International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

# Advances in Garbage Detection and Classification: A Comprehensive Study of Computer Vision Algorithms

**Amruta Hingmire[1], Dr. Uma Pujeri[2]**

**Submitted**: 28/08/2023 **Revised**: 23/10/2023 **Accepted**: 01/11/2023

**Abstract:** Effective waste detection and classification are crucial for addressing waste management challenges and promoting recycling and reuse of waste materials. The long-term environmental impacts of plastic, metal, and glass-based waste highlight the importance of proper identification, sorting, and utilization of these waste categories. Although various deep learning algorithms have been developed for waste detection, they often struggle to detect multiple garbage categories from a single input image. This research focuses on utilizing computer vision algorithms, specifically the YOLO (You Only Look Once) approach and its variant, which incorporates Convolutional Neural Network (CNN) models, for garbage detection and classification. The efficacy of these models is demonstrated through their impressive performance in waste management tasks. In summary, this research underscores the prowess of Tiny YOLOv4, not only amplifying waste detection capabilities but also envisioning its transformative role in advancing automated waste management practices.

***Keywords:*** *Computer Vision, Garbage Detection, Object Detection, Single-Shot Learning, Transfer Learning, Waste Classification, Waste Detection, YOLOV4, etc.*

## 1. Introduction

The accumulation of non-biodegradable waste has a profound impact on the environment, necessitating the implementation of safe and resource-efficient waste disposal methods. Among these concerns, the management of Municipal Solid Waste (MSW) has emerged as a critical issue due to its association with environmental degradation, resource depletion, and health hazards arising from improper waste disposal practices. Of particular interest within the category of MSW is the dry waste segment, encompassing materials like metal, paper, plastic, and glass, which hold significant potential for reuse and recycling initiatives. Improper handling of dry waste not only contributes to environmental pollution but also squanders opportunities for resource conservation, thereby exacerbating carbon emissions. In this context, the establishment of an efficient garbage classification system for dry waste becomes imperative to promote sustainable waste management practices. Unfortunately, the current approach to waste inspection relies heavily on human intervention and is plagued by time-consuming processes. As exemplified by data released by the Central Pollution Control Board Delhi in 2021, India confronts a substantial challenge, generating a daily total of 160,038.9 tonnes of solid waste. While 95.4% of this waste is successfully collected at a rate of 152,749.5 tones, 50% undergoes treatment, and 18.4% is relegated to landfills. Alarmingly, a sizable proportion of 31.7% (equivalent to 50,655.4 tonnes) remains unaccounted for [1]. The following table illustrates the per capita solid waste generation in India between 2015 and 2021.

**Table 1.** Per Capita Solid Waste Generation in India from 2015-2021[1]

| Year | Annual Solid Waste Generation per Capita (grams per day) |
|---|---|
| 2015-16 | 118.68 |
| 2016-17 | 132.78 |
| 2017-18 | 98.79 |
| 2018-19 | 121.54 |
| 2019-20 | 119.26 |
| 2020-21 | 119.07 |

The proper segregation of waste is crucial regardless of waste generation trends. Waste collection and segregation often go hand in hand in India, making segregation a vital

*1 School of Computer Engineering and Technology, MIT World Peace University, India*
*ORCID ID: 0000-0001-6921-8823*
*2 School of Computer Engineering and Technology, MIT World Peace University, India*
*ORCID ID: 0000-0002-4228-0034*
** Corresponding Author Email: amrutahingmire@gmail.com*

step in determining the recyclable and reusable waste. Consequently, waste management has attracted researchers' attention, resulting in significant research on real-time waste detection. Computer Vision and Deep Learning, particularly using Convolutional Neural Networks (CNN), has gained recognition and commercial application. The need for precise object detection models has led to various approaches. Notable advancements have been made in object detection through influential models such as R-CNN [2], Fast R-CNN [3], Faster R-CNN [4], R-FCN [5], SSD [6], and YOLO [7]. These models have been extensively studied and evaluated in recent research [8, 9].

The paper's structure is outlined as follows: Section 2 reviews pertinent literature, Section 3 details the dataset assembly methodology, Section 4 explains neural network training, Section 5 presents study findings, and Section 6 concludes and suggests future research directions.

## 2. Related Work

Extensive research is being conducted to address the global impact of waste, focusing on developing approaches and techniques for effective waste management and sorting. A common design approach involves a two-stage process: generating object region proposals using traditional Computer Vision algorithms or deep networks, and performing object categorization through bounding-box regression based on extracted features.

### 2.1 Deep Learning-based Object Detection Techniques

In 2016, He et al. introduced the Residual Networks, or ResNet, are a type of deep neural network architecture in 2015, which won the ImageNet Large Scale Visual Recognition Competition (ILSVRC) on the ImageNet dataset, which utilize residual connections to enable efficient training of deep networks without introducing additional parameters or computational overhead [10]. ResNet was designed to solve accuracy problems that arise when deep neural networks become too complex. As layers are stacked to create deeper networks, training, and test errors often increase. The model effectively addressed this issue by introducing residual blocks, which allowed the creation of deep convolutional neural networks. These blocks consist of three convolutional layers, batch normalization, and a rectified linear unit (ReLU) activation function; going beyond the conventional stacking of convolutional layers [11].This connectivity structure facilitates smooth gradient flow, allowing for the successful training of networks with hundreds of layers. Building upon this idea, Huang et al. proposed DenseNet in 2017[12]. The DenseNet architecture takes advantage of feature maps from all preceding layers as inputs, promoting feature reuse, enhancing feature propagation, and reducing the number of parameters. This approach effectively addresses the vanishing gradient problem and improves both training efficiency and accuracy. Another notable architecture is EfficientNet [13], incorporates modules designed using a neural architecture search procedure, which aims to optimize both floating-point operations (FLOPS) and model accuracy. EfficientNet achieves state-of-the-art performance on a variety of applications while maintaining computational efficiency by carefully balancing network depth, width, and resolution. In summary, the ResNet family introduced residual connections for efficient training, DenseNet enhanced feature propagation and reuse, and EfficientNet leveraged neural architecture search to strike a balance between FLOPS and accuracy, resulting in highly effective and efficient deep learning models.

Gary Thung used Support Vector Machine (SVM) and CNN algorithms in [8] to classify waste. By collecting photos of single pieces of trash from six different categories, CNN and SVM were the two methods used to train the models and assess their accuracy. The outcome showed that SVM offers greater accuracy than CNN.

**Table 2.** Datasets for Solid Waste Classification and Detection with Dataset Information and Characteristics

| Name of Dataset | No. of Images | Size of Images | Source of Images | Classification Labels |
|---|---|---|---|---|
| TrashNet [14] | 2,534 | 512 x 384 | Created by authors | 6 (cardboard, glass, metal, paper, plastic, trash) |
| D-SWASTE [15] | 2,240 | 512 x 384 | Collected from internet | 5 (bio, glass, metal, paper, plastic) |
| WASTE-CNN [16] | 12,000 | 224 x 224 | Collected from internet | 2 (recyclable, non-recyclable) |
| UEC FOOD 100 [17] | 10,000 | 256 x 256 | Collected from internet | 100 food categories (for food waste) |

ResNet50 is a variation of the ResNet architecture that has been widely used for image classification tasks. Its input dimensions are 224 by 224 (Li et al., 2021). The use of ResNet50 in image classification tasks has shown better feature extraction compared to previous versions of the ResNet architecture, thanks to its deep structure and batch normalization between residual blocks [18].

Author Chen et al [2] Suggested the fusion object detection approach based on Faster R-CNN. The RGB and

Depth data are input through two feature extraction modules in the two-stream fusion object identification network model that they designed. The two modules are combined after the FC layer at the ROI Pooling layer, where the fusion layer uses the concatenation layer to combine the CNN features and produce the detection result. The Faster R-CNN network uses the anchor box design and the VGG network framework; it does not use the additional stages of the candidate frame technique. As a result, when compared to previous approaches, their method can offer greater accuracy and greater speed.

According to recent research (Li et al., 2021)[5], traditional machine learning methods such as histogram of directed gradients, scale-invariant feature transforms, and the Viola-Jones object detection algorithm were commonly used for object detection and classification before the rise of Deep Learning. These methods involved identifying recurring features in images and categorizing those using algorithms such as random forests and logistic regression. Some of the popular Deep Learning models used for image classification include AlexNet, VGG16, ResNet, MobileNet, Inception-ResNet, and DenseNet [5]. These architectures have been pre-trained using the COCO dataset, which is a popular dataset for object detection tasks [8].

### *2.2 Single-shot Object Detection*

Object detection techniques with only one stage are YOLO (2016) [19], SSD (2016), RetinaNet (2017) [21], YOLOv3 (2018) [22], YOLOv4 (2020) [23], and YOLOR (2021) [24]. One-stage detectors forecast bounding boxes over the pictures without a region proposal phase. It is possible to apply this process in real-time applications since it is faster.

The supervised learning YOLO model requires object information to be passed during the training phase. So, for each image, the model requires an image and a text file containing bounding box details in a specific format. The Tiny YOLOv4 algorithm [25] is a lightweight object detection algorithm that is specifically designed for real-time applications on low-power devices such as smart phones and embedded systems. It uses a grid-based approach to divide the input image into smaller cells and predicts bounding boxes for objects within each cell.

- Four descriptors are employed to characterize each bounding box:
- Width (bw): This descriptor represents the horizontal extent of the bounding box.
- Height (bh): It denotes the vertical extent of the bounding box.
- Center coordinates (bx, by): These descriptions indicate the position of the bounding box's center.
- Class representation (c): The class of the specific object is denoted by the value assigned to this descriptor, such as cardboard, glass, plastic, metal, trash, etc.

The backbone network of convolutional layers extracts features maps from the input image, which are then processed by detection heads to make predictions about the location and class of objects within each cell.

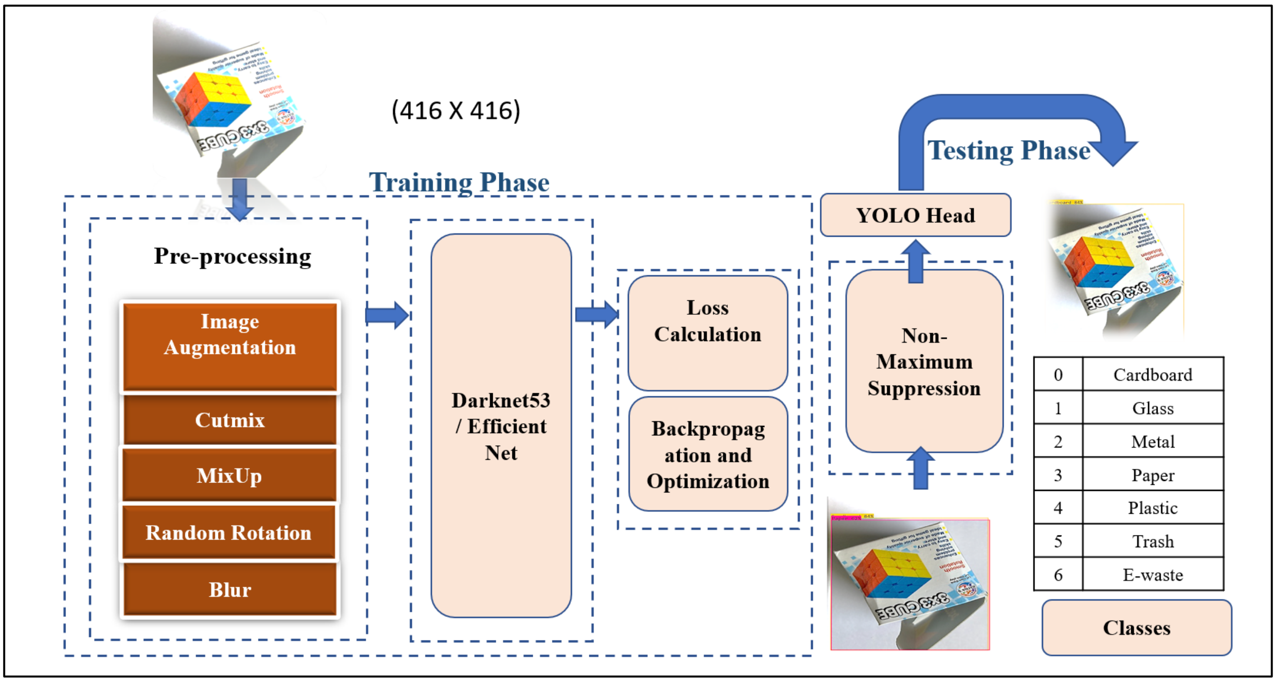


**Fig.1.** Block Diagram for Object Detection using Darknet CSP Architecture

To improve the accuracy of the predictions, the algorithm employs anchor boxes and feature pyramid networks (FPN). Anchor boxes are pre-defined bounding boxes that guide the algorithm's predictions, while FPN helps detect objects at different scales. Furthermore, the Tiny YOLOv4 algorithm incorporates several optimizations, such as shortcut connections and channel-wise convolutional layers, to improve its speed and efficiency. These optimizations enable the algorithm to achieve state-of-the-art performance on real-time object detection tasks while using fewer computational resources than other object detection algorithms. In summary, the Tiny YOLOv4 algorithm is a powerful and efficient object detection algorithm suitable for low-power devices, making it an ideal choice for real-time applications. Here's a table showing different versions of YOLO models and their improvements:

**Table 3.** YOLO Architectures

| **YOLO Version** | **Year** | **Backbone Architecture** | **Improvements** |
|---|---|---|---|
| YOLOv1 [19] | 2016 | DarkNet-19 | The first version of YOLO, achieved real-time object detection |
| YOLOv2 [20] | 2017 | DarkNet-19<br>DarkNet-53 | Introduced anchor boxes, batch normalization, and increased speed and accuracy |
| YOLOv3 [22] | 2018 | DarkNet-53 | Feature pyramid network (FPN), multiple input sizes, and improved accuracy |
| YOLOv4 [23] | 2020 | CSPDarkNet-53 | Introduced spatial pyramid pooling, path aggregation network (PAN), and Mish activation function, achieving state-of-the-art performance |
| Tiny YOLOv4[25] | 2020 | CSPDarkNet-53 | Introduced scaled-yolov4 backbone architecture, and optimizations for speed and efficiency, while maintaining high accuracy |

***2.3 Performance Metric of Object Detection Methods***

As depicted in Figure 1, the output of a detector typically includes a list of bounding boxes, corresponding confidence levels, and class labels for detected objects. Object detection methodologies heavily rely on the concepts of precision (P) and recall (R). Precision refers to the model's capacity to accurately identify only the relevant objects in the scene. It is quantified as the percentage of correct positive predictions, indicating how precise the model's detections are.

Recall is a term that describes a model's ability to locate all relevant cases. It's the percentage of positive predictions that are right based on all given ground realities.

Mean Average Precision (mAP) is a prevalent performance metric in target detection. It involves computing the area under Precision-Recall (P-R) curves for various object categories and then averaging across all categories. This comprehensive evaluation considers individual class average precision (AP) values and combines them for an overall assessment.

$$mAP = 1/n \sum_{i=1}^{n} AP_i$$

## 3. Experimental setup

To prepare a dataset for classification using the YOLO algorithm, the following steps are required.

***Collect images and annotate them:*** Gather a collection of images that represent the various objects you want to detect. Annotate each image by drawing bounding boxes around the objects of interest and labeling them with the appropriate class. To prepare a dataset for our analysis, we utilized the TrashNet dataset. Additionally, we utilized image augmentations such as collages of images to increase the size and diversity of our dataset. All experiments were performed on a single environment with access to a Tesla K80 GPU with 12GB of VRAM.

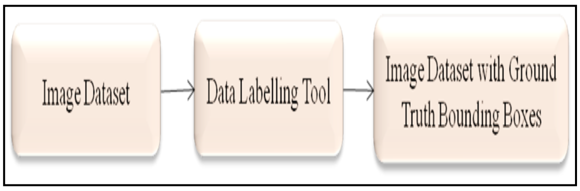


**Fig.2.** Data Preparation Process for Object

Figure 2 illustrates the data preparation process for object detection algorithms. To label the images, a range of tools are accessible, and in this research, the dataset was meticulously annotated using the Roboflow image annotation tool [26].

***Divide the dataset into training and validation sets:*** The dataset is divided into two distinct groups: a training set and a validation set. The training set is used exclusively for training the YOLO model, enabling it to learn from the data. Conversely, the validation set is employed to assess the performance of the trained model, providing insights into its effectiveness. This division ensures a systematic approach to model development, allowing for accurate evaluation and fine-tuning as necessary.

***Convert annotations to YOLO format****:* Convert the annotations for each image into YOLO format. In YOLO v4 tiny format, each annotation file contains one row for each object in the corresponding image, with the class label and normalized coordinates of the object's bounding box (center_x, center_y, width, height) listed in that order, separated by spaces as shown in figure 3.

Type of Object: **Glass**
**Label Number: 1**
**Ground Truth Coordinates:**
0.6057692
0.5096153
0.7848557
0.9711538

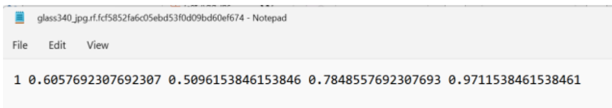
glass340_jpg.rf.fcf5852fa6c05ebd53f0d09bd60ef674 - Notepad
File Edit View
1 0.6057692307692307 0.5096153846153846 0.7848557692307693 0.9711538461538461

**Fig. 3.** Examples of waste image and its Ground truth box coordinates

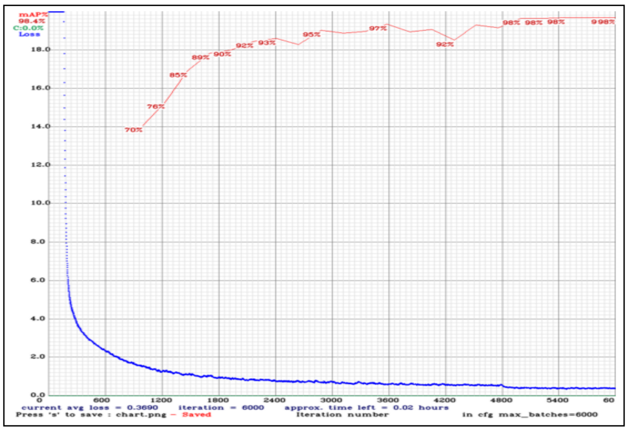


**Fig.4.** Average Loss and mAP YOLOv4-tiny iteration 1000 – 6000

***Create YOLO configuration files:*** Create the YOLO configuration files, which define the YOLO model's architecture, the number of classes, the anchors (default box sizes), and other parameters.

***Train the YOLO model:*** Train the YOLO model on the training set utilizing the generated configuration files. This entails adjusting the model's weights to minimize the loss function. Figure 4 illustrates the progression of Average Loss and mAP for YOLOv4-tiny across iterations 1000 to 6000.

***Evaluate the YOLO model****:* The performance of the YOLO model on the validation set is assessed by calculating key metrics, including mean average precision (mAP) and intersection over union (IoU). These metrics provide valuable insights into the accuracy and robustness of the model's object detection capabilities. The mAP metric measures the precision and recall trade-off, indicating how well the model detects objects across various categories. On the other hand, IoU measures the overlap between predicted bounding boxes and ground truth annotations, providing a measure of the model's spatial accuracy. By evaluating the YOLO model using these metrics, a comprehensive assessment of its performance on the validation set can be obtained.

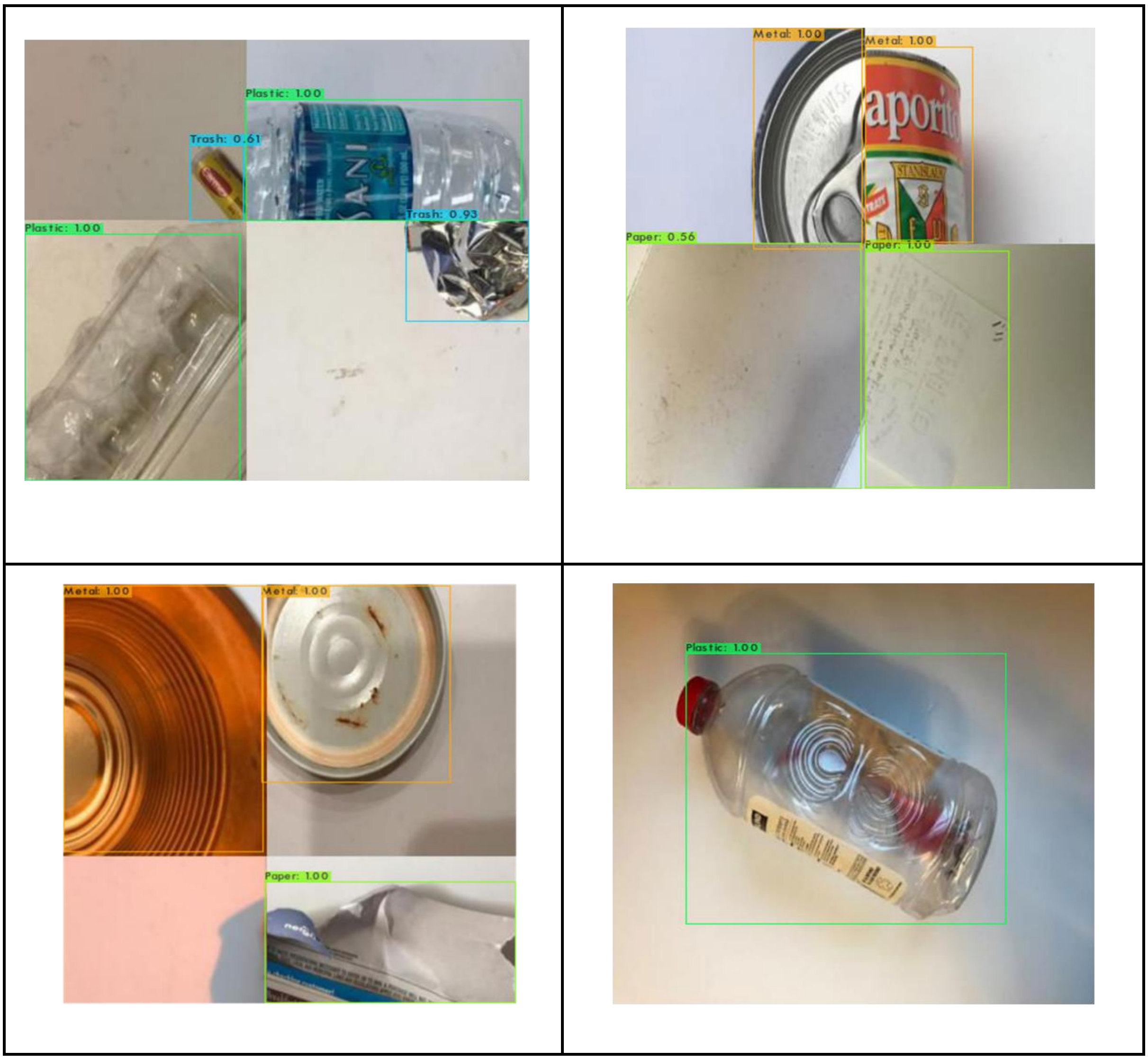


**Fig.5.** Example YOLO -v4-tiny predictions for TrashNet waste datasets

***Use the YOLO model for object detection:*** Once the YOLO model has been trained and validated, it can be used to detect objects in new images. Figure 5 displays several illustrative predictions made by YOLOv4-tiny on the TrashNet waste dataset.

**Dataset used:** Collage-based TrashNet dataset
**Image size:** 512 x 384 pixels, with a file size of approximately 14.4 KB per image
**Total number of images:** 4180
**Number of images used for training:** 3762
**Number of images used for testing:** 418

The utilization of a GPU during the neural network training process in our experiment has significantly expedited the convergence of the network. To evaluate the performance of the model, we conducted a comparative analysis of the precision rate, recall rate, average precision rate, and average detection time before and after implementing improvements to the model. This evaluation was performed on the test set, allowing us to assess the effectiveness of the model enhancements and gain insights into its overall performance.

***1. YOLO Algorithm:***
**Notations used:**
$X= \{x_1,x_2,x_3\ldots.x_n\}$ // n number of images
$a= \{a_1,a_2,a_3\ldots.a_n\}$ // text file containing Labels and bounding box of each object in an image sample
Training Dataset: $\mathbf{X}_{\mathbf{Train}} = \{(x_1, a_1), (x_2,a_2),\ldots.(x_m, a_m)\}$
Testing Dataset: $\mathbf{X}_{\mathbf{Test}} = \{(x_1), (x_2),..(x_n)\}$

***Input:*** Read the Input image I,
C: Classes of Waste C= {Cardboard, Glass, Metal, Paper, Plastic, Trash}

B: Number of Bounding Boxes

***Output:*** Processed image along with bounding box and class of object

***Begin***
***for each*** input image I from X
$I_g$= Divide image I into n grids of equal size (s x s);
***for each*** $I_g$ in Image-grid set
Predict B bounding boxes;
Find vector $Y_i$= {$P_c$, $B_x$, $B_y$, $B_h$, $B_w$, Cardboard, Glass, Metal, Paper, Plastic, Trash};
// $Yi^{th}$ bounding box for selected grid
//Pc is the objectiveness score
***end for***
***for each*** predicted bounding box PBs **do**
***if*** (P

$$IoU = \frac{Area\ of\ the\ intersection(AB \cap PB)}{Area\ of\ the\ union\ (AB \cup PB)}$$

tual box(Ground truth box);
// Delete noninteresting PBs using Non-Max Suppression.
Non-Max Suppression (PBs);
***end if***
***end for***
***end for***
***end***

Non-maximum suppression (NMS) is a technique used in object detection to filter out redundant bounding box detections. It compares the IoU among bounding boxes and removes those with lower confidence scores. NMS ensures that only the highly confident and non-overlapping bounding boxes are retained, improving the accuracy and efficiency of object detection models. If the IoU exceeds a predefined threshold $T_{limit}$ (e.g., 0.5), remove the bounding box with the lower confidence score as it likely represents the same object.

***2. Non-Max Suppression Algorithm***

***Input:*** PBs= Predicted Bounding Boxes/Anchor Boxes
$T_{limit}$ = Threshold Limit
Pc = Objectiness Score of predicted boxes

***Output:*** Final Predicted Bounding Box for Each Object in an image

***begin***
***for each*** PBs from set of Predicted Boxes
***if*** (Pc is below Threshold)
Delete the boxes;
***end if***
***end for***
Sort PBs in decreasing order of Pcs;
$Final_{BB}$=Select First box (Highest PC score)
***for each PBs from remaining Boxes***
***repeat***
Calculate IoU of $Final_{BB}$ with PB;
***if***(IoU > $T_{limit}$)
Delete the box;
***End if***
**until** (all the boxes (PBs) have either been selected or compressed)
***end for***
***end***


**Fig.6.** Example YOLO -v4-tiny predictions for random grid images on the internet

Figure 6 presents the results obtained from the YOLOv4-tiny model applied to random grid images of the TrashNet dataset found on the internet. The outcome reveals that the model tends to learn from the background, leading to incorrect predictions. Removing the background could prove beneficial in improving the model's performance.


**Fig.7**. Results of Object Tracking and Background Learning for sample images

The outcomes achieved from the model are depicted in figure 7. The model demonstrates the capability to track composite objects by systematically lowering the confidence scores while simultaneously increasing the threshold for overlap, known as the Intersection over Union. This strategy allows the model to effectively monitor objects that are composed of multiple components.

Moreover, during the learning process, the model tends to gather information from the background, leading to instances where background elements are misidentified as objects. For example, certain cases illustrate background objects being misinterpreted as pieces of paper. Additionally, performing a grayscale transformation on highly transparent images can aid in their identification.

## 4. Interpretation of Result

The evaluation of the model has yielded remarkable results, showcasing its exceptional proficiency in object detection and classification tasks. The model achieved an impressive accuracy rate of 98.39%, indicating its high level of precision in identifying objects within images. This accuracy rate demonstrates the model's ability to recognise and classify a wide range of items included in the visual input.

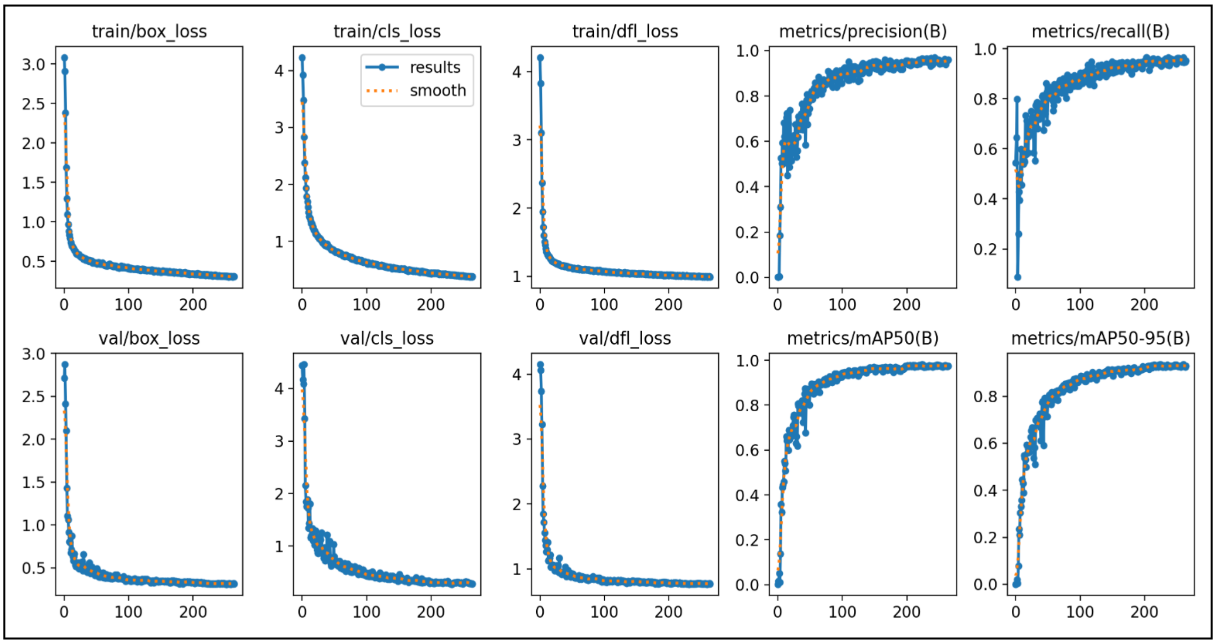


**Fig.8.** Loss and mAP results after training the YOLO tiny V4 model

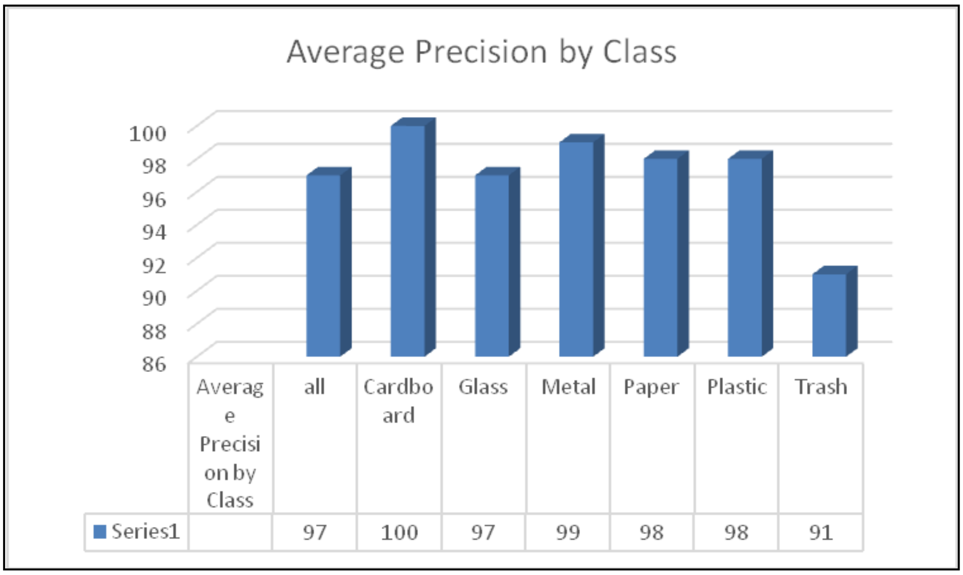


**Fig.9.** Average precision by different classes of waste

In addition to its exceptional performance metrics, the model also excels in terms of efficiency. With a detection time of merely 1 second, it showcases impressive speed in analyzing and identifying objects within images. This rapid processing capability enhances its practicality and usability in real-time applications, enabling swift and accurate detection even in time-sensitive scenarios. Figure 8 illustrates the outcomes of the training process for the YOLO Tiny V4 model, displaying the recorded Loss values and the calculated mAP. In Figure 9, the distribution of Average Precision across distinct waste classes is presented, providing insights into the model's performance for each waste category

## 5. Conclusion

In conclusion, the proposed work establishes a solid foundation for future research on garbage identification and classification. The impressive precision achieved in locating litter across various scenarios demonstrates the potential of neural networks for trash monitoring in cities or detecting illegal dumpsites in natural areas, using technologies such as drones. This breakthrough opens doors to automate environmental monitoring, enabling the measurement of the degree of pollution automatically. Inaccessible and highly contaminated areas can be efficiently cleaned with the aid of specialized robots equipped with garbage detection and classification modules, effectively reducing the cost of maintaining cleanliness in our surroundings.

The experimental results strongly support the effectiveness of the YOLOv4 and its tiny version model when applied to the waste dataset. In conclusion, the results of our investigation into the performance of Tiny YOLOv4 have demonstrated a noteworthy advantage in terms of detection speed when compared to its counterparts. The architectural optimizations and design choices made in the development of Tiny YOLOv4 have proven to be highly effective in achieving efficient object detection while maintaining a remarkable level of accuracy. The integration of robotic arms into future waste management facilities could greatly benefit from the model's capabilities, automating the sorting process and enabling the distinction between various object classes without human intervention.

## 6. References


[1] Central Pollution Control Board. (2021). Municipal Solid Waste Management in India: Annual Report 2020-21 [PDF]. Retrieved from https://cpcb.nic.in/uploads/MSW/MSW_AnnualReport_2020-21.pdf

[2] Girshick, Ross. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), edited by John Smith and Jane Doe, 580-587. IEEE, 2014.

[3] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1440-1448.

[4] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), 91-99.

[5] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), 379-387.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: single shot multibox detector," in Proceedings of the European Conference on Computer Vision (ECCV), 2016, pp. 21–37.

[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.

[8] Thung, G., & Yang, M. (2016). Classification of Trash for Recyclability Status. CS 229, Stanford University.

[9] Chen, X., Kundu, K., Zhu, Y., et al. (2017). 3D object proposals using stereo imagery for accurate object class detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(5), 1259-1272.

[10] He, Kaiming, et al. "Deep Residual Learning for Image Recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.

[11] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7310-7311.

[12] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. "Densely Connected Convolutional Networks."Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[13] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4 Tiny: A Reduced Network for Object Detection. arXiv preprint arXiv:2011.08036.

[14] Thung, G., Yang, M. (2021). TrashNet Dataset. Retrieved from https://github.com/garythung/TrashNet

[15] D-SWASTE Dataset. Retrieved from https://ieee-dataport.org/open-access/d-swaste-dataset-deep-learning-based-classification-and-segmentation-solid-waste

[16] Waste-CNN Dataset. Retrieved from https://ieee-dataport.org/open-access/waste-cnn-dataset-image-classification-solid-waste

[17] FoodCam 256 Dataset. Retrieved from http://foodcam.mobi/dataset100.html

[18] Li, X., Zhang, M., Huang, Z., Liu, J., Zhou, H., Wang, X., & Wu, J. (2021). A review on computer vision technologies for waste management. Environmental Science and Pollution Research, 28(5), 5142-5161.

[19] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection." arXiv.org, May 9, 2016. https://arxiv.org/abs/1506.02640.

[20] Joseph Redmon and Ali Farhadi. "YOLO9000: Better, Faster, Stronger." arXiv, 2016. https://arxiv.org/abs/1612.08242.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal Loss for Dense Object Detection." Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017

[22] YOLOv3: An Incremental Improvement." Https://Pjreddie.Com/. https://doi.org/1506.02640.

[23] Redmon, Joseph and Alexey Bochkovskiy. "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv, April 2020. https://arxiv.org/abs/2004.10934.

[24] Chien-Yao Wang. "You Only Learn One Representation: Unified Network for Multiple Tasks." (2021). https://doi.org/10.48550/arXiv.2105.04206.

[25] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4 Tiny: A Reduced Network for Object Detection. arXiv preprint arXiv:2011.08036.

[26] JSPM. (2023). Tiny YOLO for Trashnet Dataset [Open Source Dataset]. In Roboflow Universe. Roboflow.