Using Image Classification to Create an Intelligent Recycling Bin

Samadhi Chandrasena

Earlham College, Richmond, Indiana, skchand22@earlham.edu

Abstract :

Improper waste sorting leads to contaminated recyclables, increased landfill waste, and higher waste management costs. To address these issues, this project proposes the design and development of a smart trash sorting system that combines image classification with physical actuation to automatically categorise and direct waste into the appropriate bins. A camera captures real-time images of incoming waste, which are then processed by a pre-trained machine learning model to identify the type of material (e.g., plastic or paper). Based on the classification result, a microcontroller controls a simple mechanical system (e.g., servo-powered gate or rotating platform) to sort the item accordingly. The prototype will be tested for model accuracy, sorting success rate, and system response time. By integrating machine learning with hardware, this project aims to demonstrate a low-cost, scalable solution that can reduce human error and improve efficiency in household or small-scale recycling processes.

Index Terms: Smart trash sorting system, Machine learning, Waste management automation

1 Introduction

In many public and household settings, people often dispose of items incorrectly due to confusion, lack of awareness, or insufficient infrastructure, which reduces the effectiveness of recycling efforts. Manual sorting can help, but it is time-consuming, inconsistent, and often impractical in high-traffic environments.

This project aims to address these challenges by developing a small-scale, automated waste sorting system that uses image classification and simple hardware control. The system will detect and identify common waste types using a camera and a machine learning model. Based on the classification results, the system will then activate a basic actuator mechanism controlled by an Arduino to direct the waste item into the appropriate bin.

The proposed system could be adapted for use in smart homes, schools, or public waste stations, particularly in areas where environmental education and recycling habits are still developing.

1.1 Contribution

This paper has the follow contributions:

- A trained and tested image classifier optimised for small datasets that can sort trash items into two or more categories.
- A fully working prototype that classifies and physically sorts waste items into separate bins.

2 Related Work

2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have proven highly effective in automated waste classification systems. Several studies demonstrate that CNN-based models like YOLOv8, VGG16, and EfficientNet consistently achieve high accuracy in classifying different waste materials. Akbar, 2023 et al. (2023) tested multiple CNN models, reporting that YOLOv8 achieved the highest precision at 96.5% on the TrashNet dataset. YOLOv8's strong performance in object detection and its efficiency in real-time applications make it especially suitable for waste sorting systems where speed and resource constraints are critical. Similarly, VGG16 and EfficientNet have shown strong classification capabilities in related contexts. These results highlight the importance of choosing a deep learning model that balances accuracy and processing efficiency for real-world deployments.

2.2 Lightweight Models for Low-Resource Deployment

Many real-world waste classification systems are meant to run on small devices like Raspberry Pi or Arduino, which have limited memory and processing power. To make this possible, researchers use lightweight CNN models that require less computation but still give good results. For example, Deeluea, 2022 et al. (2022) used EfficientNet-Lite0 — a compact model designed for mobile or edge devices — and achieved 93% accuracy on a dataset of over 2,800 images. Similarly, Mendy et al., 2020 et al. (2020) showed that even small models can be effective for classifying waste when used in smart bins. These lightweight models are ideal for projects like this one, where we want to deploy the image classifier on low-power hardware without relying on a separate server.

2.3 Data Augmentation

A recurring challenge in waste classification is the limited availability of large, high-quality datasets. Data augmentation has emerged as a key technique to address this problem by artificially expanding training datasets. Studies such as Akbar, 2023 et al. (2023) and Deeluea, 2022 et al. (2022) applied augmentation strategies like random rotation, flipping, and scaling to prevent overfitting and improve model generalization. Rad et al. (2017) similarly reported a 12% increase in accuracy using augmentation techniques when training on a small dataset. These enhancements allow AI models to handle variations in lighting, positioning, and object orientation — all of which are common in real-world waste sorting. Incorporating data augmentation is therefore a practical and essential step for improving robustness when training on limited datasets.

2.4 Dataset

The TrashNet dataset is frequently used in studies involving AI-based waste classification. It contains images of waste across a few categories such as paper, plastic, metal, and glass. While useful for benchmarking, researchers often note its limitations, particularly the small dataset size and limited diversity. This makes data augmentation or custom dataset creation a necessary step in many projects. Some researchers also supplement TrashNet with additional images or curate their own datasets to increase robustness.

Figure 1. Sample images with labels from the TrashNet Dataset

However, TrashNet's limited diversity has led researchers to explore more complex datasets like TACO (Trash Annotations in Context), which includes annotated images of litter found in real-world environments with cluttered backgrounds, occlusion, and varying lighting conditions. Other datasets such as WasteNet and EcoWasteNet offer larger and more heterogeneous collections, allowing for the training of more generalised models. These datasets enable comparative analysis of different models' performance and help advance the development of more resilient waste-sorting systems.

For this project, we will take two out of six classification types (plastic and trash) to perform a binary classification.

2.5 Hardware

In addition to software advances, several projects have explored hardware implementations for automated waste sorting. Many of these use simple actuators or servo motors connected to microcontrollers such as Arduino or Raspberry Pi to physically redirect waste into appropriate bins. While some systems rely on basic metrics like weight or sensor-based triggers, fewer integrate visual classification with mechanical actuation. This combination presents challenges, particularly due to the limited processing power of microcontrollers, which makes real-time image classification difficult to perform on-device. As a result, many systems adopt a hybrid approach—using a PC or cloud service to handle classification, while the Arduino or similar device performs the physical sorting. This method balances the strengths of powerful image recognition with low-cost, accessible hardware control.

Some notable implementations of hardware-based waste sorting systems demonstrate the practicality of this hybrid approach. For example, projects like the one by Mendy et al. (2020) combined a lightweight CNN model running on a computer with servo motors controlled by an Arduino to sort items into recyclable and non-recyclable categories. Other designs, such as those outlined in Deeluea et al. (2022), prioritized low-cost and portable systems by deploying image classification on mobile devices or Raspberry Pi units, while keeping mechanical actuation simple—often just rotating arms or sliding trays. These designs emphasize accessibility and affordability, making them suitable for educational settings or small-scale deployment. However, limitations such as slow sorting speeds, limited model inference capabilities on embedded devices, and the need for consistent lighting and camera angles remain common challenges across many hardware-oriented solutions.

2.6 Approach for this Project

Building on these prior studies, this project implements a lightweight Convolutional Neural Network trained to classify images into plastic or trash categories. Unlike high-complexity models like YOLOv8, this approach prioritises deployability on a Raspberry Pi, balancing accuracy with low processing requirements. The dataset will be a curated, custom dataset of labeled waste images from the TrashNet dataset, incorporating data augmentation techniques (such as rotation and flipping) to address the limited size and variability of available datasets. The trained model is then deployed on the Raspberry Pi, which handles image preprocessing and inference in real time. Based on the classification result, the system controls servo motors to route waste items into the correct bin.

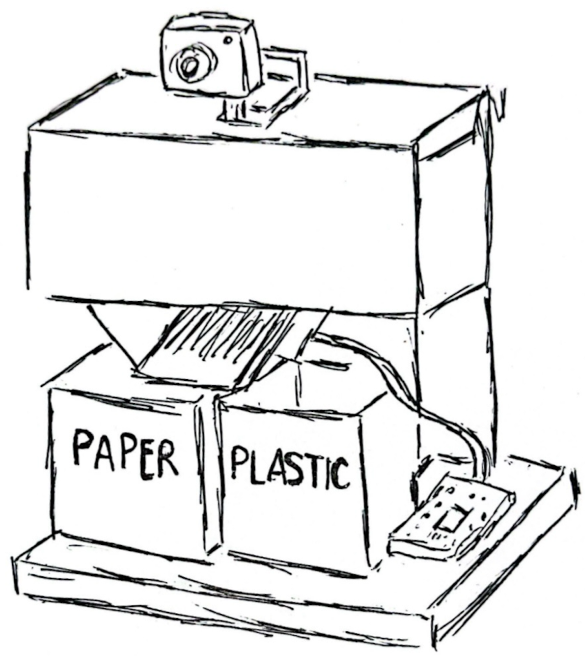

Figure 2. Visual Depiction of System Set-Up

3 Design and Implementation

The proposed system is an end-to-end, autonomous waste classification and sorting unit powered entirely by a Raspberry Pi. It combines real-time image classification using a trained convolutional neural network (CNN) model with physical actuation via servo motors to sort objects into predefined bins based on material type (e.g., plastic, paper, or trash). This system is built to demonstrate the practical deployment of machine learning in resource-constrained environments, with the entire pipeline—from image capture to mechanical sorting—handled on-device.

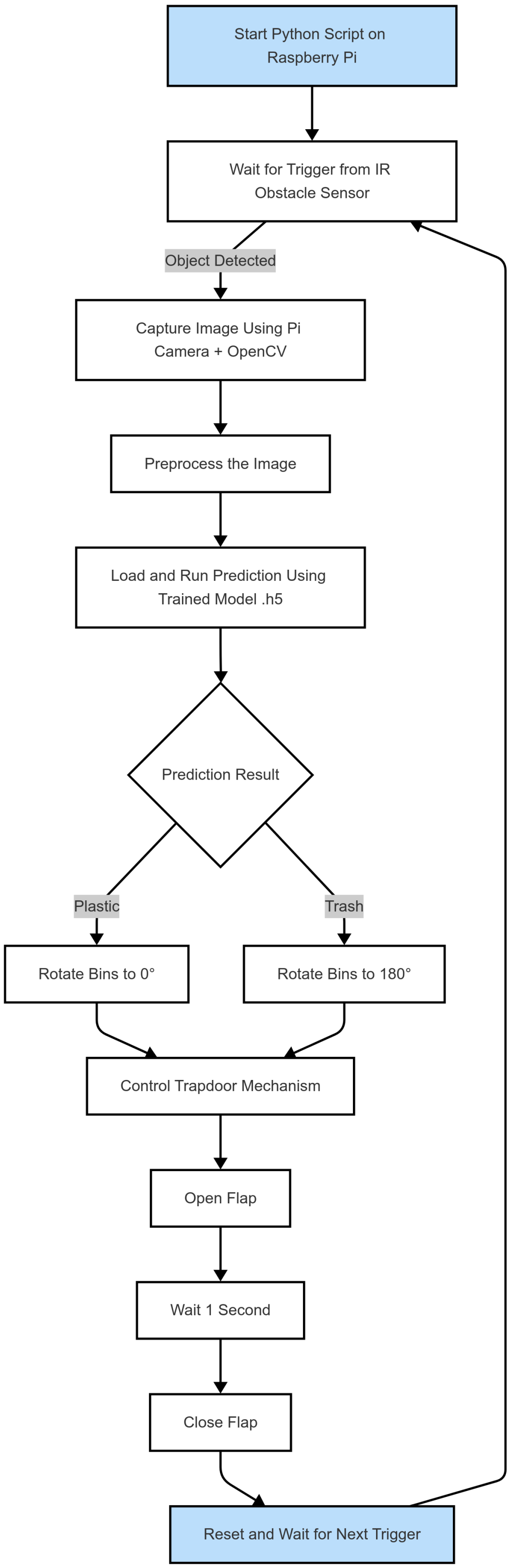

Figure 3. Process Flowchart

3.1 System Initialisation and Monitoring

The process begins by initialising a Python script on the Raspberry Pi, which serves as the central control unit for both the software and hardware subsystems. On startup, the script configures the GPIO pins and sets up the necessary peripherals, including the IR obstacle sensor and the Pi Camera. The IR sensor is connected using the necessary pins, and the system constantly monitors the state of the OUT GPIO pin.

When an object is introduced into the sorting chute and comes into the IR sensor's detection range, the output signal on the OUT pin drops from HIGH to LOW. This transition triggers the classification and sorting routine.

3.2 Image Capture and Preprocessing

Upon detecting an object, the Pi captures an image using either the built-in PiCamera module or OpenCV's cv2.VideoCapture(0) interface. The frame is captured in real time and stored as a NumPy array representing RGB pixel values. The image then undergoes a preprocessing pipeline tailored to the model's input specifications. This includes:

Resizing the image to the required input size.

Normalising pixel values to a range of 0–1 for consistency with the training set.

(Optional) Color space adjustments, contrast enhancement, or noise reduction, depending on the model's training conditions.

This preprocessing ensures the model receives input in the same format and quality as it was trained on, thereby improving classification reliability.

3.3 Model Loading and Inference

Once preprocessed, the image is passed to a trained CNN model, saved in .h5 format and loaded using TensorFlow/Keras. If resource constraints demand a lighter model, a TensorFlow Lite version (.tflite) can be substituted. The model performs inference and outputs a class prediction—for instance, 0 = plastic, 1 = paper, 2 = trash.

The output class index is then mapped to a human-readable label and passed on to the sorting mechanism.

3.4 Hardware Control for Sorting

The physical sorting mechanism comprises two servo-controlled components:

Rotating Platform: A continuous rotation servo motor is used to align the correct bin underneath the drop zone. Depending on the predicted class:

Plastic → Rotate to 0°

Trash → Rotate to 180°

The rotation is calculated using time-based calibration or angle-based control using a PWM signal. The system includes a short delay to ensure the platform settles into the correct position before the item is released.

Trapdoor Flap: Positioned at the base of the chute is a hinged flap controlled by a positional servo motor. Once the platform is correctly aligned, the flap opens to 90°, allowing the item to fall into the bin below. After a short delay (e.g., 1 second), the flap closes to 0°, sealing the chamber for the next item.

3.5 System Reset and Readiness

After the item is sorted, the system reinitialises all motors to their default positions (if needed), clears the camera buffer, and resumes monitoring the IR sensor. This closed-loop cycle continues indefinitely, allowing for continuous autonomous operation.

3.6 Resource Optimisation and Contingency Planning

While the Raspberry Pi is capable of handling lightweight inference models like MobileNet or EfficientNet-Lite0, it may encounter latency or memory issues with heavier architectures. In such cases, fallback strategies include:

Model optimisation via pruning

Offloading inference to an external GPU or server via local network

Simulated output: If hardware components fail or are incomplete, the system can print the predicted class and intended bin to the terminal as a demonstration of the classification pipeline.

4 Preliminary Evaluation

To evaluate the system's performance, several experiments will be conducted:

Model Accuracy: Measured on a labeled test dataset (e.g., TrashNet + additional images).

Sorting Success Rate: Based on how reliably the mechanism delivers items to the correct bin.

Latency: The total time from object detection to sorting completion.

The system will be considered successful if the model achieves at least 85% accuracy, the sorting is correct at least 90% of the time, and the latency remains under 2 seconds per item. These thresholds may be refined during development based on observed system behavior.

5 Timeline

Week 1–3: Dataset Collection and Model Development

- Collect and clean image dataset from sources like TrashNet and manually taken photos
- Perform data augmentation to improve model generalization
- Train and evaluate a lightweight classifier (e.g., MobileNet or TensorFlow Lite model)
- Save model in .h5 or .tflite format compatible with Raspberry Pi

Week 3–4: Raspberry Pi Setup and Sensor Testing

- Configure Raspberry Pi environment with required Python packages (OpenCV, TensorFlow, GPIO libraries)
- Connect and test IR obstacle sensor for object detection using GPIO
- Validate detection trigger logic in Python

Week 4–5: Pi Camera Integration and Image Pipeline

- Capture images with Pi Camera using OpenCV
- Implement image preprocessing (resize, normalize)
- Run inference using trained model on Pi
- Map model predictions to waste categories

Week 5–6: Hardware Control via GPIO

- Connect and test continuous rotation servo for rotating bin platform
- Connect and test standard servo for trapdoor mechanism
- Write control logic in Python for motor response based on classification output

Week 6–8: Full System Integration

- Combine all steps into one Python script: sensor trigger → image capture → classification → actuation
- Fine-tune bin rotation angles and flap timing
- Test complete sorting pipeline with real objects

Week 8–10: Optimization and Robustness

- Address latency, misclassification, or mechanical failures
- Optimize image pipeline or model size if needed
- Log classification and sorting outcomes for evaluation

Week 10–12: Final Deliverables

- Record demo video of working system
- Write and format final report
- Ensure system is reproducible with clear documentation and wiring diagrams

6 Major Risks

Several potential risks could impact the success of this project, particularly concerning classification accuracy, real-world testing, and hardware integration.

First, the image classification model may perform poorly on real-world trash due to background clutter or poor lighting, leading to inaccurate predictions. This will be mitigated by validating the model on real trash photos early (Week 2) and expanding the dataset with more diverse, manually collected images.

Second, model accuracy may remain too low despite improvements. In that case, the classification task will be simplified—such as reducing it to binary categories (e.g., recyclable vs. non-recyclable)—to boost accuracy and robustness. A lightweight model like MobileNet or YOLOv8n will be prioritised to balance speed and accuracy.

Third, servo actuation or the mechanical bin design may prove unreliable or overly complex. This will be tested independently (Weeks 3–5), and if mechanical issues persist, a virtual demo using on-screen bins will serve as a backup to demonstrate the classification and sorting logic.

Finally, integration issues may arise during the transition from model testing to full system assembly (Week 4). Buffer time in Weeks 5–8 is reserved for troubleshooting and optimisation, ensuring sufficient flexibility to address unforeseen integration problems.

Acknowledgements

I would like to thank Dr. Charlie Peck and Dr. Yunting Ying for their assistance in this proposal.

References

Akbar, Mohammad Zubair (2023). *Intelligent Solid Waste Classification System Using Deep Learning*. URL: https://hammer.purdue.edu/articles/thesis/INTELLIGENT_SOLID_WASTE_CLASSIFICATION_SYSTEM_USING_DEEP_LEARNING/23786484?file=41773260.

Chen, H. (2021). "Optimization of an Intelligent Sorting and Recycling System for Solid Waste Based on Image Recognition Technology". In: *Hindawi*.

Cheng, Tianhao et al. (2023). "Smart waste management system based on image recognition and machine learning: A case study of the recycling process in Japan". In: *Sustainability* 16.23, p. 10155.

Deeluea, Saran (2022). "Automatic Waste Sorting Trash Bin by Image Classification". In: *Medium*. URL: https://medium.com/@saran.deeluea/automatic-waste-sorting-trash-bin-by-image-classification-79f22d96498e.

Feng, Zhicheng et al. (2022). "An Intelligent Waste-Sorting and Recycling Device Based on Improved EfficientNet". In: *International Journal of Environmental Research and Public Health* 19.23, p. 15987. URL: https://www.mdpi.com/1660-4601/19/23/15987.

Hani Abu-Qdais, Nawras Shatnawi and Esra'a AL-Alami (2023). "Intelligent System for Solid Waste Classification Using Combination of Image Processing and Machine Learning Models". In: *Journal of Environmental Science and Technology*.

Kang, Xiaoqing, Ying Li, and Liang Zhang (2021). "Domain adaptation techniques for waste classification tasks". In: *2021 International Conference on Machine Learning and Cybernetics (ICMLC)*. IEEE, pp. 98–103.

Mendy, Gabriel et al. (2020). "Smart waste bin: Smart waste management system based on embedded systems, image classification, and IoT". In: *International Journal of Advanced Computer Science and Applications* 11.7.

Olawade, David B. et al. (2023). "Smart Waste Management: A Paradigm Shift Enabled by Artificial Intelligence". In: *Sustainability* 16.23, p. 10155.

Rad, Mahsa Rezanejad et al. (2017). "A deep learning approach for waste classification". In: *2017 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE, pp. 1–5.

Ruíz, V. et al (2019). "Automatic Image-Based Waste Classification". In: *Procedia Computer Science* 153, pp. 422–431.

Zhu, Junya, Rui Zhang, and Jiajun Jiang (2019). "Classification of Garbage Waste Images Using Convolutional Neural Networks". In: *International Journal of Environmental Research and Public Health* 16.17, p. 3148.