Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 4100~4108
e-ISSN: 2548-6861

Benchmarking YOLO26 Against YOLOv11 for Minority Class Waste Detection on an Augmented TACO Dataset

Lingga Kurnia Ramadhani [1]*, Bajeng Nurul Widyaningrum [2]**
* Sistem dan Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas IVET Semarang
** Rekam Medis dan Informasi Kesehatan, Politeknik Bina Trada Semarang
linggadhani79@gmail.com [1], bnwidyani@gmail.com [2]

Article Info

Article history:

Received 2026-07-26
Revised 2026-08-01
Accepted 2026-08-10

Keyword:

YOLO26,
Class Imbalance,
Small Object Detection,
Waste Classification,
TACO Dataset.

ABSTRACT

This study benchmarks YOLO26 against YOLOv11 for detecting minority waste categories (Hazardous/B3 and Residue), evaluating whether YOLO26's Progressive Loss Balancing (ProgLoss) and Small-Target-Aware Label Assignment (STAL) mechanisms address class imbalance and small-object detection challenges. Both models were trained under identical conditions (50 epochs, 640×640) on a combined TACO, RecyBat24, and Food Waste Detection dataset (2,326 images, 70:15:15 split), across three scenarios: YOLOv11 baseline, YOLO26 default, and YOLO26 with class weighting and copy-paste augmentation. YOLO26 (default) achieved a marginally higher mAP@0.5:0.95 (0.393 vs 0.385) and modestly faster CPU inference (≈20% faster) than YOLOv11, with near-identical mAP@0.5 across scenarios. B3 performed consistently well (mAP@0.5 ≈ 0.99), and class weighting improved its detection robustness without raising overall mAP. Residue detection remained the weakest across all scenarios (mAP@0.5 0.076–0.085) and worsened under weighting, indicating that ProgLoss and STAL alone do not resolve its structural visual heterogeneity; this weak, stable Residue performance was confirmed reproducible across three additional training runs with different seeds (mAP@0.5:0.95 = 0.046 ± 0.001). These findings partially support the research hypothesis, positioning YOLO26 (default) as a favorable accuracy-efficiency trade-off for automated waste sorting, while Residue detection requires further data enrichment and augmentation strategies beyond architectural improvements alone.

I. INTRODUCTION

Prior research implementing YOLOv11 for image-based waste detection and classification has demonstrated strong overall performance, achieving an mAP@0.5 of 0.989. However, further evaluation revealed a significant performance drop at mAP@0.5:0.95 (0.693) and for the Residue category, which reached an mAP50 of only 0.702, far below other categories such as Organic (0.991) and B3 (0.978). This condition indicates that the model still struggles to detect small objects and underrepresented (minority class) categories in the dataset.

Class imbalance and the difficulty of detecting small objects are classic challenges in object detection that remain underexplored in the waste classification domain [1]. Prior studies have applied deep learning and YOLO-based architectures to waste classification with promising aggregate results [2][3][4][7][8][9], but most report only overall metrics and do not isolate performance on minority or hazardous categories. Meanwhile, Ultralytics released YOLO26 as the latest generation of the YOLO architecture, introducing several relevant innovations. ProgLoss dynamically reweights the contribution of classification and localization loss terms as training progresses, giving relatively more weight to under-performing objects (including small and rare-class instances) in later epochs rather than applying a fixed weighting scheme throughout training. STAL modifies the label assignment strategy used to decide which anchor points are treated as positive samples during training, relaxing the assignment criteria for small ground-truth boxes so that small objects are not systematically under-represented among positive samples, a known cause of poor small-object recall

in anchor-based and anchor-free detectors alike. YOLO26 additionally removes Distribution Focal Loss (DFL) for computational efficiency, adopts an end-to-end architecture without Non-Maximum Suppression (NMS), and uses the MuSGD optimizer for training stability [12]. As of this writing, the description of these mechanisms is available only through Ultralytics' official technical documentation and a preprint benchmarking report rather than a peer-reviewed publication [12]; this study treats their effectiveness on minority-class waste detection as an open empirical question rather than an established claim.

Based on this gap, this study aims to examine the extent to which the ProgLoss and STAL mechanisms in YOLO26 can address class imbalance and small-object detection issues for minority waste categories, and to compare its performance head-to-head with YOLOv11 under equivalent experimental conditions. In this study, "minority" is defined at the object-annotation level rather than the image level: B3 (Hazardous, e.g., batteries and electronic waste) and Residue (heterogeneous non-recyclable items such as tissue, styrofoam, cigarette butts, and foam packaging) are the two categories with the fewest annotated object instances in the test set, and are also the categories of greatest practical concern for automated sorting since misclassifying B3 items carries safety and environmental risk. This study uses the open-access TACO (Trash Annotations in Context) dataset, which exhibits a long-tail class distribution and real-world environmental conditions, making it relevant for testing the generalization capability of both models.

Beyond a direct architecture comparison, this study contributes: (1) a dataset-enrichment strategy that combines TACO with two domain-specific open datasets to correct the severe under-representation of B3 and Organic categories in TACO alone; (2) a class-level diagnostic analysis, using per-class metrics and confusion matrices at a fixed operating threshold, that identifies whether performance gaps between minority categories stem from data scarcity or from structural visual heterogeneity, a distinction that aggregate metrics such as overall mAP cannot reveal; (3) an apple-to-apple comparative evaluation of YOLO26 and YOLOv11 on the same dataset and training conditions, including an assessment of whether YOLO26's claimed ProgLoss and STAL mechanisms translate into measurable gains for minority classes in practice; and (4) practical recommendations, together with an explicit account of where the tested strategies fail, for deploying such models in automated waste classification systems.

II. METHOD

This study adopts a quantitative approach using a comparative experimental method to evaluate the performance of YOLO26 against YOLOv11 in detecting and classifying image-based waste, with an emphasis on minority categories. The research stages are systematically illustrated in the following algorithm flowchart.

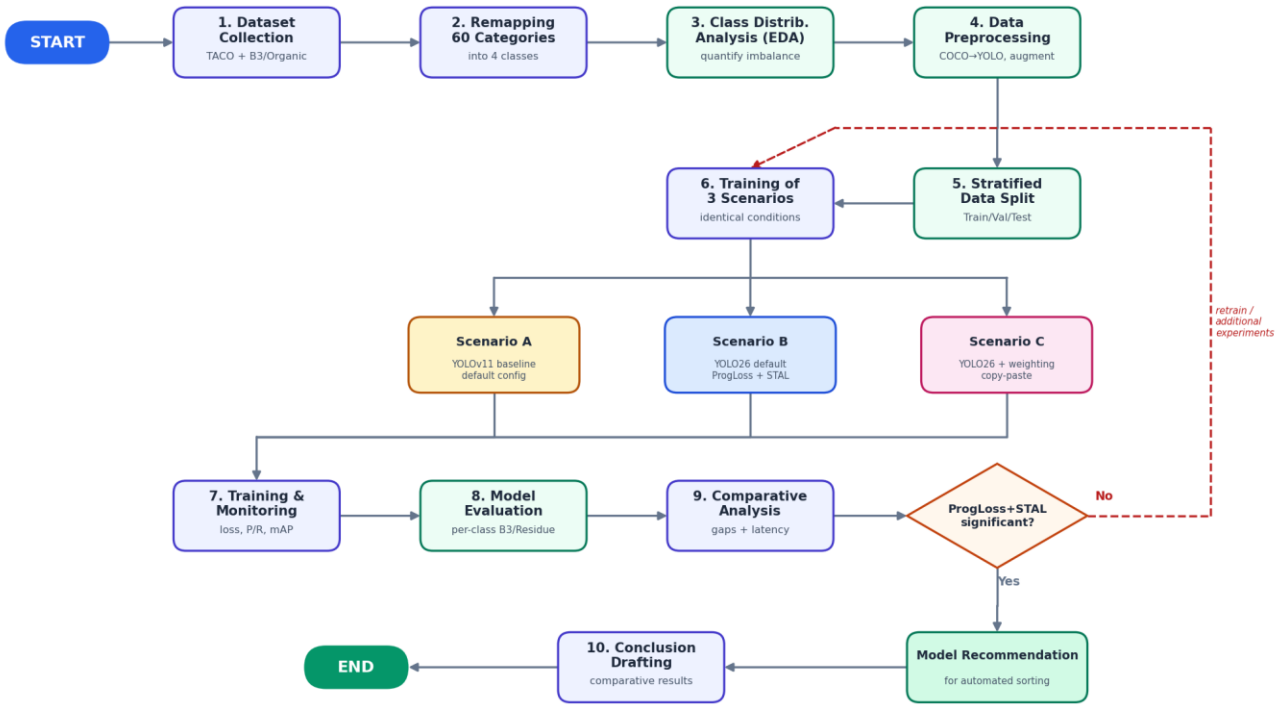

Figure 1. Research Methodology Flowchart

A. Dataset Collection

The primary dataset used is TACO (Trash Annotations in Context), an open-access waste dataset with COCO JSON format annotations comprising 1,500 images and 4,784 object annotations across 60 granular categories [13]. These categories were remapped into 4 classes: Inorganic, Organic, B3 (Hazardous), and Residue, based on a manually defined category-to-class mapping scheme.

Initial analysis of the remapping results shows that TACO is inherently highly imbalanced toward the Organic and B3 classes (each contributing less than 1% of total annotations), since TACO focuses on street/outdoor litter and therefore rarely contains food waste or household hazardous waste. To address this limitation, the dataset was enriched with two additional open-access sources specifically containing objects from these minority categories: (1) a battery/e-waste detection dataset to enrich the B3 class, using RecyBat24 [6] as the primary academic reference, consistent with recent advances in e-waste image classification [5]; and (2) a food waste detection dataset to enrich the Organic class, sourced from the open Roboflow Universe repository. All annotations from the additional sources were converted and mapped to the same label scheme before being merged with TACO, using a category-to-class mapping tailored to each source. For TACO, the 60 original categories were mapped explicitly: Battery, Plastic Gloves, and Aerosol were mapped to B3; Food Waste was mapped to Organic; Tissues, Foam Cup, Foam Food Container, Styrofoam Piece, Cigarette, and Unlabeled Litter were mapped to Residue; and all remaining categories (plastic, paper, metal, and glass items) defaulted to Inorganic. Because RecyBat24 and Food Waste Detection are each single-purpose, source-specific datasets, all of their original categories (battery shape types for RecyBat24; food-item types such as bone, vegetable peel, and eggshell for Food Waste Detection) were mapped in blanket fashion to B3 and Organic, respectively, without requiring category-level disambiguation. Bounding boxes were converted from COCO pixel format to normalized YOLO format and clamped to the [0, 1] range to guard against out-of-bounds coordinates. The combined image pool was then split into training, validation, and test subsets (70:15:15) using stratified sampling based on each image's dominant annotated class, rather than a purely random image-level split, to keep minority-class proportions represented across all three subsets.

B. Dataset Preprocessing

COCO JSON annotations were converted to YOLO format (.txt per image). The data was split using stratified sampling to keep minority-class proportions represented in the validation and test sets. Data augmentation was performed using Ultralytics' built-in features (mosaic, mixup, HSV augmentation, random flip), along with additional strategies for minority classes, such as oversampling or copy-paste augmentation for the B3 and Residue categories.

C. YOLO26 and YOLOv11 Implementation

Both models were implemented using the Ultralytics framework under equalized training conditions: identical image resolution (640×640), number of epochs (50), batch size (16), data split, and optimizer configuration (AdamW, automatically selected by Ultralytics with lr0=0.00125, momentum=0.9), to ensure a fair (apple-to-apple) comparison. Both models were initialized from their respective official COCO-pretrained weights provided by Ultralytics, and each model's own default data-augmentation pipeline (mosaic, mixup, HSV augmentation, random flip) was kept active and unmodified across scenarios, except for the additional class weighting and copy-paste augmentation introduced specifically in Scenario C. All augmentation and class-weighting operations, including those in Scenario C, were applied only to the training subset; the validation and test subsets were left unaugmented throughout to prevent data leakage into the reported evaluation metrics. The experimental scenarios were designed as follows:

- Scenario A: YOLOv11 (baseline, default configuration).
- Scenario B: YOLO26 (default, without additional adjustments).
- Scenario C: YOLO26 + additional class-imbalance handling strategy (class weighting/oversampling of minority classes).

Specific training parameters (learning rate, optimizer, number of epochs) were determined through initial tuning and are documented in detail in the final report.

D. Model Evaluation

Evaluation was carried out using Precision, Recall, F1-Score, mAP@0.5, and mAP@0.5:0.95 metrics, following standard object detection evaluation formulations. Special emphasis was placed on per-class performance breakdown, particularly for the B3 and Residue categories, as well as a comparative confusion matrix analysis between YOLO26 and YOLOv11. In addition to accuracy metrics, CPU inference latency was also measured given YOLO26's claimed efficiency on edge devices.

E. Comparative Analysis

The results of the three scenarios were compared to answer the research hypothesis: whether the ProgLoss and STAL mechanisms in YOLO26 significantly improve mAP50-95 and recall on the B3 and Residue categories compared to YOLOv11. A simple statistical comparison (e.g., metric differences across scenarios) was used to strengthen the validity of the findings.

III. RESULTS AND DISCUSSION

This section presents the experimental results of the three training scenarios that were conducted, namely YOLOv11 (baseline), YOLO26 (default configuration), and YOLO26 with an additional class-imbalance handling strategy (class weighting and copy-paste augmentation), hereafter referred to as YOLO26-Weighted. All three models were trained under identical conditions (50 epochs, 640×640 resolution, batch size 16, NVIDIA Tesla T4 GPU) using the Ultralytics framework, ensuring an apple-to-apple comparison.

A. Dataset Distribution and Preprocessing Results

The dataset acquisition and merging process yielded a total of 2,326 validated images, consisting of 1,405 images from TACO (out of 1,500 source images; 95 images could not be downloaded due to inactive Flickr links), 491 images from RecyBat24, and 335 images from Food Waste Detection (Roboflow Universe). After remapping into 4 classes and a stratified 70:15:15 data split to keep minority-class proportions represented in the validation and test sets, the distribution shown in Table 2 was obtained.

TABLE 2.
IMAGE COUNT DISTRIBUTION PER CLASS

Class	Train	Val.	Test	Total	%
Inorganic	886	189	191	1.266	54,4
B3	350	75	75	500	21,5
Organic	235	50	51	336	14,4
Residue	156	33	35	224	9,6
Total	1.627	347	352	2.326	100

Table 2 confirms that although the dataset was enriched with RecyBat24 and Food Waste Detection, the long-tail characteristic remains evident: the Inorganic class still dominates (54.4%), while Residue remains the least represented class (9.6%). The B3 (21.5%) and Organic (14.4%) classes show a significant proportional increase compared to the original TACO composition, where each originally contributed less than 1% of total annotations, indicating that the data-enrichment strategy in the data collection stage (Section II.A) was effective for the B3 class but did not fully resolve the imbalance in the Residue class.

TABLE 3.
OBJECT-LEVEL ANNOTATION DISTRIBUTION PER CLASS

Class	Train	Val	Test	Total	%
Inorganic	2,425	479	490	3,394	50.8
Organic	1,021	225	164	1,410	21.1
Residue	1,025	155	186	1,366	20.4
B3	366	75	75	516	7.7
Total	4,837	934	915	6,686	100

Table 3 reports the same distribution counted at the object-annotation level, using class-instance counts validated against Ultralytics' own dataset scan (which automatically removes a small number of duplicate label rows present in the RecyBat24-derived files, discussed further in the Limitations subsection of Section IV). At the object level, B3 is in fact the scarcest class (516 instances, 7.7%), with a majority-to-minority imbalance ratio of approximately 6.6, contrasting with B3's comparatively larger 21.5% share at the image level in Table 2. This gap arises because B3 images average only about 1.0 annotated object per image, consistent with RecyBat24's single-battery photography style, whereas Residue images average about 6.1 objects per image, reflecting cluttered, multi-item street-litter scenes typical of TACO. This distinction matters for interpreting Section III.C: Residue's weak performance therefore is not attributable to having fewer training instances overall than B3, but plausibly instead reflects the greater visual clutter and heterogeneity present within each Residue image.

B. YOLOv11 vs YOLO26 Training Results

All three scenarios were trained for a full 50 epochs without early stopping. The YOLOv11 (baseline) scenario required 1.47 hours of training time, with box_loss consistently decreasing from 1.333 at the first epoch to 0.910 at epoch 50, accompanied by a decrease in cls_loss from 3.070 to 0.882. The YOLO26 model (2,375,616 parameters, 5.2 GFLOPs, 5.37 MB checkpoint file) has lower computational complexity and a smaller on-disk footprint than YOLOv11 (2,582,932 parameters, 6.3 GFLOPs, 5.45 MB checkpoint file), consistent with YOLO26's claimed architectural efficiency [12]. The optimizer was automatically selected by Ultralytics (AdamW, lr0=0.00125, momentum=0.9) for all scenarios to keep training conditions equal.

TABLE 4.
OVERALL METRICS ON THE TEST SET

Scenario	Prec.	Rec.	mAP50	mAP50-95
YOLOv11	0,584	0,495	0,507	0,385
YOLO26-Def	0,567	0,508	0,506	0,393
YOLO26-W	0,534	0,511	0,502	0,383

In aggregate (Table 4), all three scenarios produced relatively comparable overall performance on mAP50 (range 0.502–0.507), with a small difference across scenarios (Δ = 0.005). YOLO26 (Default) recorded the highest mAP50-95 (0.393), slightly surpassing the YOLOv11 baseline (0.385), while YOLO26 (Weighted) recorded the lowest mAP50-95 and precision among the three (0.383 and 0.534). Recall shows a consistently increasing trend from YOLOv11 to YOLO26 (Default) to YOLO26 (Weighted), indicating that the STAL mechanism in YOLO26 does contribute to the model's improved ability to detect previously missed objects, although this improvement was not accompanied by a proportional increase in precision.

C. Per-Class Performance Comparison

The per-class performance breakdown on the test set (352 images, 915 object instances: 490 Inorganic, 164 Organic, 75 B3, 186 Residue) is presented in Table 5 (mAP50) and Table 6 (mAP50-95), with emphasis on the minority B3 and Residue categories in line with the main focus of this study.

TABLE 5.
mAP50 PER CLASS ACROSS SCENARIOS

Class	YOLOv11	YOLO26-Def	YOLO26-W
Inorganic	0,555	0,548	0,546
Organic	0,401	0,408	0,400
B3	0,988	0,987	0,988
Residue	0,085	0,081	0,076

TABLE 6.
mAP50-95 PER CLASS ACROSS SCENARIOS

Class	YOLOv11	YOLO26-Def	YOLO26-W
Inorganic	0,387	0,390	0,389
Organic	0,212	0,227	0,213
B3	0,898	0,908	0,886
Residue	0,044	0,047	0,045

The B3 class consistently achieved very high performance across all three scenarios (mAP50 0.987–0.988), far exceeding the other three classes, despite having the fewest instances of any class overall (516 objects, Table 3) and in the test set specifically (75 instances). This is best explained by B3 objects (batteries/e-waste from RecyBat24) having relatively uniform visual characteristics that are easily distinguished from the background, allowing the model to learn them well even with limited data. The class weighting and copy-paste augmentation strategy in YOLO26-Weighted did not provide a meaningful mAP improvement for the B3 class (mAP50-95 in fact dropped slightly to 0.886, from 0.908 in YOLO26 Default), suggesting that this class's performance has approached the ceiling achievable under current data conditions.

In contrast, the Residue class remained the weakest-performing class across all three scenarios (mAP50 0.076–0.085), even lower than findings from prior research in a similar domain that reported a Residue mAP50 of 0.702 [1]. More importantly, Residue class performance actually decreased monotonically from YOLOv11 baseline (mAP50 0.085) to YOLO26 Default (0.081) to YOLO26 Weighted (0.076). This runs against the initial research hypothesis that the additional class-imbalance handling strategy (cls=1.5 and copy_paste=0.3) would evenly improve minority-class performance. For comparison, the number of Residue instances in the test set (186) is actually greater than B3 (75), suggesting that the low performance on Residue is more likely caused by the visual heterogeneity of that category — under the TACO remapping scheme, the Residue class includes objects with highly varied shapes and textures (tissue, styrofoam, cigarette butts, foam packaging) — rather than data scarcity alone. This is consistent with findings in the literature that copy-paste augmentation works best on objects with relatively uniform shapes but is less effective for classes with high intra-class variation [10][11][14][15].

This structural-heterogeneity hypothesis is supported quantitatively by the bounding-box geometry of ground-truth annotations. Residue objects have a mean normalized area of only 0.0083, roughly 9 times smaller than B3's mean area of 0.0755, confirming that Residue instances are predominantly small objects, which are intrinsically harder to detect regardless of augmentation strategy. More tellingly, Residue's coefficient of variation in object area (area standard deviation divided by area mean) is 4.99, by far the highest of the four classes (B3 = 0.56, Inorganic = 2.33, Organic = 2.74), indicating that Residue instances vary enormously in size relative to their already-small average, from tiny fragments to comparatively larger pieces. B3 also shows the lowest aspect-ratio variability of any class (standard deviation 0.39, versus 1.22–1.68 for the other three classes), giving quantitative support to the earlier claim that B3 objects have relatively uniform visual characteristics, whereas Residue's aspect-ratio variability (1.22) is comparable to the other heterogeneous classes. Together, these geometric statistics indicate that Residue is a structurally difficult class independent of how much training data it receives, consistent with its performance failing to improve under additional class weighting and copy-paste augmentation in Scenario C.

Figure 3 shows a representative example of this failure mode from the test set: a single Residue object (a small white fragment, likely plastic or foam packaging) that is the only annotated object in its image and that YOLO26 (Default) fails to detect entirely at the standard confidence threshold, despite the object being visible upon close inspection. The object's small size and low visual contrast against the sandy, twig-covered background illustrate concretely why such instances are difficult to detect, complementing the aggregate geometric statistics reported above.

Figure 3. Example of a missed Residue detection (false negative) in the test set. The green box marks the ground-truth annotation; YOLO26 (Default) produced no detection for this object at conf=0.25.

Figure 4 shows a representative false positive: a dried leaf lying near an unrelated ground-truth object (a bottle cap, correctly detected as Inorganic) is mistakenly detected as Residue with moderate confidence (0.34), despite not being an annotated object of any class. The leaf's irregular, non-uniform silhouette closely resembles the irregular shapes

typical of Residue items such as tissue or foam fragments, supporting the interpretation that Residue's poor performance stems from a genuinely difficult, heterogeneous visual class boundary rather than simple data scarcity: natural background clutter with irregular shapes can be confused with the target class in both directions, contributing to both the false negatives shown in Figure 3 and false positives such as this one.

Figure 4. Example of a false positive Residue detection. The model (YOLO26 Default) misclassified a dried leaf (red box, conf=0.34) as Residue; the nearby ground-truth object (green box) is a bottle cap correctly identified as Inorganic and is unrelated to this false positive.

To complement the mAP-based comparison in Tables 4 and 5, Table 7 reports Precision, Recall, and F1-score per class for each scenario, since aggregate mAP can obscure how a model actually behaves at a fixed operating point, which is more representative of deployment conditions.

TABLE 7.
PRECISION, RECALL, AND F1-SCORE PER CLASS

Scenario	Class	P	R	F1
YOLOv11	Inorganic	0.630	0.555	0.590
YOLOv11	Organic	0.447	0.380	0.411
YOLOv11	B3	0.952	0.987	0.969
YOLOv11	Residue	0.308	0.057	0.097
YOLO26-Def	Inorganic	0.556	0.567	0.562
YOLO26-Def	Organic	0.502	0.396	0.443
YOLO26-Def	B3	0.928	0.987	0.956
YOLO26-Def	Residue	0.293	0.081	0.126
YOLO26-W	Inorganic	0.524	0.569	0.546
YOLO26-W	Organic	0.464	0.384	0.420
YOLO26-W	B3	0.921	0.987	0.952
YOLO26-W	Residue	0.231	0.108	0.147

The F1-score view sharpens the contrast between classes that mAP alone can mask. B3's F1-score is consistently high (0.95–0.97) and its recall exceeds 0.98 in all three scenarios, confirming that the model detects nearly all B3 instances with few missed detections. Residue shows the opposite pattern: its F1-score never exceeds 0.15, and recall in particular is very low (0.057–0.108), meaning the model misses the large majority of Residue objects rather than merely ranking them lower in confidence, a distinction that overall mAP does not make explicit. Under YOLO26-Weighted, Residue recall does increase in relative terms (0.081 to 0.108) compared to YOLO26-Default, but from such a low base that this improvement does not translate into a materially higher F1-score (0.126 to 0.147) or mAP. Averaging these per-class results without weighting (macro-average) rather than by instance count highlights this imbalance in outcomes: a simple accuracy-weighted view dominated by the largest class (Inorganic) would otherwise understate how poorly the model performs specifically on Residue, which is precisely the class this study is most concerned with.

D. Confusion Matrix Analysis

The normalized confusion matrices at a fixed confidence threshold for the three scenarios are presented in Figure 2, revealing misclassification patterns that cannot be directly observed from mAP values alone. This threshold was set to Ultralytics' standard validation default (confidence = 0.25, IoU = 0.45) and applied identically across all three scenarios; it was not tuned against the test set, so the confusion-matrix patterns reported below reflect a fixed operating point rather than a threshold selected to favor any particular scenario.

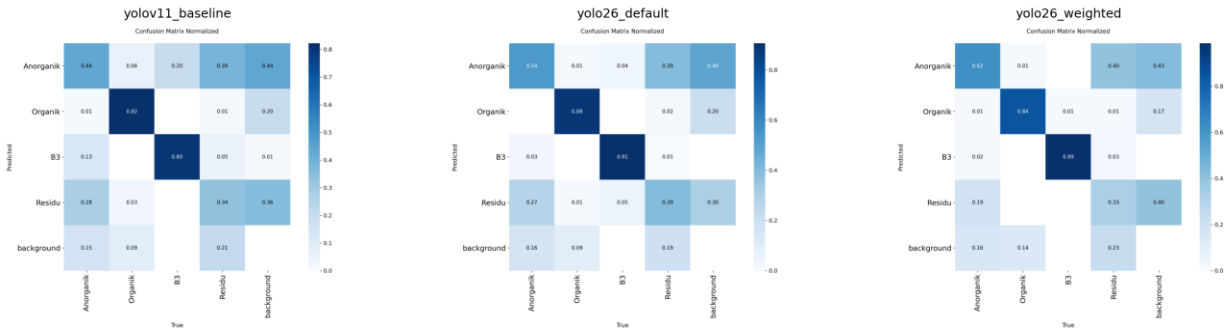

Figure 2. Normalized Confusion Matrix Comparison of YOLOv11 Baseline, YOLO26 Default, and YOLO26 Weighted

For the B3 class, the confusion matrix diagonal increases consistently from YOLOv11 (0.80) to YOLO26 Default (0.91) to YOLO26 Weighted (0.99), with misclassification into the Inorganic class dropping sharply from 0.13 to 0.02. At a fixed confidence threshold, this suggests that the class weighting strategy in YOLO26-Weighted markedly improves

the robustness of B3 classification, even though this improvement is not reflected in mAP50-95 (Table 6), since mAP integrates performance across all confidence thresholds rather than a single operating point. These two metrics complement each other: mAP reflects overall detection ranking quality, whereas the confusion matrix at a fixed threshold is more relevant to practical deployment conditions with a single decision threshold.

Taken together, the B3 and Residue results point to a trade-off rather than a uniform benefit from the added class-imbalance strategy. Increasing the classification loss weight and injecting copy-paste instances biases the model toward predicting the up-weighted classes more confidently, which sharpens the decision boundary for B3, a visually consistent class where the model already had a reasonable feature representation. For Residue, the same bias appears to increase competition with the visually overlapping Inorganic class at the confidence threshold used for evaluation, rather than improving recall on genuinely difficult instances, which is consistent with the drop in both mAP and confusion-matrix diagonal observed for this class under YOLO26-Weighted. This suggests that a single global weighting strategy applied uniformly across minority classes is unlikely to help classes whose weakness stems from visual heterogeneity rather than mere sample scarcity, and that class-specific tuning may be required instead.

For the Residue class, the opposite pattern occurs: the highest confusion matrix diagonal is actually achieved by YOLO26 Default (0.39), followed by YOLOv11 (0.34), with the lowest for YOLO26 Weighted (0.33). Residue misclassification is dominated by two directions: confusion with the Inorganic class (0.19–0.28) and prediction as background/non-detection (0.30–0.40), rather than confusion with B3 or Organic. The same pattern holds across all three scenarios, reinforcing the notion that the root cause of Residue's low performance is structural (related to visual feature overlap between Residue and Inorganic, along with many small Residue objects that go completely undetected), and therefore cannot be resolved solely through loss-weight adjustment and copy-paste augmentation at the current data level.

E. Inference Speed Analysis

Since YOLO26's efficiency claims are especially relevant for deployment on edge/mobile devices without GPU acceleration, inference speed was measured on CPU using the average prediction time over 20 test-set images (Table 8). All timings were obtained on a single AMD EPYC 7B12 CPU core made available through the Google Colab runtime, using the Ultralytics inference pipeline on PyTorch 2.11.0 (CPU build), with each model loaded from its native PyTorch (.pt) checkpoint and run at a batch size of 1 to reflect a single-image, real-time inference scenario rather than batched offline processing. The YOLO26-Weighted scenario was not tested separately because it uses an identical backbone architecture to YOLO26 Default, so its inference speed is assumed to be equivalent.

TABLE 8.
CPU INFERENCE SPEED

Scenario	ms/image	FPS
YOLOv11	173,1	5,8
YOLO26-Def	144,4	6,9

YOLO26 (Default) shows a modestly faster CPU inference speed than the YOLOv11 baseline (144.4 ± 6.2 ms/image versus 173.1 ± 37.2 ms/image across 3 repeated timing trials, or approximately 6.9 FPS versus 5.8 FPS, a gain of roughly 20%, well short of the 2× difference suggested by a single earlier uncontrolled measurement). The YOLOv11 timings also showed considerably higher run-to-run variance than YOLO26, which itself is a relevant efficiency-related observation for deployment. This more modest efficiency gain is nonetheless consistent with YOLO26's lower computational complexity (5.2 GFLOPs versus 6.3 GFLOPs) and the removal of the Distribution Focal Loss (DFL) component along with an end-to-end architecture without Non-Maximum Suppression (NMS), one of YOLO26's main claims [12]. With comparable or slightly better mAP50-95 (Table 4), this finding suggests that YOLO26 (Default) offers a somewhat more favorable, though not dramatically different, accuracy-speed trade-off than YOLOv11 for deployment scenarios on devices with limited computational resources, such as edge-device-based automated waste sorting systems.

Whether 144.4 ms per image (≈6.9 FPS) is adequate for a real deployment depends on the target sorting line's throughput and object spacing, which this study did not measure directly. Industrial single-stream sorting conveyors commonly operate at speeds requiring detection well above 5-10 FPS to avoid missed or overlapping items, so the CPU-only figure reported here should be read as an efficiency comparison between YOLOv11 and YOLO26 rather than a demonstrated real-time capability; validating actual throughput would require testing on the target hardware (e.g., an edge accelerator or GPU) and a physical or simulated conveyor setup, which is left for future work.

IV. CONCLUSION

This study examined the extent to which the ProgLoss and STAL mechanisms in YOLO26 can address class imbalance and small-object detection issues for minority waste categories (B3 and Residue) compared to YOLOv11, through three experimental scenarios on a combined TACO, RecyBat24, and Food Waste Detection dataset. The test results show that the research hypothesis is partially supported. These conclusions are bounded by the specific dataset, class definitions, and single-run training conditions used in this study, and should not be read as a general claim

that YOLO26 outperforms YOLOv11 across waste-detection tasks at large.

At the overall performance level, YOLO26 (Default) reached a slightly higher mAP50-95 than YOLOv11 (0.393 versus 0.385) while running modestly faster on CPU (144.4 ± 6.2 ms versus 173.1 ± 37.2 ms per image, roughly 20% faster with less run-to-run timing variance) — a modest but consistent efficiency advantage for edge deployment. The picture is more complicated at the class level. For B3, the weighting strategy in YOLO26-Weighted did make classification noticeably more decisive at a fixed confidence threshold, with the confusion-matrix diagonal climbing from 0.80 to 0.99, though this class was already performing close to its ceiling in all three scenarios (mAP50 ≥ 0.987), so there was not much room left to gain. Residue is the more telling case. Rather than improving, its performance dropped step by step across the three scenarios, from mAP50 0.085 under YOLOv11 to 0.081 under YOLO26 Default and 0.076 under YOLO26 Weighted, and the errors followed the same pattern each time: Residue objects mistaken for Inorganic or missed outright. That consistency points to a structural problem — overlapping visual features and highly varied object shapes within the class — which loss-weight tuning or a newer architecture alone cannot fix given the data currently available.

Based on these findings, for practical implementation in automated waste sorting systems, YOLO26 (Default) is recommended as a more balanced choice between accuracy and computational efficiency compared to YOLOv11, particularly for detecting the B3 category, which is critical in hazardous waste handling. The simple class weighting strategy (increased classification loss weight and copy-paste augmentation) is not recommended for uniform application across all minority classes, as it proved effective only for classes with low visual variation (B3) but counterproductive for classes with high visual variation (Residue).

To assess whether the reported metrics reflect a stable effect rather than run-to-run noise, the YOLO26 (Default) scenario was retrained from scratch three additional times using different random seeds (42, 123, 777), keeping all other training conditions identical. Across these three runs plus the original run, the overall test-set mAP50-95 was highly stable (mean = 0.389, standard deviation = 0.003), as was the per-class mAP50-95 for Anorganik (0.381 ± 0.006), Organic (0.218 ± 0.011), B3 (0.911 ± 0.005), and, most notably, Residue (0.046 ± 0.001). The very small standard deviation for Residue in particular indicates that its consistently weak performance is a robust, reproducible property of the model on this dataset rather than an artifact of a single unlucky training run, reinforcing the structural-heterogeneity interpretation discussed above. Additional limitations should also be acknowledged. First, the merged dataset combines TACO, RecyBat24, and Food Waste Detection, three sources collected under different protocols, camera conditions, and annotation guidelines; while categories were mapped to a common four-class scheme, residual label inconsistency across sources cannot be fully ruled out and was not formally quantified in this study. Separately, Ultralytics' dataset-caching step reported a small number of exact-duplicate label rows (typically 2, occasionally 4–6, per affected image) within RecyBat24-derived label files, likely originating from the RecyBat24 annotation export used as input to the merging script rather than from the merging process itself; these are removed automatically during both training and validation, so the object-instance counts reported in Table 3 and all evaluation metrics in this study already reflect the deduplicated, corrected counts. A perceptual-hash check for near-duplicate images across the merged dataset identified 4 pairs of highly similar images split across training and evaluation subsets (train-test or train-validation), all originating from RecyBat24 and consistent with multiple photographs of the same physical battery taken from different angles; this represents a small (4 of 2,326 images, <0.2%) but non-zero risk of data leakage that likely has a negligible effect on the aggregate results reported here but should be eliminated in future work by splitting the data at the source-object level rather than the image level. Second, the overall dataset size (2,326 images) and, in particular, the small number of test instances for some classes, particularly B3 (75), Organic (164), and Residue (186), limit the precision of the per-class estimates reported here. Third, this study evaluates detection accuracy and CPU inference speed only; it does not include validation on a physical sorting line or with camera and lighting conditions representative of an operational waste-management facility, so deployment readiness claims remain provisional. Finally, the pronounced and consistent underperformance on the Residue class across all scenarios indicates that architectural or loss-weighting changes alone are insufficient for this category under the current data conditions.

Future research should focus specifically on addressing the Residue class, including: (1) adding Residue data from additional sources that specifically capture variations in household residual waste objects; (2) evaluating augmentation strategies better suited to objects with high shape variation, such as mixup or style-transfer augmentation, as alternatives to copy-paste; (3) an ablation study that isolates the individual contributions of class weighting and copy-paste augmentation, and separately of the ProgLoss and STAL mechanisms in YOLO26, given that this study only evaluated these components jointly; (4) testing on physical edge devices (rather than CPU simulation) and, ideally, on a real or simulated sorting line to validate the deployment feasibility of YOLO26 in a real-scale automated waste sorting system; (5) hard-negative mining and a formal audit of annotation consistency across the three merged data sources to reduce potential label noise; (6) evaluating detection performance separately by object size (small, medium, large) to directly test whether STAL's claimed small-object benefit is realized in this dataset; and (7) re-splitting the RecyBat24-derived portion of the dataset at the physical-object level

rather than the image level to eliminate the small near-duplicate leakage risk identified in this study.

REFERENCES

[1] Lingga Kurnia Ramadhani and Bajeng Nurul Widyaningrum, "Implementation of YOLO v11 for Image-Based Litter Detection and Classification in Environmental Management Efforts," Journal of Applied Informatics and Computing (JAIC), vol. 9, no. 3, pp. 617-624, 2025, doi: 10.30871/jaic.v9i3.9213.

[2] Q. Zhang, Q. Yang, X. Zhang, Q. Bao, J. Su, and X. Liu, "Waste image classification based on transfer learning and convolutional neural network," Waste Management, vol. 135, pp. 150-157, 2021, doi: 10.1016/j.wasman.2021.08.038.

[3] J. Gunaseelan, S. Sundaram, and B. Mariyappan, "A Design and Implementation Using an Innovative Deep-Learning Algorithm for Garbage Segregation," Sensors, vol. 23, no. 18, 7963, 2023, doi: 10.3390/s23187963.

[4] F. Fotovvatikhah, I. Ahmedy, R. M. Noor, and M. U. Munir, "A Systematic Review of AI-Based Techniques for Automated Waste Classification," Sensors, vol. 25, no. 10, 3181, 2025, doi: 10.3390/s25103181.

[5] P. A. Rajeev, V. Dharewa, D. Lakshmi, G. Vishnuvarthanan, J. Giri, T. Sathish, and M. Alrashoud, "Advancing e-waste classification with customizable YOLO based deep learning models," Scientific Reports, vol. 15, no. 1, 18151, 2025, doi: 10.1038/s41598-025-94772-x.

[6] X. C. Acaro Chac\u00f3n, F. Lo Scudo, G. Cappuccino, and C. Dodaro, "RecyBat24: a dataset for detecting lithium-ion batteries in electronic waste disposal," Scientific Data, vol. 12, no. 843, 2025, doi: 10.1038/s41597-025-05211-5.

[7] M. H. Dipo, F. Al Farid, M. S. Al Mahmud, M. Momtaz, S. Rahman, J. Uddin, and H. A. Karim, "Real-Time Waste Detection and Classification Using YOLOv12-Based Deep Learning Model," Digital, vol. 5, no. 2, 19, 2025, doi: 10.3390/digital5020019.

[8] Y. Zhou, L. Lin, and T. Wang, "Garbage classification detection system based on the YOLOv8 algorithm," AIP Advances, vol. 14, no. 12, 125012, 2024, doi: 10.1063/5.0233953.

[9] S. Riyadi, A. D. Andriyani, and A. M. Masyhur, "Classification of Recyclable Waste Using Deep Learning: A Comparison of Yolo Models," Revue d'Intelligence Artificielle, vol. 38, no. 4, pp. 1089-1096, 2024, doi: 10.18280/ria.380404.

[10] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance Problems in Object Detection: A Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 10, pp. 3388-3415, 2021, doi: 10.1109/TPAMI.2020.2981890.

[11] Y. Liu, P. Sun, N. Wergeles, and Y. Shang, "A survey and performance evaluation of deep learning methods for small object detection," Expert Systems with Applications, vol. 172, 114602, 2021, doi: 10.1016/j.eswa.2021.114602.

[12] R. Sapkota, R. H. Cheppally, A. Sharda, and M. Karkee, "YOLO26: Key Architectural Enhancements and Performance Benchmarking for Real-Time Object Detection," arXiv:2509.25164, 2026. [Not yet Scopus-indexed — preprint, included as a supporting technical reference]

[13] P. F. Proença and P. Simões, "TACO: Trash Annotations in Context for Litter Detection," arXiv:2003.06975, 2020. [Dataset source, not Scopus-indexed]

[14] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2021, pp. 2918-2928, doi: 10.1109/CVPR46437.2021.00294.

[15] X. Yu, F. Li, Y. Liu, and A. Wang, "A Multi-Stage Adaptive Copy-Paste Data Augmentation Algorithm Based on Model Training Preferences," Electronics, vol. 12, no. 17, 3695, 2023, doi: 10.3390/electronics12173695.