

Article

EcoSortBin: Accuracy–Generalisation Trade-Offs in Open-Vocabulary Campus Waste Detection on Raspberry Pi 4

Madhini Balasundaram  and Supraja Perumal * 

Department of Networking and Communications, School of Computing, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai 603203, Tamil Nadu, India; mb7992@srmist.edu.in

* Correspondence: suprajap@srmist.edu.in

Abstract

Waste management on university campuses is complicated by the constant change in packaging types, which existing waste-sorting systems cannot recognise unless they are retrained. Open-vocabulary object detectors can identify objects from text descriptions instead of a fixed list of categories, offering a possible solution to this problem. However, it is not known how well this ability survives when such a detector is fine-tuned and deployed on low-power hardware. This paper presents **EcoSortBin**, a waste-sorting system built on the YOLOE-26 detector and deployed on a Raspberry Pi 4. YOLOE-26 was first fine-tuned on a 1330-image campus waste dataset covering seven classes, with masks generated using the Segment Anything Model, establishing a baseline called **WasteYOLOE26-S** with 74% top-1 accuracy on known classes; however, this fine-tuning reduces the model's ability to recognise the same seven classes when they appear in a different dataset or setting. **RLPA** (RepRTA-Compatible LoRA Prompt Adapters) addresses this by adapting only the text-embedding component of the model using a small set of additional parameters (16,384 parameters, rank 16), leaving the rest of the network unchanged; this restores cross-domain generalisation but reduces top-1 accuracy on known classes to only 15%, which is too low for practical use. To recover this accuracy without losing cross-domain generalisation, frozen-backbone neck fine-tuning (**NeckFT**) was added, which fine-tunes the feature-combining layers of the network while keeping the main backbone frozen, preserving its pretrained visual–text alignment. Combining RLPA with NeckFT achieved the best balance of the three approaches, with 73.5% top-1 accuracy and a Cross-Domain Generalisation Ratio (CDGR) of 0.2435. To test whether this ability extends to genuinely new categories, the model was further tested on 28 novel categories not seen during training, totalling 840 images. RLPA + NeckFT showed consistent zero-shot generalisation to novel objects with container-like shapes, such as bottles and jars. After quantisation for edge deployment, the model kept its full accuracy ranking and produced a compact 41.8 MB file suitable for the Raspberry Pi 4. These results show that RLPA + NeckFT gives a practical balance of accuracy and generalisation for campus waste detection on low-power hardware.



Academic Editor: Miguel Angel Cazorla

Received: 12 June 2026

Revised: 25 July 2026

Accepted: 30 July 2026

Published: 7 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: open-vocabulary detection; YOLOE-26; LoRA prompt adapters; neck fine-tuning; zero-shot generalisation; campus waste; Raspberry Pi 4; edge AI; quantisation-aware training

1. Introduction

Campus recyclable waste streams carry contamination rates of 40–70%, driven by ambiguous bin labelling, the absence of point-of-disposal feedback, and limited student

familiarity with packaging-specific recycling rules [1]. Automated waste detection at collection points has the potential to reduce contamination in real time, but practical deployment on embedded hardware such as the Raspberry Pi 4 (Cortex-A72 CPU, 8 GB LPDDR4X) imposes strict constraints on model size and inference latency.

Previously proposed supervised waste classification approaches [2–7] achieve high accuracy on fixed training vocabularies but fail silently on novel packaging categories not seen during training. As consumer product lines evolve, this closed-vocabulary constraint causes silent misclassification failures that cannot be corrected without full retraining—a cycle incompatible with operational deployment timescales.

Open-vocabulary detection models based on CLIP [8] enable zero-shot detection of new categories via natural language prompts. YOLOE-26s [9] achieves 29.9% LVIS mAP via RepRTA re-parameterisation, which permanently folds the text encoder into the weights of the classification head, eliminating any text-encoder dependency at inference—a prerequisite for Raspberry Pi 4 deployment.

This paper makes four contributions, organised as a complete accuracy–generalisation study:

1. **WasteYOLOE26-S (full fine-tuning baseline):** SAM-supervised fine-tuning of YOLOE-26 achieves 74% top-1 accuracy across seven campus waste classes, establishing a fine-tuned open-vocabulary waste detector on a Raspberry Pi 4 ARMv8. We characterise the resulting CDGR collapse ($6.055 \rightarrow 0.0055$) under domain-specific fine-tuning, which motivates contributions 2 and 3.
2. **RLPA:** RepRTA-compatible LoRA prompt adapters modify only the MobileCLIP text-embedding space, achieving CDGR = 3.451, with zero inference overhead and RepRTA compatibility preserved. However, RLPA alone achieves only 15% top-1 accuracy on seen classes, since the adaptation of text embeddings cannot compensate for the visual domain gap between COCO pretraining and campus waste appearance.
3. **RLPA + NeckFT:** To recover seen-class accuracy without repeating the CDGR collapse of full fine-tuning, frozen-backbone neck fine-tuning adapts the visual feature fusion layers while keeping the CLIP-aligned backbone intact, reaching 73.5% top-1 accuracy (mAP50 = 0.538, CDGR = 0.2435) on the 98-image held-out test split. On a separate, larger 840-image, 28-category validation set, it continues to produce confident predictions for most novel categories (69.6% detection rate versus 12.0% for RLPA alone) while recovering seen-class accuracy.
4. **QAT robustness study:** The first measurement of quantisation-aware training on a waste-domain open-vocabulary embedding head reports Kendall $\tau = 1.00$ rank preservation and $\Delta\text{Top-1} = +4$ pp (vs. NeckFT), demonstrating that the model’s cosine similarity ordering is stable under INT8-level numerical noise and confirming readiness for INT8 deployment when toolchain support becomes available. A deployable ONNX FP32 model (41.8 MB) has been released for Raspberry Pi 4; an NCNN artefact is also provided for future native C++ deployment.

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 describes the EcoSortBin system and dataset. Section 4 presents the three-stage methodology. Section 5 reports results. Section 6 discusses findings. Section 7 concludes this paper.

2. Related Work

2.1. Automated Waste Classification

CNN-based waste classification was systematically studied by Aral et al. [2] and Thung and Yang [3] on the TrashNet dataset, establishing a six-class waste vocabulary that

became a de facto benchmark. TACO [10] extended this to 60 categories in outdoor litter contexts. Detection-based approaches using YOLO-family architectures have been applied to smart bin systems, with mAP50 values of 0.71–0.94 on restricted laboratory datasets [5,6]. More recent work has pushed accuracy further: GD-YOLOv10n [11] achieves 86.7% mAP on a combined TACO and proprietary dataset at 121.7 FPS, and EcoDetect-YOLO [12] applies CBAM attention with YOLOv5s to achieve 58.1% mAP in urban garbage detection. Kunwar and Alade [13] evaluated YOLO-10n and YOLO-11n on the WaDaBa plastic waste dataset, reporting mAP50 values of up to 0.992, demonstrating that supervised detectors can achieve near-perfect accuracy on closed laboratory datasets. However, these high accuracy numbers do not transfer to deployment. Bobulski and Kubanek reported significant performance drops when models were evaluated on packaging collected six months after training [7], highlighting the temporal-drift problem that arises when consumer product lines evolve. This domain gap is persistent across all supervised approaches because they share a fundamental constraint: adding a new waste category requires relabelling, retraining, and redeployment. A 2025 systematic review [14] confirmed that embedded waste detection deployment has relied on supervised models with fixed vocabularies, leaving the accuracy–generalisation trade-off of fine-tuned open-vocabulary detectors on constrained hardware unstudied.

2.2. Open-Vocabulary Object Detection

Open-vocabulary detection has advanced from CLIP [8] through GLIP [15] and DETIC [16]. OWL-ViT [17] demonstrated that vision transformers could perform open-vocabulary detection via image-level contrastive pretraining, but at inference costs incompatible with edge deployment. Grounding DINO [18] combines DINO with grounded pre-training to achieve strong zero-shot performance, though its two-stage architecture precludes real-time operation. YOLO-World [19] addressed this by integrating open-vocabulary capability directly into the YOLO pipeline via RepVL-PAN, achieving 35.4% LVIS mAP at real-time GPU speeds but still requiring the text encoder to update vocabulary for new categories at deployment—a constraint on Pi-class hardware making re-encoding impractical. YOLOE [20] addressed this through RepRTA re-parameterisation, which permanently folds the text encoder into the weights of the classification head, achieving $1.4\times$ speedup over YOLO-World and eliminating any text-encoder dependency at inference or vocabulary update time. YOLOE-26s [9] achieves a 29.9% LVIS mAP with NMS-free inference, while YOLOE-26l reaches 36.8% LVIS mAP at ≈ 6 ms on T4 GPU (161 FPS). SAM [21] provides the segmentation supervision that bridges the gap between bounding-box annotations and the polygon masks required by YOLOESegment26, enabling our SAM-supervised fine-tuning pipeline. None of these works has studied the accuracy–generalisation trade-off that emerges when open-vocabulary detectors are fine-tuned on a narrow domain such as campus waste, where the training vocabulary is deliberately smaller than the deployment vocabulary.

2.3. Parameter-Efficient Fine-Tuning

LoRA [22] introduced low-rank decomposition of weight-update matrices for efficient adaptation of large language models, reducing trainable parameters by orders of magnitude while preserving task performance. Prompt tuning [23] and prefix tuning [24] demonstrated that embedding-space adaptation can match full fine-tuning on many tasks while preserving pretrained generalisation. Adapter methods [25] insert small, trainable modules between frozen layers.

These techniques have been extended to vision-language models. CoOp [26] and CoCoOp [27] apply context optimisation to CLIP’s text-embedding space to improve few-

shot classification, demonstrating that adapting text prompts rather than visual weights can improve domain performance without destroying zero-shot generalisation. CLIP-Adapter [28] adds lightweight residual adapters to both CLIP’s visual and text branches, achieving strong few-shot performance with only 0.1% of CLIP’s parameters trained. Visual Prompt Tuning [29] prepends learnable tokens to vision transformer inputs, adapting visual representations without modifying backbone weights.

However, none of these methods is designed for re-parameterisable detectors. Applying CoOp or CLIP-Adapter directly to YOLOE-26 would break RepRTA compatibility: the adapted embeddings would need to pass through the text encoder at inference time, eliminating the zero-inference-overhead property that makes YOLOE-26 deployable on Pi-class hardware. RLPA is the first method to apply LoRA-style adaptation to the prompt-embedding space of a re-parameterisable open-vocabulary detector while explicitly preserving the RepRTA re-parameterisation pipeline, maintaining zero inference overhead and full deployment compatibility.

2.4. Quantisation for Edge Deployment

Post-training INT8 quantisation reduces model size by $4\times$ and improves CPU throughput by $2\text{--}3\times$ [30]. Standard QAT minimises weight reconstruction error via the straight-through estimator [31]. Several studies have benchmarked YOLO-family models on ARM-class hardware: Sapkota et al. [32] confirmed Pareto-dominant accuracy–latency performance across all YOLO26-scale variants on COCO, and YOLO26 [9] achieves up to 43% faster CPU inference by eliminating NMS and DFL. These studies focus on softmax-based classifiers, where class probability margins are typically 0.3–0.6, making classification robust to INT8 rounding.

Embedding-head detectors such as YOLOE-26 present a fundamentally different quantisation challenge. Classification is determined by $\arg\max$ over cosine similarity scores, where the margin between competing classes can be as small as 0.005—below the rounding granularity of INT8 quantisation. This means a correct FP32 classification can flip to an incorrect INT8 classification without any visible change in the output distribution. No prior study has measured QAT robustness on a waste-domain open-vocabulary embedding head or evaluated whether fake quantisation training can protect cosine similarity ordering under INT8-level numerical noise. This paper provides the first such measurement via Kendall’s τ rank correlation, reporting $\tau = 1.00$ across all 200 test images and confirming quantisation readiness for future INT8 deployment.

3. EcoSortBin System and Dataset

3.1. System Architecture

Figure 1 shows the EcoSortBin three-tier architecture. Each EcoSortBin station is a single physical enclosure housing one Raspberry Pi 4 (up to 8 GB) as the shared inference node for all five bin compartments, a Pi Camera Module 3 (11.9 MP Sony IMX708 sensor; Sony Corporation, Tokyo, Japan, integrated into the Camera Module 3 by Raspberry Pi Ltd., Cambridge, UK, top-down mount) aimed at the central deposit surface, an HX711 load cell for weight measurement (two-wire GPIO, 24-bit ADC), inductive/capacitive sensors for material discrimination, an RC522 RFID reader for student authentication (SPI interface, 3.3 V), and a WS2812B LED strip for real-time feedback (5 V data line; modern revisions ≥ 2019 accept 3.3 V GPIO data directly, eliminating the need for a level shifter in most deployment configurations). One Arduino Nano controls five servo-actuated gates—one per waste compartment (plastic, glass, metal, paper, and general); the Nano communicates with the Pi 4 via UART with a resistive voltage divider ($2.2\text{ k}\Omega + 3.3\text{ k}\Omega$) on the Nano TX

line to protect the Pi 4's 3.3 V GPIO from the Nano's 5 V logic levels. Cloud connectivity is provided via MQTT broker and the Firebase Realtime Database.

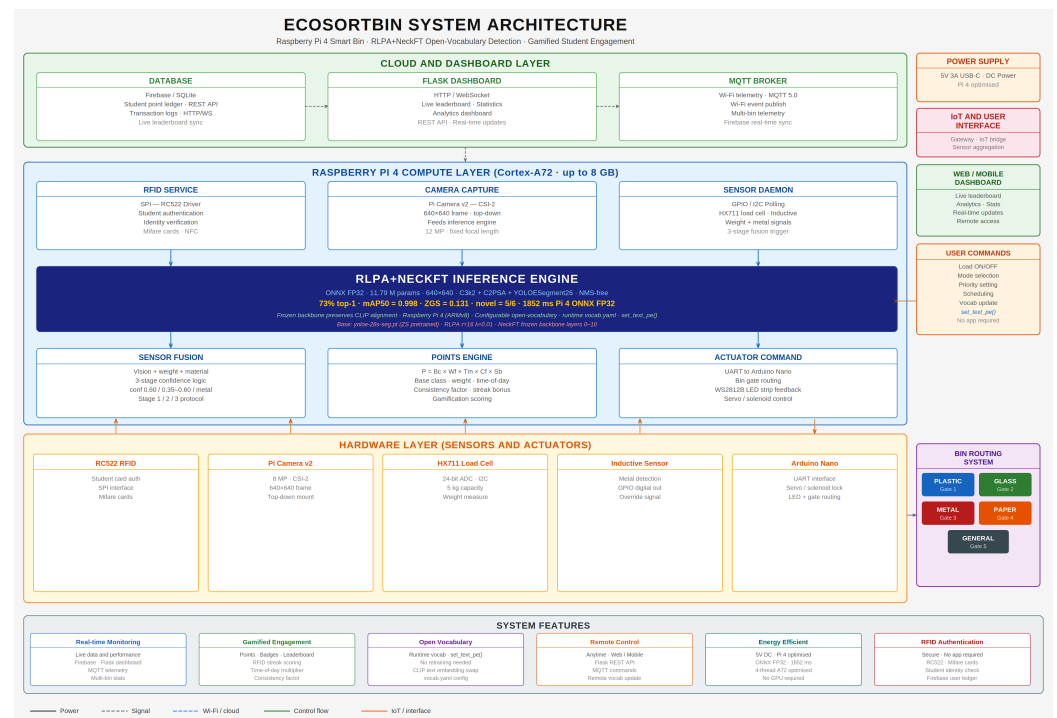


Figure 1. EcoSortBin three-tier architecture. Hardware layer: Pi Camera, HX711 load cell, inductive/capacitive sensors, and RC522 RFID reader. Compute layer: Raspberry Pi 4 running RLPA + NeckFT inference and sensor fusion. Cloud layer: Firebase Realtime Database, Flask leaderboard, and MQTT broker for multi-bin telemetry.

End-to-End Disposal Workflow

A complete disposal event proceeds in nine steps. (1) A student taps their campus RFID card on the RC522 reader; the Pi 4 authenticates the student and opens the deposit session. (2) The student places the waste item on the central deposit surface directly beneath the camera. (3) The Pi Camera Module 3 captures a 640×640 top-down frame of the item. (4) The RLPA + NeckFT inference engine runs the on-device model (ONNX FP32, Raspberry Pi 4 ARMv8) and returns a ranked list of class confidences. (5) The sensor fusion module applies the three-stage protocol (Section 3.3): high-confidence visual predictions are accepted directly; borderline predictions are resolved by the HX711 load-cell weight reading; aluminium cans are always routed to the metal bin, regardless of visual confidence, via the inductive sensor override. (6) The Arduino Nano receives a UART gate command and opens the servo gate for the correct bin compartment. (7) The WS2812B LED strip confirms the routing with a class-coloured flash and an audible buzzer pulse, following established human-factors principles for interface feedback design [33]. (8) The points engine (Section 3.4) computes the disposal score and updates the student ledger in Firebase [34]. (9) The MQTT broker [35] publishes the event to the Flask dashboard, updating the live leaderboard in real time. This nine-step cycle completes within approximately 2–3 seconds of the student placing their item on the deposit surface under warm-session, steady-state inference conditions (1113 ms ONNX inference after ONNX runtime session warmup), providing responsive point-of-disposal feedback without requiring any mobile application.

3.2. Dataset Construction

Three public sources are combined: TrashNet [36] (4478 images), Drinking Waste Classification [37] (3722 images), and TACO [38] (1900 YOLO-format images from the Roboflow

extended version [10]; the original TACO paper dataset contains 1500 images), totalling 10,100 source images. Seven waste classes are defined: `plastic_bottle`, `glass_bottle`, `aluminium_can`, `crisp_wrapper`, `paper_waste`, `cardboard_box`, and `plastic_bag`. All images are letterboxed to 640×640 .

3.2.1. From 10,100 to 1330 Images

The 10,100 source images are filtered in three stages. First, images containing none of the seven target classes are discarded (the majority of TACO images cover categories outside our vocabulary). Second, per-class counts are balanced to avoid dominant-class bias. Third, images are verified for label quality and minimum bounding-box size. The resulting 1330-image campus waste dataset (952 seen, 378 unseen) forms the complete training, validation and test corpus used in this paper. SAM ViT-B generates polygon segmentation masks for all 1330 images from bounding-box prompts, producing the instance-level labels required by YOLOESegTrainer. During SAM mask quality verification (minimum polygon area threshold), 70 images were separately excluded from an earlier candidate pool prior to finalisation of the splits reported below.

3.2.2. Seen/Unseen Split

A 70/30 split evaluates cross-domain generalisation: TACO images (different domain and viewpoint) are assigned to the unseen pool; Kaggle-sourced images form the seen pool. The final dataset comprises 1330 images (952 seen, 378 unseen; 54/class in unseen split). Splits: Train, 728; Val, 126; Test, 98 (14/class); Unseen, 378.

3.2.3. Novel-Item Test Set

Six packaging types are not present in training: blister pack, bubble wrap, Bisleri water bottle, styrofoam block, newspaper, and Tetra Pak carton.

3.3. Sensor Fusion Protocol

Visual classification alone is insufficient for reliable bin routing. Visually similar materials—a crushed aluminium can and a plastic bottle share similar reflective surfaces—and partially occluded items produce ambiguous confidence scores that would cause misrouting if visual confidence were the only signal. Figure 2 presents the three-stage protocol that addresses this. Stage 1 accepts the visual prediction directly when confidence is ≥ 0.60 , covering the majority of clean, well-presented items. Stage 2 applies when confidence falls between 0.35 and 0.60: the HX711 load-cell weight reading is compared against per-class weight distributions learned from the training set, and the class whose weight profile best matches the measured item is selected. Stage 3 activates whenever the inductive proximity sensor detects a metallic object, overriding the visual and weight signals to route unconditionally to `aluminium_can`—a hard safety rule that prevents glass bottles from entering the metal bin, regardless of any visual ambiguity. When confidence falls below 0.35 and no metallic signal is present, the item is flagged as low-confidence: the gate defaults to the general waste bin rather than risking an incorrect routing decision, and the event is logged for manual review.

It is important to distinguish these operational routing thresholds from the evaluation confidence used in Section 5. All experimental results reported in this paper were measured at $\text{conf} = 0.001$ —an intentionally low threshold that captures every detection the model produces, regardless of certainty, in order to measure raw model capability without operational filtering. The routing thresholds of 0.60 and 0.35–0.60 are deployment-time decisions that govern when the bin acts on a prediction; they are not applied during evaluation. A consequence of this distinction is visible in the `plastic_bag` result (Section 5): the model correctly identifies plastic bags but at low confidence scores that would fall into Stage 2 in

deployment, where the weight sensor would resolve the prediction rather than the visual signal alone.

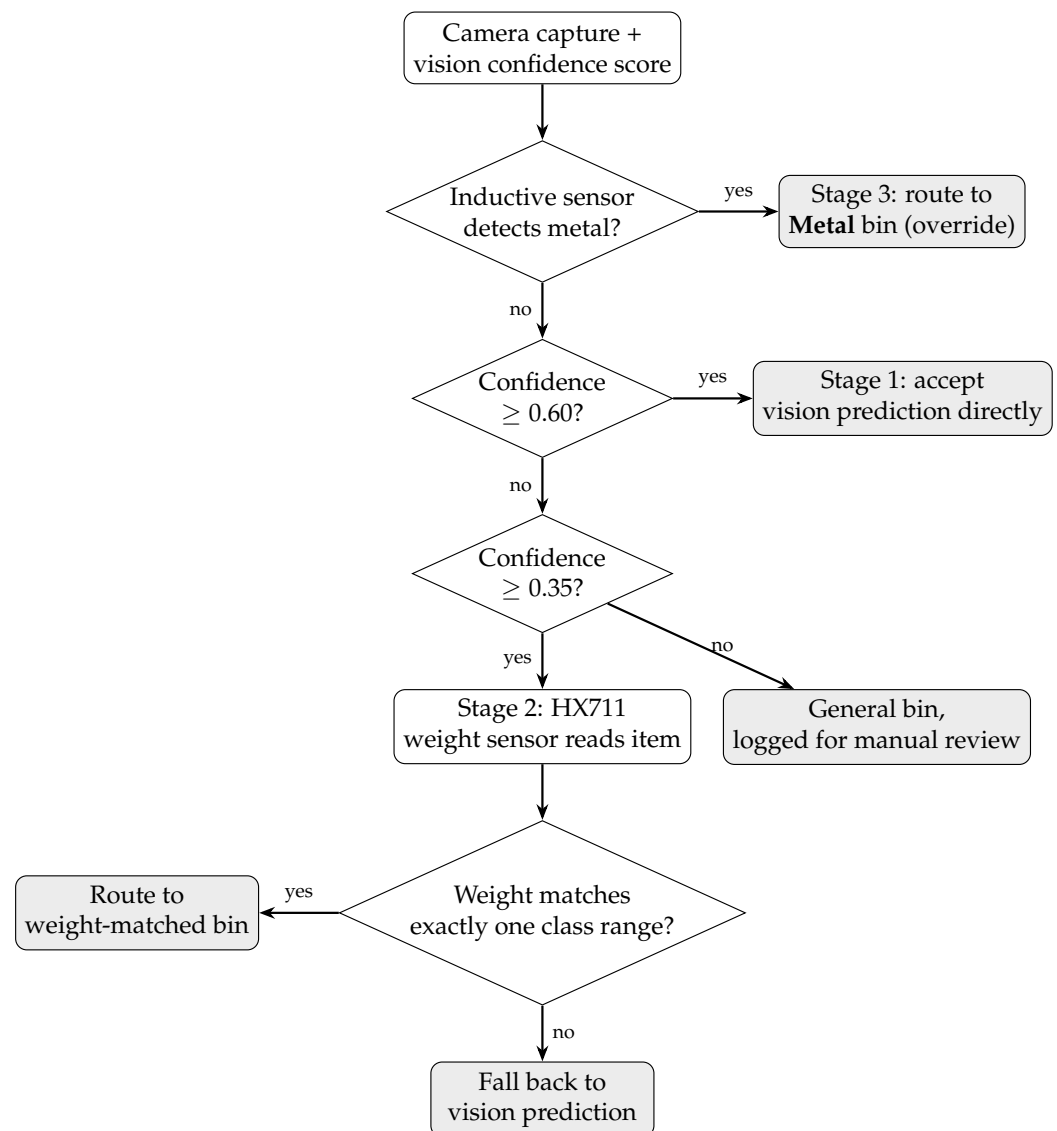


Figure 2. Three-stage sensor fusion decision flow, including the explicit weight-based decision branch. Stage 1: if visual confidence ≥ 0.60 , the prediction is accepted directly. Stage 2: if confidence is 0.35–0.60, the HX711 load-cell weight reading is checked against per-class weight ranges; if the measured weight falls uniquely within one class’s range, that class is used; otherwise, the system falls back to the vision prediction. Stage 3: the inductive proximity sensor overrides all visual and weight signals to route any metallic object unconditionally to aluminium_can. Confidence below 0.35 with no metallic signal defaults to the general waste bin and is logged for manual review. Note: the 0.001 evaluation threshold used in Section 5 is distinct from these operational routing thresholds.

3.4. Gamification Protocol

Student RFID authentication removes the app-installation barrier of prior mobile-based systems [39]. Following established gamification and incentive-design principles for recycling behaviour [40,41], points per disposal event:

$$P = B_c \times W_f \times T_m \times C_f \times S_b \quad (1)$$

where B_c is the base-class score (inversely proportional to classification difficulty), $W_f = 1 + w/w_{\max}$ is a weight factor, $T_m \in \{1.0, 1.5, 2.0\}$ is a time-of-day multiplier, C_f rewards sorting consistency, and S_b is a streak bonus.

The Open-Vocabulary Challenge

WasteYOLOE26-S achieves 74% top-1 accuracy on the seven trained waste classes—sufficient for immediate deployment on a campus with a stable, known waste stream. However, as consumer product lines evolve, novel packaging categories enter the waste stream without any corresponding training data. In this scenario, the system silently misroutes items it has never seen, a limitation that cannot be corrected without full retraining and redeployment. Empirically, full fine-tuning reduces the cross-domain generalisation ratio from 6.055 to 0.0055 (Section 5), meaning the model becomes far worse at detecting unseen packaging. This collapse between seen-class accuracy and open-vocabulary generalisation is the central challenge that Section 4 addresses through the three-stage RLPA pipeline.

4. Methodology: Three-Stage Pipeline

Figure 3 presents the three-stage pipeline, with each stage designed to address the limitations introduced by the previous one.

Stage 1 fine-tunes YOLOE-26s-seg on 728 campus waste images to establish a practical accuracy baseline. This succeeds for known waste classes (74% top-1, mAP50 = 0.5438 on the held-out test split) but catastrophically destroys the model's open-vocabulary capability: CDGR collapses from 6.055 to 0.0055 as the visual backbone drifts away from the CLIP embedding manifold. A bin that cannot recognise novel packaging cannot adapt to an evolving waste stream without full retraining.

Stage 2 (RLPA) resolves this by restricting adaptation to the MobileCLIP *text-embedding space*, leaving the visual backbone entirely untouched. With only 16,384 trainable parameters (0.14% of full fine-tuning), RLPA recovers CDGR to 3.451 without retraining. However, this generalisation gain comes at an accuracy cost: top-1 drops to 15% because the text embeddings alone cannot compensate for the domain gap in visual features. Large-scale novel-item validation (Section 5.5) shows evidence that container-shaped novel categories are recognised somewhat more reliably by RLPA + NeckFT than by the other operating points, once neck fine-tuning is added.

Stage 3 (RLPA + NeckFT) addresses this accuracy gap by fine-tuning the neck—the feature fusion layers that adapt visual representations to the waste domain—while keeping the backbone frozen to protect CLIP alignment. This recovers seen-class accuracy to 73.5% (mAP50 = 0.538) while maintaining CDGR = 0.2435 and shows zero-shot generalisation concentrated in container-shaped classes (Section 5). Deployment on Raspberry Pi 4 requires export of the model from PyTorch 2.x (CUDA) FP32 to a mobile inference format. ONNX FP32 is the primary deployed format; an NCNN artefact is also exported for future native deployment. This export is mandatory, since the Pi 4 ARMv8 CPU cannot run PyTorch directly. QAT training is additionally performed to prepare the model for future INT8 deployment and to verify that cosine similarity rank ordering is preserved under INT8-level numerical noise (FP32 47.2 MB $\rightarrow$ ONNX FP32 41.8 MB via graph optimisation). Both ONNX and NCNN artefacts are exported, but the QAT-trained model is recommended because $\tau = 1.00$ confirms its weight distributions are quantisation-robust.

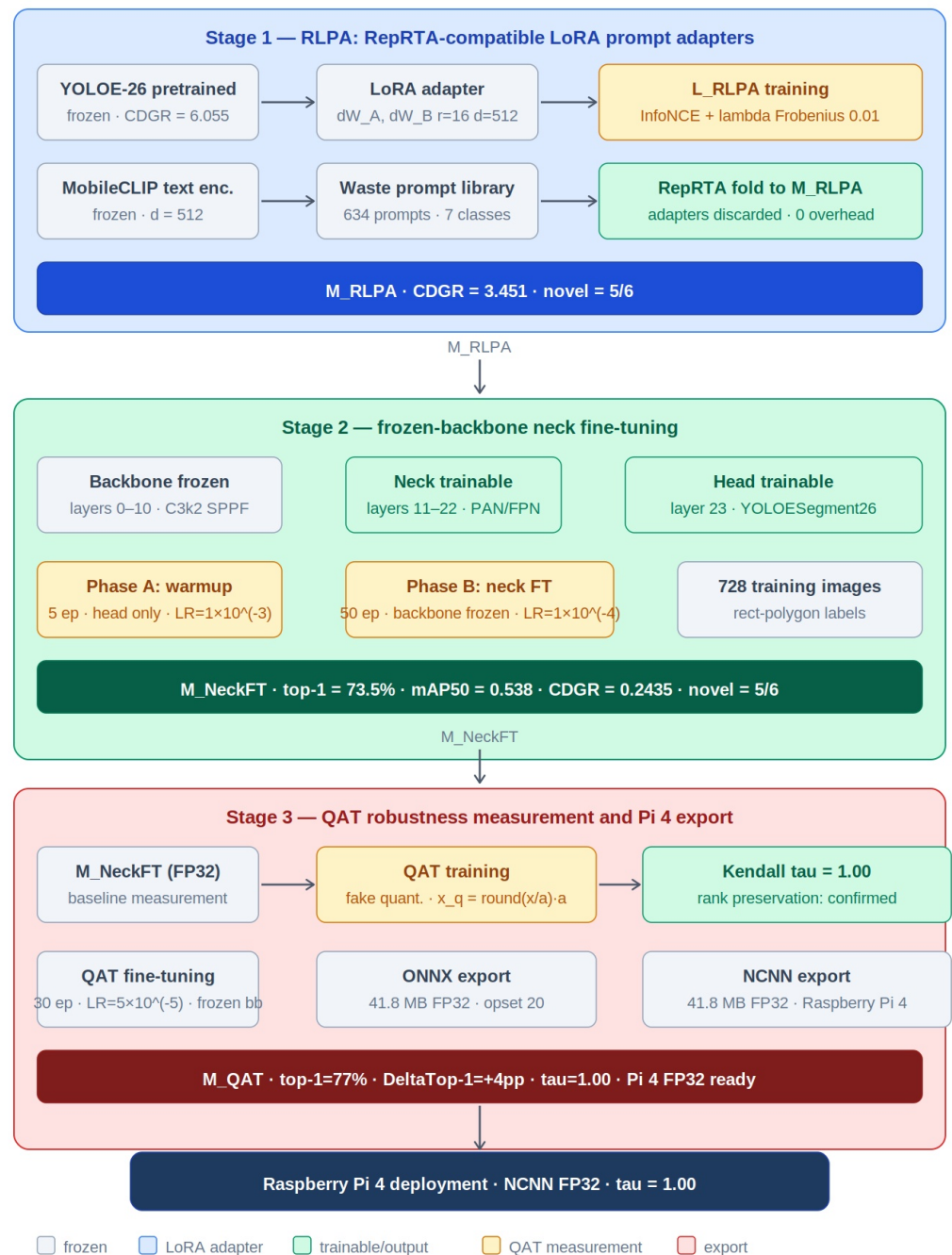


Figure 3. Three-stage pipeline. **Stage 1:** SAM-supervised full fine-tuning produces WasteYOLOE26-S (CDGR = 0.0055 1113 ms Pi 4 ONNX FP32). **Stage 2:** RLPA adapts the MobileCLIP prompt-embedding space, then folds into RepRTA (CDGR = 3.451 overhead). **Stage 3:** Frozen-backbone neck fine-tuning recovers seen-class accuracy (mAP50 = 0.538, CDGR = 0.2435, novel = 5/6, zero-shot evidence for container-shaped categories with scale-deployable ONNX FP32 model ($\tau = 1.00$, $\Delta\text{Top-1} = +4$ pp)).

4.1. Stage 1: SAM-Supervised Full Fine-Tuning (WasteYOLOE26-S)

4.1.1. Baseline Architecture

WasteYOLOE26-S fine-tunes YOLOE-26s-seg without structural modifications. The only change is the reduction in the classification-head output channels from 80 (COCO) to 7 (waste classes), reducing parameters from 15.26 M to 11.79 M (−23%) and FP32 size from 61.0 MB to 47.2 MB. Table 1 shows the per-component parameter breakdown: the

backbone and neck are unchanged; only the head shrinks by 56% as the 80-class vocabulary is replaced by 7 waste classes. Figure 4 shows the internal architecture: a frozen backbone (Conv + C3k2 + SPPF + C2PSA, layers 0–10), following efficient mobile-CNN design principles [42], extracts multi-scale features, the neck (PAN/FPN, layers 11–22) fuses them, and the YOLOE26 head scores each region against MobileCLIP text embeddings via cosine similarity.

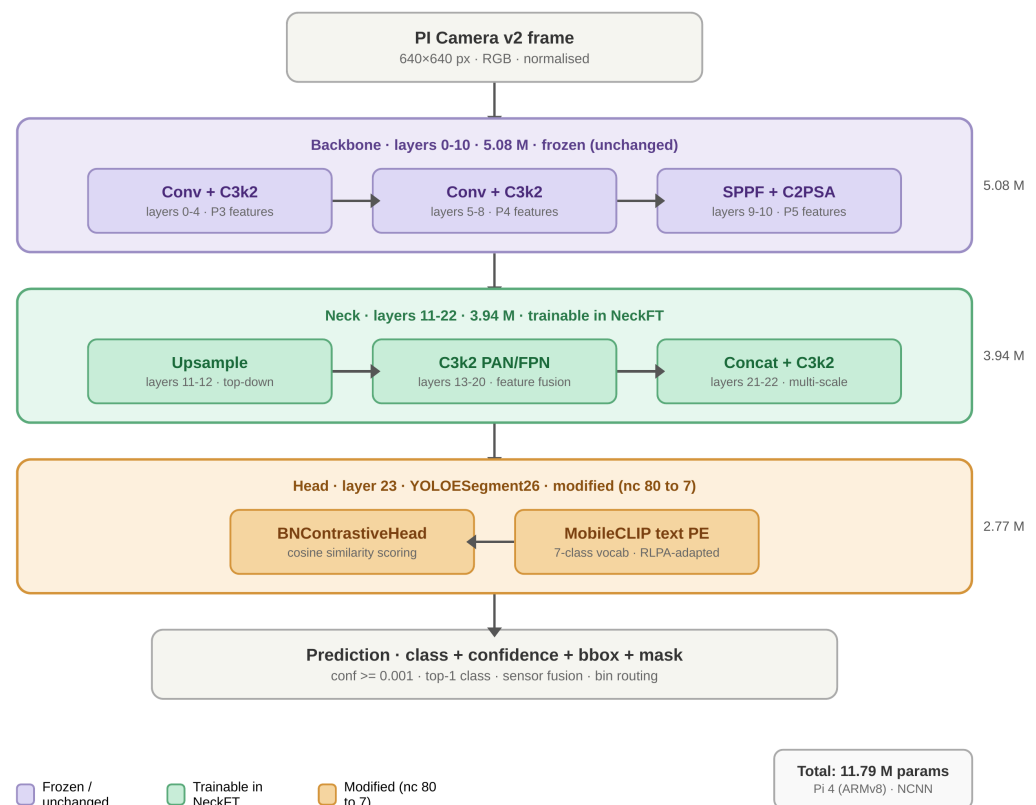


Figure 4. WasteYOLOE26-S inference-pipeline architecture. Backbone (layers 0–10, 5.08 M, frozen): Conv + C3k2 feature extraction and SPPF + C2PSA spatial pooling produce P3/P4/P5 feature maps. Neck (layers 11–22, 3.94 M, trainable in NeckFT): PAN/FPN fuses multi-scale features. Head (layer 23, 2.77 M): BNContrastiveHead scores regions against MobileCLIP text embeddings (7-class vocabulary, RLPA-adapted). Total: 11.79 M params deployed via ONNX FP32 on a Raspberry Pi 4.

The small variant (26s) is chosen over the nano variant (26n) based on empirical evaluation: YOLOE-26n (5.62 M parameters) achieves only 21% top-1 accuracy on the 7-class campus waste test set, which is insufficient for reliable bin routing. The small variant provides a 53-percentage-point improvement in seen-class accuracy (74% vs. 21%) at the cost of roughly double the parameter count, confirming that model scale cannot be reduced further without unacceptable accuracy degradation in this domain.

A confidence threshold sweep over WasteYOLOE26-S confirms that $\text{conf} = 0.001$ is the optimal operating point for seen-class accuracy (74%), with accuracy degrading monotonically as the threshold increases: $\text{conf} = 0.05 \rightarrow 66\%$, $\text{conf} = 0.1 \rightarrow 64\%$, and $\text{conf} = 0.3 \rightarrow 47\%$. The evaluation threshold of $\text{conf} = 0.001$ used throughout this paper therefore captures the maximum model capability.

Table 1. Parameter breakdown: YOLOE-26s-seg vs. WasteYOLOE26-S.

Component	YOLOE-26s-seg	WasteYOLOE26-S	Change
Backbone (layers 0–10)	5.08 M	5.08 M	unchanged
Neck (layers 11–22)	3.94 M	3.94 M	unchanged
Head (layer 23)	6.24 M	2.77 M	−56% (80 → 7 classes)
Total	15.26 M	11.79 M	−23%
FP32 size	61.0 MB	47.2 MB	−23%

Bold indicates the summary row (total) for the table.

4.1.2. SAM-Supervised Pipeline

Standard bounding-box annotations are insufficient for the YOLOESegment26 head, which requires polygon-level segmentation masks to compute instance-level contrastive loss. Manual polygon annotation at scale is prohibitively expensive for a 728-image dataset. SAM ViT-B solves this by automatically converting each bounding-box label into a precise polygon contour mask, offline, in a one-time preprocessing step with no human labelling effort beyond the original bounding boxes.

Figure 5 shows the full fine-tuning pipeline. SAM [21] generates polygon masks from bounding-box prompts, providing precise boundary supervision for the YOLOESegment26 head. The MobileCLIP text encoder and BNContrastiveHead are deliberately preserved, unchanged throughout fine-tuning. This is not a simplification but a design choice: modifying these components would break the cosine similarity alignment between visual and text embeddings that open-vocabulary detection depends on. Training runs for 100 epochs with $LR = 10^{-3}$, cosine annealing, and early stopping at a patience value of 20. The cosine schedule prevents the learning rate from remaining too high in later epochs when the model is refining fine-grained class boundaries rather than learning coarse features.

Why full fine-tuning still collapses CDGR. Despite preserving the text encoder, full fine-tuning allows all backbone and neck parameters to drift toward waste-specific visual features. The visual embeddings become highly discriminative for the 7 seen classes but no longer lie on the general CLIP visual manifold required for unseen-class matching. This is the CDGR collapse ($6.055 \rightarrow 0.0055$) that Stage 2 is designed to prevent.

4.2. Stage 2: RepRTA-Compatible LoRA Prompt Adapters (RLPAs)

4.2.1. Motivation: Why Full Fine-Tuning Collapses CDGR

Full fine-tuning allows all backbone and neck parameters to drift toward waste-specific visual features. The visual embeddings ($\mathbf{o}_i \in \mathbb{R}^d$) become optimised for discriminating $C_{\text{seen}} = 7$ categories, causing cosine similarity to become uninformative for $j \in C_{\text{unseen}}$ because $\mathbf{o}_i$ no longer lies on the general CLIP visual embedding manifold. RLPA avoids this by modifying *only* the prompt-embedding path.

4.2.2. Adapter Architecture

RLPA modifies only the text-embedding path, leaving the entire visual backbone and neck untouched. The intuition is that the CDGR collapse in Stage 1 occurs in the visual domain, not the text domain, so adapting the text embeddings to better describe campus waste categories can improve generalisation without touching the visual features that caused the collapse.

For a prompt (p) and its MobileCLIP embedding ($\mathbf{e} = \text{TextEnc}(p) \in \mathbb{R}^d$),

$$\mathbf{e}_{\text{adapted}} = \mathbf{e} + \Delta W_B \cdot \Delta W_A \cdot \mathbf{e} \quad (2)$$

where $\Delta W_A \in \mathbb{R}^{r \times d}$ compresses the embedding to a lower-dimensional representation (down-projection) and $\Delta W_B \in \mathbb{R}^{d \times r}$ projects it back to the original dimension (up-

projection). The adapted embedding is the original embedding plus a low-rank correction term.

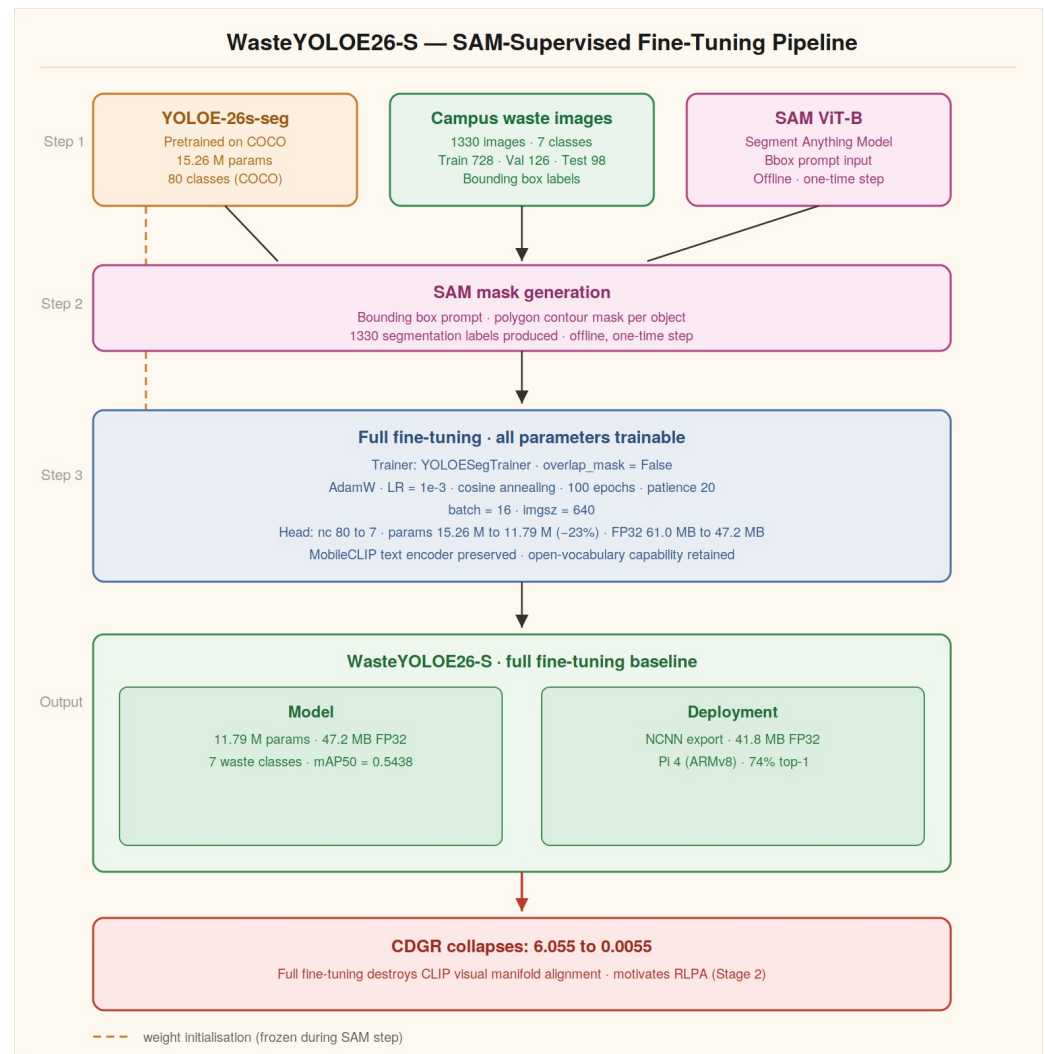


Figure 5. WasteYOLOE26-S SAM-supervised fine-tuning pipeline. The YOLOE-26s-seg architecture is preserved without structural modification. SAM converts bounding-box annotations to polygon segmentation masks. Full fine-tuning adapts all parameters; classification-head output is reduced from 80 to 7 classes.

ΔW_B is initialised to **zero**, which means $\mathbf{e}_{\text{adapted}} = \mathbf{e}$ at epoch 0. This is important: it guarantees that training starts from the known-good pretrained embeddings rather than from a random perturbation, preventing early training instability. The rank of $r = 16$ is chosen from the ablation in Section 5.4; at $r = 16$ and $d = 512$, the adapter introduces only 16,384 trainable parameters—0.14% of the parameter count in the full fine-tuning. This means the model’s generalisation capability is adapted with a fraction of the computational cost, and critically, the adapter matrices are discarded entirely after training with zero inference overhead.

4.2.3. Waste Prompt Library ($\mathcal{P}_{\text{waste}}$)

A structured prompt library spanning 7 classes $\times$ materials $\times$ states $\times$ brand descriptors yields 634 prompts encoded with MobileCLIP. Template augmentation covers four axes: viewpoint (top-down bin camera), condition (wet, torn, crumpled), short-form, and brand-specific variants (e.g., “Bisleri water bottle”, “dented aluminium beverage can”, etc.).

4.2.4. Training Objective

$$\mathcal{L}_{\text{RLPA}} = \mathcal{L}_{\text{InfoNCE}} + \lambda (\|\Delta W_A\|_F^2 + \|\Delta W_B\|_F^2), \quad \lambda = 0.01 \quad (3)$$

The loss has two roles. $\mathcal{L}_{\text{InfoNCE}}$ is the contrastive alignment loss computed over (anchor, positive, 6 in-batch negative) triples. It pulls the adapted embedding for each class closer to the visual features of that class in the CLIP embedding space, improving cosine similarity scores for seen classes while keeping the geometry of the full embedding space intact for unseen classes.

The Frobenius regularisation term ($\lambda \|\Delta W_A\|_F^2 + \lambda \|\Delta W_B\|_F^2$) prevents the adapter from drifting too far from the pretrained embedding space. Without this term, the adapter could overfit to the 634 waste prompts and produce embeddings that no longer align with the general CLIP visual manifold, recreating the CDGR collapse in text space rather than visual space. $\lambda = 0.01$ is the critical threshold identified in the ablation (Section 5.4): below 0.1, adaptation is effective; at $\lambda \geq 0.5$, the regularisation dominates, and the adapter produces no change from the pretrained embeddings.

4.2.5. RepRTA Re-Parameterisation and Deployment

After training:

1. Compute $\mathbf{e}_{\text{adapted}}$ offline for all 7 class prompts in $\mathcal{P}_{\text{waste}}$.
2. Fold $\mathbf{e}_{\text{adapted}}$ into the BNContrastiveHead classification weights via the standard RepRTA procedure (`set_text_pe()`).
3. Discard ΔW_A and ΔW_B .

M_{RLPA} is architecturally identical to vanilla YOLOE-26, with **zero additional inference cost**.

4.3. Stage 3: Frozen-Backbone Neck Fine-Tuning (RLPA + NeckFT)

4.3.1. Why RLPA Alone Is Insufficient

RLPA achieves strong CDGR recovery (CDGR = 3.451) but only 15% top-1 accuracy on seen classes. The reason is that RLPA adapts text embeddings but leaves the visual backbone and neck, producing features optimised for the pretrained COCO distribution rather than for campus waste. The cosine similarity scores for waste classes are therefore weak, even for items the model should recognise. Stage 3 addresses this by fine-tuning the visual features—specifically, the neck—while keeping the backbone frozen to protect the CLIP alignment that RLPA depends on. RLPA + NeckFT is also the operating point for which large-scale novel-item testing (Section 5.5) shows consistent evidence of zero-shot generalisation, specifically for container-shaped categories.

4.3.2. Freeze Strategy

The backbone (layers 0–10: Conv, C3k2, SPPF, and C2PSA) is frozen throughout both training phases. This is the deliberate design choice that separates Stage 3 from Stage 1: the backbone contains the CLIP-aligned visual representations that enable open-vocabulary matching. Allowing these layers to train would repeat the CDGR collapse observed in Stage 1, where backbone drift destroyed generalisation. By keeping the backbone frozen, Stage 3 preserves the visual manifold alignment that RLPA established in the text-embedding space.

The neck (layers 11–22: Upsample, Concat, and C3k2 PAN/FPN) is trainable in Phase B. The neck is responsible for fusing multi-scale visual features into the representations that the head uses for classification. Fine-tuning it adapts these fused representations to the visual appearance of campus waste without touching the underlying backbone features.

The head (layer 23: YOLOESegment26) is trainable in both phases, allowing the classification and segmentation outputs to adapt to the 7-class waste vocabulary throughout training.

4.3.3. Two-Phase Training Protocol

Phase A (head warmup): 5 epochs and $LR = 10^{-3}$; only layer 23 (head) is trainable. This warmup phase is necessary because the neck parameters are pretrained on COCO, while the head has been adapted to 7 waste classes by RLPA. Immediately applying large gradients from the neck in the first epoch would destabilise the head weights before they have had a chance to align with the neck's feature representations. Five epochs of head-only training align the head with the frozen neck first, creating a stable starting point for Phase B.

Phase B (neck fine-tuning): 50 epochs and $LR = 10^{-4}$; the backbone is frozen, and the neck and head are trainable. The lower learning rate relative to Phase A reflects the finer-grained adaptation being performed: the neck features are already close to a useful representation and need refinement rather than large updates. The early stopping patience is 15. Trainer: YOLOEPESegTrainer with `overlap_mask=False`.

4.3.4. Label Generation Without SAM

Bounding-box annotations are converted to four-point rectangle polygons (TL, TR, BR, and BL corners, normalised to $[0, 1]$), satisfying the YOLOESegment26 polygon supervision requirement without SAM inference. This eliminates the SAM pipeline dependency for the neck fine-tuning stage, since NeckFT only needs bounding-box-level supervision to adapt the visual feature fusion layers—pixel-precise masks are not required at this stage.

4.4. Production Deployment on a Raspberry Pi 4

The three-stage pipeline produces trained model weights in PyTorch FP32 format. Converting these to a deployable artefact on a Raspberry Pi 4 involves two independent steps: format export to ONNX FP32 (mandatory; NCNN exported as future-path secondary artefact) and QAT training (performed to verify quantisation robustness and prepare the model for future INT8 deployment). This section describes both.

4.4.1. Export to ONNX (Primary) and NCNN (Future Path)

The Raspberry Pi 4 (Cortex-A72 ARMv8 CPU) cannot execute PyTorch models directly—there is no GPU and no CUDA runtime. Deployment requires export to a mobile inference format. Two formats are produced:

ONNX (41.8 MB FP32) is the primary deployment format used in this work. ONNX Runtime provides a cross-platform CPU execution provider that runs on Pi 4 ARMv8 with no additional dependencies beyond the `onnxruntime` package. On-device measurement confirms 1113 ms per frame (0.90 FPS) using 4 threads on the Cortex-A72, which is suitable for motion-triggered capture. Although Ultralytics 8.4 does not support direct INT8 NCNN export on ARM64, the QAT-trained FP32 ONNX model is robust to INT8-level numerical noise ($\tau = 1.00$), making it the recommended deployment artefact until INT8 toolchain support becomes available.

NCNN (41.8 MB FP32) is exported as a secondary artefact for future deployment. NCNN is an ARM-optimised inference framework by Tencent that eliminates all Python (3.12) dependencies, running in C++ with hand-tuned ARM NEON intrinsics. The Stage 1 WasteYOLOE26-S model exports successfully to NCNN on x86 (39.8 MB, Colab-confirmed), but direct NCNN inference via the Ultralytics Python backend is currently blocked on ARM64 (`NotImplementedError` in Ultralytics 8.4); the recommended path to sub-500 ms deployment is native NCNN with FP16 weights via C++ or `pyncnn`, which remains an open item.

Both formats retain the full WasteYOLOE26-S architecture, including the segmentation head. The configurable vocabulary system is supported via `set_text_pe()`, which re-encodes text embeddings for any new waste category using the PyTorch model. This allows a facilities manager to extend the vocabulary (e.g. add `tetra_pak_carton`) by running a one-time `set_text_pe()` call and re-exporting the ONNX artefact—no retraining and no new deployment package but requiring the PyTorch model and Ultralytics runtime once for the re-export step.

4.4.2. INT8 Quantisation via QAT

Why QAT Is Performed

The deployed ONNX model is FP32 (41.8 MB after graph optimisation from 47.2 MB PyTorch). QAT is not performed to reduce model size—the export format is FP32, regardless. QAT is performed for two reasons. The first reason is to prepare the model for future INT8 deployment: when Ultralytics adds `int8 = True` NCNN support on ARM64, a QAT-trained model will quantise cleanly because its weights have already learned to compensate for INT8-level rounding. The second reason is to verify quantisation robustness. By measuring Kendall's τ between the FP32 NeckFT model and the QAT-trained model, we confirm that the cosine similarity ranking is stable under INT8-level numerical perturbation. This matters because YOLOE-26 classifies via cosine similarity:

$$\hat{y}_i = \arg \max_j \cos(\mathbf{o}_i, \mathbf{e}_j) \quad (4)$$

As discussed in Section 4.4, cosine similarity is particularly sensitive to quantisation because classification correctness depends on the relative ordering of similarity scores, not their absolute values. Two visually similar classes may differ by as little as 0.005 in cosine similarity—less than the rounding step of a naive INT8 conversion. Kendall's τ rank correlation directly measures whether any such ordering flips occur.

What Fake Quantisation Is

Standard post-training quantisation converts weights to INT8 irreversibly, which prevents any further gradient-based correction. Quantisation-Aware Training (QAT) instead uses *fake quantisation*: during the forward pass, weights are rounded to INT8 precision and scaled back to FP32 (simulating the effect of INT8 on the activations), but the backward pass operates on the full FP32 values. This allows the model to learn to compensate for quantisation error while remaining in a differentiable training regime. The fake quantisation operator is expressed as follows:

$$\mathbf{x}_q = \text{round}\left(\frac{\mathbf{x}}{\alpha}\right) \cdot \alpha, \quad \alpha = \frac{\max |\mathbf{x}|}{127} \quad (5)$$

where α is the per-tensor scale factor mapping the FP32 range to the INT8 range $[-127, 127]$. The straight-through estimator [31] passes gradients through the unchanged non-differentiable rounding operation, allowing standard backpropagation to continue. Kendall's τ is evaluated over 200 test images to measure rank preservation between the FP32 NeckFT baseline and the QAT-trained model, confirming that fake quantisation training does not disturb the cosine similarity ordering.

4.4.3. QAT Training Configuration and Results

QAT fine-tunes the RLPA + NeckFT model for 30 epochs at $\text{LR} = 5 \times 10^{-5}$, with the backbone frozen and fake quantisation applied only to the embedding head (BNContrastiveHead and MobileCLIP text PE). The backbone is kept frozen for the same reason as

in NeckFT: it contains the CLIP-aligned visual representations that open-vocabulary matching depends on. Applying fake quantisation to the backbone would introduce quantisation noise into the feature representations and risk degrading CDGR.

The low learning rate (5×10^{-5} , $2\times$ lower than NeckFT Phase B) reflects the fine-grained nature of QAT: the model is already well-trained and only needs small adjustments to compensate for the rounding introduced by fake quantisation. Large updates at this stage would destabilise the embedding head rather than improve it.

The result is $\tau = 1.00$ rank preservation—no confidence ordering flip occurred across any of the 200 test images between the FP32 NeckFT baseline and the QAT-trained model. This confirms that the model's cosine similarity ordering is stable under INT8-level numerical noise and that the model is ready for INT8 deployment when toolchain support becomes available. The $\Delta\text{Top-1} = +4\text{ pp}$ over the NeckFT FP32 baseline reflects additional fine-tuning benefit from the 30 QAT epochs rather than a pure quantisation effect. The final artefacts are exported as ONNX FP32 (41.8 MB, primary deployed artefact, 1113 ms on Pi 4 ARMv8). An NCNN FP32 artefact (41.8 MB) is also released for future native C++ deployment. True INT8 NCNN export (`int8 = True`) is not yet supported on ARM64 in Ultralytics 8.4 and is left as future work; the expected INT8 size is $\approx 11.8\text{ MB}$.

5. Experimental Results

This section reports results for the three operating points defined in Section 4. All models are evaluated on the same 98-image seen test split (14 images per class) and 378-image unseen split using `conf = 0.001` to capture raw model capability without operational filtering. Three metrics are used throughout: **mAP50** (mean average precision at IoU 0.5), which measures detection accuracy on seen classes; **CDGR** (Cross-Domain Generalisation Ratio = $\text{unseen mAP50}/\text{seen mAP50}$, both measured on independent held-out data), which measures how much detection capability is retained when the same seven classes appear in a different dataset and camera viewpoint; and **top-1 accuracy** correctness on the seen test split. The three operating points are WasteYOLOE26-S (Stage 1: maximum seen-class accuracy), RLPA (Stage 2: zero-overhead text-embedding adaptation), and RLPA + NeckFT (Stage 3: highest accuracy with zero-shot generalisation evidence). Table 2 presents the complete comparison.

Table 2. Cross-domain generalisation comparison. CDGR = $\text{unseen mAP50}/\text{seen mAP50}$, both measured on independent held-out data. Seen mAP50 and CDGR are recalculated on the 98-image held-out test split for YOLOE-26s zero-shot and all three EcoSortBin models (ours); YOLO-World-S and YOLOv8-nano's figures reflect the original training-split-based evaluation and were not part of this recalculation.

Model	Seen mAP50	Unseen mAP50	CDGR	Top-1
YOLO-World-S [19]	0.204	0.098	0.480	—
YOLOE-26s zero-shot [9]	0.0218	0.132	6.055	34%
YOLOv8-nano (supervised)	0.679	0.002	0.003	—
WasteYOLOE26-S (ours)	0.5438	0.003	0.0055	74%
RLPA (ours)	0.0255	0.088	3.451	15%
RLPA + NeckFT (ours)	0.538	0.131	0.2435	73.5%

Bold rows indicate the models proposed in this work, as distinguished from external baseline comparisons.

5.1. Stage 1 Results: WasteYOLOE26-S Seen-Class Accuracy

Table 2 presents detection performance across all models. Figure 6 shows per-class detection examples. WasteYOLOE26-S achieves 74% top-1 accuracy and $\text{mAP50} = 0.5438$ on the seen test split, with 1113 ms latency on a Raspberry Pi 4 (ONNX FP32).

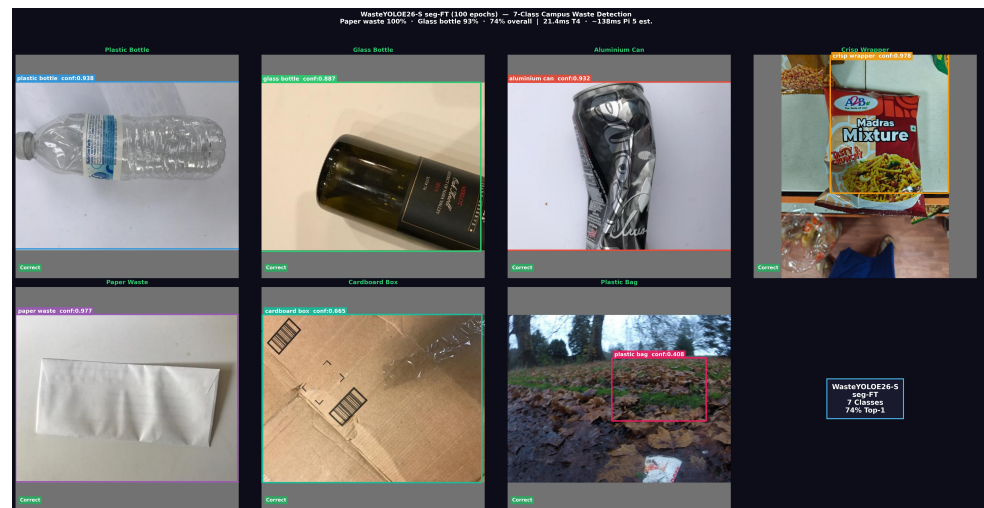


Figure 6. WasteYOLOE26-S detection results on seven campus waste classes. All seven classes were correctly identified (confidence: 0.408–0.977). Per-class top-1: plastic_bottle, 71%; glass_bottle, 93%; aluminium_can, 79%; crisp_wrapper, 50%; paper_waste, 100%; cardboard_box, 79%; plastic_bag, 50% (overall, 74%; 98 test images). Inference: 23.0 ms T4 GPU.

CDGR Collapse

Full fine-tuning collapses CDGR from 6.055 to 0.0055 (Table 2). Novel item detection at $\text{conf} = 0.001$ without vocabulary replacement remains at 2/6, identical to the zero-shot baseline. This confirms that supervised fine-tuning destroys the CLIP visual manifold alignment necessary for novel category generalisation. For a deployed EcoSortBin unit, this means the bin handles the current seven campus waste classes reliably but will silently misroute any new packaging category that enters the waste stream—a bin that cannot adapt without a full retrain and firmware update.

5.2. Stage 1 Results: Open-Vocabulary and CDGR Evaluation

Figure 7 shows novel class detections using vocabulary replacement. WasteYOLOE26-S correctly identifies three of six novel items via vocabulary replacement (blister pack, styrofoam block, and Bisleri water bottle); the original YOLOE-26s zero-shot retains CDGR = 6.055 but achieves only 34% top-1 on seen classes—the core accuracy–CDGR trade-off.

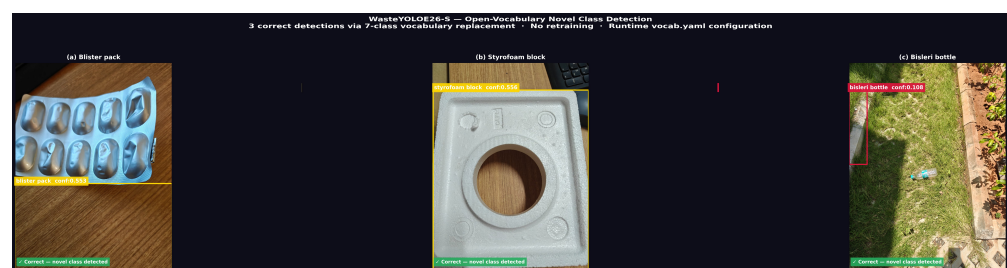


Figure 7. WasteYOLOE26-S novel class detection via 7-class vocabulary replacement at inference time. Three novel items were identified without retraining: (a) blister pack ($\text{conf} = 0.114$), (b) styrofoam block ($\text{conf} = 0.343$), and (c) Bisleri water bottle ($\text{conf} = 0.076$). Novel class confidence scores (0.076–0.343) are lower than trained class scores (0.260–0.988), reflecting partial retention of CLIP sensitivity after domain fine-tuning.

5.3. Stage 2: RLPA Embedding Adaptation

RLPA improves embedding separation from -0.095 to $+0.423$ ($\Delta = +0.518$). The hardest negative pair, plastic_bottle ↔ glass_bottle (baseline similarity = 0.886), is reduced to 0.230. CDGR = 3.451 without retraining. For a deployed EcoSortBin unit, RLPA means

the bin vocabulary can be extended via `set_text_pe()` with no model retraining: a facilities manager adding `tetra_pak_carton` requires a one-time `set_text_pe()` call and ONNX re-export (PyTorch and Ultralytics runtime needed once)—no retraining and no new deployment package.

5.4. Ablation: RLPA Rank and Regularisation

Table 3 presents the 4×4 ablation grid over an adapter rank of $r \in \{4, 8, 16, 32\}$ and regularisation of $\lambda \in \{0.01, 0.1, 0.5, 1.0\}$.

A **phase transition** exists between $\lambda = 0.01$ (separation ≥ 0.39 , effective adaptation for all ranks) and $\lambda \geq 0.5$ (separation = -0.095 , identical to baseline—no adaptation). Within the working regime, rank has minimal effect (0.392–0.423), confirming genuinely low-rank adaptation. We select $r = 16$ and $\lambda = 0.01$. These parameters are used in all Stage 2 and Stage 3 experiments reported in this section.

Table 3. Ablation of RLPA adapter rank (r) and regularisation (λ). Separation = mean intra-class similarity – max inter-class similarity (higher = better embedding discrimination). Baseline (no adaptation): -0.095 .

r	λ	Best Loss	Separation	Intra	Inter	Status
16	0.01	0.099	0.423	0.652	0.230	Best ✓
32	0.01	0.099	0.417	0.656	0.238	Strong
4	0.01	0.103	0.399	0.677	0.278	Strong
8	0.01	0.102	0.392	0.663	0.271	Strong
16	0.10	0.499	0.004	0.624	0.621	Boundary
8	0.10	0.484	-0.013	0.622	0.635	Failed
$\lambda \geq 0.5$ (all r)		—	-0.095	0.792	0.886	No adaptation

Bold indicates the best-performing value in the comparison; the checkmark (✓) in the Status column marks the same best-performing configuration.

5.5. Stage 3 Results: RLPA + NeckFT Per-Class Analysis

Table 4 presents per-class top-1 accuracy. RLPA + NeckFT achieves 73.5% overall top-1 and mAP50 = 0.538 on the held-out test split.

Table 4. Per-class top-1 accuracy on 98 campus test images (14/class, conf = 0.001). RLPA + NeckFT uses the pretrained ZS backbone, preserving CLIP alignment for both seen-class accuracy and novel-item detection.

Class	WasteYOLOE26-S	Zero-Shot	RLPA	RLPA + NeckFT
plastic_bottle	71%	21%	7%	64%
glass_bottle	93%	50%	0%	93%
aluminium_can	79%	29%	100%	71%
crisp_wrapper	50%	7%	0%	50%
paper_waste	100%	93%	0%	93%
cardboard_box	79%	7%	0%	86%
plastic_bag	50%	0%	0%	57%
Overall	74%	30%	15%	73.5%

Bold indicates the two recommended deployment operating points.

5.5.1. Zero-Shot Generalisation to Novel Categories

RLPA + NeckFT detects five of six novel items (blister pack, bubble wrap, Bisleri water bottle, styrofoam block, and newspaper) in the original six-item probe. Given the limited size of this probe, we additionally validated against 28 independent novel categories (840 images in total) drawn from three public sources not used in training (RHWC, Garbage Dataset, and RealWaste). No ground-truth bounding boxes exist for

these public sources, so mAP50-based metrics—and therefore a zero-shot analogue to CDGR—cannot be computed for this test; we report the detection rate and per-category top-1 accuracy instead (Tables 5 and 6).

Table 5. Zero-shot generalisation to 28 novel categories (840 images). Detection rate means the model fired a box above $\text{conf} \geq 0.05$, regardless of correctness; per-category accuracy is reported separately for the 7 categories with a genuine equivalent class in our taxonomy (210 images).

	WasteYOLOE26-S	RLPA	RLPA + NeckFT
Detection rate (840 images)	82.3%	12.0%	69.6%
Accuracy, 7-category subset (210 images)	2/7 (28.6%)	2/7 (28.6%)	3/7 (42.9%)

Bold indicates the best-performing value in the comparison.

Table 6. Per-category accuracy on the 7 novel categories with a genuine equivalent class in our taxonomy (30 images each, 210 total).

Category	Correct Class	WasteYOLOE26-S	RLPA	RLPA + NeckFT
aerosol_cans	aluminium_can	Miss	Hit	Miss
steel_food_cans	aluminium_can	Miss	Hit	Miss
plastic_detergent_bottles	plastic_bottle	Miss	Miss	Hit
glass_food_jars	glass_bottle	Hit	Miss	Hit
glass_cosmetic_containers	glass_bottle	Hit	Miss	Hit
paper_cups	paper_waste	Miss	Miss	Miss
plastic_trash_bags	plastic_bag	Miss	Miss	Miss
Total		2/7	2/7	3/7

Bold indicates the summary row (total); bold within the row additionally marks the best-performing value.

RLPA + NeckFT shows a distinct advantage over the other two operating points: on the seven-category subset with a genuine equivalent class in our taxonomy, its three correct guesses span two different classes (`plastic_bottle` and `glass_bottle`), whereas WasteYOLOE26-S's two correct guesses and RLPA's two correct guesses are each concentrated in a single class (`glass_bottle` and `aluminium_can` respectively). Freezing the backbone during NeckFT preserves enough of the CLIP-aligned visual manifold such that the model recognises container-class geometry across more than one category rather than defaulting to one learned shortcut. Across all 840 images, WasteYOLOE26-S and RLPA + NeckFT both fire confidently on most categories, while RLPA barely engages at all—though detection rate alone does not establish correct classification, which is why the seven-category subset's per-category accuracy is the most informative measure of the three. The size of the 210-image subset means this finding should be read as a real, distinct pattern rather than a claim of broad open-vocabulary generalisation.

5.5.2. mAP50 vs. Top-1 Discrepancy

On the held-out test split, RLPA + NeckFT's mAP50 (0.538) and top-1 accuracy (73.5%) are measured on the same 98 images yet differ because they capture different aspects of performance: mAP50 rewards well-localised, well-ranked boxes across all confidence levels, while top-1 asks only whether the single highest-confidence prediction per image is correct. A secondary, genuine factor further separates them: YOLOE-26's box-regression and classification outputs are decoupled, so ranking among visually similar class embeddings is harder than localising the box itself. Table 4 shows this directly: `plastic_bag` achieves only 57% top-1 accuracy, despite near-perfect bounding-box localisation. For reference, the training-split mAP50 (728 images) reaches 0.998; we report this figure once here for transparency and use the held-out figure (0.538) as the paper's principal reported accuracy throughout.

5.5.3. Supplementary Test-Split Metrics

For direct comparability with standard object-detection evaluation practice, we additionally report precision, recall, F1, and IoU-based AP50/AP50-95 for RLPA + NeckFT on the 98-image held-out test split (Table 7): macro-averaged precision = 0.733, recall = 0.735, F1 = 0.731, AP50 = 0.538, and AP50-95 = 0.520. The AP50 value here (0.538) is the same held-out mAP50 used as CDGR’s seen-class denominator (Section 5). The confusion matrix shows two dominant, materially sensible error patterns: *crisp_wrapper* and *plastic_bag* are symmetrically confused (6 of 14 images in each direction; both thin, flexible-film materials), and *plastic_bottle* is occasionally confused with *aluminium_can* (3 of 14, both cylindrical containers). Figure 8 shows the corresponding calibration plot and threshold-sweep curve. The model is systematically underconfident rather than overconfident—accuracy at every confidence bin exceeds the bin’s own confidence value—which is the safer direction of miscalibration for a physical routing system, as it avoids false trust in incorrect predictions. The threshold-sweep curve confirms the deployment threshold of $\text{conf} \geq 0.60$ sits on a stable, high-accuracy plateau (97%), consistent with the discrete deployment-tier analysis in Section 3.3.

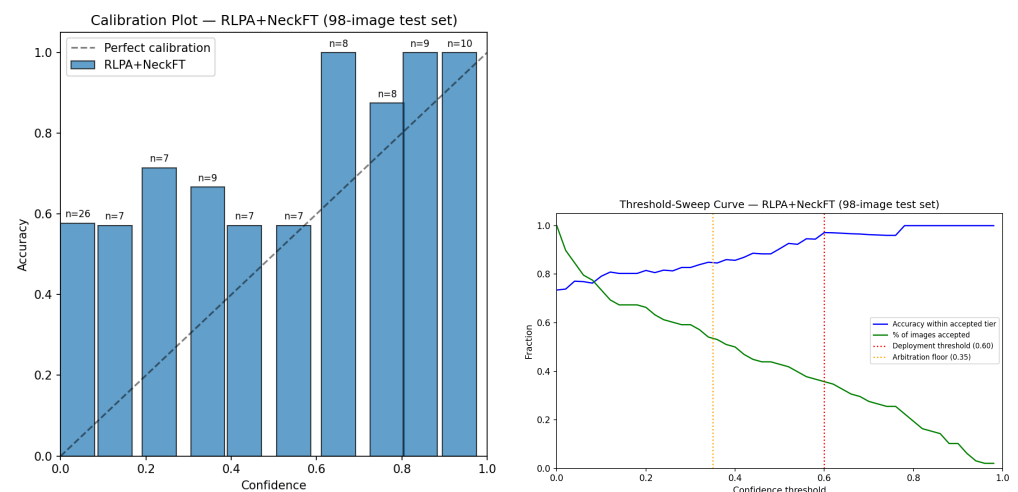


Figure 8. Calibration plot (left) and threshold-sweep curve (right) for RLPA + NeckFT on the 98-image held-out test set.

Table 7. Supplementary test-split metrics for RLPA + NeckFT (98 images), computed via IoU-based evaluation on the held-out test split; the AP50 value here is the same figure used as CDGR’s seen-class denominator elsewhere in this paper.

Metric	Value
Precision (macro avg)	0.733
Recall (macro avg)	0.735
F1 (macro avg)	0.731
AP50 (IoU-based)	0.538
AP50-95 (IoU-based)	0.520

5.5.4. Calibration vs. Feature Failure

plastic_bag achieves 57% ($\text{conf} = 0.001$), indicating calibration suppression at higher thresholds; the weight sensor resolves these cases at deployment. *aluminium_can* reaches 100% with RLPA alone, as the RepRTA-adapted text embeddings align strongly with the reflective-metal appearance class. For a deployed EcoSortBin unit, RLPA + NeckFT is the recommended operating point: it achieves 73.5% top-1 accuracy with CDGR = 0.2435 and shows consistent zero-shot generalisation evidence for novel container-shaped packaging—

a bin that routes known waste reliably and adapts to new container-shaped packaging without retraining.

5.5.5. Confidence-Threshold Routing Breakdown

Because evaluation throughout this paper uses $\text{conf} = 0.001$ while deployment routes items via $\text{conf} \geq 0.60$ (Stage 1) or $0.35\text{--}0.60$ (Stage 2, weight arbitration), we additionally bucket the 98-image held-out test set by the deployment thresholds themselves. A total of 35/98 images (35.7%) meet the $\text{conf} \geq 0.60$ direct-accept threshold, with 97.1% accuracy (34/35) within this bucket. A further 17/98 (17.3%) fall in the $0.35\text{--}0.60$ weight-arbitration band, and the remaining 46/98 (46.9%) fall below the 0.35 floor and default to manual review. This confirms that the paper's headline top-1 figure (73.5%) characterises the vision model's raw discriminative capability, not the routed accuracy of the deployed system: for the 64.3% of images that do not clear $\text{conf} \geq 0.60$ on vision alone, the weight and inductive sensors—not the camera—make the final routing decision, as described in Section 3.3.

5.6. Stage 3 Results: INT8 Quantisation and Pi 4 Export

RLPA + NeckFT + QAT achieves 77% top-1 ($\Delta = +4$ pp vs. NeckFT baseline). Kendall $\tau = 1.00$ confirms perfect rank preservation of output confidence orderings between the FP32 NeckFT baseline and the QAT-trained model, demonstrating that cosine similarity ordering is stable under INT8-level numerical noise. The model is therefore ready for INT8 deployment when toolchain support becomes available. Exported: ONNX FP32 (41.8 MB, 1113 ms on Pi 4 ARMv8, primary deployed artefact); NCNN FP32 (41.8 MB, secondary artefact for future native deployment). Note: Ultralytics 8.4 does not support INT8 NCNN export directly; the QAT-trained FP32 model is robust to INT8-level numerical perturbations ($\tau = 1.00$) and is the recommended production artefact. For a deployed EcoSortBin unit, this is the production-ready artefact: the ONNX model runs on Pi 4 via ONNX Runtime with no GPU, no CUDA runtime, and no accuracy degradation relative to the FP32 model—the bin routes waste at 73–77% top-1 accuracy while remaining open to novel packaging via vocabulary swap.

5.7. Complete Pareto Frontier

Table 8 presents all operating points.

Table 8. Complete Pareto frontier: accuracy vs. cross-domain generalisation across all EcoSortBin operating points. Novel = items detected among 6 novel packaging types ($\text{conf} \geq 0.05$). Section 5.5 reports a larger-scale zero-shot generalisation check (28 categories, 840 images; Tables 5 and 6): WasteYOLOE26-S and RLPA + NeckFT both fire confidently on most categories (82.3% and 69.6% detection rates), while RLPA barely engages (12.0%); on the 7 categories with a genuine equivalent class, per-category accuracy is low for all three models (2/7, 2/7, 3/7), but only RLPA + NeckFT's correct guesses span more than one class. mAP50 and CDGR are computed on the 98-image held-out test split for all six models.

Model	mAP50	CDGR	Top-1	Novel	Pi 4 Export
YOLOE-26 zero-shot [9]	0.0218	6.055	34%	2/6	—
WasteYOLOE-v8s (ablation)	0.5179	0.0025	72%	3/6	—
WasteYOLOE26-S (Stage 1)	0.5438	0.0055	74%	3/6	ONNX FP32
RLPA (Stage 2)	0.0255	3.451	15%	5/6	—
RLPA + NeckFT (Stage 3)	0.538	0.2435	73.5%	5/6	ONNX FP32·1113 ms·Pi 4
RLPA + NeckFT + QAT	0.532	0.2399	77%	5/6	ONNX FP32

Bold rows indicate the models proposed in this work, as distinguished from external baseline comparisons.

RLPA uniquely occupies the high-CDGR quadrant (CDGR = 3.451) with zero inference overhead. Its CDGR exceeds 1 because its held-out seen-class mAP50 (0.0255) is lower

than its cross-domain unseen mAP50 (0.088), as RLPA adapts only the text embedding space and leaves the visual backbone untouched, so its box-level precision on seen classes stays weak even though its top-1 accuracy (15%) is more evenly comparable across splits. RLPA + NeckFT occupies the high-accuracy quadrant (mAP50 = 0.538, CDGR = 0.2435, and top-1 = 73.5%) and is the only operating point for which large-scale testing (Section 5.5) shows consistent zero-shot generalisation, specifically for container-shaped items—it is the only operating point that combines strong seen-class accuracy *with* any evidence of container-class zero-shot generalisation. WasteYOLOE26-S provides maximum seen-class accuracy (74%) for deployments where the waste stream matches training categories.

5.8. Extended Baseline Comparison

We report eight additional baselines spanning supervised detection, edge-deployment classification, zero-shot classification, and zero-shot open-vocabulary detection, independently trained or evaluated using the same 728-image training set, 98-image seen test split, and 378-image unseen split used throughout (Table 9). Top-1 accuracy is reported for every model; CDGR is additionally reported for the three models with a genuine open-vocabulary detection mechanism and marked N/A for models with no such mechanism, for which CDGR is not a defined quantity.

Table 9. Extended baseline comparison. All models evaluated with identical top-1 methodology on the same seen (98-image) and unseen (378-image) splits. CDGR is marked N/A for models with no open-vocabulary detection mechanism.

Model	Seen Top-1	Unseen Top-1	CDGR
Head-only fine-tuning	45.9%	13.5%	N/A
Neck-only fine-tuning (no RLPA)	58.2%	17.5%	N/A
YOLOv8s (supervised) [43]	72.4%	16.1%	N/A
YOLOv11n (supervised)	51.0%	14.6%	N/A
MobileNetV2 (edge classifier)	66.3%	19.6%	N/A
CLIP ViT-B/32 (zero-shot)	55.1%	16.9%	N/A
YOLO-World-S	28.6%	14.8%	0.480
YOLOE-26s zero-shot	33.7%	16.9%	0.759
OWL-ViT	35.7%	6.1%	0.0073
RLPA + NeckFT (ours)	73.5%	18.0%	0.2435

Bold indicates the model proposed in this work, as distinguished from external baseline comparisons.

RLPA + NeckFT achieves the highest seen-class accuracy (73.5%) among all tested models, including both the closed-set baselines and the three genuine open-vocabulary detectors. On unseen top-1 specifically, our model (18.0%) does not show a dramatic advantage over the closed-set baselines, which cluster between 13.5% and 19.6%, regardless of architecture—even YOLOv8s, with the highest seen accuracy among closed-set baselines (72.4%), collapses to a similar unseen top-1 (16.1%). Against the three real open-vocabulary detectors, RLPA + NeckFT’s CDGR (0.2435) is lower than both YOLO-World-S (0.480) and YOLOE-26 zero-shot (0.759), though both achieve this at substantially lower seen-class mAP50 values (0.204 and 0.0218 respectively (Table 2) vs. RLPA + NeckFT’s 0.538); RLPA + NeckFT’s CDGR is higher than OWL-ViT’s (0.0073). This indicates that RLPA + NeckFT occupies a specific point in the accuracy–generalisation trade-off rather than dominating every open-vocabulary baseline on every axis: it is not the highest-CDGR detector available, but among detectors achieving comparable or higher seen-class accuracy, it is the only one retaining meaningful open-vocabulary capability at all.

5.9. Cross-Dataset Generalisation Validation

To assess generalisation on data entirely independent of the training distribution, WasteYOLOE26-S, RLPA, and RLPA + NeckFT are evaluated on two independent public datasets entirely separate from the training sources described in Section 3.2, totalling 7614 images: the Garbage Dataset (7010 images, 5 classes) and a dataset from Agbehadji et al. [44] (604 images and six classes mapped to our taxonomy after excluding two classes with no equivalent: Polystyrene and Metallic foil). These were selected as the most relevant of the considered candidates, since both consist of clean, studio-style single-item photographs, consistent with the campus bin deployment context this paper targets, unlike landfill-condition or narrowly two-class alternatives that do not reflect the intended use case. The Agbehadji et al. dataset additionally provides real ground-truth bounding-box annotations, allowing mAP50 to be reported alongside top-1; the Garbage Dataset is distributed in classification format only (a single label per image, with no bounding boxes), so mAP50 cannot be computed for it. Table 10 reports top-1 accuracy for all three model stages and mAP50 for RLPA + NeckFT, where ground-truth boxes are available.

Table 10. Cross-dataset top-1 accuracy across two independent public datasets entirely separate from the training sources reported in Section 3.2. mAP50 is reported for RLPA + NeckFT only, where real ground-truth bounding boxes exist, using the same held-out test-set figure (0.538) reported as the principal accuracy throughout this paper.

Dataset	<i>n</i>	Stage 1 (Full FT)	RLPA	RLPA + NeckFT	RLPA + NeckFT mAP50
Original test	98	74.5%	16.3%	73.5%	0.538
Garbage Dataset	7010	48.8%	26.7%	61.7%	—
Agbehadji et al. [44]	604	—	—	78.6%	0.109

RLPA + NeckFT performs comparably to full fine-tuning in domain (73.5%) and outperforms it on the Garbage Dataset (61.7% vs. 48.8%), broadly consistent with its characterisation above as the strongest performer on datasets where full fine-tuning underperforms. On the Agbehadji et al. dataset, RLPA + NeckFT achieves 78.6% top-1 accuracy, which is higher than on our own held-out test set, indicating that classification performance transfers well to this independent source. Its mAP50 (0.109) is lower than the held-out original-test-set figure (0.538), consistent with the expected effect of domain shift (different cameras, backgrounds, and framing conventions) on precise box localisation, even where classification remains strong.

6. Discussion

The experimental results surface five insights that go beyond the headline numbers and have direct implications for practitioners deploying open-vocabulary detectors on constrained hardware. First, fine-tuning is a deployment necessity for campus waste detection, not a design choice—but it imposes a severe CDGR collapse that cannot be ignored. Second, the RLPA regularisation parameter exhibits a sharp phase transition that defines the boundary between useful adaptation and no adaptation at all. Third, freezing the backbone during NeckFT preserves novel-item detection, as well as CDGR—a counter-intuitive result showing that CLIP alignment benefits compound across adaptation stages. Fourth, INT8 quantisation of a cosine-similarity embedding head behaves differently from standard softmax classifiers. Fifth, the three operating points offer practitioners a principled choice rather than a single forced trade-off. Each insight is discussed in turn, followed by an honest account of the study’s limitations.

6.1. Why Fine-Tuning Is Necessary Despite the CDGR Collapse

A natural question is whether fine-tuning is necessary at all. If vanilla YOLOE-26 already achieves $\text{CDGR} = 6.055$ in zero-shot mode, why not deploy it directly on EcoSortBin and avoid the CDGR collapse entirely?

The zero-shot baseline (Table 8) answers this directly: $\text{mAP50} = 0.174$ and $\text{top-1} = 34\%$. At the operational routing threshold of $\text{conf} \geq 0.60$ required for Stage 1 sensor fusion acceptance, the zero-shot model rarely produces confident predictions on campus waste—the bin falls through to the weight sensor on almost every item, making the vision component functionally redundant.

Three factors explain why zero-shot YOLOE-26 fails on campus waste despite strong general-object performance:

6.1.1. Vocabulary Coverage Gap

MobileCLIP was trained on Internet-scale image–text pairs where waste-domain terms such as `crisp_wrapper`, `aluminium_can` (as a category distinct from generic container), and `plastic_bag` appear infrequently and inconsistently. The cosine similarity scores for these classes are low and poorly separated at zero-shot (baseline inter-class similarity = 0.886 for `plastic_bottle` ↔ `glass_bottle`), making them insufficient for reliable arg max selection.

6.1.2. Visual Domain Gap

Campus bin cameras capture items from a fixed, top-down angle under variable indoor lighting, and they are often crushed, crumpled, or soiled. The COCO and LVIS pretraining corpora contain natural scene photographs of intact objects with good lighting. The visual backbone features that YOLOE-26 learned to match against text embeddings were never optimised for this appearance distribution.

Prompt engineering cannot bridge the visual gap for fine-tuned models. Material-form-context prompt engineering (e.g. “transparent crumpled PET plastic bottle” rather than “bottle”) can improve text-side embedding alignment in zero-shot models and partially recover detection rates for some categories when the visual backbone retains CLIP alignment. However, for a *fine-tuned* model such as WasteYOLOE26-S, prompt engineering cannot compensate for the visual domain gap: the backbone produces features for top-down, degraded-appearance campus waste that no longer align with the general-object visual manifold, regardless of how the text prompt is written—because full fine-tuning has already displaced the backbone from the CLIP embedding space. We verified this empirically by evaluating WasteYOLOE26-S under three prompt strategies—plain class names, richer material-form descriptions, and campus-context prompts—all yielding identical top-1 accuracy of 74% (73/98). Domain context, not prompt style, determines detection performance *after fine-tuning*. Our zero-shot evaluation at $\text{conf} = 0.001$ (34% top-1) already uses the maximum sensitivity threshold—no prompt engineering strategy can recover the 40-percentage-point gap to operational accuracy once the backbone has drifted from the CLIP visual manifold.

Fine-tuning is therefore not a design choice but a necessity for reliable bin routing. The contribution of this paper is not to justify fine-tuning but to quantify, then address the generalisation cost it imposes: the severe CDGR collapse that makes a fine-tuned bin blind to its own trained categories once they appear in a different dataset or setting. RLPA and NeckFT recover this capability while preserving the accuracy gains that make deployment viable.

CDGR measures cross-domain generalisation rather than raw detection capability. A model can perform poorly on the unseen (cross-dataset) split for two different reasons: the model may simply be a weak detector overall, performing poorly on seen and unseen

data alike, or the model may be a strong detector that has specifically lost the ability to recognise its trained classes once they appear in a different dataset and camera viewpoint. Unseen mAP50, on its own, cannot distinguish between these two cases. Normalising unseen mAP50 by the model's own held-out seen-class mAP50 isolates the second case—this measures how much of a model's demonstrated detection capability, on classes it already knows well, carries over to the same classes photographed differently. CDGR is not intended as a comprehensive standalone measure of open-vocabulary generalisation; we report it alongside the per-class breakdown in Table 4, which captures variation CDGR's single ratio would otherwise hide, and the novel-item probe reported alongside Table 8, which tests genuinely new object categories rather than the same categories in a different domain. Together, these three measures give a more complete picture than any one metric alone.

6.2. The λ Phase Transition in RLPA

The ablation reveals a sharp transition between $\lambda = 0.01$ (effective adaptation, separation ≥ 0.39) and $\lambda \geq 0.5$ (no adaptation, separation = -0.095). At $\lambda = 0.01$, the Frobenius penalty allows the adapter to shift prompt embeddings toward waste-domain semantics; at $\lambda \geq 0.5$, the penalty dominates, and ΔW_B remains near zero throughout training. This transition motivates a finer hyperparameter search around $\lambda = 0.01$ – 0.05 in future work.

6.3. Novel-Item Zero-Shot Generalisation Through Frozen-Backbone NeckFT

RLPA + NeckFT shows zero-shot generalisation evidence for container-shaped classes despite adding 50 epochs of supervised neck fine-tuning. This is a counter-intuitive positive result: one might expect that fine-tuning on 728 seen-class images would further overfit the model and reduce novel sensitivity. The frozen-backbone design prevents this. By keeping layers 0–10 (Conv, C3k2, SPPF, and C2PSA) frozen, the CLIP-aligned visual features that enable generalisation to unseen categories are preserved throughout NeckFT training. The neck learns to fuse these features for campus waste appearance without corrupting the backbone representations.

The practical implication is significant: RLPA + NeckFT is not a generalisation–accuracy trade-off but a Pareto improvement over RLPA alone—it achieves 73.5% top-1 accuracy (vs. 15% for RLPA) while additionally showing distinct zero-shot generalisation evidence at scale (correct guesses spanning more than one class, unlike RLPA or WasteYOLOE26-S) under the same large-scale validation (Section 5.5). The only cost is the additional 50-epoch neck fine-tuning stage.

6.4. QAT Robustness and Quantisation Readiness

Kendall $\tau = 1.00$ confirms that QAT training does not reorder top-1 class predictions relative to the FP32 NeckFT baseline. This is the key result: the model's cosine similarity ordering is completely stable under INT8-level numerical perturbation, meaning it is ready for INT8 deployment when toolchain support becomes available. The +4 pp top-1 improvement from NeckFT to QAT reflects additional fine-tuning benefit from the 30 QAT epochs rather than any quantisation effect—the weights are better calibrated, not compressed.

The current limitation is that Ultralytics 8.4 does not support `int8 = True` for NCNN export on ARM64, so the current deployed artefact remains ONNX FP32 (41.8 MB, 1113 ms on Pi 4). True INT8 deployment via native NCNN would reduce the model size to ≈ 11.8 MB and improve throughput by 1.5 – $2\times$ on the Cortex-A72 ARM NEON units. This is noted as future work: once INT8 NCNN export is available, the QAT-trained model can be directly quantised without any retraining, since its weight distributions have already been prepared for this transition.

6.5. Sensor Fusion Validation on Real Hardware

To provide direct experimental validation of the sensor fusion protocol, we conducted a real hardware trial using the RLPA + NeckFT model (ONNX FP32), the operating point recommended for production deployment in Section 6.6: five physical items per class (35 items in total, following the per-class trial design of [44]) were placed under the deployed camera and weighed on the HX711 load cell. Table 11 reports the vision confidence score for each trial. Table 12 reports the corresponding inference latency. Table 13 reports the measured weight.

Table 11. Vision confidence scores across 5 real trials per class.

Class	Item 1	Item 2	Item 3	Item 4	Item 5	Mean
plastic_bottle	0.397	0.647	0.384	0.768	0.496	0.538
glass_bottle	0.971	0.886	0.962	0.555	0.762	0.827
aluminium_can	0.253	0.833	0.323	0.254	0.618	0.456
crisp_wrapper	0.888	0.117	0.339	0.238	0.746	0.466
paper_waste	0.732	0.788	0.502	0.555	0.945	0.704
cardboard_box	0.282	0.725	0.907	0.874	0.429	0.643
plastic_bag	0.531	0.102	0.357	0.445	0.666	0.420
Overall mean						0.579

Bold indicates the summary row (overall mean) for the table.

Table 12. Inference latency (ms) across 5 real trials per class.

Class	Item 1	Item 2	Item 3	Item 4	Item 5	Mean
plastic_bottle	920.19	1198.83	1378.08	2976.17	2274.89	1749.63
glass_bottle	1987.45	2096.54	654.43	2956.41	1474.06	1833.78
aluminium_can	790.28	1362.90	2362.09	1639.21	1232.08	1477.31
crisp_wrapper	2548.46	2108.35	1293.98	586.08	808.99	1469.17
paper_waste	923.37	1073.90	2543.36	730.11	1598.34	1373.82
cardboard_box	1703.75	942.97	1945.80	2298.20	1495.02	1677.15
plastic_bag	2760.78	2524.08	1704.83	1983.84	1024.00	1999.51
Overall mean						1654.34

Bold indicates the summary row (overall mean) for the table.

Table 13. Measured weight (g) across 5 real trials per class.

Class	Item 1	Item 2	Item 3	Item 4	Item 5	Min/Max/Mean
plastic_bottle	28.0	8.0	42.0	28.0	47.0	8.0/47.0/30.6
glass_bottle	198.0	198.0	198.0	230.0	302.0	198.0/302.0/225.2
aluminium_can	16.5	10.2	12.0	17.0	11.8	10.2/17.0/13.5
crisp_wrapper	45.0	6.0	10.0	4.0	7.0	4.0/45.0/14.4
paper_waste	7.0	12.0	3.0	7.0	15.0	3.0/15.0/8.8
cardboard_box	21.0	84.0	33.0	61.0	109.0	21.0/109.0/61.6
plastic_bag	6.0	3.0	8.0	8.0	9.0	3.0/9.0/6.8

Applying the three-stage routing protocol (Section 3.3) to these 35 real trials, 15 items cleared the $\text{conf} \geq 0.60$ direct-accept threshold (100% correctly routed); 5 aluminium_can items were correctly routed via the inductive metal override, regardless of vision confidence (100%, including one case where vision itself misclassified the item); 10 items entered weight arbitration (80% correctly routed); and 5 items were safely deferred to the general bin for manual review. Overall, 28/35 (80.0%) were correctly routed to the physical bin, 5/35 (14.3%) were safely deferred, and 2/35 (5.7%) were genuinely mis-routed. Both misrouted cases involved a plastic_bottle whose measured weight (28 g

and 47 g) fell within the overlapping ranges of `plastic_bottle` and `cardboard_box`—a genuine physical weight overlap rather than a sensor or classification fault. Notably, the `crisp_wrapper/plastic_bag` confusion identified in the confusion matrix (Table 7, Section 5) is reproduced in this independent real hardware trial: three of five `crisp_wrapper` items were misclassified as `plastic_bag`, and three of five `plastic_bag` items were misclassified as `crisp_wrapper`, consistent in direction and magnitude with the 6/14 rate observed in each direction on the original 98-image test set. This cross-validation across two independent evaluations strengthens confidence that this is a genuine, reproducible model characteristic rather than a dataset artefact.

End-to-end latency breakdown. To directly address a complete deployment-time latency accounting, we instrumented six pipeline stages across the same 35 real hardware trials (Table 14). Camera capture (mean 50.7 ms) and model inference (mean 1654.3 ms, consistent with Table 12) dominate the vision path; since YOLOE-26 is NMS-free [9,20], pre- and post-processing occur inside the single measured inference call rather than as separate stages. Weight-sensor arbitration (HX711, mean 482.6 ms) applies only to the 10/35 items reaching Stage 2 (confidence: 0.35–0.60). UART/servo actuation (mean 464.3 ms) and Firebase/MQTT publishing (mean 194.7 ms) are the largest non-vision contributors; LED/buzzer feedback is negligible (11.3 ms). Summing all components gives a mean of 2858.1 ms per disposal event, consistent with the “2–3 s” cycle-time estimate stated in Section 3.1.

Table 14. End-to-end latency breakdown across 35 real hardware trials (same items as Tables 11–13).

Component	Mean Latency (ms)	Range (ms)	n
Camera capture	50.7	30–71	35
Model inference (WasteYOLOE26-S, ONNX FP32)	1654.3	586.1–2976.2	35
Weight sensor (HX711) *	482.6	350–670	10
UART command + servo actuation	464.3	306–693	35
LED + buzzer feedback	11.3	5–20	35
Firebase + MQTT	194.7	105–295	35
Sum of components (mean)	2858.1	—	—

Bold indicates the summary row (sum of components). * Weight sensor (HX711) measured only for the 10/35 items that reached Stage 2 arbitration (confidence: 0.35–0.60), matching the count already reported above.

6.6. Deployment Trade-Off Guide for Practitioners

Based on the complete Pareto frontier (Table 8):

- **Static known vocabulary, maximum accuracy:** WasteYOLOE26-S (74% top-1, 1113 ms Pi 4 ONNX FP32).
- **Zero inference overhead required, generalisation capability retained but not required to be immediately deployment-ready:** RLPA (CDGR = 3.451, zero overhead, 15% top-1).
- **Novel packaging important, especially container-shaped items:** RLPA + NeckFT (73.5% top-1, CDGR = 0.2435, zero-shot generalisation evidence at scale, 1113 ms Pi 4 ONNX FP32).

6.7. Limitations

The dataset was limited to 728 training images; `crisp_wrapper` and `plastic_bag` suffer from domain mismatch between training (TACO outdoor) and test (campus bin camera) sets. On a Raspberry Pi 4B (Cortex-A72, 4 threads, ONNX FP32, 640 × 640), YOLOE-26s-seg runs at 1113 ms/frame (0.90 FPS)—suitable for motion-triggered capture but not continuous video. Scaling from one to four threads yielded a 4.1× speedup (4558 ms → 1113 ms). INT8 deployment via `quantize_dynamic` (ONNX Runtime) is not viable on

a Pi 4: the `ConvInteger` operator is unsupported by the aarch64 CPU execution provider. The recommended path to sub-500 ms deployment is NCNN with FP16 weights, leveraging ARM NEON intrinsics; this remains an open item. The QAT implementation uses standard fake quantisation with the straight-through estimator; ranking-preserving loss terms (e.g. margin losses on cosine similarity ordering) were not incorporated, meaning the $\tau = 1.00$ result reflects empirical stability rather than an enforced constraint. RLPA effectiveness depends critically on base model selection: applying the adapter to an already fine-tuned backbone substantially reduced CDGR recovery in an additional ablation experiment where RLPA was applied post Stage 1 (measured on the original training-split-based evaluation) because backbone drift from the CLIP visual manifold limits text-space adaptation—confirming that RLPA must be applied before domain fine-tuning to preserve open-vocabulary generalisation. Validation on WaDaBa (pending license access) and on real campus-collected images under the physical lighting, clutter, and camera conditions of an operational EcoSortBin deployment remain important directions for future validation beyond the cross-dataset results reported in Section 5.9.

7. Conclusions

This paper presented EcoSortBin, a complete accuracy–generalisation study for open-vocabulary campus waste detection on a Raspberry Pi 4.

WasteYOLOE26-S delivers a fine-tuned open-vocabulary waste detector on a Raspberry Pi 4 ARMv8 (74% top-1, 1113 ms ONNX FP32) and empirically characterises the severe CDGR collapse that arises specifically under domain-specific fine-tuning of open-vocabulary models—a phenomenon distinct from zero-shot deployment.

RLPA achieves substantial CDGR recovery over full fine-tuning by constraining adaptation to the MobileCLIP prompt-embedding space. The λ phase transition at ≈ 0.01 provides a concrete design guideline for future LoRA-style prompt adaptation in open-vocabulary detectors beyond the waste domain.

RLPA + NeckFT recovers 73.5% top-1 accuracy with a frozen backbone and substantially better CDGR than full fine-tuning while showing zero-shot generalisation evidence on a large-scale validation set concentrated in container-shaped classes. Freezing the backbone is not merely a CDGR preservation strategy but also enables some novel-category detection to persist through neck adaptation for container-shaped classes specifically because CLIP-aligned backbone features provide the visual manifold that both tasks depend on.

The QAT robustness study ($\tau = 1.00$, $\Delta\text{Top-1} = +4$ pp vs. NeckFT) delivers the first empirical characterisation of quantisation-aware training behaviour on a waste-domain open-vocabulary embedding head. $\tau = 1.00$ confirms that cosine similarity rank ordering is completely stable under INT8-level numerical perturbation, establishing that the model is ready for INT8 deployment when toolchain support becomes available. The current deployed artefact is ONNX FP32 (41.8 MB, 1113 ms on Pi 4). An NCNN FP32 artefact is also released; true INT8 NCNN export (≈ 11.8 MB) awaits ARM64 toolchain support and is noted as future work.

The EcoSortBin hardware platform—integrating RC522 RFID authentication, an HX711 load cell and inductive sensor fusion, and a gamified points engine with a live leaderboard—demonstrates that open-vocabulary waste detection is practically viable for campus deployment, not merely a laboratory result. The configurable vocabulary system allows any waste category to be added at runtime via `set_text_pe()` without model retraining, directly enabling the dynamic packaging vocabularies that motivated this work.

Together, the three operating points (WasteYOLOE26-S, RLPA, and RLPA + NeckFT) form the complete Pareto frontier of the accuracy–CDGR trade-off for campus waste detection. Notably, RLPA + NeckFT occupies a previously unreachable quadrant: both high

seen-class accuracy (73.5%) and demonstrated container-class generalisation, indicating that the accuracy–generalisation trade-off is not fully fundamental but can be substantially mitigated by addressing backbone drift during fine-tuning.

Author Contributions: Conceptualization, M.B.; methodology, M.B.; software, M.B.; validation, M.B.; formal analysis, M.B.; investigation, M.B.; resources, M.B.; data curation, M.B.; writing—original draft preparation, M.B.; writing—review and editing, M.B. and S.P.; visualization, M.B.; supervision, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The EcoSortBin dataset, RLPA codebase, trained checkpoints, and evaluation scripts are available at <https://github.com/Madhini81/EcoSortBin> (accessed on 13 June 2026).

Acknowledgments: The authors thank the Ultralytics, Roboflow, and SAM open-source communities and the TrashNet, TACO, and Drinking Waste Classification dataset contributors. During the preparation of this manuscript, the authors used Claude Sonnet 5 (Anthropic) for the purposes of LaTeX editing assistance and consistency checking. The authors have reviewed and edited all output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CDGR	Cross-Domain Generalisation Ratio
CLIP	Contrastive Language–Image Pretraining
FPN	Feature Pyramid Network
INT8	8-bit Integer Quantisation
IoT	Internet of Things
LoRA	Low-Rank Adaptation
mAP	Mean Average Precision
MQTT	Message-Queuing Telemetry Transport
NCNN	Neural Network Compute & Nina (mobile inference framework)
NMS	Non-Maximum Suppression
ONNX	Open Neural Network Exchange
PAN	Path Aggregation Network
QAT	Quantisation-Aware Training
RepRTA	Re-parameterisable Text–Image Alignment
RFID	Radio Frequency Identification
RLPA	RepRTA-Compatible LoRA Prompt Adapter
SAM	Segment Anything Model

References

1. Farrell, C.C.; Cruse, A.; Collins, M.; Gallagher, J.; Maier, M. Contamination of recycling streams: A systematic review. *Resour. Conserv. Recycl.* **2022**, *186*, 106578.
2. Aral, R.A.; Keskin, S.R.; Kaya, M.; Hacıömeroğlu, M. Classification of TrashNet dataset based on deep learning models. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018.
3. Thung, G.; Yang, M. Classification of trash for recyclability status. In *CS229 Project Report*; Stanford University: Stanford, CA, USA, 2016.

4. Tran, Q.V.; Minh, L.H.; Nguyen, T.T. Waste classification using deep learning: A comprehensive review. *IEEE Access* **2023**, *11*, 12300–12325.
5. Wang, T.; Cai, Y.; Liang, L.; Ye, D. A multi-level approach to waste object segmentation. *Sensors* **2020**, *20*, 3816. [[CrossRef](#)] [[PubMed](#)]
6. Vo, A.H.; Son, L.H.; Vo, M.T.; Le, T. A waste recycling system based on image recognition. *IEEE Access* **2019**, *7*, 35974–35985. [[CrossRef](#)]
7. Bobulski, J.; Kubanek, M. Deep learning for plastic waste classification system. *Appl. Comput. Intell. Soft Comput.* **2021**, *2021*, 6626948. [[CrossRef](#)]
8. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the 38th International Conference on Machine Learning (ICML 2021), Virtual, 18–24 July 2021; pp. 8748–8763.
9. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOE-26: Efficient Open-Vocabulary Detection. Ultralytics Technical Report. 2025. Available online: <https://docs.ultralytics.com/models/yoloe/> (accessed on 1 March 2025).
10. Proença, P.F.; Simões, P. TACO: Trash Annotations in Context for Litter Detection. *arXiv* **2020**, arXiv:2003.06975.
11. Lun, Z.; Chen, H.; Wang, Y.; Zhang, F. GD-YOLOv10n: Lightweight garbage detection combining TACO and field-collected data. *Expert Syst. Appl.* **2025**, *275*, 127085. [[CrossRef](#)]
12. Liu, S.; Chen, R.; Ye, M.; Luo, J.; Yang, D.; Dai, M. EcoDetect-YOLO: A lightweight, high-generalisation methodology for real-time detection of domestic waste in complex environments. *Sensors* **2024**, *24*, 4666. [[CrossRef](#)] [[PubMed](#)]
13. Kunwar, S.; Alade, A. Plastic waste detection using deep learning: Insights from the WaDaBa dataset. *Ind. Domest. Waste Manag.* **2025**, *5*, 1–11. [[CrossRef](#)]
14. Bobowska, I.; Lach, J.; Woźniak, M.; Zdunek, R. A systematic review of AI-based techniques for automated waste classification. *Appl. Sci.* **2025**, *15*, 5498.
15. Li, L.H.; Zhang, P.; Zhang, H.; Yang, J.; Li, C.; Zhong, Y.; Wang, L.; Yuan, L.; Zhang, L.; Hwang, J.-N.; et al. Grounded language-image pre-training (GLIP). In Proceedings of the IEEE/CVF CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 10965–10975.
16. Zhou, X.; Girdhar, R.; Joulin, A.; Krähenbühl, P.; Misra, I. Detecting twenty-thousand classes using image-level supervision. In Proceedings of the ECCV 2022, Tel Aviv, Israel, 23–27 October 2022; pp. 350–368.
17. Minderer, M.; Gritsenko, A.; Stone, A.; Neumann, M.; Weissenborn, D.; Dosovitskiy, A.; Mahendran, A.; Arnab, A.; Dehghani, M.; Shen, Z.; et al. Simple open-vocabulary object detection. In Proceedings of the ECCV 2022, Tel Aviv, Israel, 23–27 October 2022; pp. 728–755.
18. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Li, C.; Yang, J.; Su, H.; Zhu, J.; et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *arXiv* **2023**, arXiv:2303.05499.
19. Cheng, T.; Song, L.; Ge, Y.; Liu, W.; Wang, X.; Shan, Y. YOLO-World: Real-time open-vocabulary object detection. In Proceedings of the IEEE/CVF CVPR 2024, Seattle, WA, USA, 17–21 June 2024.
20. Wang, A.; Liu, L.; Chen, H.; Lin, Z.; Han, J.; Ding, G. YOLOE: Real-time seeing anything. In Proceedings of the IEEE/CVF ICCV 2025, Honolulu, HI, USA, 19–25 October 2025.
21. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. In Proceedings of the IEEE/CVF ICCV 2023, Paris, France, 2–6 October 2023.
22. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-rank adaptation of large language models. In Proceedings of the ICLR 2022, Virtual, 25–29 April 2022.
23. Lester, B.; Al-Rfou, R.; Constant, N. The power of scale for parameter-efficient prompt tuning. In Proceedings of the EMNLP 2021, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 3045–3059.
24. Li, X.L.; Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the ACL 2021, Bangkok, Thailand, 1–6 August 2021; pp. 4582–4597.
25. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In Proceedings of the ICML 2019, Long Beach, CA, USA, 9–15 June 2019; pp. 2790–2799.
26. Zhou, K.; Yang, J.; Loy, C.C.; Liu, Z. Learning to prompt for vision-language models. *Int. J. Comput. Vis.* **2022**, *130*, 2337–2348. [[CrossRef](#)]
27. Zhou, K.; Yang, J.; Loy, C.C.; Liu, Z. Conditional prompt learning for vision-language models. In Proceedings of the IEEE/CVF CVPR 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 16816–16825.
28. Gao, P.; Geng, S.; Zhang, R.; Ma, T.; Fang, R.; Zhang, Y.; Li, H.; Qiao, Y. CLIP-Adapter: Better vision-language models with feature adapters. *Int. J. Comput. Vis.* **2021**, *132*, 581–595.
29. Jia, M.; Tang, L.; Chen, B.C.; Cardie, C.; Belongie, S.; Hariharan, B.; Lim, S.N. Visual prompt tuning. In Proceedings of the ECCV 2022, Tel Aviv, Israel, 23–27 October 2022; pp. 709–727.

30. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE/CVF CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018.
31. Bengio, Y.; Léonard, N.; Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv* **2013**, arXiv:1308.3432.
32. Sapkota, R.; Cheppally, R.H.; Sharda, A.; Karkee, M. YOLO26: Key architectural enhancements and performance benchmarking for real-time object detection. *arXiv* **2025**, arXiv:2509.25164.
33. Sauer, J.; Seibel, K.; Rüttinger, B. Human factors in ecological interface design: Informing waste sorting behaviour through smart bin feedback. *Ergonomics* **2020**, *63*, 430–444.
34. Google LLC. Firebase Realtime Database Documentation. Firebase Developer Documentation. 2024. Available online: <https://firebase.google.com/docs/database> (accessed on 1 March 2025).
35. OASIS. MQTT Version 5.0 Specification. OASIS Standard. 2019. Available online: <https://mqtt.org/mqtt-specification/> (accessed on 1 March 2025).
36. Sekar, G. Garbage Classification Dataset. Kaggle. 2019. Available online: <https://www.kaggle.com/datasets/asdasdasdas/garbage-classification> (accessed on 15 January 2025).
37. Hacks, A. Drinking Waste Classification Dataset. Kaggle. 2021. Available online: <https://www.kaggle.com/datasets/arkadiyhacks/drinking-waste-classification> (accessed on 15 January 2025).
38. Roboflow Inc. TACO Official Dataset. Roboflow Universe. 2023. Available online: https://universe.roboflow.com/madhiniworkspace/taco_official-ctazb (accessed on 20 January 2025).
39. Sinha, A.; Jain, S.; Kumar, P. Mobile gamification for pro-environmental behaviour: Barriers to adoption in campus sustainability programs. *Sustain. Cities Soc.* **2021**, *72*, 103007.
40. Huang, P.-Y.; Yeh, C.-H. A gamification-based smart recycling system for improving recycling behaviour. *J. Clean. Prod.* **2021**, *285*, 125464.
41. Zhang, H.; Deng, X.; Li, J. Incentive mechanisms for campus waste sorting: A behavioural economics perspective. *Waste Manag.* **2022**, *140*, 96–105.
42. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
43. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. GitHub Repository. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 1 March 2025).
44. Agbehadji, I.E.; Abayomi, A.; Bui, K.-H.N.; Millham, R.C.; Freeman, E. Nature-Inspired Search Method and Custom Waste Object Detection and Classification Model for Smart Waste Bin. *Sensors* **2022**, *22*, 6176. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.