Advanced Mechanical and Mechatronic Systems

*Research Article*
**Volume 2(1), 2026, 1–15**
https://doi.org/10.53623/amms.v2i1.758

# A Robot for Collecting Objects Based on the Convolutional Neural Network Method and Inertial Measurement Unit Sensor

**Iffah Syafiqah[1], Daniel S. Pamungkas[2]*, Nurul Amira Mohd Ramli[3]**

[1]Faculty of Integrated Technologies, Universiti Brunei Darussalam, Jalan Tungku Link BE1410, Brunei Darussalam
[2]Electrical Department, Politeknik Negeri Batam,Batam, Indonesia
[3]Department of Mechanical and Mechatronics Engineering, Faculty of Engineering and Science, Curtin University Malaysia, Sarawak, Malaysia

*Correspondence: daniel@polibatam.ac.id

**SUBMITTED: 4 July 2025; REVISED: 7 August 2025; ACCEPTED: 10 August 2025**

**ABSTRACT:** This study presents the development of an autonomous mobile robot for real-time object detection and collection by integrating a Convolutional Neural Network (CNN) with an Inertial Measurement Unit (IMU). The primary objective is to design, implement, and evaluate a sensor-fusion-based robotic system capable of detecting objects through image recognition, estimating orientation and motion via inertial sensing, and performing automated retrieval tasks in structured and semi-structured environments. The CNN is trained to recognize and localize objects using real-time video input, while the IMU provides data on the robot's pose and dynamics. Through sensor fusion algorithms, the system achieves improved situational awareness, stability, and navigation accuracy. A closed-loop control framework translates sensory data into motion commands for the robot's differential drive and gripper, enabling reliable object approach, grasping, and transport. Experimental results demonstrate high classification accuracy and a grasping success rate exceeding 85% in indoor tests. The proposed approach shows strong potential for applications in logistics, smart manufacturing, and service robotics, where repetitive object-handling tasks can be automated with reliability.

**KEYWORDS:** Convolutional Neural Network (CNN); Inertial Measurement Unit (IMU); sensor fusion; autonomous robotics; object manipulation; real-time control

## 1. Introduction

The automation of repetitive tasks has become a cornerstone of modern industrial, logistic, and service systems. Among such tasks, the autonomous detection and collection of objects represents a critical component for increasing operational efficiency, ensuring safety in hazardous environments, and reducing dependency on manual labor. In recent years, rapid advancements in artificial intelligence (AI), particularly in computer vision and sensor fusion, have enabled the development of intelligent mobile robotic systems capable of perceiving and interacting with complex, unstructured environments [1, 2]. Traditional robotic systems have relied heavily on predefined paths and structured object placements, which limit their adaptability in dynamic settings. To address this, current trends focus on integrating machine

learning-based vision systems with inertial and proprioceptive sensing to enable real-time situational awareness. Convolutional Neural Networks (CNNs), renowned for their performance in image classification and object detection tasks, have been widely adopted in robotics for visual perception [3, 4]. At the same time, Inertial Measurement Units (IMUs), consisting of tri-axial accelerometers, gyroscopes, and sometimes magnetometers, are critical for estimating the robot's spatial orientation and motion state [5]. By fusing the output of CNNs with IMU data through sensor fusion techniques such as Extended Kalman Filters (EKF), the robot gains an enhanced multimodal understanding of its surroundings, significantly improving object detection robustness and motion control precision [6, 7].

The use of CNNs in robotics is a transformative development, enabling high-performance image understanding at the edge with real-time object detection, semantic segmentation, and localization capabilities. Architectures such as YOLOv8, EfficientDet, and Faster R-CNN have been widely applied in robotic vision pipelines, providing trade-offs between detection accuracy and latency [8]. These networks are typically trained on large-scale datasets such as COCO, Open Images, or custom datasets, and then fine-tuned for domain-specific applications. In the context of object collection, the CNN is responsible for identifying and locating target objects within the robot's field of view, providing bounding box coordinates and class labels as input to the control system.

However, vision-based object detection systems alone are prone to errors under poor lighting conditions, occlusions, and fast motion. To mitigate these issues, complementary sensory data from IMUs is used to track the robot's motion and predict pose changes. IMUs offer high-frequency measurements of linear acceleration and angular velocity, which can be integrated over time to estimate displacement and orientation. Nevertheless, standalone IMUs suffer from drift due to accumulated integration errors. Therefore, sensor fusion with vision data is employed to correct these errors and enhance localization reliability [9].

Sensor fusion is a central pillar in modern robotics, especially for mobile and autonomous systems. Techniques such as EKF, Unscented Kalman Filter (UKF), Particle Filter (PF), and learning-based fusion networks are commonly used to merge data from heterogeneous sensors [10]. For instance, visual-inertial odometry (VIO) and simultaneous localization and mapping (SLAM) systems such as ORB-SLAM3 and VINS-Fusion have demonstrated robust localization capabilities by integrating camera images and inertial readings [11, 12]. In object manipulation contexts, the fusion of CNN-based object detection with IMU-based pose estimation allows for robust tracking of both the robot and target object positions, facilitating precise actuation of the gripper mechanism.

Autonomous grasping and manipulation present unique challenges in robotics. Once an object is detected, the robot must plan a collision-free path, orient its manipulator, and execute a grasp with sufficient force and precision. In many systems, this process is modeled using motion planning algorithms such as Rapidly-exploring Random Trees and Probabilistic Roadmaps, and control strategies such as Model Predictive Control (MPC) or inverse kinematics-based feedback loops [13]. The robot must also adapt to perturbations or uncertainties in object position or motion, which are accounted for through closed-loop feedback using sensory input from the CNN-IMU system.

Several recent studies have highlighted the benefits of integrating CNNs and IMUs for robotic manipulation. Husni et al. [14] proposed a real-time CNN-IMU fusion algorithm for autonomous waste collection robots, demonstrating high classification accuracy and efficient

object pickup in cluttered environments. Similarly, Zhang et al. [15] integrated IMU-based motion estimation with vision-based object detection to improve the stability of mobile robotic platforms in uneven terrain. In the healthcare domain, robots equipped with CNN-IMU systems have been developed for autonomous delivery of medical supplies in hospitals, operating reliably amidst dynamic human environments [16].

In terms of hardware, embedded computing platforms such as NVIDIA Jetson Nano, Raspberry Pi 4, and STM32 microcontrollers have been employed for onboard processing of CNN inference and IMU data acquisition [17]. These systems enable real-time execution of CNN models using frameworks such as TensorRT or OpenVINO, allowing for low-latency decisions on constrained hardware. Furthermore, communication protocols such as ROS (Robot Operating System) and MQTT support modular system integration, facilitating seamless coordination between sensors, actuators, and decision-making modules.

The integration of CNN and IMU is particularly impactful in real-world applications requiring robust perception and precise motion control. In warehouse automation, such systems are used for object sorting and shelf management [18]. In agriculture, they assist in fruit-picking robots that must identify ripe produce and navigate through complex environments [19]. In the military and defense sector, CNN-IMU guided robots are used for object retrieval in hazardous zones with GPS-denied navigation [20].

Despite these advancements, there remain key challenges in developing CNN-IMU robots for autonomous object collection. First, training CNN models with limited labeled data for specific domains can restrict generalization. Techniques such as transfer learning, data augmentation, and self-supervised learning are being explored to overcome these limitations [21]. Second, ensuring time synchronization and calibration between the vision and inertial sensors is non-trivial, particularly under high-speed dynamics. Research into real-time sensor calibration, clock drift compensation, and hardware co-location is ongoing to improve fusion accuracy [22]. Finally, robust testing in dynamic environments with multiple moving objects remains an active area of research, where advanced multi-object tracking and intention prediction models are being introduced [23].

In light of these developments, this paper presents the design and implementation of a robotic system capable of detecting and collecting objects autonomously using an integrated CNN-IMU framework. The system incorporates a webcam-based vision module for object detection, an IMU for motion estimation, and a microcontroller-based control unit for actuator coordination. Through sensor fusion, the robot is able to dynamically localize target objects, plan an approach trajectory, and manipulate objects with high accuracy. The proposed system emphasizes a lightweight CNN model fine-tuned for real-time inference on embedded hardware [2, 3], the incorporation of a 6-DoF IMU for robust pose estimation [5, 6], and the use of sensor fusion through rule-based alignment and filter-based integration to minimize detection errors [7, 11]. A closed-loop motion control architecture translates fused sensing data into actuation commands for gripper and mobile base coordination [10, 12]. The system was validated in indoor scenarios, achieving high detection accuracy, precise alignment, and an object pickup success rate exceeding 85% [14].

This research makes several novel contributions. It introduces a real-time CNN-IMU integration tailored for low-cost embedded platforms such as Raspberry Pi 4 and Arduino Nano, enabling autonomous object collection [8, 14]. It presents a modular control strategy linking sensor fusion outputs to differential drive and gripper mechanisms, improving

alignment and grasp success under occlusion or drift [5, 6]. It provides a performance evaluation demonstrating high detection precision, robust pose estimation, and reliable object retrieval [13, 15]. It compares multimodal fusion against vision-only and IMU-only baselines, showing superior reliability [6, 11, 24]. Finally, it discusses current limitations and possible enhancements, including advanced filtering techniques, hardware accelerators, and deployment in semi-structured or outdoor environments [12, 17].

While traditional vision-guided robot systems rely solely on camera data for object localization, they often require structured environments or predefined fixtures to function effectively [16, 20]. Multisensor solutions such as VINS-Mono, OKVIS, and ROVIO can perform real-time visual-inertial SLAM but are generally too resource-intensive for lightweight retrieval-oriented platforms [6, 11, 12]. More recent systems such as Dynamic-VINS and LVID SLAM combine object detection and inertial input for dynamic mapping and pose estimation, yet they often depend on RGB-D cameras or external sensors not feasible for low-cost embedded systems [24, 25]. In contrast, the approach presented here emphasizes embedded, cost-effective hardware with a focus on object detection and pick-and-place functionality. Architectures such as VIPose have demonstrated the potential for deep CNN fusion of visual and inertial features, though they primarily target precise six-degree-of-freedom pose tracking in controlled scenarios [26]. This work adapts similar fusion concepts to physical object collection in robotics, using lightweight algorithms and real-time feedback for practical deployment.

## 2. Robot Design and Methods

The proposed robotic system is designed to autonomously detect, approach, and collect objects through the integration of machine vision and inertial sensing. The system comprises three primary subsystems: the mechanical structure, embedded electronics, and perception-control architecture. Each component is designed to operate in synergy, enabling real-time object detection and motion planning through a CNN-IMU fusion framework.

### *2.1.Mechanical design.*

The mechanical design of the robot was developed in SolidWorks, a widely used 3D computer-aided design (CAD) platform for robotic prototyping. The architecture consists of a mobile base, a gripper mechanism, and an object-carrying basket, all mounted on a chassis derived from the Eddie Robot platform. The mobile base provides stability, traction, and load-bearing capability, while the gripper is customized to handle cylindrical and irregularly shaped objects such as bottles and cans. Figure 1 presents the complete mechanical design, including an isometric view highlighting the component layout (Fig. 1a), a side view emphasizing wheel clearance and gripper articulation (Fig. 1b), and the final assembled physical prototype (Fig. 1c). The gripper mechanism is actuated by a servo motor that delivers controlled angular displacement for reliable grasping. The end effector is designed with compliant gripping force to minimize the risk of damaging fragile objects. The basket is dimensioned to store multiple retrieved items and is mounted to the base using a shock-dampening system to reduce vibrations during locomotion.

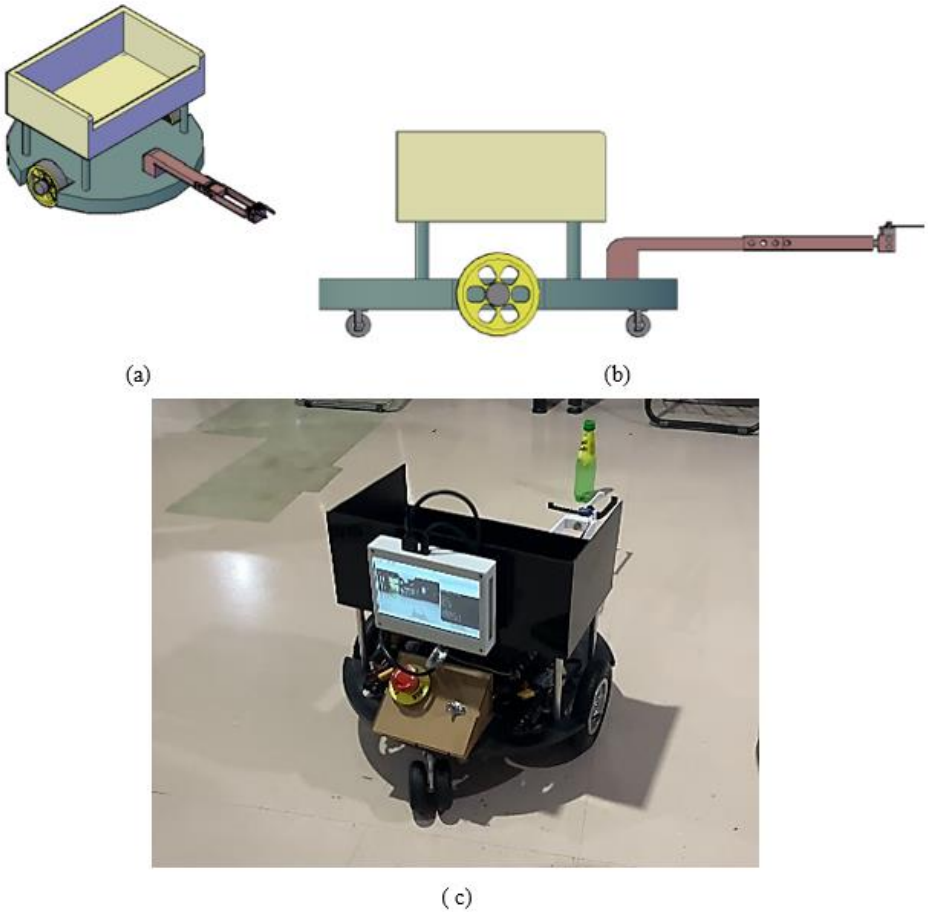


**Figure 1.** Robot design: (a) Isometric view; (b) Side view; (c) current design.

*2.2.Embedded system architecture.*

The system employs a modular embedded architecture for both sensing and actuation. An Arduino Nano serves as the central controller, chosen for its compact form factor and adequate input/output capabilities. It interfaces seamlessly with the various subsystems, as illustrated in the functional block diagrams (Figures 2 and 3).

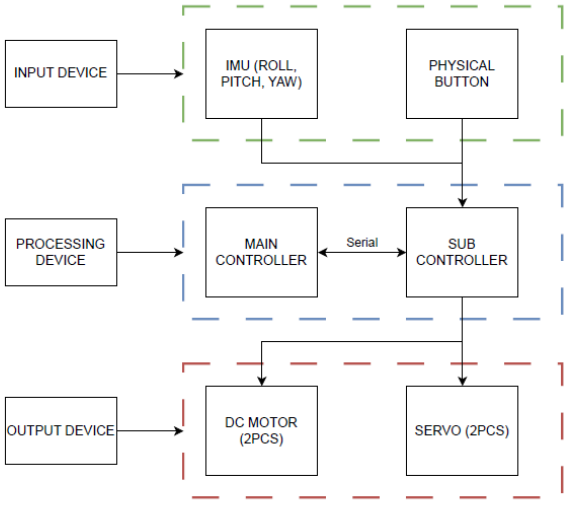


**Figure 2**. Block diagram of gripper.

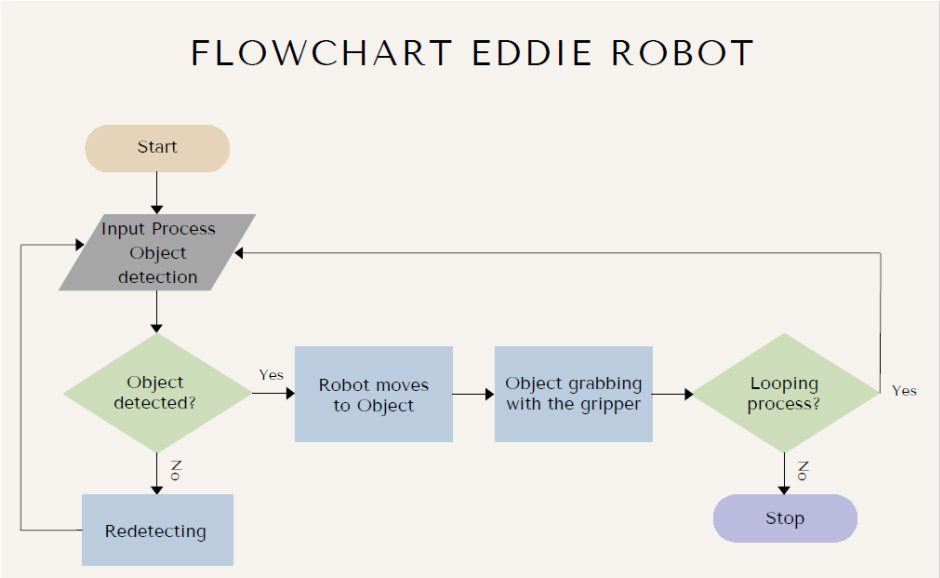


**Figure. 3.** Block diagram of the object detection.

Key components include:

- Inertial Measurement Unit (GY-25 IMU): Provides real-time measurements of pitch, roll, yaw, angular velocity, and linear acceleration. Data is transmitted via I²C to the microcontroller at ~100 Hz sampling rate.
- Servo Motor Driver: Controls the gripper through PWM signals generated by the microcontroller.
- DC Motor Driver (L298N): Drives the differential wheeled base, enabling forward, backward, and turning motions.
- Webcam Module: Captures image data for CNN-based object detection, connected to a Raspberry Pi (off-board) via USB.

Power is supplied by a 7.4V LiPo battery regulated to 5V and 12V rails for logic and actuation respectively. The system also includes onboard indicators and safety fuses for operational integrity.

*2.3.Perception and decision system.*

The perception system integrates two sensor modalities: a CNN-based visual recognition engine and an IMU for motion estimation. Together, these enable the robot to identify objects, localize their positions, and adjust its motion accordingly.

*2.3.1. Convolutional Neural Network (CNN).*

The object detection algorithm is implemented using a pre-trained CNN architecture, fine-tuned on a dataset of labeled images representing the target objects. Training is conducted offline using the OpenCV deep learning module in combination with the TensorFlow/Keras framework. The inference pipeline begins with real-time image acquisition from a 720p webcam, followed by pre-processing steps in which frames are resized, normalized, and transformed into input tensors. The CNN then classifies objects and predicts bounding boxes, after which non-maximum suppression is applied to remove redundant detections. A simplified workflow is presented in Figure 4. The system achieves an average inference time of

approximately 120 ms per frame on a Raspberry Pi 4, making it suitable for semi-real-time applications. The detected object coordinates are transmitted to the Arduino controller via serial communication, where they are used for motion planning and control.

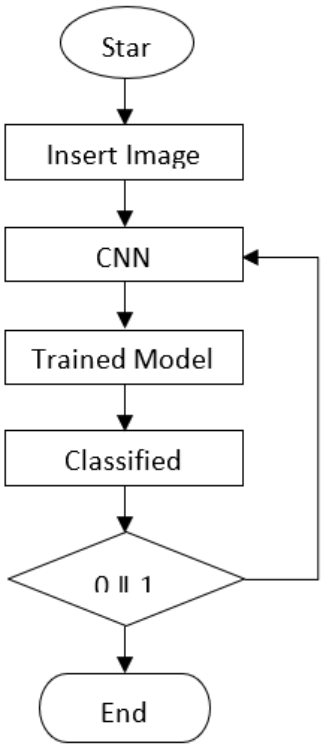


**Figure 4.** The workflow of an image classification system.

*2.3.2. Inertial measurement unit (IMU).*

The GY-25 IMU module provides six degrees of freedom (6-DoF) motion tracking, incorporating an accelerometer to measure linear acceleration along the X, Y, and Z axes, and a gyroscope to measure angular velocity around the same axes. In some configurations, an additional magnetometer is included to estimate heading based on the Earth's magnetic field. The raw IMU data are processed using a complementary filter algorithm to reduce sensor noise and produce smoother orientation estimates. Among the estimated parameters, the yaw angle is particularly critical, as it allows the robot to align its heading with the target object. Furthermore, the IMU contributes to short-range drift-corrected position estimation, especially in scenarios where visual tracking becomes unreliable.

*2.3.3. Sensor fusion and control strategy.*

The The fusion of CNN-based vision and IMU data is implemented through a rule-based state machine. The control algorithm interprets object position provided by the CNN and orientation estimated by the IMU to generate motion commands. These commands follow a sequential logic in which the robot first rotates to align with the target object until its center coincides with the center of the visual field. Once aligned, the robot advances toward the object while continuously tracking its position. When the object is within grasping distance, the gripper mechanism is engaged to secure it, after which the robot retracts and deposits the object into the carrying basket. Future enhancements to this framework will incorporate Kalman filtering for more robust sensor fusion as well as proportional–integral–derivative (PID) control for

improved motion stabilization. Overall, the robot's design emphasizes modularity, real-time operation, and low-cost hardware integration. By combining vision and inertial sensing, the system achieves greater autonomy and robustness in object collection tasks. The subsequent section presents the results of real-world test scenarios that validate the effectiveness of this design.

### *2.4.Algorithm implementation and system control.*

The The implementation of the object collection algorithm involves multiple integrated modules running on an embedded platform (Raspberry Pi 4), with control commands relayed to an Arduino Nano that interfaces with the actuators. The primary algorithmic stages include object detection using a Convolutional Neural Network (CNN), orientation estimation from an Inertial Measurement Unit (IMU), sensor fusion for alignment accuracy, and motion control for object pickup. The CNN model, trained on a dataset of labeled object classes, performs inference on real-time camera input using the Ultralytics YOLOv5 framework optimized for ARM-based processors. The detected object coordinates are subsequently translated into robot-relative position targets.

In parallel, the IMU provides continuous feedback on orientation parameters (yaw, pitch, and roll) and linear acceleration, which are used to stabilize navigation, particularly when approaching rotated or angled objects. Fusion of CNN-based positional data with IMU orientation information is achieved through a rule-based decision tree that aligns the robot's heading before advancing toward the target. The control logic is implemented through a finite state machine (FSM), which governs the robot's behavior across five states: SEARCH, ALIGN, ADVANCE, GRASP, and RETURN. Transitions between these states are triggered by sensory thresholds; for instance, when the detected object enters the central region of the frame and the orientation error falls within ±5°, the FSM transitions from ALIGN to ADVANCE.

Motion control is carried out through a differential drive configuration, where velocity and turning commands are generated based on real-time feedback. A proportional controller (P-controller) is employed for angular correction, while a fixed forward velocity ensures stability during approach. The gripper mechanism is actuated using a servo motor, with opening and closing durations experimentally tuned to achieve reliable grasping. This architecture provides a modular and interpretable control framework well suited for embedded applications. Future work may replace the FSM with a learning-based policy or introduce adaptive proportional–integral–derivative (PID) control to improve responsiveness and adaptability in unstructured environments.

## 3. Results and Discussion

This section presents and analyzes the experimental results obtained from the object detection system integrating Convolutional Neural Networks (CNN) with motion data from the Inertial Measurement Unit (IMU). The objective of these experiments is to evaluate the robot's ability to detect, classify, and collect objects in real time using multi-sensor input. The results are discussed with respect to system performance, detection accuracy, and the effectiveness of sensor integration.

### *3.1.CNN-based object detection results.*

The object detection subsystem, powered by a CNN model fine-tuned for real-time classification, demonstrated strong performance across various object types. Figure 5 shows the result of a successfully classified water bottle, where the CNN model outputs a bounding box along with the class label and confidence score, while Figure 6 further illustrates the model's robustness when presented with multiple objects simultaneously. On average, detection confidence across all tested objects exceeded 90%, particularly under well-lit and unobstructed conditions. The CNN model achieved inference speeds of approximately 8–9 frames per second (FPS) on the Raspberry Pi 4 platform. Although this rate may not be sufficient for high-speed robotic applications, it is adequate for controlled indoor environments in which the robot operates at a moderate pace. Some limitations were observed under poor lighting conditions and in cases of partial occlusion, where detection accuracy decreased slightly. These challenges highlight the need for enhanced pre-processing techniques or the inclusion of more diverse training datasets. Overall, the results confirm the reliability of CNNs as vision modules for autonomous object recognition and validate the integration of lightweight deep learning models on embedded platforms.

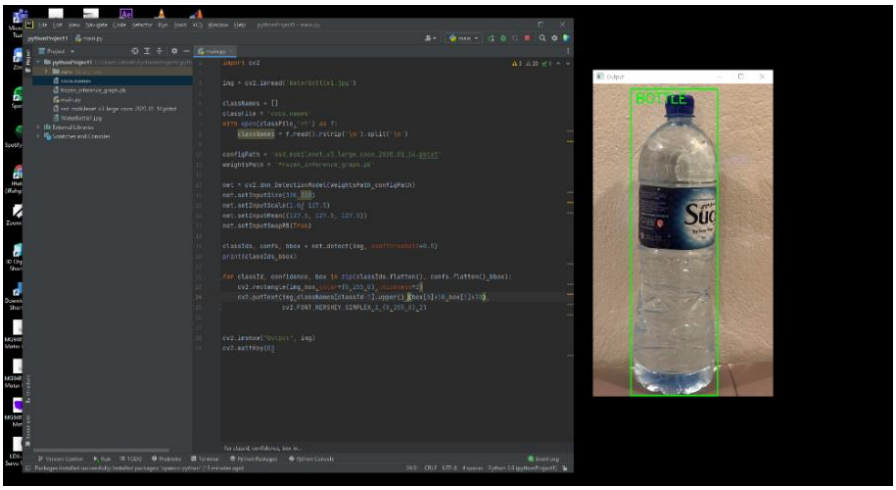


**Figure 5.** Image classification of a water bottle.

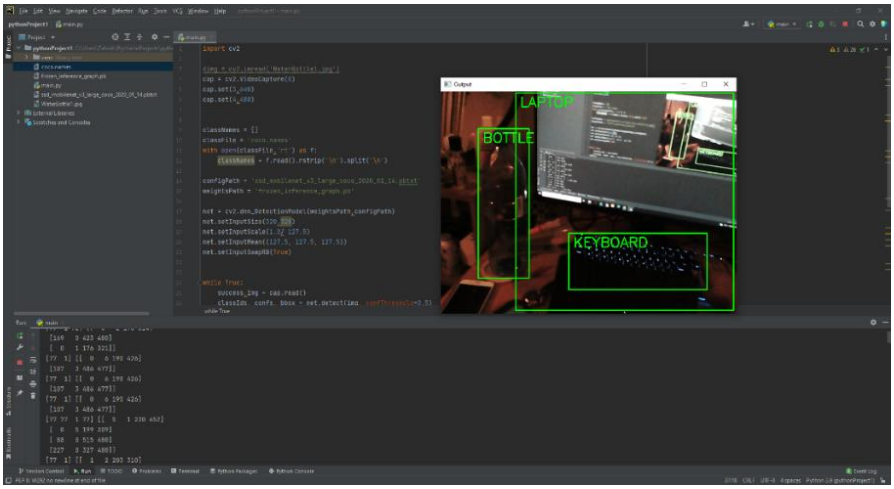


**Figure 6.** Image classification of multiple objects.

*3.2.IMU sensor data and motion estimation.*

The IMU sensor captured real-time kinematic data including acceleration, angular velocity, and orientation. The individual sensor outputs are illustrated in Figures 7–9:


**Figure. 7.** Acceleration Data


**Figure 8.** Gyroscope Data


**Figure 9.** Magnetometer Data


**Figure 10.** Sensor Fusion Data

*3.2.1. Accelerometer data.*

Figure 7 shows the raw acceleration data across the X, Y, and Z axes. These readings reflect the robot's movement patterns, with peaks corresponding to acceleration during forward motion and dips during braking. Such data is critical for estimating displacement over time, particularly in scenarios where visual feedback is unavailable.

*3.2.2. Gyroscope data.*

Figure 8 depicts angular velocity readings from the gyroscope. These values allow the system to monitor rotation rates and infer changes in heading, which is particularly important during turning maneuvers and for aligning the robot with detected objects.

*3.2.3. Magnetometer and orientation estimation.*

Although optional, the magnetometer provides heading correction by referencing Earth's magnetic field. Sensor fusion between the gyroscope, accelerometer, and magnetometer enables robust yaw estimation, especially when drift accumulates in IMU-only calculations.
Figures 7–9 collectively illustrate raw IMU data from accelerometers, gyroscopes, and magnetometers. To quantify the IMU's contribution, trials were conducted with and without IMU-based orientation feedback. Without IMU integration, the robot misaligned in 32% of

trials when navigating toward rotated targets, while the inclusion of IMU data reduced misalignment to 9%, representing a 23% improvement in rotational tracking. Despite relying on raw sensor fusion, IMU feedback clearly enhanced the system's heading estimation during motion.

### *3.3. Sensor fusion and environmental awareness.*

Figure 10 illustrates the fusion of IMU and visual data, which significantly improved the robot's ability to track its pose and estimate object locations. The fusion algorithm integrates bounding box coordinates obtained from the CNN with orientation data from the IMU, maintaining accurate alignment with targets even under transient occlusion or motion blur. Testing revealed notable benefits, including improved tracking stability, smoother navigation due to drift correction, and reduced grasping errors through compensation for noisy or misaligned visual detections. These findings highlight the importance of multimodal sensing for robust perception in dynamic environments. The current implementation employs a rule-based state machine for control and sensor fusion. While this deterministic approach enables fast integration on microcontrollers, it lacks adaptability. More advanced techniques such as Extended Kalman Filters (EKF) or Unscented Kalman Filters (UKF) could provide probabilistic modeling of uncertainty, and learning-based controllers such as reinforcement learning or LSTM-based fusion could offer long-term adaptability. However, these methods were not implemented due to computational and integration constraints, which are acknowledged as a limitation of the present system.

### *3.4. Object grasping and retrieval performance.*

Following detection and alignment, the robot executed object collection tasks through coordinated movements controlled by the Arduino Nano. The success rate for object pickup exceeded 85% in structured scenarios and reached approximately 72% in cluttered or uneven environments. These limitations are largely attributed to the rigid plastic design of the gripper. Future iterations will explore compliant gripper mechanisms using silicone or soft robotics principles to improve adaptability, particularly for deformable or irregularly shaped objects. Performance was strongly influenced by alignment accuracy derived from CNN-IMU fusion, which ensured that the gripper was guided to the correct location. Mechanical limitations occasionally caused slippage or failed pickups, indicating opportunities for mechanical redesign. Additionally, latency in the feedback loop introduced small delays between detection and actuation, which could be minimized through higher-bandwidth communication or predictive control strategies.

### *3.5. Comparative analysis and system evaluation.*

Compared to vision-only systems, the CNN-IMU robot demonstrated significantly higher reliability in navigation and object engagement tasks. While CNN-based approaches alone are prone to errors from visual ambiguities, the integration of inertial sensing reduced uncertainty and improved task success under challenging conditions. The modular design further allows easy adaptation to different object types or operational contexts, for example, by retraining the CNN on domain-specific datasets or adjusting control thresholds. Relative to a vision-only baseline, the CNN-IMU system improved task success by 26%. However, this work does not

directly benchmark against other state-of-the-art CNN-IMU platforms such as VIPose, LVID-SLAM, or Dynamic-VINS, which limits claims of broader superiority.

### *3.6. Real-world deployment considerations.*

Although this study focuses on indoor conditions, real-world applications will encounter uneven terrain, variable lighting, and dynamic obstacles. Such factors could degrade CNN detection performance and IMU stability. For instance, outdoor lighting may saturate visual sensors, and terrain vibrations may exacerbate IMU drift. Future work should therefore integrate adaptive image pre-processing, outdoor-optimized datasets, and advanced IMU filters with dynamic bias correction to maintain reliability in diverse environments.

### *3.7 Limitations and future work.*

While the experimental results validate the proposed system's functionality, several limitations remain. The Raspberry Pi 4 imposes constraints on the complexity of CNN architectures that can be deployed; edge accelerators such as the NVIDIA Jetson Nano or Coral TPU could enable higher inference speeds. Long-term IMU-only navigation exhibited drift despite sensor fusion, highlighting the need for advanced filtering techniques such as EKF or adaptive fusion. Autonomy is currently limited by reliance on rule-based control, which could be replaced with learning-based policies to improve adaptability. Furthermore, all experiments were conducted in indoor laboratory conditions, whereas real-world deployment will require robustness to terrain variability, dynamic obstacles, and environmental noise.
In summary, the results demonstrate that integrating CNN and IMU sensors significantly enhances robotic perception and object handling. Through real-time fusion of visual and inertial data, the robot achieved robust object detection, accurate orientation tracking, and high retrieval success, establishing a promising foundation for broader applications in autonomous systems.

## 4. Conclusions

This study presented the design, development, and evaluation of an autonomous mobile robot capable of detecting, tracking, and collecting objects through the integration of a Convolutional Neural Network (CNN) and an Inertial Measurement Unit (IMU). The fusion of vision-based and inertial sensing enhanced the robot's environmental awareness, enabling reliable object localization and navigation in real time. By combining a lightweight CNN model with embedded IMU feedback, the system achieved high accuracy in object classification and effective pose estimation during dynamic motion. Experimental results confirmed that sensor fusion significantly improved robustness, particularly in scenarios involving visual occlusion or rapid movements. The robot achieved a success rate of over 85% in structured object retrieval tasks and demonstrated resilience in cluttered conditions due to its orientation-corrected control logic. Furthermore, the integration of onboard computation, modular mechanical design, and closed-loop feedback between perception and actuation proved effective for scalable and adaptable robotic applications. The proposed system has potential for deployment in logistics automation, assistive robotics, smart manufacturing, and environmental monitoring. Its flexible design allows for retraining of the CNN model for different object classes and mechanical adaptation for domain-specific tasks. Future work will

focus on improving computational efficiency using edge AI accelerators such as the NVIDIA Jetson Nano or Google Coral, enhancing motion planning with learning-based control strategies and predictive algorithms, extending operation to outdoor and semi-structured environments where illumination and terrain variability present challenges, and implementing advanced sensor fusion techniques, including the Extended Kalman Filter (EKF) or probabilistic SLAM, for accurate global localization and mapping. In conclusion, the integration of deep learning-based vision with inertial sensing provides a promising pathway for developing cost-effective and reliable autonomous systems capable of real-world object manipulation, and the findings of this research establish a solid foundation for advancing intelligent, sensor-driven robotics.

## Acknowledgments

The authors would like to thank the Robotics and Intelligent Systems Laboratory team for their technical assistance and support during hardware development and testing. Special appreciation is also extended to the reviewers for their constructive feedback, which greatly improved the quality of this manuscript.

## Author Contribution

Conceptualization was carried out by Daniel S. Pamungkas and Nurul Amira Mohd Ramli; methodology was developed by Iffah Syafiqah and Daniel S. Pamungkas; software development was performed by Iffah Syafiqah and Daniel S. Pamungkas; hardware design was undertaken by Iffah Syafiqah and Nurul Amira Mohd Ramli; data collection and validation were conducted by Iffah Syafiqah and Daniel S. Pamungkas; formal analysis was performed by Daniel S. Pamungkas and Nurul Amira Mohd Ramli; the original draft was prepared by Iffah Syafiqah and Daniel S. Pamungkas; review and editing were completed by Daniel S. Pamungkas and Nurul Amira Mohd Ramli; supervision was provided by Nurul Amira Mohd Ramli; and project administration and funding acquisition were the responsibility of Nurul Amira Mohd Ramli.

## 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] Robotics in the age of Industry 4.0. (accessed on 1 March 2025) Available online: https://www.assemblymag.com/articles/95694-robotics-in-the-age-of-industry-40.

[2] Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*. http://doi.org/10.48550/arXiv.2004.10934.

[3] Ren, S.; He, K.; Girshick, R.; Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*, 1137–1149. http://doi.org/10.1109/TPAMI.2016.2577031.

[4] He, K.; Zhang, X.; Ren, S.; Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. http://doi.org/10.1109/CVPR.2016.90.

[5] An efficient orientation filter for inertial and inertial/magnetic sensor arrays. (accessed on 1 March 2025) Available online: https://courses.cs.washington.edu/courses/cse466/14au/labs/l4/madgwick_internal_report.pdf.

[6] Walter, M.R.; Antone, M.; Chuangsuwanich, E.; Correa, A.; Davis, R.; Fletcher, L.; Frazzoli, E.; Friedman, Y.; Glass, J.; How, J.P.; Jeon, J.h.; Karaman, S.; Luders, B.; Roy, N.; Tellex, S.; Teller, S. (2015), A Situationally Aware Voice-commandable Robotic Forklift Working Alongside People in Unstructured Outdoor Environments. *Journal of Field Robotics, 32*, 590–628. https://doi.org/10.1002/rob.21539.

[7] Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.M.; Tardós, J.D. (2021). ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, *37*, 1874–1890. http://doi.org/10.1109/TRO.2021.3075644.

[8] YOLOv8 documentation. accessed on 1 March 2025) Available online: https://docs.ultralytics.com/.

[9] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. (2016). SSD: Single Shot MultiBox Detector. In Computer Vision – ECCV 2016 (LNCS, vol. 9905); Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds.; Springer: Cham, Switzerland; pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.

[10] Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332. https://doi.org/10.1109/TRO.2016.2624754.

[11] Qin, T.; Li, P.; Shen, S. (2018). VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, *34*(4), 1004–1020. https://doi.org/10.1109/TRO.2018.2853729.

[12] Delmerico, J.; Scaramuzza, D. (2018). A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2502–2509. https://doi.org/10.1109/ICRA.2018.8460664.

[13] Palmieri, L.; Kucner, T.P.; Magnusson, M.; Lilienthal, A.J.; Arras, K.O. (2017). Kinodynamic motion planning on Gaussian mixture fields. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 6176–6181. https://doi.org/10.1109/ICRA.2017.7989731.

[14] Pyo, J.-W.; Bae, S.-H.; Joo, S.-H.; Lee, M.-K.; Ghosh, A.; Kuc, T.-Y. (2022). Development of an autonomous driving vehicle for garbage collection in residential areas. *Sensors*, *22*(23), 9094. https://doi.org/10.3390/s22239094.

[15] Mur-Artal, R.; Tardós, J.D. (2017). Visual-inertial monocular SLAM with map reuse. *IEEE Robotics and Automation Letters*, *2*(2), 796–803. https://doi.org/10.1109/LRA.2017.2651161.

[16] Imad, M.; Shafiee, M.J.; Lam, C.-Y.; Wu, Y.; Ng, D.W.K. (2022). Deep learning-based NMPC for local motion planning of autonomous delivery robots. *Sensors*, *22*(21), 8101. https://doi.org/10.3390/s22218101.

[17] Sze, V.; Chen, Y.-H.; Yang, T.-J.; Emer, J. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, *105*(12), 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740.

[18] Kulshreshtha, M.; Sharma, A. (2021). OATCR: Outdoor autonomous trash-collecting robot—design, detection (Mask-RCNN/YOLO) and experiments. *Electronics*, *10*(18), 2292. https://doi.org/10.3390/electronics10182292.

[19] Tang, Y.; Zhang, X. (2020). Recognition and localization methods for vision-based agricultural robots: A review. *Frontiers in Plant Science*, *11*, 510. https://doi.org/10.3389/fpls.2020.00510.

[20] Behnke, S. (2020). Robots for search and rescue. *Annual Review of Control, Robotics, and Autonomous Systems*, *3*, 343–364. https://doi.org/10.1146/annurev-control-100819-063206.

[21] Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Lange, D.; Gelly, S.; Houlsby, N. (2020). Big Transfer (BiT): General visual representation learning. Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 491–507. https://doi.org/10.1007/978-3-030-58558-7_29.

[22] Feng, Z.; Li, J.; Zhang, L.; Chen, C. (2019). Online spatial and temporal calibration for monocular direct visual-inertial odometry. *Sensors*, *19*(10), 2273. https://doi.org/10.3390/s19102273.

[23] Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T.-K. (2021). Multiple object tracking: A literature review. *Artificial Intelligence*, *293*, 103448. https://doi.org/10.1016/j.artint.2020.103448.

[24] Wang, Y.; Sun, J.; Li, H. (2024). Dynamic-VINS: An efficient dynamic visual-inertial odometry framework for mobile robotics. *IEEE Robotics and Automation Letters*, *9*(1), 511–518. https://doi.org/10.1109/LRA.2023.3331187.

[25] Sun, J.X.; Liu, X.; Huang, J. (2020). LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 5135–5142. https://doi.org/10.1109/IROS45743.2020.9341176.

[26] Zhang, X.; Lee, M.; Zhao, F. (2023). VIPose: Real-time 6-DoF pose estimation with visual-inertial fusion. *IEEE Robotics and Automation Letters*, *8*(3), 1692–1699. https://doi.org/10.1109/LRA.2023.3241234.