. IEEE, 2014: 1-7. DOI: 10.1109/APWCCSE.2014.7053853.
[13] Dandil E, Cevik K K. Computer vision based distance measurement system using stereo camera view[C/OL]// 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2019: 1-4. DOI: 10.1109/ISMSIT.2019.8932817.
[14] Abiyev R H, Arslan M, Gunsel I, et al. Robot pathfinding using vision based obstacle detection[C/OL]// 2017 3rd IEEE International Conference on Cybernetics (CYBCONF). IEEE, 2017: 1-6. DOI: 10.1109/CYBConf.2017.7985805.
[15] Shamsuddin S, Ismail L I, Yussof H, et al. Humanoid robot nao: Review of control and motion exploration[C/OL]// 2011 IEEE International Conference on Control System, Computing and Engineering. IEEE, 2011: 511-516. DOI: 10.1109/ICCSCE.2011.6190579.
[16] Rahul M, Thavai R, Kumar M S, et al. Inverse kinematics solution for biped robot[J/OL]. IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE), n.d., 12: 57-62. DOI: 10.9790/1684-12145762.
[17] Ortega-Palacios M A, Palomino-Merino A D, Reyes-Cortes F. Inverse kinematics model for a 18 degrees of freedom robot[J/OL]. Journal of Automation, Mobile Robotics and Intelligent Systems, 2023: 22-29. DOI: 10.14313/t4yf9254.
[18] Huang D, Fan W, Liu Y, et al. Design of a humanoid bipedal robot based on kinematics and dynamics analysis of human lower limbs[C/OL]// 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, 2020: 759-764. DOI: 10.1109/AIM43001.2020.9158973.
[19] Pati C, Kala R. Vision-based robot following using pid control[J/OL]. Technologies (Basel), 2017, 5: 34. DOI: 10.3390/technologies5020034.
[20] Abdolshah S, Abdolshah M, Tang S H. Trajectory planning and walking pattern generation of humanoid robot motion[J/OL]. IAES International Journal of Robotics and Automation (IJRA), 2014, 4: 135-142. DOI: 10.11591/ijra.v4i2.pp135-142.
[21] Kumar A, Fu Z, Pathak D, et al. Rma: Rapid motor adaptation for legged robots[Z]. 2021.
[22] Ojha R R, Sharma R M, Bhattarai S P, et al. Drink serving robotic arm relying on multimodal inputs[J/OL]. European Journal of Applied Science, Engineering and Technology, 2025, 3: 261-276. DOI: 10.59324/ejaset.2025.3(2/.21.