Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 3s (2025)

WasteScan: Efficient Waste Detection and Multi-label Classification Leveraging Object Detection Methods

[1]Amruta Hingmire, [2]Uma Pujeri
[1,2]School of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India.
[1]amrutahingmire@gmail.com, [2]uma.pujeri@mitwpu.edu.in

Article History:

Received: *03-10-2024*

Revised: *24-11-2024*

Accepted: *02-12-2024*

Abstract:

This study presents an efficient waste detection and multilabel classification algorithm using single-shot object detection techniques. The objective is to accurately identify and classify various waste objects according to their material, even in scenarios in which objects are densely clustered. The proposed algorithm demonstrates promising outcomes, achieving both exceptional detection accuracy and efficient computational performance. To achieve efficient waste detection, we utilize the YOLOv5, YOLOv7 and YOLOv8 models, known for their ability to detect objects with high precision. These models utilise a one-shot detection technique to predict bounding boxes and class probabilities for many objects in a single run, enabling real-time image processing. This study explored the effectiveness of an incremental learning approach, showcasing notable performance gains for a newly added class. This characteristic is especially crucial for applications that require timely decision-making or monitoring waste in dynamic environments. Furthermore, the models' computational speed guarantees real-time performance, which qualifies them for waste management applications which involve real-time sorting functionalities.

Keywords: Waste Classification, Deep Learning, Computer Vision, Object Detection, Incremental Learning, Single-shot Detection, etc.

1. INTRODUCTION

Urban waste, stemming from both natural and human sources, accumulates on city streets as trash, leaves, illegal dumping, and other materials, resulting in significant economic, environmental, and social consequences. Beyond contributing to air and water pollution, unmanaged waste harms a country's economy, reputation, and tourism. Ineffective waste management exacerbates problems such as landfill- induced ecosystem degradation, clogged drains, and the spread of disease. To combat these issues, recycling is increasingly seen as a solution, but it requires efficient waste sorting, a process that is both costly and time-consuming. Object detection techniques play an important part in improving recycling efforts by ensuring accurate waste classification. The application of deep learning techniques has significantly advanced garbage classification, surpassing traditional methods in both performance and accuracy. Deep learning offers superior detection precision and improved classification capabilities compared to conventional approaches. However, challenges remain in optimizing these methods for waste sorting. To tackle these issues, researchers have explored various detection algorithms and network models, conducting extensive comparisons to identify the most efficient and accurate approaches for garbage classification. Through these investigations, the goal is to refine deep learning methods and determine the best solutions for enhancing waste management systems.

One key challenge lies in improving the detection capabilities of network models for small and multiple targets. Researchers have focused on developing novel techniques to enhance the network's ability to detect such targets more accurately. Additionally, reducing the complexity of network models is crucial for optimizing real-time performance. Researchers have explored methods to simplify the models without compromising detection accuracy, thereby meeting the requirements for real-time garbage detection. The optimization of the detection output prediction bounding box is another area of interest. Researchers have proposed techniques to refine the bounding box predictions, resulting in more precise and reliable outputs. Several studies have proposed improved methods and models to address these challenges. These techniques aim to construct multilevel deep learning models that are fast and precise, with a focus on recognising and categorising common trash items that people come into contact with daily.

This work contributes to the development of an advanced waste management solution using object detection for multilabel classification and efficient trash detection. By incorporating incremental learning for enhanced system efficiency and adaptability, ensuring real-time processing with exceptional computational speed and proving accuracy in detecting waste items, even in densely packed scenarios.

2. BACKGROUND & RELATED RESEARCH

Manually identifying and separating waste based on its material is a crucial step in waste management. To simplify this process, a smart waste material classification system is proposed. In recent research, using a ResNet-50 Convolutional Neural Network (CNN) model as a capable feature extractor has gained popularity. This model, when coupled with a Support Vector Machine (SVM), has demonstrated promising results in classifying waste into distinct categories, including glass, metal, paper, and plastic. The attained outcomes showcase a notable 87% accuracy rate during evaluations conducted on a dataset encompassing diverse trash images. The successful integration of this system can significantly boost the efficiency and expediency of waste separation processes, thereby mitigating the necessity for extensive human intervention [1].

Traditional waste classification and segregation methods have limitations, leading to the investigation of alternative computational approaches. Artificial intelligence (AI) and image processing have emerged as powerful methods for tackling solid waste management issues. However, the application of deep learning techniques in this field is still relatively underexplored. This study presents an intelligent model that employs Convolutional Neural Networks (CNNs) for waste categorization. Three CNN architectures-AlexNet, DenseNet121, and SqueezeNet-were used for classification tasks. The results showed notable success, with DenseNet121 achieving the highest accuracy of 0.9415. These findings underscore the potential of deep learning strategies to enhance waste management by enabling more accurate waste classification [2].

A multilayer hybrid deep-learning system (MHS) was proposed by Chu et al. for automating waste sorting in metropolitan public places [3]. The MHS combines sensors and a camera to capture images of waste and collect more feature data. The system utilises a CNN-based algorithm to extract features from images and then combines these features with additional relevant information using a multilayer perceptron (MLP) method to classify waste into recyclable and non-recyclable categories. In two distinct testing scenarios, the MHS exhibits an impressive overall classification accuracy of over 90%, outperforming a reference CNN-based method that is dependent only on image inputs. The results highlight the effectiveness of the MHS in autonomously sorting waste in real-world settings.

Joseph Redman, Santosh Divvala, Ross Girshick, and Ali Farhadi's study have some interesting findings, one of which is that Fast R-CNN tends to produce a lot more background errors than YOLO (You Look Only Once) [4]. Their study responds to this result by significantly improving Fast R-CNN by successfully removing background detections with the use of YOLO's capabilities. The approach involves a methodical assessment of YOLO's forecasts concerning the bounding boxes produced by Fast R-CNN. Researchers change the likelihood of a forecast if a particular prediction from YOLO closely resembles a bounding box produced by Fast R-CNN, taking into account both YOLO's prediction and the degree of overlap between the two bounding boxes. This novel method significantly boosts performance, constituting a substantial advance in the field of object detection.

In their study, Yadav, Jindal, Rani, and their team explored the role of computer vision in object detection. They highlighted the effectiveness of combining fast CNNs with YOLO can effectively handle low image quality, motion control, and similar object recognition. To identify the best feature-accuracy combination, they compared YOLO, Faster Region-Based Convolutional Neural Networks (R-CNN), and Single-Shot Detectors. Notably, R-CNN outperformed in accuracy, recall, precision, and loss. The study underscored the practicality of deploying these models on mobile devices for real-time object detection, critical for swift and accurate identification. Using a Custom Chess Piece Dataset, the researchers showcased the effectiveness of these models in chess piece identification, substantiating the research findings. This study combines academic rigour with practical applicability, offering valuable insights into the evolving field of object detection [5].

Li et al. reported considerable advances in trash categorisation using deep learning approaches in their 2022 work Deep GarbageNet: Advancements in trash categorisation using CNNs [6]. The study concentrated on employing Convolutional Neural Networks, a core architecture in deep learning, to enhance the accuracy of garbage classification tasks. By discussing the limitations of traditional garbage classification methods and highlighting how deep learning can address these challenges. The authors emphasize that deep learning offers improved detection accuracy and enhanced classification capabilities compared to conventional techniques. A key contribution of the study is the comprehensive experimental evaluation of popular deep learning models, such as YOLO and SSD, in the context of waste classification. Through detailed experimental comparisons, the authors assess the performance and precision of these models. Additionally, they compare these deep learning approaches with mainstream detection techniques to identify the most effective and accurate methods for garbage classification.

The research emphasises the need for enhanced detection abilities, particularly for small and many targets inside the garbage images, to address a significant difficulty in garbage categorization. The authors suggest cutting-edge methods to improve the network's capacity to detect these targets more precisely, a big step towards improving garbage collection technology.

Deep learning and artificial intelligence have advanced significantly in several domains. Patel et al. [7] used object detection models to create a garbage identification system that can automatically detect and locate trash in real-world pictures and videos. Five models for waste object detection are being tested: SSD ResNet-50 V1, Faster R-CNN ResNet-101 V1, EfficientDet-D1, CenterNet ResNet-101 V1, and YOLOv5M. YOLOv5M has the best performance after optimization and hyperparameter adjustment, with a Mean Average Precision of 0.5 mAP is 0.61. This study emphasises YOLOv5M's efficiency in waste detection tasks, as it strikes a balance between detection accuracy and computing efficiency, which is consistent with earlier research demonstrating YOLO's robustness in real-time object detection.

The YOLOv4 and YOLOv4-tiny Darknet-53 versions are tested with various inputs, including pictures, video clips, and live webcams [8]. The YOLOv4-tiny model undergoes experimental hyperparameter tuning, which includes subdivision values and mosaic data augmentation. The results show that YOLOv4 outperforms YOLOv4-tiny in terms of object detection performance, despite YOLOv4-tiny's better computing speed. The YOLOv4 model obtained the best performance with an average IoU of 64.01%, mAP of 89.59%, precision of 0.76, recall of 0.90, and F1-score of 0.82. The best results for YOLOv4-tiny are 81.84% mAP, 0.59 precision, 0.83 recall, 0.69 F1-score, and 48.35% Average IoU. The study also found that models with reduced subdivision values and mosaic data augmentation perform best. In current scenario, deep leaning based models produced high performance in different image-based pattern classification domain [24][25][26].

3. METHODOLOGY

3.1 Object Detection

Image classification assigns a single label to an image based on identified content. Object detection extends image classification by not only identifying objects but also locating them within an image. The detection of objects is a crucial task in computer vision that involves the identification along with localization of objects from input images and videos [9]. These detection algorithms are classified based on their approach to processing input images. For example, single-shot object detection [10] analyzes each image just once, making it ideal for real-time applications in environments with limited resources. However, this method often struggles with accuracy, especially when dealing with the detection of small objects.

Single-shot and two-stage detection algorithms are the two main categories into which visual object detection techniques fall. While two-stage models often provide better recognition performance, single-shot models are faster. A Faster R-CNN is a well-known two-stage architecture and has become a benchmark in object detection. This two-stage method, which calls for object categorization and bounding box regression, creates region recommendations by selective search, as shown in Figure 1. Faster R-CNN works more slowly than single-shot models but provides greater accuracy.

YOLO revolutionised object detection by introducing a one-shot model that predicts both bounding boxes and class probabilities in a single forward pass through the network. Unlike previous methods, which divide the task into distinct parts, YOLO uses an end-to-end neural network to analyse the entire image at once, substantially speeding up the detection process. It uses the full input image as the starting point and immediately regresses the location and bounding box at the output layer. The image is first divided into s x s grids using YOLO models, which then forecast each grid's boundary and determine if it corresponds to the detected object's location and confidence. Over time, various iterations of the YOLO model have emerged, each building upon the previous version's capabilities. Table 1 presents an overview of the architectural components and loss functions employed across these different YOLO model versions. Notably, YOLOv6 [10], YOLOv7 [11], and YOLOv8 [12] have incorporated advanced elements like RepVGG, CSPRepStack, EELAN, and innovative head structures. These enhancements can significantly elevate the performance of object detection as compared with the earlier YOLOv4 and YOLOv5 iterations [13].

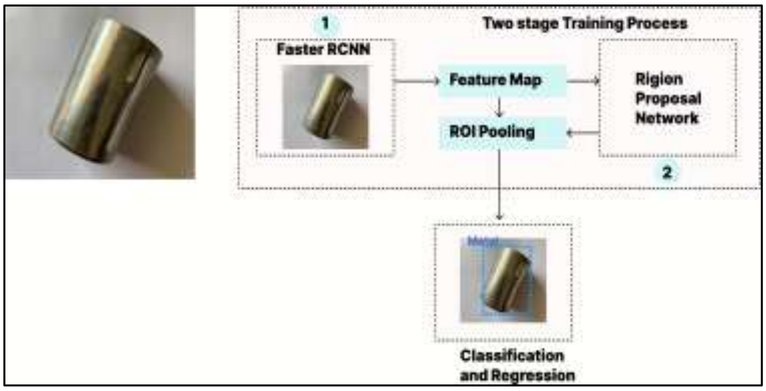

Fig 1: The Two-stage Object Detection Process

Single-shot YOLO models proposed a groundbreaking approach, using an end-to-end neural network to predict bounding boxes and class probabilities simultaneously. It uses the full input image as the starting point and immediately regresses the location and bounding box at the output layer. The input image is divided into s x s grids using YOLO models, which then forecast each grid's boundary and determine if it corresponds to the location and confidence of the object being detected. Over time, various iterations of the YOLO model have emerged, each building upon the previous version's capabilities.

Table 1 presents a comprehensive overview of the architectural components and loss functions employed across these different YOLO model versions. Notably, YOLOv6[10], YOLOv7[11], and YOLOv8[12] have incorporated advanced elements like RepVGG, CSPRepStack, EELAN, and innovative head structures. These enhancements hold the potential to significantly elevate the performance of object detection tasks when compared to the earlier YOLOv4 and YOLOv5 iterations [13].

Table 1: Comparative Analysis of YOLO Object Detection Architectures and Components

Layers	**YOLOv4**	**YOLOv5**	**YOLOv6**	**YOLOv7**	**YOLOv8**
Backbone	CSPDarknet53	CSPDarknet53	RepVGG and CSPRepStack	EELAN	CSPDarknet53
Neck	PANet (Path Aggregation Network) and SPP (Spatial Pyramid Pooling)	**PANet:** To get feature pyramids	RepPAN - Rep operator, which allows structural reparameterization	PANet	YOLO neck in place of the conventional C2f module
Head	$B * (5 + C))$ B: No. of Bounding Boxes C : No. of Classes	$B * (5 + C))$ B: No. of Bounding Boxes C : No. of Classes	Decoupled Classification and Detection Head	Auxiliary Head for middle layer output and Lead Head for final output	Decoupled head, anchor-free model that can handle objectness, classification, and regression tasks separately.
Loss function	**BCE** (Binary Cross Entropy)	**BCE WITH LOGITS LOSS** (Binary Cross Entropy & Logit Loss Function) - for numerical stability.	Varifocal Loss for the Distribution and Classification Task, Focal Loss in Object Detection	Binary Cross Entropy with focal Loss: Classification, IoU **Loss**: Detection	Classification loss: Binary Cross-Entropy; Bounding Box loss: CIoU and DFL loss functions.

3.2 YOLO v8 Evolution

The 'clan' of object detectors known as YOLO has rapidly evolved in the field of computer vision. YOLOv8 is an enhancement over the previous version of YOLO, which makes the model quick, accurate, and simple to use. Alterations were made to the foundational structure of the system, primarily with the introduction of C2f instead of C3. This shift also involved the substitution of the initial 6x6 convolution within the stem with a more compact 3x3 convolution. A distinguishing factor that emerges in the handling of outputs within these outputs arises from the amalgamation of

Bottleneck results, where each Bottleneck integrates two 3x3 convolutions along with residual connections. This contrasts with the approach in C3, where solely the output from the ultimate Bottleneck was employed.

The most recent manifestation of YOLOv8 exhibits a plethora of novel attributes. Of noteworthy mention are the user-friendly Command-Line Interface (CLI) and the comprehensive GitHub repository. This version encompasses a diverse spectrum of applications, encompassing object detection, instance segmentation, and image classification.

3.3 Anchor-free Detections

An object detection model that predicts an object's centre directly, as opposed to using an offset from a specified anchor box, is known as anchor-free detection. In conventional methods, objects of the proper scale and aspect ratio are identified using anchor boxes, which are predetermined sets of boxes with specified heights and widths. During detection, these anchor boxes are chosen based on object sizes from the training data and are tiled throughout the image. The anchor boxes are adjusted based on the model's predictions of each tile's probability and properties, such as background, Intersection over Union (IoU), and offsets. For objects of various sizes, the network may generate probabilities and attributes for each anchor box, which can be utilized as a starting point.

However, in anchor-free detection, there is no need to manually define anchor boxes, making the process more flexible and efficient. This manual anchor specification, used in earlier versions of YOLO such as v1 and v2, often led to suboptimal performance. Anchor-free methods eliminate this issue, providing a more streamlined approach to object detection.

4. RESULTS & DISCUSSION

We collected all solid waste images from the TrashNet[14] dataset using data augmentation techniques, and 3921 images total were produced as a result. Out of them, 1000 were chosen for the test set and 2921 images were used for training. Annotations from six categories-cardboard, glass, paper, plastic, metal, and trash-were used in our study. The prepared dataset was used to evaluate the object detection capabilities of YOLOv5, YOLOv7, and YOLOv8 models on Colab, utilizing the Tesla T4 GPU. These models exhibited notably higher detection accuracy compared to earlier iterations of YOLO. Furthermore, the trained models demonstrated the ability to achieve benchmark-worthy outcomes on unfamiliar datasets, showcasing improved speed and accuracy [15].

The mAP metric evaluates the accuracy of object detection models by considering both precision and recall across all detection thresholds.

$$AP = \int_0^1 Precision(Recall)\, dRecall \qquad (1)$$

$$mAP = \frac{1}{6}\sum_{c=1}^{6} AP_c \qquad (2)$$

where C is the number of classes and AP_c is the AP for class c.

Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 28 No. 3s (2025)

4.1 YOLOv5 Performance: Precision, Recall, and mAP Analysis:

YOLOv5 is an object recognition model integrating bounding box estimation and object classification into a single end-to-end differentiable network [16]. It was developed and maintained within the Darknet environment. As the first YOLO model built using the PyTorch framework, YOLOv5 offers a more lightweight and user-friendly experience compared to its predecessors.

The F1 confidence plot for YOLOv5 illustrates how the model's F1 score changes across various confidence thresholds, aiding in the selection of an optimal threshold for detection.

As shown in figure 2, the precision-recall curve for YOLOv5 illustrates the trade-off between precision and recall at different confidence levels. The graph depicts the model's performance across multiple-choice criteria. The YOLOv5 confusion matrix details each class's true positives, true negatives, false positives, and false negatives. It gives a complete assessment of the model's performance across a variety of waste categories.

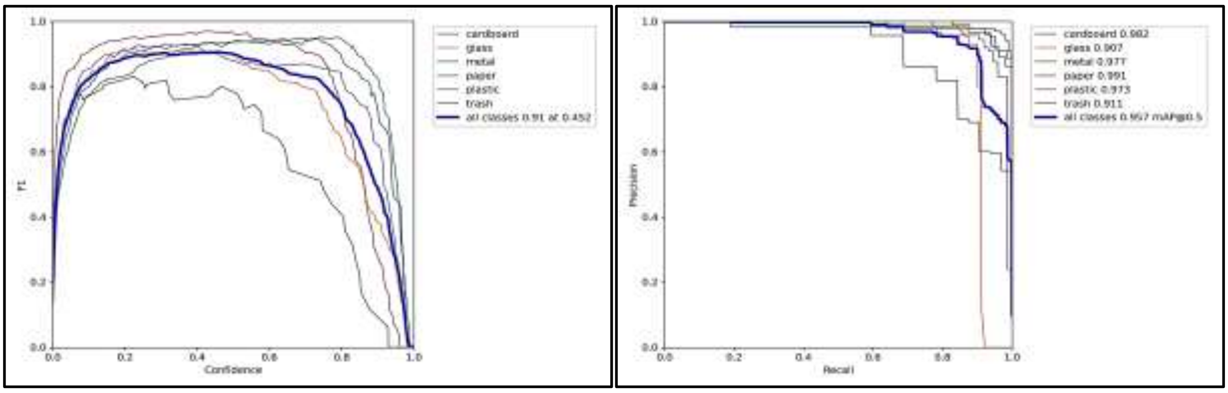

Fig.2. (a) F1 confidence plot for YOLOv5, (b) Precision-Recall curve for YOLOv5

4.2 Performance Evaluation of YOLOv7: Precision, Recall, and mAP

YOLOv7 is the newest and improved detector in the YOLO family. The trainable bag-of-freebies in this network allows real-time detectors to achieve significant gains in precision without increasing the cost of inference. Combining extend and compound scaling, allows the target detector to compute fewer parameters and achieve a much faster detection rate. YOLOv7 performs faster (5 to 160 frames per second) and more accurately than standard object detectors. It also offers a ready-to-use set of freebies and simplifies the process of fine-tuning detection models. It's easy to add more modules and make new models with the YOLOv7 configuration file.

4.3 YOLOv7 Architecture

The YOLOv7 architecture is derived from earlier models such as YOLOv4, Scaled YOLOv4[17], and YOLO- R [18]. To effectively detect small objects, YOLOv7 employs a pre-processing strategy that combines techniques from both YOLOv5 and its design. In terms of architecture, an enhanced version of ELAN, known as E-ELAN, serves as a fundamental component of YOLOv7. This design aims to improve the network's learning capabilities while maintaining a clear gradient path by utilizing techniques such as expansion, shuffling, and merging cardinality. Within the architecture of the computational block, group convolution is employed to enhance both the channel dimensions and cardinality. Each set of computational blocks is assigned specific instructions to learn various features. Additionally, YOLOv7 incorporates compound model scaling for models based on concatenation.

Setting up the model: The 'YOLOv7' code from GitHub is obtained and cloned to install the YOLO package. The latest version of 'YOLO v7' is supported by Torch and is simple to implement with the aid of 'Google Colab' harnessing the processing power of a Tesla T4 GPU.

YOLOv7 exhibited significantly enhanced object detection capabilities compared to its predecessors, showcasing notably higher detection accuracy.

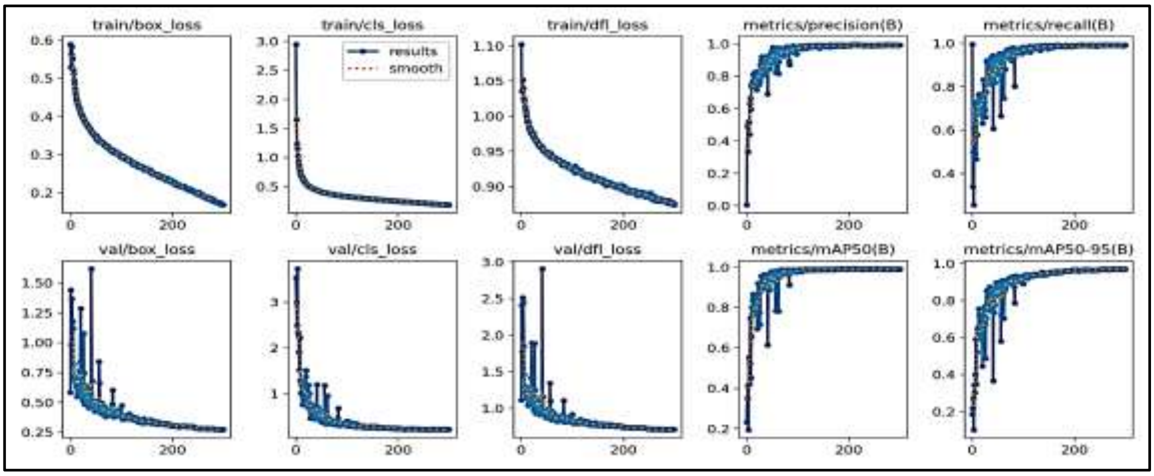

Fig.3. Plots of the YOLOv7 training epochs for the box loss, abjectness loss, classification loss, accuracy, recall, and mAP in the training and validation sets.

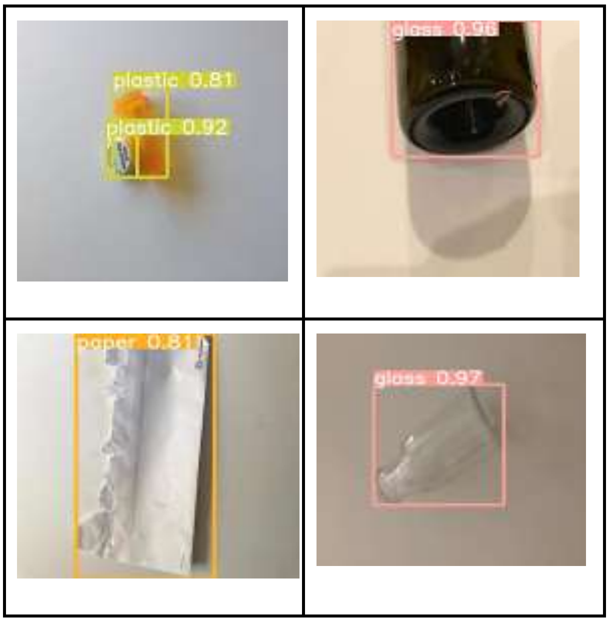

Fig.4. YOLOv8 predictions for TrashNet waste dataset

4.4 Assessing YOLOv8 Performance: Precision, Recall, and mAP Comparison

Strong performance on COCO [19] demonstrates that YOLOv8 is quite accurate. It offers features that are helpful to developers, such as a simple CLI and a well-organized Python package. The computer vision community is active and expanding, which provides the model with plenty of support and direction. Utilising a modified Darknet annotation format, YOLOv8 employs the YOLOv5 PyTorch TXT annotation format.

Loss Function

YOLOv8 introduces improvements, like anchor-free predictions and adaptive weight assignment. The loss remains modular:

- Bounding Box Loss: GIoU (Generalized IoU) or CIoU.
- Objectness Loss: BCE or Focal Loss with adaptive thresholding.
- Classification Loss: BCE or Focal Loss.

The total loss can be represented as:

$$L_{total} = L_{box} + L_{Obj} + L_{class} \quad (3)$$

In all cases:

L_{box}Measures how accurately the bounding boxes overlap with ground truth.

L_{Obj}Measures the confidence of object presence.

L_{class} Measures the accuracy of class prediction.

The model specifically achieves a remarkable 50.2% mAP on COCO. Additionally, YOLOv8 improves the entire development experience compared to earlier models by streamlining the training process with a user-friendly CLI and a Python package.

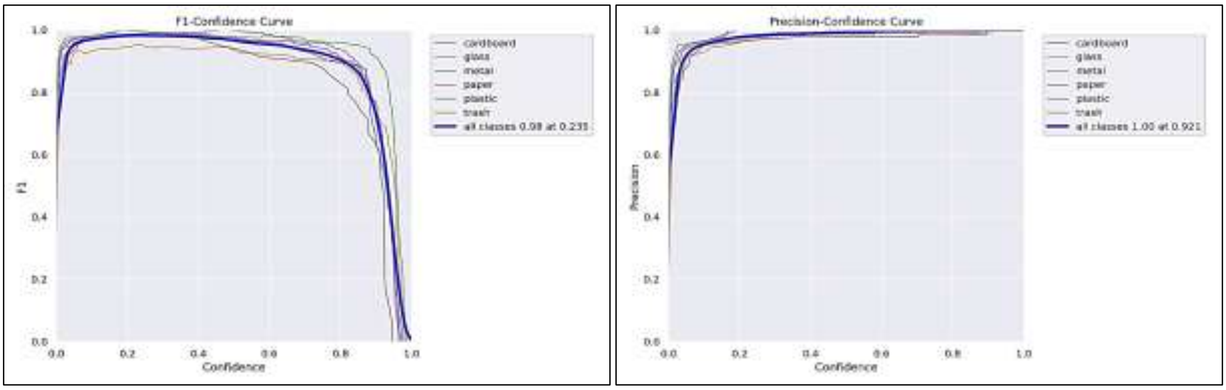

Fig.5. (a) F1 confidence plot for YOLOv8, (b) Precision-Recall curve for YOLOv8

The comparison demonstrates that at an Intersection over the Union (IoU) with a 0.5 threshold, YOLOv7 outperforms in terms of precision, recall, and mAP. With the highest precision and recall ratings, this version generates a remarkable MAP score of 99.0%. Additionally, YOLOv8 has strong performance across all measures, with the top Detection speed, Precision, Recall, and MAP scores. For this waste identification job utilising the TrashNet dataset, YOLOv4 and YOLOv5 provide comparable results [20], but they lag somewhat behind YOLOv7 and YOLOv8 in terms of overall accuracy.

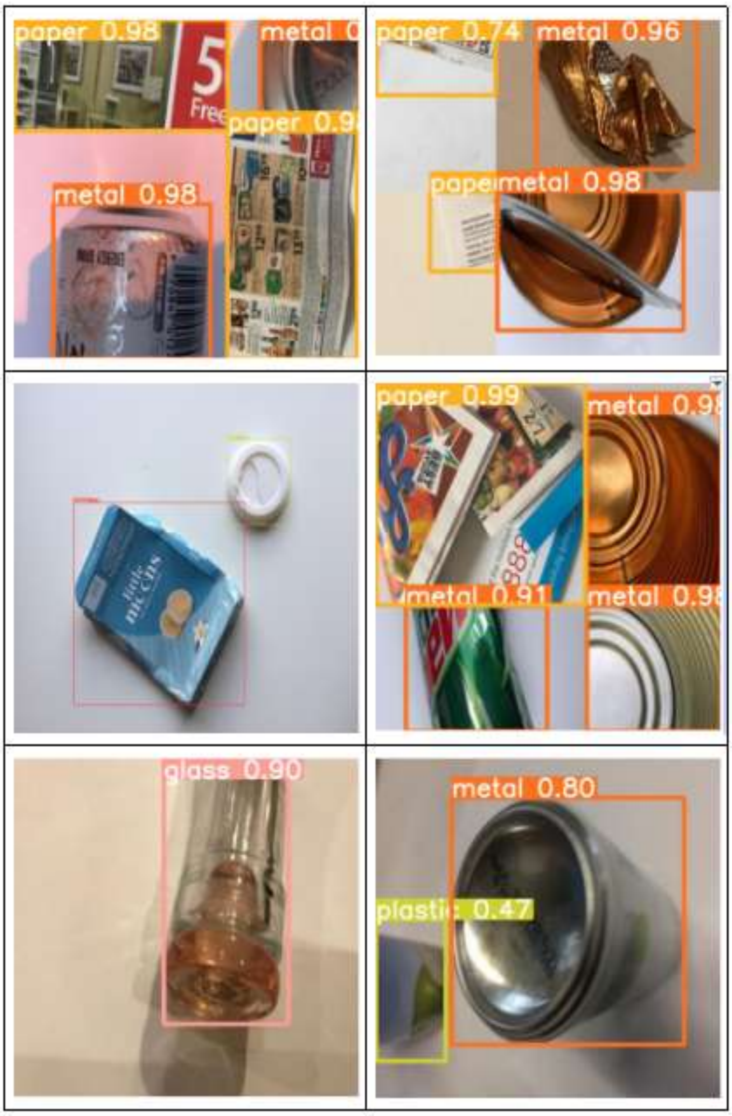

Fig.6. YOLOv8 multilabel waste predictions for TrashNet waste dataset

Table 2: Analysing YOLO Object Detection Models: Layer Complexity, Parameter Counts, and Computational Efficiency

YOLO Version	Layers	Parameters	GFLOPs
YOLOv5	232	7,260,003	16.8
YOLOv8	218	25,843,234	78.7
YOLOv7	314	36,508,742	103.2

Table 3: Performance Metrics of YOLO Object Detection Models for TrashNet Dataset

Model	Precision	Recall	mAP @ 0.5
YOLOv4	96.4	93.6	98.1
YOLOv5	91.1	94.7	96.2
YOLOv7	99.3	99.2	99.0
YOLOv8	98.4	98.4	99.1

Table 2 shows that YOLOv7 has the highest number of layers (314), which is more than YOLOv8 (218) and YOLOv5 (232), suggesting a more complex architecture. YOLOv7's high parameter count may enable it to capture more intricate features, but it may also increase the likelihood of overfitting, particularly on smaller datasets. In comparison to YOLOv7 (103.2 GFLOPs) and YOLOv8 (78.7 GFLOPs), YOLOv5 has the fewest parameters and is computationally more efficient (16.8 GFLOPs). This implies that YOLOv5 uses a more compact architecture to achieve its performance, which speeds up inference. Compared to YOLOv7, OLOv8 achieves a balance between computational efficiency (GFLOPs) and complexity (number of layers). YOLOv5 may be the better choice if computational efficiency is crucial.

4.5 Incremental Learning

When deep neural networks are exposed to new tasks or data distributions, they tend to lose previously learned tasks or skills, a process known as catastrophic forgetting [21]. The model gains knowledge gradually as new instances are added, changing its pre-existing weights to account for the new data, frequently at the expense of performance on previous tasks. Because models must adjust without wiping existing knowledge, this problem is especially present in situations where continual learning is necessary. This problem is addressed by incremental learning, which involves gradually updating a deep learning model with fresh data or classes while training the model in phases. Incremental learning allows the model to change over time while keeping previously learned information, in contrast to classical learning, which necessitates retraining the model from scratch [22]. This procedure is particularly helpful when an existing model needs to be expanded to include additional classes or categories, such as newly discovered waste types [23].

Table 4: Comparison of Incremental Learning Types – Key Features, Techniques, Challenges, and Applications

Type	Description	Key Techniques	Advantages	Challenges	Applications
Instance-Based	Updates model sequentially with small data instances.	Online SGD, Online SVM	Real-time updates, low memory usage	Overfitting, sensitive to noisy data	Streaming data, IoT systems
Class-Based	Learns new classes while retaining old knowledge.	EWC, Knowledge Distillation, Replay	Prevents catastrophic forgetting	Balancing old vs. new knowledge	Incremental classification tasks
Task-Based	Learns distinct tasks with task-specific isolation.	Multi-head Networks, Progressive Networks	Handles multiple tasks separately	Model complexity increases with tasks	Multi-task systems, robotics

Memory-Based	Retains past data or uses generative models for replay.	Experience Replay, Generative Replay	Combines old and new knowledge effectively	Trade-off between memory and accuracy	Time-series prediction, reinforcement learning
Dynamic	Adapts to data distribution changes (concept drift).	Drift Detection, Online Ensembles	Handles non-stationary data effectively	Detecting and reacting to drift	Fraud detection, adaptive systems
Hybrid	Combines multiple incremental learning strategies.	Replay + Ensembles, Adaptive Networks	Versatile and robust	Higher computational and memory overhead	Complex adaptive systems, healthcare

Class-based incremental learning is ideal for this work due to its ability to adapt to new classes without forgetting previously learned ones. In the context of waste classification, as new categories (such as e-waste) emerge, class-based incremental learning allows the model to expand its recognition capabilities incrementally, ensuring that existing classes like cardboard, glass, or plastic are still accurately detected. To start with this, first set of model parameters, represented by θ_0, is a pre-trained model before the incremental learning process starts.

Initial Model Parameters

Let θ_0 represent the initial model parameters.

a. First Training Step (TrashNet Classes)

Train the model on a dataset labelled with the six original classes. The training weights are stored in a file named Yolo8mbest.pt, and they belong to the following six classes: cardboard, glass, paper, plastic, metal and trash. Update the model parameters to obtain θ_1.

$$\theta_1 = Train\ (\theta_0, Dataset\ TrashNet) \quad (4)$$

b. Update Weights from Step 1

Save the weights obtained from the first training step as Yolo8mbest.pt. These weights will be used as the starting point for the next training step.

c. Second Training Step (e-waste Class)

Train the model on a new dataset that includes the e-waste class, using the updated weights from Step 1. Update the model parameters to obtain θ_2.

$$\theta_2 = Train\ (\theta_1, Dataset\ e - Waste) \quad (5)$$

d. Final Model Parameters:

Save the weights obtained from the second training step as best.pt or last.pt.

The overall incremental learning process can be represented as:

$$\theta_0\,(Pretrained\ Model) \rightarrow\ \theta_1\,(Trained\ on\ TrashNet\,) \rightarrow\ \theta_2\,(Trained\ on\ eWaste\,) \quad (6)$$

The YOLOv8 model initially trained on six classes from the TrashNet dataset, underwent incremental training to accommodate a new class, e-waste. The resulting performance metrics are presented in Table 4.

Table 5: Performance Metrics of YOLOv8 with Incremental Learning (Epochs: 15)

Model	**Precision**	**Recall**	mAP **@ 0.5**
YOLOv8	92.5	89.1	94.6
YOLOv8(Incremental)	93.3	84.4	90.5

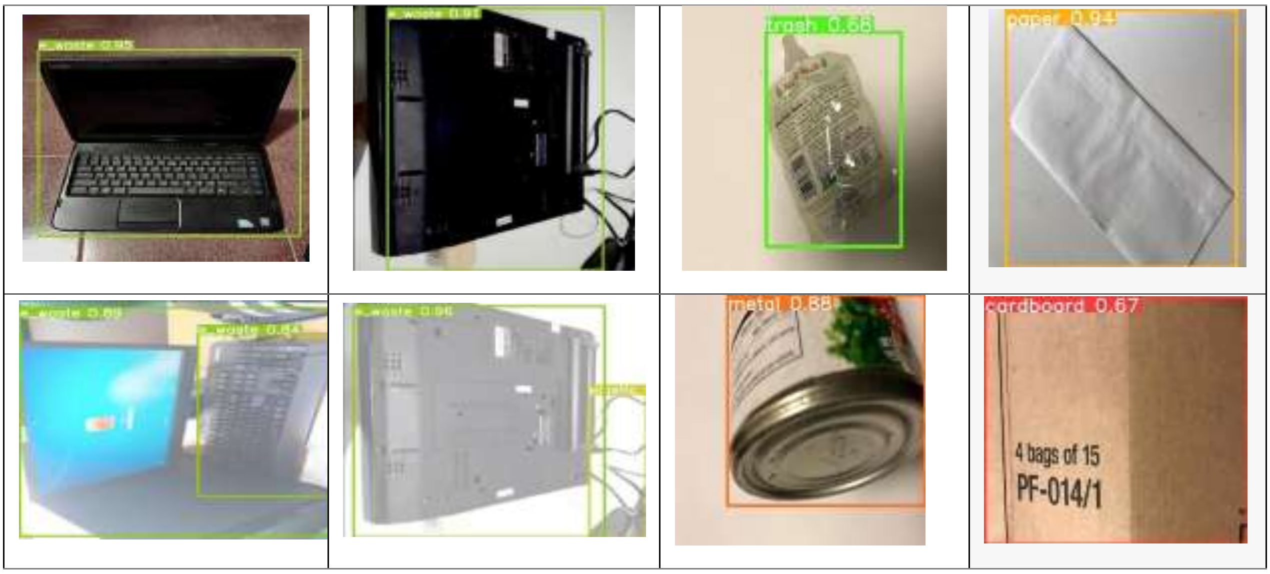

Fig.7. Incremental Learning: Detections on Newly Added e-waste Class Retaining Previously Trained Knowledge

Figure 7 illustrates the detection results for the newly added e-waste class, demonstrating the model's continued accurate detection of previously learned classes.

5. CONCLUSION

This study uses the most recent single-shot item detection models, specifically YOLOv5, YOLOv7, and YOLOv8, to present an effective garbage detection and multilabel classification method. Our primary goal was to effectively recognise and categorise a diverse spectrum of waste products, particularly in circumstances involving closely packed waste objects. Our study's findings demonstrate the efficacy of the proposed technique in detection accuracy and computational speed. We utilized YOLOv5, YOLOv7, and YOLOv8, renowned for their outstanding accuracy in object identification tasks, to effectively detect waste. By simultaneously predicting bounding boxes and class probabilities for multiple objects at the same time, these models, which employ a single-shot detection method, enable real-time image processing. This finding highlights the potential of incremental learning methodologies in improving the adaptability and efficiency of classification systems retaining knowledge of previously learned classes. This characteristic is particularly important for applications that require rapid decision-making, such as waste monitoring in dynamic and unpredictable environments. The remarkable precision, recall, and mAP @0.5 scores demonstrate the exceptional accuracy with which our algorithm consistently detects waste items, even when objects are densely grouped. These models are especially well suited for time-sensitive waste identification and classification tasks in dynamic, real-world scenarios since the computational speed of the models guarantees real-time performance.

Declarations

Funding

No funding was received to assist with the preparation of this manuscript.

Conflicts of interests

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Data Availability Statement:

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

6. REFERENCES

[1] "Intelligent Waste Classification System Using Deep Learning Convolutional Neural Network," *Procedia Manufacturing*, vol. 35, 2019, doi: 10.1016/j.promfg.2019.05.086.

[2] "Solid Waste Classification Using Deep Learning Techniques," *Proceedings of the International Conference on Open Source Technology and Education Networks (ICOTEN)*, Yemen, 2021, pp. 1–5, doi: 10.1109/ICOTEN52080.2021.9493430.

[3] Y. Chu, "Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling," *Advances in Multimedia*, vol. 2018, Art. no. 5060857, 2018, doi: 10.1155/2018/5060857.

[4] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[5] S. P. Yadav et al., "An Improved Deep Learning-Based Optimal Object Detection System from Images," *Multimedia Tools and Applications*, 2023, doi: 10.1007/s11042-023-16736-5.

[6] "GarbageNet: A Unified Learning Framework for Robust Garbage Classification," *IEEE Transactions on Artificial Intelligence*, vol. 40, 2021, doi: 10.1109/TAI.2021.3081055.

[7] A. Patel et al., "Garbage Detection Using Advanced Object Detection Techniques," in *Proceedings of the International Conference on Advances in Computing, Communication, and Information Science (ICAIS)*, India, 2021, pp. 526–531, doi: 10.1109/ICAIS50930.2021.9395916.

[8] A. P. Saputra et al., "Waste Object Detection and Classification Using Deep Learning Algorithm: YOLOv4 and YOLOv4-tiny," *International Journal of Artificial Intelligence*, vol. 12, pp. 1666–1677, 2021.

[9] J. Yao et al., "Two-Stage Detection Algorithm for Kiwifruit Leaf Diseases Based on Deep Learning," *Plants*, vol. 11, no. 6, Art. no. 768, 2022, doi: 10.3390/plants11060768.

[10] C. Li et al., "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," [Online]. Available: https://doi.org

[11] C.-Y. Wang et al., "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, doi: 10.1109/CVPR52729.2023.00721.

[12] G. Jocher et al., "YOLO by Ultralytics," 2023. [Online]. Available: https://ultralytics.com

[13] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machines*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[14] A. Thung et al., "TrashNet," 2016. [Online]. Available: https://github.com/garythung/trashnet

[15] A. Bochkovskiy et al., "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020, doi: 10.48550/arXiv.2004.10934.

[16] "YOLOv5," Ultralytics, 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[17] C.-Y. Wang et al., "Scaled-YOLOv4: Scaling Cross Stage Partial Network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13029–13038, doi: 10.1109/CVPR46437.2021.01283.

[18] C.-Y. Wang et al., "You Only Learn One Representation: Unified Network for Multiple Tasks," *arXiv preprint arXiv:2105.04206*, 2021, doi: 10.48550/arXiv.2105.04206.

[19] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, doi: 10.1007/978-3-319-10602-1_48.

[20] A. Hingmire et al., "Advances in Garbage Detection and Classification: A Comprehensive Study of Computer Vision Algorithms," *Journal of Computer Vision Applications*, vol. 12, pp. 767–777, 2023.

[21] F. M. Zenke et al., "Continual Learning Through Synaptic Intelligence," *Proceedings of the Neural Information Processing Systems (NeurIPS)*, vol. 70, pp. 3987–3995, 2017.

[22] G. M. van de Ven et al., "Three Types of Incremental Learning," *Nature Machine Intelligence*, vol. 4, pp. 1185–1197, 2022, doi: 10.1038/s42256-022-00568-3.

[23] A. Hingmire and U. Pujeri, "Chasing Pelican Based Deep Learning for Multiple Object Detection from Single Input Trash Image," *Multimedia Tools and Applications*, 2024, doi: 10.1007/s11042-024-19718-3.

[24] D. Mane, N. Londhe, N. Patil, O. Patil, and P. Vidhate, “A Survey on Diabetic Retinopathy Detection Using Deep Learning,” Data Engineering for Smart Systems, Springer, Singapore, 2022, pp. 621–637. doi: 10.1007/978-981-16-2641-8_59

[25] Mane, D.T., & Kulkarni, U.V. (2017). A Survey on Supervised Convolutional Neural Network and Its Major Applications. *Int. J. Rough Sets Data Anal., 4*, 71-82.

[26] Mane, D., Ashtagi, R., Suryawanshi, R., Kaulage, A.N., Hedaoo, A.N., Kulkarni, P.V., Gandhi, Y. (2024). Diabetic retinopathy recognition and classification using transfer learning deep neural networks. Traitement du Signal, Vol. 41, No. 5, pp. 2683-2691. https://doi.org/10.18280/ts.410541