

## Development of Computer Vision Algorithms for Multi-class Waste Segregation and Their Analysis

Neeraja Narayanswamy<sup>1</sup>, A. R. Abdul Rajak<sup>2\*</sup>, Shazia Hasan<sup>2</sup>

<sup>1</sup> MS-student, Department of Electrical and Computer Engineering, New York University, New York, United States.

<sup>2</sup> Assistant Professor, Department of Electronics and Communication, Birla Institute of Technology and Science, Pilani - Dubai Campus, Dubai, United Arab Emirates.

### Abstract

Classification of waste for recycling has been a focal point for scientists interested in the field of conservation of the environment. Recycling consists of numerous steps, of which one of the most crucial is the segregation of recyclables from all other waste. Due to a lack of safety standards in developing countries, waste collection is often done manually by domestic helpers, or "rag-pickers". Such a process risks individual and public health. The waste collection methods may ultimately cause waste to become non-recyclable due to cross-contamination. Literature shows that research in this direction focuses on a single class of waste detection. The proposed work investigates CNN, YOLO, and faster RCNN-based multi-class classification methods to detect different types of waste at the collecting point. The smart dustbin proposed employs these computer vision methods with a Raspberry Pi microcontroller and camera module. The experimental results for multi-class classification show that the CNN has 80% of accuracy with 60% of the loss. Whereas the YOLO algorithm shows an accuracy of 88% and a loss of 40%. But the best results were obtained from faster RCNN object detection with API, with an accuracy of 91% and a loss of 16%. There is already an existing method for making a smart dustbin, so the results are compared to show how computer vision can be used to make a smart dustbin. This shows how computer vision can be used to make a smart dustbin.

### Keywords:

Computer Vision;  
Object Detection;  
Classification;  
Recycling;  
Waste Disposal.

### Article History:

<b>Received:</b>	29	November	2021
<b>Revised:</b>	13	February	2022
<b>Accepted:</b>	02	March	2022
<b>Available online:</b>	19	April	2022

## 1- Introduction

Municipal Solid Wastes, also known as Garbage, is the clutter of unsorted waste from various sources such as homes, parks, and industries that gets accumulated in landfills or dumped into oceans. The consequences of such practices lead to pollution in the environment. Therefore, recycling is essential. Segregation of waste is the first step to recycling. One of the crucial aspects of segregation is the classification of waste at the source point. Even though segregation is a challenging task, it is vital to prevent cross-contamination [1]. There are multiple ways to classify waste, such as wet and dry, recyclable and non-recyclable, degradable and non-degradable, etc. The methods discussed in the research article aim to segregate different categories of recyclable waste through computer vision (CV). The CV leads to the reduction of health and safety risks for waste pickers and improves the recycling standards in plants. Though segregation is still a relatively recent approach, informal waste collection is predominant in developing countries. The itinerant waste buyers collect and pay for source-separated materials by various means. The household servants handle the waste without protection for their hands and faces. Therefore, automated waste segregation is crucial for segregation at source,

\* **CONTACT:** [abdulrazak@dubai.bits-pilani.ac.in](mailto:abdulrazak@dubai.bits-pilani.ac.in)

**DOI:** <http://dx.doi.org/10.28991/ESJ-2022-06-03-015>

© 2022 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<https://creativecommons.org/licenses/by/4.0/>).

as it helps to promote better public health and also increases livelihood through the improved quality of recycled materials.

Due to the growth of the urban population globally, waste generation is increasing in record numbers. However, in developing areas, for example, Varanasi city in India, only 24% of the population segregates waste for recycling [2]. Because 89% of the city's population had never received any formal education on waste segregation. People do not know about the advantages of segregation and the disadvantages of the cumulation of waste together. Hence, a problem arises since household or community bins generate waste, which falls on the local municipality to manage waste segregation as shown in Figure 1. a sample of a landfill makes all wastes generate cross-contamination. Moreover, waste collection has become an enormous task, especially in big metropolitan cities like Mumbai, which generates 9400 tons of solid waste every day [3]. This huge amount of waste pollutes the environment and ground water as well. So, it is necessary to find a solution to reduce pollution.



**Figure 1. A landfill site**

The research article aims to find optimal multiclass waste segregation methods at the source point. Three image classification algorithms are taken into consideration to classify the waste. These algorithms aim to have high accuracy and reduce computational time and the complexity of the hardware required. This research proposes building smart dustbins that use these algorithms to segregate waste at the source point.

The rest of the sections in this study are arranged as follows: Section 2 deals with a detailed literature review. Section 3 explains the methods of the proposed work, whereas Section 4 discusses the development of algorithms to detect material for waste classification. In Section 5, analysis of algorithms presents results obtained. The paper is concluded, and the future scope of the work is presented in Section 6.

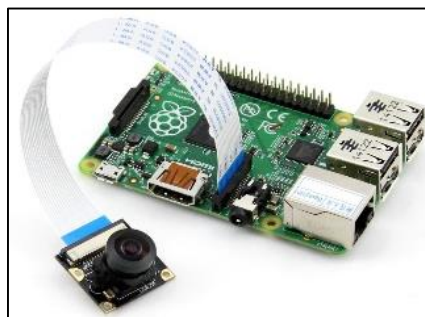
## **2- Literature Review**

Balamurugan et al. [4] described a low-power waste management system that uses cannibals to collect the trash that is decomposed daily. The GSM module and the Arduino Uno microcontroller send information regarding three levels of waste. Jain et al. [5] analysed problems concerning waste management in the world. The improper planning of waste management and the lack of technical support are the main reasons affecting citizens' health conditions. The waste management system has four models based on size, budget, route, and waste processing machines. The risk management module helps the municipal corporation of a city manage waste environmentally and economically. Sreejith et al. [6] designed a dustbin that avoids contamination of waste during rain and sends information about its level. The method mentioned is effective when solid waste disposal utilizes robotic technology. Saha et al. (2017) explained waste management strategies [7]. Such a process gives revenue. The suggested IOT technology treats the waste for animal feeding, recycling, composting, fermentation, landfills, and burning. Adam et al. [8] suggested the use of wireless sensor networks and IoT technologies to manage waste and also suggested real-time monitoring of containers and their levels.

Cristina [9] et al. analyzed waste management in two ways. The first one was the application of panel data order, and the second was the use of bootstrapped truncated regression. The results obtained show that certain local governments' political and social-economic factors increase cost-efficiency. White and Beaven [10] briefed on the LDAT landfill model. The input and output data are obtained from calculations using the full sets of data to convert conventional waste characteristics into degradation. Ferronato et al. [11] modelled waste management using different data normalizations. In this respect, economic scale is confirmed, along with the critical role of an adequate waste facility in cost minimization. Ferronato et al. [12] explained the geographic information system that enables the selective collection of municipal solid waste in developing cities. The study shows that the implementation of formal and informal recycling is the main advantage. Viau et al. [13] aimed at a life cycle assessment that needs to be critically analysed to recover recyclables from the municipal solid waste management systems. Laura et al. followed the pioneering approach [14] to obtain a global inefficiency score and an individual inefficiency score for each variable integrated into the model. The results indicated that one-third of the municipalities evaluated were eco-efficient in the provision of services. Riedewald et al. [15] explained the eco-efficiency assessment of municipal solid waste services by exogenous variables. The results obtained show that a reliable and accessible market for solid waste is available. Kumar et al. [16] estimated the generation rate of different plastic wastes by machine learning and made revenue recovery from the recycling process. Xu et al. [17] studied artificial neural networks to solve solid waste-related issues. ANN is widely used in the literature for waste generation and technological parameter prediction. Jain et al. [18] explained that heavy metal content in soil declined from landfills and claimed that solid waste was characterized by the concentration of various heavy metals. Funch et al. [19] classify glass and metal waste using convolutional neural networks. The obtained results support the CNN method for real-time waste classification. The SSD (Single Shot Multi Box Detector) method was suggested by Liu et al. [20] for detecting objects in images using a single deep neural network. The SSD method creates bounding boxes of different aspect ratios and scales for discretising the output.

### 3- Methodology

The research article aims to find the most suitable algorithm for optimal multiclass waste segregation. Three image classification algorithms are taken into consideration to classify waste. There are six most common categories of waste, such as cardboard, glass, metal, paper, plastic, and other trash. The implementation looks at accuracy, computational time, and the complexity of the hardware required. Such algorithms aim to reduce the involvement of humans in waste management and to provide safer working environments. The reduction in human effort also increases the quality and quantity of waste while segregating it. Hence, it is possible to build smart dustbins that use these algorithms to segregate waste at the source point. The smart dust bin employs the Raspberry Pi microcontroller with a camera module as shown in Figure 2 for an economical design. Such smart bins will make segregation easier for the local municipality to manage waste collection. The municipality can further transport segregated waste directly to recycling plants. The recycling plants get materials without much effort through such operations.



**Figure 2. Raspberry Pi with Camera module**

#### 3-1- Concepts of Convolutional Neural Networks

The convolutional neural network (CNN) is the class of neural networks very commonly applied to image classification [21]. The biological neural networks inspire a convolutional neural network. They are regularized multi-layer perceptrons with fully connected networks. The pattern connection in the neural network resembles the pattern of connection of neurons in the visual cortex of animals [22]. The convolutional neural network uses less pre-processing of images when compared to other image classification algorithms. The CNN consists of an input layer, an output layer, and multiple hidden layers as shown in Figure 3. The hidden layers convolve using a dot product and the Rectified Linear Activation function (*ReLU*). The input is usually an image tensor that is primarily several images  $\times$  height  $\times$  width  $\times$  depth, and the output is the image with the appropriate bounding box and label. Each neuron computes the output value by applying this activation function to the value it gets as input from the previous layer. The CNN uses pooling to reduce the dimensionality of the data by combining the outputs of the neurons of one layer into single input to the next layer.

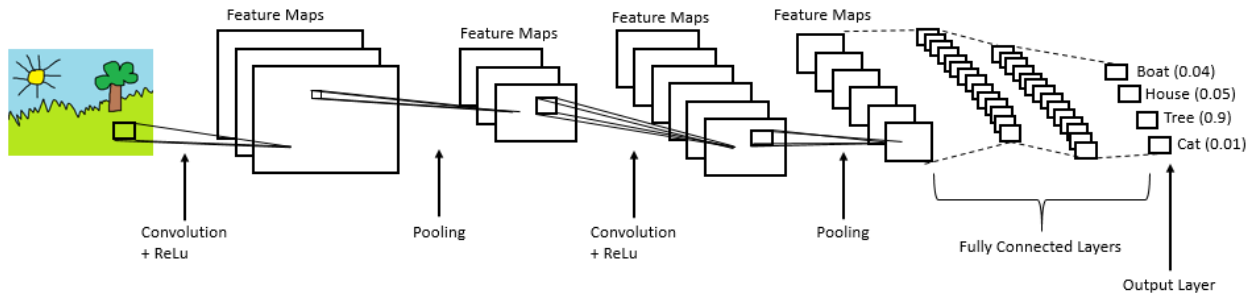


Figure 3. Convolutional neural network framework [23]

Figure 4 shows that how convolutional neural networks takes an input image, converts it into a vector of labels, and exhibits the phenomena of pooling. The initial image is divided into an 8x8 vector and then converted into a 7x5 vector by the previous layer. The current layer converts into a 2x2 vector. This process continues until the output is a n x 1 vector, where n is the number of images.

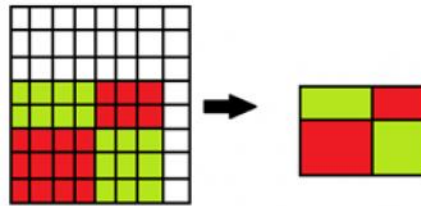


Figure 4. Pooling

When analyzing the performance [24], the Equations 1 and 2 could summarize all data and output performance in *tiles/ms* with a given layer, neurons and depth:

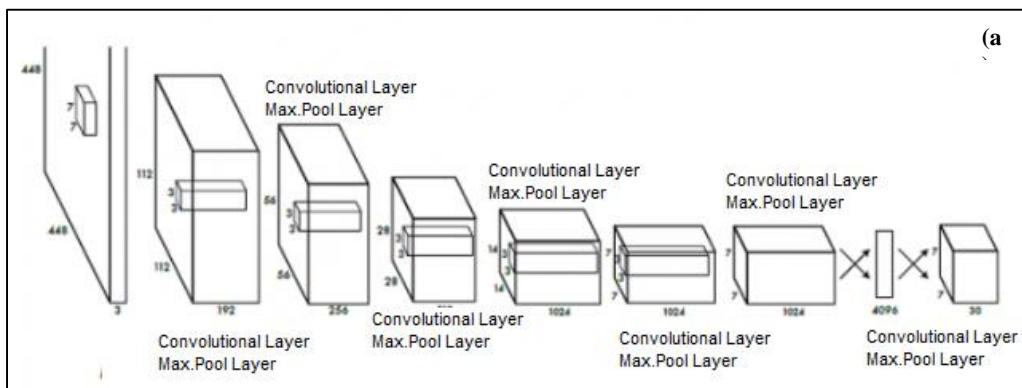
$$ms/tile = C1 * FC * Neurons + C2 * FC * Neurons + C3 * Convolutional\ layer * Depth + C4 * Convolutional\ layer * Depth + C5 * Convolutional\ layer * Depth + C6 * Convolutional\ layer * Depth + C7 \tag{1}$$

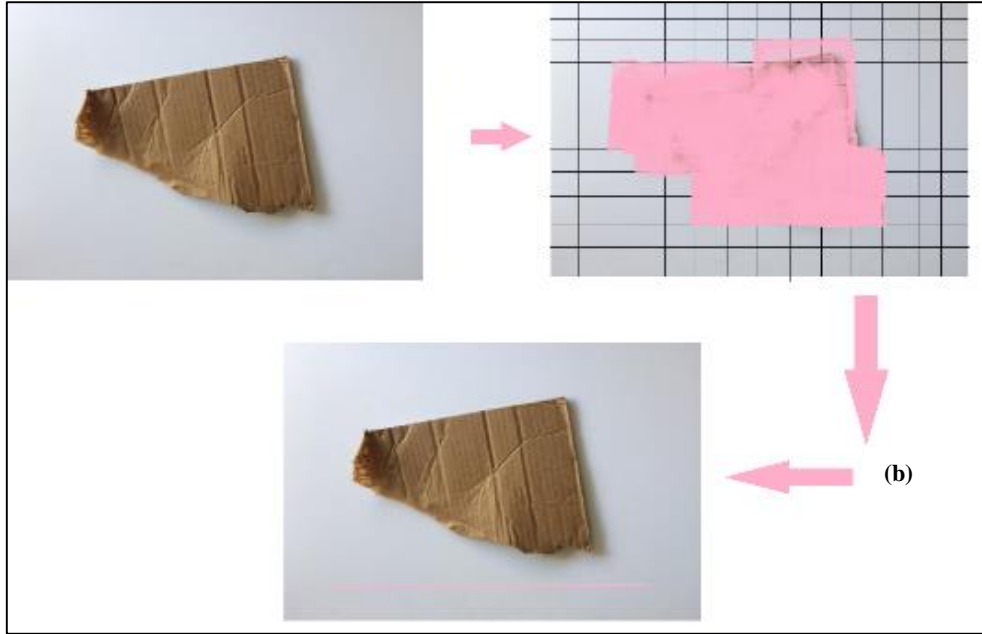
where, C1 to C7 are constants, C7 being the combined overhead for all operations:

$$ms/tile = 3.03 * 10^{-8} * Depth^2 * \sum_{n=2}^5 (tile\ size - 4n)^2 \tag{2}$$

3-2- Theory of YOLO

You Only Look Once Algorithm (YOLO) is an object detection algorithm that comes in four versions. The proposed research work employs version three. The CNN becomes the base for YOLO. CNN divides an image into an SxS grid and draws a bounding box around the parts of the image classified during training. The YOLO has twenty-four convolutional layers, followed by two fully connected layers. It reasons globally about the image when making the predictions [25, 26]. Figures 5-a and 5-b demonstrate how an image gets divided into a grid. In the grid, each box is given a number by calculating the probability. The box contains the object that can be detected. After calculating all probabilities, a bounding box is drawn to encompass the object, and then a label is given. The advantage of YOLO over CNN is that it performs object detection directly over the images rather than simultaneously predicting multiple bounding boxes and class probabilities for those boxes [27].





**Figure 5. a) The YOLO V3-Block Diagram; b) The YOLO process**

The class specific probability [28, 29] for each grid cell is defined as in Equation 3:

$$P_r(Class|Object) * P_r(Object) * IOU_{pred}^{truth} = P_r(Class) * IOU_{pred}^{truth} \quad (3)$$

The output pixel  $y$  is calculated as follows in Equation 4 to 7:

$$Sigmoid\ function = \frac{1}{1+e^{-y}} \quad (4)$$

ReLU function  $f = \max(0, y)$ :

$$Leaky\ ReLU\ function \begin{cases} y & y \geq 0 \\ 0.1y, 0 & y < 0 \end{cases} \quad (5)$$

$$Performance = \min \left\{ \begin{array}{l} Operation\ intensity \times Bandwidth \\ Peak\ Performance \end{array} \right\}$$

$$Peak\ Performance = \frac{Total\ number\ of\ operations}{Total\ amount\ of\ Execution\ cycles} = \frac{2xRxCxMxNxKxK}{\frac{M}{T_m} \times \frac{N}{T_n} \times \frac{R}{T_r} \times \frac{C}{T_c} \times (T_m \times T_n \times K \times K + T)} \quad (6)$$

$$Operation\ Intensity = \frac{Total\ number\ of\ operations}{Total\ amount\ of\ external\ data\ access} = \frac{2xRxCxMxNxKxK}{T_{in} \times DS_{in} \times T_{weight} \times DS_{weight} \times (T_{out} \times DS_{out})} \quad (7)$$

where  $T$  represents time of data transmission,  $2xRxCxMxNxKxK$  are operations,  $T_{in}$ ,  $T_{weight}$  and  $T_{out}$  are trips counts and  $DS_{in}$ ,  $DS_{weight}$  and  $DS_{out}$  data block sizes. The Loss function is expressed in Equation 8:

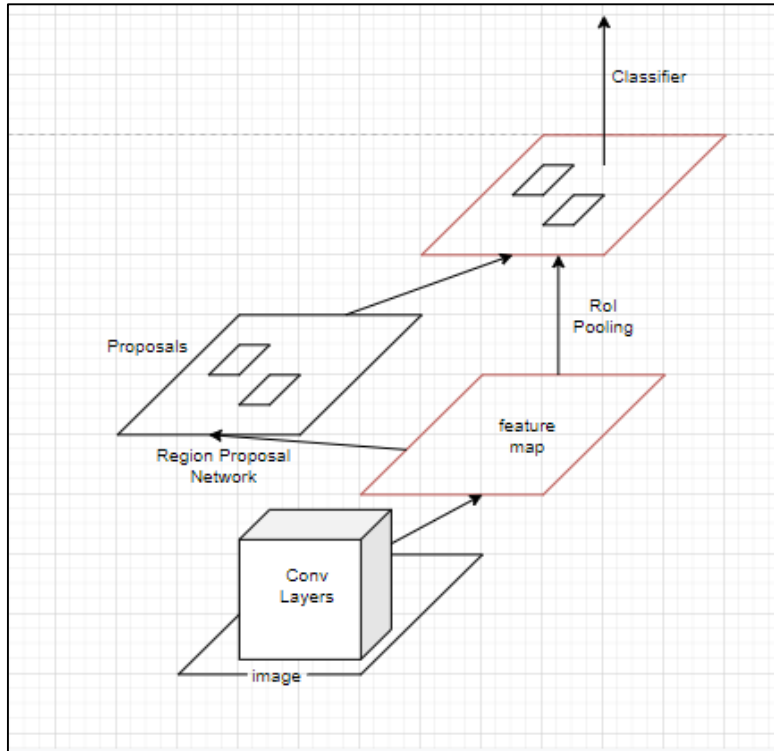
$$Loss = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^A I_{ij}^{obj} [(b_{xi} - b_{\hat{x}i})^2 + (b_{yi} - b_{\hat{y}i})^2] + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^A I_{ij}^{obj} [(\sqrt{b_{wi}} - \sqrt{b_{\hat{w}i}})^2 + (\sqrt{b_{hi}} - \sqrt{b_{\hat{h}i}})^2] + \sum_{i=0}^{s^2} \sum_{j=0}^A I_{ij}^{obj} [(C_i - \hat{C}_i)^2] + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^A I_{ij}^{noobj} [(C_i - \hat{C}_i)^2] + \sum_{i=0}^{s^2} \sum_{j=0}^A I_{ij}^{obj} + \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \quad (8)$$

where,  $b_x$  and  $b_y$  are variables refer to centre of prediction,  $b_w$ ,  $b_h$  are dimensions of bounding boxes,  $\lambda_{coord}$ ,  $\lambda_{noobj}$  are to increase emphasis,  $C$  refers to confidence( $c$ ) is classification of prediction,  $I_{ij}^{obj}$  is  $j$ th bounding box in the  $I$  th cell,  $I_{ij}^{noobj}$  is the  $I$  th cell.

### 3-3- Theory of Faster RCNN

Faster RCNN is a Region based Convolutional Neural Network (RCNN) used for real time object detection. It is a search region proposal network. The RCNN uses anchors on images to centralize bounding boxes around the image and it detects potential regions of the target object and eliminates them by probability. Faster RCNN becomes useful because both region proposal and object detection are made simultaneously [30]. Such action increases the speed. Hence the algorithm delivers results faster. Region Proposal Networks (RPN) based on CNN is used instead of RPN based on

Selective Search which was used by its predecessors RCNN and Fast RCNN [31]. More over the detection network also uses CNN. While conducting region proposals, the RPN [32, 33] uses a sliding window to specify each location on the feature map. These regions use anchor boxes, which are dependent on the scales and aspect ratios to generate region proposals. The subsequent two layers identify an object in those regions and the bounding box needed for those objects. After the RPN returns its results, CNN then classifies the object detected by the RPN. Figure 6 displays the working of the Faster RCNN as explained above. Its loss function is like YOLO, but the model is more accurate as total loss eventually plateaus with each iteration.



**Figure 6. The working of a Faster RCNN**

The distribution probability of the anchor on Image I can be expressed by Equation 9:

$$p(x, y, w, h|I) = p(x, y|I)p(w, h|x, y, I) \quad (9)$$

where  $x, y$  are the coordinate of the image and  $w$  and  $H$  are width and height of the image.

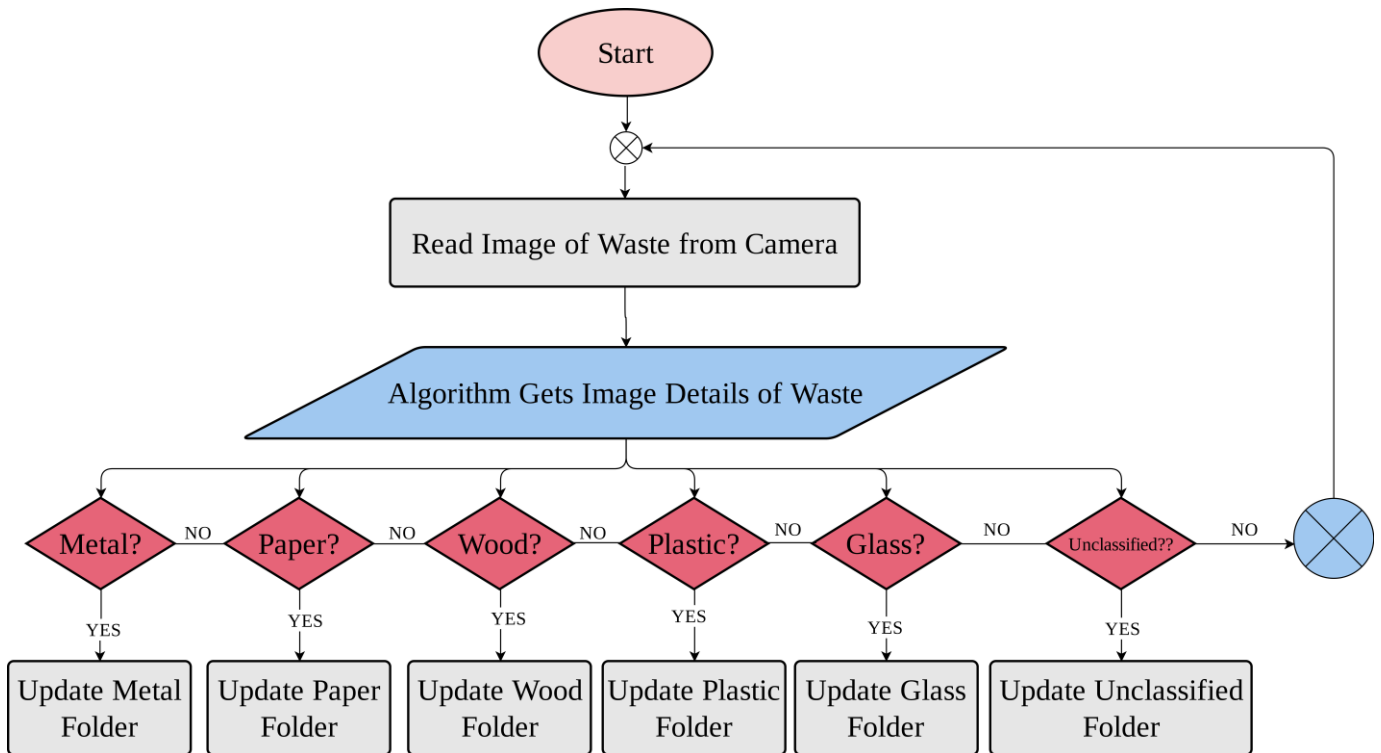
The loss function is given in Equation 10 as:

$$L = \frac{1}{N_c} \sum_k L_k (P_k P_k^*) \frac{\lambda}{N_r} P_k^* L_t(t_k, t_k^*) \quad (10)$$

where;  $L$ ,  $L_c$ , and  $L_t$  are the joint loss, classification loss and regression loss,  $N_c$  and  $N_r$  are number of categories,  $\lambda$  and  $k$  represent the weight coefficient,  $p_k$  and  $p_k^*$  is the probability that box  $k$  is the object,  $t_k$  is the predicted offset box  $t_k^*$  is the offset between the anchor box and actual box.

#### 4- Development of Algorithms to Detect Material for Waste Classification

In order to develop algorithm for its implementation level, simulations were run on a machine with Windows 10 as the operating system. The programs for simulation were written in *Python 3.7* using *Keras* and *TensorFlow 2.0* to create, train and test the models. Microsoft *VoTT* (Visual Object Tagging Tool) is used to label the images. Figure 7 shows how algorithm implementation classifies the waste by flowchart.



**Figure 7. Flowchart of implementation**

#### 4-1- Implementing CNN Model for Waste classification

Following steps shows implementation of CNN model:

- **Step 1.** All the images are saved first in separate folders according to classification. For example, a cardboard image goes into the cardboard folder.
- **Step 2.** Then the image paths are converted into an array called image lists.
- **Step 3.** The images under test are augmented using this array in batches. For computers with low processing power, the reduction of batch size avoids resource exhaustion error.
- **Step 4.** Label folder with training images.
- **Step 5.** The machine runs the model, the time taken to complete the process depends on the processing power of the machine.
- **Step 6.** The detector gets the results after completing its training.

#### 4-2- Implementing the “You Only Look Once” Object Detection Model for waste classification

The series of steps to be followed for implementing algorithm in YOLO.

- **Step 1.** Images get splitted and stored in two folders having names as test and train.
- **Step 2.** Then, the images are annotated in the train folder using VoTT which gives an output in the form of a CSV File.
- **Step 3.** This CSV file creates a text file that is readable by the training program. YOLO Algorithm gets trained by the text file which contains the dimension of bounding boxes
- **Step 4.** Then the weights are downloaded for training code that uses the darknet weights.
- **Step 5.** Then the Object Detector is trained as indicated in Figure 8.
- **Step 6.** After Training is complete, the code is run for labelling the images based on a trained dataset on the images in the Test folder as revealed in Figure 9.
- **Step 7.** After all the images in the test dataset are labelled, new images with the bounding box are generated and stored in the results folder.

```

Epoch 41/51
128/128 [=====] - 4039s 32s/step - loss: 14.4774 - val_loss: 15.0259
Epoch 42/51
128/128 [=====] - 4044s 32s/step - loss: 14.4297 - val_loss: 14.6007
Epoch 43/51
128/128 [=====] - 4043s 32s/step - loss: 14.4993 - val_loss: 14.8587
Epoch 44/51
128/128 [=====] - 4034s 32s/step - loss: 14.4246 - val_loss: 14.8670
Epoch 45/51
128/128 [=====] - 4038s 32s/step - loss: 14.3803 - val_loss: 14.3563
Epoch 46/51
128/128 [=====] - 4030s 31s/step - loss: 14.4793 - val_loss: 14.9387
Epoch 47/51
128/128 [=====] - 3944s 31s/step - loss: 14.4275 - val_loss: 14.4461
Epoch 48/51
128/128 [=====] - 3894s 30s/step - loss: 14.4135 - val_loss: 14.7571
Epoch 49/51
128/128 [=====] - 3889s 30s/step - loss: 14.3958 - val_loss: 14.6579
Epoch 50/51
128/128 [=====] - 3891s 30s/step - loss: 14.3017 - val_loss: 14.6626
Epoch 51/51
128/128 [=====] - 3897s 30s/step - loss: 14.2782 - val_loss: 14.2667
Unfreeze all layers.
Train on 2048 samples, val on 227 samples, with batch size 4.
Epoch 52/102
512/512 [=====] - 8957s 17s/step - loss: 14.4679 - val_loss: 14.2909
Epoch 53/102
512/512 [=====] - 8970s 18s/step - loss: 13.4543 - val_loss: 14.0281
Epoch 54/102
512/512 [=====] - 8997s 18s/step - loss: 13.1005 - val_loss: 14.7199
Epoch 55/102
512/512 [=====] - 10890s 21s/step - loss: 12.9375 - val_loss: 13.0157
Epoch 56/102
512/512 [=====] - 9007s 18s/step - loss: 12.6304 - val_loss: 12.9093
Epoch 57/102
512/512 [=====] - 8988s 18s/step - loss: 12.3300 - val_loss: 13.0315
Epoch 58/102
512/512 [=====] - 8963s 18s/step - loss: 12.2160 - val_loss: 11.9847
Epoch 59/102
512/512 [=====] - 8836s 17s/step - loss: 11.9511 - val_loss: 11.7786
Epoch 60/102
512/512 [=====] - 8820s 17s/step - loss: 11.6763 - val_loss: 12.0900
Epoch 61/102
512/512 [=====] - 8825s 17s/step - loss: 11.3216 - val_loss: 11.2808
Epoch 62/102
512/512 [=====] - 8904s 17s/step - loss: 11.1115 - val_loss: 11.8973
Epoch 63/102
512/512 [=====] - 10990s 21s/step - loss: 10.9005 - val_loss: 10.9549
Epoch 64/102
445/512 [=====>.....] - ETA: 19:56 - loss: 10.7124

```

Figure 8. Image of YOLO training process

```

Found 1 boxes for img
Cardboard 1.00 (0, 1) (512, 384)
Time spent: 3.389sec
C:\Users\neera\TrainYourOwnYOLO\Data\Source_Images\Test_Images\cardboard365.jpg
(416, 416, 3)
Found 1 boxes for img
Cardboard 1.00 (2, 0) (511, 384)
Time spent: 1.516sec
C:\Users\neera\TrainYourOwnYOLO\Data\Source_Images\Test_Images\cardboard366.jpg
(416, 416, 3)
Found 1 boxes for img
Cardboard 1.00 (11, 0) (511, 384)
Time spent: 1.466sec
C:\Users\neera\TrainYourOwnYOLO\Data\Source_Images\Test_Images\cardboard367.jpg
(416, 416, 3)
Found 1 boxes for img
Cardboard 1.00 (3, 7) (509, 377)
Time spent: 1.448sec
C:\Users\neera\TrainYourOwnYOLO\Data\Source_Images\Test_Images\cardboard368.jpg
(416, 416, 3)
Found 1 boxes for img
Cardboard 1.00 (0, 2) (512, 384)
Time spent: 1.452sec
C:\Users\neera\TrainYourOwnYOLO\Data\Source_Images\Test_Images\cardboard369.jpg
(416, 416, 3)
Found 1 boxes for img
Cardboard 1.00 (8, 15) (512, 372)
Time spent: 1.437sec
C:\Users\neera\TrainYourOwnYOLO\Data\Source_Images\Test_Images\cardboard370.jpg
(416, 416, 3)
Found 1 boxes for img
Cardboard 0.98 (0, 0) (512, 382)
Time spent: 1.432sec

```

Figure 9. Image of Test folder created



### 4-3- Implementing the Faster RCNN for Training and Classification

The following steps explain algorithmic ways for the implementation of faster RCNN:

- **Step 1.** Both the training and the testing images get annotated using labelling, resulting in an XML file as output
- **Step 2.** The information has two CSV files, like the one made for the YOLO algorithm.
- **Step 3.** These files generate tf. records and maps to the config file.
- **Step 4.** The config file gives details like input and output directory, learning rate, and neural network structure.
- **Step 5.** Then the training is done using config file and the iterations are executed as conveyed in Figure 10.
- **Step 6.** The training is stopped, and the inference graph is frozen.
- **Step 7.** Then the trained model is run through test images, and the output is evaluated.

```

Anaconda Prompt (Miniconda3) - python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_pets.co...
I1215 11:20:38.001220 14452 learning.py:507] global step 15871: loss = 0.1588 (5.357 sec/step)
I1215 11:20:42.490267 14452 learning.py:507] global step 15872: loss = 0.1318 (4.487 sec/step)
I1215 11:20:46.970289 14452 learning.py:507] global step 15873: loss = 0.2272 (4.480 sec/step)
I1215 11:20:51.518131 14452 learning.py:507] global step 15874: loss = 0.2234 (4.546 sec/step)
I1215 11:20:56.930663 14452 learning.py:507] global step 15875: loss = 0.5723 (5.411 sec/step)
I1215 11:21:01.489569 14452 learning.py:507] global step 15876: loss = 0.1851 (4.559 sec/step)
I1215 11:21:05.940256 14452 learning.py:507] global step 15877: loss = 0.2495 (4.450 sec/step)
I1215 11:21:10.403425 14452 learning.py:507] global step 15878: loss = 0.2747 (4.448 sec/step)
I1215 11:21:14.762413 14452 learning.py:507] global step 15879: loss = 0.3445 (4.357 sec/step)
I1215 11:21:19.234840 14452 learning.py:507] global step 15880: loss = 0.3632 (4.471 sec/step)
I1215 11:21:23.956217 14452 learning.py:507] global step 15881: loss = 0.2002 (4.719 sec/step)
I1215 11:21:29.009707 14452 learning.py:507] global step 15882: loss = 0.1658 (5.052 sec/step)
I1215 11:21:34.261666 14452 learning.py:507] global step 15883: loss = 0.5017 (5.251 sec/step)
I1215 11:21:39.038761 14452 learning.py:507] global step 15884: loss = 0.2917 (4.775 sec/step)
I1215 11:21:44.297703 14452 learning.py:507] global step 15885: loss = 0.0201 (5.258 sec/step)
I1215 11:21:48.969137 14452 learning.py:507] global step 15886: loss = 0.0232 (4.669 sec/step)
I1215 11:21:54.002682 14452 learning.py:507] global step 15887: loss = 0.4244 (5.032 sec/step)
I1215 11:22:00.436482 14452 learning.py:507] global step 15888: loss = 0.4229 (6.433 sec/step)
I1215 11:22:05.649931 14452 learning.py:507] global step 15889: loss = 0.0209 (5.211 sec/step)
I1215 11:22:10.601526 14452 learning.py:507] global step 15890: loss = 0.3701 (4.950 sec/step)
I1215 11:22:15.785390 14452 learning.py:507] global step 15891: loss = 0.3794 (5.183 sec/step)
I1215 11:22:20.758916 14452 learning.py:507] global step 15892: loss = 0.2400 (4.973 sec/step)
I1215 11:22:26.789793 14452 learning.py:507] global step 15893: loss = 0.5118 (6.029 sec/step)
I1215 11:22:32.093617 14452 learning.py:507] global step 15894: loss = 0.1675 (5.300 sec/step)
I1215 11:22:37.877506 16644 supervisor.py:1050] Recording summary at step 15894.
I1215 11:22:38.200640 14452 learning.py:507] global step 15895: loss = 0.3035 (6.107 sec/step)
I1215 11:22:43.544002 14452 learning.py:507] global step 15896: loss = 0.1114 (5.341 sec/step)
I1215 11:22:48.576549 14452 learning.py:507] global step 15897: loss = 0.0942 (5.033 sec/step)
I1215 11:22:53.328844 14452 learning.py:507] global step 15898: loss = 0.0671 (4.751 sec/step)

```

Figure 10. Image of Training by Iterations

## 5- Analysis of Algorithms

CNN, YOLO and RCNN algorithms get images of waste for their processing. These three algorithms get trained for classifying waste. Once identification and classification of waste are done, the image file of each waste is stored in appropriate folder.

### 5-1- Results and Discussion of CNN

Figure 11 expresses classification of the images by CNN and their probability graphs. The probability graph indicates the possible classification of images. The glass gets classified with maximum probability. In most of the graphs, it is seen that the algorithm predicts only one class. However, the algorithm predicts glass with two other classes metal and plastic because of their similarities between current image and the images used to train the network to identify metal and plastic. The network was trained on a total of 2500 training images with a train/test split of 90/10. During training, a batch size of 32 is taken initially. However, due to the system's capacity under training the network, the batch size was reduced to 8.

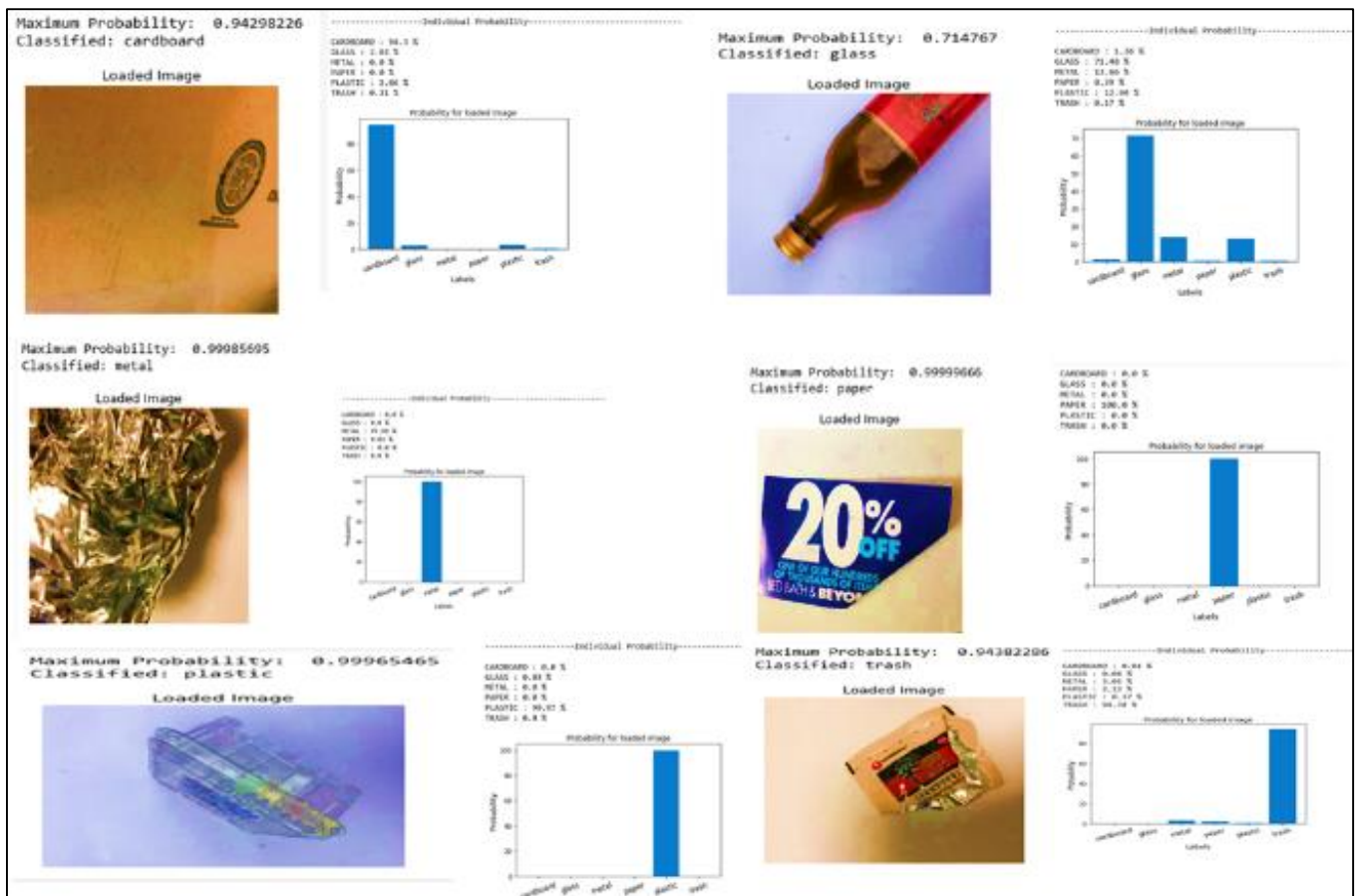


Figure 11. Classified Images and their Probability Graphs

The graph in Figure 12 displays the accuracy when the network is trained. It is observed that the accuracy goes through small fluctuations initially, but overall, it follows a trend of eventually increasing until the final accuracy of 80% is obtained.



Figure 12. Accuracy Graph for CNN

Figure 13 represents the overall loss of CNN while training and looks like the accuracy graph, as shown in Figure 12, where the loss keeps fluctuating per iteration, but the overall performance of loss decreases. However, towards the 100th epoch, there is an increase in the difference between training and validation loss. This difference indicates that the model is moving towards over fitting and measures are needed to prevent over fitting in future runs.



Figure 13. Loss graph for CNN

### 5-2- Results and Discussions of YOLO

The timing details of YOLO network to finish training are monitored. Initially, like the CNN algorithm, the batch size had to be reduced from 32 to 8 due to the computer system's capabilities. The network initially trained over 51 epochs with a specific learning rate. Again, the network gets trained with another 49 epochs at a reduced learning rate. This method is adopted in order to prevent overfitting by reducing the learning rate when the loss was plateauing. After the training was over, the algorithm was tested for over two hundred images. The results showed an accuracy of 88%. Thus, although the algorithm took more time than the simple CNN for training, it has better accuracy.

Figure 14 displays six tested images among two hundred images. The accuracy shows that YOLO algorithm is better than the CNN algorithm. The algorithm also is better in classifying multiple wastes in the same frame. Such method of classifying waste proves that YOLO will be a better algorithm for rovers going into landfills and conducting waste segregation.



Figure 14. Objects detected by using the YOLO Algorithm

Figure 15 proclaims the loss of YOLO over ninety iterations. The first ten iterations are not shown in the above graph as the loss was high, and therefore, this high loss would hide the tiny fluctuations in the loss for the next iterations. Similar to CNN, there are small fluctuations in the validation loss but overall loss tends to decrease. However, unlike CNN, there are no fluctuations in the training loss. The difference between training loss and validation loss is constant which indicates that the network is not overfitting over the images. However, the loss from YOLO is much higher than the loss due to CNN because YOLO requires more training images to classify the images into six classes. Further test results in the same trend where images are predicted more accurately for some classes than others. There are more similar images available in one class than the other class as shown in the graph.

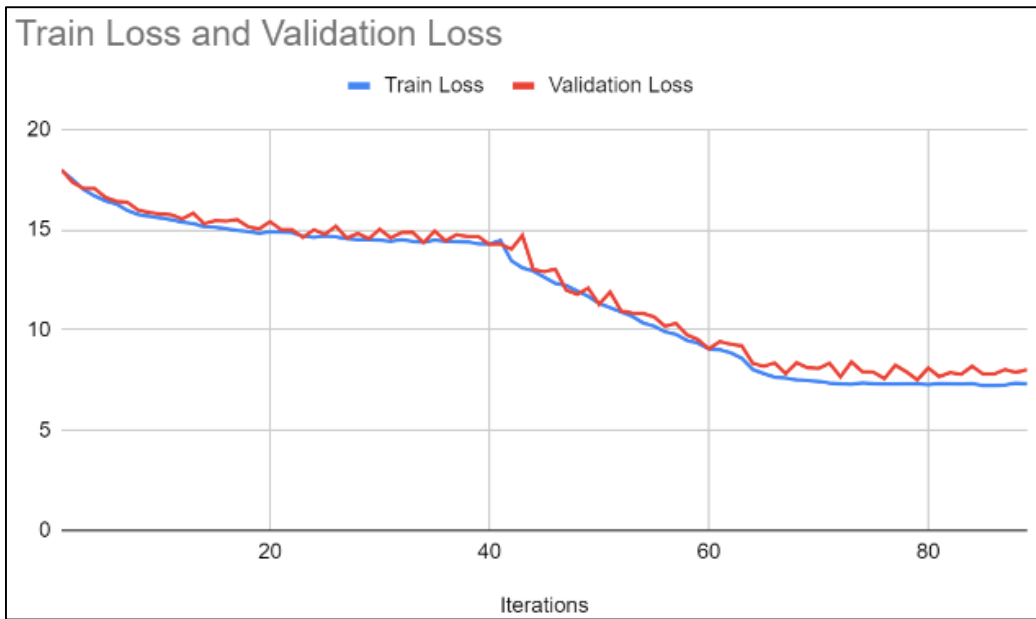


Figure 15. Loss Graph for YOLO

Figure 16 exhibits the percentage of images correctly classified for each class and having 252 images. The classifier correctly classifies cardboard and metal with 88% accuracy. The paper and the plastic get classified with least accuracy due to plenty of paper images are similar to cardboard, and many plastic images show some resemblance to glass. To mitigate this problem, an extensive dataset of images is needed, which can be easily obtained by local municipal authorities.

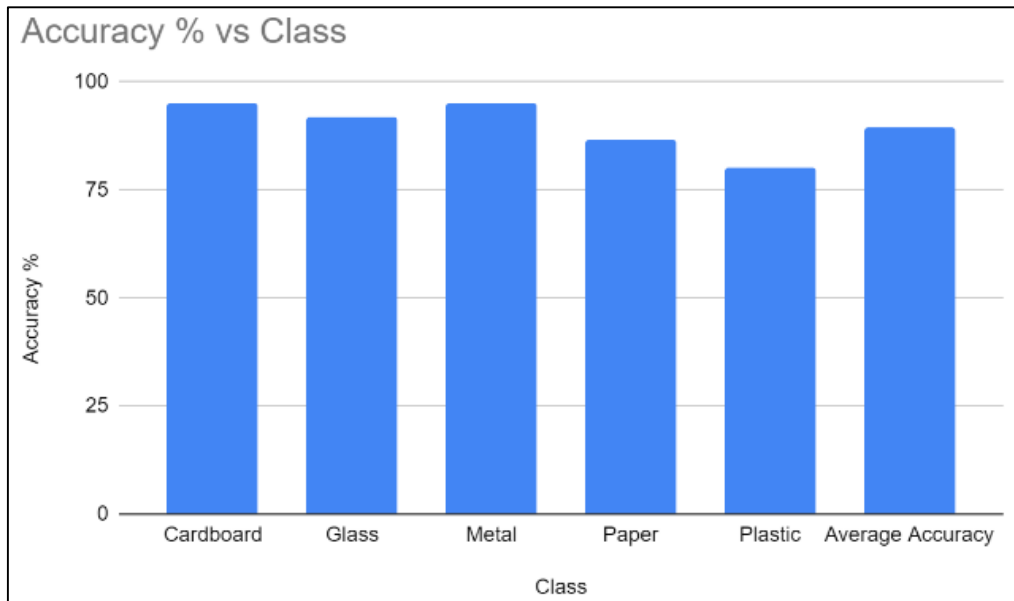


Figure 16. Accuracy graph for each of the six classes

### 5-3- Results and Discussions of Faster RCNN

Along with faster RCNN, *TensorFlow* employs Application Programming Interface (API) for object detection. This API helps to implement various object detection algorithms. For example, for those municipal corporations that cannot afford high end equipment for every dustbin in the city, lower cost equipment for object detection mechanism [9] can be used. The iterations in number 16000 were performed for training. Then the model runs through the testing images. The obtained results showed an accuracy of 91%. The loss by execution of the model was around 0.16 while making the best classifier so far. However, this algorithm requires high power and an extensive amount of data to work. Initially, the loss was oscillating from almost zero to exponential values. However, the learning rate was reduced after 2500 iterations, and loss decreased to 16 % after 16000 iterations, as shown in Figure 17.

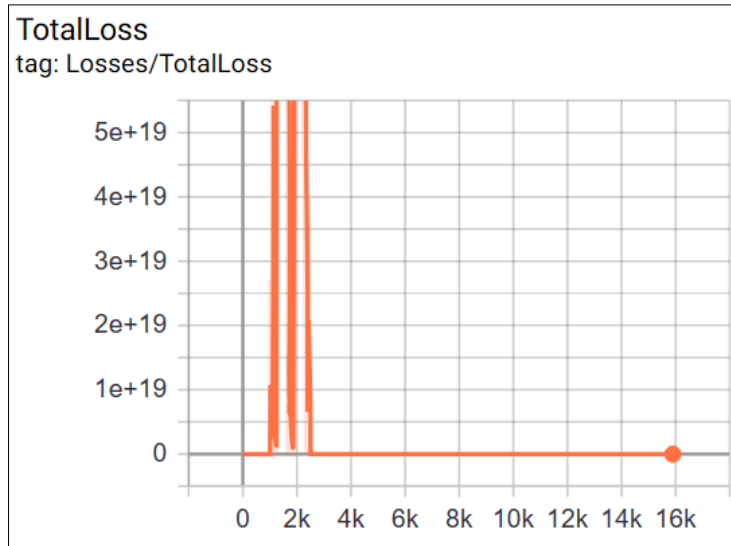


Figure 17. Total Loss

The training gets stopped after the loss metric was found as shown inference graph in Figure 17 python trains these objects to recognise them from various photographs of common waste.

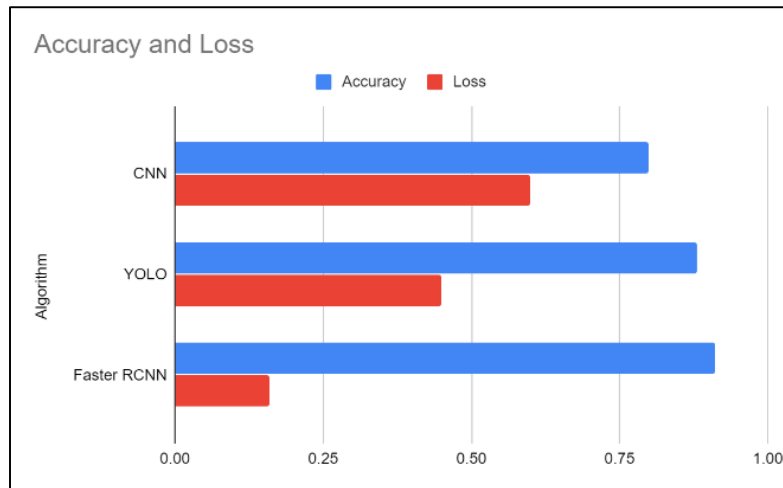
Figure 18 demonstrate that these algorithms are accurate enough to classify wastes. They can be run even on live video footage, which would be used in smart dustbins.



Figure 18. Correctly Classified Junk by Faster R-CNN

## 6- Conclusion

Figure 19 compares the accuracy and loss of various models used to train smart dustbins, and their performance values are available in Table 1. CNN shows 80% accuracy when classifying waste. At the same time, the loss was found to be 60%. The accuracy of 88% and loss of 40% are exhibited by the YOLO algorithm. The faster RCNN object detection with API generates an accuracy of 91% and the loss is 16%. It is seen that the faster-RCNN object detection model has the highest accuracy, and loss is less. *TensorFlow* object detection via the API can also be used with other models, such as low-resolution mobile cameras.



**Figure 19. Comparison of CNN vs. YOLO vs. Faster RCNN**

**Table 1. Comparison of performance of CNN vs. YOLO vs. Faster RCNN**

	CNN	YOLO	RCNN	SSD
<b>Accuracy</b>	80%	88%	91%	63.6%
<b>Loss</b>	60%	40%	16%	22%

CNN does not require annotation and has the fastest training time. However, it has the lowest accuracy and maximum loss. YOLO takes more time and requires annotation, but is more accurate than CNN and can be used on systems with less power. Faster RCNN has the highest accuracy and the most negligible loss. However, it requires the most computational power and an extensive dataset. Therefore, it is the most computationally complex algorithm. *TensorFlow* object detection using the API provides a great infrastructure that can host a variety of models for the segregation of waste. It provides scripts in which trained models can be used in image and live video feeds, making it the best infrastructure for the smart dustbin. Therefore, depending on the budget and the capability of existing systems, the municipal corporation can design a dustbin by employing any one of several algorithms. Since computer vision is employed to identify and classify waste, it becomes a strength of this research if a proper image of the waste is analysed. The limitation of research is that images of waste should be the same as those of stored images. If the shape of waste is changed and the same images are processed, it leads to erroneous results.

### 6-1- Further Research

The PC system with high configuration employs these algorithms to classify the waste. These algorithms can also build a smart dustbin using a microcontroller with a camera model so that its cost is lower and more affordable. Research on the combination of two or more computer vision algorithms may be implemented to increase accuracy. Furthermore, the dataset used for this study is very limited in terms of the shapes and materials of waste available. Therefore, testing with an extensive dataset is crucial. Lastly, our team wishes to combine the CV algorithm tested here with the sensors currently available on the market to continue our research further to make an economically viable smart dustbin.

## 7- Declarations

### 7-1- Author Contributions

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

### 7-2- Data Availability Statement

Data sharing is not applicable to this article.

### 7-3- Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

#### 7-4- Acknowledgements

The authors would like to thank the Director, Birla Institute of Technology and Science Pilani, Dubai campus. This research work was supported by the innovation laboratory at Birla Institute of Technology and Science Pilani, Dubai campus.

#### 7-5- Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

#### 8- References

- [1] Wilson, D. C., Rodic, L., Modak, P., Soos, R., Carpintero, A., Velis, K., Iyer, M. & Simonett, O. (2015). Global waste management outlook. United Nations Environment Programme (UNEP), Nairobi, Kenya.
- [2] Istrate, I.-R., Iribarren, D., Gálvez-Martos, J.-L., & Dufour, J. (2020). Review of life-cycle environmental consequences of waste-to-energy solutions on the municipal solid waste management system. *Resources, Conservation and Recycling*, 157, 104778. doi:10.1016/j.resconrec.2020.104778.
- [3] Anjali Lukose. (2015). Dirty Mumbai: 6,400 tonnes of solid waste, 40 pc sewage go untreated, *The Indian Express*, Mumbai. Available online: <http://indianexpress.com/article/cities/mumbai/dirty-mumbai-6400-tonnes-of-solid-waste-40-pc-sewage-go-untreated/> (accessed on January 2022).
- [4] Balamurugan, S., Ajithx, A., Ratnakaran, S., Balaji, S., & Marimuthu, R. (2017). Design of smart waste management system. 2017 International Conference on Microelectronic Devices, Circuits and Systems, ICMDCS 2017, Vellore, India, 1–4. doi:10.1109/ICMDCS.2017.8211709.
- [5] Jain, R., Garg, S., Agrawal, T., Gangal, S., Chawla, I., & Jain, S. (2018). Sustainable Waste Management Model. 2018 11th International Conference on Contemporary Computing, IC3 2018, Noida, India, 1–6. doi:10.1109/IC3.2018.8530544.
- [6] Sreejith, S., Ramya, R., Roja, R., & Sanjay Kumar, A. (2019). Smart Bin for Waste Management System. 2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019, Coimbatore, India, 1079–1082. doi:10.1109/ICACCS.2019.8728531.
- [7] Saha, H. N., Auddy, S., Pal, S., Kumar, S., Pandey, S., Singh, R., Singh, A. K., Banerjee, S., Ghosh, D., & Saha, S. (2017). Waste management using Internet of Things (IoT). 2017 8th Industrial Automation and Electromechanical Engineering Conference, IEMECON 2017, Bangkok, Thailand, 359–363. doi:10.1109/IEMECON.2017.8079623.
- [8] Adam, M., Okasha, M. E., Tawfeeq, O. M., Margan, M. A., & Nasreldeen, B. (2018). Waste Management System Using IoT. 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering, ICCCEEE 2018, Khartoum, Sudan, 1–4. doi:10.1109/ICCCEEE.2018.8515871.
- [9] Campos-Alba, C. M., Garrido-Rodríguez, J. C., Plata-Díaz, A. M., & Pérez-López, G. (2021). The selective collection of municipal solid waste and other factors determining cost efficiency. An analysis of service provision by Spanish municipalities. *Waste Management*, 134, 11–20. doi:10.1016/j.wasman.2021.07.039.
- [10] White, J. K., & Beaven, R. P. (2013). Developments to a landfill processes model following its application to two landfill modelling challenges. *Waste Management*, 33(10), 1969–1981. doi:10.1016/j.wasman.2012.12.006.
- [11] Di Foggia, G., & Beccarello, M. (2020). Drivers of municipal solid waste management cost based on cost models inherent to sorted and unsorted waste. *Waste Management*, 114, 202–214. doi:10.1016/j.wasman.2020.07.012.
- [12] Ferronato, N., Preziosi, G., Gorrity Portillo, M. A., Guisbert Lizarazu, E. G., & Torretta, V. (2020). Assessment of municipal solid waste selective collection scenarios with geographic information systems in Bolivia. *Waste Management*, 102, 919–931. doi:10.1016/j.wasman.2019.12.010.
- [13] Viau, S., Majeau-Bettez, G., Spreutels, L., Legros, R., Margni, M., & Samson, R. (2020). Substitution modelling in life cycle assessment of municipal solid waste management. *Waste Management*, 102, 795–803. doi:10.1016/j.wasman.2019.11.042.
- [14] Delgado-Antequera, L., Gémar, G., Molinos-Senante, M., Gómez, T., Caballero, R., & Sala-Garrido, R. (2021). Eco-efficiency assessment of municipal solid waste services: Influence of exogenous variables. *Waste Management*, 130, 136–146. doi:10.1016/j.wasman.2021.05.022.
- [15] Riedewald, F., Patel, Y., Wilson, E., Santos, S., & Sousa-Gallagher, M. (2021). Economic assessment of a 40,000 t/y mixed plastic waste pyrolysis plant using direct heat treatment with molten metal: A case study of a plant located in Belgium. *Waste Management*, 120, 698–707. doi:10.1016/j.wasman.2020.10.039.

- [16] Kumar, A., Samadder, S. R., Kumar, N., & Singh, C. (2018). Estimation of the generation rate of different types of plastic wastes and possible revenue recovery from informal recycling. *Waste Management*, 79, 781–790. doi:10.1016/j.wasman.2018.08.045.
- [17] Xu, A., Chang, H., Xu, Y., Li, R., Li, X., & Zhao, Y. (2021). Applying artificial neural networks (ANNs) to solve solid waste-related issues: A critical review. *Waste Management*, 124, 385–402. doi:10.1016/j.wasman.2021.02.029.
- [18] Jain, P., Kim, H., & Townsend, T. G. (2005). Heavy metal content in soil reclaimed from a municipal solid waste landfill. *Waste Management*, 25(1), 25-35. doi:10.1016/j.wasman.2004.08.009.
- [19] Funch, O. I., Marhaug, R., Kohtala, S., & Steinert, M. (2021). Detecting glass and metal in consumer trash bags during waste collection using convolutional neural networks. *Waste Management*, 119, 30–38. doi:10.1016/j.wasman.2020.09.032.
- [20] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS, Amsterdam, Netherland, 21–37. doi:10.1007/978-3-319-46448-0\_2.
- [21] Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V., & Chervyakov, N. I. (2020). Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177, 232–243. doi:10.1016/j.matcom.2020.04.031.
- [22] Fukushima, K. (2007). Noncognition. *Scholarpedia*, 2(1), 1717, Kansai University, Osaka, Japan. doi:10.4249/1717.
- [23] Prabhu, R. (2018). Understanding of Convolutional Neural Network (CNN) Deep Learning. *Medium.Com*, 1–11. Available online: <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148> (accessed on January 2022).
- [24] Mattsson, N. (2016). Classification Performance of Convolutional Neural Networks. *Uppsala University Publications*, Sweden. Available online: <http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-305342> (accessed on January 2022).
- [25] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Las Vegas, United States, 779–788. doi:10.1109/CVPR.2016.91.
- [26] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In F. Better & Stronger (Eds.), *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Honolulu, United States, (pp. 6517–6525). doi:10.1109/CVPR.2017.690.
- [27] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. doi:10.1109/TPAMI.2016.2577031.
- [28] Huang, H., Liu, Z., Chen, T., Hu, X., Zhang, Q., & Xiong, X. (2020). Design space exploration for yolo neural network accelerator. *Electronics*, 9(11), 1–15. doi:10.3390/electronics9111921.
- [29] Huang, R., Pedoeem, J., & Chen, C. (2019). YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data*, Seattle, United States, 2503–2510. doi:10.1109/BigData.2018.8621865.
- [30] Xiao, Y., Wang, X., Zhang, P., Meng, F., & Shao, F. (2020). Object detection based on faster r-cnn algorithm with skip pooling and fusion of contextual information. *Sensors*, 20(19), 1–20. doi:10.3390/s20195490.
- [31] Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., & Ouni, K. (2019). Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*. doi:10.1109/uvs.2019.8658300
- [32] Xu, X., Zhang, X., Yu, B., Hu, X. S., Rowen, C., Hu, J., & Shi, Y. (2019). Dac-sdc low power object detection challenge for uav applications. *IEEE transactions on pattern analysis and machine intelligence*, 43(2), 392–403. doi:10.1109/TPAMI.2019.2932429.
- [33] Abdul Rajak, A. R., Hasan, S., & Mahmood, B. (2019). Automatic waste detection by deep learning and disposal system design. *Journal of Environmental Engineering and Science*, 15(1), 38–44. doi:10.1680/jenes.19.00023.