Deep Learning-Based Waste Classification System for Efficient Waste Management

by

Abdullah Al Nakib
17101145
Md. Nayem Talukder
17201026
Chinmoy Majumder
18201108
Soptorshi Biswas
17301073
Jabid Hassan
17201056

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
October 2021

Declaration

We hereby declare that

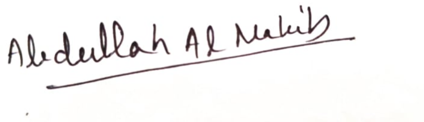
1. The submitted thesis is our original work during the completion of the degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, unless it is appropriately cited by means of a complete and accurate reference.

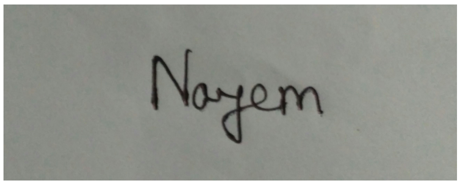
3. The thesis does not contain material that has been accepted or presented for any other degree or diploma from a university or other institution.

4. We have recognized all major sources of assistance.

Student's Full Name & Signature:



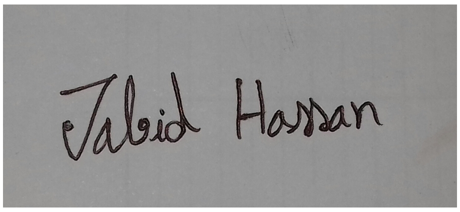
Abdullah Al Nakib
17101145



Md. Nayem Talukder
17201026

Chinmoy Majumder
18201108

Soptorshi Biswas
17301073



Jabid Hassan
17201056

Approval

The thesis titled "Deep learning-based waste classification system for efficient waste management" submitted by

1. Abdullah Al Nakib(17101145)
2. Md. Nayem Talukder(17201026)
3. Chinmoy Majumder(18201108)
4. Soptorshi Biswas(17301073)
5. Jabid Hassan(17201056)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on October 05, 2021.

Examining Committee:

Supervisor:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

A smart waste management system plays a vital role in building cleanliness, hygienic, and healthier living for the inhabitants of a city. However, the inherent problems of the waste management system are still a matter of great concern even amid this cutting edge of science and technologies. The root cause of this problem points to one fact - which is too much manual labor in the garbage collection, separation, and recycling process. In this research, we have used the Deep Learning-based model 'Mask R-CNN' to detect and classify Kitchen Waste, Glass Waste, Metal Waste, Paper Waste, and Plastic Waste from garbage dump waste images for the automation of the waste management system. We have also used the Explainable AI algorithm 'Grad-CAM' to introduce explainability to our model which helped to identify the most important features of each object and understand decisions of Mask R-CNN. Mask R-CNN model achieved 92.58% accuracy in classifying the 5 waste categories.

Keywords: CNN, Mask R-CNN, ResNet-101, Grad-CAM, Deep learning, Waste classification.

Dedication

We dedicate this work to our beloved parents who supported and guided us throughout our lives, to dear friends and respected teachers who helped and motivated us and to Dr. Golam Rabiul Alam for guiding and inspiring us to complete this work.

Acknowledgement

We would like to thank our research supervisor Associate Professor Dr. Md. Golam Rabiul Alam. Without his patience, guidance, and support we wouldn't be able to complete the paper. His inspirational words motivated us to keep working hard. Also, we are extremely grateful that you lead us and continued to have faith on us over the year. Furthermore, The success and outcome of this project were possible through contributions of all the team members. The paper required a lot of effort from each individual involved in this project. We are also grateful to our parents and friends for supporting us.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ANN Artificial Neural Network

CNN Convolutional neural network

$COCO$ Common Object in Context

$ConvNet$ Convolutional neural network

FPN Feature Pyramid Network

GPU Graphics Processing

$JSON$ JavaScript Object Notation

MHS Multilayer hybrid deep-learning framework

MLP Multilayer Perceptrons

NMS Network Management System

$R-CNN$ Region-based Convolutional Neural Network

$ReLU$ Rectified Linear Unit

$ResNet$ Residual neural network

RGB Red Green Blue

ROI Region of Interest

$RoIPool$ Region of interest pooling

RPN Region Proposal Network

VGG Visual Graphics Generator

Chapter 1

Introduction

Waste is a great concern in today's fast-growing world. The world produces 2.01 billion tonnes of solid waste every year from metropolitan areas, where more than of wastes are managed conservatively which severely hampers nature. By 2050, the amount of global waste is expected to rise to 3.4 billion tonnes [20]. So, effective waste management is a must to manage this huge amount of waste materials.

However, most of the developing countries are still using traditional waste management systems where most of the waste doesn't go through the recycling process and wastes are simply landfilled or incarnated which is not an environmentally friendly process. Insignificant amounts of waste are classified into different categories and hence recycled or properly disposed of. But these waste management and recycling processes are managed by people who handpick and categorize wastes which is very hazardous for their health [1]. Moreover, traditional waste management systems are not capable of handling this huge amount of waste efficiently.

Classifying and recycling waste properly is beneficial for the environment and also for the economy of the country. So, a smart waste management system is needed where different kinds of waste are going to be classified easily using automatic systems, and then depending on the kind of waste, it is going to be recycled or landfilled which is safer and environment friendly. Taking this into consideration we want to make a system using deep learning-based-convolutional neural networks that can detect 5 types of wastes such as kitchen waste, glass waste, metal waste, paper waste, and plastic waste from garbage dump waste images for the automation of the waste management system.

1.1 Research Problem

A clean and healthy environment is very important for the sound living of human beings. But keeping the environment clean and healthy especially in city areas is a big challenge but when it is a big city of a developing country it's even harder. Every day thousands of tonnes of solid waste are collected from different parts of the city. The solid waste of towns consists of solid wastes from public places, streets, houses, shops, offices, and hospitals and these wastes are usually the responsibility of municipal or other governmental authorities [1].

Most of the time these large amounts of waste are processed manually in developing countries like Bangladesh. Dhaka is one of the fastest-growing cities in the world which is the capital of Bangladesh. Every day around 3 thousand to 4 thousand tonnes of waste is generated in Dhaka city where 60% to 70% of solid waste is organic, 8-10% paper, 3-5 % plastic, 1-2% metal, 0.5-1.5% glass and 10-15% other wastes. Dhaka city still follows a traditional waste management system which is very expensive and 5-20% of the annual budget is used for this [3], [19], [2]. A large quantity of manpower and skill is needed for the various processes involved in solid waste management. However, the work is less efficient and environmentally friendly and consumes a huge amount of time and only 10-15% of waste is recycled [3]. The rest of the waste is landfilled. But if these wastes can be classified in different categories and recycled properly, the government can earn a huge amount of money which they can spend on making waste management more effective. The leftover waste can then be disposed of in an environmentally friendly process depending on its category since each category needs a different process to dispose of properly. So classifying these wastes into different categories is very important. If wastes go through a classification process and are separated early depending on the category, it is going to be easy to recycle and process the solid waste.

To do the classification process safely and efficiently, cities of developing countries need an efficient automatic system that can classify different types of waste in a short time and help to separate the waste. To implement an automatic system to classify the waste we prefer to use Deep Learning-based Convolutional Neural Network commonly known as CNN or ConvNet methods. We think using CNN to solve the problem is very effective since Convolutional neural networks (CNN) have been extensively used for object detection and applied in analyzing visual images. CNN is a kind of feed-forward neural network and operates on the principle of weight sharing [17]. Generally, CNN needs images containing investigated items as inputs and classifies images into different categories. CNN is like a 3D volume of a neuron with width, height, and depth. CNN consists of a series of convolutional , pooling , fully connected, and normalization layers [12]. A more dense CNN architecture can achieve an exponentially increased capability to detect objects compared to other traditional shallow models and greater accuracy can be achieved by fine-tuning the layers and adding a few extra details [13]. Furthermore, we want to add an Explainable AI(Artificial Intelligence) algorithm in our model to explain the action and results from the CNN model. Also explainable AI will help to get an idea of the importance of extracted features from the objects of the CNN model.

1.2 Research Objectives

This research aims to classify wastes into five categories consisting of kitchen, glass, paper, metal, and plastic waste. Automation in waste management would be helpful for cost minimization as it would reduce the dependency on manpower and management can be done more efficiently within a short period.

The objectives of this research are:

1. To understand the challenges of waste management systems in developing countries.
2. Develop a model that can detect the specified 5 kinds of waste based on CNN and apply it in the waste management system.
3. Add an Explainable AI algorithm.
4. Test and evaluate the performance of the model.
5. Explain the extracted features and decisions of our model using explainable AI algorithm.
6. Find ways to improve the model.

Chapter 2

Literature Review

In the field of Machine Learning, Deep Learning is an emerging field that aims towards making Artificial Intelligence closer than ever [6]. When classical machine learning techniques are used, the feature vector needs to be extracted at the very beginning, and to do this, advice from specialists is a must. Also, raw data can't be used without pre-processing while using machine learning techniques. So preprocessing also takes a great deal of time [11]. Deep Learning has eliminated this problem and made huge progress where machine learning was struggling for years. Unlike traditional machine learning and image processing techniques, even the raw data can be used to train the deep network. Deep learning-based models can learn the features, characteristics and attributes by its own deep architecture. Deep architecture maintains networks with multiple layers and the neighbor layers are connected which makes it a great choice to use in solving the problems of representation and classification through constant learning. Perceptron has some weight that combines all the known features and characteristics for identifying the objects [6].

Image classification is a set of techniques for labeling the image depending on the object present in it. The main drawbacks of image classification is that using image classification techniques only one object in the image can be classified and presence of multiple objects in an image can cause errors in result. Moreover, the location of the detected object is unknown since no bounding box or locator is not used to localize the object. But with the advancement of deep learning it has become easier than ever to classify and localize an object in the image. Object detection is a set of techniques for labeling the identified object and localizing the object using a bounding box. Object detection is a more complex operation than image classification, since it does both classification and localization. Using object detection multiple objects in a single image can be recognized.

Since we are going to use the system in waste management multiple wastes need to be recognized at the same time. There are many CNN models that can do object detection tasks very well with a high degree of accuracy. So, in order to implement a smart waste classification system we prefer to use Deep learning-based CNN or ConvNet. Since it has many advantages in object detection over other Deep Learning methods.

2.1 Convolutional Neural Networks (CNN)

CNN is a deep learning model which processes the data similar to the animal visual cortex. CNN models are designed to automatically and adaptively learn the most important features of the data from low to high-level patterns through a backpropagation algorithm [14]. CNN is like a black box model; the decision it takes can't be explained, even by the ones who built the network.

2.1.1 CNN Architecture

CNN is mainly composed of 3 kinds of layer:

Convolutional layer: in order to predict the class probabilities for every function,the whole image is scanned by applying filters to produce a feature map of the image.

Pooling layer: this is like a down-scaling operation. The information from the functions of the convolutional layer is reduced but the most important information is kept.

Fully connected layer: To predict an accurate mark, weights are applied over the input created by the feature analysis and the final probabilities for determining a class for the image are created by the completely connected output layer. [16]

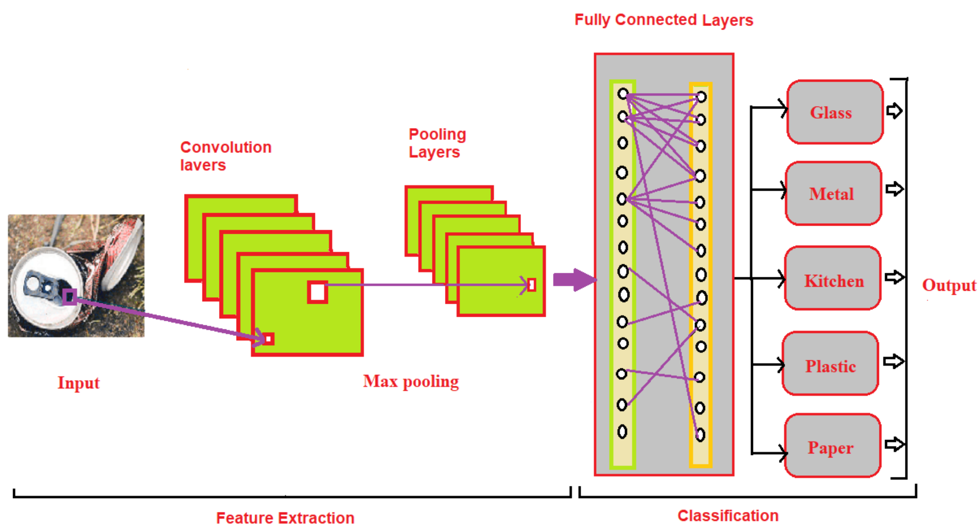


Figure 2.1: CNN Architecture

2.2 Related Works

We have studied various techniques used to classify waste in smart waste management. Every research paper illustrates a different set of techniques to classify or detect waste.

The research work [18] comes up with a solution to the conventional waste management system and introduces IoT-based smart bins. All bins consist of a Raspberry Pi with an ultrasonic sensor and pi camera that can detect and classify garbage from bins using a CNN-based Yolo algorithm. However, the efficiency of the system is still unknown since they haven't provided any data about the results.

This research [11] uses a Trashnet dataset and tests it on different deep learning models to classify different categories of waste. However, they only showed results of the 3 most successful models. First, they tried it on an Inception-V4 model with 100 epochs and got a test accuracy of 89 %. Later they tried the DenseNet169 model with 150 epochs which gave them 84 % accurate results. At last, on MobileNet with 150 epochs they got a similar result like the previous model. This paper also suggests fine-tuning in different CNN-based models to improve the accuracy of the model and classify waste into different categories. Overall, this paper will surely help us to select the best Deep CNN model for our research work.

It is a very important task to maintain a clean and hygienic civic climate, especially in developing countries. This research paper [5] introduces us to a smartphone app that can detect and coarsely segment garbage regions in a user-clicked geotagged image. The app uses a fully convolutional neural network approach to detect garbage regions. The model provides a result with 87.69% accuracy on their newly introduced dataset. In this paper, some recommendations were also given to improve the accuracy of the model. However, in this model classification of garbage is absent, it can only detect a chunk of garbage region which makes it different from our research purpose of classifying waste in different categories.

A multilayer hybrid deep-learning framework (MHS) was introduced in this research paper [12] to sort waste into different categories automatically. Since CNN relies only on the data derived from the image, the accuracy of prediction results is dependent on the quality of images. But this model only classifies images into 3 categories but in our research, we are going to classify 5 types of waste with their recyclability.

This paper [15] used Mask Scoring R-CNN for classifying and recognizing 4 types of waste and compared the result with Cascade R-CNN. Mask Scoring R-CNN is a modified version of Mask R-CNN with a MaskIoU which classifies, localizes, and creates a segmentation mask around the object. The quality of the mask depends on the confidence of the object in Mask Scoring R-CNN. However, Cascade R-CNN only classifies and localizes an object. Moreover, The dataset used for the research is prepared by the author. But the accuracy of the test set is comparatively low for both Mask Scoring R-CNN and Cascade R-CNN.

However, from the above discussion, it can be observed that most of the researchers who worked on waste management systems have not classified waste in various categories. They either simply detect wastes as a garbage region or classify waste into few categories using Deep Learning or other Machine Learning approaches. However, the researcher should classify the waste in various categories to simplify the recycling process and fasten the waste management system through automation.

Chapter 3

Methodology Selection

There are many convolutional neural network methods available to detect and classify objects from images and videos. We wanted to work on our thesis with a state-of-the-art method. In 2016, Mask R-CNN(Mask Regional Convolutional Neural Network) outperformed all existing, single-model methodologies available in almost every task and also became the winner of the COCO challenge 2016. [7] Since it provides a mask for every detected object it would be a handy feature for knowing the real shape of the object. If this model is used, in the future our work can be used in automatic garbage collecting robots. By identifying the mask color on objects, the robots can pick up the objects easily. Since this model can infer fast, objects can be detected very quickly. So we chose Mask R-CNN to use in our model. Mask R-CNN is a black box model, so there is a lack of transparency in decision making. The decisions of this kind of model are very hard to explain. So to understand the decisions and introduce explainability to our model, we have decided to add an explainable AI algorithm Grad-CAM(Gradient-weighted Class Activation Mapping) [10] in our model. Grad CAM generates a heat map on the most important areas of the object. We can understand the decisions of our model by analyzing the heat map of objects from different categories.

3.1 Mask R-CNN

Mask R-CNN is a deep learning algorithm that detects objects in an image and parallelly generates a high-quality segmentation mask for each instance. Mask R-CNN is an extended version of Faster R-CNN with a mask for each detected object. The mask helps to acquire the shape of the object which only the rectangular bounding box can't provide. Faster R-CNN mainly consists of two stages. The first stage is Region Proposal Network (RPN) which detects candidate objects with bounding boxes. The second stage extracts features using RoIPool from each candidate box and performs classification and bounding-box regression. Both stages can share their used features for faster inference. Mask R-CNN inherits the two stages from Faster R-CNN with an extra mask generated for each instance in the second stage. Our Mask R-CNN model is based on a ResNet network with 101 layers and an advanced Feature Pyramid Network (FPN). The ResNet-FPN backbone achieves great accuracy and speed for feature extraction.

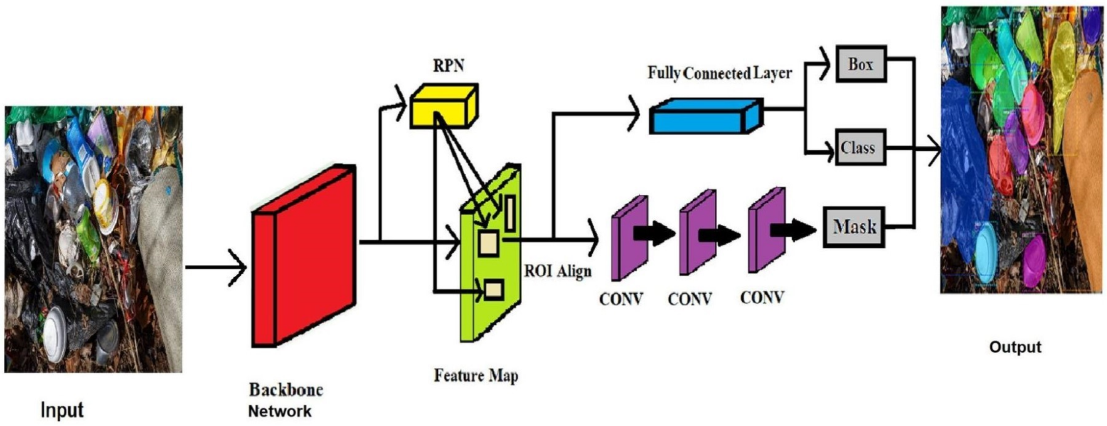


Figure 3.1: High-level overview of Mask R-CNN architecture.

3.1.1 ResNet-101

Deep Residual Network is one of the most groundbreaking works in the field of deep learning in the decade. Using the Deep Residual Network blocks, Resnet solves the vanishing gradient problem breaking the barrier to train deep network layers. Which makes it possible to train to hundreds or even thousands of layers without compromising with the performance. The performance of many computer vision tasks such as object detection has been significantly increased using the powerful representation ability of Resnet.

ResNet introduces a revolutionary idea called "identity shortcut connection" that jumps one or more layers if necessary.

ResNet uses Deeper Bottleneck architecture modifying the Deep Residual Network block to reduce training time. For each residual function F, a stack of 3 layers is used. The three layers are 1×1, 3×3, and 1×1 convolutions. Reducing and increasing dimensions is performed on 1 x 1 layers, the 3×3 layer a bottleneck is kept to deal with smaller input/output dimensions. [4]

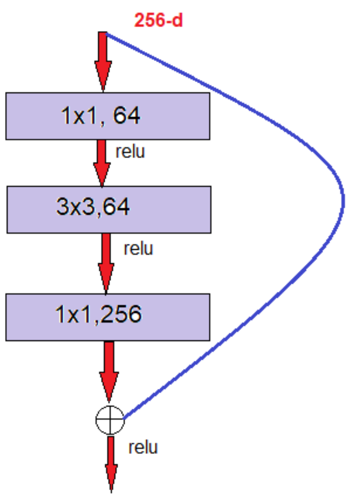


Figure 3.2: 'Bottleneck' building block for ResNet-50/101.

Depending on the depth of layers, the name of ResNet variants is defined. ResNet-101 has 101 layers of depth. ((conv1)1*1 + (conv2_x)1*1+3*3 + (conv3_x)3*4 + (conv4_x)3*23 + (conv5_x)3*3) = (1+1+9+12+69+9) = 101 layers. The conv3 1, conv4 1, and conv5 1 with a stride of 2 performs downsampling tasks.

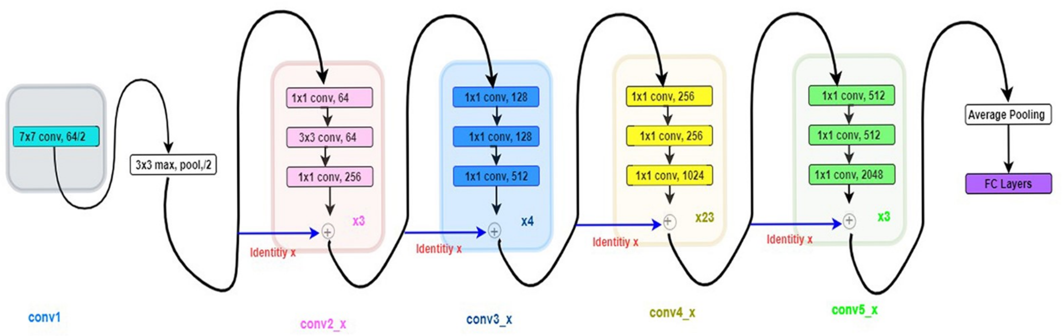


Figure 3.3: High-level overview of ResNet 101 architecture.

This ResNet-101 works as the backbone of the Mask R-CNN and serves as a feature extractor. While passing through this network from bottom to top an RGB image is converted to a feature map of shape 32x32x2048 and this feature map is passed to the next network layers as input.

3.1.2 Feature Pyramid Network (FPN)

In Mask R-CNN, Feature Pyramid Network proposed for Faster R-CNN by Lin et al [8] is used which achieves a good result in object detecting. A top-down architecture with lateral connections is used to build an in-network feature pyramid from a single-scale input. Moreover, the FPN backbone performs amazingly well while extracting RoI features from different levels of the feature pyramid depending on their scale.[8] FPN Pyramid gets the high-level features of the images from the first ResNet-101 pyramid and then sends them to the lower layers. Both lower and higher-level features of the image are available for every level using this technique.

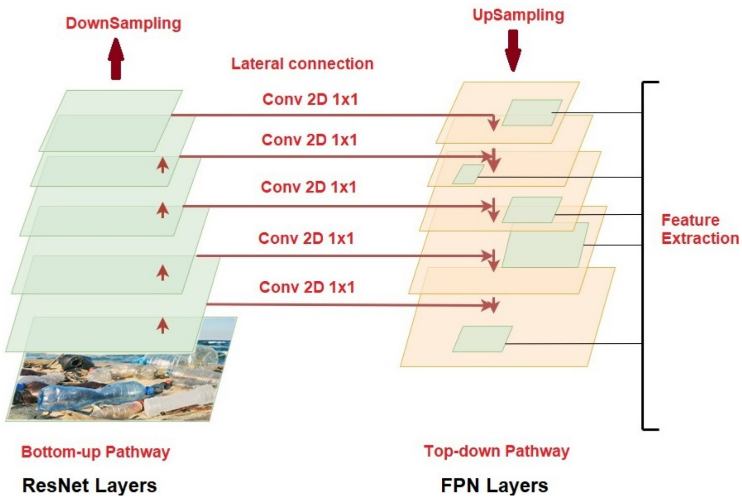


Figure 3.4: FPN building block with a top-down approach.

3.1.3 Region Proposal Network (RPN)

The RPN is a comparatively simple fully connected network that scans over the feature map provided by the backbone in a sliding-window fashion and finds areas that contain objects. The RPN generates an anchor box of different sizes in the image which is later analyzed to detect the foreground and the background of the image. These anchor boxes cover the whole image and even overlap. Since the size of the anchor box varies, it can take thousands of boxes to cover an image. For foreground anchor boxes or the positive anchors, RPN calculates delta which change the shape of the anchor box to perfectly fit with the object. By the help of RPN prediction, only the foreground anchors are selected. Also, the selected anchor's size and location are refined. The anchor box with the highest foreground is selected if there are too many overlapping boxes (NMS). The regions of interest are then sent to the next level.

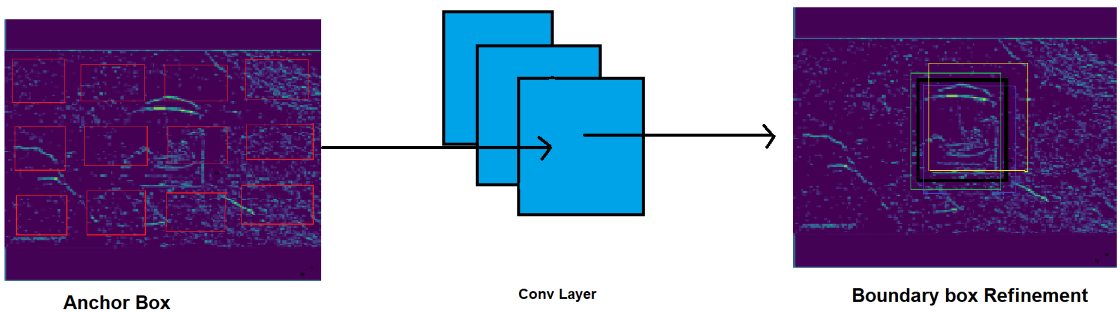


Figure 3.5: Region Proposal Network.

3.1.4 ROI Align

RoIPooling takes parts of the feature map and then resizes it to a fixed size. It is important to provide classifiers a fixed size input since the classifiers don't perform well in variable-size inputs. While pooling the alignment of the extracted feature and input can be compromised. So RoIAlign[7] takes samples from different points of the feature map and applies bilinear interpolation to properly align the extracted features with input. This increases the classification performance even more.

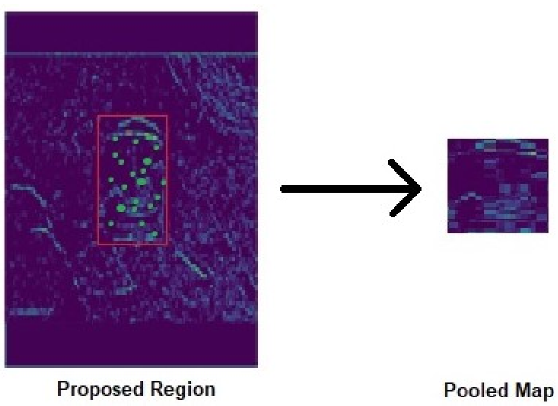


Figure 3.6: ROI Pooling.

3.1.5 ROI Classifier Bounding Box Regressor

The RoIs proposed by the RPN are scanned at this level. This stage has similar functionality like RPN. This network is deeper and it tries to predict the actual class of the proposed region. The proposed region gets ignored if it gets classified as background. This network also calculates to refine the bounding box of the object in the RoI like RPN.

3.1.6 Segmentation Masks

Masks are important to observe the actual shape and size of the object which the boundary box can't provide. The mask creating stage is a simple CNN network that takes only the selected positive regions by the RoI classifier to produce a mask for each detected instance in the image. The produced masks are soft masks that hold floating-point numbers in each pixel. So the created 28x28 low-resolution masks can be scaled up to the size of the actual object to fit properly.

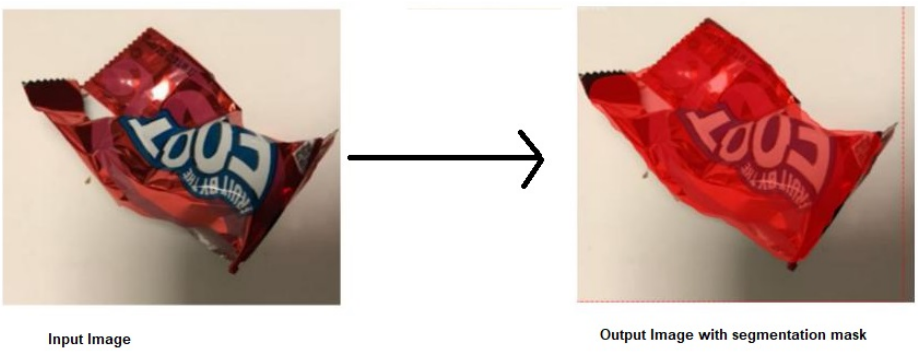


Figure 3.7: Segmentation Mask.

3.2 Calculation of loss function

Mask R-CNN is a very complex neural network model. Since it has to classify, localize and also generate a mask for the object, the calculation of loss function is complex as well. The loss function is calculated as followed:

$$L = L_{cls} + L_{\text{box}} + L_{\text{mask}} \tag{3.1}$$

loss = w1 × rpn_class_loss + w2 × rpn_bbox_loss + w3 × class_loss + w4 × bbox_loss + w5 × mask_loss

`Rpn_class_loss`: It indicates the loss generated by the improper classification of an object's presence on the anchor boxes by RPN.
`Rpn_bbox_loss`: It indicates loss in the localization accuracy of the boundary box.
`Mrcnn_class_loss`: This loss signifies the improper classification of the object present in the proposed region.
`Mrcnn_bbox_loss`: This loss represents the error of the localization of the bounding box of an identified object.

`Mrcnn_mask_loss`: This loss represents the inaccuracy on the identified object's created mask.

$$L\left(p, u, t^{u}, v\right)=L_{cls}(p \cdot u)+\lambda[u \geq 1] L_{loc}\left(t^{u}, v\right) \tag{3.2}$$

Here, p is Predicted Class, u is GT Class, t^u Predicted Bounding Box for class and v is GT Bounding Box.

$$L_{cls}(p, u)=-\log p_{u}(\text{ Log Loss }) \tag{3.3}$$

$$\begin{gathered} L_{loc}\left(t^{u}, v\right)=\sum_{i \varepsilon\{x, y, w, h\}} \operatorname{smooth}_{L1}\left(t_{i}^{u}-v_{i}\right) \\ \operatorname{smooth}_{L1}(x)=\left\{\begin{array}{cc} -0.5 x^{2} & \text{if } |x|<1 \\ |x|-0.5 \text{ otherwise} & \end{array}\right\} \\ \text{Smooth L1 Loss} \end{gathered} \tag{3.4}$$

Mask

- $K \cdot(m \times m)$ sigmoid outputs:
  - pixel-wise binary classification
  - one mask for each class, no competition
- L_{mask} : mean binary cross-entropy

3.3 Grad CAM

Mask R-CNN is a very complex deep learning model which not only detects and classifies the object but also produces a mask around the boundary of the object. Mask R-CNN is a black-box model in machine learning that even its designers cannot explain the reason behind its special decisions. For this kind of model Explainable AI is used to explain the model's specific decision. Grad-CAM is a very effective Explainable AI algorithm which can help to understand the decisions of any CNN based model with distinguishable layers. Grad-CAM [10] generates class-specific heat maps from the input image on a trained CNN model. We will use the Grad-CAM on Mask R-CNN model to understand which part of the object is important for classification and to assess the model's prediction, localization and segmentation.

Grad-CAM follows simple 3 steps to produce heatmap in the activation centre:

At first, it computes the gradient of class yc. Gradient is calculated from the feature map activations Ak of the convolutional layer. The gradient calculated in this step is 3D for a 2D image. As there are k numbers of feature maps of height v and width u, the shape of the gradient would be [k,v,u].

Then, the alphas are calculated by global average pooling over the height and width dimension to get the weight for the next step. Since the shape of our gradient was [k,v,u]. After pooling the shape will reduce in height and width dimension to [k,m,n]. Here, m < h and n < u.

Finally, We calculate the Grad-CAM heatmap by a weighted combination of forward activation maps followed by a ReLU. The alpha values of the previous step are used as weight on each feature map to calculate the final Grad-CAM heatmap. Then a simple upscaling algorithm is used to accurately fit the heat map on the original image.

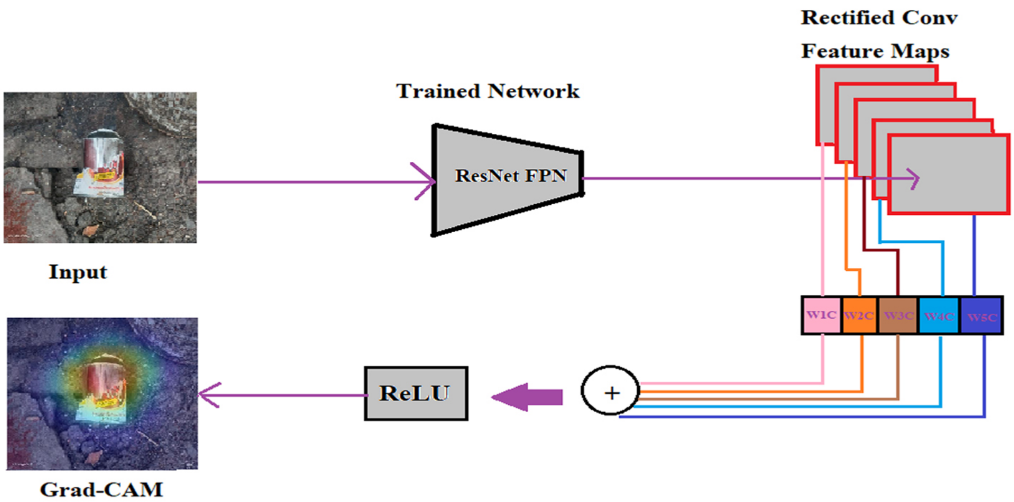


Figure 3.8: High-level overview of Grad-Cam Architecture

Chapter 4

Dataset Preparation

We aim to detect waste materials and classify them into 5 categories. In order to train our model to classify wastes accurately, we need a dataset with a large number of images of different types of wastes. Since we are going to use our model for the waste management system, we need real-life images of garbage collection, dustbins, etc where a lot of garbage objects are together. There are many datasets publicly available on the internet for free for waste detection and classification purposes like trashnet dataset, TACO dataset, etc. But these datasets alone don't fulfill our requirements for the project. Since the object count per image of these datasets is very low. Most of the images have one object. Assembling a dataset is the first barrier of any object detection project using deep learning. The challenge becomes harder when we do not have a standard dataset to follow in our project. So we decided to prepare a realistic dataset for our research from scratch.

4.1 Data Collection

4.1.1 Capturing Images

Firstly, we have captured images of wastes from different areas of our hometowns. We captured 2000 images using the camera of our mobile phones roaming around different areas. These images mainly consist of 5 types of waste mentioned earlier on the paper.

Figure 4.1: Several captured images (1)

Figure 4.2: Several captured images (2)

4.1.2 Collecting images from web scraping

There are thousands of waste images containing waste materials in different internet sources. We hand-picked around 600 best suitable images and included them in the dataset.

Figure 4.3: Different web scraped images

4.2 Preprocessing

After collecting all the data, we ensured that each image has a unique id by renaming them all. Then we carefully examined each image and removed all the blurry and low quality images. We went through the images and cropped the images if necessary. Since we collected the images from different sources, the resolution and the format of the images were different. So we converted the images in jpeg format and resized all the images to 512x384 pixels. Furthermore, we compressed the images to keep the dataset file size small without compromising the features of the images. As datasets with a smaller file size can be trained significantly faster. After all these preprocessing steps we only kept 1323 images captured by our phones and kept 553 images from web scraping for our dataset.

4.3 Annotation

We annotated most of the objects in the images which fall into the 5 categories. We annotated the images using VGG Image Annotator which is an open-source application. While annotating, we used polygon to define the boundary of the object. We assigned the category and the subcategory of each object. The produced annotation file was in JSON format.We have annotated a total of 10284 objects and found 48 sub categories.

Figure 4.4: Annotation of different images (1)

Figure 4.5: Annotation of different images (2)

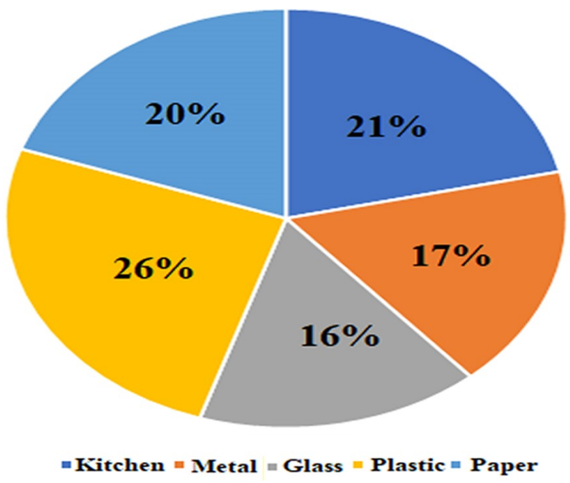


Figure 4.6: Object percentage per category

Total Number of annotation along with sub category is given below:

Category	Sub-category	Object no
Plastic	plastic_container_waste	159
	plastic_drinks_bottle_waste	747
	plastic_polythene_bag_waste	391
	plastic_polythene_packet_waste	607
	single_use_food_box_waste	162
	single_use_plastic_cup_waste	67
	plastic_laboratory + medical_waste	86
	plastic_pipe_waste	16
	plastic_zipper_bag_waste	16
	plastic_cleaner_bottle_waste	23
	plastic_shampoo_bottle_waste	43
	plastic_toothpaste_waste	7
	plastic_scrap_waste	276
	Total	2612
Kitchen	kitchen_biodegradable_waste	285
	kitchen_organic_waste	423
	kitchen_tea_bag_waste	125
	kitchen_vegetable_waste	512
	kitchen_fruit_waste	315
	kitchen_food_waste	543
	Total	2203
Glass	glass_broken_jar_waste	238
	broken_glass_waste	647
	broken_plate_waste	361
	broken_water_glass_waste	282
	crushed_glass_waste	92
	glass_light_bulb_waste	63
	Total	1683

Category	Sub-category	Object no
Paper	paper_cardboard_waste	119
	paper_newspaper_waste	157
	paper_soap_packet_waste	140
	used_tissue_paper_waste	169
	kitchen_fruit_waste	315
	kitchen_food_waste	543
	paper_craft_paper_waste	164
	scrap_paper_waste	383
	shredded_paper_waste	267
	magazine_paper_waste	170
	paper_light_box_waste	163
	single_use_tea_cup_waste	17
	paper_bekary_box_waste	124
	used_mask_paper_waste	64
	paper_food_pack_waste	43
	paper_cork_sheet_waste	39
	Total	2019
Metal	metal_alloy_scrap_waste	303
	metal_crushed_cans_waste	703
	metal_electronic_scrap_waste	305
	metal_household_scrap_waste	246
	metal_rod_waste	31
	metal_drums_waste	39
	metal_bottle	140
	Total	1767

Table 4.3: Total Number of annotations

Chapter 5

Implementation

5.1 Project set-up

We set up the Mask R-CNN on Google Colab Pro online platform. Colab Pro provides an Intel(R) Xeon(R) CPU with 2.20GHz clock speed, 24 GB ram, and an Nvidia Tesla T4 with 14 GB memory as GPU. The Mask R-CNN code was built on Python version 3.7, Tensorflow version 1.14, and Keras version 2.2.5.

We also added Grad-CAM on the Mask-RCNN. We used the feature map produced by the ResNet-FPN backbone of the trained network as the input of our Grad-CAM model.

5.2 Data Augmentation

We split the dataset in train,validation and testset in 14:3:3 ratio. Deep learning models require a lot of data while training to achieve a very good accuracy in the test phase. Using data augmentation we can increase the amount of training data and reduce overfitting on models as well. Data augmentation is not a new field and there are a lot of data augmentation techniques available. Multiple data augmentation techniques can be used in a specific problem [9]. Since we do not have a huge amount of images of waste, we have used data augmentation in our dataset. We have used 3 data augmentation techniques on our training set. They are scale, rotation, and flip.

Figure 5.1: Output images after applying different augmentation techniques

5.3 Training

We trained our Mask R-CNN model on the training set. While training initially we used COCO pre-trained weight for Mask R-CNN. Using pre-trained weight is an efficient and faster way to train our model to get to a good accuracy. Anchor plays a vital role in separating the object from the background. Depending on the object separation region proposal and boundary box is generated. So anchor size is very important to achieve a very good accuracy for the model. Since the resolution of the images of our dataset is low, the anchor size should be small as well. For our model, we set the anchor box size to [4, 8, 16, 32, 64, 128, 256] pixels. The anchor boxes are square in shape. We have used only one GPU and the image per GPU was set to 2, hence the Batch size=2. For each epoch, we have set the steps per epoch to be 2000 and also set the weight decay and momentum to be 0.0001 and 0.9 respectively.

Initially, we trained the head layers of the network for 20 epochs at different learning rates from 0.002 to 0.0002 and at different RPN-NMS thresholds from 0.6 to 0.8. So that our model can learn the most out of the dataset.

Then we trained all the layers of the network except the head layers for another 12 epochs with a stable learning rate 0.0001, RPN-NMS thresholds 0.8 and other parameters. At last, we trained all the layers of the network for 8 epochs with a stable learning rate 0.0001, RPN-NMS thresholds 0.8 and other parameters.

We also tuned other parameters on every 4 epochs to improve the accuracy depending on the results of the model after being tested.

Epochs	Learning Rate	RPN-NMS Threshold
1-4	0.002	0.6
5-8	0.001	0.6
9-12	0.0008	0.7
13-16	0.0005	0.8
17-20	0.0002	0.8
21-52	0.0001	0.8
33-40	0.0001	0.8

Table 5.2: Learning rate and RPN-NMS threshold during training

Since we have tuned the parameters during training both the training and validation loss curve are not smooth. Both the training and validation loss become very low after the training sessions of 40 epochs which indicates that the model has been trained well.

Figure 5.2: Training Loss.

Figure 5.3: Validation Loss.

Chapter 6

Result Analysis

Figure 6.1: Output images of Mask R-CNN on testset (1)

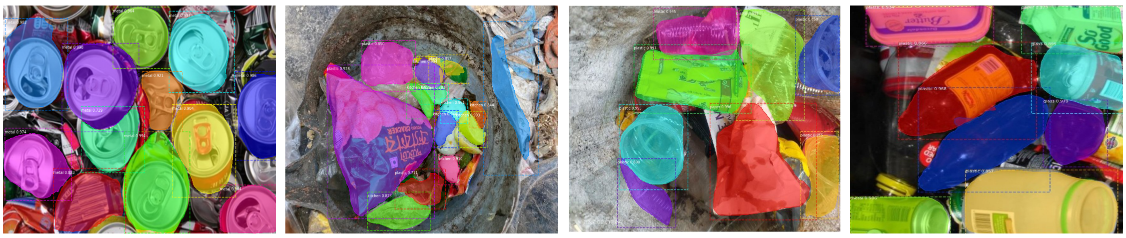


Figure 6.2: Output images of Mask R-CNN on testset (2)

6.1 Accuracy

Total number of images in the testset is 264 and the total number of objects present in the testset is 2378. So the average number of objects per image is 9. Since our research is based on a classification problem, we are evaluating the model on the classification metrics. So our main concern is classification performance of our model.

Accuracy: The number of test samples correctly identified by the model as either truly positive or truly negative out of the total number of test samples. The formula for checking accuracy is

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6.1}$$

The total number of detected objects is 2371 and the final testset accuracy is found to be 92.58%. Since in our dataset the classes are imbalanced. So we cannot judge our model by only accuracy.

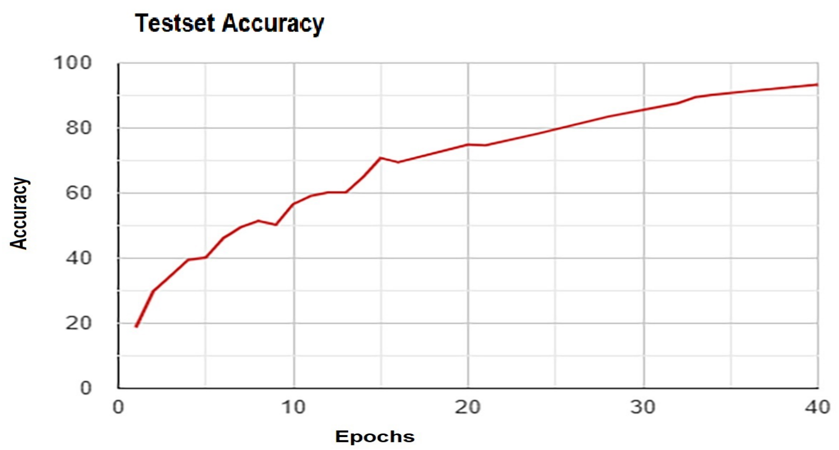


Figure 6.3: Testset Accuracy.

6.2 Confusion matrix

In Figure 6.4, we can see the confusion matrix of Mask R-CNN on our testset. The columns of the matrix present the ground truth or the actual number of objects and the rows of the matrix present the prediction of the model.

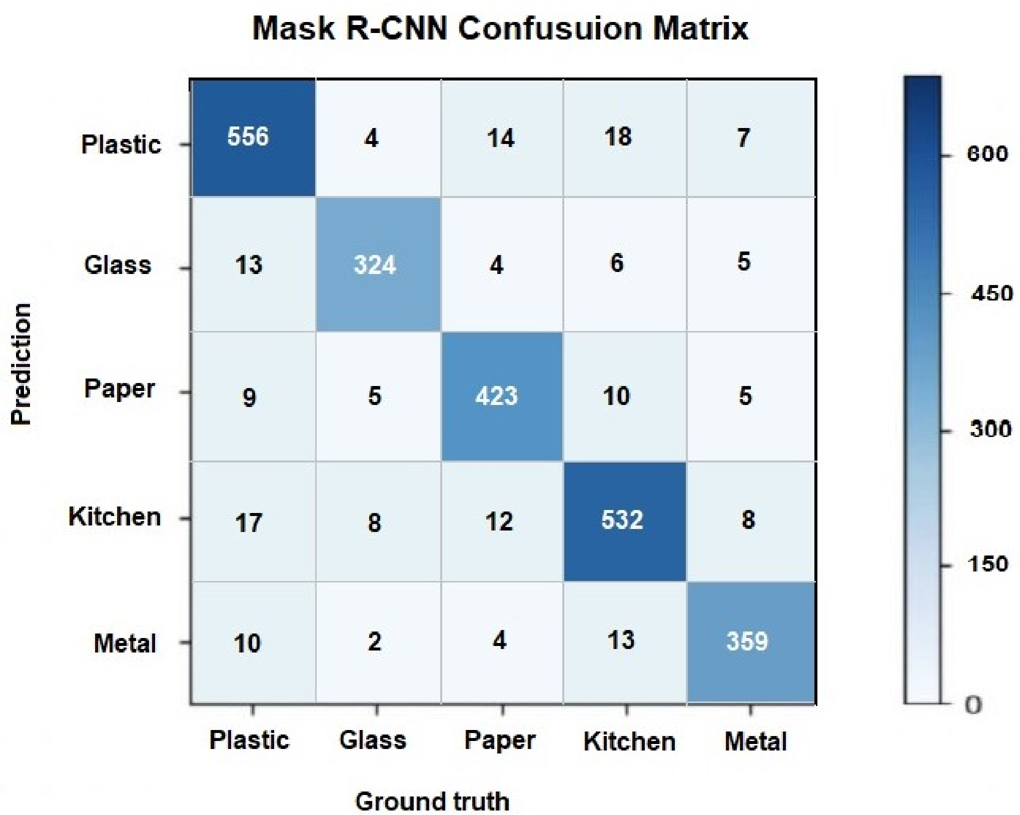


Figure 6.4: Confusion matrix of Mask R-CNN on testset.

6.3 Precision, Recall, Specificity, F1-Score

True Positives(TP)- predicted the test sample as positive for a certain class and actually the sample is positive which means the sample belongs to the certain class.

True Negatives(TN)- predicted the test sample as negative for a certain class and actually the sample is negative which means the sample doesn't belong to the certain class.

False Positives(FP)-predicted the test sample as positive for a certain class but actually the sample is negative for that class.

False Negatives(FN)- predicted the test sample as negative for a certain class and actually the sample is negative for that class.

Precision- the probability of a positively identified test sample to be actually positive. In other words Percentage of actual positive samples out of the total predicted positive instances. The formula of precision is:

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

True Positive Rate(TPR)/ Sensitivity/ Recall-the probability of a actually positive sample to be recognized as positive. The Percentage of predicted actual positive samples out of the total actual positive samples. The formula of recall:

$$Recall = \frac{TP}{TP + FN} \tag{6.3}$$

True Negative Rate(TNR)/ Specificity-the probability of a negative sample to be truly negative. The Percentage of negative samples out of the total actual negative samples. The formula of specificity:

$$Specificity = \frac{TN}{TN + FP} \tag{6.4}$$

F1 score: the harmonic mean of precision and recall. This gives the balance between precision and recall. The formula of F1 score:

$$F1Score = \frac{2 * \text{ precision } * \text{ recall}}{\text{precision } + \text{ recall}} \tag{6.5}$$

	TP	FP	FN	Precision	Recall	Specificity	F1-Score
Plastic	556	43	49	0.928	0.915	0.975	0.921
Glass	324	28	19	0.920	0.918	0.983	0.919
Paper	423	29	34	0.925	0.921	0.984	0.923
Kitchen	532	45	47	0.922	0.917	0.974	0.919
Metal	369	32	25	0.918	0.918	0.985	0.918

Table 6.2: Category Wise Precision, Recall, Specificity and F1-Score

6.4 Overview of the performance of the Mask R-CNN

The accuracy of the model in testset is 92.58%, precision 92.10%, recall or sensitivity is 91.80%, F1 score is 91.45% and mAP score is 92.07%. From all the evaluation metrics the performance of the model looks excellent.

6.5 Explainability using Grad-CAM

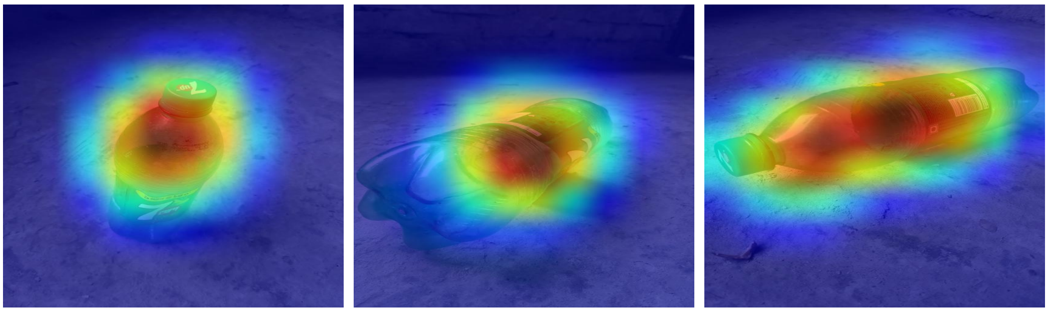


Figure 6.5: Grad-CAM output of a bottle (1).

In Figure. 6.5, we can see that the most important part of the bottle is marked red and the less important areas are marked using yellow, green. The blue color is used for the background. We can see that the most important feature is the upper or the lower part of the bottle depending on the position of the bottle in the image. We can see that our system is using the sticker part of the bottle as the least important feature. Using this sticker part as an important feature of the bottle will create confusion with the paper category or a bottle without any sticker will not be recognized.

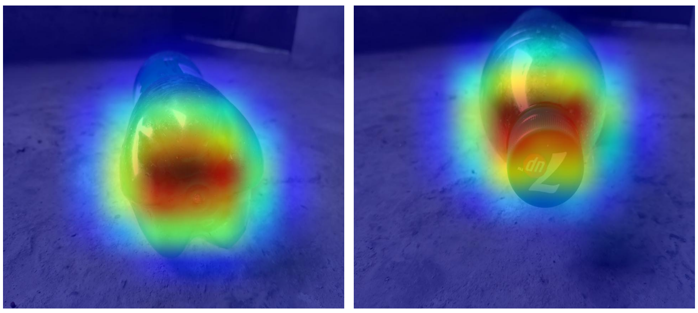


Figure 6.6: Grad-CAM output of a bottle (2).

If we see Figure. 6.6, we can see the important feature of the bottle from other angles, but still our model is not using the bottle cap's logo as the most important feature. This clearly signifies that our trained model is intentionally ignoring the sticker or the logo of the bottle to avoid confusion with other categories to maximize the chance of detecting any plastic bottle in any situation.

Analyzing the Grad-CAM result from each class signifies that our model is extracting the most important features from the objects of each class. Since the feature extraction of our model is accurate, we can say that our model is performing well in terms of decision making of classifying objects.

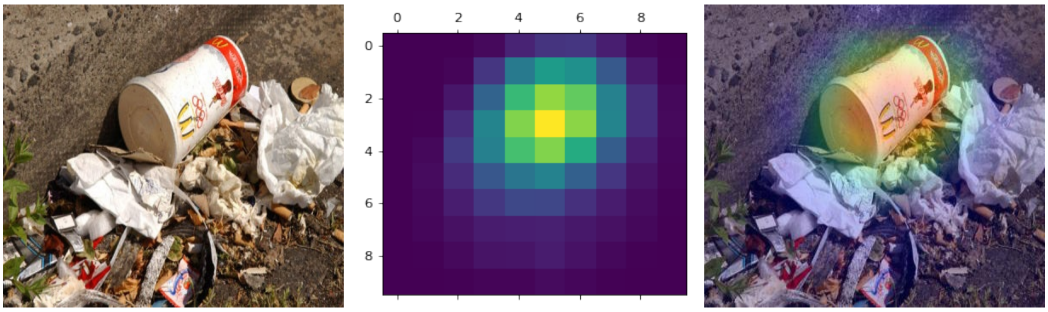


Figure 6.7: Grad-CAM results(class specific) on testset image (1) from the test set(from left: original image, activation map,Grad-CAM output).

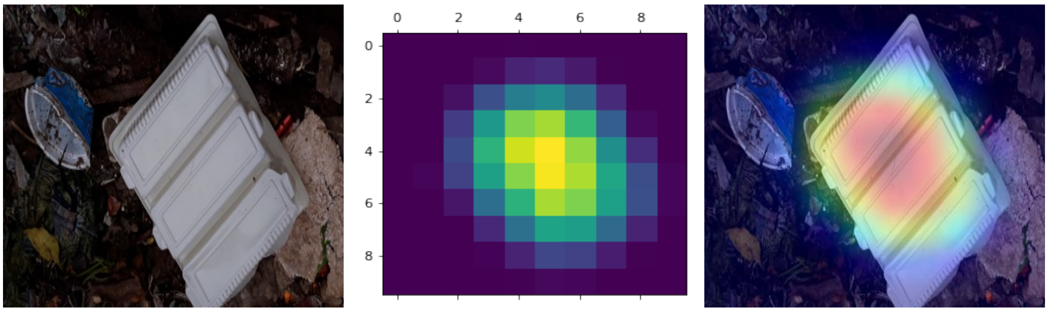


Figure 6.8: Grad-CAM results on testset image (2) from the test set(from left: original image, activation map,Grad-CAM output).

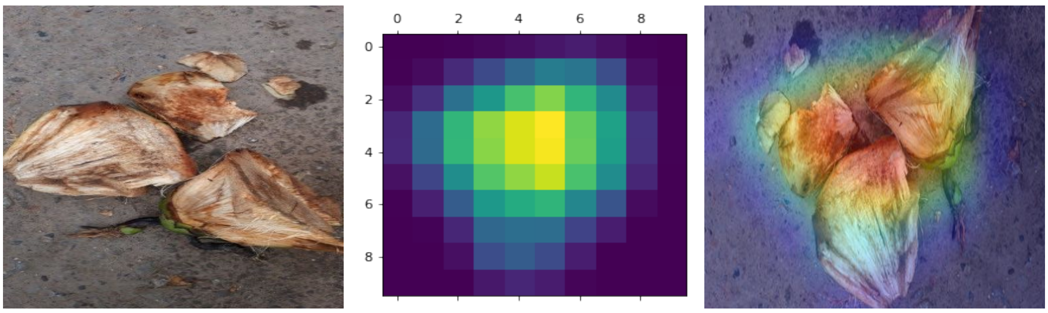


Figure 6.9: Grad-CAM results on testset image (3) from the test set(from left: original image, activation map,Grad-CAM output).

Chapter 7

Conclusion and Future Works

Major cities of developing countries are growing rapidly. But the authorities are struggling to keep the city clean and hygienic even after spending a lot since the waste management system is traditional and depends on manual labor-based processing. This is one of the sectors that is still deprived of the blessings of modern science and technology and the amount of research done to modernize this area is insufficient. Our research aims to modernize the classification and separation of solid waste and help to recycle the waste more effectively with a low cost and efficient system. Since our Mask R-CNN-based model can classify and also mark the objects in the frame with great accuracy. This can help in building an automated system to separate the wastes and help to recycle a greater amount of waste. Thus this research can help to keep the environment clean and make cities a better place to live. We want to make the dataset of the project more realistic and improve the accuracy of the model. Furthermore, we want to add more categories and subcategories to our system in the future.

Bibliography

[1] C. Zurbrugg, "Solid waste management in developing countries," *SWM introductory text on www. sanicon. net*, vol. 5, 2003.

[2] M. Zahur, "Solid waste management of dhaka city: Public private community partnership," 2007.

[3] M. A. Abedin and M. Jahiruddin, "Waste generation and management in bangladesh: An overview," *Asian Journal of Medical and Biological Research*, vol. 1, no. 1, pp. 114–120, 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[5] G. Mittal, K. B. Yagnik, M. Garg, and N. C. Krishnan, "Spotgarbage: Smartphone app to detect garbage using deep learning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 940–945.

[6] S. Sudha, M. Vidhyalakshmi, K. Pavithra, K. Sangeetha, and V. Swaathi, "An automatic classification method for environment: Friendly waste segregation using deep learning," in *2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, IEEE, 2016, pp. 65–70.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[8] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[9] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[10] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[11] R. A. Aral, Ş. R. Keskin, M. Kaya, and M. Hacıömeroğlu, "Classification of trashnet dataset based on deep learning models," in *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, 2018, pp. 2058–2062.

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

[13] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[14] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.

[15] S. Li, M. Yan, and J. Xu, "Garbage object recognition and classification based on mask scoring rcnn," in *2020 International Conference on Culture-oriented Science & Technology (ICCST)*, IEEE, 2020, pp. 54–58.

[16] V. Maeda-Gutierrez, C. E. Galvan-Tejada, L. A. Zanella-Calzada, J. M. Celaya-Padilla, J. I. Galván-Tejada, H. Gamboa-Rosales, H. Luna-Garcia, R. Magallanes-Quintanar, C. A. Guerrero Mendez, and C. A. Olvera-Olvera, "Comparison of convolutional neural network architectures for classification of tomato plant diseases," *Applied Sciences*, vol. 10, no. 4, p. 1245, 2020.

[17] B. Qiang, R. Chen, M. Zhou, Y. Pang, Y. Zhai, and M. Yang, "Convolutional neural networks-based object detection algorithm by jointing semantic segmentation for images," *Sensors*, vol. 20, no. 18, p. 5080, 2020.

[18] D. Vinodha, J. Sangeetha, B. C. Sherin, and M. Renukadevi, "Smart garbage system with garbage separation using object detection," *International Journal of Research in Engineering, Science and Management*, 2020.

[19] *Solid waste management in dhaka*, https://www.waste.ccacoalition.org/sites/default/files/files/city_fact_sheet/Dhaka_MSW_FactSheet_0.pdf, Accessed: 2021-09-24.

[20] *Trends in solid waste management*, https://datatopics.worldbank.org/what-a-waste/trends$_{i}n_{s}olid_{w}aste_{m}anagement.html?fbclid = IwAR1jo5TyNP9KBJmVQcNkN1RA$
Accessed: 2021-09-24.