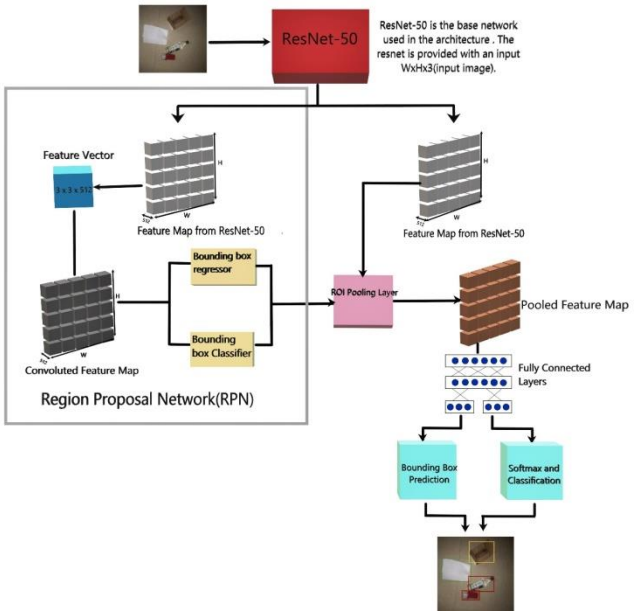
Global NEST Journal, Vol 24, No 4, pp 743-751

Multi object detection and classification in solid waste management using region proposal network and YOLO model

Jansi Rani S.V.*, Raghu Raman V., Rahul Ram M., and Prithvi Raj A.
Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu, India
Received: 30/09/2022, Accepted: 23/10/2022, Available online: 31/10/2022
*to whom all correspondence should be addressed: e-mail: svjansi@ssn.edu.in
https://doi.org/10.30955/gnj.004501

Graphical abstract



Abstract

Rapid urbanization has given us many benefits in terms of giving us a better standard of life, but it sure has brought a lot of problems with it. Solid waste management is a major issue that has been in the forefront of the issues caused by urbanization. It has brought a lot of domains together namely social, environmental, and climate together. Technology has been used occasionally but has not made significant advances in this domain. This domain is relatively new in terms of deep learning. This is due to some of the issues like lack of proper dataset, effective architecture to classify multiple objects and so on. The goal is to build multi-class dataset in this domain and perform detection and classification using both single stage and two stage object detection networks. The single stage network that is to be implemented is YOLOv5 and the two-stage network that is to be implemented is Faster Region based CNN using Resnet50. The single object dataset used is TrashNet dataset and the multi-object dataset used is Waste-mart self-built dataset. The result obtained shows mAP around 0.84 for the two-stage network and mAP around 0.98 with IoU threshold placed at 0.5 for both the systems.

Keywords: Classification, detection solid waste, faster RCNN, multi object, region proposal network, YOLO

1. Introduction

Solid waste management is one of the growing concerns that looms over the society. It cannot be avoided as it poses a real threat when it is not handled properly. Classification and segregation of waste has been involving human contact since the dawn of time, but to automate it to at least to a certain extent, we need to start incorporating deep learning in the domain. The pandemic has accelerated generation of solid waste. There has been an increase of plastic generation from 4.4 to 15.1 million tons (Peng *et al.,* 2021) dominated mostly by mismanaged plastics from medical industry. Most of this will be led into the ocean affecting the overall ecosystem. Direct Human contact with this hazardous medical waste will lead to increase in disease infections. According to the report form CPCD for 2019-20, the waste generation in India is at 1,52,076 TPD (Metric tons per day) (CBCB 2022). Although initiatives have been taken to start the processing of the waste at the early cycle of the solid waste, it must be noted that most of the solid waste generated ends in dumping yard without being processed. The overall management of the solid waste generated is a labor-intensive and time-consuming.

According to Environmental Protection Agency (EPA), the recycling rates of Municipal Solid Waste (MSW) have fallen from 35 percent in 2017 to 32.3 percent in 2018 (EPA 2022) Classification and segregation at the early stage of solid waste management cycle not only reduces the direct human contact, but also leads to increase in the recycling rates in solid waste. Since the domain is relatively new in the field of deep learning, single-object detection and classification has been done multiple times in the past. The dataset for single object detection and classification has been TrashNet dataset (Yang and Thung 2016). Single object detection is not scalable in terms of real-life implementation. To make it a more efficient process, multi-object detection should be achieved, and further classification should be expected. The main

motivating factor behind the proposed system is to achieve efficient system for multi-object detection and classification for solid waste. Once the before mentioned objective is achieved, it will be easier to segregate the waste into the required classes before processing it further. This can act as the first step taken to automate the whole process and make human contact as less as possible.

The dataset predominantly used in this field has been TrashNet (Yang and Thung 2016). It has 2527 images. Each image has one object in it. There are 6 classes namely glass, plastic, paper, trash, cardboard and metal. This has been used in multiple system to achieve single object detection and classification. The system that produces on of the highest accuracy score is optimized DenseNet121 with accuracy around 99.6 percent (Mao *et al.,* 2021). Here, the optimization is done by encoding the hyper-parameters of the CNN as the hyper-parameters of Genetic Algorithm. The unoptimized version of the same architecture returns an accuracy of 89.24 percent. The unoptimized version of Dense121 has over 7 million parameters. Another implementation has used an architecture named 'RecycleNet' (Bircanoğlu *et al.,* 2018), which fine tunes the architecture to achieve hyper parameters count of 3 million. The accuracy drops to 81 percent in this. The prediction time is relatively good at 15.6 ms (Xia *et al.,* 2022). The model does not provide multi-object detection.

Self-built data sets have been used in single object detection. A hybrid system containing CNN(AlexNet) and Multi-Layer Perceptrons (MLP)(Chu *et al.,* 2018) compares the new Multi-Layer Hybrid Systems (MHS) and the CNN. The result shows the MHS outperforming the CNN on all evaluation parameters: accuracy, precision and recall. As the problem involves image recognition, deep learning models (CNN) have been implemented in this domain. Residual Neural Networks have been used in single-object detection. ResNext architecture by Anh H. Vo (Vo *et al.,* 2019) a modified residual neural network with a cardinality value is worked on TrashNet dataset, and accuracy of 90 percent has been achieved. Similarly, the work done by Olugboja Adedeji *et al.*, (Adedeji and Wang 2019) uses ResNet-50 to the initial processing of the image, and uses SVM classifier to classify the data. The system achieves an accuracy of 87 percent on the Trashnet dataset. InceptionV3 has been used in the work done by Fatin Amanina Azis et al. (Azis *et al.,* 2020) and it produces an accuracy of 92.5 on the TrashNet dataset. Different versions of CNN have dominated image detection in this domain, the paper by Liu (Liu 2020) compares InceptionV3, DenseNet, Resnet, Xception and MobileNet on the trashNet dataset. The conclusion was shown as denseNet leading the accuracy with 95 percent, with second place held by Inception and resnet with accuracy of 94 percent. A review paper by Wanjun Xia et al., (Xia *et al.*, 2021) gives an overview of all the single object detection implementations used in solid waste classification. It compares different implementations of Machine Algorithms like ANNs, SVMs and KNNs. There was a real-life implementation using IoT, a CNN is used to send signals to a IoT system for real time monitoring (Rahman *et al.,* 2020) It is concluded that the 'future research should improve the effectiveness of the proposed framework in actual systems and achieve a balance between running time and accuracy.

As for multi-object detection in this domain, there's no definite multiclass dataset that can be used to compare different implementations. Most of the work in this domain has been either based on self-built data, or ImageNet data (as it is comparable with real life objects). The paper by Yayu Chen *et al.*, on multi-object classification returns a mAP score of 84.1 percent. The paper compares Faster RCNN architecture with Resnet50 as base network as well as MobileNet as the base architecture. With the residual neural network Resnet50 producing better results in comparison with MobileNet. Similar architecture was used in the Smart Street Litter detection and classification (Ping *et al.,* 2020) with mean accuracies ranging from 73 to 97 percent. The dataset in this implementation comprises of images collected from Google Street View data, ImageNet etc.

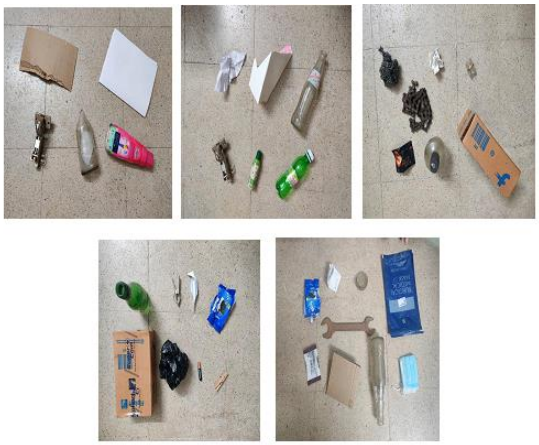


Figure 1. Sample images from Waste-Mart Dataset

Current implementations of classification problem have been done with CNN models like DenseNet121, DenseNet169, InceptionV3 and Alexnet. Although higher accuracies are obtained, multi-object detection and long testing times are not desirable. The solution to solving the problem can be using a version of R-CNN (Faster R-CNN) and YOLO to classify the classes. This will enable multi-object detection and faster testing time. We must obtain annotated data for the TrashNet Dataset. Annotated data helps in localizing the objects in the image. The annotated data is trained along with the images from the dataset and the model should be built for detecting multiple objects to classify the solid waste.

The main contribution of the paper is given below:

1. Building a deep learning model for classifying the solid waste
2. Building a multi class dataset with different sets of numbers of objects
3. Improving the mAP for the solid waste classification in the architecture with custom dataset

4. Tabulating the experimental results comparison with existing work helping us to check the mAP for different number of objects in the image and analyzing the efficiency, improving the mAP for the solid waste classification in the architecture with our dataset

5. Checking the impact of different residual networks for the dataset.

The paper can be summarized as follows, Section 2 discuss about literature, Section 3 describes the proposed system using Region Proposal Network and residual neural networks and YOLOv5, and Section 4 about the results and analysis.

2. Materials and methods

2.1. Dataset

From the literature review, it can be inferred that building a dataset for the multi-object detection and classification model will be helpful. The dataset that is to be built must have multiple objects in single image. The dataset will comprise classes from the standard dataset that has been used in this domain, i.e., TrashNet.

2.1.1. TrashNet dataset

The dataset has six main classes. They are glass, metal, cardboard, plastic, paper, trash as in Table 1. These images contain only one object in it. This can be used to implement models which can predict and classify a single object in the image.

Table 1. Images in the dataset

Class	Number of Images
Glass	501
Cardboard	403
Metal	410
Plastic	482
Paper	592
Trash	137

2.1.2. Waste-mart dataset

The Waste-mart Dataset is custom made dataset consisting of objects collected from the nearby Waste Marts (Figure 1) and hence we named it as Wastemart Dataset. The total number of images in the dataset is 3200 images. The 3200 images were classified into six different sets. Each set of 500 images had objects ranging from 4 to 10. The total number of objects in the self-built dataset will be 21,500 (500 X (4+5+6+7+8+9)+(10*200)). The dataset will be pre-processed to obtain the required dimensions of 480*600 for each image

2.1.3. Device configuration

OnePlus 7T with 48 MegaPixel. Images obtained of resolution 3000*4000. It is obtained and processed using OpenCv2. ImResize function is used to convert it into the desired 480 * 600.

2.1.4. Manual annotation

The working of the architecture relies on feeding it with annotation data along with the images. After procuring the objects from the local waste mart, the dataset is built. To annotate the data, a tool named 'LabelImg' (Tzutalin 2015) is used. The tool allows us to annotate the data and store in various formats for different architectures in machine learning. The format we used was PASCAL-VOC format, the default format. The PASCAL-VOC format is the format used in the Faster RCNN architecture. The annotation format used for YOLOv5 is YOLO annotation format.

2.2. Architecture of the classification and recognition model

The proposed system is based on the architecture of Faster Region-based Convolutional Neural Network (Faster R-CNN) (He *et al.,* 2016). Variations of the architecture has been built based on two different ideas. The first one will be changing the base network. The base networks that are to be used in the architecture will be ResNet50 as shown in Figure 2. This is the architectural diagram of the proposed system. The architecture diagram has been rendered using ResNet50 as the base network. The Faster R-CNN (Ren *et al.,* 2015) has used VGG as the base network. Further iterations have preferred residual neural networks. From Figure 2, we can see that the input image is fed into the base network. It is noted that the image is pre-processed using the OpenCV2 package and all images are made into the size of 416*554. Thus, it can be seen that an input tensor of 416*554*3, where 3 depicts the color channels (Red, Green, Blue). Note that the input dimensions can be alternating with either 416*554 or 554*416.

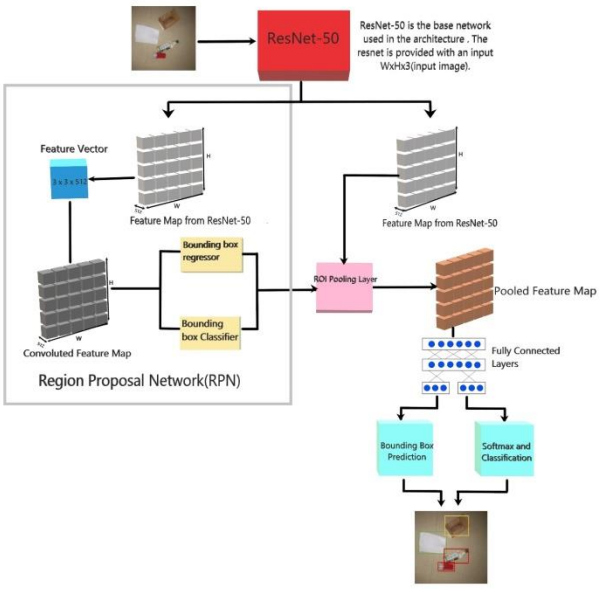


Figure 2. Faster R-CNN network for Solid Waste Classification

2.3. Algorithms

2.3.1. ResNet50

The Resnet50 is an example of residual neural networks as shown in Figure 3, used by Shaoqing Ren (He *et al.,* 2016) on the need for residual networks in image recognition. The reason for opting for a residual neural network over any other neural network is because it overcomes the

problem of 'vanishing gradient'. As the number of layers in a neural network, the back propagation process gets affected as there is a case of the diminishing gradient over each iteration of the back propagation. This will result in a gradient approaching zero over the layers.

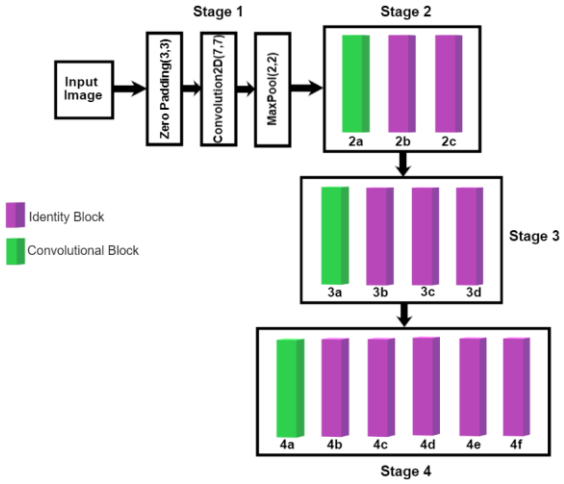


Figure 3. Architecture of ResNet50

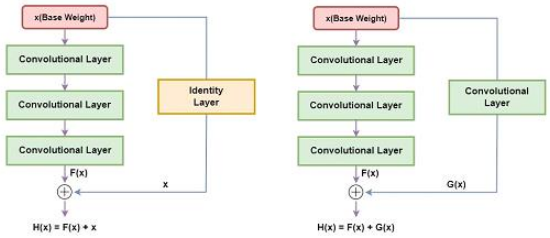


Figure 4. Identity and Convolutional Block

This is the reason why there is a decrease in the error rate as the layers are increased in the residual neural network while the opposite happens in the plain neural network. This is the effect of the vanishing gradient in a plain neural network. The way a residual neural network deals with this is by having identity blocks and convolutional blocks. An identity block is nothing but the propagation of weights from the input to the output function f(x). The identity block has 'skip connections. The convolutional blocks also propagate weights, but they pass it through a convolutional layer before adding them to the end. The diagrammatic representation of identity blocks and convolutional blocks is given below in Figure 4.

There are some alterations made to the Resnet50. In the last stage of the network, average pooling and the softmax layers are removed from the network. The convoluted feature map of W*H*512 is obtained, where W is the width of the original image, H is the height of the original image and 512 is the number of channels. Transfer Learning is used, and weights of no-top version of the base network trained on ImageNet is obtained. The weights are not frozen, back-propagation still occurs in the architecture when trained on our dataset. The next stages in the architecture are region proposal networks, followed by the ROI pooling layer. The Region Proposal Network (RPN) is an important step in the system. It will classify the processed convoluted feature map into two classes (foreground or background), and it also regresses the bounding box around the object. Anchor boxes form the base of the Region Proposal Network.

2.3.2. Transfer learning

Transfer Learning is the process of using knowledge(information) acquired from solving a problem and using it to solve a similar problem. The weights are obtained as a h5 file and passed as an input to the base network. The Figure 5 describes the transfer learning occurring in the proposed system.

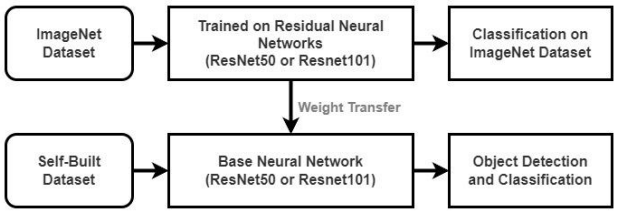


Figure 5. Transfer Learning for resnet50 and resnet101

2.3.3. Anchor box

Anchors are fixed rectangular boxes that are placed throughout the image with different sizes and ratios, that are going to be used to localize and detect the objects when first predicting object locations. The convoluted feature map will be of Width of the image X Height of the image X ConvDepth. It is important to understand that the anchors are drawn against the convoluted feature map. The feature maps are of the same length, and width as the input image. The anchor boxes will fit the image. The defined set of sizes for the anchor boxes are 64px, 128px, 256px, and the defined set of ratios between width and height of boxes are 0.5, 1, 1.5, and combinations of these two sets have been used to generate the anchor boxes. The ratios used in the system are (1:2, √2 2:1 and 1:√2 2). Thus, there are nine different boxes per anchor.

2.3.4. Region proposal network

The Region Proposal Network has two levels. The first level involves processing the anchor boxes, from the image to obtain a similarly shaped feature map. This level contains 512 convolutional kernels of dimensions 3*3*512. The first '3' stands for the size of the boxes, and the second '3' stands for the number of aspect ratios used. Now, the number of anchor boxes obtained per image can be calculated. Height = 416, Width = 554, Stride =16. Consider 16 as the stride length, the number of anchor boxes will be equal to ((416\16 ∗ 554\16) ∗ 9 = 8102). This is the first level of the RPN. The new feature map containing all the information from the image is sent to the second level. The second level of the Region Proposal Network contains two layers, a Regression layer and a classification layer. The feature map containing the proposals is processed by using 36 (9*4) convolutional kernels of filter size 1*1. Thus, bounding box parameters are obtained for W*H*9 proposals. This can be seen in Figure 6.

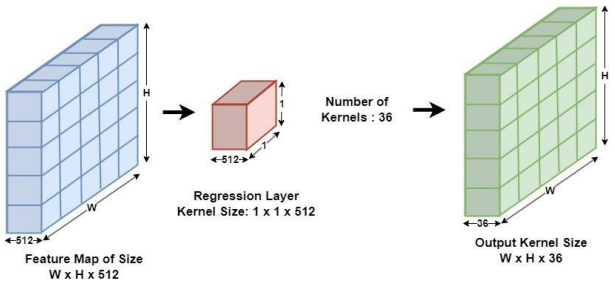


Figure 6. Regression layer in the second level of RPN.

The classification layer in the RPN contains 18 convolutional kernels of filter size 1*1 (Figure 7). This will provide the proposals in the image with an objectness score. The objectness score is calculated by Intersection over Union function (Figure 8). The IoU functions a value between 0 to 1 for each proposal. The threshold of the IoU function has been different in different implementations. The IoU is usually set to 0.5, where an object is classified as foreground if IoU > 0.5, the background is IoU < 0.3.

As it can be seen, the RPN does not care about the classes of the object. It only classifies the object as foreground or background, hence in simple terms a binary classifier. The RPN only sends an output of 256 proposals with a balanced ratio between positive anchors (foreground) and negative anchors (background).

2.3.5. Region of interest pooling layer (ROI)

As it can be seen, a set of object proposals is obtained form th RPN. The ROI pooling layer also gets its input from the base network as well. A balanced ratio of positive anchors(foreground) and negative anchors(background) is. The region of Interests containing the object is obtained. For every object proposal in our hand, the ROI takes the corresponding part of the input feature map, and formats into a fixed size using OpenCV. The fixed size that is used in our system is 7*7. Thus, a convolutional feature map of dimensions 7*7*conv-depth for each proposal is obtained. Non-max Suppression (Hosang 2017) is also carried out in this part of the system. Thus, a feature map with multiple ROIs pooled into it, is obtained. The bounding box co-ordinates for each of the proposal is provided with each proposal.

The last few layers of the proposed system will be the part where the bounding box is regressed more, and classify the objects into different classes. Thus, multi-object detection and classification can be achieved.

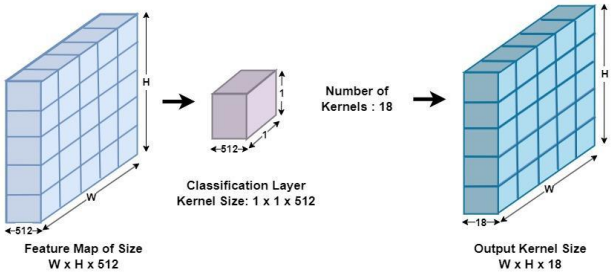


Figure 7. Classification layer in the second level of RPN.

2.3.6. R-CNN layer

The convolutional feature maps are passed through RoI pooling layers, object proposals along with images are obtained, and now must be classified into the specified classes. This is done in this stage of the architecture where the R-CNN will flatten the feature map and passed through two fully connected layers of 4096 size. Rectified Linear Unit is used as the activation function. The two fully connected layers are given below:

1. First one is the fully connected with N (in our case six classes) + 1 activation units which will classify the object. The additional one is the background class.

2. Second fully connected network 4 * N units where N is the number of classes. This regressor will further adjust the proposal box for better proposals.

2.3.7. YOLOv5

The YOLOv5 architecture (He *et al.,* 2016) follows a similar structure when it comes to object detection as shown in Figure 9. It consists of three different segments ordered sequentially to achieve object section in the image. The first segment will be the architecture backbone. This segment helps in feature extraction. Dense block networks are usually preferred as they also overcome vanishing gradient issue. The second segment consists of the architecture neck. The architecture neck uses the PANet to aggregate all the feature extrcated and construct a 'pyramid of object proposals' over the image. The final segment is the architecture head. The architecture head used in YOLOv5 is same as the model head used in YOLOv3 and YOLOv4.

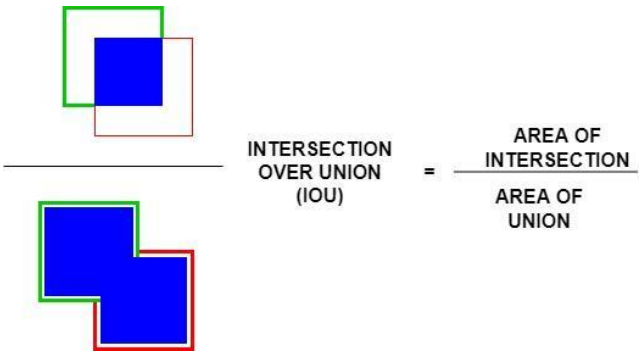


Figure 8. Intersection over Union

2.3.8. Model backbone - CSPDarkNet

The CSPDarkNet is a DenseNet based architecture. This uses the dense blocks to control the flow of the gradient thereby maintaining the learning rate. The bottleneck layers inside the dense blocks help in checking the flow of the gradient. Keeping a constant gradient growth rate not only helps in propagating the feature extracted in each layer and also reduces the number of operations that is to be performed. The constant gradient growth also means that the vanishing gradient problem is over come. The DenseNet is modified to obtain CSPNet. The CSPNet uses only a partition of the weights inside the dense block, so that the number of operations performed is decreased (10- 20 percent) in comparison with DenseNet network.

The transition layer also uses only a partition of the channels that is propagated forward. This makes the CSPNet more suitable for object detection algorithms as the features are extracted at a better rate with less computational power. This makes the CSPNet usable even in low performance devices. In YOLOv5, CSPDarkNet acts as the backbone to YOLOv5 creating feature maps.

2.3.9. Model neck - PANet

The Path Aggregation network uses the feature maps to produce feature pyramids containing the receptive field (Area of Interest). This aids in object scaling as same object of various sizes can be localized and identified. The PANet uses bottom-up. Path augmentation creating a shorter route in order to connect bottom and top layers. The PANet also uses 'Adaptive feature Pooling to pool all the features from different levels.

2.3.10. Model head-YOLOv3

The model neck for Yolov5 is same as the neck from previous iterations. This part of the architecture helps in regressing the bounding box around the object using regression loss that reduces with increase in certainty. It is also used to classify the object into their respective class based on the objectness score designated to the object.

2.4. Losses in the proposed system

Losses in the field of deep learning are used to increase the effectiveness of the model. The losses used and discussed are Binary cross entropy loss Categorical cross entropy loss, SmoothL1 loss (Azis *et al.,* 2020). The binary cross-entropy loss is used in the RPN part of the architecture. The RPN does the binary classification. The loss can be considered as the uncertainty that prevails in the model to classify an object into any label. To understand this, each prediction is compared to the actual ground value, and the distance between them is pushed through a negative logarithmic function to obtain the loss value. The lesser the loss value, the more certain the model is that it can classify the object into any label. The mathematical function is given below. In RPN structure, the image loss function of solid waste is:

$$L(\{p_i\},\{o_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls1}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(o_i, o_i^*)$$

$$where\ p_i^* = \{0\ negative\ sample\ 1\ positive\ sample \quad (1)$$

p_i represents the probability that the i^{th} suggestion box is predicted to be the true label. O_i, represents the prediction of the bounding box regression parameter of the i^{th} suggestion box, and O^*_irepresents the bounding box regression parameters of the true label corresponding to the i^{th} suggestion box. N_{cls} represents all 256 samples in a mini batch. N reg represents the number of suggested box positions. The λ is the balance coefficient. The waste image classification loss function is shown below:

$$L_{cls1} = -[p_i^* loglog(p_i) + (1-p_i^*) loglog(1-p_i,)] \quad (2)$$

Another loss in the RPN structure is the bounding box loss. It is defined by:

$$L_{reg}(o_i, o_i^*)\sum_i smooth_{L1}(o_i - o_i^*)\ where\ o_i = \{o_i = [o_x, o_y, o_w, o_h]\ o_i^* = [o_x^*, o_y^*, o_w^*, o_h^*] \quad (3)$$

$$here\{o_x = \frac{(x-x_{a)}}{w_a}\ o_y = \frac{(y-y_{a)}}{h_a}\ o_w = log\frac{w}{w_a}\ o_h = log\frac{h}{h_a}\ \{o_x^* = \frac{(x^*-x_{a)}}{w_a}\ o_y^* = \frac{(y^*-y_{a)}}{h_a}\ o_w^* = log\frac{w^*}{w_a}\ o_h^* = log\frac{h^*}{h_a}$$

$$smooth_{L1}(x) = \{0.5x^2\ if\ |x| < 1\ |x| - 0.5,\quad otherwise \quad (4)$$

x, y, w, h, respectively, are the center coordinates of the predicted bounding box and the width and height of the bounding box; x_a, y_a, w_a, h_a, respectively, represent the center coordinates of the candidate proposal box and the width and height of the bounding box; x^*, y^*, w^*, h^*, respectively, represent the center coordinates of the gtbox and the width and height of the bounding box; Smooth L1 is a loss function with good robustness. The reason for choosing Smooth L1 loss function is that it can act as both L1 and L2 loss when required. The L1 loss is obtained by calculating Least Absolute Deviations (LAB). The L2 loss is obtained by calculating the Least Squared Errors (LSE). These two will be used alternatively in the architecture. The L1 loss will be used when the values are outliers present in the data, and L2 loss is preferred when the deviations are nearing zero.

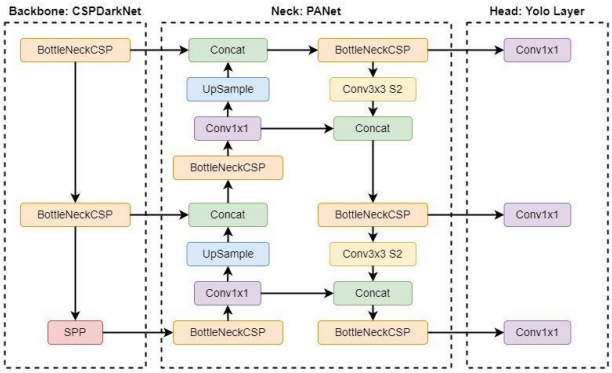


Figure 9. YOLOv5 Architecture

The above-mentioned loss functions are the loss functions base on RPN classifier and Regressor. Similarly, there are loss functions for the R-CNN classifier and R-CNN regressor.

$$L(p,u,t^u,v) = L_{cls2}(p,u) + \lambda[u]L_{loc}(t^u,v)$$
$$where\ L_{cls2}(p,u) = -log(p_u) \quad (5)$$

L_{cls2} is the loss function of the classifier L_{loc} is the loss function of bounding box regression. p is the probability distribution predicted by the classifier p = (p0, p1, . . . , pk) where p0 is the probability that the candidate region is the background, p1, . . . , pk are the probabilities that the candidate region is different category; u is the corresponding object true category label. P and u are used to find the classifier loss. Now, t^u is the regression parameter, v is the bounding box regression parameter v = (v_x, v_y, v_w, v_h) corresponding to the real target, λ is the balance coefficient, [u] is the inversion parenthesis, given below

$$|u| = \{1\ \ if\ u\ is\ true\ 0\ otherwise \quad (6)$$

The bounding box regression loss function of the waste image in the Faster RCNN model is given by

$$L_{cls2}(o_u,v) = \sum_{i[x,y,w,h]} smooth_{L1}(o_i^u - v_i) \quad (7)$$

The above mentioned are the loss functions used in the architecture.

3. Results

3.1. Device configuration

The proposed system was executed in Spyder IDE in Anaconda Environment. The device has Windows 10 system, with AMD Ryzen 7 3750H, 2300 MHz, 4 cores. The GPU in the device is NVIDIA GEFORCE GTX 1660Ti with Max-Q Design with 6 GB RAM.

3.2. Performance

The training of the proposed system uses transfer learning to begin with. The H5 files are obtained and passed into the program through system command. The dataset is split into 80 percent training set and 20 percent test set. A csv file containing all the file path for the images and the corresponding bounding boxes will be used to access each image. The proposed system has three data augmentation options in horizontal flipping, vertical flipping, and rotation of image by 90 degrees. The image input size is 480 * 600 or 600*480 according to the orientation of the image (landscape or portrait). The dataset will be trained on different iterations of the proposed system, and the weights will be saved and used for testing.

Table 2. Performance of Faster RCNN

Model (Faster RCNN)	Cardboard AP	Paper AP	Metal AP	Plastic AP	Glass AP	Trash AP	mAP
ResNet50	0.8882	0.8783	0.8796	0.7774	0.8216	0.8447	0.8483
ResNet101	0.8659	0.7285	0.8355	0.7802	0.8110	0.7572	0.7964

Table 3. Single object detection

S. No	Model	Accuracy
1	ResNet50 (Adedeji and Wang 2019)	0.94
2	DenseNet169 (Zang *et al.,* 2021)	0.82
3	Faster RCNN - ResNet50	0.848
4	YOLO v5	0.88

Table 4. Multi object detection

S. No	Model	Accuracy
1	Faster RCNN - MobileNet50 v2 (Chen *et al.,* 2021)	0.74
2	ZF + RPN (Awe *et al.,* 2017)	0.683
3	Faster RCNN - ResNet50	0.848
4	YOLO v5	0.98

The losses such as binary cross entropy, regression loss for RPN and categorical cross entropy, regression loss for RCNN discussed in the previous section are calculated for the proposed system and shown in Figure 10. As the epochs increases, the total loss of the system decreases.

The evaluation parameters are 1) mean Average Precision, 2) Average Precision, 3) Average Recall, 4) Average Precision across Scales (APAS), 5) Average Recall across Scales (ARAS) (Chen *et al.,* 2021). The scales refer to the size of the object in the image. If object is less than 32*32, then it is categorized as 'small', if it is between 32*32 and 96*96, then it is categorized as 'medium' and the rest is categorized as 'large'. This will give an insight into the working of the model for different scales. The following table will show us the mean Average precision, Average Precision, Average Recall, APAS, ARAS for the different iterations of the proposed system.

The precision and recall graph is very important in object detection domain. The precision tells us how many objects, identified as positive, are legitimately correct. The recall tells how many of the actual positives are identified as actual positives. The model cannot have a low precision but high recall value as well as high precision but low recall value. A balance must be struck between the precision and recall value for a model to be considered as a good model. The precision vs recall graphs give us insight into the working of the model. The ResNet50 has a relatively better precision and recall balance as seen from the graph. It produces high precision and recall values for all the classes in the model. This means that most of the objects are localized in the image and the predictions are

more accurate when compared to the model with Resnet101. The Figure 11 shows the accuracy, APAS, ARAS and Precisio vs Recall graph of RCNN. The mAP for both the models can be seen from Table 2.

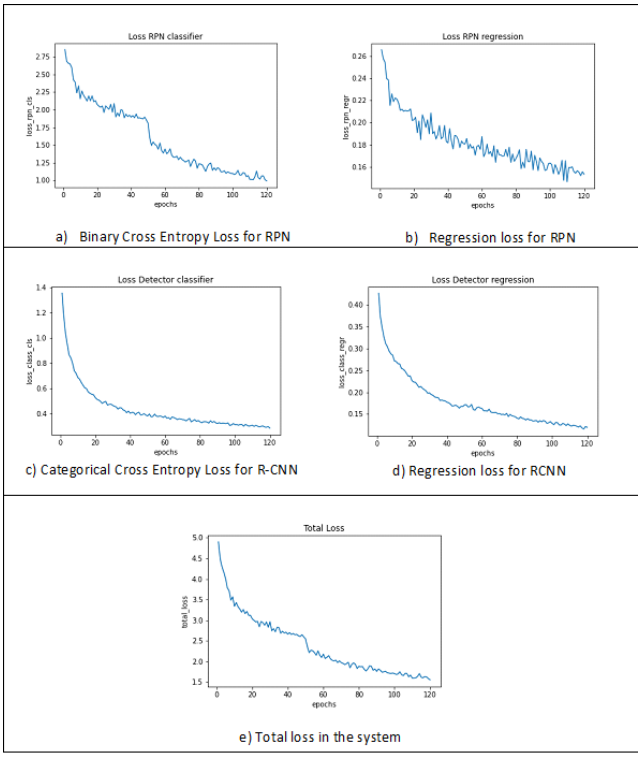


Figure 10 Losses for RCNN

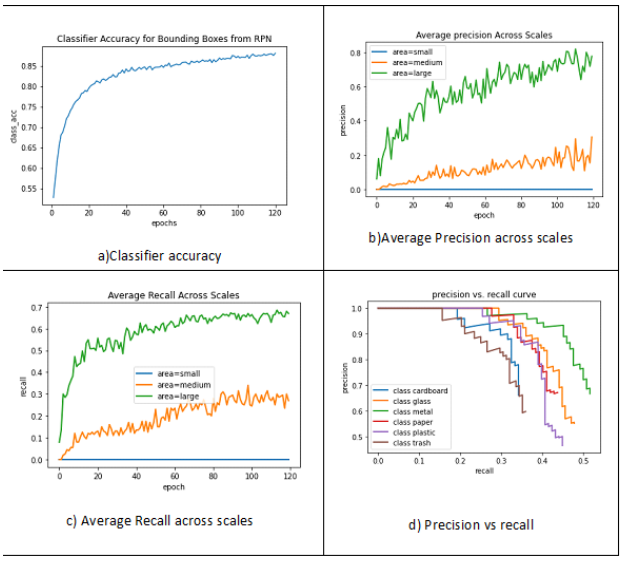


Figure 11 Performance Values for RCNN

The YOLO version 5 (Zhang *et al.,* 2021) works very well with the dataset. At 0.5 Intersection over Union (IoU) threshold, the mean Average Precision reaches a value of 0.98. But with average of IoU varying from 0.5 to 0.95, it fairs at a mean Average Precision of 0.73. This is done by taking the average of IoU set at 0.5 with an increment of 0.5 to reach 0.95 and shown in Figure 12.

3.3. Comparison with existing approaches

The models are compared with other models in the literature as shown in Tables 3 and 4. Faster RCNN produces an output at a mean Average Precision of 84.8 percent. The testing time for each image with multiple objects takes anywhere between 0.3 - 0.4 s. It can very well be implemented in real life scenario with solid waste being detected and classified at rate which can be useful in terms of reducing the overall time spent for managing the solid waste. The main drawback is in fact it is a two-stage network. The computational time taken for generating bounding boxes in the two-stage network will cost some extra time when compared to a single stage network like YOLOv5. The YOLOv5 takes around half the minimum taken by the Faster RCNN. The testing times for YOLO v5 ranges from 0.15 - 0.2 seconds. A frame gain of 2 frames per second is achieved in comparison with Faster RCNN using residual neural network (Resnet50).

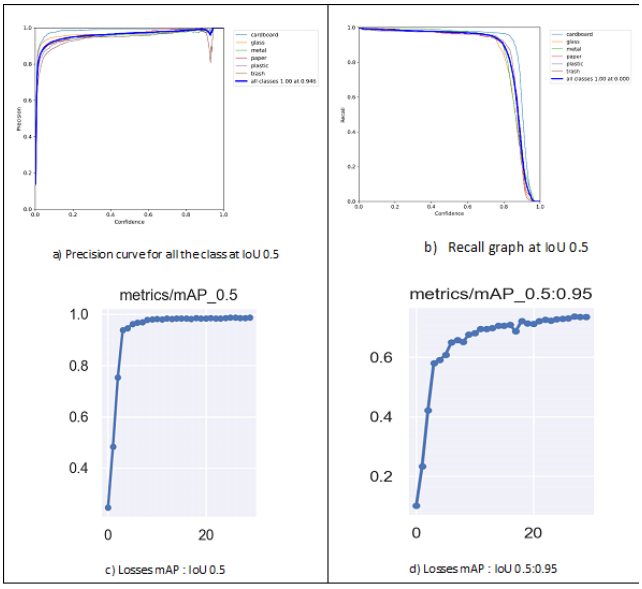


Figure 12 Performance Values for YOLOv5

The dataset has helped in producing the multi object detection and classification. The single stage network has outperformed the two-stage network in both single and multi-objects detection. The lack of availability of dataset with multi objects in single image was the main factor in building our own dataset. The annotations were done in two formats YOLO v5 format for the YOLOv5 architecture and the PASCAL VOC format for the Faster RCNN. The advantage provided by the CSPDarkNet over the DenseNet is very evident in terms of reduced computations. Reduction of the computation by a percentage of 10-20 means that dense blocks are trained better on relatively lower computations.

4. Conclusion

To improve the technology driven solid waste management, a model is proposed to detect multi objects present in solid wastes and also to build the multi-class dataset. In view of detecting multi objects, a custom dataset Waste-mart dataset is created and used. The two-stage network using Faster RCNN with ResNet50 is constructed and experimented with proper ablation study. To reduce time and improve accuracy, single stage network of YOLO v5 is experimented under different ablations. The result obtained shows mAP around 0.84 for the two-stage network and mAP around 0.98 with IoU

threshold placed at 0.5 for both the systems to detecting multiple objects. YOLOv5 outperforms the other models Faster RCNN with both ResNet50 and MobileNet50 v2 for multiple object classification. Future improves in the domain can be made in terms of implementing light weight detection architectures using the multi object dataset. The lightweight networks can be used in minicomputers, even on raspberry pi3 due their efficiency and the very small weight data for testing the images. These networks can be very well-fitted in the devices like drone so that objects can be classified on the go.

References

Adedeji O., and Wang Z. (2019). Intelligent waste classification system using deep learning convolutional neural network. *Procedia Manufacturing*, ***35***, 607–612.

Awe O., Mengistu R., and Sreedhar V. (2017). Smart trash net: Waste localization and classification. *arXiv preprint*.

Azis FA., Suhaimi H., and Abas E. (2020). Waste classification using convolutional neural network. In *Proceedings of the 2020 2nd International Conference on Information Technology and Computer Communications* (9–13).

Bircanoğlu C., Atay M., Beşer F., Genç Ö., and Kizrak M.A. (2018). RecycleNet: Intelligent waste sorting using deep neural networks. In *2018 Innovations in intelligent systems and applications (INISTA)* (1–7). IEEE.

Chen Y., Sun J., Bi S., Meng C., and Guo F. (2021). Multi-objective solid waste classification and identification model based on transfer learning method. *Journal of Material Cycles and Waste Management*, ***23***(6), 2179–2191.

Chu Y., Huang C., Xie X., Tan B., Kamal S., and Xiong X. (2018). Multilayer hybrid deep-learning method for waste classification and recycling. *Computational Intelligence and Neuroscience*.

CPCB India, Annual report 2019-2021, https://cpcb.nic.in/annual-report.php (2021). Date of Access: 09/09/2022.

EPA USA, National overview: Facts and figures on materials, wastes and recycling, https://www.epa.gov/facts-and-figures-aboutmaterials-waste-and-recycling/national-overview-facts-andfigures-materials. Date of Access: 09/09/2022.

He K., Zhang X., Ren S., and Sun J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* (770–778).

Hosang J., Benenson R., and Schiele B. (2017). Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* (4507–4515).

Liu J. (2020). An intelligent garbage classifier based on deep learning models. In *MIPPR Remote Sensing Image Processing, Geographic Information Systems, and Other Applications,* (**11432** 341–347). SPIE.

Mao W.L., Chen W.C., Wang C.T., and Lin Y.H. (2021). Recycling waste classification using optimized convolutional neural network. *Resources, Conservation and Recycling*, ***164***, 105132.

Peng Y., Wu P., Schartup A.T., and Zhang Y. (2021). Plastic waste release caused by COVID-19 and its fate in the global ocean. *Proceedings of the National Academy of Sciences*, ***118***(47), e2111530118.

Ping P., Kumala E., Gao J., and Xu G. (2020). Smart street litter detection and classification based on faster R-CNN and edge computing. *International Journal of Software Engineering and Knowledge Engineering*, ***30***(04), 537–553.

Rahman M.W., Islam R., Hasan A., Bithi N.I., Hasan M.M., and Rahman M.M. (2020). Intelligent waste management system using deep learning with IoT. *Journal of King Saud University-Computer and Information Sciences*.

Ren S., He K., Girshick R., and Sun J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, ***28***.

Tzutalin, Labelimg, https://github.com/tzutalin/labelImg (2015). Date of Access: 09/09/2022.

Vo A.H., Vo M.T., and Le T. (2019). A novel framework for trash classification using deep transfer learning. *IEEE Access*, ***7***, 178631–178639.

Xia W., Jiang Y., Chen X., and Zhao R. (2022). Application of machine learning algorithms in municipal solid waste management: A mini review. *Waste Management & Research*, ***40***(6), 609–624.

Yang M., and Thung G. (2016). Classification of trash for recyclability status. *CS229 project report*, **3**(1).

Zhang Q., Yang Q., Zhang X., Bao Q., Su J., and Liu X. (2021). Waste image classification based on transfer learning and convolutional neural network. *Waste Management*, ***135***, 150–157.