FPGA-Based Autonomous Rag Picking Robot for Smart Waste Management

V. Thanush Kumar Reddy[1], M. Yaswanth Kumar[2], Venkata Gopinadh Pallikala[3]

[1,2,3]Department of Electronics and Communication Engineering Vasireddy Venkatadri Institute of Technology, Guntur, India

***Abstract*—With the rapid increase in urban population, efficient waste man- agement has become a significant challenge. Traditional rag picking methods are manual, time-consuming, and expose workers to haz- ardous environments. To overcome these limitations, this paper presents the design and implementation of an FPGA-based autonomous rag- picking robot for smart waste management.**

The proposed system uses a camera to capture real-time images and detect plastic waste using an object detection method. The DE10- Nano FPGA board acts as the main controller, enabling fast process- ing and real-time decision making. Ultrasonic sensors are used for obstacle detection and navigation.

Based on the detected object position, the robot moves towards the waste and performs pick-and-place operation using a robotic arm controlled by servo motors. The system demonstrates reliable object detection, smooth navigation, and effective waste collection.

The proposed solution reduces human effort, improves safety, and provides an efficient approach for smart waste management applica- tions.

***Index Terms*—FPGA, Autonomous Robot, Waste Management, Object Detection, Robotic Arm, Ultrasonic Sensor**

I. INTRODUCTION

The exponential growth of urban populations has led to a substantial increase in solid waste generation across cities. Improper waste disposal and inefficient collection methods contribute to environmental degradation, groundwater contamination, and severe health risks [13], [16]. Municipal corporations often struggle with timely and hygienic waste collection due to limitations in infrastructure and labor-intensive practices.

Manual rag picking, still widely practiced in many regions, exposes work- ers to biohazards, sharp objects, toxic chemicals, and unsanitary conditions. These workers often operate without protective equipment, making them highly vulnerable to infections, injuries, and respiratory issues [16]. The un- safe and unsustainable nature of manual rag picking calls for a technological intervention that ensures both operational efficiency and worker safety.

The rapid evolution of automation, computer vision, and intelligent con- trol systems has paved the way for smart solutions in waste management. Robotics, when integrated with embedded systems and artificial intelligence, can offer efficient, contactless, and autonomous garbage collection [10], [11]. Furthermore, the use of cloud platforms enables real-time monitoring, ana- lytics, and large-scale coordination essential components of modern smart city infrastructure [3], [17].

This paper presents the development of an Autonomous Rag Pick- ing Bot, a robotic system designed to detect, collect, and segregate waste autonomously. It employs a webcam for real-time image acquisition, an SSD- based object detection model running on a DE10-Nano FPGA board [2], [4], and a Raspberry Pi-controlled robotic arm for physical waste collection [10]. The bot uses ultrasonic sensors and GPS for navigation, while real-time sta- tus updates, including bin capacity and location data, are transmitted to Microsoft Azure Cloud via an RFS module [3], [12].

The remainder of this paper is structured as follows: Section III discusses related work; Section IV presents the system architecture and hardware components; Section V explains the detection methodology and control logic; Section VI covers implementation details; Section VII outlines the results and observations; Section VIII highlights the system's sustainability impact; and Section IX concludes the paper with future directions.

II. RELATED WORK

In recent years, automation in waste management has gained significant at- tention, leading to the development of various robotic systems aimed at en- hancing efficiency and safety. Traditional waste collection robots often utilize microcontroller-based architectures, which, while cost-effective, may not of- fer the processing power required for advanced tasks such as real-time image processing and complex decision-making. Some systems incorporate basic sensors for navigation but lack sophisticated vision capabilities for precise waste identification and sorting.

The integration of Field-Programmable Gate Arrays (FPGAs) with Ar- tificial Intelligence (AI) has been explored to overcome the limitations of microcontroller-based systems. FPGA-based designs provide parallel pro- cessing capabilities, enabling real-time data processing essential for dynamic environments. However, many existing FPGA implementations do not fully leverage the potential of cloud connectivity, limiting real-time monitoring and data analytics.

Our approach distinguishes itself by combining cloud monitoring with real-time Single Shot Multibox Detector (SSD) object detection on an FPGA platform. This integration allows for efficient data processing and remote ac- cessibility, addressing the shortcomings of previous systems. The use of the DE10-Nano FPGA platform is particularly advantageous due to its dual-core ARM Cortex-A9 processor operating at 800 MHz and 110K Logic Elements (LEs). This configuration provides a balanced mix of processing power and reconfigurability, making it well-suited for handling complex tasks such as real-time image processing and adaptive control in waste management appli- cations.

Table 1 summarizes the key features of existing systems compared to our proposed solution, highlighting the advancements introduced by our design. The DE10-Nano platform's combination of a dual-core ARM processor and FPGA fabric offers a unique advantage for our application, enabling efficient processing and flexibility. This configuration allows for the implementation of complex algorithms and real-time data processing, which are essential for the dynamic nature of waste management tasks. By leveraging this platform, our system achieves a balance between performance and adaptability, setting it apart from existing solutions.

Feature	Existing Systems	Proposed Bot	
Architecture	Microcontrolle based	rFPGA (DE10-Nano) + Em- bedded ARM	
Object Detection	Manual/IR-based	SSD with Camera Vision	
Navigation	Basic Sensor Only	Ultrasonic GPS	+
Robotic Arm Control	Static Logic	2-DOF Feedback	with
Cloud Integration	Limited or None	Full Azure Connectivity	
Decision Speed	Moderate	Real-Time (FPGA-Accelerated)	

Table 1: Comparison of Existing Systems vs. Proposed Bot

III. SYSTEM DESIGN

The Autonomous Rag Picking Bot is built using a modular hardware-software architecture to perform real-time waste detection, collection, and segregation. It integrates FPGA-based control, machine learning, and cloud connectivity. Figure 1 illustrates the system's overall architecture.

3.1 Hardware Components

The system comprises the following key modules:

- DE10-Nano Board: Serves as the main controller, handling SSD- based object detection, decision logic, and GPIO control.

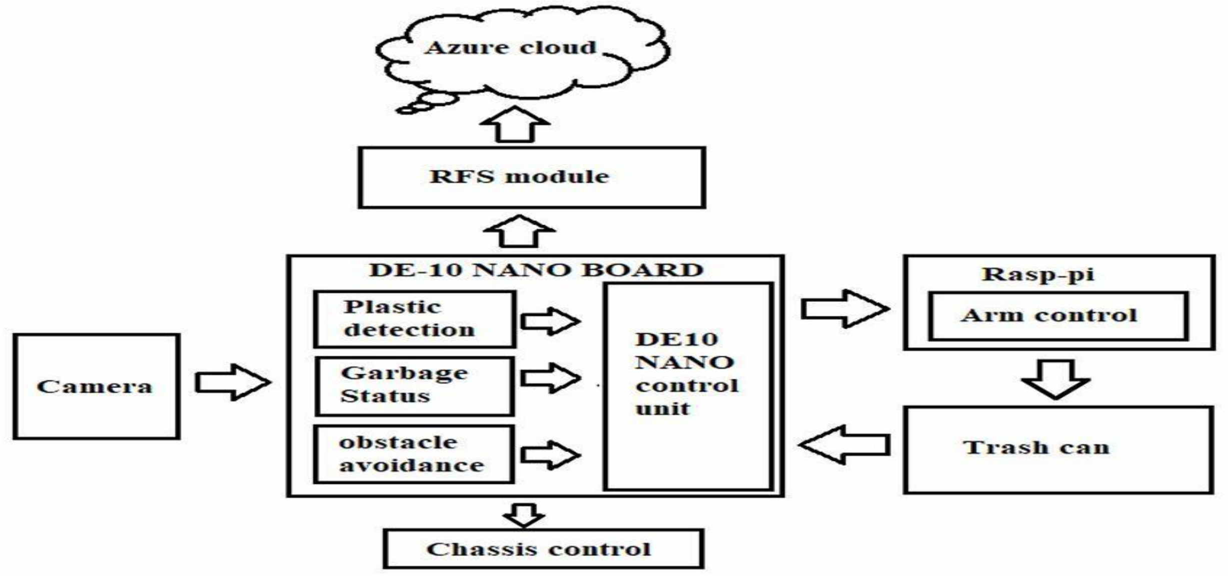

Figure 1: Block diagram of the Autonomous Rag Picking Bot.

- Arduino Microcontroller: Receives movement commands from the FPGA and controls the chassis motors via motor drivers.
- Raspberry Pi Board: Drives the 2-DOF robotic arm by generating PWM signals to control the MG995 servo motors.
- C270 Webcam: Captures real-time video at 640×480 resolution, used for plastic bottle detection.
- Ultrasonic Sensor: Measures distance between the bot and the object to prevent collisions and assist in navigation.
- GPS Module: Tracks the bot's location for cloud reporting and smart deployment.
- RFS Module: Transmits bin and location data to the Microsoft Azure cloud platform.
- Battery: Powers all components including motors, FPGA board, Pi, and sensors.

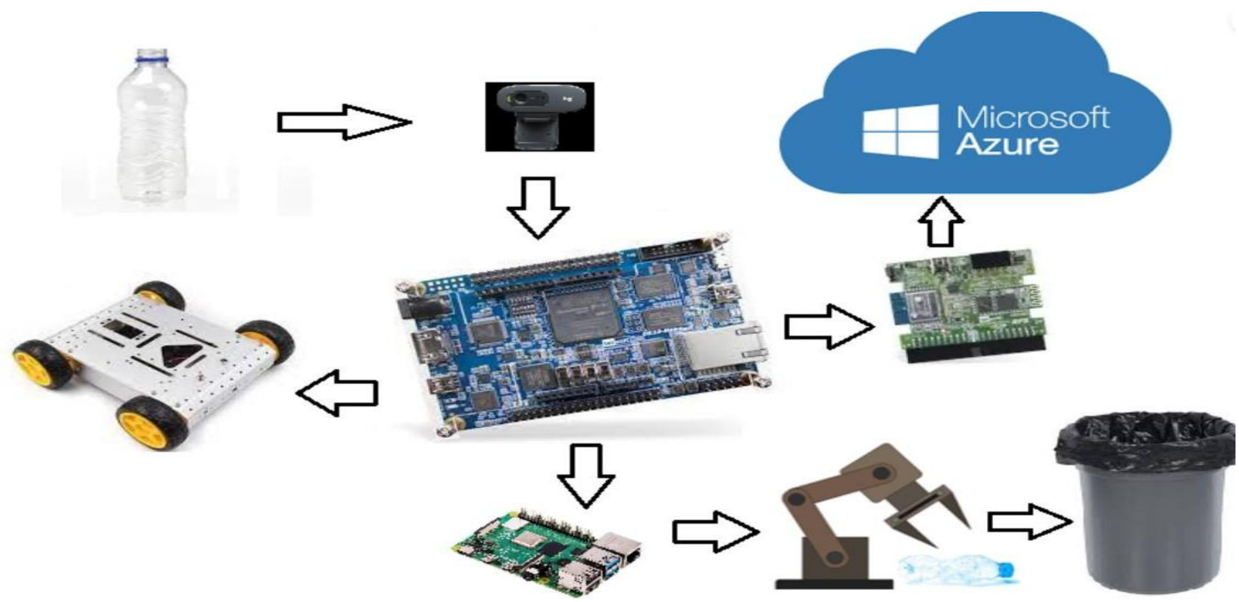

Figure 2: Physical overview of key hardware components in the bot (Camera, DE10-Nano, Raspberry Pi, GPS, Arm, Sensor, Battery, etc.).

3.2 Software Components

The software system includes real-time image processing, sensor feedback interpretation, and cloud integration:

- SSD Object Detection: Trained using the COCO dataset and im- plemented on the FPGA, the SSD model detects plastic bottles and outputs bounding boxes.
- Centroid Calculation and Motion Decision: The centroid of the bounding box is compared with a fixed region in the image. Based on deviation, the bot executes movement commands (left, right, forward, stop, etc.).
- Robotic Arm Control: A synchronized dual-servo arm mechanism is controlled by the Raspberry Pi to lift and dump the bottle.
- Cloud Data Upload: Bin capacity status and GPS coordinates are sent to Azure for real-time tracking, analytics, and alerts.

3.3 Control Logic Summary

The control logic is handled through GPIO pins of the DE10-Nano board, with the following mapping:

Table 2: GPIO Motor Control Logic

Direction	GPIO1803	GPIO1804	GPIO1805	GPIO1806
Forward	HIGH	LOW	HIGH	LOW
Backward	LOW	HIGH	LOW	HIGH
Left	LOW	HIGH	HIGH	LOW
Right	HIGH	LOW	LOW	HIGH
Stop	LOW	LOW	LOW	LOW

3.4 Workflow Overview

1. The camera captures frames and passes them to the SSD model for object detection.
2. Bounding boxes and centroid are calculated to determine the object's location.
3. Navigation commands are issued through GPIO based on deviation from the center.
4. When aligned, the bot stops and activates the robotic arm for collec- tion.
5. Bin status and GPS coordinates are sent to Azure via the RFS module.

3.5 Design Highlights

- Parallel Processing with FPGA: Enables real-time inference and fast control.
- Cloud Monitoring: Azure cloud integration ensures centralized con- trol and analytics.
- Robust Arm Mechanics: 2-DOF arm synchronizes left and right movement using MG995 servos.
- Modular Architecture: Allows future enhancements, such as multi- bot coordination.

3.6 Image Processing and Control Flow

The bot's core functionality is driven by a combination of computer vision, neural network inference, and embedded control logic. The entire detection pipeline is outlined below:

1. Image Frame: The process begins with capturing a live video stream from the C270 webcam connected to the DE10-Nano board. This raw image serves as the input for object detection. The resolution and clarity of this frame directly impact the bot's ability to identify waste accurately. It forms the foundation for all subsequent image processing tasks.
2. Image Resizing: The captured image is resized to a standard res- olution to optimize processing speed and reduce computational load. Resizing ensures compatibility with the SSD neural network while re- taining sufficient detail for reliable detection.
3. DNN Algorithm Using TensorFlow: A Deep Neural Network (DNN), implemented using the TensorFlow framework, processes the resized image to extract feature maps. The network uses pretrained layers to recognize features associated with plastic bottles, transform- ing raw pixel data into high-level object representations.
4. Single Shot Detector (SSD) Model: The SSD

model divides the image into a grid and evaluates each region for the presence of a plastic bottle. Confidence scores are assigned and bounding boxes are gener- ated around detected objects. The lightweight SSD architecture enables real-time inference on the FPGA with detection accuracy above 85%.

5. Bounding Boxes: Bounding boxes highlight the spatial location of detected objects in the frame. They are used to compute the centroids, which guide the bot's movement toward the object.
6. Centroid Calculation: The centroid (x_c, y_c) of the bounding box is computed as:

$$x_c = \frac{x_{\min} + x_{\max}}{2} \qquad y_c = \frac{y_{\min} + y_{\max}}{2}$$

This centroid is compared with the center of the frame to decide on the required direction of movement.

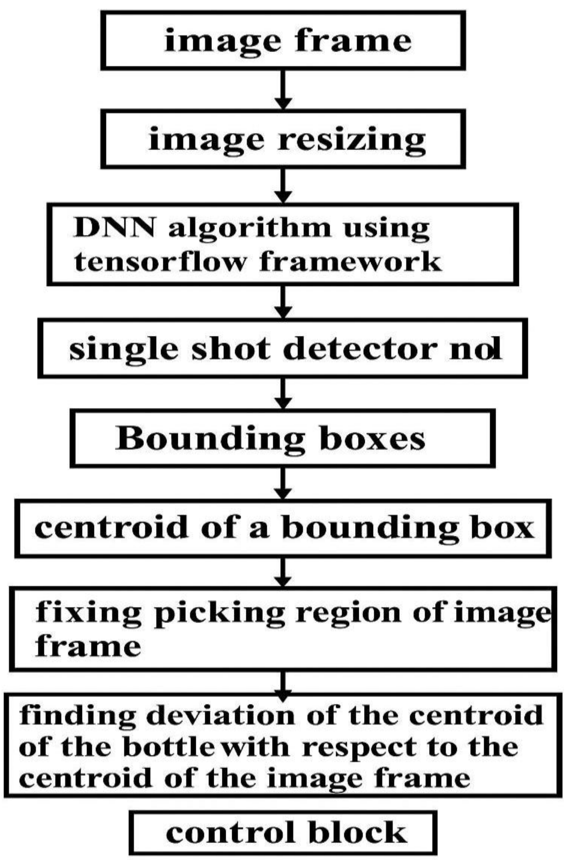

Figure 3: Flow diagram for image processing and control logic of the bot.

7. Picking Region: A fixed rectangular "picking zone" is defined at the center of the image frame. When the detected centroid lies inside this region, the bot halts and activates the arm for pickup. This region ensures the object is within reachable range for the arm.
8. Deviation Calculation: The Euclidean deviation between the cen- troid and the image center (x_0, y_0) is calculated using:

$$\text{Deviation} = \sqrt{(x_c - x_0)^2 + (y_c - y_0)^2}$$

where (x_0, y_0) is the fixed center of the image frame and (x_c, y_c) is the centroid of the detected object (bottle). This distance informs how far the bot needs to move in order to align with the object.

9. Control Block: Based on the deviation, the control logic generates GPIO signals for motor direction (forward, backward, left, right). It synchronizes chassis movement (via Arduino) and arm actuation (via Raspberry Pi). Once aligned, the bot proceeds to pick the object.

IV. IMPLEMENTATION

The implementation phase involves the practical realization of the system us- ing hardware-software integration. The bot operates through a three-layered structure: data gathering, decision-making, and actuation, ensuring real-time performance and autonomous operation.

4.1 Data Gathering and Object Detection

The C270 webcam captures real-time video input at 640×480 resolution. Each frame is resized and passed to a TensorFlow-based SSD model de- ployed on the DE10-Nano FPGA. The SSD divides the image into grids and assigns bounding boxes around detected plastic bottles with confidence scores exceeding 85%.

The centroid (x_c, y_c) of each bounding box is computed using:

$$x_c = \frac{x_{\min} + x_{\max}}{2} \qquad y_c = \frac{y_{\min} + y_{\max}}{2}$$

The centroid's location relative to the center of the image frame is used to generate navigation commands.

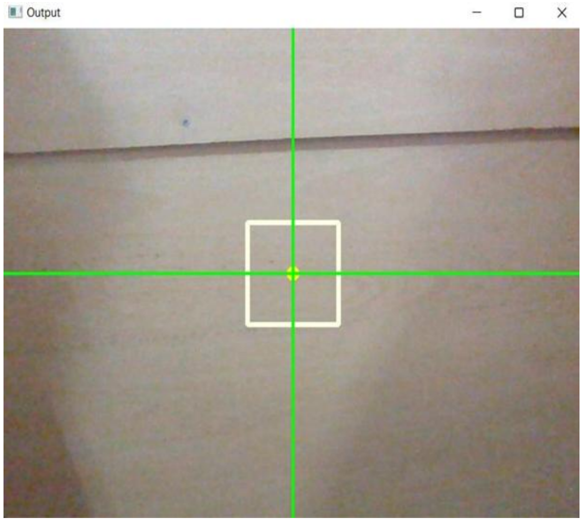

Figure 4: Bot movement logic using bounding box and centroid alignment for object tracking and navigation.

4.2 Image Processing and Object Detection

The core of the waste detection pipeline relies on image frames captured by the Logitech C270 webcam. This camera streams live video to the DE10- Nano board, where a pre-trained Single Shot Detector (SSD) model, built on the TensorFlow framework, processes the images to identify plastic waste.

The SSD model detects objects by dividing each frame into a grid and generating bounding boxes around detected items. The centroid of these bounding boxes guides the bot's navigation and alignment. Once the object is centered in the fixed "picking region," the bot triggers the pick-and-dump sequence using the servo-controlled robotic arm.

For enhanced object recognition, we experimented with the YOLOv3 model. YOLO (You Only Look Once) offers real-time detection and supports multiple object identification within a single frame. It is lightweight and suitable for embedded systems. Both SSD and YOLO were trained using the COCO (Common Objects in Context) dataset, which contains labeled data for over 80 object classes including bottles, bags, and waste items.

Key Advantages:

- SSD: Faster and more suitable for single-object detection in a con- strained field of view.
- YOLO: More robust when detecting multiple objects at once, although it requires slightly more compute.
- COCO Dataset: Provides standardized labels for object detection and image captioning, making it ideal for training both models.

This combination of vision models and training datasets enables high-accuracy detection and robust adaptability for smart urban environments.

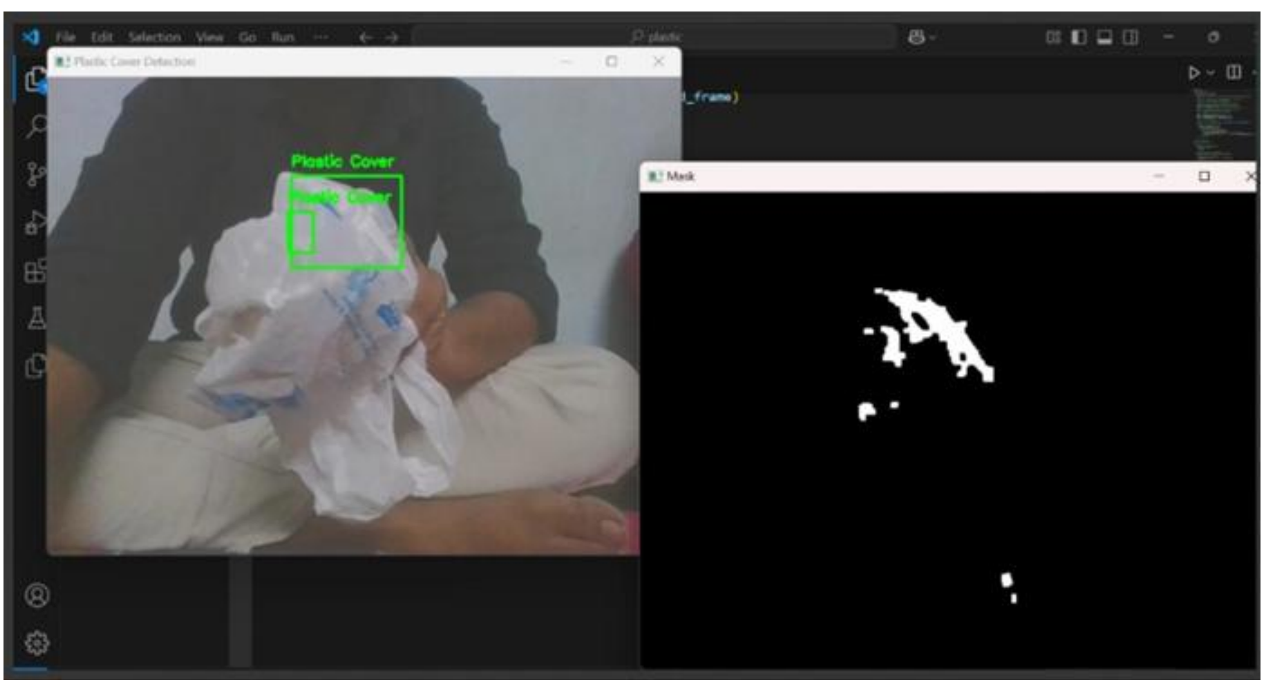

Figure 5: Object detection and image processing implementation.

4.3 Navigation and Control

The bot compares the centroid of the detected object with a predefined "pick- ing region" in the center of the frame. Based on its position, the DE10-Nanoboard generates GPIO signals that are interpreted by an Arduino microcon- troller, which in turn drives BTS7960 motor drivers to control the chassis.

Control decisions follow this logic:

- Centroid inside picking region: Execute STOP.
- Centroid above center: Move FORWARD.
- Centroid below center: Move BACKWARD.
- Centroid left of center: Turn LEFT.
- Centroid right of center: Turn RIGHT.
- No object detected: Begin rotational search by turning left. The GPIO logic for motor control is mapped in Table 2.

4.4 Arm Control for Waste Collection

A 2-DOF robotic arm, controlled by a Raspberry Pi and powered by four MG995 servo motors (two per arm), is responsible for picking and dumping detected waste. The arm is synchronized as follows:

- Right Arm:
  - Upper servo: 90° to 0° sweep to grip the bottle.
  - Lower servo: 90° to 15° downward to pick, then up to 90° to dump.

- Left Arm:
  - Upper servo: 0° to 90° sweep.
  - Lower servo: 90° to 15° downward to pick, then lift to dump.

Servo movement is coordinated using PWM signals from the Raspberry Pi, ensuring successful pick-and-place actions.

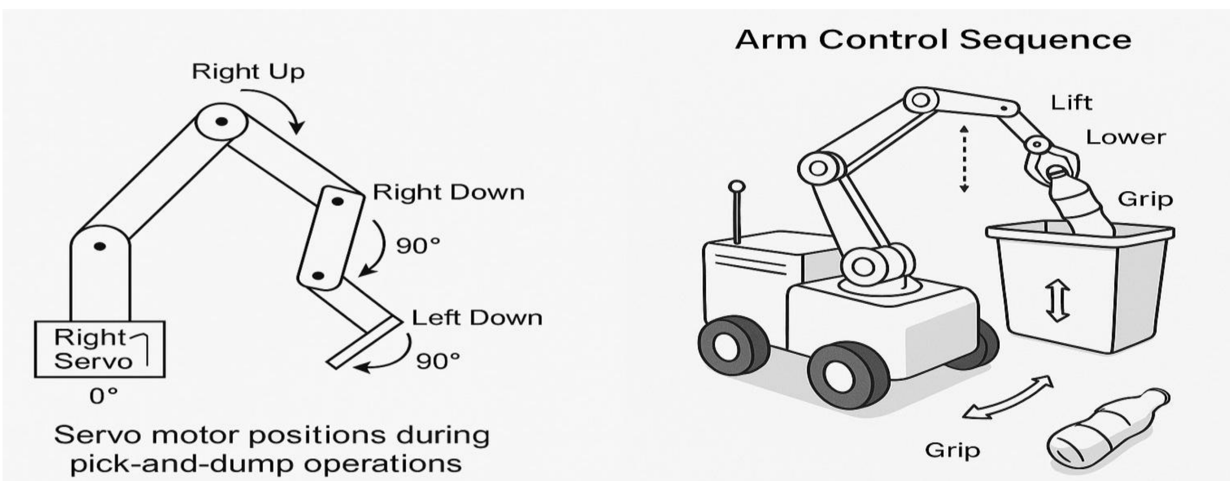

Figure 6: Robotic arm control sequence showing servo actuation during pick- and-dump operations.

4.5 Cloud Communication and Monitoring

Post collection, the bot sends bin capacity status and GPS coordinates to Microsoft Azure Cloud via the RFS module. The HPS (Hard Processor System) on the DE10-Nano handles communication, leveraging Azure's IoT services for:

- Real-time bot location tracking.
- Bin status monitoring (empty/full).
- Fault alerts and diagnostic logging.

The Azure dashboard provides centralized control and facilitates large- scale bot deployment in urban waste zones.

4.6 Workflow Summary

The end-to-end flow is described as:

1. Live video is captured and resized.
2. SSD model detects plastic waste and computes centroids.
3. DE10-Nano computes deviation and sends movement commands to Ar- duino.
4. Once aligned, the robotic arm picks and dumps the object.
5. Bin status and location are transmitted to the cloud.

This layered approach ensures modularity, efficiency, and future scalability.

V. RESULTS AND DISCUSSION

The Autonomous Rag Picking Bot was developed and tested in controlled environments with varying lighting and background conditions. The system demonstrated real-time object detection, precise navigation, and robust waste collection using SSD-based image processing deployed on the DE10-Nano FPGA.

5.1 Prototype Demonstration

The bot's integrated 2-DOF robotic arm successfully performed pick-and- place actions. The entire workflow from detection to collection was executed autonomously, with system feedback uploaded to the Azure Cloud platform.

5.2 Object Detection Output

The SSD model achieved a detection accuracy of over 85% on plastic objects (e.g., bottles, covers) and processed frames at approximately 17 FPS. The centroid-based navigation system provided accurate alignment for the robotic arm.

5.3 Performance Metrics

5.4 Sustainability Impact

- Reduced manual labor and exposure to hazardous environments.
- Live GPS tracking and bin status updates via Microsoft Azure.
- Clean, eco-friendly operation suitable for smart urban environments.
- Scalable and modular design allows swarm-based deployment of bots.

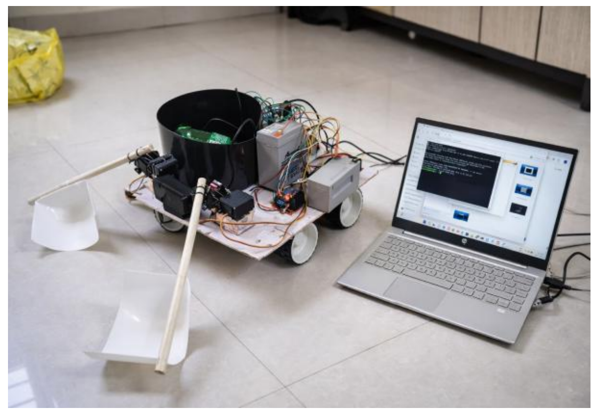

Figure 7: Prototype of the implemented Autonomous Rag Picking Bot.

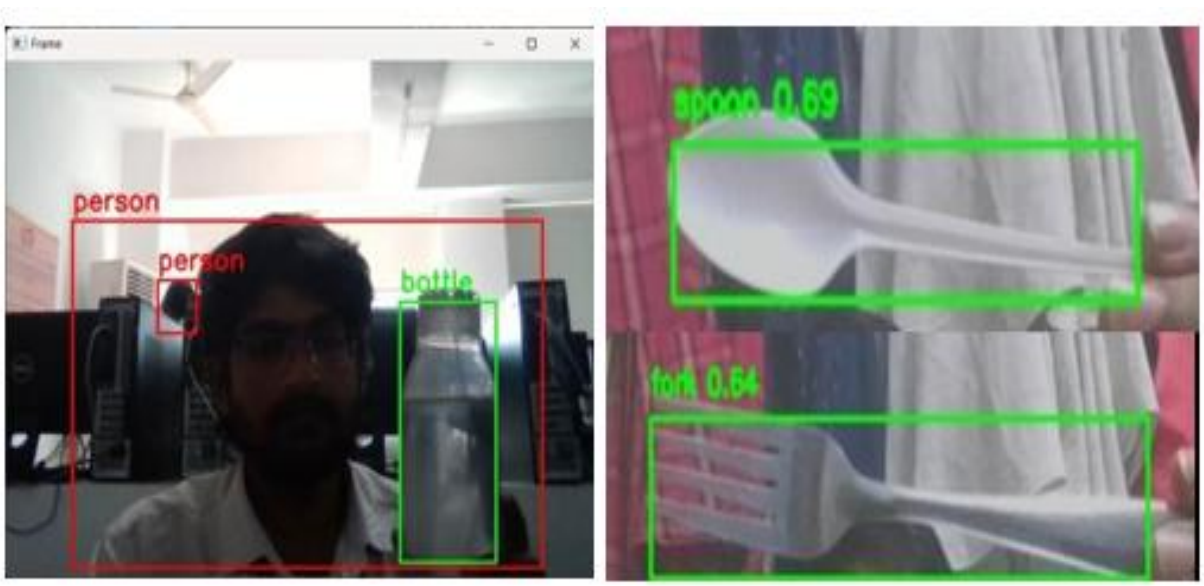

Figure 8: Sample outputs from SSD model showing plastic object detection with bounding boxes.

Metric	Observed Value
Object Detection Accuracy	85.4%
Inference Time per Frame	57 ms
Servo Arm Operation Time	1.2 s (avg)
Navigation Response Delay	180 ms
Cloud Sync Latency (Azure)	200 ms
Average Power Consumption	14.6 W

Table 3: Performance Evaluation Metrics

VI. CONCLUSION

This paper presented the design and implementation of an FPGA-enabled Autonomous Rag Picking Bot aimed at transforming urban waste management through intelligent automation. The proposed system integrates real- time image processing on the DE10-Nano FPGA platform using an SSD model, along with robotic actuation via a Raspberry Pi-controlled 2-DOF arm and movement logic driven by Arduino-based motor control.

The bot effectively demonstrated real-time object detection, autonomous navigation, and reliable pick-and-place actions for plastic waste under varying environmental conditions. Its cloud integration via Microsoft Azure enabled live tracking, bin status monitoring, and data-driven insights, thereby improving scalability and smart city integration.

By combining edge AI, hardware acceleration through FPGA, and cloud analytics, the system offers a sustainable, contactless, and modular solution to urban sanitation challenges. It reduces human exposure to hazardous waste environments and provides a foundation for scalable deployment in smart city infrastructure.

VII. FUTURE WORK

To further improve the system's performance and adaptability, the following extensions are proposed:

- Multi-object Detection: Integration of lightweight YOLO models (e.g., YOLOv4-Tiny) to detect multiple waste items in a single frame.
- Material Classification: AI-based classification to segregate waste as biodegradable or non-biodegradable.
- Swarm Coordination: Deployment of multiple bots with inter-bot communication for large-area coverage.
- Energy Optimization: Implementing dynamic power scaling and sleep modes to improve battery life during idle states.
- Mobile App Interface: A real-time mobile dashboard for both control, monitoring, and alert systems.

These improvements will extend the system's functionality and make it more adaptable to real-world, high-volume waste collection scenarios.

ACKNOWLEDGMENT

The authors would like to thank their institute and faculty mentors for their guidance and support throughout the development of this project. Special thanks to Intel, Microsoft Azure, and Terasic for providing tools and plat- forms essential to this research. We also extend our gratitude to SSIT Pvt. Ltd. for their valuable support and contributions to this project.

REFERENCES

[1] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767, 2018.

[2] W. Liu et al., "SSD: Single Shot MultiBox Detector," in ECCV, 2016.

[3] Microsoft Azure Cloud Platform, Available: https://azure. microsoft.com

[4] DE10-Nano User Guide, Terasic Technologies, 2022. Available: https://www.terasic.com

[5] Raspberry Pi Documentation, Available: https://www.raspberrypi. com/documentation

[6] Intel, "FPGA Cloud Connectivity Kit Overview," Intel White Paper, 2022.

[7] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in CVPR, 2005.

[8] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in NeurIPS, 2012.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real- Time Object Detection," in NeurIPS, 2015.

[10] H. Li et al., "Design and Control of a Robotic

Arm for Waste Sorting," IEEE Access, vol. 8, 2020.

[11] A. Sangiovanni-Vincentelli, "The Role of Embedded Systems in Smart City Applications," IEEE Design & Test, vol. 33, no. 5, 2016.

[12] Z. Zhang, "Embedded Machine Learning for Edge Computing," ACM Computing Surveys, vol. 54, no. 8, 2022.

[13] S. Kamble, A. Gunasekaran, and R. Sharma, "Smart Waste Management in Cities Using IoT," IEEE Smart Cities, vol. 9, no. 3, pp. 1–8, 2019.

[14] M. Kaur and A. Kaur, "Comparative Analysis of YOLO and SSD for Real-Time Object Detection," in IEEE ICMLA, 2021.

[15] T. Dinh et al., "FPGA-based Real-Time Object Tracking for Autonomous Robots," in IEEE SoC Conference, 2020.

[16] M. Srivastava, T. Teixeira, and S. Madden, "Sensor Network Applications in Urban Environments," IEEE Communications Magazine, vol. 49, no. 11, 2011.

[17] J. Verma and P. Kumar, "IoT-Based Smart Waste Management System Using Cloud Computing," in IEEE ICCSP, 2021.

[18] Y. Bengio et al., "Deep Learning for Real-Time Applications: Efficiency, Latency, and Power," IEEE TPAMI, vol. 44, no. 8, 2022.

[19] V. R. Borkar et al., "Smart Garbage Collection System Using AI and FPGA," in IEEE ICEECCOT, 2020.

[20] COCO Dataset – Common Objects in Context, Microsoft Research. Available: https://cocodataset.org