sustainability

MDPI

Article

Sustainable Waste Management Through Deep Learning: A Knowledge Distillation Framework for Real-Time Garbage Classification

Nawanol Theera-Ampornpunt, Panisa Treepong *, Panuwat Jannu and Apimet Sritongkul

College of Computing, Prince of Songkla University, Phuket 83120, Thailand
* Correspondence: panisa.t@phuket.psu.ac.th

Abstract

Effective waste sorting is central to circular economy goals and sustainable waste management: it maximizes recycling yields, diverts waste from landfills, and reduces the environmental burden of solid waste disposal. Accurate automated sorting using deep learning can achieve this at scale, yet high-performing classifiers are too computationally demanding for the low-cost embedded hardware used in sorting facilities. We propose the KD-Garbage Framework, which applies knowledge distillation to transfer predictive knowledge from a high-capacity teacher model to a lightweight student model, enabling deployment-ready classifiers that approach or exceed teacher-level accuracy without any added inference cost. We also introduce a 15,681-image garbage dataset organized into 13 classes defined by recycling and disposal pathway, assembled from 12 public sources and original photography, with all labels manually verified. Two teacher models were paired with 16 lightweight convolutional neural network (CNN) student architectures and benchmarked on a Raspberry Pi 5 at a minimum throughput of five frames per second. Knowledge distillation reduced misclassification rates by 10–25% across all student architectures. The best-performing student, RegNetY-1.6GF, achieved a balanced accuracy of 0.9129, surpassing both teacher models while sustaining real-time throughput on the target hardware, demonstrating a practical pathway toward scalable, AI-enabled sustainable waste management.

Keywords: sustainable waste management; circular economy; recycling; garbage classification; knowledge distillation; deep learning; convolutional neural network; edge deployment; machine vision

Academic Editor: Cristina Trois

Received: 18 May 2026
Revised: 12 June 2026
Accepted: 16 June 2026
Published: 23 June 2026

1. Introduction

Municipal solid waste (MSW) generation has emerged as one of the defining environmental challenges of the twenty-first century. The World Bank estimates that approximately 2.01 billion tonnes of solid waste are generated globally each year, a figure projected to reach 3.40 billion tonnes by 2050 as urbanization and economic growth accelerate [1]. At present, at least one-third of this waste is managed in ways that threaten public health and ecosystems [2,3]. Effective waste sorting is a critical bottleneck in the recycling supply chain: mixing recyclable and non-recyclable materials drives up processing costs, reduces material recovery rates, and ultimately increases the volume of waste directed to landfills [4]. Conventional sorting relies heavily on manual labor, which is slow, inconsistent, and difficult to scale. These limitations have motivated growing interest in automated

solutions, with computer vision and deep learning emerging as particularly promising tools for visual waste classification.

Convolutional neural networks (CNNs) have become the dominant paradigm for image classification, and their application to garbage classification has been extensively studied. Aral et al. [5] evaluated several established CNN architectures on the TrashNet dataset [6], demonstrating that deep networks substantially outperform traditional feature-engineering baselines. Ahmed et al. [7] and Gude et al. [8] applied standard CNN architectures to garbage classification and detection, and several studies have proposed CNN architectures specifically designed for waste sorting [9–12]. A consistent finding across these works is that classification accuracy scales with model capacity. More recently, Vision Transformers (ViTs) [13] have challenged CNN dominance by replacing convolutional operations with self-attention mechanisms that capture long-range dependencies across image patches. Hierarchical variants such as the Swin Transformer [14] address the quadratic complexity of global attention by computing self-attention within local shifted windows, yielding a more favorable accuracy-to-cost ratio.

High-performing models such as EfficientNetV2-S [15] and SwinV2-T [16], however, require tens of millions of parameters and substantial floating-point operations per forward pass. Garbage sorting hardware is frequently deployed in environments with strict computational constraints—collection-point kiosks, conveyor-belt inspection systems, and mobile units—where real-time throughput is essential and cloud offloading may not be feasible. Lightweight CNN architectures have been designed for exactly these scenarios; the MobileNet family [17,18], EfficientNet [19], RegNet [20], and ShuffleNet V2 [21] achieve competitive accuracy with only 1–11 million parameters by employing techniques such as depthwise separable convolutions, grouped convolutions, and channel shuffling. However, lightweight models trained independently typically fall short of their heavier counterparts in classification accuracy, and bridging this gap while preserving deployment efficiency remains an open challenge.

Knowledge distillation (KD), introduced by Hinton et al. [22], provides a principled mechanism for transferring the predictive knowledge of a large, high-capacity teacher model to a smaller, more efficient student model. Rather than training the student on hard one-hot labels alone, KD exposes it to the soft probability distributions produced by the teacher, which encode rich inter-class relationship information absent from hard labels. The KD literature has expanded considerably; Gou et al. [23] surveyed the field and organized it into response-based methods [22], feature-based methods [24], and relation-based methods [25]. KD has been successfully applied across a range of vision tasks: Chen et al. [26] showed that distilling from a high-capacity detector substantially improves lightweight detection models at no additional inference cost, and Cho and Hariharan [27] demonstrated that the capacity gap between teacher and student is a key determinant of distillation effectiveness.

Despite this progress, the existing literature on automated garbage classification reveals four important gaps. First, widely used datasets such as TrashNet [6] are limited in scale and class coverage, and their taxonomies are defined by material composition rather than by recycling or disposal pathway—a mismatch with the operational logic of sorting facilities, where the relevant question is not what the material is but where it should go [28–31]. Second, the application of KD to garbage and waste classification remains largely unexplored. Third, no prior study has systematically benchmarked a diverse range of lightweight student architectures under a unified KD framework on a large-scale, practically motivated garbage dataset. Fourth, there is no established guidance for practitioners on which lightweight architecture to select when deploying a garbage classifier under resource constraints.

This paper addresses all four gaps by making the following contributions:

- The KD-Garbage Framework, a KD pipeline that pairs two teacher models—SwinV2-T [16] and EfficientNetV2-S [15]—with 16 lightweight CNN student architectures spanning the MobileNet, EfficientNet, RegNet, and ShuffleNet V2 families.
- A new garbage classification dataset comprising 15,681 images drawn from 12 public sources, social media, and original photography, with all labels manually verified for correctness. The dataset adopts a 13-class taxonomy aligned with real-world recycling and disposal pathways, offering greater scale, visual diversity, and operational relevance than existing benchmarks.
- Empirical evidence that offline KD enables lightweight student models to match or exceed teacher-level garbage classification accuracy without adding any computational overhead at deployment.
- Practical model-selection recommendations identifying which student architectures offer the best accuracy-efficiency trade-off for resource-constrained garbage sorting applications.

The remainder of this paper is organized as follows. Section 2 presents the KD-Garbage Framework, the deep-learning models used, the proposed dataset, and the experimental setup. Section 3 reports the results and discusses the implications for system design. Section 4 concludes the paper with a summary of findings and directions for future work.

2. Materials and Methods

2.1. The KD-Garbage Framework

Figure 1 illustrates the components and steps of the KD-Garbage Framework. The framework follows the response-based KD paradigm introduced by Hinton et al. [22] and proceeds in two sequential training stages. In the first stage, the teacher model is trained from scratch on the garbage dataset using hard labels, in which the ground-truth probability for the correct class is 1 and all remaining classes are 0. In the second stage, the trained teacher generates soft labels for each training image according to the temperature-scaled softmax

$$\hat{y}_i(\mathbf{x}|T) = \frac{\exp\left(\frac{z_i(\mathbf{x})}{T}\right)}{\sum_j \exp\left(\frac{z_j(\mathbf{x})}{T}\right)}, \tag{1}$$

where $\hat{y}_i(\mathbf{x}|T)$ is the probability the teacher assigns to class i for input image $\mathbf{x}$, $z_i(\mathbf{x})$ is the corresponding teacher logit, and T is the temperature hyperparameter. The student model is then trained jointly on these soft labels and the original hard labels using the composite loss

$$\ell(\mathbf{x}|T) = -T^2 \sum_i \hat{y}_i(\mathbf{x}|T) \log y_i(\mathbf{x}|T) - \sum_i \bar{y}_i \log y_i(\mathbf{x}|1), \tag{2}$$

where $y_i(\mathbf{x}|T)$ is the student's temperature-scaled class probability, and $\bar{y}_i$ is the hard ground-truth label. The first term penalizes the divergence between the student's output distribution and the teacher's soft distribution, while the second term penalizes deviation from the ground-truth labels. The T^2 coefficient on the first term compensates for the reduction in gradient magnitude that temperature scaling introduces, ensuring that the two loss terms remain comparably weighted throughout training.

The temperature parameter T controls the sharpness of the teacher's output distribution. At $T = 1$, the soft labels coincide with the standard softmax probabilities. As T increases, the distribution becomes softer, assigning non-negligible probability mass to incorrect classes. These inter-class probability relationships encode knowledge that hard

labels cannot convey. For instance, a piece of foam packaging may visually resemble both plastic and foam items. A soft label of 60% foam and 40% plastic reflects this ambiguity more faithfully than a hard label of 100% foam. Training on such soft labels encourages the student to produce appropriately calibrated, rather than overconfident, predictions.

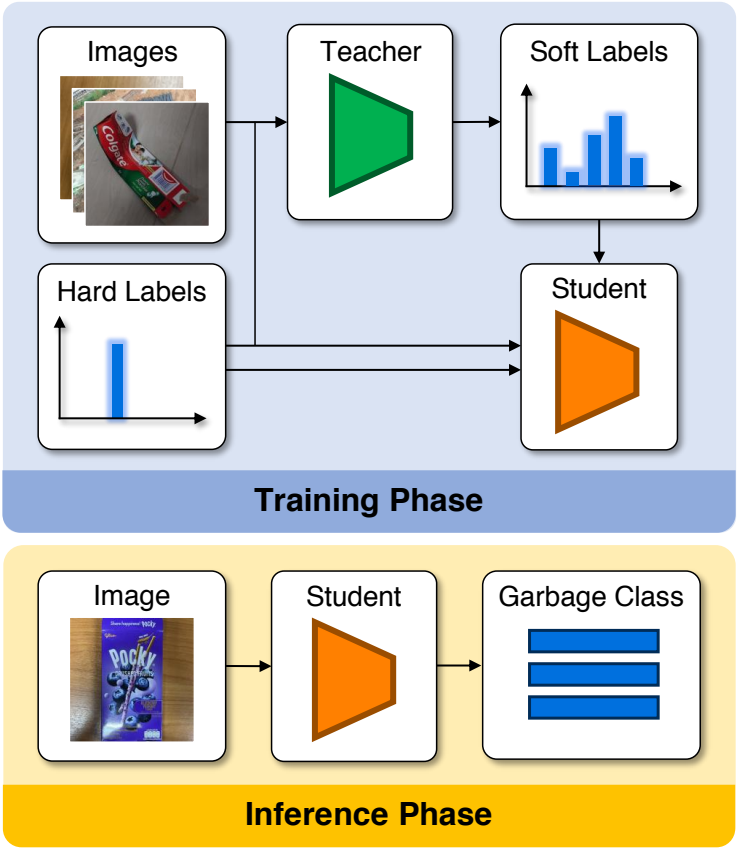


Figure 1. Overview diagram of the KD-Garbage Framework.

Once student training is complete, the teacher model and the training dataset are no longer required and can be discarded. The deployed student operates at its native inference speed, incurring none of the computational overhead associated with the teacher. All additional computation introduced by the framework—teacher training, soft label generation, and composite-loss student training—occurs entirely offline and can be performed on high-performance hardware or cloud infrastructure prior to deployment.

2.2. Models

The vision models used in this study are divided into two groups: teacher models and lightweight student models, as listed in Table 1. Lightweight models were selected on the basis of a minimum throughput requirement of five frames per second (FPS) on a budget Raspberry Pi 5 device, ensuring practical deployability in real-world sorting environments. This number is chosen to match the speed of conveyor sorter which we estimate to be five items per second. Teacher models were chosen to be accurate yet only moderately larger than the student models, since an excessively large capacity gap between teacher and student has been shown to impair knowledge transfer [27].

Table 1. Deep-learning architectures used in the study.

CNN/ViT Architecture	Parameters
Teacher Models:	
SwinV2-T [16]	28.4M
EfficientNetV2-S [15]	20.3M
Lightweight Models:	
EfficientNet-Lite0 [19]	4.7M
EfficientNet-Lite1 [19]	5.4M
EfficientNet-Lite2 [19]	6.1M
EfficientNet-B0 [19]	5.3M
EfficientNet-B1 [19]	7.8M
EfficientNet-B2 [19]	9.1M
MobileNetV2 [17]	3.5M
MobileNetV3-Small [18]	2.5M
MobileNetV3-Large [18]	5.5M
RegNetY-400MF [20]	4.3M
RegNetY-800MF [20]	6.4M
RegNetY-1.6GF [20]	11.2M
ShuffleNet V2 0.5× [21]	1.4M
ShuffleNet V2 1× [21]	2.3M
ShuffleNet V2 1.5× [21]	3.5M
ShuffleNet V2 2× [21]	7.4M

2.2.1. MobileNet V2

MobileNetV2 [17] is a lightweight CNN designed for efficient inference on mobile and edge devices. Its central contribution is the inverted residual block, which departs from the conventional residual design by placing the skip connection between narrow, low-dimensional bottleneck layers rather than wide ones. Within each block, a pointwise convolution first expands the channel dimension by a fixed factor, a depthwise convolution then captures spatial features at low computational cost, and a second pointwise convolution projects the representation back to the bottleneck width. A linear activation is applied after this final projection to prevent information loss in low-dimensional spaces, a design choice the authors term the linear bottleneck. These structural decisions yield a model that achieves strong image-classification accuracy while remaining suitable for deployment in resource-constrained environments.

2.2.2. MobileNet V3

MobileNetV3 [18] advances the MobileNet architecture family by combining NAS with a set of hand-crafted architectural refinements. The NAS procedure, based on platform-aware search, optimizes the layer configuration for latency on mobile hardware, while manual modifications address components that automated search handles poorly. Key innovations include the integration of the hard-swish activation function, which approximates the computationally expensive swish non-linearity using only integer arithmetic, and the incorporation of Squeeze-and-Excitation (SE) modules [32] into selected inverted residual blocks to enable channel-wise feature recalibration. The redesigned classification head, which replaces the conventional average-pooling and fully connected layers with a more efficient structure, further reduces latency without sacrificing accuracy. Two variants are released—MobileNetV3-Large and MobileNetV3-Small—targeting different computational budgets. The present study employs both architectures.

2.2.3. EfficientNet V1

EfficientNet [19] introduces a principled method for scaling CNNs along three orthogonal dimensions—depth, width, and input resolution—simultaneously and in a balanced manner. Prior work typically scaled only one of these dimensions in isolation; the authors demonstrate that co-scaling all three according to a fixed set of compound coefficients consistently improves accuracy and efficiency. The base architecture, EfficientNet-B0, is obtained via NAS and uses Mobile Inverted Bottleneck Convolution (MBConv) blocks as its primary building unit, augmenting standard inverted residuals with SE modules. Scaling B0 with increasing compound coefficients yields the B1–B7 architectures. This study additionally employs EfficientNet-Lite, a derivative variant in which SE blocks are removed and the swish activation is replaced with ReLU6 to improve compatibility with mobile accelerators. We employ three smallest architectures within each variant as lightweight student models in the proposed KD-Garbage Framework.

2.2.4. EfficientNet V2

EfficientNetV2 [15] revisits the EfficientNet scaling strategy with the explicit goals of accelerating training and improving parameter efficiency. Through a combination of NAS and careful analysis of training-speed bottlenecks, the authors identify that the large depthwise convolution kernels in earlier EfficientNet stages contribute disproportionately to training time. Accordingly, EfficientNetV2 replaces these layers with Fused-MBConv blocks—which merge the depthwise and expansion convolutions into a single standard convolution—in the early network stages, reserving the original MBConv design for deeper layers where it remains efficient. The model family is further trained with progressive learning, a curriculum that gradually increases image resolution and regularisation strength during training. EfficientNetV2-S, the smallest architecture of the family, is employed as a teacher model in this study due to its high classification performance and strong feature representations.

2.2.5. RegNet

RegNet [20] is a family of networks derived from a systematic design-space exploration aimed at identifying simple, general principles that govern efficient network architectures. Rather than searching for a single optimal architecture, the authors parameterize a large space of residual networks and iteratively refine it by fitting low-dimensional models to the empirical accuracy–complexity frontier. This process yields a regular network design governed by a small set of linear constraints on width and depth, giving rise to the name RegNet. All models in the family share a consistent structure of four stages, each composed of groups of X-blocks (standard residual bottleneck blocks with grouped convolutions). Two sub-families are considered: RegNetX, which uses plain grouped convolutions, and RegNetY, which augments each block with an SE module. The present study employs three RegNetY variants—RegNetY-400MF, RegNetY-800MF, and RegNetY-1.6GF, where the suffix denotes the approximate floating-point operations at inference.

2.2.6. ShuffleNet V2

ShuffleNet V2 [21] is a highly efficient CNN architecture designed by analyzing practical runtime cost on hardware rather than relying solely on theoretical metrics such as Floating Point Operations Per Second (FLOPS). The authors identified four practical guidelines for efficient network design—equal channel-width, balanced group convolution, reduced network fragmentation, and minimized element-wise operations—and showed that earlier efficient architectures violate several of these principles. ShuffleNet V2 addresses these shortcomings through two key operations: channel split and channel shuffle. Channel

split partitions the input feature map into two branches at the start of each block. One passes through as an identity shortcut and the other is processed by three convolutions. Channel shuffle enables cross-channel information flow between branches after concatenation. This design achieves a strong accuracy-efficiency trade-off and scales gracefully through width multipliers. All four variants are evaluated in this study—ShuffleNet V2 0.5×, 1×, 1.5×, and 2×.

2.2.7. Swin Transformer V2

Swin Transformer V2 [16] is a hierarchical Vision Transformer (ViT) that extends the original Swin Transformer [14] to improve training stability at scale and to close the performance gap when transferring from small-resolution pre-training to high-resolution downstream tasks. The architecture retains the core mechanism of shifted window-based multi-head self-attention (SW-MSA), which computes attention within local non-overlapping windows and alternates window partitions across layers to allow cross-window communication, yielding linear computational complexity with respect to image size. SwinV2 introduces three principal modifications: a residual post-norm configuration that stabilizes gradient flow in large models, a scaled cosine attention function that decouples attention weights from feature magnitude, and a log-spaced continuous positional bias (Log-CPB) that enables flexible transfer to higher resolutions without performance degradation. SwinV2-T serves as one of the two teacher models in the KD-Garbage Framework, providing rich hierarchical feature representations to guide the training of the lightweight student models.

2.3. Dataset

For this work, we constructed a new dataset by first consolidating images from 12 publicly available sources and then supplementing them with images collected from social media platforms and photographs taken directly by the authors. The full list of sources is provided in Table 2.

Table 2. Dataset sources.

Number	Source
1	TrashNet [6]
2	cd_ver3 Computer Vision Dataset [33]
3	Garbage Classification [34]
4	Drinking Waste Classification [35]
5	MJU-Waste [36]
6	paper Computer Vision Dataset [37]
7	TACO [38]
8	Waste Images from Sushi Restaurant [39]
9	styrofoam Computer Vision Dataset [40]
10	Food Waste Detection Computer Vision Dataset [41]
11	dish-segmentation Computer Vision Dataset [42]
12	Garbage Dataset [43]
13	Social media platforms
14	Photographs taken by the authors

The consolidated images were classified into 13 categories, listed in Table 3. The taxonomy uses the MSW classification of the World Bank and the United States Environmental Protection Agency (EPA) [44,45] as its basis, with certain categories subdivided to better reflect distinct recycling and disposal pathways. The criteria and rationale for each category are as follows.

Table 3. Garbage categories and number of images.

Category	Number of Images
Appliance	875
Food Scrap	879
Foam	1283
Glass	926
Cardboard	822
Milk Carton	243
Paper-White	607
Paper-Other	788
Plastic	7142
Construction-Metal	257
Construction-Nonmetal	208
Metal	1093
Mixed/Other	558
Total	15,681

- Appliance encompasses electronic and electrical equipment, including components such as circuit boards and batteries, which require dedicated collection streams due to their hazardous constituents.
- Food Scrap covers directly visible leftover food waste in any container or packaging. Because food container always belongs to another class, we consider the container part of the background.
- Foam, although chemically a type of plastic, is designated as a separate class because its recycling requires specialized machinery distinct from that used for conventional plastics.
- Glass includes all food and beverage container, product packaging, and other items made primarily of glass. As glass bottles and containers often contains metal lids or caps, we do not take into consideration the material of the lids and caps.
- Cardboard also includes unbleached brown kraft paper, which shares the same processing pathway.
- Milk Carton covers paperboard cartons coated with polyethylene, and, for shelf-stable variants, an additional layer of aluminum foil. This composite construction means that milk cartons can only be recycled at specialized facilities, warranting their own class.
- Paper-White is restricted to non-glossy white paper printed exclusively in black ink, as this grade can be recycled to a higher standard than mixed paper.
- Paper-Other groups together all other paper types, namely glossy paper and paper printed with color ink.
- Plastic recycling or disposal process is determined by its polymer type, which cannot be reliably inferred from visual appearance alone. Although Resin Identification Codes (RICs) are molded into many plastic products to facilitate sorting, these symbols are typically too small to identify from the distances at which sorting systems operate. Accordingly, all plastic waste except foam is consolidated into a single class.
- Construction-Metal covers used or leftover construction materials made of metal.
- Construction-Nonmetal covers demolition and building waste such as concrete, bricks, and tiles. Construction waste is split along material lines because these two categories are processed differently.
- Metal covers all other metal items. Finer alloy classification is not possible from appearance alone, so all types are grouped together.
- Mixed/Other encompasses items composed of two or more materials from different classes, as well as materials that do not fit any other category, such as rubber, textiles, and ceramics.

Regardless of source, all images were subjected to the following standardized screening and preprocessing pipeline:

1. Filtering: Each image was manually inspected. Images that were low-quality, blurred, or unrelated to the target categories were removed. Images containing multiple items belonging to different categories were either cropped to isolate a single item or excluded entirely, to prevent ambiguous training signals.
2. Cropping: Each image was manually cropped to a square aspect ratio, centering on the garbage item as closely as possible. This step produces spatially consistent inputs and removes extraneous background content that could introduce confounding visual features.
3. Labeling: Each image was assigned a category label according to the 13-class taxonomy. Images that could not be confidently assigned to a single class were excluded from the dataset.
4. Resizing: All images were resized to 224×224 pixels to match the input resolution expected by the model architectures used in this study.

The dataset was partitioned into training, validation, and test subsets comprising 65%, 15%, and 20% of all images, respectively. During hyperparameter optimization, models were trained on the training subset and evaluated on the validation subset. For the final performance evaluation, models were retrained on the combined training and validation subsets and evaluated on the held-out test subset. This two-phase protocol ensures that hyperparameters are selected without any exposure to the test data, so that the reported metrics provide an unbiased estimate of generalization performance.

Representative examples of images from each category are shown in Figure 2.

2.4. Experimental Setup

All models were initialized with weights pre-trained on ImageNet-1K [46] before being fine-tuned for garbage classification. The final classification layer of each model was replaced with a new linear layer with 13 outputs corresponding to the target classes. A global average pooling layer and a dropout layer were appended where not already present. Each model was then trained in two successive phases. In the transfer learning phase, only the new output layer was trained while all pre-trained weights were kept frozen, allowing the model to adapt its output space to the new class structure without disrupting the learned feature representations. In the subsequent fine-tuning phase, the output layer and the latter three-quarters of the backbone's main stages were jointly optimized with a lower learning rate, enabling task-specific refinement of higher-level features while preserving the low-level representations learned from ImageNet. The full set of training hyperparameters is reported in Table 4. Learning rates were optimized separately for each run.

Table 4. Transfer learning and finetuning hyperparameters for all models.

Hyperparameter	Value
Image Resolution	224×224
Batch Size	128
Optimizer	AdamW
Momentum	None
Weight Decay	$\{1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}, 1 \times 10^{-6}\}$
LR Schedule	CLR triangular policy [47] with initial LR of 0.05 of maximum LR
Training Length	20 epochs
Data Augmentation	Random horizontal flip
Dropout Rate	0.2
Label Smoothing	None

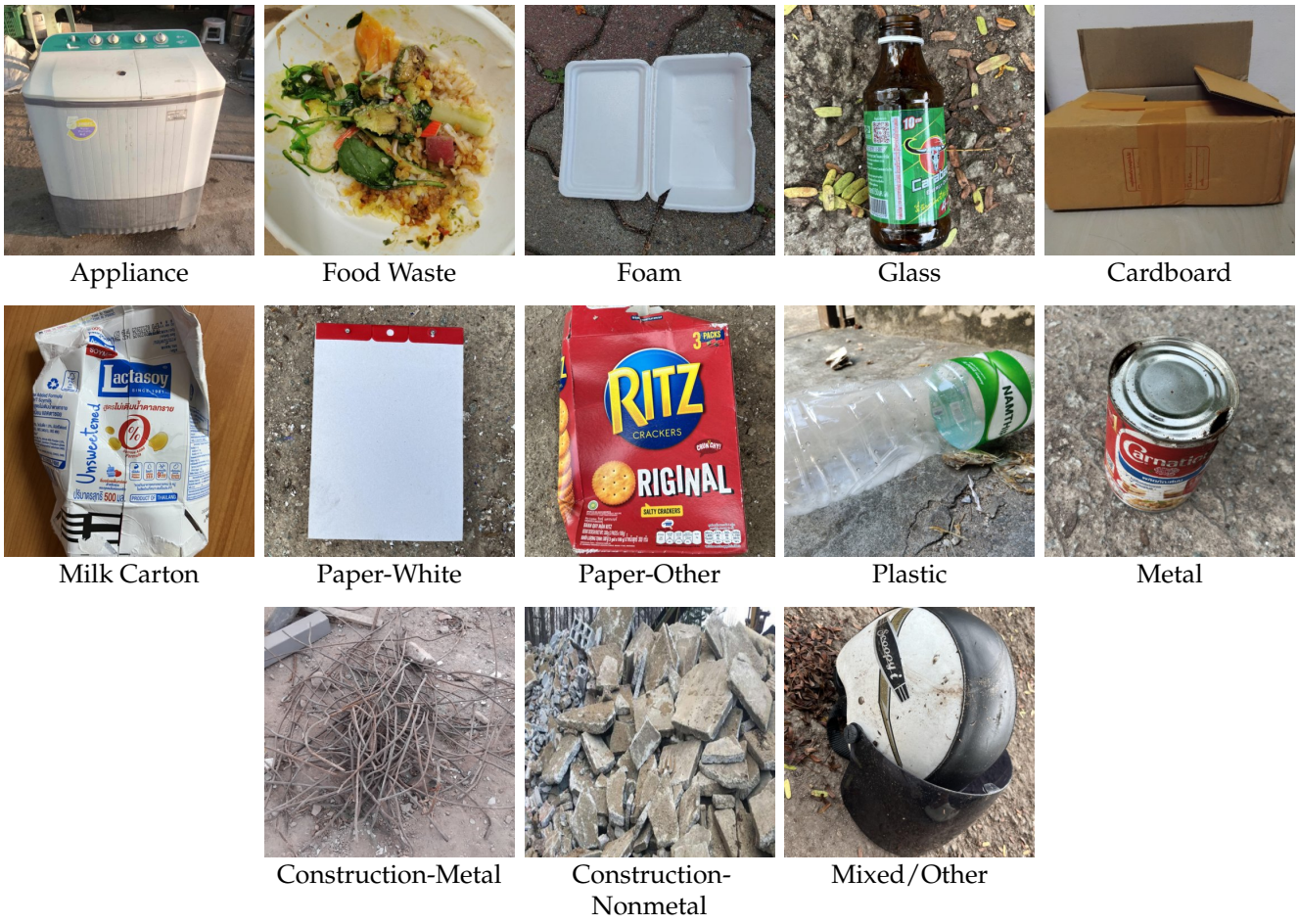


Figure 2. Representative examples of images in the garbage dataset.

All preprocessing, training, and evaluation procedures were implemented in PyTorch 2.9.0 [48]. Experiments were conducted on a workstation equipped with an Intel Core i5-12400F (Intel Corporation, Santa Clara, CA, USA) processor, 64 GB of system memory, and an NVIDIA GeForce RTX 5090 GPU (NVIDIA Corporation, Santa Clara, CA, USA). To account for variance in the training process arising from random weight initialization and data shuffling, each training configuration was repeated three times, and all reported metrics are averages over these three runs. Statistical significance test was performed using one-tailed Welch's *t*-test.

Because the dataset is class-imbalanced (see Table 3), standard accuracy—which weights each sample equally regardless of class—would favor models that perform well on the dominant Plastic class at the expense of minority classes. We therefore adopted balanced accuracy (BA) as the primary evaluation metric. BA is defined as

$$\mathrm{BA} = \frac{1}{m}\sum_{i=1}^{m}\frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}, \tag{3}$$

where m is the number of classes and TP_i and FN_i are the numbers of true positives and false negatives for class i, respectively. BA is equivalent to the macro-averaged recall, assigning equal weight to each class irrespective of its size. It ranges from 0 to 1, with higher values indicating fewer misclassifications. To align the training objective with this metric, the cross-entropy loss was weighted inversely proportional to class size, penalizing errors on underrepresented classes more heavily during optimization.

3. Results and Discussions

3.1. Model Efficiency

For an AI-based garbage sorting system to be economically viable, the classifier must run in real time on commodity embedded hardware. In this experiment, we benchmarked all candidate architectures on a Raspberry Pi 5 single-board computer—a platform priced from 45 USD that offers a favorable balance of cost, computing power, and connectivity for edge deployment. Although the Raspberry Pi 5 includes a GPU, the absence of compatible machine learning inference libraries at the time of writing necessitated CPU-only evaluation. Thermal throttling was prevented by installing the manufacturer's active cooler.

Inference latency was measured at a fixed input resolution of 224 × 224 pixels and a batch size of one, reflecting single-item sorting at deployment. Each model was run on 100 images. The maximum latency observed over the latter 50 images is reported to characterize sustained throughput while excluding the initial warm-up period. Memory consumption was also concurrently recorded.

The results are presented in Table 5. Architectures not specifically designed for mobile deployment—SwinV2-T, MaxViT-T, and DenseNet-121—all fall well below the 5 FPS threshold, confirming that generalist architectures cannot be directly deployed in this setting without further optimization. Within the EfficientNet and EfficientNet-Lite families, only the three smallest variants, B0–B2 and Lite0–Lite2 respectively, meet the throughput requirement. Notably, each EfficientNet-Lite variant achieved only marginally higher throughput than its standard EfficientNet counterpart at the same scale, suggesting that the removal of SE modules and the substitution of ReLU6 for Swish in the Lite variants offer limited practical latency benefits on CPU-based ARM hardware. Among the RegNet family, models up to and including RegNetY-1.6GF satisfy the requirement, while RegNetY-3.2GF does not. MobileNetV2, both MobileNetV3 variants, and all four ShuffleNet V2 variants comfortably exceed the 5 FPS threshold, a direct reflection of their optimization for mobile CPU inference.

Table 5. Model inference latency for a single image on a Raspberry Pi 5 single-board computer. FPS denotes frames per second.

Model	Latency (ms)	FPS	Memory (MB)
DenseNet-121	251	3.99	660
EfficientNet-Lite0	121	8.29	585
EfficientNet-Lite1	140	7.13	639
EfficientNet-Lite2	147	6.81	645
EfficientNet-Lite3	202	4.96	736
EfficientNet-B0	126	7.92	562
EfficientNet-B1	174	5.73	650
EfficientNet-B2	188	5.32	707
EfficientNet-B3	229	4.37	763
EfficientNetV2-S	266	3.75	836
MaxViT-T	603	1.66	1148
MobileNetV2	112	8.90	520
MobileNetV3-Large	100	9.96	467
MobileNetV3-Small	65	15.36	378
RegNetY-400MF	112	8.96	431
RegNetY-800MF	117	8.58	458
RegNetY-1.6GF	165	6.04	546
RegNetY-3.2GF	238	4.21	629
ShuffleNet V2 0.5×	51	19.62	368
ShuffleNet V2 1×	78	12.82	393
ShuffleNet V2 1.5×	92	10.84	425
ShuffleNet V2 2×	104	9.60	447
SwinV2-T	314	3.19	772

Regarding memory consumption, all models operated comfortably within the memory envelope of even the lowest-tier Raspberry Pi 5 configuration (1 GB), with the sole exception of MaxViT-T at 1148 MB. This result indicates that RAM capacity is unlikely to be a binding constraint when selecting models for this application on comparable embedded platforms. Based on these results, the 16 architectures that satisfy the 5 FPS requirement were carried forward for classification evaluation: EfficientNet-Lite0 to Lite2, EfficientNet-B0 to B2, MobileNetV2, MobileNetV3-Small, MobileNetV3-Large, RegNetY-400MF, RegNetY-800MF, RegNetY-1.6GF, and all four ShuffleNet V2 variants.

3.2. Architecture Comparison

In this experiment, we compare the classification performance of all candidate architectures on the proposed garbage dataset, both with and without the KD-Garbage Framework. The BA results are reported in Table 6.

Table 6. Classification performance of each architecture with and without the KD-Garbage Framework, measured by BA. The best result in each column among the lightweight models is highlighted in bold.

Architecture	Without KD-Garbage	With KD-Garbage
Teacher Models:		
SwinV2-T	0.9037	–
EfficientNetV2-S	0.9053	–
Lightweight Models:		
EfficientNet-Lite0	0.8559	0.8921
EfficientNet-Lite1	0.8691	0.8962
EfficientNet-Lite2	0.8612	0.8954
EfficientNet-B0	0.8896	0.9019
EfficientNet-B1	**0.8911**	0.9109
EfficientNet-B2	0.8877	0.9070
MobileNetV2	0.8616	0.8897
MobileNetV3-Small	0.8343	0.8697
MobileNetV3-Large	0.8719	0.8992
RegNetY-400MF	0.8705	0.8976
RegNetY-800MF	0.8790	0.9053
RegNetY-1.6GF	0.8888	**0.9129**
ShuffleNet V2 0.5×	0.8173	0.8470
ShuffleNet V2 1×	0.8497	0.8782
ShuffleNet V2 1.5×	0.8646	0.8863
ShuffleNet V2 2×	0.8775	0.8995

Without the KD-Garbage Framework, EfficientNet-B1 achieved the highest BA among the lightweight models at 0.8911, closely followed by EfficientNet-B0 and RegNetY-1.6GF at 0.8896 and 0.8888, respectively. EfficientNet-B1's higher accuracy is not statistically significant when compared to EfficientNet-B2 (p-value $= 0.25$) but significant compared to all other models (p-value ≤ 0.013). The two teacher models, SwinV2-T and EfficientNetV2-S, attained 0.9037 and 0.9053, respectively, exceeding all lightweight models as expected given their higher parameter counts. Across all lightweight architectures, the baseline BAs ranged from 0.8173 (ShuffleNet V2 0.5×) to 0.8911 (EfficientNet-B1), a spread of approximately seven percentage points, indicating that architecture choice has a meaningful impact on classification performance in the absence of KD.

With the KD-Garbage Framework applied, all lightweight models improved with statistical significance (p-value ≤ 0.015). RegNetY-1.6GF emerged as the top performer at 0.9129, followed by EfficientNet-B1 and EfficientNet-B2 at 0.9109 and 0.9070, respectively. These three models surpassed both teacher models, while RegNetY-800MF matched EfficientNetV2-S exactly at 0.9053. These results are noteworthy: a student model trained

with KD can exceed the classification accuracy of its teacher because the benefits of distillation (richer training signal from soft labels) can outweigh the structural capacity advantage of the teacher, provided the capacity gap between the two is not too large [27]. Put differently, distillation and model size are two distinct and partially independent levers for improving accuracy. A student that gains more from the former than it loses from the latter will outperform its teacher. The ranking of architectures was largely preserved after distillation, with a notable exception: RegNetY-1.6GF rose to the top from the third place. This suggests that RegNetY-1.6GF is well compatible with KD, achieving significant accuracy gain despite high baseline performance.

Figure 3 places these results in the context of computational efficiency, plotting BA against inference latency for all lightweight models with the KD-Garbage Framework applied.

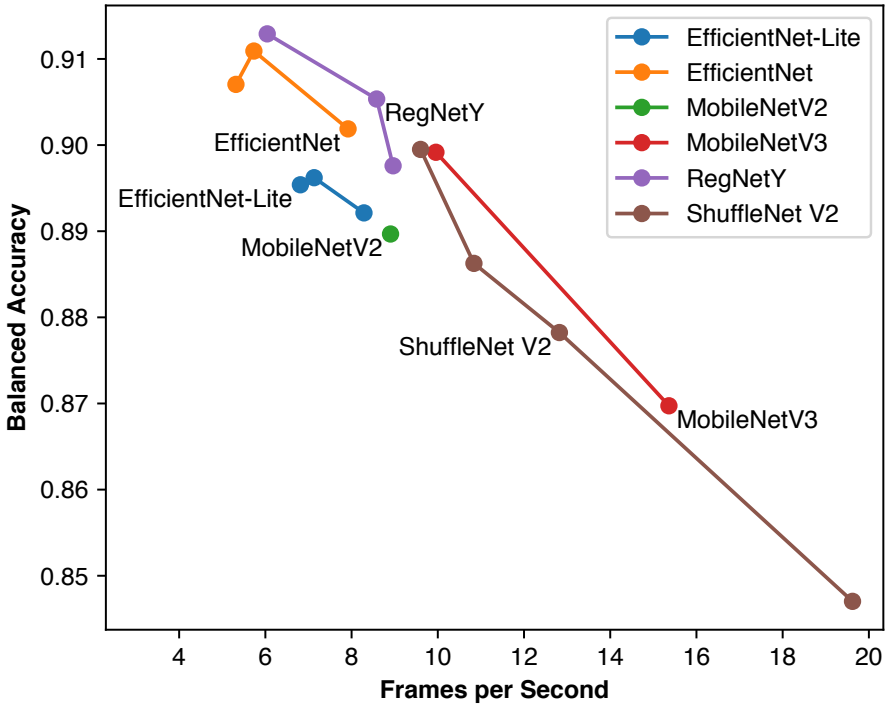


Figure 3. Classification accuracy vs. computational efficiency for all lightweight architectures trained with the KD-Garbage Framework. Points closer to the top-right corner represent better accuracy–efficiency trade-offs.

The Pareto-optimal architectures—those for which no other model achieves both higher accuracy and higher throughput—are RegNetY-1.6GF, RegNetY-800MF, both MobileNetV3 variants, and all ShuffleNet V2 variants. RegNetY-1.6GF offers the highest BA overall at 0.9129, making it the recommended choice when 5 FPS is sufficient. For applications requiring approximately 10 FPS, MobileNetV3-Large provides the best accuracy at that operating point with a modest drop relative to RegNetY-1.6GF. Pushing throughput further to 11–20 FPS with the ShuffleNet V2 family and MobileNetV3-Small incurs a steeper accuracy cost. In cases where the available throughput exceeds the minimum requirement, the surplus computational budget can alternatively be invested in increasing the input image resolution. Although our 224×224 baseline already achieves competitive accuracy, higher-resolution inputs would expose finer surface detail—such as resin identification codes on plastics or ink color on paper—that may benefit challenging classes. However, because inference latency scales linearly with the number of pixels, doubling both spatial dimensions would quadruple the latency, quickly exhausting the available budget.

3.3. Effectiveness of the KD-Garbage Framework

Having established the absolute performance of each architecture, we now investigate the conditions under which the KD-Garbage Framework is most effective. Specifically, we examine the choice of teacher model, the sensitivity of distillation performance to the temperature hyperparameter, and the effect of KD on training dynamics.

3.3.1. Choice of Teacher Model

Table 7 reports the improvement in BA attributable to KD for each student–teacher pair.

Table 7. Improvement in BA from the KD-Garbage Framework for each student–teacher pair. The better result for each student is highlighted in bold.

Model	Teacher	
	SwinV2-T	**EfficientNetV2-S**
EfficientNet-Lite0	+**0.0363**	+0.0298
EfficientNet-Lite1	+**0.0271**	+0.0237
EfficientNet-Lite2	+0.0228	+**0.0342**
EfficientNet-B0	+**0.0123**	+0.0113
EfficientNet-B1	+**0.0198**	+0.0182
EfficientNet-B2	+0.0188	+**0.0193**
MobileNetV2	+**0.0281**	+0.0259
MobileNetV3-Small	+0.0348	+**0.0354**
MobileNetV3-Large	+**0.0273**	+0.0257
RegNetY-400MF	+**0.0271**	+0.0226
RegNetY-800MF	+0.0243	+**0.0263**
RegNetY-1.6GF	+**0.0241**	+0.0211
ShuffleNet V2 0.5×	+**0.0297**	+0.0200
ShuffleNet V2 1×	+0.0280	+**0.0286**
ShuffleNet V2 1.5×	+0.0197	+**0.0217**
ShuffleNet V2 2×	+0.0190	+**0.0220**
Average	+**0.0249**	+0.0241

The KD-Garbage Framework improved the BA of every student architecture under both teacher configurations (p-value $\leq$ 0.015), confirming that KD is consistently beneficial across a diverse range of student and teacher designs. SwinV2-T was the more effective teacher in 9 out of 16 cases and produced a higher average improvement (+0.0249 vs. +0.0241) despite marginally lower classification accuracy than EfficientNetV2-S. This finding is consistent with the observation that a higher-accuracy teacher does not automatically translate to better distillation outcomes [27]. The structural diversity between teacher and student, as well as the nature of the inter-class relationships encoded in the teacher's soft outputs, also play important roles. SwinV2-T, as a hierarchical Vision Transformer, produces soft label distributions that reflect a qualitatively different inductive bias from the CNN-based students, which may provide complementary information not already captured by the student's own feature extractor.

A clear pattern emerges with respect to student capacity: smaller and less accurate models tended to benefit more from distillation. The most pronounced example is EfficientNet-Lite0, whose improvement of +0.0363 under SwinV2-T was nearly three times that of the architecturally similar but more capable EfficientNet-B0 (+0.0123). Expressed in terms of error reduction, the KD-Garbage Framework reduced the misclassification rate of EfficientNet-Lite0 by 25%—the largest single improvement observed—while the average reduction across all students was 19% and 18% for SwinV2-T and EfficientNetV2-S, respectively. Even RegNetY-1.6GF, already the strongest baseline student, achieved a strong 22% error reduction under SwinV2-T, demonstrating that KD adds value even when the student is close to the teacher in absolute accuracy. The difference in improvement between the two teachers was relatively small and was only statistically significant for EfficientNet-

Lite0, EfficientNet-Lite2, RegNetY-400MF, and ShuffleNet V2 0.5× (p-value $= 6 \times 10^{-5}$, 0.0018, 0.045, and 0.045, respectively). This suggests that either teacher is a viable choice in practice, though SwinV2-T is recommended as a default given higher overall improvement.

3.3.2. Effect of Temperature

KD introduces the temperature hyperparameter T, which controls the sharpness of the teacher's soft label distribution. Figure 4 shows the effect of temperature on student BA for both teacher configurations.

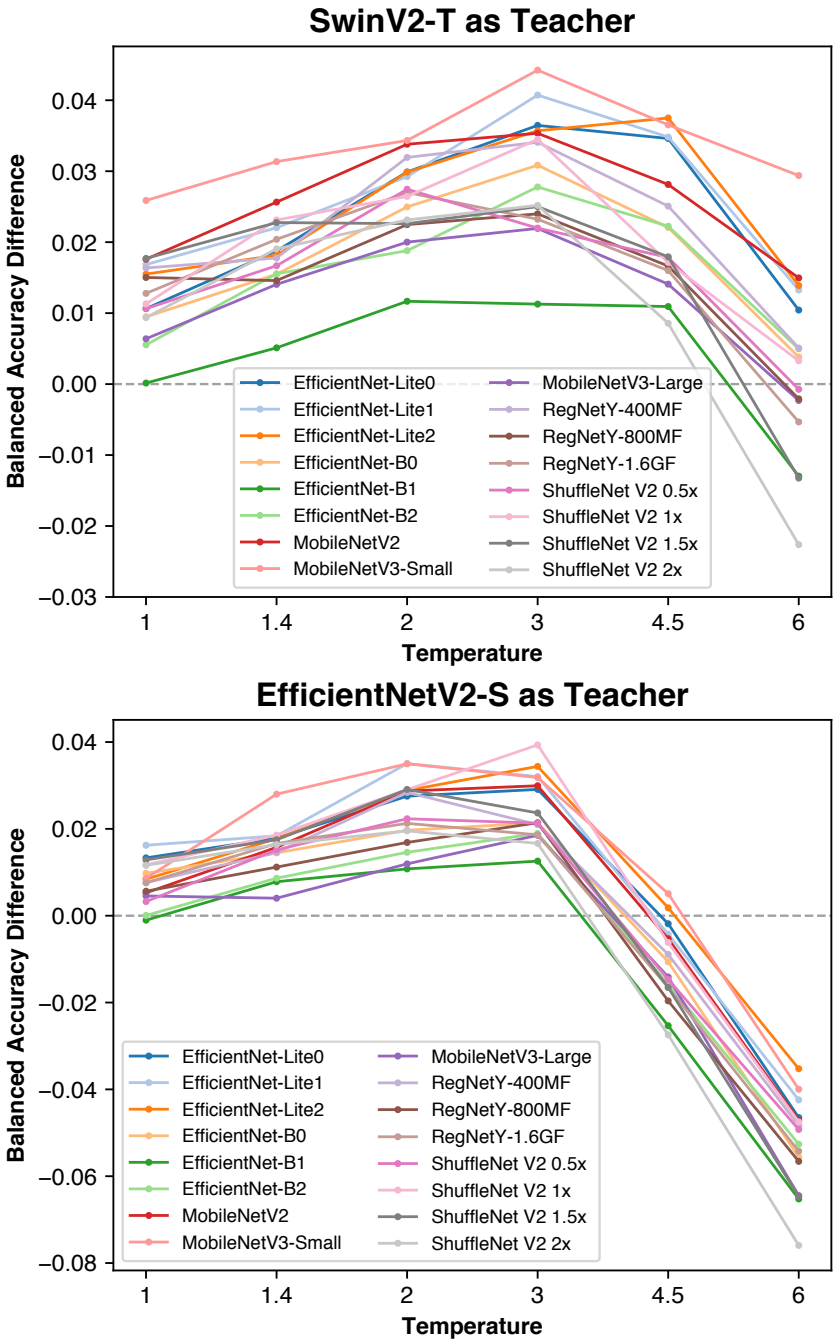


Figure 4. Effect of the temperature hyperparameter on student BA for each teacher model. Performance is shown relative to the same student trained without distillation (dashed baseline at 0).

All models exhibited a consistent unimodal response to temperature. The optimal value was $T = 3$ in 12 out of 16 cases with SwinV2-T as the teacher, and in 9 out of 16 cases with EfficientNetV2-S. $T = 4.5$ was optimal in one case, while the remaining

cases favored $T = 2$. Distillation remained beneficial even at suboptimal temperatures: low values ($T < 3$) still yielded improvements over the no-distillation baseline, whereas high values ($T \gg 3$) caused appreciable accuracy degradation. The deterioration at high temperatures is consistent with the theoretical expectation: as T increases, the soft labels become increasingly uniform, reducing the discriminative information they carry. The unimodal, convex shape of the temperature–accuracy curve has a useful practical implication: gradient-free optimization strategies such as ternary search or simple line search are guaranteed to locate the global optimum efficiently. Based on these results, $T = 3$ is a reliable default starting point for the garbage classification task, with little risk of substantially suboptimal performance at adjacent values.

3.3.3. Training Dynamics

To understand how KD influences the optimization process beyond its effect on final accuracy, we inspect the training curves of RegNetY-1.6GF across both training phases, with and without the KD-Garbage Framework. Figure 5 presents the BA and loss on the training and validation sets at each epoch.

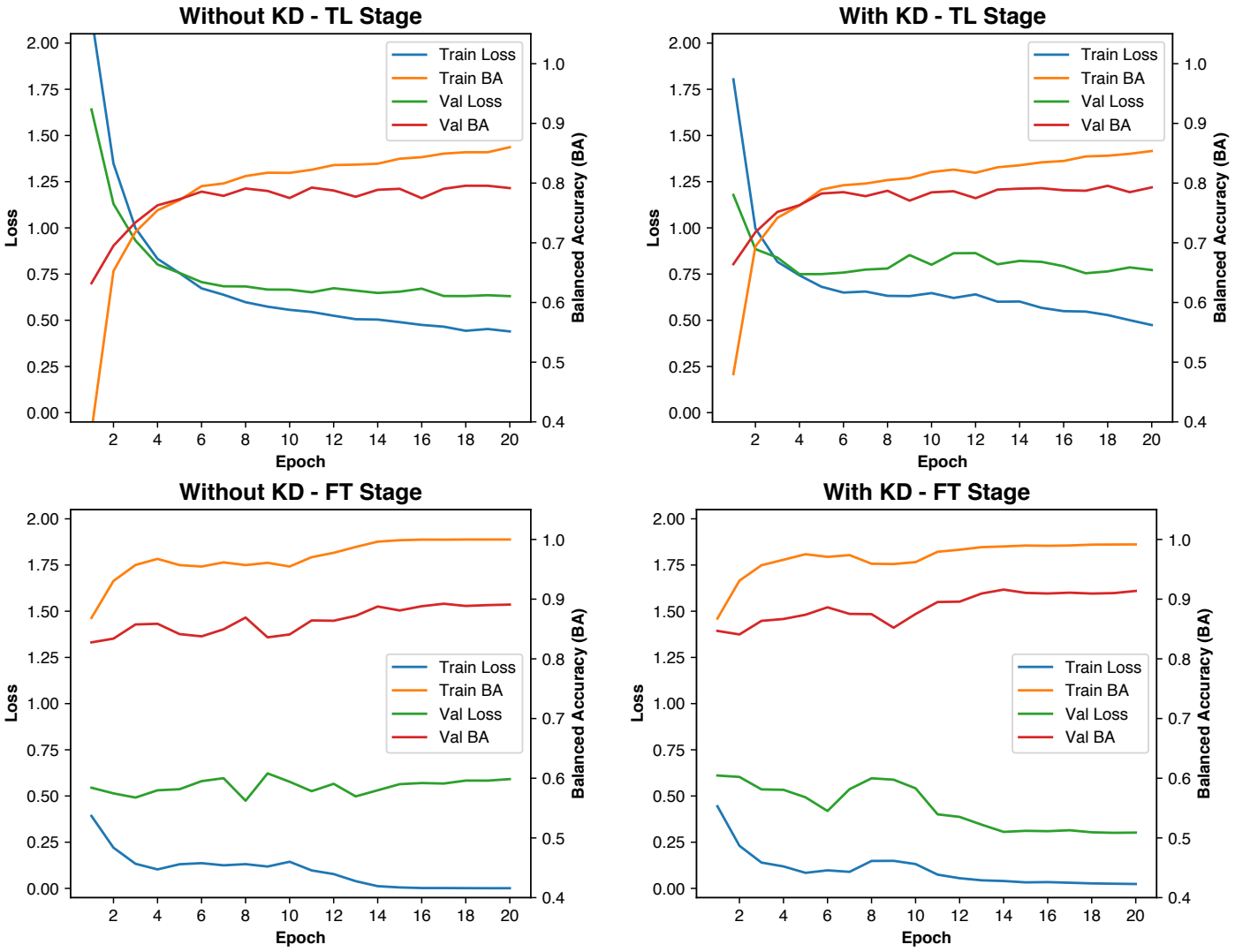


Figure 5. Training curves of RegNetY-1.6GF during the transfer learning (TL) and fine-tuning (FT) phases, with and without the KD-Garbage Framework. All loss values are computed using the standard cross-entropy loss on hard labels to allow direct comparison between the two training configurations.

During the transfer learning phase, validation BA was comparable between the two configurations, indicating that both models learned to map the pre-trained feature representations to the 13-class output space at a similar rate. The model trained with KD, however, exhibited noticeably higher validation loss during this phase. This is expected: the KD loss encourages the student to match the teacher's soft, distributed output rather than committing sharply to a single class. Consequently, the student's output probabilities are less peaked, resulting in higher cross-entropy loss against the hard ground-truth labels even when the predicted class is correct. This behavior reflects a form of implicit output-space regularization rather than a degradation in classification quality.

The distinction between the two configurations became more pronounced during the fine-tuning phase. The model trained with KD achieved higher validation BA at every epoch, while the baseline model exhibited a progressive divergence between training and validation loss in the latter half of the phase—a characteristic signature of overfitting. By contrast, the model trained with KD maintained a closer alignment between training and validation loss throughout, suggesting that the soft labels act as an effective regularizer that smooths the loss landscape and prevents the model from memorizing idiosyncrasies of the training set. Taken together, these results indicate that the accuracy gains from KD are a result of improved generalization through reduced overfitting, attributable to the richer label information.

3.4. Detailed Analysis

To gain deeper insight into the strengths and limitations of the proposed framework, we perform a fine-grained analysis of the classification behavior of the best-performing model, RegNetY-1.6GF trained with the KD-Garbage Framework using SwinV2-T as the teacher. We examine the normalized confusion matrix, investigate the most common misclassification patterns, and discuss their practical implications for system design.

3.4.1. Per-Class Performance

Figure 6 presents the normalized confusion matrix. Normalization applies row-wise, so that each cell reports the percentage of ground-truth instances of the row class predicted as the column class, regardless of class size. The diagonal entries therefore correspond directly to per-class recall.

The distribution of the model's predictions among the 13 classes is nearly uniform, indicating that the weighted loss function successfully equalized emphasis across the classes. 10 of the 13 classes achieved recall of 90% or above, indicating that the model generalizes reliably across the majority of the category space. The three classes with lower recall were Plastic (85%), Paper-Other (78%), and Mixed/Other (78%). The relatively lower performance on these three classes is attributable to distinct but identifiable factors, discussed in the next section.

When the confusion matrices of both the student model and the teacher model are compared, it is evident that both models have some overlap in misclassification patterns. Nevertheless, the two models exhibit complementary strengths. The teacher, with its larger parameter count, achieves higher recall on Plastic, a class characterized by exceptional visual heterogeneity in shape, size, surface texture, and color, where greater representational capacity translates directly into more reliable discrimination. The student model, in comparison, achieves better recall for Milk Carton, which has limited training data and its members shares shape and texture characteristics with some other categories. This advantage may be attributed to the regularizing effect of knowledge distillation. Taken together, these results show that model capacity and knowledge distillation address different parts of a classification problem and that they are most effective when used together.

ID	True Class	Predicted Class 1	2	3	4	5	6	7	8	9	10	11	12	13	Recall
1	Appliance	94	0	0.6	0	0	0.6	0.6	0.6	1.1	0	0.6	0.6	1.1	94.29
2	Food Scrap	0	99	0	0	0	0	0	0	0	0	0.6	0	0	99.43
3	Foam	0	0	97	0	0	0.4	0.4	0	0	0	1.6	0	0.4	97.28
4	Glass	1.1	0.5	0	91	0	0.5	0.5	0	1.6	0	0.5	4.3	0	90.81
5	Cardboard	0	0	0	0	95	0	0	3.7	0.6	0	0	0	0.6	95.12
6	Milk Carton	0	0	0	0	0	94	0	2	0	0	0	0	4.1	93.88
7	Paper-White	0	0	1.7	0	0	0	91	7.4	0	0	0	0	0	90.91
8	Paper-Other	0	0	0	0	0.6	5.7	7	78	0	0	0	0	8.9	77.85
9	Plastic	0.4	0.4	0.6	1.7	0.5	1.9	1.4	0.4	85	0.2	0.4	1.6	5.3	85.36
10	Construction-Metal	0	0	0	0	0	0	2	0	0	96	2	0	0	96.08
11	Construction-Nonmetal	0	0	2.4	0	4.8	0	0	0	0	0	90	0	2.4	90.48
12	Metal	0.5	0	0	0.5	0	0.9	0.5	1.4	1.4	0	0.5	92	2.3	92.24
13	Mixed/Other	1.8	0.9	0	0	0	2.7	2.7	2.7	5.4	0	0.9	5.4	78	77.68
	Precision	96	98	95	98	94	88	86	81	89	100	93	89	76	

(**a**)

ID	True Class	Predicted Class 1	2	3	4	5	6	7	8	9	10	11	12	13	Recall
1	Appliance	94	0	0.6	0.6	0	0	0	0.6	2.3	0	0	1.1	1.1	93.71
2	Food Scrap	0	98	0	0	0	0	0	0.6	0.6	0	0.6	0	0	98.30
3	Foam	0	0	98	0	0	0	0.8	0	0.8	0	0.4	0.4	0	97.67
4	Glass	1.1	1.1	0	90	0	0	0	0	4.3	0	0	3.2	0	90.27
5	Cardboard	0	0.6	0	0	95	0.6	0	3	1.2	0	0	0	0	94.51
6	Milk Carton	0	0	0	0	0	84	0	10	2	0	0	0	4.1	83.67
7	Paper-White	0	0	0.8	0	0	0	95	1.7	2.5	0	0	0	0	95.04
8	Paper-Other	0	0	0	0	0	3.8	5.1	82	4.4	0	0	0.6	3.8	82.28
9	Plastic	0.2	0.1	0.3	0.6	0.1	0.3	0.6	0.3	93	0.1	0	1.1	3	93.49
10	Construction-Metal	0	0	0	0	0	0	0	2	3.9	92	2	0	0	92.16
11	Construction-Nonmetal	0	0	0	0	4.8	0	0	2.4	0	0	90	0	2.4	90.48
12	Metal	0.5	0	0	1.4	0.5	0.9	0	0.5	5	0	0	91	0.5	90.87
13	Mixed/Other	1.8	0.9	0	0	0	0	0.9	1.8	16	0.9	0	4.5	73	73.21
	Precision	96	97	98	97	95	94	93	78	68	99	97	89	83	

(**b**)

Figure 6. Normalized confusion matrix of the best model (RegNetY-1.6GF trained with KD-Garbage using SwinV2-T as the teacher) (**a**) and the teacher model (**b**). Each cell reports the fraction of ground-truth instances of the row class predicted as the column class. Diagonal entries equal per-class recall. Precision is computed from the normalized values. Blue shades indicate correct classification; red shades indicate incorrect classification.

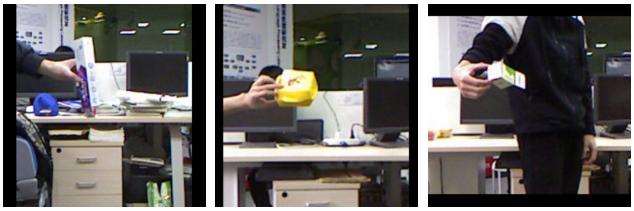
3.4.2. Misclassification Analysis

Paper-Other misclassified as Mixed/Other

The first row of Figure 7 illustrates representative examples of this pattern. Three subtypes account for the majority of these errors. First, paper items bearing patterns or color combinations that were underrepresented in the training set were confused with visually complex Mixed/Other items, suggesting that the training distribution for Paper-Other does

not yet adequately cover the full diversity of paper types. Second, highly glossy paper items that produced strong specular reflections were confused with plastic or metal, both of which also exhibit high-gloss surfaces. Surface gloss is an unreliable discriminator without controlled lighting. Third, images in which the central item was partially overlapped or visually associated with a background object from a different category caused the model to predict Mixed/Other, consistent with the class definition. These errors underscore the importance of physical deployment design: a sorting station equipped with uniform, diffuse illumination and a clean, single-color conveyor surface would eliminate the second and third sub-types, respectively.

Paper-Other misclassified as Mixed/Other



Paper-Other misclassified as Milk Carton

Figure 7. Representative examples of misclassified images.

Paper-Other Misclassified as Milk Carton

The second row of Figure 7 shows examples of this confusion. The misclassified items consistently share two visual characteristics with milk cartons: a rectangular shape and color schemes dominated by white, yellow, and blue—common palettes in food and beverage packaging. At the resolution and distance used for inference, the surface-texture differences between coated paperboard (milk carton) and uncoated paper (Paper-Other) are difficult to resolve. Deploying the classification system with a high-resolution or high-magnification camera would help expose the fine-grained surface details necessary to distinguish these classes more reliably.

Confusion Between Paper-White and Paper-Other

This confusion is primarily driven by two visual properties that are inherently difficult to capture at moderate image resolution: paper glossiness and ink color. Paper-White is defined as non-glossy white paper printed exclusively with black ink. Items printed with any color ink, or exhibiting surface gloss, belong to Paper-Other. Both glossiness and the presence of faint or small-area color printing can be easily lost at 224×224 pixel resolution, particularly when lighting is not carefully controlled. Improved illumination and a higher-resolution imaging setup are again the most direct paths to reducing this error.

Misclassification of Plastic

The dominant source of Plastic misclassification is confusion with Mixed/Other, arising from a specific structural ambiguity: many plastic food and beverage containers have a thin aluminum film laminated to their interior surfaces to provide a moisture and oxygen barrier. Because this internal coating is not visible from the outside, it is often impossible to determine from a single exterior image alone whether a container is pure

plastic or a plastic–aluminum composite, the latter of which belongs to Mixed/Other. This ambiguity is not unique to the model; it frequently challenges human sorters as well. A secondary source of Plastic errors is the sheer visual heterogeneity of the class itself: plastics span an enormous range of colors, surface finishes, shapes, and sizes, making it inherently difficult for the model to establish a compact decision boundary around the class.

Misclassification of Mixed/Other

The lower recall of the Mixed/Other class is, to a significant degree, structurally unavoidable. By definition, Mixed/Other is a catch-all category encompassing items made of two or more materials from different classes, as well as less common materials such as rubber, textiles, and ceramics. The resulting intra-class visual diversity is extreme, and the class lacks the consistent shape, texture, or color cues that anchor the model's predictions for more homogeneous categories. In addition, multi-material items are particularly susceptible to background-object interference: if the model considers part of the object as background, it may correctly identify the dominant material but fail to recognize the composite nature of the item, yielding a single-class prediction instead of Mixed/Other. This error mode is the mirror image of the background confusion observed for Paper-Other, and is similarly mitigated by ensuring physical separation of items from the background during imaging.

3.4.3. Implications for System Design

The misclassification analysis reveals that a substantial proportion of remaining errors are attributable to controllable deployment conditions or inherent class definitions rather than fundamental limitations of the model. Four interventions are recommended, spanning hardware design, physical setup, and data collection strategy.

First, diffuse and uniform illumination would reduce specular reflections on glossy surfaces, producing more consistent appearance for high-gloss plastics, metals, and glossy paper. This directly addresses the confusion between Paper-Other, Plastic, and Metal, and would also improve the differentiation between Paper-White and Paper-Other by making surface glossiness a more reliable visual cue.

Second, a clean, uniformly colored background—such as a monochrome conveyor belt surface—combined with a physical handling protocol that isolates one item at a time would eliminate background-object interference. This would reduce false Mixed/Other predictions for Paper-Other items, and conversely improve the recall of Mixed/Other itself by ensuring that composite items are not misidentified as single-material classes due to the model confusing part of the object as a background element.

Third, a higher-resolution or higher-magnification camera would expose fine-grained surface details that are currently below the effective resolution threshold. The most direct beneficiaries would be the Paper-White/Paper-Other boundary (ink color and glossiness), the Paper-Other/Milk Carton boundary (surface texture and coating), and Plastic identification (resin identification codes). To maximize detail without increasing computation, the image can be cropped so that the object fills the entire frame. This step only adds a negligible amount of computation as long as the background is uniform. This intervention requires no change to the model or the inference hardware.

Fourth, the confusion between Plastic and Mixed/Other caused by interior aluminum coatings cannot be resolved from exterior images alone and therefore calls for a different approach. A mechanical handling mechanism that exposes the interior of containers—such as an automated tilting or flipping stage on the conveyor—would allow the camera to capture both surfaces, enabling the model to detect the reflective aluminum lining and classify the item correctly.

Together, these interventions target the root causes identified in the misclassification analysis and are consistent with standard practice in industrial machine vision systems. Importantly, they are complementary: each addresses a distinct failure mode, and their combined effect would be expected to substantially reduce errors in the classes that currently fall below 90% recall.

4. Conclusions

This study introduced the KD-Garbage Framework, a KD pipeline designed to close the accuracy gap between high-capacity teacher models and lightweight student architectures for garbage image classification, without introducing any computational overhead at deployment. The framework was evaluated on a newly constructed 15,681-image dataset spanning 13 recycling- and disposal-aligned categories—a resource larger, more diverse, and more operationally representative than existing public benchmarks.

The key findings can be summarized as follows. First, KD improved the BA of every student architecture evaluated, with error reductions ranging from 10% to 25%. Second, RegNetY-1.6GF trained with the KD-Garbage Framework achieved the highest BA of 0.9129—exceeding both teacher models—while sustaining 6 FPS on a Raspberry Pi 5, confirming that the framework enables lightweight models to match or surpass the accuracy of their teachers without any inference-time cost. Third, SwinV2-T, despite being marginally less accurate than EfficientNetV2-S as a standalone classifier, proved to be the more effective teacher in the majority of student configurations, suggesting that structural diversity between teacher and student is a meaningful driver of distillation quality. Fourth, a temperature of $T = 3$ was found to be optimal or near-optimal across all configurations, providing a practical default for future applications of this framework. Fifth, the analysis of the confusion matrix identified three categories—Plastic, Paper-Other, and Mixed/Other—as the primary sources of remaining error, and traced the majority of misclassifications to controllable deployment factors such as illumination quality, background uniformity, image resolution, and viewing angle.

Taken together, these results demonstrate that the KD-Garbage Framework is an effective and practical approach for deploying high-accuracy garbage classifiers on resource-constrained hardware. Future work should investigate feature-based and relation-based distillation strategies, which transfer intermediate representations rather than output logits, as these may yield further improvements, particularly for smaller student architectures with larger capacity gaps relative to the teacher. The impact of higher-resolution inputs, richer data augmentation strategies, and targeted dataset expansion for the underperforming classes also merit systematic investigation. Finally, evaluating the framework on additional embedded platforms—including microcontrollers with dedicated neural processing units—would further characterize its practical deployment envelope.

Author Contributions: Conceptualization, P.J. and A.S.; methodology, P.T.; software, N.T.-A. and P.T.; validation, N.T.-A. and P.T.; formal analysis, N.T.-A. and P.T.; investigation, N.T.-A., P.T., P.J. and A.S.; resources, N.T.-A.; data curation, P.J. and A.S.; writing—original draft preparation, N.T.-A.; writing—review and editing, N.T.-A. and P.T.; visualization, P.T.; supervision, N.T.-A.; project administration, N.T.-A.; funding acquisition, N.T.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the College of Computing, Prince of Songkla University grant number COC6604032S. The APC was funded by Prince of Songkla University.

Data Availability Statement: The original data presented in the study are openly available on GitHub at https://github.com/NawanolT/Garbage-Dataset (accessed on 17 May 2026).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

MSW	Municipal solid waste
CNN	Convolutional neural network
ViT	Vision transformer
KD	Knowledge distillation
exp	Exponential function
log	Logarithm
FPS	Frames per second
M	Million
NAS	Neural architecture search
SE	Squeeze-and-Excitation
MBConv	Mobile inverted bottleneck convolution
FLOPS	Floating point operations per second
SW-MSA	shifted window-based multi-head self-attention
Log-CPB	log-spaced continuous positional bias
EPA	United States Environmental Protection Agency
RICs	Resin identification codes
LR	Learning rates
CLR	Cyclical learning rates
GB	Gigabytes
GPU	Graphics processing unit
BA	Balanced accuracy
TP	True positive
FN	False negative
AI	Artificial intelligence
USD	United States dollars
CPU	Central processing unit
ms	Millisecond
MB	Megabytes
RAM	Random-access memory
TL	Transfer learning
FT	Fine-tuning

References

1. Kaza, S.; Yao, L.; Bhada-Tata, P.; Van Woerden, F. *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050*; World Bank Publications: Washington, DC, USA, 2018.
2. Lloyd's Register Foundation. A World of Waste: Risks and Opportunities in Household Waste Management. 2024. Available online: https://www.lrfoundation.org.uk/publications/a-world-of-waste-risks-and-opportunities-in-household-waste-management (accessed on 23 January 2025).
3. World Health Organization. *Throwing Away Our Health: The Impacts of Solid Waste on Human Health—Evidence, Knowledge Gaps and Health Sector Responses*; World Health Organization: Geneva, Switzerland, 2025.
4. Hoornweg, D.; Bhada-Tata, P. What a Waste: A Global Review of Solid Waste Management. 2012. Available online: https://openknowledge.worldbank.org/entities/publication/1a464650-9d7a-58bb-b0ea-33ac4cd1f73c (accessed on 23 January 2025).
5. Aral, R.A.; Keskin, Ş.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)*; IEEE: Piscataway, NJ, USA, 2018; pp. 2058–2062.
6. Yang, M.; Thung, G. Classification of trash for recyclability status. *CS229 Proj. Rep.* **2016**, *2016*, 3. [CrossRef]

7. Ahmed, M.I.B.; Alotaibi, R.B.; Al-Qahtani, R.A.; Al-Qahtani, R.S.; Al-Hetela, S.S.; Al-Matar, K.A.; Al-Saqer, N.K.; Rahman, A.; Saraireh, L.; Youldash, M.; et al. Deep learning approach to recyclable products classification: Towards sustainable waste management. *Sustainability* **2023**, *15*, 11138. [CrossRef]
8. Gude, D.K.; Bandari, H.; Challa, A.K.R.; Tasneem, S.; Tasneem, Z.; Bhattacharjee, S.B.; Lalit, M.; Flores, M.A.L.; Goyal, N. Transforming urban sanitation: Enhancing sustainability through machine learning-driven waste processing. *Sustainability* **2024**, *16*, 7626. [CrossRef]
9. Bircanoğlu, C.; Atay, M.; Beşer, F.; Genç, Ö.; Kızrak, M.A. RecycleNet: Intelligent waste sorting using deep neural networks. In *Proceedings of the 2018 Innovations in Intelligent Systems and Applications (INISTA)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1–7.
10. Chen, Z.; Yang, J.; Chen, L.; Jiao, H. Garbage classification system based on improved ShuffleNet v2. *Resour. Conserv. Recycl.* **2022**, *178*, 106090. [CrossRef]
11. Tian, X.; Shi, L.; Luo, Y.; Zhang, X. Garbage classification algorithm based on improved mobilenetv3. *IEEE Access* **2024**, *12*, 44799–44807. [CrossRef]
12. Kang, Z.; Yang, J.; Li, G.; Zhang, Z. An automatic garbage classification system based on deep learning. *IEEE Access* **2020**, *8*, 140019–140029. [CrossRef]
13. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations (ICLR) 2021, Vienna, Austria, 4–8 May 2021.
14. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
15. Tan, M.; Le, Q. EfficientNetV2: Smaller models and faster training. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2021; pp. 10096–10106.
16. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin transformer v2: Scaling up capacity and resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12009–12019.
17. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
18. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
19. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2019; pp. 6105–6114.
20. Radosavovic, I.; Kosaraju, R.P.; Girshick, R.; He, K.; Dollár, P. Designing network design spaces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10428–10436.
21. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
22. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
23. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [CrossRef]
24. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. FitNets: Hints for thin deep nets. *arXiv* **2014**, arXiv:1412.6550.
25. Park, W.; Kim, D.; Lu, Y.; Cho, M. Relational knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3967–3976.
26. Chen, G.; Choi, W.; Yu, X.; Han, T.; Chandraker, M. Learning efficient object detection models with knowledge distillation. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–10.
27. Cho, J.H.; Hariharan, B. On the efficacy of knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 4794–4802.
28. Wilson, T. Plastic Recycling: Why Are Some Types Harder to Process Than Others? 2026. Available online: https://www.bywaters.co.uk/sustainability/plastic-recycling-why-are-some-types-harder-to-process-than-others (accessed on 11 June 2026).
29. American Forest & Paper Association. Do You Know How to Recycle Milk Cartons? 2022. Available online: https://www.afandpa.org/news/2022/do-you-know-how-recycle-milk-cartons (accessed on 11 June 2026).
30. RecyClass. Design for Recycling Guidelines. Available online: https://recyclass.eu/protocols-guidelines/design-for-recycling-guidelines/ (accessed on 11 June 2026).

31. RegSurance. PPWR Recyclability Grades Explained: A-to-E Packaging Compliance, Deadlines & Business Action Plan. Available online: https://regsurance.com/ppwr-recyclability-grades-explained-a-to-e-packaging-compliance-deadlines-business-action-plan/ (accessed on 11 June 2026).
32. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18-23 June 2018; pp. 7132–7141.
33. lulia. cd_ver3 Computer Vision Dataset. 2025. Available online: https://universe.roboflow.com/iulia/cd_ver3 (accessed on 23 January 2025).
34. Mohamed, M. Garbage Classification. 2021. Available online: https://www.kaggle.com/datasets/mostafaabla/garbage-classification (accessed on 23 January 2025).
35. Serezhkin, A. Drinking Waste Classification. 2020. Available online: https://www.kaggle.com/datasets/arkadiyhacks/drinking-waste-classification (accessed on 23 January 2025).
36. Wang, T.; Cai, Y.; Liang, L.; Ye, D. A multi-level approach to waste object segmentation. *Sensors* **2020**, *20*, 3816. [CrossRef] [PubMed]
37. Modern Academy for Engineering. Paper Computer Vision Dataset. 2023. Available online: https://universe.roboflow.com/modern-academy-for-engineering-enkza/paper-iybve (accessed on 23 January 2025).
38. Proença, P.F.; Simões, P. TACO: Trash annotations in context for litter detection. *arXiv* **2020**, arXiv:2003.06975.
39. Cen, A. Waste Images from Sushi Restaurant. 2020. Available online: https://www.kaggle.com/datasets/arthurcen/waste-images-from-sushi-restaurant (accessed on 23 January 2025).
40. school2. Styrofoam Computer Vision Dataset. 2023. Available online: https://universe.roboflow.com/school2-iqyyf/styrofoam-l0ihw (accessed on 23 January 2025).
41. Food. Food Waste Detection Computer Vision Dataset. 2024. Available online: https://universe.roboflow.com/food-1b74y/food-waste-detection-jghxg (accessed on 23 January 2025).
42. Thaifood. dish-segmentation Computer Vision Dataset. 2024. Available online: https://universe.roboflow.com/thaifood-iiosc/dish-segmentation-97jtv (accessed on 23 January 2025).
43. Kunwar, S. The Garbage Dataset (GD): A Multi-Class Image Benchmark for Automated Waste Segregation. *arXiv* **2026**, arXiv:2602.10500.
44. United States Environmental Protection Agency. Advancing Sustainable Materials Management: 2018 Fact Sheet. 2020. Available online: https://www.epa.gov/sites/default/files/2021-01/documents/2018_ff_fact_sheet_dec_2020_fnl_508.pdf (accessed on 11 June 2026).
45. Ionkova, K.M. Ten Charts that Explain the Global Waste Crisis. 2026. Available online: https://blogs.worldbank.org/en/sustainablecities/what-a-waste-3-charts (accessed on 11 June 2026).
46. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
47. Smith, L.N. Cyclical learning rates for training neural networks. In *Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*; IEEE: Piscataway, NJ, USA, 2017; pp. 464–472.
48. Paszke, A. Pytorch: An imperative style, high-performance deep learning library. *arXiv* **2019**, arXiv:1912.01703.