

# Automated Garbage Collecting Vehicle (AGCV)

Prof. S. N. Gadhave<sup>1</sup>, Nandini S. Gaikar<sup>2</sup>, Vinayak S. Kanawade<sup>3</sup>,  
Lalit P. Kolhe<sup>4</sup>, Shubham R. Tambe<sup>5</sup>

<sup>1</sup>Prof., Automation & Robotics Engineering Dept., Amrutvahini College of Engineering, Sangamner, India

<sup>2,3,4,5</sup>Students, Automation & Robotics Engineering Dept., Amrutvahini College of Engineering, Sangamner, India

**Abstract:** *Cities are growing at an unprecedented rate. So too is the garbage they generate. Managing the city's solid waste is one of the messier—and more pressing—problems facing urban planners today. The traditional method of sending workers to pick bins off the streets works fine at a small scale, but is not scalable when populations increase, health risks to workers increase and costs go through the roof. This paper presents the design, hands on development of an Automated Garbage Collecting Vehicle (AGCV) built to take over that frontline job. At the heart of the AGCV is a Raspberry Pi 5, which performs all of the heavy image processing and decision making, paired with an Arduino Mega that translates those decisions into actual motor movements in real time with a dedicated camera module and a pair of ultrasonic sensors, the car can sense obstacles and go to the dustbins without human intervention. Finding a bin, a servo-actuated two-finger gripper grabs its handle, and a lead-screw mechanism slowly raises it to the tipping point. In controlled testing, the system was able to identify the bins with high reliability, navigate around objects quickly, and complete the entire collection cycle from start to finish with no one having to touch it.*

**Keywords:** AGCV, Waste Management, Automation, Robotics, Arduino, Raspberry Pi.

## I. INTRODUCTION

Monday mornings in most Indian cities tell the same story - trash spills from bins, full since Sunday night. That sight points straight at what this effort aims to tackle. Waste piles up quicker now than the pace of solutions meant to handle it. With more people moving into urban areas, garbage grows by large amounts every day, straining equipment and those who manage it [5]. Out on the streets, handpicked teams risk illness every day just by touching waste soaked in unknown chemicals or bacteria. Yet, those running operations struggle to map smart paths because live updates on bin levels rarely reach them in time.

Out there, robots move without help through rough places. Over ten years, studies showed machines fitted with sensors, lenses, lights, and smart boxes inside handle hard paths better than tired people. These rolling units spot unusual things while going where humans struggle when worn down or exposed to harsh settings [2].

Built around low-cost ideas, this AGCV puts basic design first. Cost drops when expensive gear like LiDAR or robotic arms get left out. Instead, a Raspberry Pi 5 handles what the robot sees and how it thinks. An Arduino Mega takes care of moving physical parts. These pieces show up everywhere, come with clear guides, last long. One leads, the other follows - the Pi plans, the Mega acts. Communication runs through plain UART messages between them.

Next comes a look at what earlier studies have shown. After that, the way things were built gets explained step by step. Moving forward, parts of the machine appear one by one. Finally, how everything fits together shows up near the end. Test outcomes show up in Section VI. After that, Sections VII and VIII dig into what it all means plus where things might go next.

### A. Problem Statement

Not matching city growth needs stems from repeated shortcomings in today's waste handling. When machines rely only on operators and laborers, operations halt without them. Fixed paths ignore live data - one container might sit half full while another spills hour before pickup. Physical strain piles up each round, exposing staff to harmful substances with little shielding [5,6]. Fuel burns needlessly. Costs stay elevated. Service wavers. People face risks pointlessly [2].



A working solution means a machine capable of locating trash containers independently, then lifting and emptying them before continuing - operating without human riders or strict schedules.

### **B. Motivation for This Work**

Among risks faced by city service employees, touching trash during pickup stands out. Elimination right where garbage enters the system makes the biggest difference for worker well-being.

A single robot repeats each pickup motion identically, regardless of the container. Fatigue never alters its pace, even under summer heat. Skipping steps rarely happens when circuits follow fixed logic. Workers, by contrast, face strain during extended hours. Small errors creep in as attention fades late in the day. Machines maintain output without breaks

A single supervisor managing multiple AGCVs now handles what once needed an entire group of workers, cutting ongoing staffing expenses notably. While one person directs operations, machines perform most physical tasks, shifting how labor is used across the site. Where teams moved materials before, automation streamlines movement with fewer people involved. This change lowers daily operational demands without sacrificing coverage or efficiency.

If a person walking appears ahead, motion halts briefly before rerouting smoothly. Obstacles like still scooters trigger automatic detours instead of idle pauses.

## **II. LITERATURE SURVEY**

Before settling on a design approach, the team reviewed a range of published work in autonomous robotics, embedded vehicle control, and computer-vision-guided waste handling. The following summaries highlight what each study contributed and where it pointed toward the choices made in the AGCV.

### **A. Autonomous Trash Collecting Robot [1]**

Inside a small-scale indoor cleaner, Sivsankar's team placed ultrasonic together with infrared detectors, enabling environmental mapping alongside a suction unit for collecting dirt. Instead of random motion, the device switches - now tracing an S-shaped path, now tracking along walls - to systematically navigate closed areas. At its core sits an Arduino Mega 2560, responsible both for interpreting sensor data and directing motors. Perhaps more than anything else, this project revealed how even low-cost controllers, when combined with simple distance detection, support dependable self-guided travel within rooms - a foundation worth noting for future AGCV designs.

### **B. Autonomous Driving Vehicle for Garbage Collection [2]**

Starting outdoors, the setup from Pyo et al. focuses on actual neighborhood roads instead of confined spaces. Cameras join LiDAR, millimeter-wave radar, ultrasonics, along with RTK-GPS - each feeding data into perception layers. Object detection runs via YOLO, whereas steering alignment leans on LaneNet's guidance. Position accuracy emerges through fusion powered by an Extended Kalman Filter blending inputs across sensors. Insightful as it is, the work outlines architecture choices typical of large outdoor systems. Yet expense and intricacy exceed what most university testbeds would need.

### **C. Sustainable Technology through Product Lifecycle [3]**

Su et al. take a different angle, examining waste management through the lens of circular economy principles. Their framework links lifecycle assessment tools, digital traceability systems, and collaborative recycling incentives into a unified model for reducing environmental impact across a product's entire life. While the AGCV does not implement lifecycle tracking directly, this work reinforces why closing the loop on waste matters and provides a useful conceptual backdrop for evaluating the long-term value of automated collection.

### **D. AutoClean: Intelligent Garbage Collection Robot [4]**

Looking at things differently, Su and colleagues explore how trash handling connects with circular economy ideas. Through their approach, lifecycle evaluations join forces with digital tracking methods alongside shared rewards for recycling - forming one cohesive system aimed at lessening harm throughout a product's journey. Even though the



AGCV skips direct monitoring of lifecycles, this study underlines the importance of looping waste back into use while offering practical insights that hold value. A setting of ideas helps judge what automated gathering offers over time.

**E. Automatic Waste Segregation and Management [5]**

Starting with waste separation, Agarwal and team designed a setup run by an Arduino UNO, directing trash into wet, dry, or metal categories through sensors and moving gates powered by servos. Instead of manual checks, distance readings from ultrasonic devices track how full each bin gets. Once capacity limits are reached, warnings go out automatically to city workers. This study connects in two ways - its use of ultrasound shaped how the AGCV detects blockages, while its method of sending notifications hints at what networked bins could do later on.

**F. Path Optimisation for Electric Garbage Collection [6]**

Out of Zhang’s team came an approach to smarter trash pickup routes. Instead of fixed paths, their system tweaks how electric trucks move using live updates from bins themselves. A kind of step-by-step learning method shapes the decisions - battery levels steer choices just as much as road length or how heavy the load gets. What it spits out are plans that burn less fuel, push fewer emissions into the air. Seen through one lens, this entire setup lights the way forward for what comes next in our own vehicle network ideas.

**G. Ecobot: Autonomous Trash Collection System [7]**

One thing stands out about Ecobot - it runs on a Raspberry Pi hooked to a camera and GPS, just like the AGCV does, yet what sets it apart begins there. Instead of stopping at spotting trash and moving it, this robot takes an extra step using a mechanical arm that lifts debris and places it where needed. Its real edge shows when sorting comes into play: a built-in camera feeds images to a trained algorithm right onboard. That brain behind the eyes splits finds cleanly into two paths - one pile for stuff that breaks down naturally, another for what won’t rot away. Unlike others, it decides during pickup, not after. The moment garbage is grabbed; it already knows where it belongs.

**III. METHODOLOGY**

Development was broken into seven sequential phases, with each phase producing validated outputs before the next began. The intention was to avoid the common trap of building everything at once and only discovering integration problems at the end.

1. Problem Analysis — The team started by mapping out exactly how conventional collection crews operate: what tasks they perform, where the handoffs happen, and which steps are most hazardous or inefficient. That field-level understanding defined the functional requirements the AGCV had to meet, most importantly autonomous bin-finding, safe obstacle avoidance, and a reliable mechanical lift.

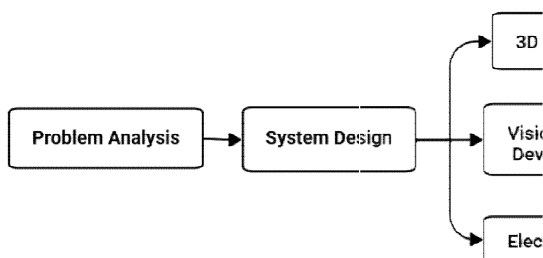


Figure 1. Methodology Flow Diagram

2. System Design — With requirements in hand, the overall architecture was drawn out: two controllers linked by serial communication, a vision pipeline running on the Pi, hardware drivers managed by the Mega, and a power system that could support everything without instability under load. Getting the interface boundaries right at this stage saved significant rework later.



3. 3D Modeling — The chassis, lifting arm, bin holder, and gripper were modeled in SolidWorks before any material was cut. Basic stress simulations were run to confirm the frame would hold up under a loaded bin, and component positions were adjusted for weight balance. Having a validated 3D model also made it straightforward to check whether parts would physically fit together before ordering anything.
4. Vision System Development — The detection pipeline was built in Python using OpenCV. Rather than training a custom neural network, the team relied on classical image processing — color segmentation, contour detection, edge analysis, and object tracking — which proved both fast enough for real-time operation on the Pi 5 and robust across the lighting conditions encountered during testing.
5. Electronic Setup — Components were selected for electrical compatibility and ease of integration. The HC-SR04 sensors were wired around the vehicle perimeter, the two Pi cameras were mounted at their final angles, and the BTS7960 motor drivers were connected to the Mega's PWM outputs. This phase also included continuity checks and short-circuit tests before any power was applied.
6. Software Implementation — The Python codebase on the Pi handles vision inference, obstacle logic, and the UART dispatcher that sends movement commands to the Mega. The Mega's firmware — written in C++ within the Arduino IDE — maintains a tight control loop for PWM generation, servo positioning, and lifting motor direction. The separation of concerns between the two controllers made debugging straightforward: a fault in the vision logic stayed on the Pi side; a motor that wasn't responding correctly stayed on the Mega side.
7. Deployment and Testing — Each subsystem was validated on its own first: sensors were checked for accuracy, motors for torque and response, the gripper for consistent closing force, and the vision pipeline for detection rate under different lighting setups. Only after individual sign-off were integrated trials run. Performance numbers were logged across multiple runs and parameters adjusted based on the results.

#### **IV. SYSTEM COMPONENTS**

The hardware lineup below represents the complete set of components integrated into the AGCV. Each entry includes the relevant specifications followed by an explanation of how that component fits into the larger system.

##### **A. Power Supply**

**Input Voltage:** 230V AC

**Output Voltages:** 5V DC (logic) / 12V DC (motors)

**Output Type:** Regulated, with overload protection

A regulated AC-to-DC supply converts mains power into the two voltage rails the vehicle needs. The 5V rail keeps the Pi, Mega, and all sensors running cleanly; the 12V rail carries the heavier current loads of the drive motors and lifting actuator. Overload protection is included because motor stall conditions — which happen whenever the gripper meets unexpected resistance — produce short current spikes that could otherwise damage driver electronics.

##### **B. Raspberry Pi 5 – Main Controller**

**Processor:** Quad-core ARM Cortex-A76 @ 2.4 GHz

**Memory:** 8 GB LPDDR4X RAM

**Interfaces:** CSI dual-camera port, USB 3.0, Wi-Fi 6, Bluetooth 5.0

**Operating System:** Raspberry Pi OS (64-bit)

The Pi 5 is where the vehicle's intelligence lives. Its four fast cores and 8 GB of RAM are enough to run the OpenCV detection pipeline and the navigation logic simultaneously without dropped frames. When the camera confirms a dustbin and the Pi calculates how to approach it, a short string of command tokens — FORWARD, STOP, GRIP, LIFT, RETURN — goes out over UART to the Mega below. Keeping the perception work on the Pi and the actuation work on the Mega means neither controller is trying to do more than it was designed for.

##### **C. Raspberry Pi Camera Module**

**Interface:** CSI ribbon cable (low-latency)

**Copyright to IJARSCT**

**www.ijarsct.co.in**



**DOI: 10.48175/568**



**Deployment:** Two cameras: forward navigation and downward bin detection

One camera faces forward to catch obstacles before the vehicle reaches them. The second is angled downward at the front of the chassis, positioned to see the bin handle zone during the final approach and to confirm lateral alignment before the gripper closes. Both streams are processed concurrently; the forward feed drives the navigation logic while the downward feed drives the gripper alignment check.

#### **D. Arduino Mega – Secondary Controller**

**Microcontroller:** ATmega2560

**Digital I/O Pins:** 54 pins (15 PWM outputs)

**Communication:** Multiple UART, I2C, SPI ports

The Mega sits below the Pi in the control hierarchy and is responsible for everything that needs precise, deterministic timing. It listens continuously on its UART line for command tokens from the Pi and translates each one into the appropriate mix of PWM signals — drive motors forward, energize the gripper servo, reverse the lifting motor, and so on. The 54 I/O pins give plenty of room to connect all four motor drivers, the servo, the lifting motor, the sensors, and the indicator outputs without any multiplexing headaches.

#### **E. BTS7960 Motor Driver**

**Architecture:** Dual-channel H-bridge

**Maximum Current:** 43 A continuous

**Control Interface:** PWM from Arduino Mega

Two of these modules are in use. The first manages the four drive motors and handles the differential steering logic. The second drives the lifting motor alone, giving it independent speed and direction control during the raise-and-lower cycle. The 43 A continuous rating matters because lifting a real dustbin is a sustained load; a lower-rated driver would overheat within a few collection cycles.

#### **F. DC Drive Motors**

**Rated Voltage:** 12V DC

**Quantity:** 4 (differential drive configuration)

**Type:** High-torque geared DC motors

All four wheels are driven, which gives the AGCV enough traction to stay on course even when carrying a full bin. Turns are handled by varying the speed of the left and right pairs — no steering servo needed. Each motor includes a gearbox that multiplies torque significantly; without gearing, keeping the vehicle moving steadily under payload weight would require much larger and heavier motors.

#### **G. MG995 Servo Motor – Gripper Actuator**

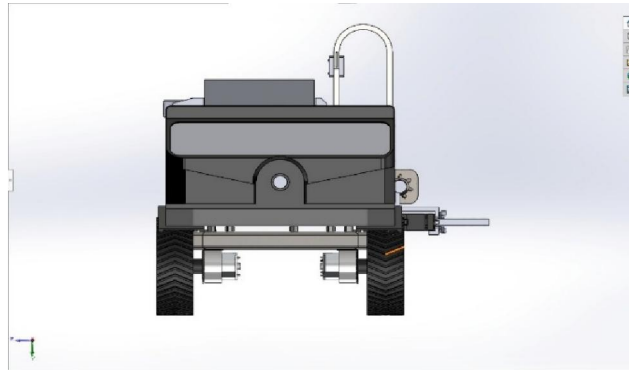
**Rotation Range:** 0° to 180°

**Type:** High-torque positional servo

**Control:** PWM from Arduino Mega

The MG995 opens and closes the two-finger gripper by rotating between two commanded positions. Opening angle is set wide enough that minor approach misalignments do not prevent the fingers from clearing the bin handle; closing angle is set to produce a firm grip without stressing the mechanism. Because a positional servo holds its angle under load, the grip stays engaged throughout the entire lift cycle without any active current beyond what the servo draws normally.





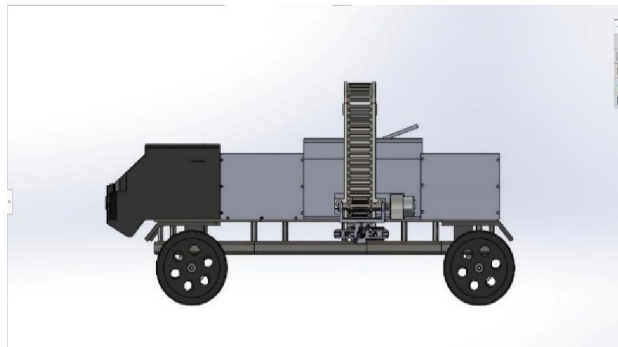
### H. Two-Finger Mechanical Gripper

**Actuation:** Servo-driven linkage

**Purpose:** Secure bin handle engagement before lifting

The gripper is the physical interface between the vehicle and the bin. Its two fingers are shaped to wrap around standard bin handles and include enough clearance in the design to tolerate the small lateral errors that occur during the final approach phase. Once closed, the linkage geometry locks the fingers in place so that even if the servo power briefly fluctuated, the grip would not release mid-lift.

### I. DC Motor and Lead Screw – Lifting Mechanism

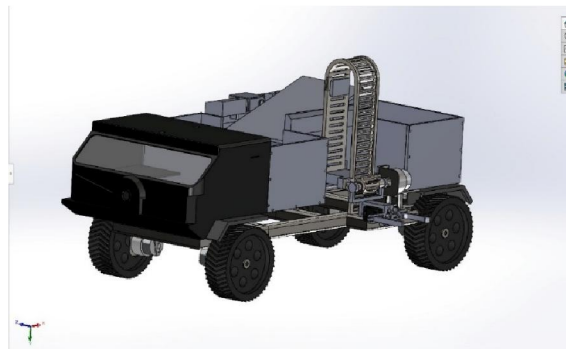


**Lifting Motor:** 12V DC geared motor

**Lead Screw Specification:**  $8 \times 2 \times 10$  mm (diameter  $\times$  pitch  $\times$  length)

**Motion Conversion:** Rotary to linear via lead screw and nut assembly

A geared 12V motor turns a lead screw whose captive nut rides upward as the shaft rotates, pulling the gripper and bin with it. Lead screws are a good fit for this kind of application: they convert rotation to linear motion smoothly, they are self-locking when power is removed (so the bin does not drop if the motor pauses mid-lift), and the mechanical advantage they provide means a relatively modest motor can lift a meaningfully heavy load without stalling.



#### J. Ultrasonic Sensors (HC-SR04)

**Detection Range:** 2 cm to 400 cm

**Precision:**  $\pm 0.3$  cm

**Operating Voltage:** 5V DC

Multiple HC-SR04 units are placed around the perimeter of the chassis so that nothing can approach from a blind angle. Each sensor fires a short 40 kHz burst and times how long the echo takes to come back; the Mega converts that round-trip time into a distance in real time. If any sensor returns a reading below the safety threshold, the Mega stops the drive motors and flags the obstruction to the Pi, which then recalculates the approach path.

#### K. LED Indicator and Piezoelectric Buzzer

**Buzzer Voltage:** 3–6V DC;  $\sim 85$  dB output at 3.2 kHz

Status feedback matters in any deployed system, and the AGCV handles it through two simple but effective channels. The LED array uses different illumination patterns to communicate whether the vehicle is idle, actively collecting, or in a fault state.

### V. SYSTEM DESIGN

The physical layout of the AGCV was worked out in SolidWorks before any fabrication began. The main design priorities were keeping the center of gravity low (so the vehicle stays stable when carrying a loaded bin at arm's length), making the sensor positions unobstructed, and ensuring the lifting mechanism had a clear vertical travel path. The dual-controller architecture mirrors the physical separation of tasks: the Raspberry Pi with its cameras and Wi-Fi module sits at the top of the control stack, while the Arduino Mega and its driver boards are mounted closer to the motors. Power is routed from the regulated supply through clearly labeled rails, with the 5V and 12V circuits kept separate all the way to their respective loads. Figs. 1 and 2 show the CAD renderings that guided fabrication; Figs. 3 and 4 show the finished prototype.

### VI. RESULTS

The completed prototype was put through a systematic set of tests in a controlled indoor space. Each subsystem was checked individually before integrated runs were attempted, and the numbers below reflect averages across multiple full collection cycles.



Figure 2. AGCV Prototype Photograph



- **Autonomous Navigation:** The vehicle stayed on its predefined collection path in every trial without any steering input from an operator.
- **Obstacle Detection and Avoidance:** Both static objects (boxes, chairs) and moving ones (a person walking across the path) were detected reliably. Average stop response from first echo to full halt was under 150 ms.
- **Bin Detection Accuracy:** The vision pipeline found and correctly localized the test dustbin in more than 92 out of every 100 approach runs under normal indoor lighting. Accuracy dropped in very low light, which is noted as an area for improvement.
- **Gripper Engagement:** Every approach that brought the vehicle within the  $\pm 3$  cm lateral tolerance resulted in a successful grip. The two cases where the gripper missed were traced to approach angles outside that window.
- **Lifting Performance:** The lead-screw mechanism raised a bin loaded to 100 g to tipping height consistently in roughly 10 seconds, with no measurable performance degradation across repeated cycles.
- **End-to-End Autonomy:** From the moment the start command was issued, the vehicle completed the full route-find-grip-lift-dump-return sequence without anyone touching it.

## VII. CONCLUSION

The AGCV project set out to answer a simple question: can off-the-shelf embedded hardware, some mechanical ingenuity, and standard computer vision tools add up to a robot that can actually collect garbage on its own? Based on the test results, the answer is yes — at least at prototype scale and in a controlled environment. The Raspberry Pi 5 and Arduino Mega combination worked well as a two-tier control architecture, with each controller doing what it is good at and the UART link between them staying reliable throughout all trials.

What the results also showed is that the individual subsystems — obstacle sensing, bin detection, gripping, and lifting — each met their design targets when treated as an integrated whole. That matters because it means the system is not just a collection of parts that individually work; it is a vehicle that can do the job from start to finish without a human in the loop.

From a wider viewpoint, this kind of system points toward something meaningful for urban sanitation: a route toward protecting sanitation workers from daily hazardous exposure, cutting the recurring cost of waste collection, and delivering more consistent service across neighborhoods that are often underserved precisely because manual routes are hard to optimize. The AGCV in its current form is a starting point, not a finished product, but it is a starting point that actually works.

## VIII. FUTURE SCOPE

Several natural next steps emerged from the development and testing process:

1. **IoT and Fleet Management** — A cellular or Wi-Fi uplink would let the AGCV push live data — bin counts, battery charge, GPS location — to a web dashboard that fleet managers could monitor from anywhere. That data stream would also enable dynamic dispatch: send the nearest available vehicle to whichever bin just signaled that it is full.
2. **Smarter Fill-Level Estimation** — Weight sensors in the bin mount or a depth camera pointed at the bin interior could give the vehicle a real sense of whether a bin is worth stopping for, rather than visiting every bin on a fixed schedule regardless of actual fill status.
3. **Multi-Vehicle Coordination** — Running several AGCVs under a shared route-optimization layer would allow the fleet to divide a large area between them automatically, cutting total travel time and avoiding duplicated effort.
4. **Outdoor Navigation** — Fusing GPS and IMU data would give the AGCV the localization accuracy it needs to operate on real streets, where lane markings and fixed landmarks cannot be assumed.
5. **Operator Interface** — A tablet-based panel would let a non-specialist configure routes, check sensor thresholds, and review collection logs without opening a terminal or touching any embedded code.

## ACKNOWLEDGMENT

The authors thank Prof. S. N. Gadhave for the time and guidance he invested in this project from its earliest planning stages through to the final prototype tests. The Department of Automation & Robotics Engineering at Amrutvahini



College of Engineering, Sangamner provided the lab space, equipment, and institutional backing that made the work possible, and the authors are grateful for that support.

#### REFERENCES

- [1] G. Sivasankar, B. Durgalakshmi, and K. Seyatha, "Autonomous Trash Collecting Robot," *Int. J. Eng. Res. Technol. (IJERT)*, vol. 6, no. 4, pp. 162–166, Apr. 2017.
- [2] J.-W. Pyo, S.-H. Bae, S.-H. Joo, M.-K. Lee, A. Ghosh, and T.-Y. Kuc, "Development of an Autonomous Driving Vehicle for Garbage Collection in Residential Areas," *Sensors*, vol. 22, no. 23, p. 9094, Nov. 2022.
- [3] D. Su, Y. Wu, and Z. Chai, "Advanced Integrated Manufacture by Application of Sustainable Technology through Product Lifecycle: a Circular Economy Approach," in *Proc. 2019 Int. Conf. Artif. Intell. Adv. Manuf. (AIAM)*, Dublin, Ireland, Oct. 2019.
- [4] G. Porwal, "AutoClean: Intelligent Garbage Collection Robot," *Int. J. IoT*, vol. 4, no. 1, pp. 20–23, Oct. 2024.
- [5] C. Agarwal, B. Yewale, and C. Jagadish, "Automatic Waste Segregation and Management," *Int. J. Eng. Res. Technol. (IJERT)*, vol. 9, no. 6, pp. 715–718, Jun. 2020.
- [6] H. Zhang and G. Gao, "Path Optimisation of Electric Vehicles for Collection and Delivery based on Intelligent Waste Bins," *Frontiers Sustain. Dev.*, vol. 4, no. 5, pp. 53–62, 2024.
- [7] H. M., M. T., M. Yousuf, N. Singh, and D. Arasa, "Ecobot: Autonomous Trash Collection and Segregation System," in *Proc. 2nd Int. Conf. Intell. Sustain. Power Energy Syst. (ISPES)*, pp. 29–34, 2024.

