A Two-Stage Deep Learning Framework for Enhanced Waste Detection and Classification

Hongbo Pang
Department of Systems and Computer Engineering
Carleton University
Ottawa Canada
hongbopang@cmail.carleton.ca

Changcheng Huang
Department of Systems and Computer Engineering
Carleton University
Ottawa Canada
huang@sce.carleton.ca

***Abstract*— With the rapid modernization progress over the past decades, waste classification has become increasingly important as cities worldwide seek to implement more sustainable waste management practices. Traditional manual sorting methods are labor-intensive, prone to inaccuracies, and hard to scale, driving the need for automated, efficient solutions. Although deep learning techniques, recognized for their ability to process complex hierarchical data, have emerged as potential aids, their successful implementation is often hampered by the challenging task of gathering diverse, large-scale, high-quality waste image datasets, leading to possible overfitting and model bias. This study proposes an innovative, two-stage waste detection framework that first identifies the bounding box of waste items, then classifies them into one of six primary categories, effectively addressing the inherent issues with previous methodologies by optimally utilizing available data and reducing overfitting and bias. Trained and evaluated on the comprehensive TACO and WaRP waste datasets, our model has demonstrated superior performance relative to existing methods, underscoring its promise as a scalable, accurate, and efficient solution for waste classification, thus offering exciting prospects for further research and practical applications in sustainable waste management.**

Keywords—Deep learning, garbage classification, waste management

I. INTRODUCTION

In the past decade, garbage classification, which involves sorting waste into categories based on properties such as biodegradability, recyclability, and toxicity, has become a pivotal aspect of sustainable development and environmental protection. As global population growth and consumerism contribute to increased waste production, garbage classification helps recover valuable resources, reduces landfill waste, and minimizes environmental impact. The integration of artificial intelligence (AI) and machine learning (ML) in garbage classification, leveraging cameras and sensors to detect and classify waste based on characteristics like shape, colour, and texture, has shown potential in enhancing sorting accuracy and efficiency, reducing human error, and improving recyclable material quality.

However, several challenges persist, including the need for a standardized waste classification and detection dataset format, the arduous task of compiling large, diverse, high-quality waste image or video datasets, and the complexities associated with unique dataset characteristics and waste item variability. Furthermore, object detection and classification, two separate deep learning tasks requiring different annotations and data structures, are often tackled in isolation, providing incomplete information and limiting real-world applicability. To address these challenges, we propose a unified framework that concurrently integrates waste detection and classification tasks, aiming to provide a more comprehensive and efficient real-world applicable waste classification solution.

II. RELATED WORK

The waste classification and detection problems are a subset of object classification/detection tasks in computer vision society. Garbage classification aims to accurately identify and categorize different types of waste materials, such as plastic, glass, metal, and paper, to facilitate proper disposal and recycling. The major challenge in garbage classification is the high variability and complexity of the waste materials themselves. Garbage items can have different shapes, colours, textures, and sizes. In addition, they can be contaminated or mixed with other materials, making it difficult for traditional rule-based or heuristic approaches to classify them accurately. As mentioned earlier, the lack of training data is one of the limitations of deep learning-based waste classification and detection, as annotating garbage images with accurate class labels and object bounding boxes can be a time-consuming and expensive and can also introduce bias and errors in the dataset. However, many scholars have continuously developed more accurate, efficient, and scalable systems and pipelines for waste detection and classification.

A. Benchmark Datasets

As garbage-based classification and detection gains more and more attention in the computer vision society, the scientific community has created and published numbers of dataset benchmarks. Datasets play a crucial role in the development of deep learning models. The dataset's quality and quantity can significantly impact a deep-learning model's accuracy and robustness. Several challenges and limitations are associated with the current datasets for garbage classification. Many existing datasets [1][4][8] only focus on a narrow range of materials, such as plastic, paper, metal, and glass, while neglecting other types of waste, such as e-waste, organic waste, and hazardous waste.

Moreover, many of the existing garbage classification datasets are relatively small, which may need to be more for training deep learning models with high accuracy when testing

1946-0759/23/$31.00 ©2023 IEEE
DOI 10.1109/ICMLA58977.2023.00304

in real-life scenarios. Standardization is also a significant issue for garbage-based classification benchmarks. Currently, no standard format or protocol for garbage classification datasets makes comparing results across different studies or applications difficult. Additionally, some datasets [2][10] may use different labelling schemes or definitions of classes, which can introduce inconsistencies and confusion.

TrashNet [1] includes six classes of garbage: waste, glass, paper, cardboard, plastic and metal. Each category contains approximately 400 images. The images were captured using a mobile phone camera in a controlled environment, with consistent lighting and background. The authors of [1] used the TrashNet dataset to train and evaluate several deep-learning models for garbage classification, including AlexNet, GoogleNet, and VGG16. They reported that the VGG16 model achieved the highest accuracy on the TrashNet dataset, with an overall accuracy of 87.2%.

TACO (Trash Annotations in Context Object Detection Dataset) [2] dataset contains more than 150,000 annotated object instances in over 6,000 images, covering 60 object classes, including various types of trash and recycling materials. It differs from other benchmarks with its annotations for both object detection and instance segmentation. In addition, TACO contains a wide range of litter types and a sizable diversity of backgrounds, from tropical beaches to London streets. The diversity of the data improves the robustness of the deep learning model in a real-life scenario. For example, the authors in [3] developed a trash detection system based on U-Net and trained and evaluated the model using the TACO dataset. They reported that their system achieved an accuracy of 94.67% on the TACO dataset, demonstrating its potential for real-world waste sorting applications.

Trash-ICRA19 [4] was signed for trash-based object detection tasks underwater. The data was collected using autonomous underwater vehicles (AUVs) in open-water locations. The dataset contains 7668 images with seven classes and corresponding labels and annotations. The author evaluated the performance of various deep learning object detection models such as YOLOv2, Tiny-YOLO, Faster RCNN and SSD. Faster RCNN outperforms other models with an mAP of 81%.

MJU-Waste [5] was created for object segmentation task, and it is the largest public benchmark available for waste object segmentation, with 1485 images for training, 248 for validation and 742 for testing. The data was collected via camera with university campus waste items held by a human in a lab environment. The authors experimented with VGG16, ResNet-50 and ResNet-101 backbones in well-known frameworks such as FCN, PSPNet, CCNet, and DeepLabv3. The result shows that the Mean pixel precision on MJU-Waste is 97.14% on the ResNet-101 backbone.

UAVVaste [6] contains 722 images collected by UAV with 3761 hand-labelled annotations of rubbish in urban and natural environments such as streets, parks, and lawns. In its paper, the authors proposed that the YOLOv4 model has the best performance in terms of accuracy and speed, with an M1 score of 78.5%.

GINI [7] dataset was collected using Bing search API. It contains 2561 images with 1496 annotations. The author proposed a new smartphone app integrated with the deep convolutional network to detect images' garbage. It achieved 87.9% mean accuracy while maintaining efficient memory usage and prediction speed.

Waste Classification Data [8] from Kaggle is one of the most popular benchmarks for the waste classification task. It contains 22,500 with two classes: organic and recyclable. Unfortunately, the data was scrapped from Google search.

WaRP (Waste recycling plants) [9] dataset consists 28 recyclable waste categories, which are divided into 17 categories of plastic bottles, three categories of glass bottles, two categories of cardboard, and four categories of detergents and cans. The dataset is captured from the waste processing belt and can be used to train detection, classification, and segmentation models.

B. Two-step Waste Detction using Deel Learning

Waste detection tasks focus on the localization and the classification of the detected wasted object in images and video frames. Differing to waste classification, waste detection can be applied to a wider range of applications in various scenarios, such as underwater conditions. One of the major challenges in waste detection is data collection. Image annotation is a time-consuming and labour-expensive task, making it difficult to build a comprehensive benchmark for all scenarios.

In 2017, the authors of [10] proposed a robotic grasping system for automatically sorting garbage using Fast R-CNN with Regional Proposal Generation (RPN) for object detection and the VGG-16 for object reignition and post-estimation. The detection of the self-made dataset results in a 3% missing rate and a 9% false rate.

Thung et.al. [11] proposed a two-step approach to the waste detection problem. The authors use Faster R-CNN for object detection with separate CNNs for object classification. The model is trained and evaluated using the Labeled Waste in the Wild dataset of 1002 images of used food trays. It investigates different architectures, including flat, material, and shaped-based methods. The flat method refers to using a single CNN for all types of waste. Material-based and shape-based methods use different CNNs for waste in different materials and shapes. The result shows that Faster R-CNN with flat CNN achieved the best (mAP 74.1%) among the methods.

A novel two-stage waste detector for e-waste is proposed in [12]. The authors proposed to use Faster R-CNN as the object detector and a deep CNN as the classifier. The model is trained and tested on the image of electrical applications such as refrigerators and washing machines. It yields recognition and classification accuracy of the selected e-waste categories ranging from 90% to 97%.

In [13], the authors proposed a two-stage deep-sea debris detection method using YOLOv3 and ResNet50. Compared with other deep-sea debris detection methods, the proposed method uses a 3-D dataset containing seven types of deep-sea debris with depth information. The ResNet50-YOLOv3 achieves the best comprehensive detection capabilities (mAP@.5 83.4%) for deep-sea debris while maintaining a low

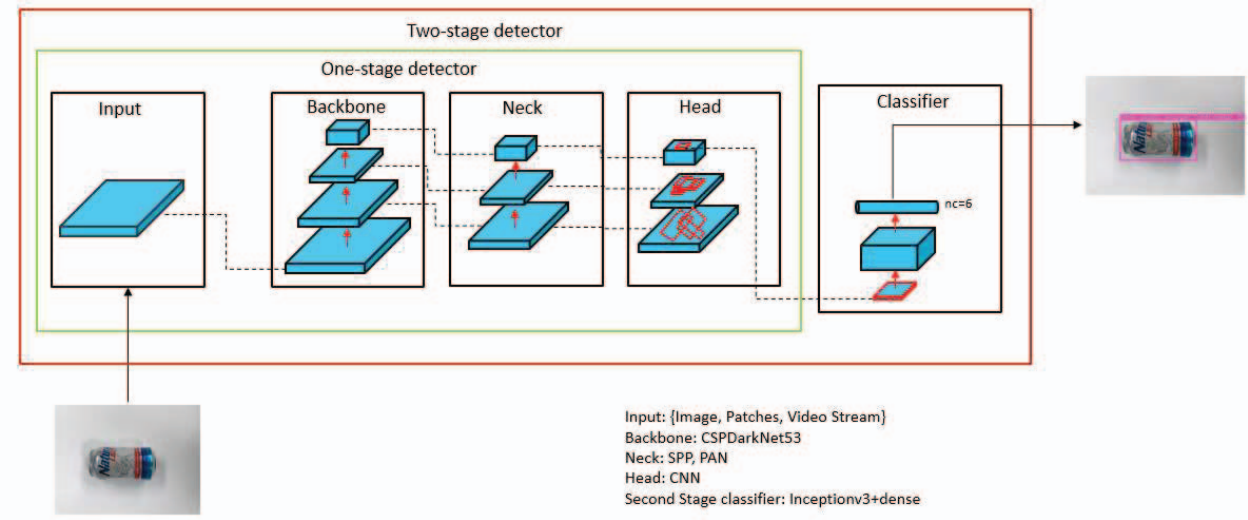

Figure 1 The proposed framework

level of confusion compared with the other two-stage architectures.

The author in [14] investigates waste detection on narrowband Internet of Things (IoT) devices. The proposed method replaced the backbone of the YOLOv2 model from VGG16 to MobileNet to perform lightweight predictions. The improved YOLOv2 model obtained the same level of detection precision with Fast R-CNN (89.1% vs 89.7%) with a much faster inference speed (5 f/s vs 42 f/s).

[15] proposed a two-step waste detector in natural and urban environments. The model uses EfficientDet-D2 to localize litter and EfficientNet-B2 to classify the waste into seven categories. The classifier is trained in a semi-supervised fashion using unlabeled data. The proposed model starts with the region proposal by the object detector. Then the cropped region is passed to the classifier. The output contains the possibility of seven litter categories and additional background classes. The EfficientDet-D2 obtained mAP50 above 90% for the TrashCan dataset and mAP50 of 16.2% for the TACO dataset. The EfficientNet-B2 obtained the best precision of 97% on the background class.

C. Summary

Waste sorting problems can be divided into two aspects: waste classification and waste detection. The training data is labelled according to distinct categories in waste classification, predominantly emphasizing the waste object. On the other hand, waste detection involves annotating the input data with class labels and the coordinates of one or several waste objects. Moreover, recent studies show that two-step waste detection is more robust to shape, form, and background varieties. However, using a single traditional two-stage detector model such as Fast R-CNN suffers in slower inference speed, more complex training pipeline, and low detection rate for small objects. Numerous research efforts have integrated one-stage detectors, such as YOLO and EfficientDet, with an independent CNN-architecture classifier to address this issue. This approach enhances both the classification accuracy and the model's robustness, fulfilling the prerequisites for deployment in real-world applications.

III. PROPOSED DETECTION FRAMEWORK

The suggested framework is composed of two modules, as illustrated in Fig. 1. The initial component employs a one-stage detector based on the YOLOv5 architecture. The model's input can consist of individual images, patches of images, or video streams. Subsequently, the input undergoes the CSPDarkNet backbone network for feature extraction. The neck layer utilizes spatial pyramid pooling and a Path Aggregation Network. Following this, the head layers of the model are tasked with region proposals. The proposed region is then cropped and fed into a separate classifier employing a pre-trained InceptionV3 model. The output of the model contains the bounding box of the detected litter and the possibility score that indicates the litters' categories (cardboard, paper, plastic, metal, glass, and trash).

A. Object Detection Module

The object detection module of the proposed framework consists of three layers: Backbone, Neck, and Head.

The **Backbone** uses the CSPDarkNet53 as the base network for object detection. It is a modified version of the DarkNet53 architecture that integrates Cross Stage Hierarchical (CSH) features to enhance efficiency and performance. The network can be divided into the convolutional build block and five CSPBlock modules. The convolutional building block with a kernel size of three and a stride of one and concatenated with the Mish layer. Mish is an activation function that can be mathematically defined as

$$f(x) = x\tanh(softplus(x)) \qquad (1)$$

where x is the input and *softplus(x)* is a smooth approximation of the *ReLu* function. The Mish layer helps prevent the vanishing gradient problem by maintaining a smooth and continuous gradient even for large input values. The CSPBlocks divide feature maps into two parts and then merge

them through a cross-phase hierarchy. This way, the gradient flow can propagate through different network paths after being separated [16]. The cross-phase hierarchy module enables better gradient flow and reduces redundancy in feature maps, leading to more efficient training and faster inference times.

The **Neck** layer is used to extract the feature pyramid from the Backbone. Firstly, a variant of Spatial Pyramid pooling has been used to handle input images of varying sizes and generate fix-size feature representation. It divides the input feature map into non-overlapping regions at multiple scales or levels, forming a spatial pyramid. By capturing spatial information at different scales and levels, the performance and generalization capabilities of CNNs is improved. Then, a Path Aggregation Network is used as a bottom-up path augmentation. It adds connections from lower-level feature maps to higher-level feature maps in the bottom-up pathway.

The **Head** consists of three convolution layers that predict the location of the bounding boxes in the form *of (x, y, height, width)*, where *(x, y)* are the normalized center coordinates. The bounding box calculation is shown in Fig. 2, where *Cx* and *Cy* are the top left coordinates; *tx* and *ty* are the output from the layer; *pw* and *ph* is the size of the proposed anchor box; *e* represents the spatial transformation of the dimension.

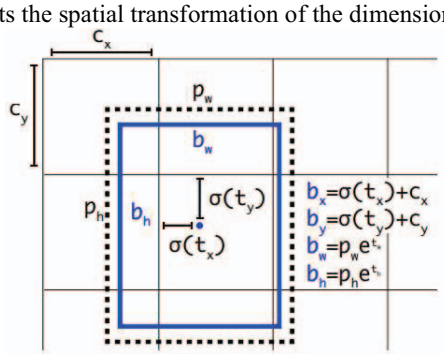

Figure 2 Boudning box prediction in YOLO [46]

The leaky ReLU and Sigmoid functions are used as the activation functions in the YOLOv5 architecture. The Leaky ReLU function is defined as:

$$LeakyReLU(x) = \max(\alpha x, x) \quad (2)$$

where where x is the input to the activation function, and α is a small positive constant, typically in the range of 0.01 to 0.3. The Leaky ReLU function aims to address the "dying ReLU" problem associated with the standard ReLU function. The dying ReLU problem occurs when a neuron gets stuck in the negative part of the ReLU function, causing it to output zero and stop learning due to the lack of gradient during backpropagation. [17].

For the Loss function, the model uses BCE (Binary Cross Entropy) to compute the class loss and the object score and CIoU (Complete Intersection over Union) loss to compute the localization loss. The BCE loss function quantifies the dissimilarity between the predicted probability distribution and the true probability distribution of the target class. It can be calculated as follows:

$$BCE(y,p) = -(1/N)\,\Sigma\,[y_i * log(p_i) + (1 - y_i) * log(1 - p_i)] \quad (3)$$

Where y represents the true labels, and p represents the predicted probabilities. The summation is performed over all samples in the dataset.

As an advanced version of the traditional IoT metric, the CIoU loss combines IoU with the center distance term and aspect ratio term, making it more sensitive to the quality of bounding box predictions. It can be defined as the following:

$$CIoU = 1 - IoU + (\rho^2(center\_pred, center\_gt)/c^2) + \alpha * v \quad (4)$$

Where IoU stands for Intersection over Union, which is the ratio of the intersection area of the predicted and ground truth bounding boxes to their union area; ρ is the Euclidean distance between the center points of the predicted and ground truth bounding boxes; c is the diagonal length of the smallest enclosing bounding box containing both the predicted and ground truth bounding boxes; α is a trade-off parameter that balances the aspect ratio term (v); v is the aspect ratio term, which measures the difference in aspect ratios between the predicted and ground truth bounding boxes. By minimizing the CIoU loss, the model is encouraged to generate bounding boxes that have a high IoU and better align with the ground truth boxes in terms of their center positions and aspect ratios.

B. Object Classifiction Module

After the region proposal in the object detection module, the region is then cropped and fed into the object classification module for further classification. Due to the poor prediction accuracy of the one-stage detector on un-seen objects, a second-stage classifier (a modified InceptionV3 model) is proposed.

Inceptionv3 [19] is a convolutional neural network architecture from the Inception family with several improvements. First, inception-v3 refines the use of auxiliary classifiers, which are additional classifiers connected to intermediate layers of the network. These classifiers help improve the gradient flow and alleviate the vanishing gradient problem, making it easier to train deeper networks. Moreover, to prevent the model from becoming overconfident in its predictions, Inception-v3 uses label smoothing, which assigns a small portion of the probability mass to incorrect labels during training. This technique regularizes the model and prevents overfitting.

Transfer learning is applied to train the fully connected layer at the top. The original Inceptionv3 outputs the shape of [None, 2048], and the modified model outputs the shape of [None, 6] to match the six categories of waste (cardboard, paper, plastic, metal, glass, and trash) by adding a dense layer with average pooling.

C. Data Augmentation and Preprocessing

To increase the robustness and generalization capabilities of our proposed framework, we will apply data augmentation techniques such as random rotations, color space adjustments,

Mosaic augmentation, and scaling to the training dataset. The main idea behind mosaic augmentation is to combine four different images into a single composite image as shown in Fig. 3. It preserves the annotations and labels of the original images and exposes the model to multiple objects and scenes in a single image, encouraging the model to learn how to handle complex scenes and object interactions. Moreover, image preprocessing techniques, such as normalization and resizing, will be used to ensure consistent input for the deep learning models.

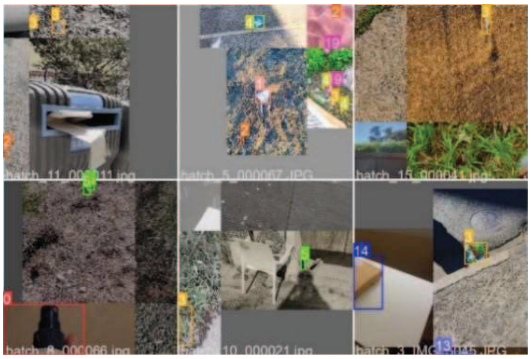

Figure 3 The Mosaic augmentation

IV. EXPRIEMENT AND RESULTS

A. Model Evaluation and Performance Metrics

The standard metrics Precision, Recall, mAP@.5 and mAP@.5:.95 are used to evaluate the object detection module and accuracy to evaluate the object classification module.

Precision measures the model's ability to correctly identify positive instances among all the instances it has predicted as positive. In other words, precision tells us how accurate the model is in its positive predictions. It can be defined as using following equation:

$$Precision = \frac{TP}{(TP + FP)} \tag{5}$$

Where True Positives (TP) refers to the number of instances where the model correctly predicts the positive class. False Positives (FP) means the number of instances where the model incorrectly predicts the positive class when the actual class is negative.

Recall measures the model's ability to correctly identify all the dataset's positive instances (i.e., instances belonging to the target class). It can be defined as the following:

$$Recall = \frac{TP}{(TP + FN)} \tag{6}$$

Where False Negatives (FN) is the number of instances where the model incorrectly predicts the negative class when the actual class is positive.

mAP@.5 (Mean Average Precision at a 0.5 Intersection over Union) is the mean average precision of IoU with a threshold of 0.5. In the context of object detection, the Intersection over Union (IoU) is a metric that measures the overlap between two bounding boxes:

$$IoU = \frac{(Area\ of\ Intersection)}{(Area\ of\ Union)} \tag{7}$$

mAP@.5 requires an IoU threshold of 0.5, which means that a predicted bounding box is considered a true positive if its IoU with the ground truth bounding box is greater than or equal to 0.5 (i.e., at least 50% overlap). The average precision is calculated using the area under the Precision-Recall curve for each object using the following equation:

$$AP = \int_0^1 p(r)dr \tag{8}$$

Where p(r) is the Precision-Recall curve by plotting the calculated precision values against the corresponding recall values, finally, Mean Average Precision (mAP) can be calculated by averaging the AP values across all object classes.

mAP@.5:.95 calculates the Mean Average Precision (mAP) at different IoU thresholds, starting from 0.5 (50% overlap) to 0.95 (95% overlap), with an interval of 0.05. Compared with mAP@.5, mAP@.5:.95 provides a more balanced and comprehensive evaluation of the model's performance.

For the classification task, accuracy measures how well the model correctly predicts the class labels for a given dataset. Accuracy is defined as the ratio of the total number of correct predictions to the total number of instances in the dataset:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{9}$$

Where False Positives (FP) is the number of instances where the model incorrectly predicts the positive class when the actual class is negative, and False Negatives (FN) is the number of instances where the model incorrectly predicts the negative class when the actual class is positive.

B. Dataset Preparation

The datasets used for training and evaluating of the proposed model are the Waste Classification dataset [25], the TACO dataset [19], and the WaRP[26] dataset. Inceptionv3 is trained in the Waste Classification dataset, which contains six categories of waste. The YOLOv5 is trained in selected categories on the TACO dataset. 22 categories that belong to six super categories are selected from the original TACO dataset. The selected categories can be found in Table 1. Both models use the pre-trained weight as the starting point, where the initial Inceptionv3 weight was trained on the ImageNet dataset, and the initial YOLOv5 weight was trained on the COCO dataset. Afterwards, the WaRP dataset is used to evaluate the performance of the integrated model in a real-life scenario.

C. Model Implementation and Training

Object Detector YOLOv5 is implemented using Pytorch framework and trained with one NVIDIA A100-SXM4-40GB GPU from Google Colab. The model contains 468 layers with 46,251,379 parameters. It took 1.7 hours to train 100 epochs with a batch size of 16 and image size of 416*416.

Object Classifier Inceptionv3 is implemented using the TensorFlow framework in Python and trained on the Waste Classification dataset with an input size of 224 x 224, batch size

Table 1 Selected Categories from TACO dataset.

Super-category	ID	Category	# of instance
plastic	4	Other plastic bottle	50
	5	Clear plastic bottle	285
	29	Other plastic	273
	36	Plastic film	451
	39	Other plastic wrapper	260
glass	6	Glass bottle	104
	23	Glass cup	6
	26	Glass jar	6
metal	12	Drink can	229
	52	Scrap metal	20
	8	Metal bottle cap	80
	28	Metal lid	10
carboard	14	Other carton	93
	16	Drink carton	45
	17	Corrugated carton	64
	18	Meal carton	30
paper	31	Tissues	42
	32	Wrapping paper	12
	33	Normal paper	82
	34	Paper bag	27
Trash	25	Food waste	8
	1	Battery	2

Figure 4 Examples from selected datasets

of 64 and epochs of 100. The dataset is split for training and validation in a 0.8, 0.2 ratio.

D. Quantitative Results

Table 2 provides a comparison of various deep learning architectures for object detection using the Waste Classification dataset. The InceptionV3 module with transfer learning achieved superior validation accuracy, balancing a medium level of model complexity. Different YOLOv5 architecture variants, specifically YOLOv5s (small) and YOLOv5l (large), with varying sizes and computational complexities, were evaluated for the object detection module. Despite similar design, these variants differ in layer quantity and feature extractor complexity. YOLOv5s, with fewer layers and a simpler architecture, is lightweight, making it optimal for use on devices with limited computational resources, like smartphones or edge devices. Conversely, YOLOv5l, with a more intricate, deeper architecture and more layers, offers increased model capacity and superior feature extraction capabilities.

Table 2 Comparison of classifier

Base Model	Train Accuracy	Validation Accuracy	Total Params
EfficientNet B0	95.81%	86.69%	4,057,250
Resnet50	99.85%	95.70%	23,600,006
VGG16	99.10%	84.45%	14,980,422
InceptionV3	99.80	97.07%	21,815,078

In terms of performance, YOLOv5l generally achieves higher accuracy and better object detection results than YOLOv5s due to its increased model capacity and more complex feature extraction capabilities. However, YOLOv5s provides faster inference speed and is more suitable for real-time applications where computational resources and latency are essential considerations.

The performances of the above models on the TACO dataset are listed in Table 3. Compared to YOLOv5s, YOLOv5l achieved better precision and lower recall. It means that the model is good at correctly predicting positive instances when it makes a positive prediction. However, it may fail to identify many of the actual positive instances in the dataset. In other words, the model tends to be conservative in making positive predictions, resulting in fewer false positives and more false negatives. In the content of waste detection, it is important to have higher precision rather than higher recall. Minimizing false positive is crucial because misclassification of waste type can result in increased workload in further processing, affecting the recycling rate. Poor performance in mAP@.5 and mAP@.5:.95 is mainly because of the imbalanced class distribution and insufficient training data. It can be solved by adding more training data into classes with fewer instances, such as Glass Cup and Glass Jar.

Table 4 shows the performance of the YOLOv5l on the WaRP dataset. The validation loss is shown in Fig, 4. The model achieved 63.6% precision and 48.6% recall. This means the model is reasonably accurate in distinguishing waste types when making a positive prediction. However, there is still room for improvement, as 37% of the time, the model needs to be more accurate to classify a waste object, potentially leading to incorrect waste sorting or processing. Higher precision might be more desirable to ensure that waste is sorted accurately and sent to the correct processing facilities.

Table 3 Comparison of YOLOv5s, YOLO5l

Base Model	P	R	mAP@.5	mAP@.5:.95
YOLOv5s	0.253	0.208	0.205	0.148
YOLOv5l	0.484	0.163	0.162	0.125

Figure 7 Complex case results

Table 4 Performance of YOLOv5l on WaRP dataset

Base Model	P	R	mAP@.5	mAP@.5:.95
YOLOv5l	0.636	0.486	0.523	0.402

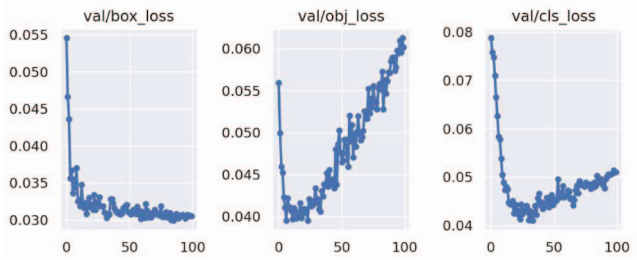

Figure 4 YOLOv5l Validation Loss on WaRP dataset

E. Qualitative Results

We selected three scenarios to evaluate the integrated model's performance. The simple scenario only includes one clear waste object with a clear background. The intermediate scenario includes smaller waste objects with a more complex background. Finally, the complex scenario includes multiple waste objects with the most complex background.

Fig 5 presents examples of simple case results where the integrated model successfully detected objects but struggled with accurate initial classification, primarily due to limited training instances for the object detection model. However, the second-stage classifier enhanced classification accuracy by reassessing the proposed region, as exemplified by a Fanta glass bottle initially misclassified as a "clear plastic bottle" but correctly identified as "glass" in the second stage.

As shown in Fig. 6, the intermediate scenario contains small waste objects in a more complex background. The performance of both the detection and classification modules is affected by the size of the input image and the resolution of the proposed region. Especially for the object classification module, the cropped object box is extremely small and not recognizable even with human eyes. In this case, the second stage classifier will have lower accuracy due to the high false positive rate. This issue can be addressed by adding more training images with various sizes to the classification dataset and using higher-resolution test images.

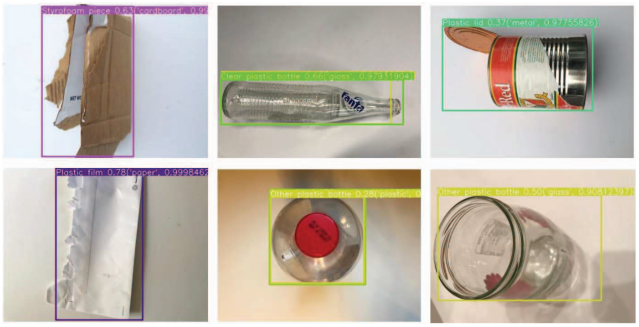

Figure 5 Simple case results

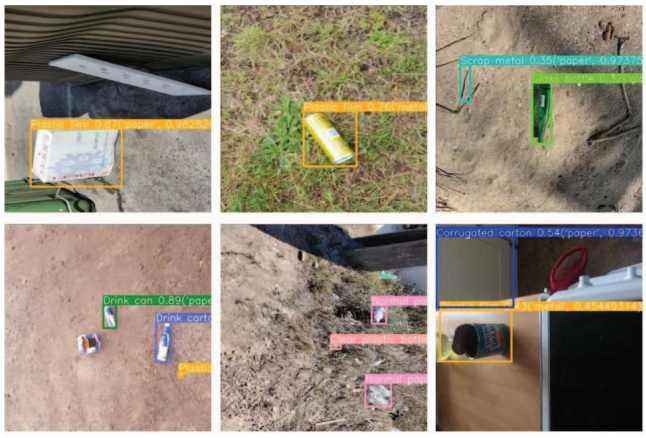

Figure 6 Intermediate scenario results

In the most complex scenario where numbers of waste objects with different shapes are stacked, as shown in Fig.7, the model performs better when the waste object is distinct from its background. For example, paper bags and plastic film are the most detected categories because the color and the light reflection made them distinct from the background. To address this, including more training images with different light conditions and shapes could be a good approach.

V. Conclusion and Future Work

Automated waste detection and classification are vital in achieving sustainable recycling practices. Utilizing deep learning methodologies for waste detection and classification allows waste management systems to enhance efficiency, precision, and adaptability across various scenarios and requirements. The study introduces a two-step waste detection system crucial for successful recycling practices. Using deep learning techniques, the proposed system improves the efficiency and adaptability of waste management across varying conditions. Our unique approach uses two separate models for detecting and classifying waste, making it stronger for real-life use. The system utilizes the YOLOv5 design for waste detection and applies transfer learning to the Inceptionv3 model for classification which further improves the classification accuracy of detected waste. As a result, the system performs well on the WaRP dataset (63.6% precision and 52.3% mAP@.5), even with a small training dataset of 819 images. Moreover, our framework can adjust to changes in waste categories, local regulations, or specific industry needs by retraining the model with new data or fine-tuning the architecture. The combination of YOLOv5 and Inceptionv3 supports scalability in terms of the number of waste categories that can be detected and classified. As more data becomes available, the models can be easily updated to perform better and manage a wider range of waste materials.

Several improvements can still be made to the performance of the proposed framework. Acquiring more labelled training data, particularly for imbalanced waste categories or challenging cases, is essential for deep learning models. A larger, more diverse training dataset can aid the model in learning a more extensive range of features, thus improving overall performance. Experimentation with alternative model architectures or adopting more recent state-of-the-art models may also yield better results.

References.

[1] G. Thung and M. Yang, "Classification of Trash for Recyclability Status," in Proceedings of the 2018 IEEE Conference on Open Systems (ICOS), 2018, pp. 1-6

[2] E. Romera et al., "TACO: Trash Annotations in Context Object Detection Dataset," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9447-9456.

[3] A. H. Vo, L. Hoang Son, M. T. Vo and T. Le, "A Novel Framework for Trash Classification Using Deep Transfer Learning," in IEEE Access, vol. 7, pp. 178631-178639, 2019, doi: 10.1109/ACCESS.2019.2959033.

[4] M. Fulton, J. Hong, M. J. Islam and J. Sattar, "Robotic Detection of Marine Litter Using Deep Visual Detection Models," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 5752-5758, doi: 10.1109/ICRA.2019.8793975..

[5] T. Wang, Y. Cai, L. Liang, and D. Ye, "A multi-level approach to waste object segmentation," *Sensors*, vol. 20, no. 14, p. 3816, 2020.

[6] M. Kraft, M. Piechocki, B. Ptak, and K. Walas, "Autonomous, onboard vision-based trash and litter detection in low altitude aerial images collected by an unmanned aerial vehicle," *Remote Sensing*, vol. 13, no. 5, p. 965, 2021.

[7] G Mittal, K B Yagnik, M Garg, and N C Krishnan, Spot Garbage: Smartphone App to Detect Garbage Using Deep Learning, ACM International Joint Conference on Pervasive and Ubiquitous Computing, 940-945, 2016

[8] S. Sekar, "Waste classification data," *Kaggle*, 16-Jun-2019. [Online]. Available: https://www.kaggle.com/datasets/techsash/waste-classification-data. [Accessed: 15-Feb-2023].

[9] Yudin, Dmitry and Zakharenko, Nikita and Smetanin, Artem and Filonov, Roman and Kichik, Margarita and Kuznetsov, Vladislav and Larichev, Dmitry and Gudov, Evgeny and Budennyy, Semen and Panov, Aleksandr, Hierarchical Waste Detection with Weakly Supervised Segmentation in Images from Recycling Plants. Available at SSRN: https://ssrn.com/abstract=4183424

[10] C. Zhihong, Z. Hebin, W. Yanbo, L. Binyan and L. Yu, "A vision-based robotic grasping system using deep learning for garbage sorting," 2017 36th Chinese Control Conference (CCC), Dalian, China, 2017, pp. 11223-11226, doi: 10.23919/ChiCC.2017.8029147.

[11] J. Sousa, A. Rebelo and J. S. Cardoso, "Automation of Waste Sorting with Deep Learning," 2019 XV Workshop de Visão Computacional (WVC), São Bernardo do Campo, Brazil, 2019, pp. 43-48, doi: 10.1109/WVC.2019.8876924.

[12] Sousa, Joao, Ana Rebelo, and Jaime S. Cardoso. "Automation of waste sorting with deep learning." 2019 XV Workshop de Visão Computacional (WVC). IEEE, 2019

[13] B. Xue et al., "An Efficient Deep-Sea Debris Detection Method Using Deep Neural Networks," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 12348-12360, 2021, doi: 10.1109/JSTARS.2021.3130238.

[14] Liu, Ying, et al. "Research on automatic garbage detection system based on deep learning and narrowband internet of things." Journal of Physics: Conference Series. Vol. 1069. No. 1. IOP Publishing, 2018.

[15] Majchrowska, Sylwia, et al. "Deep learning-based waste detection in natural and urban environments." Waste Management 138 (2022): 274-284.

[16] Xu, Pan & Li, Qingyang & Zhang, Bo & Wu, Fan & Zhao, Ke & Du, Xin & Yang, Cankun & Zhong, Ruofei. (2021). On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning. Remote Sensing. 13. 1995. 10.3390/rs13101995.

[17] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[18] Xu, Bing, et al. "Empirical evaluation of rectified activations in convolutional network." arXiv preprint arXiv:1505.00853 (2015).

[19] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE conference on computer vision and pattern recognition