Simen Owen Academic Proceedings Series

4th International Conference on Electronics, Engineering, Computer Science and Applied Development (EESD 2026)

Article

Data-Efficient Computer Vision for Environmental Monitoring in Resource-Constrained Settings

Yiqi Pan[1,*]

[1] Computer Engineering and Science, Shanghai University, Shanghai, China
* Correspondence: Yiqi Pan, Computer Engineering and Science, Shanghai University, Shanghai, China

Abstract: Computer vision can expand the scope of garbage detection and reduce the reliance on continuous manual observation, and is expected to be applied in a wider range of environmental monitoring. However, these methods still face some problems in practical applications, such as limited labeled data, significant differences in the number of samples for different categories, and limited computing power of edge devices. This study conducted experiments based on the public Trash Annotations in Context dataset. This dataset contains 1500 outdoor images and 4784 garbage instances. The research mainly focused on how to maintain good detection performance with limited labeled data and device resources. To this end, the study adopted uncertainty-diversity active selection, category-balanced focal training, and a lightweight SSDLite–MobileNetV3 model, and conducted tests under 5%, 10%, 20%, 40%, and 100% of the labeled data. When only using 20% of the labeled data, the mAP@[0.5:0.95] of the complete method reached 0.319, equivalent to 89.4% of the performance of the fully supervised SSDLite benchmark model. When the labeled data increased to 40%, this proportion further improved to 97.8%. Category-balanced training also improved the detection performance of minority categories, with the average precision increasing from 0.167 to 0.224 and the macro recall rate increasing from 0.488 to 0.568. In terms of device deployment, the latency of the INT8 model was reduced by 33.8% compared to the FP32 model, the model size was reduced by 70.5%, the peak memory usage decreased by 27.4%, at the cost of a 1.2 percentage point decrease in mAP. In summary, this method can maintain good detection performance with less labeled data and limited device resources, but there is still a certain trade-off between accuracy and deployment cost, and whether it can be applied to other environmental monitoring tasks still needs further verification.

Keywords: Data-efficient computer vision; environmental monitoring; litter detection; active learning; class imbalance; edge deployment

Received: 24 June 2026
Revised: 16 August 2026
Accepted: 31 August 2026
Published: 05 September 2026

1. Introduction

Computer vision has become an important tool for environmental monitoring as it can extract time and space-related information from a large number of images [1]. Currently, it has been used for garbage inspection in roads and public parks, waste identification in areas such as riverbanks and beaches, environmental monitoring around protected areas, analysis of vegetation coverage changes, and urban environmental anomaly detection. Drones and mobile devices have further expanded the application scenarios of image recognition [2]. These technologies can expand the monitoring scope and support more frequent environmental assessments. However, high-performance visual systems usually require a large amount of labeled data and strong computing capabilities [3]. In resource-limited environments, these conditions are often difficult to meet. For example, data labeling may require a large amount of human input, the number

of samples for different categories may vary greatly, and the targets in the environment may be small in size, partially occluded, or mixed with complex backgrounds. Additionally, some monitoring systems need to be deployed on smartphones or embedded devices, which have certain limitations in memory, processing power, and energy.

Bottles, cans and other common garbage account for a large proportion of the training data, while samples of paper and rare containers are very few [4]. The model is easy to recognize common garbage but has difficulty in stably identifying categories with fewer samples. After the annotation data is reduced, this problem becomes more obvious because when randomly selecting data, the minority categories and complex scenes may not be included [5]. Increasing the model size may improve the detection effect, but it will also make the model larger, slower to run, and consume more memory [6]. For devices with limited computing power, these requirements are difficult to meet [6]. This study uses the "Trash Annotations in Context" dataset to investigate outdoor garbage detection and discusses it in environmental monitoring scenarios such as vegetation monitoring, abnormal environment recognition, and urban change detection.

This study examined the performance of the model under different conditions after reducing the labeled data and using lightweight detectors. This method might reduce the resources required for training and operation. However, in complex scenarios, the model may have greater difficulty in detecting rare categories and small-sized target garbage. Using more annotation data and larger models helps improve the overall detection effect, but it will increase the running burden of the device. Therefore, this study compares the detection results under different annotation data quantities and device conditions, while also paying attention to the recognition effect of minority categories and the time and memory usage of the model during operation.

This study is centered around several research core issues. RQ1 focuses on how much of the original detection performance the model can retain when only some of the training images are labeled. RQ2 examines whether class-balanced training can improve the detection performance of rare categories without significantly affecting common categories. RQ3 investigates how the accuracy of the model changes and how much computational resources can be saved when using lightweight detectors and low-precision deployment. These questions cover the entire detection process from data selection, model training to device deployment.

The main contributions of this study are as follows. The study established an evaluation method for outdoor garbage detection, taking into account the amount of labeled data, category distribution, and equipment operation costs. On this basis, a data-efficient method was proposed, combining uncertainty-diversity active selection, class-balanced training, and lightweight deployment. The study also tested with different amounts of labeled data and compared the detection results of various categories. Combining ablation experiments and edge device tests, the relationship between the amount of labeled data, detection performance, and computational costs was analyzed.

2. Related Work and Research Gap

2.1. Computer Vision for Environmental Monitoring

Computer vision can serve environmental monitoring in various ways. Waste detection typically employs image classification, object detection, and instance segmentation to identify waste in streets, parks, beaches, riverbanks, and natural landscapes [7]. The Trash Annotations in Context (TACO) dataset was established for outdoor waste detection and segmentation. Although the data scale is limited, it still reflects the difficulties in data annotation for outdoor waste detection. Studies on vegetation and land cover often use multispectral, aerial, or satellite images to create pixel-level maps, estimate vegetation canopies, and predict changes [8]. Urban monitoring typically compares images from different times to identify changes in buildings, land use,

and infrastructure. Environmental anomaly detection focuses on areas in the image that deviate from normal conditions, without the need to preset fixed target categories [9].

The data and detection objects used for these tasks are different, resulting in varying final results. Therefore, this study focuses on outdoor waste detection as an environment monitoring task with limited resources. The research results cannot be directly applied to vegetation prediction, general anomaly detection, or urban change analysis.

2.2. Data-Efficient Object Detection

Transfer learning, active learning, semi-supervised learning, few-shot detection, and self-supervised pre-training can all reduce the reliance on a large amount of manual annotation [10]. Transfer learning uses the features learned from large datasets to initialize the detector. Semi-supervised detection uses both labeled and unlabeled images and utilizes unlabeled data through pseudo-labels, a common method being teacher-student learning [11]. Few-shot detection focuses on classes with only a few labeled samples. Self-supervised pre-training learns general features using data without artificial class labels.

Active learning is suitable for the scenario of this study as it can control the amount of data that needs to be labeled and select the next batch of samples to be labeled from unlabeled images. Active learning in object detection usually combines the uncertainty of the model and the differences between samples to reduce redundant labeling. However, if only the images with the least confidence of the model are selected, the selected images may come from similar scenes [12]. At the same time, images with a large number of targets are more likely to be selected, while rare categories may not be selected. Therefore, this study considers both the uncertainty of the model and the differences in features space of the images, and observes whether the data selected in each round covers different categories.

2.3. Long-Tailed Environmental Data

The categories in environmental datasets are often unevenly distributed. In TACO, some fine-grained category samples and annotations have very few, so the overall detection results of the model are prone to be affected by common categories. To address category imbalance, common methods include random over-sampling, class-aware sampling, inverse frequency weighting, Focal Loss, Class-Balanced Loss, and logit adjustment [13]. Focal Loss reduces the influence of easily classified samples on training, allowing the model to pay more attention to difficult-to-identify samples. Class-Balanced Loss adjusts weights based on the number of valid samples for each category, avoiding excessive adjustment caused by inverse frequency weighting [14]. Logit adjustment changes the classification boundaries of the model based on the frequency of category occurrence.

In long-tail waste detection, only looking at mAP may not reflect the model's performance on minority categories, as the overall score is easily influenced by common categories. Therefore, experiments also need to report Macro-AP, Macro-Recall, rare-class AP, common-class AP, and per-category AP. These indicators can help determine whether the model has truly improved the detection effect on minority categories or mainly enhanced the performance of common categories.

2.4. Lightweight Vision and Edge Deployment

Resource-constrained visual systems typically use smaller backbone networks and detection heads, and also combine pruning and low-precision inference to reduce computational costs [15]. MobileNetV3 is designed for mobile devices and, when combined with SSDLite, can be used for target detection tasks with limited computing resources. Structured pruning and FP16 or INT8 quantization can further reduce model size and computational costs, and support deployment formats such as ONNX.

However, the number of parameters and FLOPs can represent the actual running speed on the device. Hardware support, memory access, numerical precision, runtime optimization, and device architecture all affect inference speed. Therefore, when

evaluating deployment effectiveness, in addition to detection accuracy, attention should also be paid to latency, frames per second, peak memory, and model size.

2.5. Research Gap

There are still several aspects in existing research that are worth further discussion. The research on annotation efficiency mostly focuses on overall accuracy and pays less attention to the different impacts on rare and common categories after reducing the annotation data. In addition, active learning and class balance are usually studied separately. The former is responsible for selecting the images worth labeling, while the latter adjusts the training process after obtaining labels. It is unclear whether combining the two can improve performance on small-scale long-tail datasets. Finally, data efficiency and deployment efficiency are often evaluated separately. Reducing the annotation data does not reduce the computational cost during model inference. Model compression may make it more difficult to detect small targets and rare categories. This study evaluates the annotation data volume, category detection effect, and running costs on edge devices together to more comprehensively analyze the relationships between these factors.

3. Dataset and Methodology

3.1. TACO Dataset and Category Reorganization

This study utilized the TACO dataset, which contained 1,500 outdoor images and 4,784 instances of garbage. These images were taken in various environments such as roads, parks, beaches, and woodlands, and most were captured by mobile devices, thus being affected by factors such as lighting, shooting distance, background, and angle. Each garbage instance had a category label, bounding box, and a polygon mask in COCO format. The original data included 60 fine categories, which were divided into 28 subcategories. Among them, samples for some categories were scarce, while items like bottles, cans, films, and packaging materials were more common. The garbage in the images was often small in size, could be obscured, had transparent materials, be damaged, or be very similar to the surrounding environment.

To ensure a more stable sample size for each category, this study combined the original categories into six groups. Bottles included plastic bottles and glass bottles; cans included beverage cans, food cans, and spray containers; bags and packaging materials included shopping bags, films, packaging bags, and packaging membranes; cups and containers included cups and disposable containers; paper and cardboard included paper, cardboard boxes, paper bags, and paperboard; other categories were uniformly classified as "other garbage". This classification method was determined before data division and remained unchanged throughout the experiment.

After merging, there were 1,280 instances of bags and packaging materials, 1,040 instances of bottles, 900 instances of other garbage, 650 instances of cans, 500 instances of cups and containers, and 414 instances of paper and cardboard. The data was divided into 1,050 training images, 225 validation images, and 225 test images. The validation set and test set remained unchanged, and the active selection was only performed on the training set.

This study divided the instances in the training set into three frequency levels. Categories with 700 or more instances were classified as Common, those with 400 to 699 instances as Medium, and those with fewer than 400 instances as Rare. Therefore, bags and packaging materials, and bottles belonged to Common, other garbage, cans belonged to Medium, cups and containers, and paper and cardboard belonged to Rare. Table 1 summarizes the merged categories, frequency levels, data division, and annotation budget.

Table 1. Dataset Composition, Reorganized Litter Categories, and Experimental Partition

Panel a. Reorganized Categories				
Reorganized category	**Representative original labels**	**Instances**	**Proportion**	**Frequency group**
Bag and wrapper	Plastic film, carrier bag, packet, wrapper	1,280	26.76%	Common
Bottle	Plastic bottle, glass bottle	1,040	21.74%	Common
Other litter	Caps, straws, cigarettes, fragments, miscellaneous plastic	900	18.81%	Medium
Can	Drink can, food can, aerosol	650	13.59%	Medium
Cup and container	Paper cup, plastic cup, food container	500	10.45%	Rare
Paper and cardboard	Paper, cartons, paper bags, cardboard	414	8.65%	Rare
Total	-	4,784	100.00%	-
Panel B. Experimental Partition and Configuration				
Setting		**Value**		
Training pool		1,050 images, 70%		
Validation set		225 images, 15%		
Test set		225 images, 15%		
Annotation budgets		5%, 10%, 20%, 40%, and 100%		
Labeled training images		53, 105, 210, 420, and 1,050		
Random seeds		13, 21, 42, 87, and 102		
Input resolution		320 × 320 pixels		

3.2. Label-Budget Simulation

During the sample collection process, the annotation information of the training set was concealed. In the t-th round of collection, the training data was divided into two parts: annotated and unannotated:

$$\mathcal{D}_{\text{pool}} = \mathcal{D}_L^{(t)} \cup \mathcal{D}_U^{(t)}, \mathcal{D}_L^{(t)} \cap \mathcal{D}_U^{(t)} = \emptyset. \quad (1)$$

The experiment began with 53 labeled images, accounting for about 5% of the training set. Then, more labeled images were gradually released, increasing the data volume to 105, 210, and 420, corresponding to 10%, 20%, and 40% of the budget respectively. The experiment also used a complete set of 1,050 training images for fully supervised training as a performance reference. In each round of the experiment, only the selected images and their labels were used.

This setup simulates the gradual increase of labeled data using the existing labeled data, without simulating the real manual labeling process.

Figure 1 shows the entire experimental process, including category merging, fixed data division, and the settings of different labeling budgets, as well as uncertainty-diversity selection, category balance training, and three deployment methods: FP32, FP16, and INT8. The experimental records the overall accuracy of the model, the performance of rare categories, inference latency, memory usage, and model size. The following sections will introduce the detector, data selection method, loss function, and deployment settings.

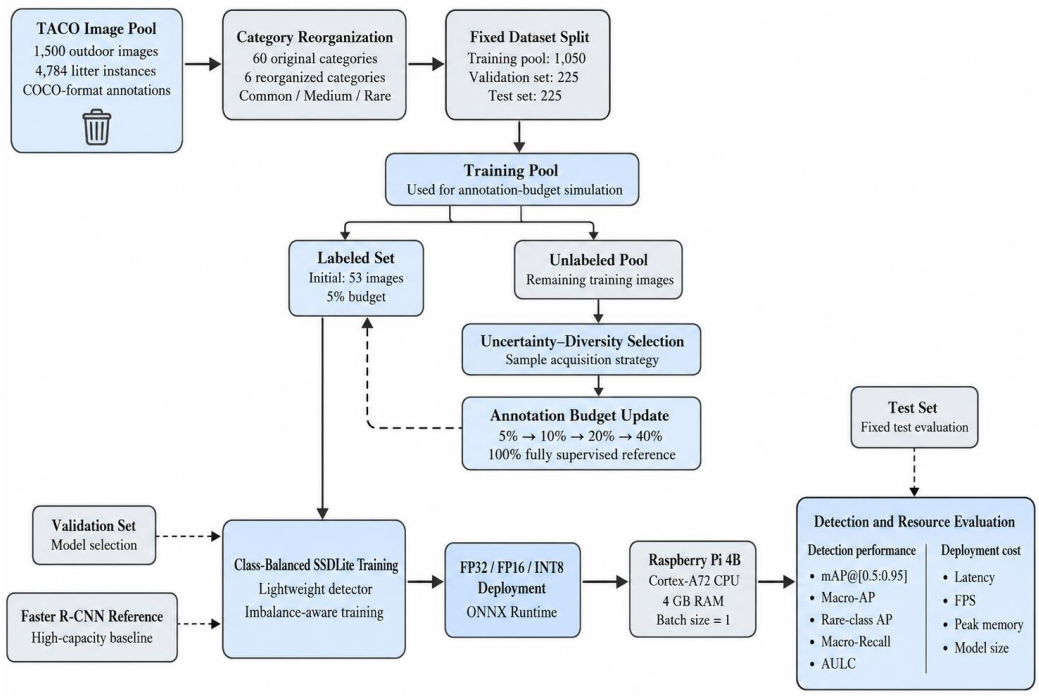

Figure 1. Overall Framework for Data-Efficient Litter Detection under Annotation, Class-Imbalance, and Deployment Constraints.

3.3. Lightweight Detection Backbone

SSDLite320–MobileNetV3-Large is as the main detector. MobileNetV3 offers a lightweight feature extraction network, suitable for running on mobile devices, while SSDLite employs a lightweight single-stage detection head. The model input size is 320 × 320, and it is initialized using COCO pre-trained weights.

Faster R-CNN–ResNet50-FPN serves as a reference for high-capacity models. The batch size of SSDLite is 16, while that of Faster R-CNN is 4. Both models use the AdamW optimizer, with an initial learning rate of 3×10^{-4} and weight decay of 10^{-4}. At the beginning of training, 5 warm-up rounds are conducted, followed by a cosine decay to adjust the learning rate. The training is conducted for a maximum of 120 cycles. If the mAP on the validation set does not continue to improve after the 15th cycle, the training is prematurely terminated.

During the training process, various simple data variations were added to increase the sample differences. Images are horizontally flipped with a 0.5 probability, and are randomly rotated by ±10 or scaled to 0.8 to 1.2 times the original size. The brightness and contrast are adjusted within a ±20% range of the original values.

3.4. Uncertainty–Diversity Active Selection

Each unlabeled image will receive a scoring value:

$$S(x) = \lambda \tilde{U}(x) + (1 - \lambda)\tilde{D}(x), \quad (2)$$

Among them, $\tilde{U}(x)$ and $\tilde{D}(x)$ represent the normalized uncertainty score and diversity score respectively. The balance coefficient is set as $\lambda = 0.6$.

Uncertainty is calculated based on the average predicted entropy of the retained detection results:

$$U(x) = \frac{1}{N_x} \sum_{j=1}^{N_x} \left[- \sum_{c=1}^{6} p_{j,c} \log\left(p_{j,c} + 10^{-8}\right) \right], \quad (3)$$

Among them, N_x represents the number of candidate objects with a confidence level higher than 0.05. Before performing non-maximum suppression, each image retains at most 100 candidates.

The diversity is calculated based on the 960-dimensional aggregated features extracted by MobileNetV3:

$$D(x) = \min_{z \in \mathcal{D}_L^{(t)}} \| f(x) - f(z) \|_2. \quad (4)$$

The greater the feature distance between the image and the current annotation set, the higher the diversity score. The image with the highest score is selected until the annotation budget for the next stage is reached. Under the same conditions, the experiment compared the methods of random selection, using only uncertainty, using only diversity, and combining both.

3.5. Class-Balanced Detection Loss

Class weights were calculated based on the number of valid samples:

$$w_c = \frac{1-\beta}{1-\beta^{n_c}}, \quad (5)$$

Among them, n_c represents the number of labeled instances of category c, and $\beta = 0.999$. The calculated weights are normalized so that the average weight of all categories is 1, and they are recalculated after each round of adding labeled data.

During training, the category weights are incorporated into the Focal Loss:

$$L_{\mathrm{CBF}} = -\frac{1}{M} \sum_{i=1}^{M} \sum_{c=1}^{6} w_c \, y_{i,c} (1 - p_{i,c})^2 \log\left(p_{i,c} + 10^{-8}\right), \quad (6)$$

Among them, M represents the number of matching anchor points, $y_{i,c}$ represents the target indicator, and $\gamma = 2$. The bounding box regression still uses the standard smooth L_1 loss. Since the category reweighting is only used during the training stage, it will not increase the computational cost during model inference.

3.6. Deployment Optimization

The deployment experiment compared the performance of Faster R-CNN FP32, the baseline SSDLite FP32, and the complete framework under FP32, FP16, and INT8. All models were exported in ONNX opset 17 format and ran in ONNX Runtime 1.18.1. INT8 used static quantization and 100 images from the training set were extracted for calibration.

The experiment was conducted on a Raspberry Pi 4B, which is equipped with a quad-core Cortex-A72 CPU and 4 GB of memory. During the test, 4 CPU threads were used and the batch size was 1. Each model was run 30 times for preheating before measuring 200 times. Each test started with image processing, followed by model inference and result processing to obtain the final result. Result processing included adjusting image size, normalization, bounding box decoding, and non-maximum suppression, with an IoU threshold of 0.5. The experiment recorded the model's running delay, frame rate, memory usage, and file size, as well as the number of parameters and the number of multiplication-accumulation operations.

3.7. Evaluation and Statistical Protocol

The performance is evaluated using mAP@0.5, mAP@[0.5:0.95], Macro-AP, Macro-Recall, rare category AP, common category AP, and category AP. The annotation efficiency is measured by the area under the label-performance curve:

$$\mathrm{AULC} = \sum_{b=1}^{4} \frac{P_b + P_{b+1}}{2} (r_{b+1} - r_b), \quad (7)$$

Among them, $r_b \in \{0.05, 0.10, 0.20, 0.40, 1.00\}$ represents the annotation ratio, and P_b represents the corresponding detection score.

For each setting, experiments were repeated using 5 random seeds, and the results were expressed as the average ± standard deviation. A two-sided paired t-test was used to compare different methods with the random sampling baseline, with the significance level set at $\alpha = 0.05$. Holm correction was used for multiple comparisons.

4. Results

4.1. Overall Performance under Different Annotation Budgets

Under random sampling, mAP@[0.5:0.95] gradually increased from 0.142 at a 5% annotation budget to 0.198, 0.261, 0.318, and 0.357 at 10%, 20%, 40%, and 100% budgets respectively. When only uncertainty or diversity selection was used, the results increased to 0.149, 0.215, 0.281, 0.331, 0.357, and 0.151, 0.219, 0.286, 0.334, and 0.357 respectively.

When both were combined, mAP further reached 0.157, 0.231, 0.302, 0.342, and 0.357. After adding the category balance focal training, the complete framework reached 0.162, 0.243, 0.319, 0.349, and 0.366 under five budgets.

With a complete framework under a 20% annotation budget, it has achieved a performance of 87.2% for the fully supervised model, equivalent to 89.4% of the fully supervised SSDLite baseline. When the budget is increased to 40%, these two ratios rise to 95.4% and 97.8% respectively. Compared with random sampling, the improvement at 20% budget was the greatest, with mAP being 5.8 percentage points higher; at 5%, 10%, and 40% budgets, it was 2.0, 4.5, and 3.1 percentage points higher respectively. The results show that when the annotation data is at a low to medium level, the benefits brought by active selection are more obvious.

Figure 2(b) shows that when the annotation data is scarce, the performance of rare categories is more susceptible to influence. Under random sampling, the AP of rare categories gradually increases from 0.081 at a 5% budget to 0.252 at a 100% budget. Uncertainty-diversity selection increases the AP to 0.089, 0.134, 0.191, 0.229 and 0.252 under the same budget. After adding category balance training, the complete framework further reaches 0.103, 0.158, 0.224, 0.259 and 0.273. At a 20% budget, category balance training makes the AP of rare categories 3.3 percentage points higher than active selection, and the overall mAP also increases by 1.7 percentage points.

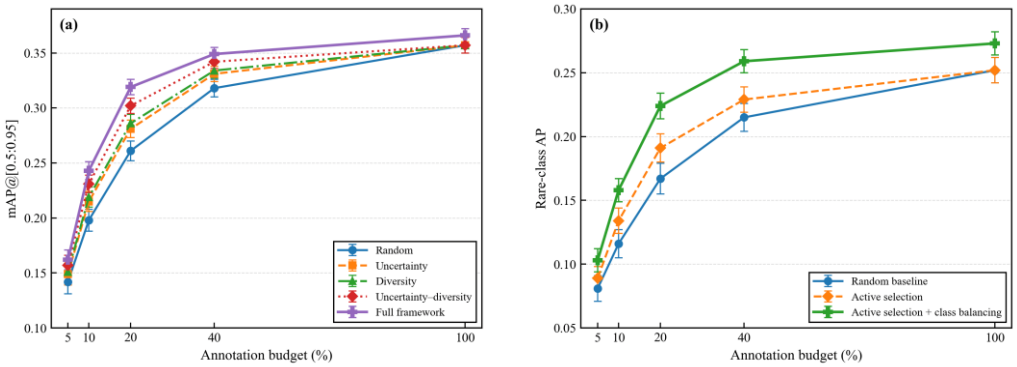

Figure 2. Detection Performance under Different Annotation Budgets: (A) Overall mAP and (B) Rare-Class AP. Error Bars Represent Standard Deviations over Five Random Seeds.

AULC also shows the same trend. The AULC of random sampling is 0.292, while using uncertainty or diversity alone is 0.302 and 0.304 respectively. When combined, it increases to 0.310, and the complete framework reaches 0.320, which is 9.5% higher than random sampling.

4.2. Performance on Imbalanced Categories.

As shown in Column A of Table 2, with a 20% annotation budget, the Macro-AP of the complete framework is 0.333. When only active selection is used, it is 0.311, and the SSDLite baseline is 0.272. Therefore, the complete framework achieved better overall performance. The class-level AP values were 0.391 for Bag and wrapper, 0.403 for Bottle, 0.371 for Other litter, 0.385 for Can, 0.229 for Cup and container, and 0.219 for Paper and cardboard. The average AP of the latter two categories was 0.224, which was higher than 0.167 of the baseline.

This improvement mainly came from the increase in recall rate. The recall rate of rare categories increased from 0.382 to 0.493, while the precision rate only decreased from 0.278 to 0.267, indicating that the increase in false positives was not significant. Common errors included identifying paper pieces as packaging materials and recognizing obscured cups as other garbage. After category balance training, the performance of rare categories improved. The AP of common categories only decreased from 0.400 to 0.397, with a very

small change. This indicates that this adjustment has a limited impact on common categories.

Table 2 summarizes the main experimental results of the model, including detection performance, ablation experiments, and device tests. Panel A and Panel B are used to present the main results and the changes brought about by different settings, while Column C focuses on the performance of the model on edge devices.

Table 2. Detection performance, ablation results, and edge-deployment costs.

Panel A. Main detection results

Method	Label budget	mAP@0.5	mAP@[0.5:0.95]	Macro-AP	Rare-class AP	Macro-Recall
Faster R-CNN	100%	0.612	0.381	0.393	0.281	0.646
SSDLite baseline	100%	0.589	0.357	0.369	0.252	0.621
SSDLite baseline	20%	0.441	0.261	0.272	0.167	0.488
SSDLite + active selection	20%	0.482	0.302	0.311	0.191	0.527
Full framework	20%	0.504	0.319	0.333	0.224	0.568
Full framework	40%	0.548	0.349	0.361	0.259	0.603

Panel B. Ablation results at the 20% budget

Active selection	Class-balanced loss	Quantization	mAP	Rare-class AP	AULC
No	No	No	0.261	0.167	0.292
Yes	No	No	0.302	0.191	0.310
No	Yes	No	0.283	0.205	0.303
Yes	Yes	No	0.319	0.224	0.320
Yes	Yes	INT8	0.308	0.214	0.309

Panel C. Edge-deployment results using the fully supervised model

Model	Precision	Size	Parameters	Latency	Peak memory	FPS	mAP	Rare AP
Faster R-CNN	FP32	167.4 MB	41.2 M	1188.0 ms	1560 MB	0.84	0.381	0.281
SSDLite baseline	FP32	13.9 MB	3.44 M	182.6 ms	312 MB	5.48	0.357	0.252
Full framework	FP32	13.9 MB	3.44 M	184.1 ms	318 MB	5.43	0.366	0.273
Full framework	FP16	7.3 MB	3.44 M	153.7 ms	274 MB	6.51	0.362	0.270
Full framework	INT8	4.1 MB	3.44 M	121.8 ms	231 MB	8.21	0.354	0.263

4.3. Deployment Cost and Accuracy–Resource Trade-Off

Table 2's C column compares the deployment results under different models and numerical precisions. The mAP of Faster R-CNN is the highest, but it takes 1,188.0 milliseconds to process one image, with a peak memory usage of 1,560 MB and a model size of 167.4 MB. In contrast, the FP32 version of the complete framework reduces the delay by 84.5%, the model size by 91.7%, and the mAP by only 1.5 percentage points.

In the complete framework, FP16 further reduces the delay by 16.5%, the model size and memory usage also decrease by 47.5% and 13.8% respectively. The mAP and rare category AP decrease by 0.4 and 0.3 percentage points respectively. The compression effect of INT8 is more obvious, with a delay reduction of 33.8%, a model size reduction of 70.5%, a memory usage decrease of 27.4%, and a throughput increase from 5.43 to 8.21 FPS. However, the cost is a 1.2 percentage point decrease in mAP and a 1.0 percentage point decrease in rare category AP.

Figure 3 shows the relationship between accuracy and delay for different models, with the size of the points representing the model storage space.

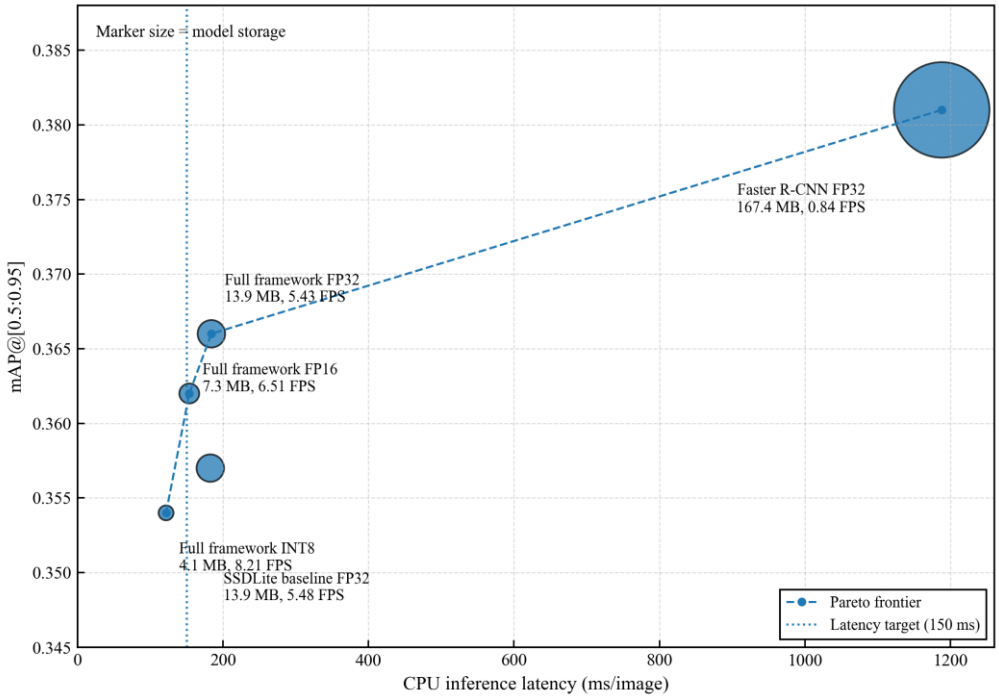

Figure 3. Accuracy–deployment Cost Trade-Off Across Detector Architectures and Numerical-Precision Configurations Measured on Raspberry Pi 4B. Marker Size Represents Model Storage.

From Figure 3, it can be seen that Faster R-CNN, the complete FP32, FP16, and INT8 have different trade-offs between accuracy and speed. The performance of the complete FP16 is better than the baseline SSDLite, reducing delay while maintaining a high mAP. INT8 meets the device requirements of a delay lower than 150 milliseconds and a peak memory lower than 256 MB, but the 8.21 FPS is still below the target of 10 FPS.

4.4. Ablation and Sensitivity Analysis

Figure 4 summarizes the ablation experiments and parameter sensitivity analysis. Figure 4(a) shows that the combination of uncertainty and diversity yields better results. With a 20% annotation budget, the mAP of the two methods used alone were 0.281 and 0.286 respectively, while when combined, it increased to 0.302, while random sampling was 0.261. This indicates that the two selection methods can complement each other. After adding category balance training, the performance of rare categories further improved, indicating that the sample selection and category balance during the training stage address different issues.

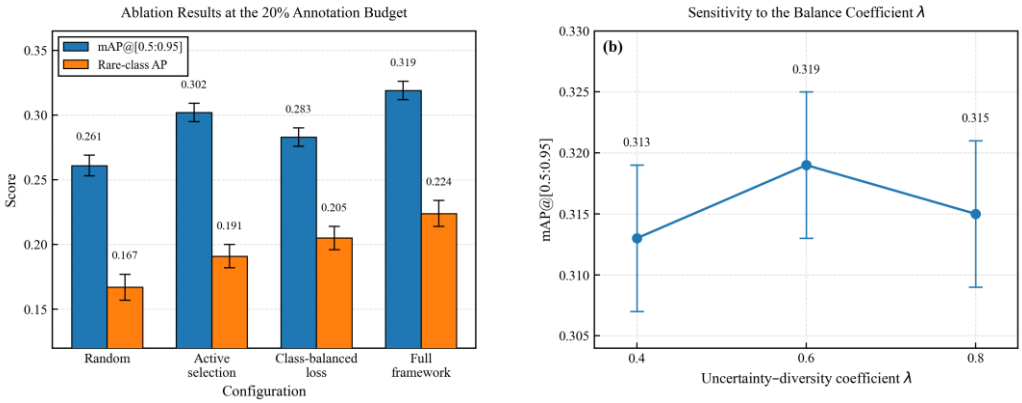

Figure 4. Ablation and Sensitivity Analysis of the Proposed Framework: (A) Ablation Results at the 20% Annotation Budget and (B) Sensitivity of mAP to the Uncertainty–diversity Coefficient Λ.

The results of different loss functions also support this. Under the standard classification loss, the AP of rare categories was 0.191; after adding focal loss or category balance, the AP increased to 0.202 and 0.216 respectively. When combined, it reached 0.224, and Macro-Recall also increased from 0.527 to 0.568. The results show that category balance has a more significant improvement on rare categories, while focal loss brings additional enhancements.

Figure 4(b) shows that when the parameter λ varies between 0.4 and 0.8, the mAP remains between 0.313 and 0.319, reaching the maximum value of 0.319 at 0.6. The changes in β and γ do not cause significant fluctuations. When β is adjusted from 0.99 to 0.9999, the AP of rare categories varies between 0.211 and 0.224; when γ is adjusted from 1 to 3, the AP varies between 0.216 and 0.224. Overall, the model is not sensitive to these parameter changes, and the current settings can maintain relatively stable results.

5. Discussion and Conclusion

5.1. Discussion of the Main Findings

The results show that the selection method of labeled data, class balance, and model size all affect the final detection performance. When the labeling budget is between 10% and 20%, the benefit brought by uncertainty-diversity selection is the most obvious. At this time, the model has learned basic garbage features, but the recognition of some difficult scenarios is still unstable. With 20% of labeled data, the performance of the fully supervised SSDLite baseline can reach 89.4%, and 40% can reach 97.8%. When the budget is low, the performance decline mainly occurs in rare categories such as cups and containers, paper and cardboard, because they are more likely to be missed during data selection. Uncertainty selection is more inclined to pick out images that the model is difficult to judge, while diversity selection can reduce the repetition of similar images. After combining the two methods, AULC increases from 0.292 in random sampling to 0.310, and further reaches 0.320 after adding category balance training.

The improvement of category balance training on rare categories is mainly reflected in recall rate. At a 20% labeling budget, the AP of rare categories increases from 0.167 to 0.224, and the Macro-Recall increases from 0.488 to 0.568. After redistributing the weights of different categories in training, the model can find more paper and containers that were originally prone to be missed. However, the slight decrease in AP also indicates an increase in false positives. For categories with few samples or large internal differences, adjusting the category weights alone is still limited, and the model lacks sufficient samples to learn the features of these categories.

Deployment results show that lightweight models can significantly reduce the computational and storage burden on devices. Compared with Faster R-CNN, the FP32 version of the complete framework reduces the delay by 84.5%, the model size by 91.7%, and the mAP by only 1.5 percentage points. Further using INT8 quantization, the delay, model size, and peak memory are reduced by 33.8%, 70.5%, and 27.4% respectively, but the mAP decreases by 1.2 percentage points, and the AP of rare categories decreases by 1.0 percentage points. This indicates that the impact of quantization may be more obvious in small targets, blurry targets, and occluded targets. At the same time, the number of parameters and FLOPs cannot fully represent the running speed of the model on real devices.

5.2. Practical Implications

This method is more suitable for deployment on devices with limited resources, such as mobile devices, roadside cameras and drones. Even when the network connection is unstable, local inference can still be carried out. The INT8 model can be used initially to screen out the junk, and the results that the model finds difficult to determine can then be handed over to human processing. This study did not test large-scale deployment or cover subsequent applications such as garbage collection, supervision or environmental impact.

5.3. Limitations and Future Work

The data volume of TACO is limited, and there are fewer samples for certain weather conditions, regions, and rare types of garbage. The study merged some of the subcategories to avoid the impact of insufficient samples on the results, but this also reduced the analysis of the subcategories. Active learning simulates the annotation process based on the existing data, so it cannot reflect the time required for real annotations and does not consider the differences among annotators. The experiment only utilized the TACO dataset and a Raspberry Pi. Nighttime, severe occlusion, and adverse weather conditions were not included in the test. If the real annotation process is incorporated and more datasets and devices are used for testing, it will be possible to further examine how the model performs under different conditions and during long-term operation.

6. Conclusion

The experimental results show that 20% of the annotated data can already achieve 89.4% of the performance of fully supervised SSDLite, and 40% reaches 97.8%. Active selection is 4.1 percentage points higher than random sampling. After adding class balance training, the performance of rare categories has improved. The AP and Macro-Recall have increased to 0.224 and 0.568 respectively. INT8 further alleviated the operating burden of the device, reducing the model size while also cutting down the computational cost. Overall, this method has achieved good results in the TACO garbage detection task. However, this cannot be used to determine its applicability to other environments. When evaluating similar systems, in addition to looking at the overall detection effect, the annotation volume and the burden on the actual operation of the device also need to be considered.

References

1. E. Alotaibi and N. Nassif, "Artificial intelligence in environmental monitoring: in-depth analysis," *Discover Artificial Intelligence*, vol. 4, no. 1, p. 84, 2024.
2. S. M. Popescu et al., "Artificial intelligence and IoT driven technologies for environmental pollution monitoring and management," *Frontiers in Environmental Science*, vol. 12, p. 1336088, 2024.
3. S. Avasthi and S. L. Tripathi, "Artificial Intelligence for Environmental Monitoring, Conservation and Waste Management," in *Waste Management and Utilization for a Sustainable Environment*, pp. 163–188, 2025.
4. I. K. Egbuna et al., "Advancing environmental sustainability through emerging AI-based monitoring and mitigation strategies for microplastic pollution in aquatic ecosystems," *Environmental Health*, vol. 2, p. 3, 2025.
5. A. K. Wani et al., "Environmental resilience through artificial intelligence: innovations in monitoring and management," *Environmental Science and Pollution Research*, vol. 31, no. 12, pp. 18379–18395, 2024.
6. D. D. Olodu, F. Inegbedion, and A. Erameh, "The Role of Artificial Intelligence and Machine Learning in Environmental Monitoring and Management," *International Journal of New Findings in Engineering, Science and Technology*, vol. 4, no. 1, pp. 15–26, 2026.
7. M. Basu and M. D. Nath, "A study of the role of artificial intelligence in monitoring environmental and health issues in the post-COVID-19 pandemic era for sustainable living," in *Artificial Intelligence for Multimedia Information Processing*, pp. 132–166, CRC Press, 2024.
8. G. Schouten, B. S. Michielsen, and B. Gravendeel, "Data-centric AI approach for automated wildflower monitoring," *PLOS ONE*, vol. 19, no. 9, p. e0302958, 2024.
9. J. Wu, J. Chen, and D. Huang, "Entropy-based active learning for object detection with progressive diversity constraint," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9387–9396, Jun. 2022.
10. M. Bartolo et al., "Litter detection from aerial imagery: a review of UAV-based approaches and deep learning techniques," *Multimedia Tools and Applications*, vol. 85, no. 3, p. 232, 2026.
11. B. M. Pati et al., "Leveraging UAV data and deep learning models for detecting waste in rivers," *IEEE Access*, vol. 13, pp. 99603–99627, 2025.
12. P. S. Bala, C. Kaviyazhiny, and S. Ajeeth, "Analysis of Emerging Deep Learning Techniques for Enhanced Real-Time Object Detection and Classification," in *Intelligent Data-Driven Systems with Innovations in Artificial Intelligence*, pp. 343–354, CRC Press, 2025.
13. O. Parami et al., "Systematic Review of Annotation Strategies for Overlapping Objects in AI Waste Management," in *2025 International Conference on Advances in Technology and Computing (ICATC)*, pp. 1–7, Dec. 2025.

14. N. Sun, K. Yang, D. Liu, and X. Guo, "Semi-supervised UAV-based river plastic detection with efficient teacher-student framework," in *2025 5th International Conference on Computer Science and Blockchain (CCSB)*, pp. 43–47, Aug. 2025.
15. D. Kumar et al., "Optimizing Litter Detection in Images: An Integration of Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs)," *IEEE Access*, vol. 14, pp. 3325–3344, 2025.