Convergence
A Journal for Young Researchers

ORIGINAL RESEARCH

Received: 8-26-25; Accepted: 10-10-25;
Managing Editor: Jonas Katona;
Action Editor: Jonas Katona

# WISE: An Adaptive YOLO Ensemble for Accurate E-Waste Object Detection

**Vishnu Muthuraman**

Westwood High School, Austin, Texas, United States of America

**Abstract**

Electronic waste (E-waste), which contains hazardous chemicals and valuable precious metals, has increased tremendously over the past few decades. The world generated 62 million tons of E-waste in 2022, and this continues to increase rapidly. Still largely relying on manual sorting, only 22.3% of E-waste was documented as collected and recycled, leaving the rest to enter landfills. There is an urgency for automated techniques to help. Object detection algorithms, traditionally, focus on improving accuracy but end up weighing false negatives (FNs) and false positives (FPs) the same. However here, FNs like batteries being labeled as non-E-waste are far more hazardous.

This paper makes three contributions to the literature on E-waste. First, we develop a new calibrated metric to help pick the best algorithm for E-waste detection. This metric provides a basis upon which even future algorithms can be evaluated. Second, we revisit existing algorithm evaluations for the specific purpose of E-waste detection. We evaluate several standard, tweaked, and ensemble methods. Finally, we propose a smart ensemble (WISE: Waste-focused Integrated Smart Ensemble) whose weights are learned using machine learning, to minimize the cost and impact of E-waste disposal. The grand goal of these developments is to help improve public health and the reusability of precious metals by enabling more efficient recovery and processing of E-waste.

**Keywords:** object detection, You Only Look Once (YOLO), electronic waste (E-waste), artificial intelligence (AI), machine learning (ML), ensemble methods, threshold calibration, false negatives

## 1. Introduction

Electronic waste, or E-waste, consists of numerous hazardous materials, including plastics, lead, mercury, cadmium, arsenic, etc. (Vats & Singh, 2014). E-waste that ends up in landfills is responsible for multiple health hazards, especially in developing countries. Examples of these hazards include fetal loss, prematurity, low birth weight, abnormal thyroid function, neurobehavioral disturbances, and genotoxicity (Noel-Brune et al., 2013). Additionally, the E-waste that ends up in landfills consists of valuable metals like aluminum (Al), gold (Ag), palladium (Pd), and platinum (Pt) (Vats & Singh, 2014) that can be extracted and reused to save money. On just a single TV board, 7% of its salvage value comes from silver, 33% from gold, and 7% from palladium (Fornalczyk et al., 2013). In a mobile phone, 11% comes from silver, 71% comes from gold, and 11% comes from palladium, totaling 93% of the phone's salvage value that comes from just precious metals (Fornalczyk et al., 2013).

In 2022, we generated 62 million metric tons of E-waste, and this is expected to escalate to 74 million metric tons by 2030 (Singh & Parimala S, 2025). Preventing hazardous materials from entering landfills is critical. It is almost the only way to stop these chemicals from eventually finding their way into animal and human bodies. Facilities and techniques exist to extract valuable metals from E-waste and dispose of the rest appropriately. However, the challenge lies in identifying E-waste and keeping up with the volume. We currently depend on manual sorting, afforded in part by cheap labor from developing and underdeveloped countries, to help prevent hazardous material from reaching landfills.

The ethical issues involved in dispatching the E-waste from developed nations to socioeconomically disadvantaged nations cannot be overstated. In developed nations, medical ailments caused by E-waste are often not seen every day. As a result, the problem remains hidden, and even conscientious citizens rarely notice it. In contrast, underprivileged localities are often the least equipped environments to cope with these imported toxic substances. They also lack the medical resources necessary to tackle the rising health crisis. Many governments have recognized this issue and have attempted to use systems and regulations that cut down on E-waste. However, even with such efforts and often challenged funding, growth in E-waste still seems to outpace our efforts to prevent it from entering our landfills and ecosystems.

Automated techniques, though not perfect, can be of huge benefit in augmenting manual sorting. Current research has already developed artificial intelligence (AI) models that have been trained and evaluated for generic object detection. Pictures of labeled objects are processed and used to train neural network models. Training involves the incremental adjustment of the neural network's weights to match the known labels. Once trained, the final network with adjusted weights is used to detect and label unlabeled objects. These neural net models also create bounding boxes for each object and estimate labels for each object. Bounding boxes are tight rectangles around detected objects. This can even facilitate the extraction of E-waste more easily, rather than debating whether an object should be classified under this name.

The problem with existing algorithms and their benchmarks is that they focus primarily on accuracy or, at most, precision and recall as measures of success. Accuracy is well warranted for generic object detection, where both false positives (FP) and false negatives (FN) are equally bad. Other variants like precision and recall are well warranted when only FPs (e.g., screening resumes) or FNs (e.g,. screening for cancer) are to be minimized. From an E-waste perspective, it is less harmful to label a banana peel as E-waste, not allowing it to go to the landfill directly by routing it to manual sorting. It is much more harmful to label a battery, full of toxic chemicals, as non-E-waste, allowing it to enter the landfill and ecosystem directly. The clear objective would be to allow FPs, like identifying the banana peel as E-waste, instead of allowing FNs like batteries to reach the landfill. Choosing a standard measure like recall will not help either, since it can be maximized by labeling everything as positive, thereby making FNs zero. This will make automation useless, as it will keep over-burdening manual sorting—the exact reason we seek automation and augmentation.

Specifically in terms of literature on AI for waste detection, deep learning models have been used for solid waste classification (Adedeji & Wang, 2019; Majchrowska et al., 2022; Oza et al., 2025). The *TrashNet* (Thung & Yang, 2017) dataset was created as part of a student-led project in 2017 and has been a popular dataset for trash classification benchmarking. However, this dataset is not relevant for E-waste classification since the only six classes included are glass, paper, cardboard, plastic, metal, and trash. The dataset contains 2527 images and has been used for benchmarking in some research papers (Khan et al., 2024; Rahim et al., 2024). Another popular direction of research has been in identifying waste in the wild. The TACO dataset (Proença & Simões, 2020, 2023) provides 1500 images of trash in various locations, from the beaches to city streets. These are used to train algorithms that can find trash in pictures taken from, for example, an automated garbage collector (Fan et al., 2023; Promboonruang et al., 2024; Song et al., 2025). Findings from these two trash-related applications demonstrate that off-the-shelf general-purpose detectors trained on natural image corpora can perform poorly. These, however, are great proof-of-concept trash classification systems being used in related applications. Motivated by this, we frame the E-waste identification problem as a cost-sensitive identification problem and focus on fine-tuning and developing variants that explicitly minimize our costs.

The YOLO (You Only Look Once) family of detectors, though created in 2016 (Redmon et al., 2016), only became a popular go-to choice recently. These one-stage detectors are very effective, specifically for integration into real-time waste sorting, such as with robotic arms (Ibrahim et al., 2023; Paudel et al., 2024). However, in terms of effective sorting methods for identifying E-waste, there exists only one recent paper (Rajeev et al., 2025) that simply compares different YOLO methods in terms of accuracy, and the reported poor performances underscore the need for improved methods.

Prior waste or E-waste detection studies predominantly optimize for accuracy, precision, or recall rather than application-level costs of mistakes. None, to our knowledge, considers the application-level nuances of differentiating the costs of FP and FN. However, in classical Machine Learning, both minimizing expected misclassifications cost (EMC) when costs of FP and FN differ (Domingos, 1999; Elkan, 2001; Sheng & Ling, 2006) and creating better performing ensemble are popular research topics (Bodla et al., 2017; Lin et al., 2017; Solovyev et al., 2021; Wu & Zhu, 2013).

### 1.1. Contributions and Outline

In this paper, we:

(1) Create a new metric, E-waste Misclassifications Cost (EMC), calibrated to help pick the best algorithm for E-waste detection, providing a basis upon which even future algorithms can be evaluated. EMC weights the FNs substantially greater than the FPs but does not completely nullify the FPs. We compute the coefficients based on accounting cost estimates. We seek a measure that gives us an estimate of the total cost of misidentification, lending itself well to our case since the additional cost of manually sorting an FP should be contrasted against the estimated environmental cost of sending a FN to a landfill.

(2) Revisit existing algorithm evaluations for the specific purpose of E-waste detection. We evaluate several standards and tweaked ensemble methods. We also analyze several algorithms. The first four algorithms are standard and popular implementations (YOLOv3, YOLOv4, YOLOv5, YOLOv8). The second four are our tweaks to the first four to reduce FNs (YRFNv3, YRFNv4, YRFNv5, YRFNv8). We further create two more ensembles that we call EYOLO and EYRFN. These estimate labels by polling the 8 previous variants and the 4 reduced FN variants.

(3) Propose the Waste-focused Integrated Smart Ensemble (WISE), whose weights are learned using machine learning to minimize the cost and impact of E-waste disposal. Ensemble models combine multiple models to make a final recommendation or decision. Since this is a machine learning algorithm, we do in-sample comparisons with other methods and also make in-sample versus out-of-sample comparisons to ensure that there is no overfitting.

### 1.2. Object Detection

This subsection provides a brief, practical primer on object detection to orient the reader to the components (backbone, neck, head) used later. We emphasize elements that matter for E-waste sorting speed, small object recall, and thresholding to motivate our choice of YOLO variants.

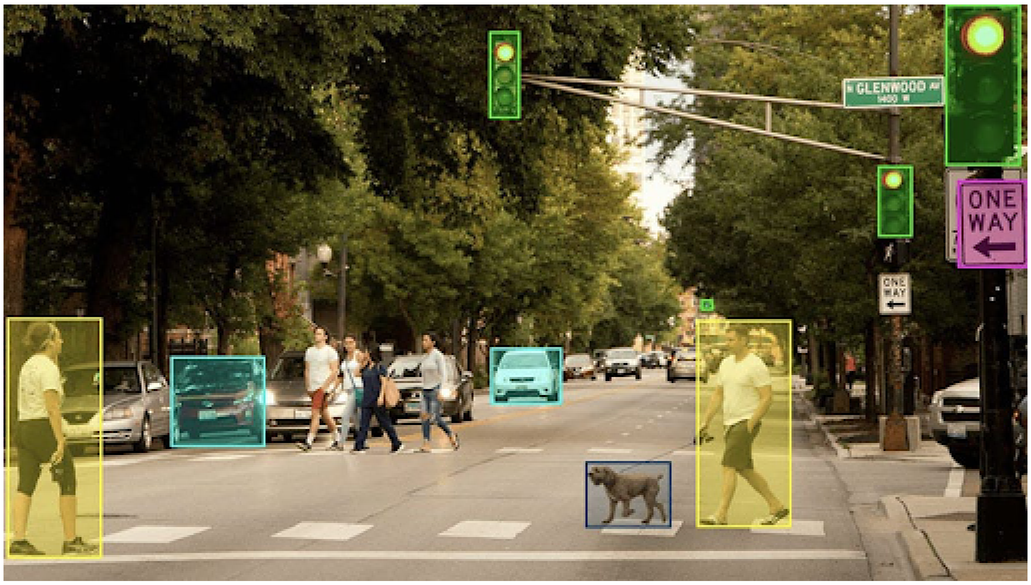


**Figure 1:** Object Detection example with colored bounding boxes: yellow = person, blue = vehicle, green = traffic light.

*Note:* Boxes illustrate localization outputs before non-maximum suppression. Source: Potter, 2022.

Object detection is the process of analyzing an image, localizing objects, and classifying what those objects are. It is utilized in numerous applications such as face, pedestrian (Figure 1), and object detection (Li & Cao, 2020). Convolutional Neural Networks (CNNs) are a specific type of neural network that play a very significant role in the object detection space. They are essentially classical feed-forward neural networks with additional layers called convolutional layers. These convolutional layers help in any kind of computer vision by applying trained averaging kernels (small weight matrices that slide across the image to compute local features). This kernel performs computations on the image's pixel values of color and intensity. By performing these computations, the CNN is able to create a feature map that consists of channels, or representations of different learned features such as lines, edges, and curves.

When an image is processed to look for certain objects, it is first transformed into a form that can be passed on as an input to a neural network (NN). This means the image must be processed, resized, and normalized for the neural network to take in the image. All inputs are also normalized to [0, 1] by dividing by 255. A trained CNN can then be used to identify objects using intermediate steps that rely on extracted visual features like edges, corners, and textures (Parti, 2024). A Region Proposal Network, a smaller CNN that utilizes extracted visual features, can be used to create bounding boxes around potential object locations. Objects encapsulated in these bounding boxes are then identified, and localization is used to calibrate the bounding boxes to exclude unnecessary parts of the image. Non-maximum suppression, a mathematical algorithm, keeps only the most confident detections, excluding false positives and improving accuracy. The bounding boxes are further refined, and object identification is finalized (Parti, 2024).

There are two main categories of CNN object detectors: two-stage and one-stage (Wu et al., 2024). In the two-stage detector (Figure 2), the algorithm first makes object proposals, guessing where the object appears in the image. The second stage classifies the object and refines the bounding box. By contrast, one-stage detectors completely skip this object proposal step. They go straight to predicting the bounding boxes and classifying objects. Overall, two-stage detectors are more accurate but computationally much more expensive and slower in comparison to one-stage detectors (Wu et al., 2024). One-stage detectors achieve favorable speed/accuracy trade-offs, often 2–5 times faster, making them practical for real-time applications. Prior work quantifies these trade-offs across architectures and scales (Huang et al., 2017; Liu et al., 2016).

You Only Look Once (YOLO) is the most popular one-stage detector that uses a CNN architecture. It uses the CNN to predict where the bounding boxes for objects should be placed. It then assigns the probability of each object in a bounding box being an object of a specific class, which are called class-probabilities. For example, an image of an animal could be classified as a dog with a probability of 0.1, and as a cat with a probability of 0.9. YOLO is known for its computational speed and high accuracy. There are numerous versions of YOLO, which all have distinct advantages and disadvantages. YOLO is the primary focus of this paper due to its availability, open-source license, and popularity.

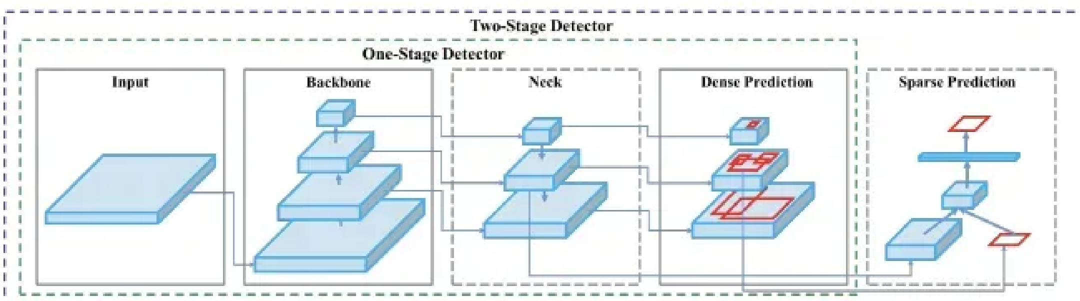


**Figure 2:** One-stage vs. two-stage detection. Source: Solawetz, 2024

*Note:* Two-stage proposes regions, then classifies and refines; one-stage predicts boxes and classes in one pass.

### 1.3. Models

We summarize the specific YOLO variants evaluated, focusing on design differences that influence E-waste detection (e.g., multi-scale heads, CSP/C2f backbones, anchor-free prediction) and their practical trade-offs.

There are eight versions of the YOLO, starting with version 1 proposed by Redmon (Redmon et al., 2016). Versions 2 and 3 are improved versions with residual blocks (Redmon & Farhadi, 2018). YOLO version 4 was released by A. Bochkovskiy, who took over after Redmon retired (Bochkovskiy et al., 2020). This included major performance improvements and new training tricks.

Version 5 (Jocher & Ultralytics, 2020) and version 8 (Jocher & Ultralytics, 2023) were developed by Ultralytics and do not have an official research paper. Version 5 reimplemented YOLO in PyTorch, while version 8 improved on version 5 by unifying the code base and modernizing the architecture. YOLO version 6 was developed by Meituan (Meituan Vision AI Department, 2022), was optimized for industrial deployment, and does not have a formal research paper either. YOLO version 7 (Wang et al., 2022) introduced several new features, such as extendable, trainable bag-of-freebies and architectural refinements. The most popular among the YOLOs are version 3 (last official Redmon release), version 4 (huge leap in accuracy), version 5 (industry standard, ease of use), and Version 8 (latest, cutting-edge, unified framework).

**YOLOv3**

The backbone structure, the part of the neural network that extracts features, utilized by YOLOv3s is Darknet-53 (Cheng et al., 2021; Redmon & Farhadi, 2018). Darknet53 is a CNN with 53 layers that uses residual connections (Figure 3), allowing for input to skip over layers to improve efficiency. Convolution blocks, layers of the CNN that extract features, use a sequence of Convolution, Batch Normalization, and Leaky ReLU (CBL). Convolution uses kernels to detect features, Batch Normalization allows CNNs to stabilize and train faster, and Leaky ReLU—a mathematical function—helps the CNN learn complex patterns. All YOLO methods include normalization of each channel to [0, 1] and do not have per-channel mean or standard deviation standardizations.

The head structure, or the end of the model, of YOLOv3 predicts whether objects are small, medium, or large. In other words, this helps the model detect small objects, medium objects, and large objects. Anchor boxes, or boxes that have been predefined at certain sizes and shapes, are placed in every grid cell to facilitate bounding box predictions. Finally, the bounding boxes are placed around the target object(s).

**YOLOv4**

YOLOv4's backbone structure is CSPDarknet53 (Bochkovskiy et al., 2020). Darknet53 extracts the features while Cross Stage Partial (CSP) splits channels from the feature map such that half the channels are able to use the residual connections, and the other half continue through the several layers. This ensures that there is a balance between performance and efficiency. The neck structure is the part of the object detection model that combines details that have been collected, allowing for a better understanding of the image. In the neck structure of YOLOv4, Spatial Pyramid Pooling (SPP)—a pooling module, reducing the size of the feature map and keeping on essential parts, in the CNN—takes the feature maps that were created from CSPDarknet53, and underscores important patterns. It also looks at the feature maps in different sizes, helping the SPP pick up minute details as well as the larger context. Path Aggregation Network (PAN), a component of the CNN, focuses on locating where an object occurs in the image.

The head structure for YOLOv4 is the same head structure used for YOLOv3 due to its efficiency. Although the head structures are the same, YOLOv4 has an improved backbone and neck structure in comparison to YOLOv3.

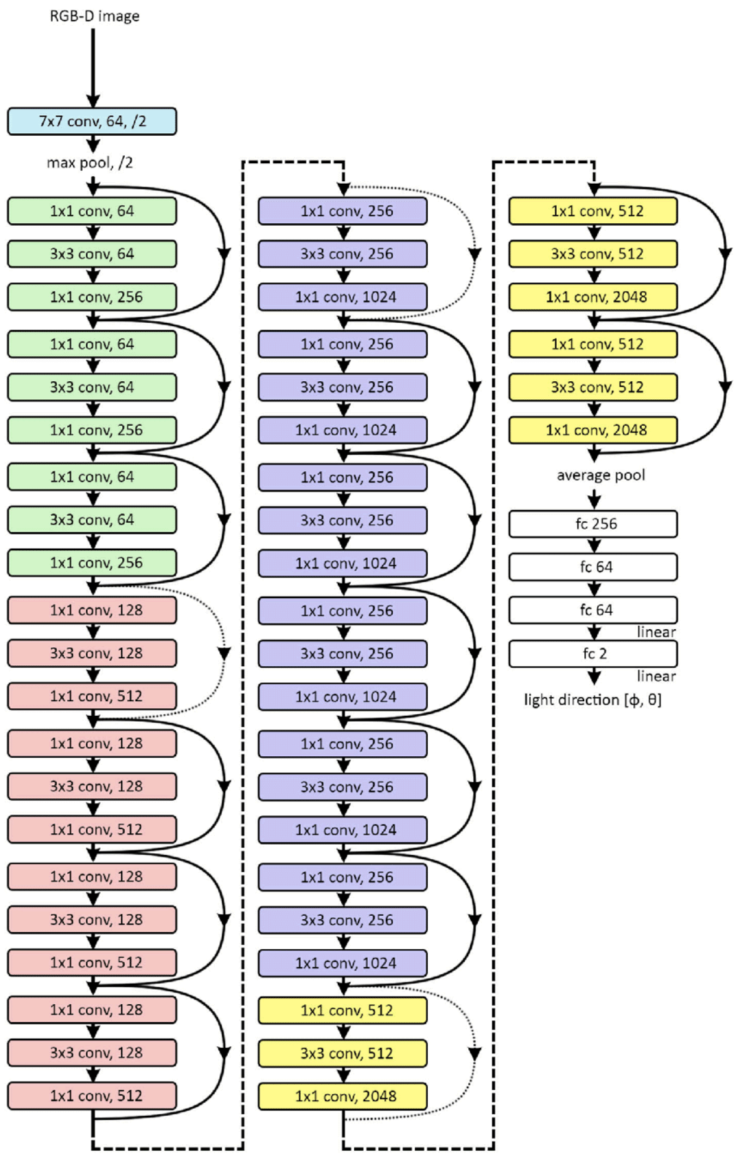


**Figure 3:** Darknet53 architecture. Source: Kán and Kaufmann, 2019

*Note:* Deep convolutional network applied to RGB-D input, using residual connections across multiple layers to extract hierarchical features for downstream object detection tasks.

**YOLOv5**

The backbone structure—the part of the neural network that extracts features—utilized by YOLOv5s is inspired by CSPDarknet53 (Feng et al., 2023; Jocher & Ultralytics, 2020). Additional convolution layers are used to extract the image features to a greater extent.

To improve its performance, YOLOv5s records its mistakes, which can fall into one of three categories: bounding box regression loss, where the bounding box is placed away from the correct spot from where the object is located, as seen in Figure 4; target confidence loss, where YOLOv5s claims an object is apparent when it is not; and classification loss, when an object is mislabeled.

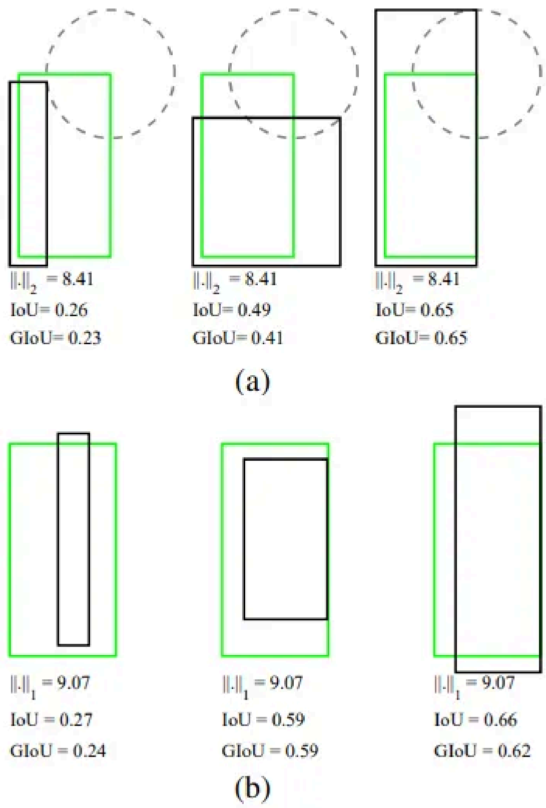


**Figure 4:** Bounding boxes and overlap measures. Source: "Bounding Box Regression Loss", n.d.

*Note:* (a) IoU examples with partial overlap between ground-truth and predicted boxes. (b) GIoU extends IoU to provide informative gradients when boxes do not overlap, improving training stability.

**YOLOv8**

YOLOv8s uses a backbone inspired by CSP. When the channels of the feature map are created, instead of CSP splitting the channels up, YOLOv8s uses Cross Stage Partial with Fused Layers (C2f). C2f splits the feature map channels and fuses the features from the channels progressively. By gradually fusing channels, the object detector is able to grab richer details rather than fusing all the channels in one shot like CSP does. A Spatial Pyramid Pooling-Faster (SPPF) module, a component in the CNN, is used to see objects in different sizes. Essentially, it uses windows—parts of the image—of different sizes to grab minute details as well as the bigger picture. For example, a small window can be used to identify that a dog's fur is more wavy than straight. By contrast, a big window would look at the dog entirely and identify that the object is in the shape of a dog.

In the head of YOLOv8s, Upsample layers (U layers in Figure 5) are used to improve the resolution of the feature maps. Doing this helps preserve more details, which can result in greater precision and accuracy. Three separate branches, or pathways, are used for *objectness*, that is, predicting if an object is apparent; classification, identifying what object that is; and regression, predicting where the bounding box should be placed.

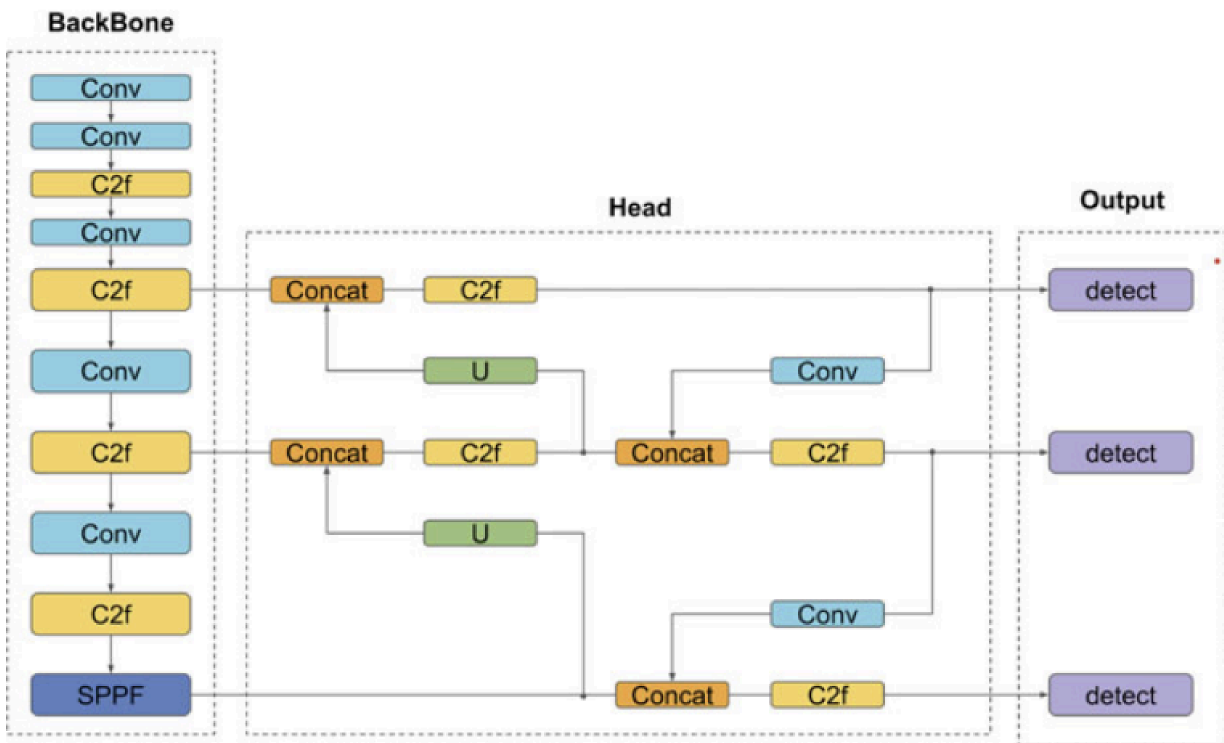


**Figure 5:** YOLOv8 architecture: the backbone extracts multi-scale features; the head aggregates features; the output layer generates object-detection predictions. Source: Jain et al., 2025

*Note*: Abbreviations: C2f = Cross Stage Partial with fused layers; SPPF = Spatial Pyramid Pooling-Faster; PAN = Path Aggregation Network; U = Upsample; Conv = Convolution.

The comparisons are summarized in Table 1. As discussed in the preceding subsections, YOLOv3, v4, v5s, and v8s each represent critical points in the evolution of the YOLO family of detection algorithms. v3 is the canonical baseline, v4 is the first significant performance jump, and v5s is the widely adopted PyTorch re-implementation. Finally, v8s is the most recent unified architecture. We picked the "s" variants for v5 and v8 to align better with the real-time, resource-constrained, conveyor-belt-based sorting scenario. This spectrum of methods allows our analysis to capture both historical and state-of-the-art behaviors under the E-waste misclassifications cost framework.

In summary, the four selected variants span a stable baseline (v3), a major performance jump (v4), a widely deployed PyTorch baseline (v5s), and a modernized anchor-free design (v8s), giving coverage of historical and state-of-the-art behaviors under EMC.

**Table 1:** Concise comparison of YOLO variants evaluated.

| Version, Year | Backbone | Neck | Head | Key innovations and training updates | Pros / Cons (especially for E-waste) |
|---|---|---|---|---|---|
| YOLOv3, 2018 | Darknet-53 (residual) | FPN-like multi-scale fusion | 3-scale heads with anchors | Stable baseline; incremental over YOLOv2; residual connections improve depth | + Robust and widely reproduced baseline. − Weaker recall for small objects, limiting E-waste sensitivity |
| YOLOv4, 2020 | CSPDarknet-53 | SPP + PAN | 3-scale heads with anchors | Introduced "bag of freebies/costlies;" CIoU/GIoU; mosaic augmentation; stronger training recipe | + Major accuracy gain, better on small/medium objects. − Higher complexity and compute demand. |
| YOLOv5s, 2020 | CSP-inspired (Py-Torch) | PANet | 3-scale heads with anchors | PyTorch re-implementation; small "s" variant optimized for speed; auto-augment, EMA; easy deployment | + Flexible, lightweight, edge-ready. − "s" favors speed over recall; in our tests, default thresholds missed subtle E-waste (see YRFNv5). |
| YOLOv8s, 2023 | C2f (CSP-fused) | SPPF + PAN | Decoupled obj/cls/reg heads (anchor-free default) | Modernized architecture; anchor-free prediction; cleaner training pipeline; improved feature reuse | + Strong precision, efficient pipeline. − In our results, default setup under-recalled E-waste; improved when FN-reduction applied (YRFNv8). |

*Note*: "s" variants (v5s, v8s) are chosen to match real-time, resource-constrained conveyor-belt settings.

## 2. Methodology

### 2.1. Standard evaluation measures

Very often, when analyzing object detection models, the common metrics utilized are accuracy, precision, and recall. Sometimes, though more rarely, F1 scores are used as well. Accuracy measures the percentage of predictions that are correct. Accuracy, given in Equation (1), is the most popular measure for classification and is perfect for cases where all misclassifications hurt the measure the same way.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

- True Positive (TP): E-waste is successfully identified by the object detector as E-waste.
- True Negative (TN): Non–E-waste is successfully identified by the object detector as non-E-waste.
- False Positive (FP): Non–E-waste is incorrectly identified by the object detector as E-waste.
- False Negative (FN): E-waste is incorrectly identified by the object detector as non-E-waste.

Precision, given by Equation (2), is also called the positive predictive value and measures how many were correct among all those predicted positive. In other words, it does not account for false negatives (FNs). False positives (FPs) alone hurt the measure. This is very useful in applications such as resume screening, where even missing out on a good resume is not as harmful as hiring a person who is not the right fit.

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

Recall, given in Equation (3), is also called the Sensitivity or True positive rate. It asks how many the model correctly detected out of all the objects that were actually positive. It tells us the total detected E-waste fraction among all E-wastes. In other words, it does account for FPs, and only FNs hurt the measure. This is perfect for applications such as cancer screening. In our case, we do include FPs as well, just less so. That is, labeling a tennis ball as E-waste (FP) is bad, but not nearly as bad as labeling a battery as a non-E-waste (FN). Moreover, recall can be trivially maximized by labeling every item E-waste. Hence, recall can never be used as an objective when training ML methods.

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

Finally, the F1-score, Equation (4), balances precision and recall, and is relevant in applications where both are equally expensive.

$$F_1 = 2\,\frac{Precision\cdot Recall}{Precision+Recall} \tag{4}$$

The F1-score is the harmonic mean of precision and recall and is large only when both are large, discouraging degenerate solutions that maximize one at the expense of the other.

### 2.2. Reduced FN Models

Appending to the list of models described in the "Models" section, we create a duplicate of every model, but lower the confidence levels from 0.5 to 0.25 to favor detection. By lowering the confidence level, the number of FNs is guaranteed to decrease since it increases the overall positivity rate. Confidence level is the minimum confidence required for the object detector to make the final decision that the target object is in the picture. We name these models YRFNv3, YRFNv4, YRFNv5, and YRFNv8, respectively. YRFN stands for YOLO with Reduced False Negatives.

### 2.3. EMC: A New Performance Metric for E-waste Detection

As we have already discussed, none of the usual measures of success used in classification lend themselves well to E-waste detection. Accuracy, for example, undermines this situation completely, considering both FNs and FPs the same. Our proposed new measure seeks to balance FNs and FPs by accounting for the costs of each type of misclassification.

To determine these costs, we looked at the cost of offsetting the environmental consequences and the costs of inspecting false positives. Cost-benefit analysis literature (Yang et al., 2021) highlighted that the cost to offset the environmental consequences was on average $4.00 USD/kg of E-waste in 2021. Adjusted for inflation, this is $5.95 USD/kg today (2025). Averaging common trash product weights (Empa - Swiss Federal Laboratories for Materials Science and Technology, 2025), we obtained $11.71 USD/item. To find the cost of false positives, we took the average worker's hourly salary, and assuming they spend five minutes on each object, we evaluated the appropriate cost. The average recycling worker's hourly salary in China is $0.17 USD (Wikipedia contributors, 2024), while it is $32,000/year in the US, according to ZipRecruiter. Though a wide variation, averaging five objects per minute, it gave us $0.67 USD/item. These are rough estimates, and hence it is important to note that this is only for comparison purposes. Our new method's calibration and performance do not depend on the exact values, only that we need non-zero values for both.

Below is the new metric that we propose, the E-waste misclassification cost (EMC):

$$EMC = \frac{11.71\, FN + 0.67\, FP}{TP + FN} \tag{5}$$

The cost coefficients of false negatives ($c_{FN}$) and false positives ($c_{FP}$) are user-settable and may vary by region and waste stream. Note that multiplying both by the same scalar leaves rankings unchanged; only their ratio matters. Later, in the results section, we also analyze the sensitivity of the results to this ratio.

### 2.4. Ensembles

An ensemble method is the combination of multiple models to make a prediction, or the wisdom of the crowd. Different models may give different results on whether they're able to detect an object or not. To make the final decision, a voting system is used such that if more models claim that an object is present, then the final prediction is that the object is present, and vice versa.

In this study, we created an ensemble of YOLOv3, YOLOv4, YOLOv5, and YOLOv8, and another ensemble of RFNv3, YRFNv4, YRFNv5, and YRFNv8. If at least three of the models detected E-waste as present, then the final decision would be that

E-waste is present. We call them EYOLO and EYRFN, respectively. These ensembles estimate labels by polling the 4 YOLOs and the 4 YRFNs, respectively.

### 2.5. WISE: Waste-focused Integrated Smart Ensemble

Consider an ensemble that does not allow equal weighting. That is, the wisdom of the crowd is not equally weighted. The better an underlying method, the more its weight. Further, we let machine learning pick the weights that minimize the total EMC as defined by Equation (5).

More specifically, let $x_i \in \{0, 1\}^8$ denote the vector of model-level detections (YOLOv3, YOLOv4, YOLOv5s, YOLOv8s; YRFNv3, YRFNv4, YRFNv5s, YRFNv8s) for image $i$, and let $y_i \in \{0, 1\}$ be the ground truth label (1 for E-waste, 0 for non-E-waste). WISE learns a weight vector $w \in R^8$ and bias $b$ by minimizing the expected misclassification cost with an $l_1$ penalty:

$$\frac{1}{N}\sum_{i=1}^{N}\left(c_{FN}\, y_i\left[1 - \hat{y}_i\right] + c_{FP}\left(1 - y_i\right)\hat{y}_i\right) + \lambda\|w\|_1 \tag{6}$$

where the prediction is made as

$$\hat{y}_i = 1\left[w^{\top}x_i + b \geq 0\right]. \tag{7}$$

We set $c_{FN} = 11.71$ and $c_{FP} = 0.67$ (as derived from Equation (5), the EMC metric). Optimization is performed with the Adam optimizer (batch size 256, learning rate $10^{-3}$), with early stopping on a stratified validation split. The threshold 0 is without loss of generality since the bias b can be appropriately adjusted.

The learned solution is sparse, with only two non-zero weights: YRFNv4 receives a 0.6 weight and YRFNv8 receives a 0.4 weight. This sparsity results from the $l_1$ penalty and is consistent with the high performance of these models in Table 2.

Note that since we are learning these weights to minimize a loss, we have to train the weights on a training subset of the data and test it on the rest of the test set. We use a standard randomization of 80/20 to split the data for testing versus training. We also tune λ over the grid $\{0, 10^{-4}, 10^{-3}, 10^{-2}\}$. For comparisons to other methods above that did not require training/testing splits, we report and compare the averages over the entire dataset. This obviously risks that the performance improvements could be biased due to the inclusion of in-sample and out-of-sample data. To ensure that this is not the case, we compare the EMC of the test set to that of the training to ensure that magnitudes are similar.

### 2.6. Dataset

A mixture of multiple datasets was used to test. We did not pick images that were used for the original YOLO training. We utilized Kaggle's E-waste dataset, which consisted of mobile phones, microwaves, keyboards, and mice. There were 300 images of each, giving a total of 1200 images of E-waste. Another 1200 images of non–E-waste were mixed with the E-waste images. These non–E-waste images came from images.cv and roboflow and fell under various categories like handbags, chairs, spoons, books, rackets, and umbrellas. Each model was given 2400 images for testing, 1200 consisting of E-waste and 1200 consisting of non–E-waste. We then counted the number of true positives, false positives, true negatives, and false negatives.

All datasets, preprocessing scripts, and trained models are publicly available at our GitHub repository: https://github.com/vishnumuthuraman/wise (GitHub Contributors, 2025).

### 2.7. Statistical Analysis (McNemar's test)

The McNemar's test is specifically designed to evaluate differences between two classifiers tested on the same set of data points.

Given two models A and B, McNemar's test constructs a 2 x 2 contingency table based on whether each classifier prediction is correct or incorrect. The key quantities are $n_{10}$ (instances where A is correct but B is incorrect) and $n_{01}$ (instances where A is incorrect but B is correct). Under the null hypothesis that both classifiers have equal error rates, these counts should be approximately equal. The test statistic is defined as

$$\chi^2 = \frac{\left(\left|n_{01} - n_{10}\right| - 1\right)^2}{n_{01} + n_{10}} \quad (8)$$

which asymptotically follows a $\chi^2$ distribution with one degree of freedom. A small $p$-value indicates that the classifiers differ significantly in their performance.

## 3. Results

Table 2 shows the confusion matrices for each of the classical methods (YOLOv3, YOLOv4, YOLOv5, YOLOv8), tweaked methods (YRFNv3, YRFNv4, YRFNv5, YRFNv8), and ensemble methods (EYOLO, EYRFN). As one can observe, the traditional YOLOs, though impressive for object detection, do not perform well for the purposes of E-waste detection. More specifically, YOLOv3 identifies every object as a non-E-Waste, resulting in 1200 TNs and 1200 FNs. This is potentially due to overfitting during the YOLOv3 training.

While YOLOv4, YOLOv5, and YOLOv8 perform better, they still weigh FPs and FNs the same. Very high false negatives (FNs) essentially cause too many toxic substances to enter landfills and find their way into living organisms. The YRFNs each have relatively lower FNs. What is also surprising is that the cutting-edge YOLO version 8, which is the go-to algorithm today for object detection, is far from the best for E-waste detection.

Table 2 also shows the confusion matrix for the newly developed smart ensemble, WISE (Waste-focused Integrated Smart Ensemble). It has the lowest FNs amongst all methods compared. The learned weights were 0 for six of the eight methods (four YOLOs and four YRFNs). The only two non-zero weights were YRFNv4 (0.6) and YRFNv8 (0.4), with a threshold almost just above zero.

A final comparison of all methods in terms of the popular measures of success and our EMC measure is shown in Table 2. As one can see, the smart ensemble outperforms every other method significantly. Moreover, the magnitude of EMC is in USD per object and can hence easily lend itself to a relatable, intuitive value. In Table 2, the best values for each measure of success are bolded.

Complete confusion matrices and additional performance breakdowns are provided in our GitHub repository: https://github.com/vishnumuthuraman/wise (GitHub Contributors, 2025).

### 3.1. Statistical Significance Comparisons

While raw performance metrics discussed in the previous section provide useful summaries, they do not by themselves establish whether observed differences are statistically significant. To rule out this possibility, we conduct pairwise McNemar's tests across all models. (McNemar, 1947)

**Table 2:** Performance comparison of different models

| Model | TP | FP | TN | FN | Accuracy | Precision | Recall | EMC |
|---|---|---|---|---|---|---|---|---|
| YOLOv3 | 0 | **0** | 1200 | 1200 | 0.50 | N/A | 0.00 | 11.71 |
| YRFNv3 | 0 | **0** | 1200 | 1200 | 0.50 | N/A | 0.00 | 11.71 |
| YOLOv4 | 756 | 4 | 1196 | 444 | 0.81 | **0.99** | 0.63 | 4.33 |
| YRFNv4 | 851 | 11 | 1189 | 349 | 0.85 | **0.99** | 0.71 | 3.41 |
| YOLOv5 | 209 | 5 | 1195 | 991 | 0.59 | 0.98 | 0.17 | 9.67 |
| YRFNv5 | 373 | 14 | 1186 | 827 | 0.65 | 0.96 | 0.31 | 8.08 |
| YOLOv8 | 176 | 6 | 1194 | 1024 | 0.57 | 0.97 | 0.15 | 10.00 |
| YRFNv8 | 340 | 22 | 1178 | 860 | 0.63 | 0.94 | 0.28 | 8.40 |
| EYOLO | 90 | 0 | 1200 | 1110 | 0.54 | 1.00 | 0.08 | 10.83 |
| EYRFN | 203 | 0 | 1200 | 997 | 0.58 | 1.00 | 0.17 | 9.73 |
| WISE - Smart (learned) | 906 | 25 | 1175 | **294** | **0.87** | 0.97 | **0.76** | **2.88** |

Figure 6 presents a heatmap of the pairwise McNemar $p$-values. It is displayed on a $-(p)$ scale for easier readability, and significantly large values are colored yellow. As shown, almost all off-diagonal entries exhibit extremely small $p$-values, well below conventional significance thresholds (e.g., $p < 0.05$). This establishes that the observed performance gaps across methods are not attributable to chance.

Methods that have identical predictions (YOLOv3 vs. YRFNv3) appear visually as blank/white in the heat map since, in such cases, McNemar's statistic is undefined and has no discordant pairs. We also notice that EYRFN is not statistically different from YOLOv5.

These findings are reassuring that the comparative rankings from Table 2 are not due to chance. Without such an analysis, improvements cannot be trusted to have come from well-grounded performance improvements.

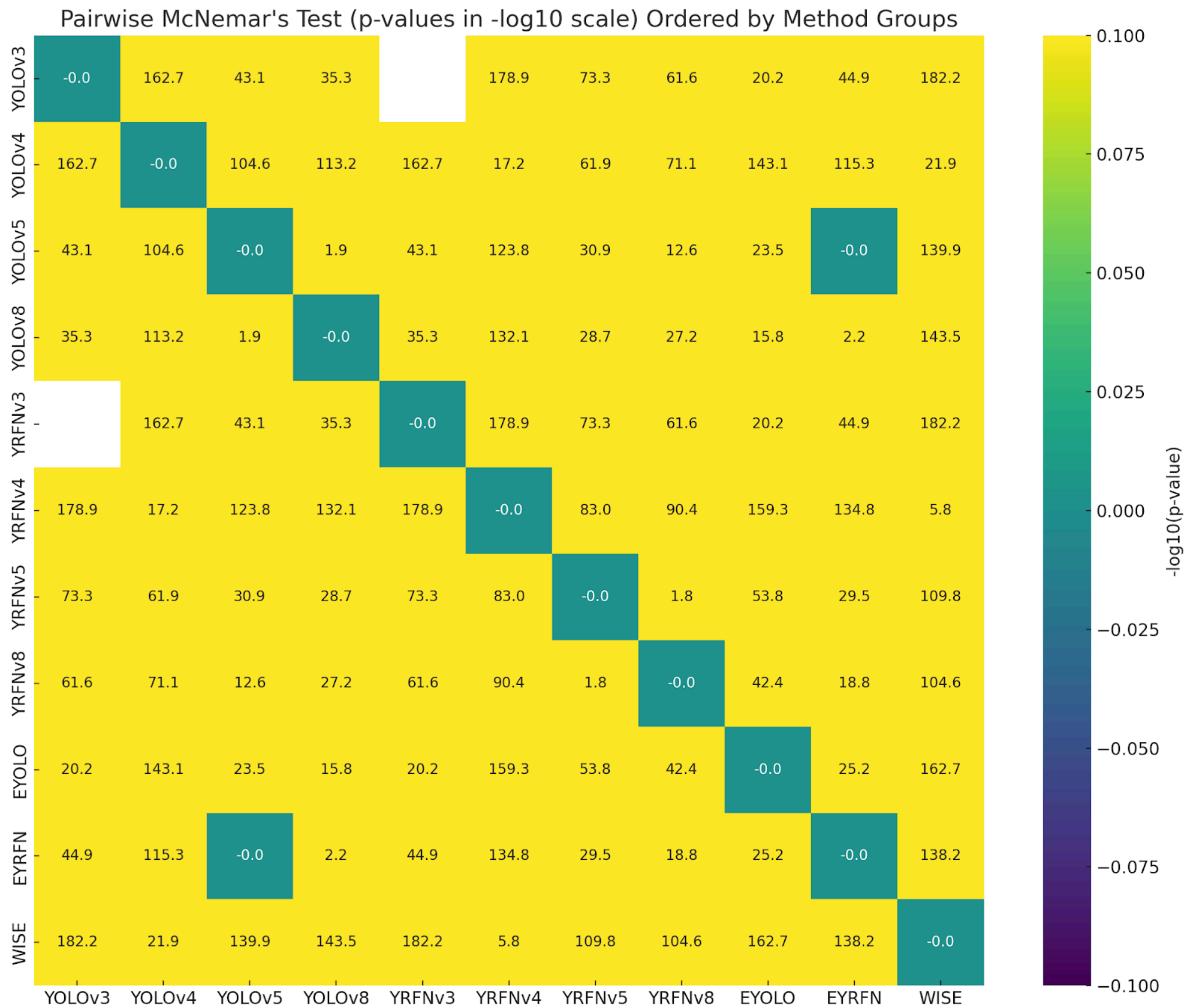


**Figure 6:** Pairwise McNemar's test results across all methods

*Note*: The heatmap displays $-\log_{10}(p)$ values, where larger magnitudes correspond to stronger evidence against the null hypothesis of equal performance.

### 3.2. Sensitivity of EMC to cost assumptions

Table 3 presents a compact sensitivity table comparing WISE to the two strongest baselines (YOLOv4 and YRFNv4) under three realistic cost-ratio scenarios ($c_{FN}$: $c_{FP}$= 10:1, 17.5:1 [current], and 30:1) and an increased manual sorting scenario (doubling $c_{FP}$). WISE remains the method with the lowest EMC; the ranking relative to YRFNv4 and YOLOv4 is unchanged.

**Table 3:** Cost sensitivities by FN:FP ratio

| Model | 10:1 | 17.5:1 (current) | 30:1 | double cFP |
|---|---|---|---|---|
| WISE | 1.66 | 2.88 | 4.94 | 2.90 |
| YRFNv4 | 1.95 | 3.41 | 5.85 | 3.42 |
| YOLOv4 | 2.48 | 4.33 | 7.44 | 4.24 |

## 4. Discussions

Our results demonstrate that aligning evaluation metrics with application-level costs provides tangible benefits for E-waste detection. The E-waste Misclassification Cost (EMC) framework allows us to explicitly trade off false negatives (FNs) and false positives (FPs) based on their real-world implications. Across all tested cost ratios and sensitivity analyses, the WISE ensemble consistently achieved the lowest EMC and the highest recall, underscoring its robustness to plausible regional or operational variations in cost assumptions. Missing E-waste items (FNs) are particularly consequential, as they increase the risk of toxic materials entering landfills and the ecosystem. EMC minimizes such high-impact errors while maintaining a manageable false-positive burden on manual sorting. This balance directly supports more sustainable E-waste operations.

From a methodological perspective, our study reinforces broader findings that cost-sensitive training and ensembling improve decision quality when misclassification costs are asymmetric. WISE's sparsity—favoring YRFNv4 and YRFNv8—suggests that leveraging a few high-performing reduced-FN models can outperform large unweighted ensembles. These results mirror classical meta-learning and ensemble-weighting theory, in which limited but well-weighted predictors dominate majority-vote systems. In practice, this means facilities can prioritize training fewer, more complementary detectors while still realizing ensemble benefits.

Many surprising observations were made in the results section. Classical cutting-edge methods like YOLOv8 (Jain et al., 2025), have terrible performance for E-waste detection, although their performance for general object detection is commendable. Further thought would reveal that this is not totally surprising, since they were trained and evaluated in a generic object detection context and with accuracy as the primary measure. Hence, it might not be fair to compare or to use such methods for E-waste detection. Next, tweaked methods outperform the classical methods whose default parameters would have been carefully chosen by the authors. This again shows that even parameters calibrated for generic object detection do not work well for E-waste detection. We also confirmed that almost all model pairs differ in a statistically robust way and, more importantly, our smart ensemble, WISE, is statistically better than all the other models.

Our results align with reports that off-the-shelf detectors can degrade under domain shift (Solovyev et al., 2021). Our results also align with broader ML findings that cost-sensitive training and ensembling can improve decision quality when costs are asymmetric (Domingos, 1999; Elkan, 2001; Wolpert, 1992). In E-waste operations, prioritizing lower FNs as opposed to FPs by an EMC-minimizing ensemble like WISE is a principled path forward. Our smart ensemble does surprisingly well in the context of E-waste detection. It not only has a significant cost reduction, it also has the lowest number of FNs in the entire comparison set. While a methodology that reduces FNs to zero while not labeling every object as FP is ideal, that would be for future research.

### 4.1. Broader Impacts and Ethics

Automating E-waste detection and sorting can yield multiple societal benefits. Reducing manual handling of hazardous materials can lower exposure risks for workers, particularly in under-resourced recycling facilities. However, automation may also shift labor demand across the recycling chain. To ensure equitable outcomes, implementation should include just-transition principles like reskilling opportunities, strengthened occupational safety standards, and fair-work policies. More broadly, the EMC framework provides a transparent way to quantify trade-offs between environmental risk (through FNs) and operational burden (through FPs). By integrating such cost-aware metrics into policy and facility design, E-waste operations can achieve both environmental and social sustainability.

### 4.2. Limitations and Future Work

Our 2,400-image corpus mixes curated E-waste and non-E-waste classes but is modest in size and lacks the nuances of in-the-wild images. There might be cluttering, stacking, off-the-frame issues, etc., on sorting lines and conveyor belts. In-situ tests and refinements are a subject of future studies.

Our current dataset, though balanced and curated, contains only four E-waste categories (mobile phones, microwaves, keyboards, and mice). Consequently, generalization to heterogeneous streams such as printed-circuit boards, cables, and monitors remains an open challenge. In real-world conveyor-belt environments, domain shift arising from clutter, occlusion, and motion blur may also affect accuracy. Nevertheless, WISE can be retrained with additional classes and cost parameters as new data becomes available. Future work should combine in-situ belt imagery with adaptive EMC weighting that accounts for item composition, contamination, and lighting variability.

Assumptions of the cost accounting for EMC use approximate unit costs from the literature and wage proxies. We do adjust the costs for inflation. However, we have one EMC measure that is calibrated to costs in the United States, which produces the largest quantity of E-waste in the world. However, we would need to refine this measure of different economies in the future.

Finally, we have not studied any real-time constraints like latency and throughput. Further improvements can help create ensembles that also consider the underlying models' latency and throughput to combine them to yield a smart ensemble that matches the expected real-time performance thresholds will be the subject of future studies.

## 5. Conclusion

We propose EMC, an application-level metric that weights e-waste detection errors by their operational/environmental costs, and WISE, a learned ensemble tuned to minimize EMC. On a 2,400-image corpus, WISE achieves the lowest EMC and highest recall, with improvements that are statistically significant by McNemar's test. A short cost-sensitivity study shows our rankings are robust to plausible $c_{FN}:c_{FP}$ ratios. The approach is practical: EMC can be re-parameterized for local conditions, and WISE can be re-trained as new classes are added. Future work will extend to in-situ belt data and integrate latency and throughput constraints.

## 6. References

Adedeji, O., & Wang, Z. (2019). Intelligent waste classification system using deep learning convolutional neural network. *Procedia Manufacturing*, 38, 48–55. https://doi.org/10.1016/j.promfg.2019.05.086

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: *Optimal speed and accuracy of object detection*. arXiv. https://arxiv.org/abs/2004.10934

Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS – Improving object detection with one line of code. *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), 5561–5569. https://openaccess.thecvf.com/content_ICCV_2017/papers/Bodla_Soft-NMS_--_Improving_ICCV_2017_paper.pdf

Bounding box regression loss. (n.d.). *CloudFactory Computer Vision Wiki*. Retrieved September 6, 2025, from https://wiki.cloudfactory.com/docs/mp-wiki/loss/bounding-box-regression-loss

Cheng, X., Cheng, Y., Gong, Z., & Chen, Y. (2021). An improved small object detection method based on YOLOv3. *Multimedia Tools and Applications*, 80(19), 29329–29344. https://doi.org/10.1007/s11042-021-10605-5

Domingos, P. (1999). MetaCost: A general method for making classifiers cost-sensitive. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD), 155–164. https://doi.org/10.1145/312129.312220

Elkan, C. (2001). The foundations of cost-sensitive learning. *Proceedings of the 17th International Joint Conference on Artificial Intelligence* (IJCAI), 973–978. https://cseweb.ucsd.edu/~elkan/rescale.pdf

Empa - Swiss Federal Laboratories for Materials Science and Technology. (2025). *E-waste guide*. Retrieved August 24, 2025, from https://www.empa.ch/web/ewaste-guide

Fan, J., Cui, L., & Fei, S. (2023). Waste detection system based on data augmentation and YOLO_EC. *Sensors*, 23(7), Article 3646. https://doi.org/10.3390/s23073646

Feng, M., Jiang, T., Wang, J., Yang, L., & Zhang, Y. (2023). A multi-scale traffic object detection algorithm for road scenes based on improved YOLOv5. *Sensors*, 23(4), Article 2045. https://doi.org/10.3390/s23042045

Fornalczyk, A., Willner, J., Francuz, K., & Cebulski, J. (2013). E-waste as a source of valuable metals. *Archives of Materials Science and Engineering*, 63(2), 87–92. http://www.amse.acmsse.h2.pl/vol63_2/6325.pdf

GitHub Contributors. (2025). WISE [Computer software]. GitHub. https://github.com/vishnumuthuraman/wise

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. *2017 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 3296–3305. https://doi.org/10.1109/CVPR.2017.351

Ibrahim, A. N., Youssif, E. S., Hassanien, A. E., Hossam, A. A., Abdo, S. S., & Saleh, A. M. D. (2023). Object detection-based automatic waste segregation using robotic arm. *International Journal of Advanced Computer Science and Applications* (IJACSA), 14(11), 599–611. https://doi.org/10.14569/IJACSA.2023.0141164

Jain, H., Jain, J., Shrivastava, V., Pandey, A., & Sharma, S. (2025). YOLOv8 and its advancements. *International Journal of Research Publication and Reviews*, 6(5), 660–666. https://ijrpr.com/uploads/V6ISSUE5/IJRPR44604.pdf

Jocher, G., & Ultralytics. (2020). *YOLOv5* [Computer software]. GitHub. https://github.com/ultralytics/yolov5

Jocher, G., & Ultralytics. (2023). *YOLOv8* [Computer software]. GitHub. https://github.com/ultralytics/ultralytics

Kán, P., & Kaufmann, H. (2019). DeepLight: Light source estimation for augmented reality using deep learning. *The Visual Computer*, 35(6–8), 873–883. https://doi.org/10.1007/s00371-019-01666-x

Khan, H. A., Naqvi, S. S., Alharbi, A. A. K., Alotaibi, S., & Alkhathami, M. (2024). Enhancing trash classification in smart cities using federated deep learning. *Scientific Reports*, 14, Article 11816. https://doi.org/10.1038/s41598-024-62003-4

Li, K., & Cao, L. (2020). A review of object detection techniques. *2020 5th International Conference on Electromechanical Control Technology and Transportation* (ICECTT), 385–390. https://doi.org/10.1109/ICECTT50890.2020.00091

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), 2999–3007. https://doi.org/10.1109/ICCV.2017.324

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer vision – ECCV 2016* (pp. 21–37). Springer. https://doi.org/10.1007/978-3-319-46448-0_2

Majchrowska, S., Mikołajczyk, A., Ferlin, M., Klawikowska, Z., Plantykow, M. A., Kwasigroch, A., & Majek, K. (2022). Deep learning-based waste detection in natural and urban environments. *Waste Management*, 138, 274–284. https://doi.org/10.1016/j.wasman.2021.12.001

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153–157. https://doi.org/10.1007/BF02295996

Meituan Vision AI Department. (2022). YOLOv6: *A single-stage object detection framework for industrial applications* [Computer software]. GitHub. https://github.com/meituan/YOLOv6

Noel-Brune, M., Goldizen, F. C., Neira, M., van den Berg, M., Lewis, N., King, M., Suk, W. A., Carpenter, D. O., Arnold, R. G., & Sly, P. D. (2013). Health effects of exposure to e-waste. *The Lancet Global Health*, 1(2), e70. https://doi.org/10.1016/S2214-109X(13)70020-2

Oza, P., Agrawal, S., Kapadia, M., & Raotole, O. (2025). Solid waste classification using deep neural network: A transfer learning approach. *Earth Science Informatics*, 18, Article 255. https://doi.org/10.1007/s12145-025-01743-x

Parti, A. (2024, March 18). *Understanding object detection: A comprehensive guide*. Pareto AI. https://pareto.ai/blog/object-detection

Paudel, P., Shrestha, S., Shrestha, S., Gurung, S., & Adhikari, S. (2024). Automated waste sorting with delta arm and YOLOv8 detection. *Journal of Artificial Intelligence and Capsule Networks*, 6(3), 299–315. https://doi.org/10.36548/jaicn.2024.3.004

Potter, R. (2022, August 23). *How bounding box object detection is taking over and what to do about it*. Medium. https://medium.com/anolytics/how-bounding-box-object-detection-is-taking-over-and-what-to-do-about-it-9130b6d6c1f1

Proença, P. F., & Simões, P. (2020). TACO: *Trash annotations in context for litter detection*. arXiv. https://arxiv.org/abs/2003.06975

Proença, P. F., & Simões, P. (2023). TACO *dataset website*. Retrieved September 5, 2025, from http://tacodataset.org

Promboonruang, S., Boonrod, T., & Ratchatawetchakul, Y. (2024). Efficient waste detection and classification based on YOLOv5 models. *Engineering Access* (MIJET), 10(1), 51–58. https://doi.org/10.14456/mijet.2024.7

Rahim, L. A. A., Abidin, N. A. Z., Aminuddin, R., Samah, K. A. F. A., Ibrahim, A. Z. M., Yusoh, S. D., & Nasir, S. D. N. M. (2024). YOLOv5 model-based real-time recyclable waste detection and classification system. In B. Alareeni & I. Elgedawy (Eds.), *Innovations in smart cities applications volume* 7 (Vol. 906, pp. 44–54). Springer. https://doi.org/10.1007/978-3-031-53824-7_5

Rajeev, P. A., Dharewa, V., Lakshmi, D., Vishnuvarthanan, G., Giri, J., Sathish, T., & Alrashoud, M. (2025). Advancing e-waste classification with customizable YOLO-based deep learning models. *Scientific Reports*, 15, Article 18151. https://doi.org/10.1038/s41598-025-94772-x

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2018). YOLOv3: *An incremental improvement*. arXiv. https://arxiv.org/abs/1804.02767

Sheng, V. S., & Ling, C. X. (2006). Thresholding for making classifiers cost-sensitive. *Proceedings of the AAAI Conference on Artificial Intelligence*, 20(1), 476–481. https://cdn.aaai.org/AAAI/2006/AAAI06-076.pdf

Singh, Y., & Parimala, S. (2025, March 26). *E-waste management* [Preprint]. SSRN. https://doi.org/10.2139/ssrn.5196804

Solawetz, J. (2024, January 4). *What is YOLOv4? A detailed breakdown*. Roboflow. https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/

Solovyev, R., Wang, W., & Gabruseva, T. (2021). Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107, Article 104117. https://doi.org/10.1016/j.imavis.2021.104117

Song, L., Yu, H., Yang, Y., Tong, Y., Ren, S., & Ye, C. (2025). YOLO-VG: An efficient real-time recyclable waste detection network. *Journal of Real-Time Image Processing*, 22, Article 74. https://doi.org/10.1007/s11554-025-01655-5

Thung, G., & Yang, M. (2017). *TrashNet* [Data set]. GitHub. Retrieved September 5, 2025, from https://github.com/garythung/trashnet

Vats, M. C., & Singh, S. K. (2014). E-waste characteristic and its disposal. *International Journal of Ecological Science and Environmental Engineering*, 1(2), 49–61.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). YOLOv7: *Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. arXiv. https://arxiv.org/abs/2207.02696

Wikipedia contributors. (2024, August 24). *Electronic waste in Guiyu*. In *Wikipedia*. https://en.wikipedia.org/wiki/Electronic_waste_in_Guiyu

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Wu, P., Chai, B., Nie, X., Yan, L., Wang, Z., Zhou, Q., Wang, B., Peng, Y., & Li, H. (2024). *Enhanced object detection: A study on vast vocabulary object detection track for* V3Det *challenge* 2024 (Version 3). arXiv. https://doi.org/10.48550/arXiv.2406.09201

Wu, T., & Zhu, S.-C. (2013). Learning near-optimal cost-sensitive decision policy for object detection. *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), 753–760. https://doi.org/10.1109/ICCV.2013.98

Yang, W.-D., Sun, Q., & Ng, H.-G. (2021). Cost-benefit analysis of metal recovery from e-waste: Implications for international policy. *Waste Management*, 123, 42–47. https://doi.org/10.1016/j.wasman.2021.01.023

## Code & Data Availability Statement

All codes and data used for this study can be found at https://github.com/vishnumuthuraman/wise.

## Author Biography

**Vishnu Muthuraman** is a senior at Westwood High School in Austin, Texas, with a strong interest in computer science, machine learning, and sustainable technology innovation. He is the founder of Gdgt+, a mobile application that helps users manage their electronic devices to promote environmentally responsible consumption and disposal, and of The Loneliness Project, a nonprofit initiative connecting young musicians with senior living communities to ease isolation through music. He is also a USA Computing Olympiad (Gold) participant, an award-winning app developer, and an ARSM Diploma pianist. His work sits at the intersection of artificial intelligence, sustainability, and human-centered design, with a vision to build technology that benefits both people and the planet.

## Mentor Contribution Statement

**Prof. Siddharth Krishnan** and **Dr. Plínio Zanini** provided valuable guidance during the development of this manuscript. Their mentorship was instrumental in refining the overall structure, strengthening the clarity of exposition, and ensuring that the research communicated its technical ideas effectively. During the early stages, they offered constructive feedback on how to frame the research problem and contextualize the study within broader work on adaptive computer vision and environmental sustainability. Their comments helped improve the logical flow between sections and clarify the motivation behind key design choices.