SMART WASTE CLASSIFICATION AND SORTING SYSTEM

Andrei-Catalin BUNESCU[1], Rodica-Cristina NEGROIU[2], Cristina-Ioana MARGHESCU[3]

Nowadays, there is increasing emphasis on waste sorting as a global issue that aims to reduce pollution and increase recycling. Manual waste sorting is often difficult and the development of automatic, precise and efficient sorting systems represents a major contribution to this global issue. We present a compact autonomous system that combines embedded deep learning and mechanical sorting. A Raspberry Pi 4 Model B with PiCamera v2 captures images that are processed by a MobileNetV2-based TensorFlow Lite model to classify plastics, metals, and glass. A VL53L0X Time-of-Flight sensor detects objects, a 28BYJ-48 stepper motor drives a 3D-printed conveyor belt and a TD-7120MG servo rotates a multi-chamber bin for sorting.

Keywords: Waste classification, Raspberry Pi, MobileNetV2, Transfer learning, 3D printing, TensorFlow Lite

1. Introduction

Due to the continuous increase in global consumption and urbanization, waste management has become an ever-greater challenge. Traditionally, sorting has been conducted manually in specialized facilities where workers physically separate recyclables based on material type. While partially effective, this approach is labor-intensive, dangerous. Modern recycling relies on consumer separation using color-coded public bins for plastic, glass, paper, and metal; however, compliance varies and human errors remain high.

Thanks to developments in industrial automation, waste sorting is now a very effective, mostly hand-free process. To identify and sort materials in real time, modern facilities use a network of robotic arms controlled by machine learning algorithms, high-speed cameras, infrared and X-ray sensors, and conveyor belts. This significantly reduces the need for manual labor and increases recycling efficiency overall [1].

[1] Student, Dept. of Applied Electronics, National University of Science and Technology Politehnica Bucharest, Romania, e-mail: andrei.bunescu@stud.etti.upb.ro
[2] Lecturer, Dept. of Electronic Technology and Reliability, National University of Science and Technology Politehnica Bucharest, Romania, e-mail: rodica.pavel@upb.ro
[3] Lecturer, Dept. of Electronic Technology and Reliability, National University of Science and Technology Politehnica Bucharest, Romania, e-mail: cristina.marghescu@upb.ro

Considerable progress has been made in the field of automated waste sorting, both in industrial applications and in academic research. Commercial systems such as those developed by “ZenRobotics” and “AMP Robotics”, use real-time sensor fusion and deep learning to recognize and classify different recyclable materials on high-speed conveyor belts. These systems are equipped with robotic arms that perform rapid sorting operations based on material type, shape, and even brand recognition. ZenRobotics has developed systems such as the Fast Picker and Heavy Picker which are designed for material recovery facilities. These robots are equipped with AI-based vision systems and multi-axis robotic arms capable of sorting materials ranging from construction and demolition waste to municipal solid waste and commercial waste streams. Their systems can perform up to 6000 picks per hour and are trained to recognize materials such as wood, metal, stone, plastic, and cardboard. ZenRobotics’ solutions are characterized by sustainability, modularity and 24/7 operation with minimal human intervention [2]. AMP Robotics, on the other hand, focuses on the combination of robotic manipulation and AI-assisted classification to automate the sorting of municipal and industrial waste. Their systems are capable of detecting materials using computer vision and deep learning and redirecting them with high-speed robotic arms. AMP’s solutions are widely used in single-stream recycling, e-waste separation, and mixed plastics recovery. The system uses programmable AI-driven models to recognize and classify each object on the conveyor belt and direct it to the appropriate bin [3].

Inspired by the core concepts used in these real-time sorting facilities, we have designed a simplified prototype that replicates their operating principles on a smaller scale. The system consists of a miniature conveyor belt (implemented as a 3D-printed treadmill mechanism), a Raspberry Pi and Pi camera for real-time software/hardware control and image acquisition, a distance sensor that detects when an object is under the camera, and a specially trained Convolutional Neural Network (CNN) capable of classifying three materials: plastic, metal, and glass as well as the background. Based on the classification result, a servomechanism is triggered to guide the object into the appropriate bin. [2, 3].

This work presents a proof-of-concept, scale automated sorting system that emulates industrial principles with low-cost hardware. The contributions include:

- A lightweight MobileNetV2-based classification pipeline optimized with TensorFlow Lite for embedded inference.
- Integration of a Raspberry Pi 4B controller with ToF sensing, stepper-driven conveyor, and servo driven sorting bin.
- 3D-printed mechanical structures for the conveyor system and electrical components.

2. Proposed method

2.1. System Design and Operation

The system architecture and workflow are illustrated in Fig. 1. The hardware components controlled by of the Raspberry Pi include:

• A 28BYJ-48 stepper motor with ULN2003 driver to drive a 3D-printed conveyor belt;
• A VL53L0X Time-of-Flight sensor for object detection and positioning.
• A PiCamera v2 for real-time image acquisition;
• A TD-7120MG servo motor to rotate a three-chamber sorting bin.
• A 0.96” I2C OLED display for user feedback.

The main actions performed by the hardware components of the system are presented below:

1. **Conveyor belt movement:** The stepper motor moves the conveyor belt continuously, passing the objects under the ToF sensor.
2. **Object Detection:** When the sensor reads a distance below the dynamically set threshold, the conveyor stops and Counts the steps until the object leaves the sensor's field of view.
3. **Re-Centering:** The conveyor reverses half the recorded steps to position the object under the camera.
4. **Image Capture & Classification:** The PiCamera captures a 224×224 RGB image. A TensorFlow Lite MobileNetV2 model classifies the material into plastic, metal, glass or background using test time augmentation (rotations ±15◦ and horizontal flips) and majority voting for robust predictions.
5. **Sorting Actuation:** Based on the predicted class, the servo rotates the bin to 0°, 120°, or 240°. The conveyor advances 2000 steps to drop the object into the correct chamber, then the servo returns to 0°.
6. **Feedback & Looping:** The OLED displays the predicted class and confidence percentage. The system resets and waits for the next object.

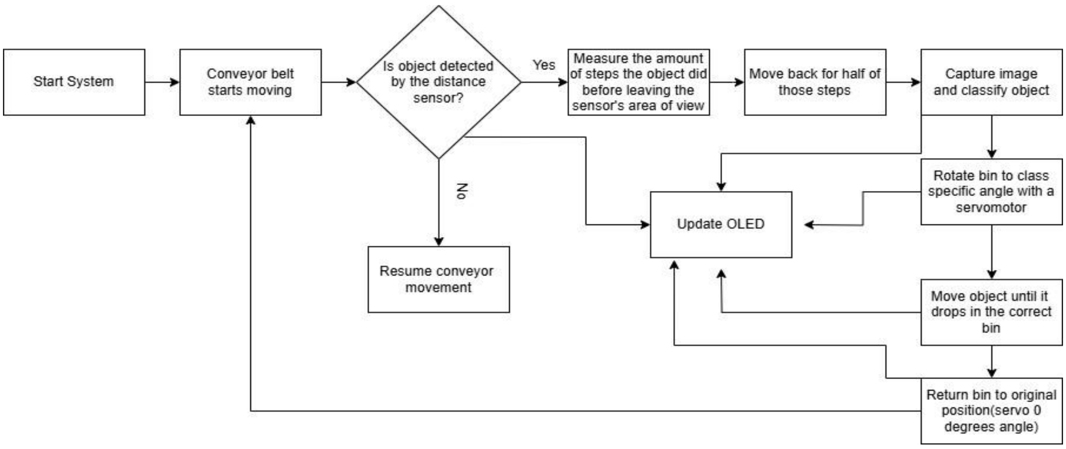


Fig. 1. Functional workflow of the automated sorting system

2.2. Transfer Learning

Transfer learning leverages a pretrained model to improve performance on a related task with limited data [4, 5]. By initializing the network with ImageNet weights, training requires fewer samples and converges faster. In our implementation: **Feature Reuse:** MobileNetV2's convolutional base (pretrained) extracts general features; **Head Customization:** A new classification head tailored to plastic, metal, and glass is trained on waste images; **Fine-Tuning:** Upper layers are unfrozen and retrained to adapt the features to the new domain.

2.3. Model Architecture and Training

We use MobileNetV2 because of its efficiency on embedded devices [6, 7]. The dataset the MobileNetV2 model was trained on is "Waste Classification of Reto: Recycle Together", this is a publicly available dataset from Kaggle and it contains 63000 images of many waste recycling objects of which we extracted the three needed for this project giving us 27000 images to work with. Also, when testing the model on the real-life system it was noticed that objects would be misclassified because of the background of the conveyor belt, which prompted the addition of another class called background comprised purely of belt images without objects on it [9].

The most important aspects are:

- **Inverted Residual Bottlenecks:** Each block expands the input channels by a factor of t, applies a depth wise convolution, and projects back, using ReLU6 activations [6, 8];
- **Linear Bottlenecks:** The projection layer uses no activation to preserve information for quantization;
- **Model Configuration:** See Table 1 for block parameters.

Table 1

MobileNetV2-based transfer learning architecture used for waste classification

Layer	Configuration
Input	224×224×3 RGB image
Feature Extraction Block: MobileNetV2 (pretrained)	weights=ImageNet, include top=False, frozen during head training
Classification Head: GlobalAveragePooling2D	—
Dense	256 units, L_2 regularization =1e−4
BatchNormalization	—
Activation	ReLU
Dropout	rate = 0.4
Dense	128 units, L_2 regularization =1e−4
BatchNormalization	—
Activation	ReLU
Dropout	rate = 0.4
Dense (Output)	4 units, Softmax

Training steps:

1. **Head Training:** Freeze convolutional base, train new dense head (5 epochs, LR=1e-4).
2. **Fine-Tuning:** Unfreeze all layers, train entire network (40 epochs, LR=1e-5).
3. **Optimization:** Use class weights for class imbalance, ReduceLROnPlateau and EarlyStopping callbacks.
4. **Quantization:** Convert to TensorFlow Lite with default optimizations, resulting in a 2.8MB model [10].

3. 3D Printing and Hardware Implementation

3.1. Electrical Components and Circuit Diagram

The electrical architecture of the system consists of the following modules:

- Raspberry Pi 4 Model B: Central controller running the classification and hardware control software [11];
- PiCamera v2: 8 MP Sony IMX219 sensor via CSI for high-resolution image capture [12];
- VL53L0X Time-of-Flight Sensor: I2C distance sensor, measurements up to 2 m for object detection [13];
- 28BYJ-48 Stepper Motor & ULN2003 Driver: Drives the conveyor belt with precise stepping [14];
- TD-7120MG Servo Motor: Rotates the multi-chamber bin to class-specific angles [15];
- 0.96” I2C OLED Display: 128×64 pixel display for status and confidence readouts [16].

Electrical connections and power distribution are summarized in Table 2.

Table 2

GPIO and Power Connections for Electrical Components

Component	GPIO / Interface	Power Supply
VL53L0X ToF Sensor	SDA/SCL (GPIO2/3)	3.3 V (Pi)
OLED Display	SDA/SCL (GPIO2/3)	3.3 V (Pi)
Stepper Driver (IN1–4)	GPIO23, GPIO18, GPIO27, GPIO22	5 V (external)
28BYJ-48 Stepper	ULN2003 output	5 V (external)
TD-7120MG Servo PWM	GPIO24	5 V (external)
PiCamera v2	CSI port	3.3 V (Pi)

To create the electrical circuit diagram we used the EasyEDA program which was developed specifically for this process. The electrical diagram is shown in Fig. 2.

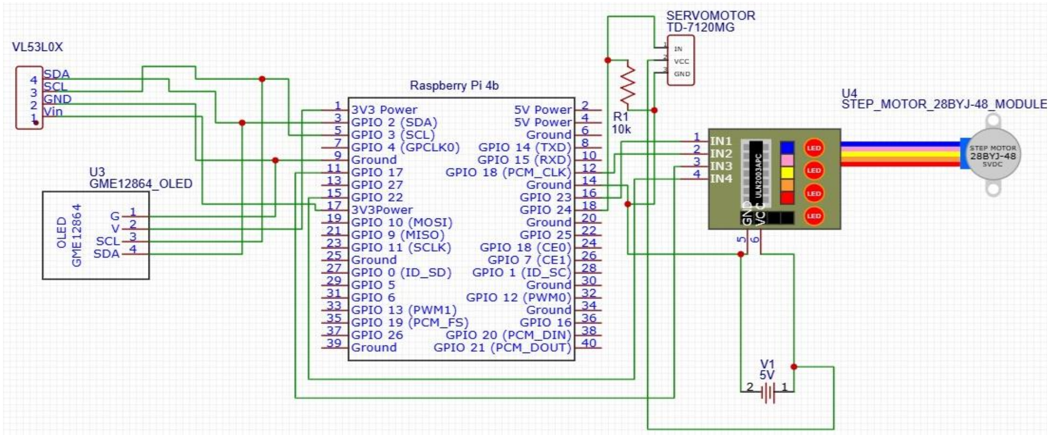


Fig. 2. Electrical diagram showing Raspberry Pi, sensors, motors, and OLED display connections

3.2. 3D-Printed Components

An important step was the 3D design of the system's components. This proved to be the most time-consuming phase. The models were designed in FreeCAD and produced using white PLA filament (1.75 mm). Table 3 lists the main 3D-printed parts of the mini-conveyor system together with their overall dimensions.

Table 3

Key dimensions of 3D-printed support parts

Part	STL file Dimensions [L×W×H] [mm]
Support Bar - V3	141.6×9.88×6.36
Cover - RaspPi	200.0×51.4×4.0
Driver Support - ULN2003	48.8×34.4×32.9
Support - VL53L0X	34.98×9.39×5.99
SteperMotor Support	142.1×30.0×6.0
Side Drum Support	141.4×30.0×6.0
Front Drum	176.0×30.0×15.0
Traction Drum	180.4×30.0×34.9

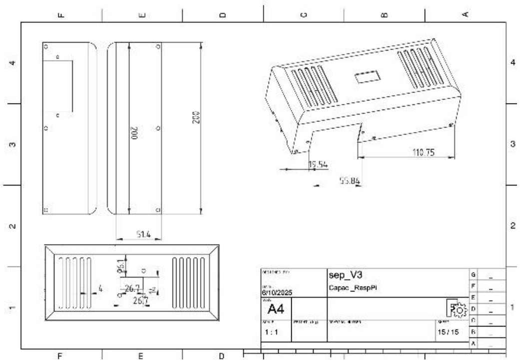


Fig. 3. Custom cover for the Raspberry Pi (CapacRasberyPi.stl)

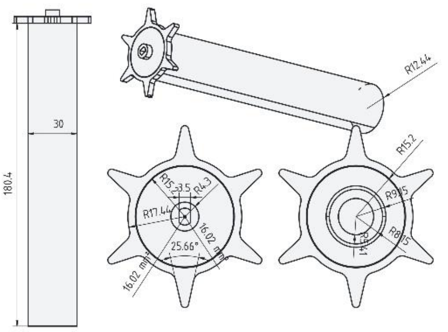
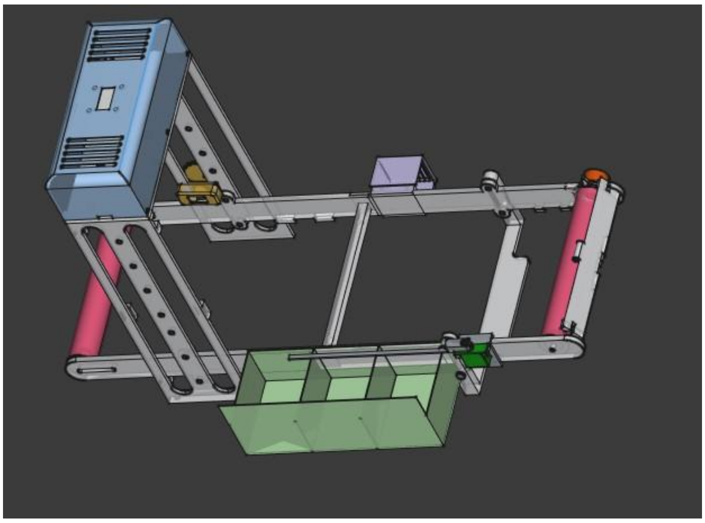


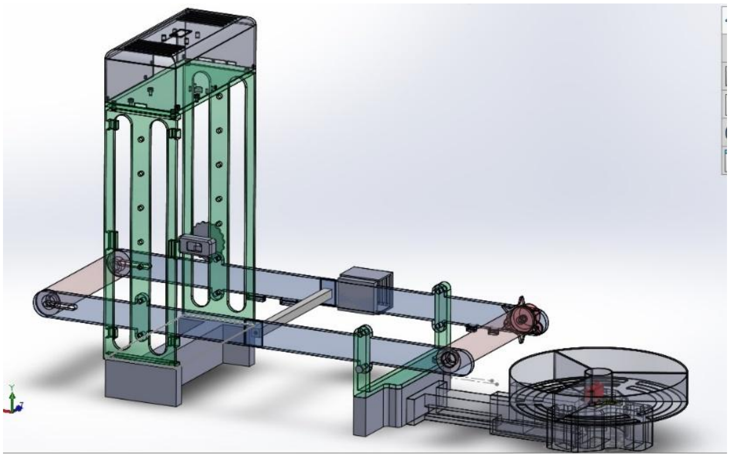
Fig. 4. Drive drum for the conveyor belt (tamburtractiune.stl)

3.3. Assembling the Prototype

Fig. 5a shows the initial prototype configuration. In this layout, after classification a MG90S servomotor pushes each object laterally off the belt into its designated bin. While this works for lightweight or uniformly shaped items, it has been observed that some objects do not fully clear the edge of the conveyor and get stuck between the lanes, resulting in miss-sorting or not falling into the bins at all.



(a) Initial prototype: servo-push sorting mechanism



(b) Upgraded design: end-of-belt rotating bin

Fig. 5. Comparison of the current (a) and upgraded (b) prototype assemblies

To address the shortcomings of the lateral servo approach, the second figure was designed, an improved version of the first module from the first (Fig. 5b) in which classified items travel to the end of the belt and then fall into a rotating multi-chamber bin. The rotation is controlled by a 270º servo TD-7120MG and is synchronized so that each compartment aligns beneath the end of the conveyor in turn, ensuring that every object—regardless of shape or mass— falls cleanly into the correct bin. This failsafe mechanism eliminates the push-force variability and greatly improves sorting reliability.

After 3D realization of the mechanical components of the system, it was assembled with great care and tested to demonstrate its efficiency. Fig. 6 shows the assembled prototype of the sorting system.

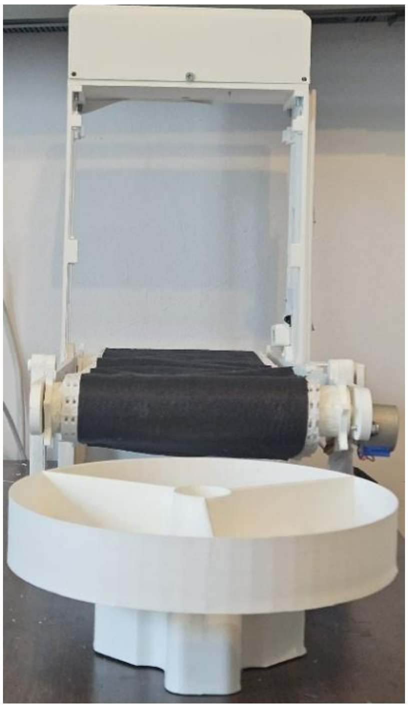


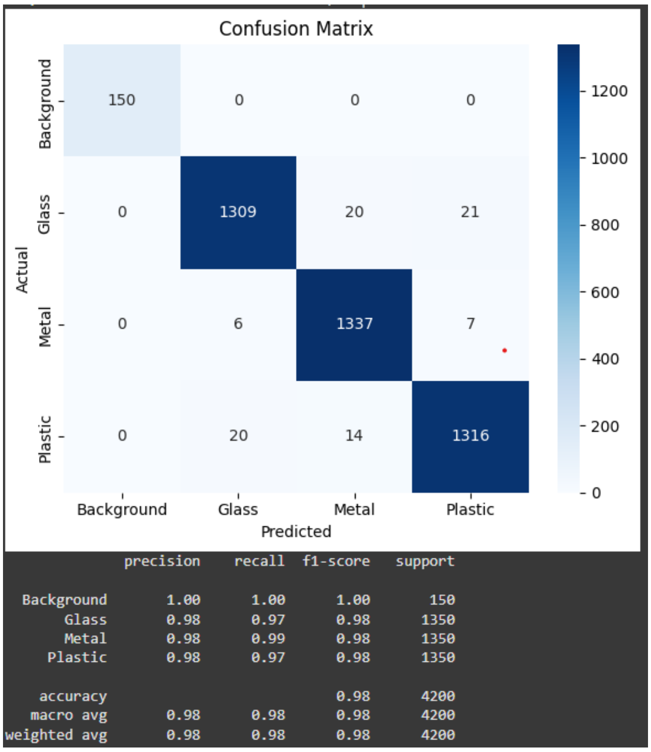
Fig. 6. Front view of the assembled sorting system prototype

4. Experimental Results

To demonstrate the functionality of the system, it was tested on 3 types of materials (plastic, glass and metal) and the results were more than satisfactory. The details of the results are presented in the following subchapters.

4.1. Model Accuracy

Our MobileNetV2 transfer-learning model achieved an accuracy of 98% on the validation set. Fig. 7 shows the confusion matrix, illustrating balanced performance across classes [17].



	precision	recall	f1-score	support
Background	1.00	1.00	1.00	150
Glass	0.98	0.97	0.98	1350
Metal	0.98	0.99	0.98	1350
Plastic	0.98	0.97	0.98	1350
accuracy			0.98	4200
macro avg	0.98	0.98	0.98	4200
weighted avg	0.98	0.98	0.98	4200

Fig. 7. Confusion matrix for the MobileNetV2 model on test data

Fig. 8 plots the training & validation accuracy and loss over epochs, confirming convergence without overfitting.

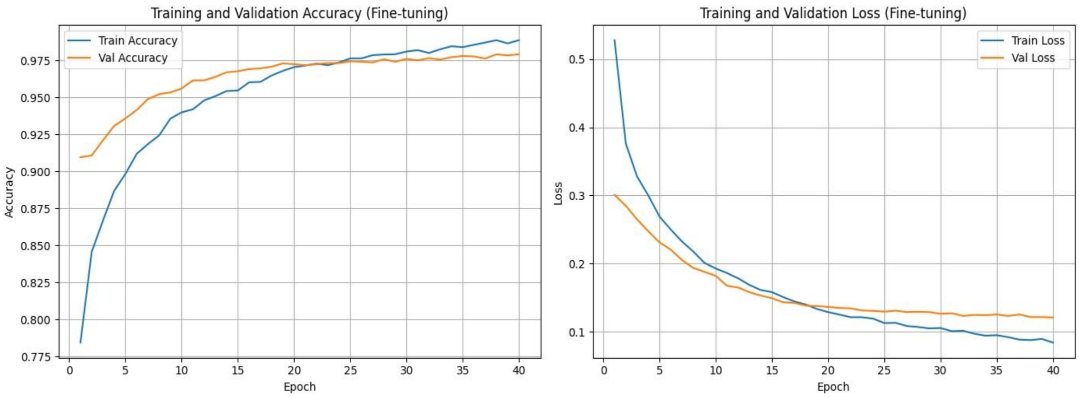
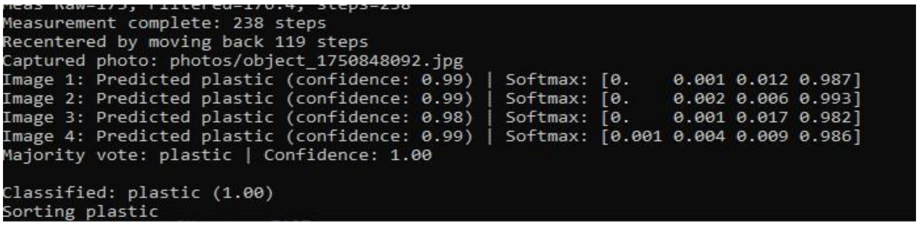


Fig. 8. Training and validation accuracy and loss across epochs

4.2. Prototype Demonstration

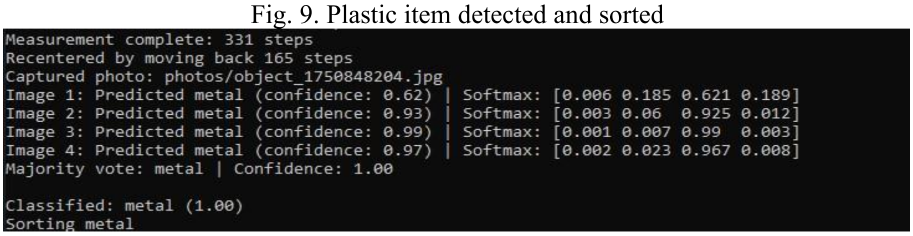
Figs. 9–11 show the system successfully classifying and sorting sample objects in real time.



```
Measurement complete: 238 steps
Recentered by moving back 119 steps
Captured photo: photos/object_1750848092.jpg
Image 1: Predicted plastic (confidence: 0.99) | Softmax: [0.    0.001 0.012 0.987]
Image 2: Predicted plastic (confidence: 0.99) | Softmax: [0.    0.002 0.006 0.993]
Image 3: Predicted plastic (confidence: 0.98) | Softmax: [0.    0.001 0.017 0.982]
Image 4: Predicted plastic (confidence: 0.99) | Softmax: [0.001 0.004 0.009 0.986]
Majority vote: plastic | Confidence: 1.00

Classified: plastic (1.00)
Sorting plastic
```

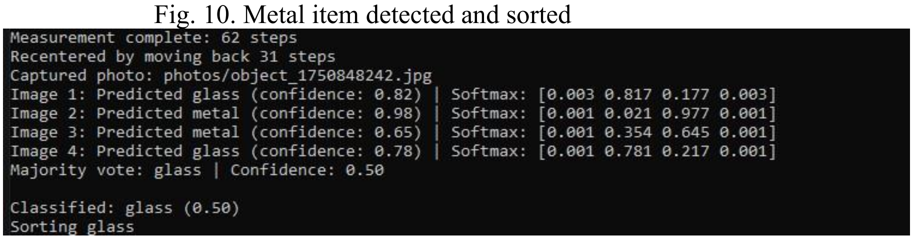
Fig. 9. Plastic item detected and sorted



```
Measurement complete: 331 steps
Recentered by moving back 165 steps
Captured photo: photos/object_1750848204.jpg
Image 1: Predicted metal (confidence: 0.62) | Softmax: [0.006 0.185 0.621 0.189]
Image 2: Predicted metal (confidence: 0.93) | Softmax: [0.003 0.06  0.925 0.012]
Image 3: Predicted metal (confidence: 0.99) | Softmax: [0.001 0.007 0.99  0.003]
Image 4: Predicted metal (confidence: 0.97) | Softmax: [0.002 0.023 0.967 0.008]
Majority vote: metal | Confidence: 1.00

Classified: metal (1.00)
Sorting metal
```

Fig. 10. Metal item detected and sorted



```
Measurement complete: 62 steps
Recentered by moving back 31 steps
Captured photo: photos/object_1750848242.jpg
Image 1: Predicted glass (confidence: 0.82) | Softmax: [0.003 0.817 0.177 0.003]
Image 2: Predicted metal (confidence: 0.98) | Softmax: [0.001 0.021 0.977 0.001]
Image 3: Predicted metal (confidence: 0.65) | Softmax: [0.001 0.354 0.645 0.001]
Image 4: Predicted glass (confidence: 0.78) | Softmax: [0.001 0.781 0.217 0.001]
Majority vote: glass | Confidence: 0.50

Classified: glass (0.50)
Sorting glass
```

Fig. 11. Glass item detected and sorted

4.3. Accuracy Evaluation

The classification accuracy of the system was evaluated using a test set consisting of 10 representative objects for each target class: glass, metal, and plastic. For the glass category, the objects included small bottles, glass shards, and small glass cups. The metal class consisted of aluminum foil, keychains, candle cups, and a metal container. The plastic class was tested with LEGO pieces, plastic bottle caps, coffee cup lids, and pill bottles. After each object was presented to the system under standard conditions, the number of correct classifications was recorded. The results are summarized in Table 4.

Table 4

Classification accuracy for each material class.

Class	Correct / Total	Accuracy (%)
Glass	9 / 10	90
Metal	7 / 10	70
Plastic	6 / 10	60
Background	10/10	100

Overall, the system demonstrated the highest accuracy for glass objects, followed by metal, with plastic items being the most challenging to classify correctly. It is important to note that the position of the objects in relation to the camera and the lighting conditions in the environment had a significant impact on the classification performance. Objects that were not centered in the camera's field of view or were tested in poor lighting conditions were more likely to be misclassified.

4.4. Final Discussions

For waste classification tasks, state-of-the-art convolutional neural network architectures such as ResNet-50 and MobileNetV2 are commonly used due to their strong performance on image recognition benchmarks. ResNet-50, with its many residual layers and higher parameter count, often achieves slightly higher accuracy. However, this comes at the cost of significantly larger model size (approximately 96 MB) and increased computational demands, which makes it less suitable for real-time inference on embedded platforms like the Raspberry Pi 4 [18].

MobileNetV2 was specifically chosen for this project due to its lightweight architecture, small memory footprint (2.8 MB after quantization), and fast inference speed. These qualities make it well-suited for deployment on resource-constrained devices without sacrificing much accuracy. Although this particular waste classification dataset has not been used in any published papers, the MobileNetV2 model achieved a high accuracy while ensuring efficient operation on embedded hardware, making it the optimal balance solution between state-of-the-art performance and practical deploy ability for our system.

The responsiveness of the system was evaluated by measuring the time required for each stage of the sorting process, from object recognition to final actuation.

• No significant bottlenecks were observed in the detection or classification stages, the longest delays were attributed to the stepper motor while the object was in the field of view of the ToF sensor.

• The system was able to reliably process and sort one object at a time with minimal latency between events.

All timing measurements confirm that the prototype is suitable for low- to medium-throughput applications, where objects are presented sequentially.

Despite the successful operation, several limitations and challenges were identified during testing:

• The VL53L0X sensor can provide inaccurate distance measurements when objects have irregular shapes, such as multiple holes in them, or when they are not tall enough for the sensor to detect.

• Model accuracy decreases when objects are outside the center of the camera's field of view or under poor lighting conditions. In addition, some classes perform better than others: glass is identified most reliably, followed by metal, with plastic being the most challenging.

• The system can handle only one object at a time and is not robust enough for simultaneous or overlapping items on the conveyor belt.

5. Conclusions

We have demonstrated a compact, cost-effective embedded system for automated waste sorting that seamlessly combines computer vision with mechanical actuation. By leveraging a Raspberry Pi platform, a distance sensor, a PiCamera v2, a stepper-driven conveyor, and a servo-controlled multi-chamber bin—coordinated through Python and TensorFlow Lite—the prototype achieves real-time classification and physical separation of plastic, metal, and glass obtaining an accuracy of 90% for glass, 70% for metal and 60% for plastic for 10 different objects for each tested class.

Key limitations include reduced performance on irregular or overlapping items, occasional misclassification of ambiguous materials, and constraint by single-item handling and stepper motor speed. Future enhancements will address these areas by expanding the training dataset, refining the deep-learning model, and mechanical redesigns for parallel sorting and faster actuation. The integration of adaptive lighting control and additional sensing modalities will further increase robustness in varied environments.

Moving forward, the following directions are proposed to address current system limitations and enhance performance:

- Creation of custom dataset

Develop a dataset specifically tailored to the conveyor-based setup, using objects that can be physically placed and moved by the system. This will improve classification by reducing discrepancies between training data and real world applications;

- Lighting Control

Integrate lighting components in the form of LED's around the camera to ensure consistent illumination, this will reduce the influence of ambient light and shadows leading to a more reliable image capture and classification and also a more reliable system which can be deployed anywhere;

- Parallel sorting mechanisms

Redesign the mechanical structure to enable the handling and sorting of multiple objects simultaneously. This will increase the system's handling ability and make the prototype more applicable to real world recycling scenarios.

REFERENCES

[1] *G. Maier, R. Gruna, T. Langle, and J. Beyerer*, A Survey of the State of the Art in Sensor-Based¨ Sorting Technology and Research, IEEE Access, vol. **12,** pp. 6477–6478, 2024.

[2] Recycling Inside, ZenRobotics Launch Fourth Generation of Waste Sorting Robots, 2024. [Online]. Available: https://recyclinginside.com/ recycling-technology/separation-and-sorting-technology/ zenrobotics-launch-fourth-generation-of-waste-sorting-robots/.

[3] Businesswire, AMP Robotics Develops Industry's First AI-Powered System for Recovery of Film and Flexible Packaging, 2022. [Online]. Available: https://www.businesswire.com/news/home/20221019005267/en/AMP-Robotics-Develops-Industrys-First-AI-Powered-System-for-Recovery-of-Film

[4] *S. J. Pan and Q. Yang*, A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering, vol. **22**, no. 10, pp. 1345–1359, 2010.

[5] *F. Zhuang et al.*, A Comprehensive Survey on Transfer Learning," Proceedings of the IEEE 109.1 (2020), pp. 43-76.

[6] *Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C*. Mobilenetv2: Inverted residuals and linear bottlenecks, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510-4520, 2018.

[7] Keras, MobileNet and MobileNetV2, 2025. [Online]. Available: https://keras.io/api/ applications/mobilenet/

[8] Stanford University, CS231n Convolutional Neural Networks for Visual Recognition. [Online]. Available: https://cs231n.github.io/convolutional-networks/

[9] D. Samuel Sitorus, Waste Classification of Reto: Recycle Together, Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/davidsamuelsitorus/waste-classification

[10] TensorFlow [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/lite/ TFLiteConverter

[11] Raspberry Pi Foundation, 'Specifications: Raspberry Pi 4 Model B," 2024. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ specifications/

[12] Raspberry Pi Foundation, 'Camera Module v2 Documentation," 2024. [Online]. Available: https://www.raspberrypi.com/documentation/accessories/camera.html

[13] ArduShop.ro, 'Modul VL53L0X – Senzor de Distan¸ta˘ (THT breakout)," 2025. [Online]. Available: https://ardushop.ro/ro/module/1861-modul-vl53l0x-senzor-de-distanta-6427854028143.html

[14] Mouser Electronics, '28BYJ-48 Stepper Motor Data Sheet." [Online]. Available: https://www.mouser.com/datasheet/2/758/stepd-01-data-sheet-1143075.pdf

[15] TinyTronics, 'Data Sheet of TD-7120MG Digital Servo Motor," 2025. [Online]. Available: https://www.tinytronics.nl/product_files/006299_Data%20Sheet% 20of%20TD-7120MG%20Digital%20Servo%20Motor.pdf

[16] Adafruit Industries, "Monochrome OLED Breakouts: Python Usage", 2024. [Online]. Available: https://learn.adafruit.com/monochrome-oled-breakouts/ python-usage-2

[17] scikit-learn developers, ‘sklearn.utils.class weight.compute class weight — scikit-learn 1.3.0 documentation,” 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_ weight.html

[18] *He, K., Zhang, X., Ren, S., & Sun, J.* Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016