Computer Vision Based Waste Classification using Deep Learning

by

S M Yeaminul Islam
19273001

A project submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
M.Engg. in Computer Science and Engineering

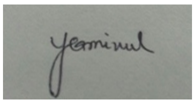
Department of Computer Science and Engineering
BRAC University
October 2020

Declaration

It is hereby declared that

1. The project submitted is my/our own original work while completing degree at Brac University.
2. The project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The project does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



S M Yeaminul Islam
19273001

Approval

The project titled "Automatic Waste Separation from Garbage Dump using Deep Learning" submitted by

S M Yeaminul Islam (19273001)

Of Summer, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Engg. in Computer Science and Engineering on October 9, 2020.

Examining Committee:

Supervisor:
(Member)

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

Program Coordinator:
(Member)

Amitabha Chakrabarty, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac University

Mahbubul Alam Majumdar, PhD
Professor and Dean, School of Data and Sciences
Department of Computer Science and Engineering
BRAC University

Abstract

Waste management systems and their inherent problems 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 - can't keep up to the pace with which garbage generation happens. In this research, We will propose a novel Deep Learning based approach of automatic separation of five kinds of waste materials namely - Kitchen Waste, Glass Waste, Metal Waste, Paper Waste, Plastic Waste, from the garbage dump for an efficient recycling process, which not only improves the efficiency of the current manual approach but also provides a scalable solution to the problem.

The contributions of this project includes a fully human labelled data set consists of 2000 images of garbage dump and a real time garbage localization and classification framework based on a single stage object detection algorithm. For the baseline, we have used YOLOv4 Object Detection Algorithm and with some fine tuning, we proposed a modified object detection framework which yields a mAP of 66.08% with an inference speed of 70 milliseconds on both images and videos.

Keywords: Waste Classification; Solid Waste Separation; Garbage Recycling; Garbage Waste Data set; Object Detection; Computer Vision; Deep Learning; Urban Environment.

Dedication

This work is dedicated to my beloved parents, loving wife, adorable daughter and Dr. Golam Rabiul Alam for guiding and bearing with me during this period with love and patience.

Acknowledgement

Firstly, We humbly praise and grateful to the Almighty, who permits us to live and accomplish tasks including the research work that is being presented here.

It is a great please to express the unfettered gratitude and profound respect to my supervisor **Dr. Golam Rabiul Alam**, Associate Professor, Department of Computer Science and Engineering, BRAC University, for his constructive suggestions, scholastic guidance, constant inspiration and kind co operation throughout the entire progress of this research work. It would have been impossible for me to accomplish my task without his invaluable advice.

I also would like to thank all the concerned teachers and stuffs of the department for their direct and indirect assistance at different events of the work.

Finally, gratitude to my Family and Friends for their encouragement and unconditional moral support.

Table of Contents

Declaration i

Approval ii

Abstract iii

Dedication iv

Acknowledgment v

Table of Contents vi

List of Figures viii

List of Tables ix

Nomenclature x

1 Introduction 1
- 1.1 Motivation 1
- 1.2 Research Problem 1
- 1.3 Aims and Objectives 2
- 1.4 Organisation of the Report 2

2 Background Study 4
- 2.1 Waste Generation 4
- 2.2 Waste Management 5
  - 2.2.1 Waste Management Systems 5
  - 2.2.2 Recycling and Composting 5
- 2.3 Sensor Based Robotic Agent 6
- 2.4 Computer Vision 7
- 2.5 Object Detection and Classification 7
  - 2.5.1 YOLO Models 8

3 Related Works 10

4 Data Set Preparation 13
- 4.1 Publicly available open labelled data sets 13
- 4.2 Web Scraping 13
- 4.3 Capturing Images 13

4.4 Using Data Augmentation 15
4.5 Synthetic Data Generation 15
4.6 Data Set Preparation Steps 15

5 Methodology **18**
5.1 Model Architecture 18
5.1.1 Input of the network 18
5.1.2 Feature Extractor backbone 18
5.1.3 Darknet-53 18
5.2 Multi-scale detection 21
5.3 Output of the Network 22
5.4 Loss Function 22
5.5 Some Additional Tweaks 23
5.5.1 Yolov4 Neck 23
5.5.2 Yolov4 Bag of Specials 23

6 Implementation & Result Analysis **25**
6.1 Implementation Details 25
6.2 Training History 26
6.3 Results Analysis 27
6.3.1 Metrics at I0U- 0.50 27
6.3.2 Metrics at I0U- 0.45 29
6.3.3 Metrics at I0U- 0.40 30
6.3.4 Metrics at I0U- 0.35 30
6.4 Checkout the model on some test data 31

7 Conclusion & Future Research **33**

Bibliography **35**

List of Figures

2.1 Total Waste Generation w.r.t Population 4
2.2 Total Waste Generation w.r.t GDP 5
2.3 Waste management framework in Bangladesh 6
2.4 Overview of Object Recognition Computer Vision Tasks 7
2.5 Image Classification vs Object Detection 8
2.6 Bounding Box Representation, Taken from: Original Yolov2 paper . . 9
2.7 High Level Network Architecture 9

3.1 Representative of waste images from 11
3.2 Representative of waste images from 11
3.3 Representative of waste images from 12
3.4 Representative of waste images from 12

4.1 Search Category granularity to accumulate garbage photos through web scrapping 14
4.2 Annotated Images of Garbage Waste - Self Taken Photographs 16
4.3 Annotated Images of Garbage Waste - Web Scrapped Images 16
4.4 Annotation percentage per category 17

5.1 The End to End Model Development Process 19
5.2 The High Level Architecture 20

6.1 Customised Anchor Box coordinates determined by running K-Means Algorithm over the whole data set images 26
6.2 Feature Visualisation from Feature Extractor Output of glass images 26
6.3 Feature Visualisation from Feature Extractor Output of plastic images 27
6.4 Feature Visualisation from Feature Extractor Output of paper images 27
6.5 Learning rate decay during training 27
6.6 Training Loss deceasing over epochs during training 28
6.7 Validation Loss and mAP Monitoring during Training 28
6.8 mAP, Average IoU & F1-Score Over different IoU Thresholds 29
6.9 Sample inference on real world data (1) 31
6.10 Sample inference on real world data (2) 32
6.11 Sample inference on real world data (3) 32

List of Tables

6.1 The highlights of accuracy metrics, inference efficiency of our model . 30

Nomenclature

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

BCE Binary Cross Entropy

CNN Convolutional Neural Network

GAN Generative Adversarial Network

$GINI$ Garbage in Images

GPU Graphics Processing Unit

IoT Internet of Things

MLP Multi Layer Perceptron

MSE Mean Square Error

SVM Support vector Machine

$TACO$ Trash Annotations in Context

$YOLO$ You Only Look Once

MSW Municipal Solid Waste

Chapter 1

Introduction

1.1 Motivation

Bangladesh is a densely populated country with it's population is on course to be about 17 crore by the end of year. This huge population comes with a lot of human made hazards, and solid wastes are surely a major part of it. An enormous volume of garbage waste is getting generated each day, particularly in urban areas and regrettably, management of solid waste is being worsen as the days pass by. The major cities of Bangladesh, generates around 16 thousand tons of waste a day, which summing up to more than 5.84 million tons a year. This amount is on the rise and will reach 47 thousand tons per day and approximately 17.16 million tons per year due to rapid expansion of urban area, population growth and hike in per capita waste generation rate. With the current amount of urban inhabitants, the waste generation rate is 0.41 kg per capita per day and the garbage collection efficiency ranges from 37% to 77% in different cities [2]. So, we are observing a big difference from wastes being generated and the wastes being treated here. Lack of awareness, technical limitations, strict law and financial constraints; a significant portion of wastes; are not duly treated, accumulated, disposed or recycled in their desired places. Consequently, it's a growing headache for the municipalities - how manage these solid waste efficiently and ensure not only a healthy environment for the residents but also make effective use of the litters as well through recycling.

1.2 Research Problem

Now, one solution to this enormous problem can be by improving the rate of treating the wastes but it's really hard for human garbage workers to catch up to the rate garbage generates. Right from collection to accumulate wastes in respective suburb's dumping house and later proper recycling or disposing takes time and effort from the individual, but it is important for the environment as it helps reduce pollution caused by waste and the need for raw materials [15]. So the major challenge, specifically with the garbage separation, turns out that it's a manual process and not many people think about it and when they do, they don't do it correctly.

Automated solution to this problem involves computer vision and sensor based approaches which will help an automatic classifier to detect and separate the waste objects as per their predefined classes. It imposes a wider range of challenges as well. The waste objects in nature can be obscured, twisted, camouflaged, transparent, transformed in different forms, aged, which makes a model extremely difficult to generalise in production environment. In spite of several attempts like - https://github.com/letsdoitworld/wade-ai, the problems seemed unresolved with a lack of a enriched data set with contextual images and a solid framework to detect and classify major categories of waste.

1.3 Aims and Objectives

A few decades back, a robotic agent, moving inside a dumping station and deliberately segregating wastes into predefined categories, will surely qualify for a science fiction movie! Fortunately, with the advent of IoT enabled smart devices and robotics, fast and remote maintenance of mechanical activities can be facilitated and waste management can surely be a part of it. But, a problem arises that those systems require accurate representation of the kinds of waste to be treated. There is some recent research works, which pointed out this problem and their solutions [7] [8] [14] [18] [21] but the major problem with those works are related to general applicability, not contextual representation of garbage object, real time detection etc. Now, detecting trash in open environment is a daunting task, as it can be twisted, pellucid, shattered, and camouflaged and adding to that, the vast feature diversity of our natural world should be taken into consideration as well. In this this

project, we will propose a solution of an automated garbage separation system using Computer Vision and Deep Learning technology to encounter this issue. Our proposed system will be able to detect, localise and classify garbage objects from piles of waste in natural environment all in real time through pictures/videos/cameras. This information then can be passed to a robotic agent immediately to automate the process using a micro controller based embedded system. So, In summary our propositions are -

1. A fully human labelled data set consists of 2000 images of garbage dump (half of them are scrapped from the internet and others are captured real time with my Redmi 4X device in broad daylight)
2. A real time garbage localization and classification framework based on a fast and accurate object detection algorithm.
3. Open source the project on GitHub, so that fellow researchers working on the same problem can collaborate and enrich the data set and also the framework.

1.4 Organisation of the Report

The report is structured in the following manner - In section II, I shall try to explain the concepts related to the subject matter in detail. In section III, a short walk through of the previous studies will be provided. Then, in section IV, I shall briefly explain the data collection procedures and considerations of our proposed garbage detection data set. Afterwards, in section V, the framework architecture and design choices will be reviewed. Finally, I'll wrap up with the experimental results and some future scopes of this research in section VI and section VII respectively.

Chapter 2

Background Study

2.1 Waste Generation

Waste generation is something we can't avoid as long as we, humans, live in this beautiful planet Earth. Waste has been generated since the outset of human civilization but during recent years, with the development of technologies, there is a significant amount of artificial materials prevail around us, which, when converted to waste make dangerous non bio degradable objects - which is the most concerning thing about waste generation now a days. As types and quantities of waste vary in different localities, I will try to depict the scenario from a Bangladesh perspective.

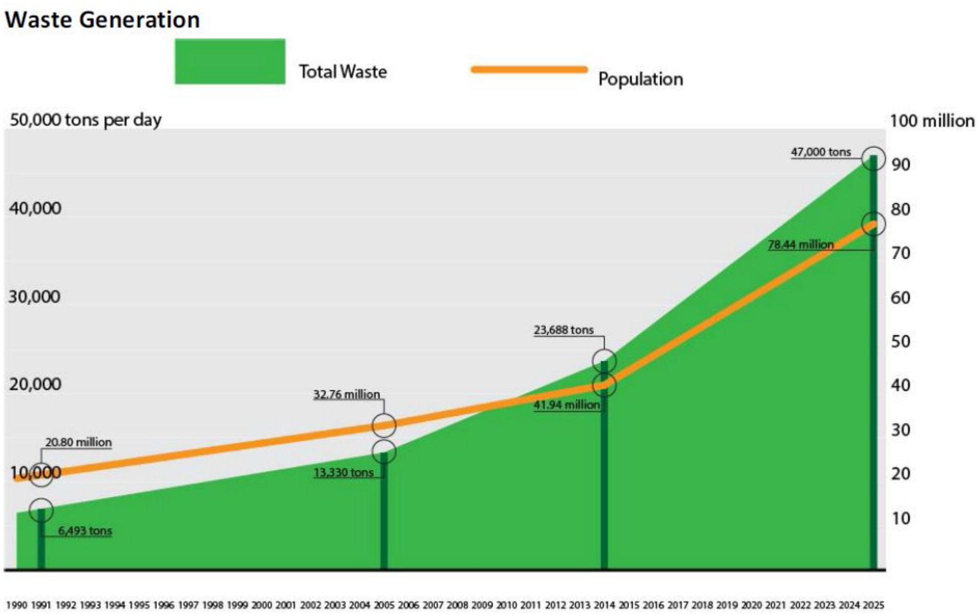


Figure 2.1: Total Waste Generation w.r.t Population [4]

In city areas, MSW generation rate is increasing with the growth of population (Figure 2.1). As reported in [20], a total of 7.7 thousand tons of solid waste is being generated from the six divisional cities of Bangladesh on a daily basis, while the capital itself has a 69% share in the total waste volume. The formation of the total stream of waste is approximately 74.4% of organic matter, 9.1% of paper, 3.5% of plastic, 1.9% of textile & wood, 0.8% of leather & rubber, 1.5% of metal, 0.8% of glass and 8% of other wastes [3]. The constituents behind this waste composition are population growth, advanced life styles, economic conditions, seasonality, weather, amount of recycling, and waste management programs. As we can see from (Figure 2.2), it's evident that, the waste generation rate is linearly evolving with respect to the GDP growth of this nation as well.

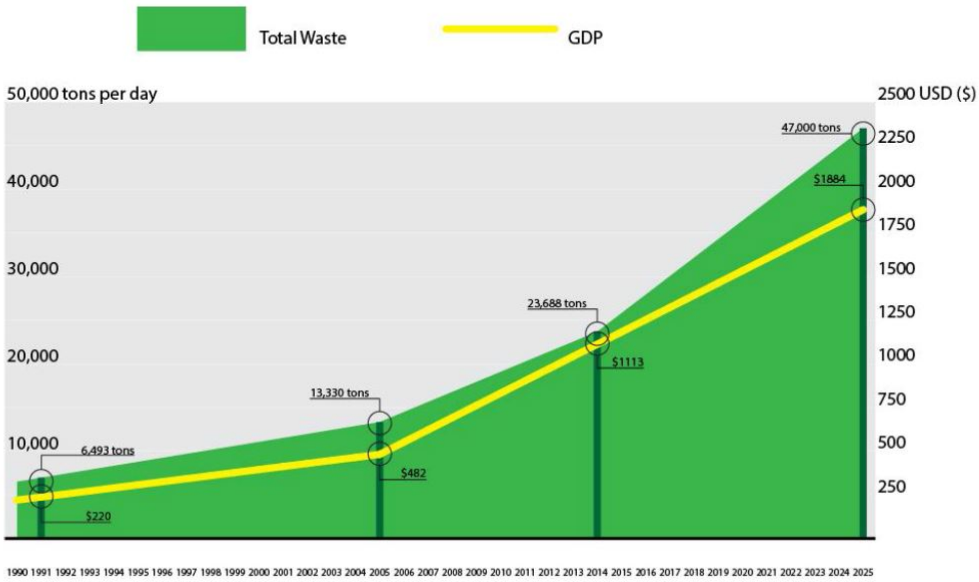


Figure 2.2: Total Waste Generation w.r.t GDP adopted from [4]

2.2 Waste Management

2.2.1 Waste Management Systems

The overall Waste handling process here is not structured in a profound manner. Figure 2.3 is a clear illustration of the existing garbage handling framework in Bangladesh.There are three roughly established process of waste management in Bangladesh - the first and the formal one is led by municipalities/city corporations where from collection to transportation and dumping of waste performed by the local agencies. In this framework, there is not recycling which is the most vital thing as the non bio degradable wastes are on the rise. The second one is the community led initiative, which is based on a small scale of MSW collection by NGOs and CBOs, Finally, the last one is comprised with the vast informal waste workers involved in the chain of solid waste recycling trade. So, we can conclude that, the recycling process is completely ignored (only 10-15% of total waste, that too by informal private channels) here as the separation is a tedious thing to do.

2.2.2 Recycling and Composting

Big, centralized and mostly automated plants along with small-scale & scattered locality based composting systems can be considered as a good option in the developing countries for treating MSW, as they drastically reduce logistic expenditures, employ cheap technologies at work, depends heavily on manual labor, and reduce problems, faced with backyard composting. There are multiple classifications for recycling. They usually consist of metal, glass, plastic, paper and organic kitchen waste. The recycling process varies from country to country depending on the waste varieties that is being generated is respective localities which makes the recycling process inconsistent across borders.

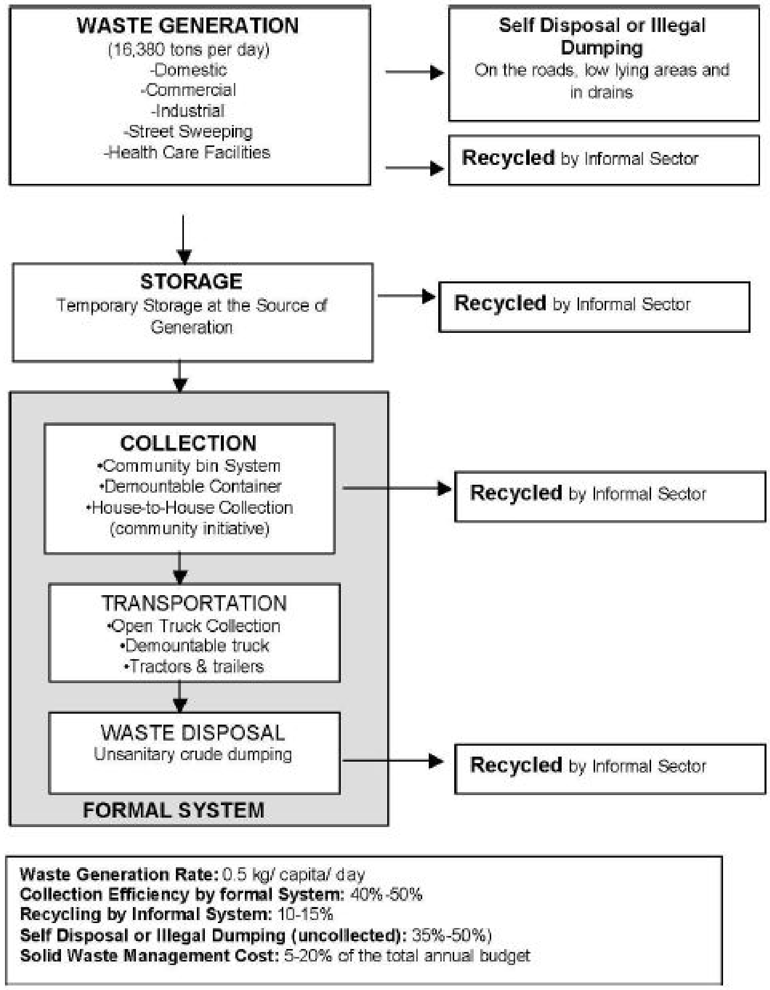


Figure 2.3: Waste management framework in Bangladesh adopted from [2]

2.3 Sensor Based Robotic Agent

Here a autonomous robotic agent has been thought of, which will have a eye camera to capture real time scenes while surfing around the designated places. The captures images will then be analysed and waste object will be detected and classified using a lite exported version of our model, having the capability of inference on a computing devices like Arduino or Rasberry Pi. Then the robotic agent will auto separate the garbage objects is their respective categories for recycling or permanent disposal.

2.4 Computer Vision

Now, People, who use cameras as the sensors of the micro controllers, usually rely on images to give them information about the objects that they were looking at. In general, they use computer vision to identify objects within an image. The name computer vision is kind of self explanatory which enables a computer to identify objects autonomously. From Satellite imaging to self driving cars and underwater image processing, computer vision has a wide range of applications which require the ability to identify objects from a scene.

2.5 Object Detection and Classification

Before diving into object detection, let's spend a few words on a more generalised topic - object recognition. Object recognition can be defined as an array of related computer vision tasks that involve recognising objects in images. Now, only classifying the whole image is relatively simple, but the task of localization and detection can be a bit tricky because in this case, first, the desired objects from the images should be detected and then the detected objects can be classified as per maximum likelihood among the predefined categories.

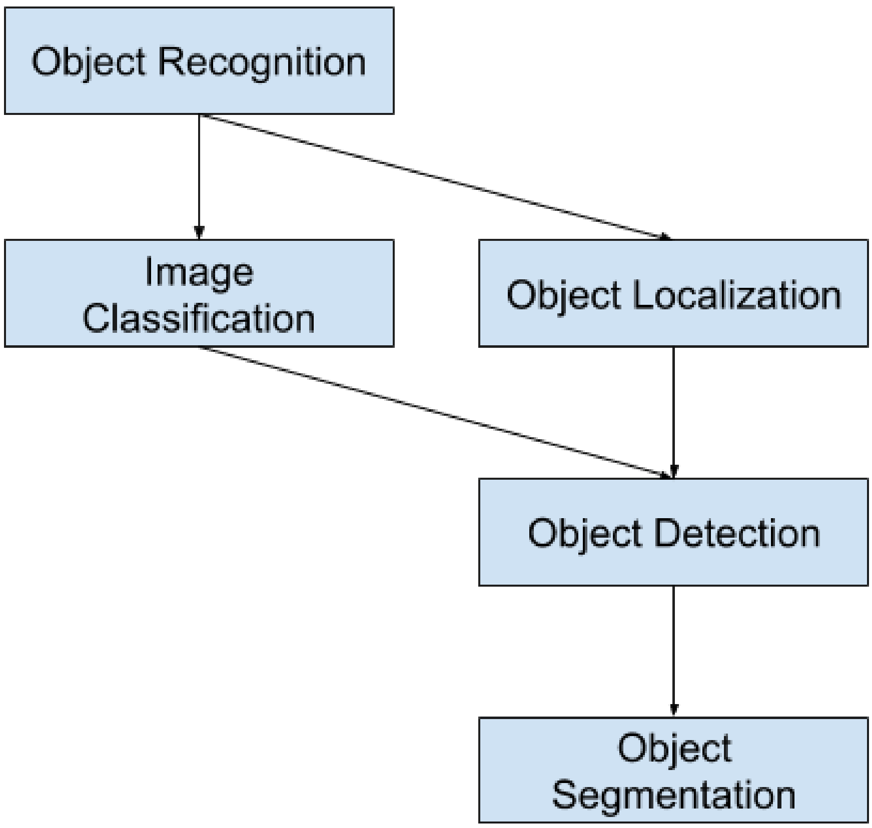


Figure 2.4: Overview of Object Recognition Computer Vision Tasks

So we can define the tasks as below -

- **Image classification:** Task requirement to output a list of categories, that are present in a particular image. Image classification algorithms take an image as input, transforms it in a giant one dimensional vector, process the input and yields the predicted label of that image with some metric like loss, probability, accuracy, etc.

- **Object detection**: Task requirement to output a list of categories of object, that are present in the image or frame along with a bounding box (commonly rectangular in shape), that indicates the position and scale of the instances of the object categories. Object Detection algorithms are a combo of both object localization & classification task. It takes an image as input, transforms it in a giant one dimensional vector, process it and outputs one or more bounding boxes with the class labels, depicted above each bounding box.

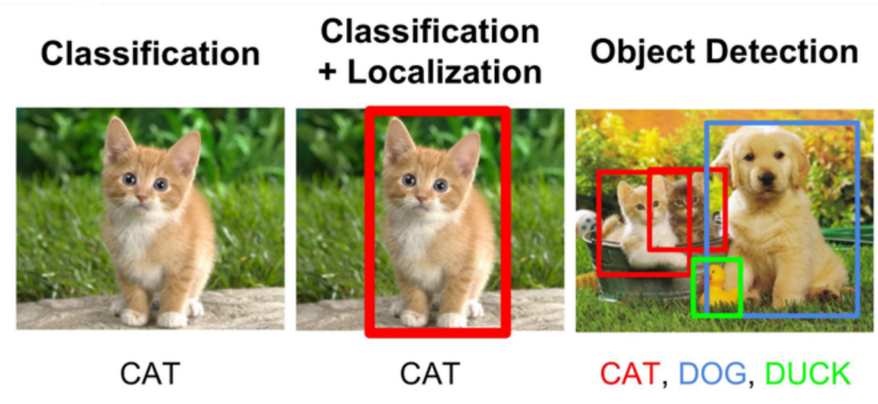


Figure 2.5: Image Classification vs Object Detection

Now, the family of object detection algorithms can be further classified into 2 major categories namely - (1) Single Stage Detectors and (2) Multi Stage Detectors

1. **Single Stage Detectors:** It refers to the family of algorithms that detects and classifies the object using a single network and one forward pass. This is relatively recent invention and attracted many more researchers mainly because of it's inference speed and practical applicability. These models can also be optimised in such a way that they can be used in mobile devices. In this project, I worked with a single stage detector - YOLOv3.

2. **Multi Stage Detectors**: Multistage detectors employ two networks - one for predicting bounding boxes around detected objects and another for classifying the object in the box. These algorithm is based on region proposals and comparatively slow to make predictions than their single stage counterparts. But they are more accurate than the single stage detectors. Examples - R-CNN, Masked R-CNN, Fast R-CNN etc.

2.5.1 YOLO Models

YOLO is an algorithm that uses CNN for object detection. Although it's not the most precise detectors out there, it is a very good choice the focus is on real time detection without losing too much of accuracy. This algorithm employs a single neural network to full image that effectively means - the network breaks down the frame into multiple local regions & predicts the bounding boxes and probabilities of those regions.The first version of this model, published in 2015, yields lower prediction accuracy, for example - greater localization errors, although operates at 45 FPS and up to an astounding number of 155 FPS for a speed-optimized lite version of the model. In YOLOv2, the authors adopted the idea of the anchor boxes from the R-CNN paper [6]. The choice of anchor boxes for the images are pre-determined running a clustering algorithm on the whole data set. The representation of the bounding boxes is modified a bit (instead of estimating the bounding box position and size directly, offset values are predicted for moving and reshaping the pre-defined anchor boxes relative to a grid cell) to allow small changes which eventually expected to have a less dramatic effect on the predictions, resulting in a more stable model.

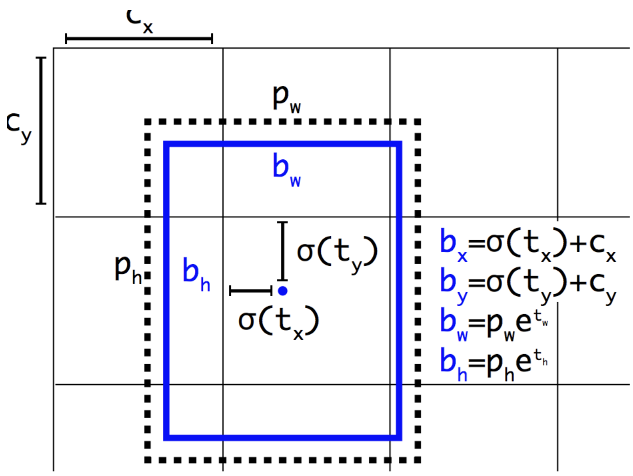


Figure 2.6: Bounding Box Representation, Taken from: Original Yolov2 paper

Now, our implemented YOLOv3 is an important improvement over the legacy models. It supports detection in multiple scales, has a stronger network as feature extractor and a few tweaks in the objective function as well. Consequently, the network is capable of detecting many more targets, from large to small.

Figure 2.7: High Level Network Architecture

Chapter 3

Related Works

Our local researchers pointed out the problem and it's severity in [2] & [5]. In [5], authors proposed a high level frameworks combining legal bindings, economic instruments and public awareness to tackle this problem. And in [2] authors portray the existing process of garbage collection to separation and processing scheme and expressed their proposition to haunt them down (mostly high level proposition demanding process changes). In [3] authors proposed an electro-mechanical sensor based system using micro controllers and operational amplifiers to auto separate common forms of wastes. But the practical applicability of this system is in question mainly because of the diversity of the waste objects possess which is difficult to capture with the amount of information that these sensors provide to make a prediction.

In [12], authors dealt with slightly easier problem which is tagging street cleanliness. They have used street cameras from different localities and tried to develop an array of localised models for area specific inference. That indicates training a lot of discrete model which doesn't seem like a scalable solution of the problem. Moreover, they used Naïve Bayes and Adaboost classifiers to make the prediction observing the whole image, which missed the detection granularity down to object level that is subjective for object separator to work. Recently, deep learning techniques have become growingly popular because of their capabilities to extract rich features automatically from images, highly efficient parallel processing on GPUs to satisfy these resource hungry algorithms, fast and efficient optimizing techniques and a wonderful research community. That is why most of the modern literature tried to address the problem using the same. In [10], authors used end to end deep learning based system to do a similar task for training efficiency - they have employed a pre-trained GoogleNet model as the feature extractor backbone. Moreover, this addressed the detection granularity down to object level which makes it a better candidate for waste separator model. They have used hand annotated images and also segregated the single objects and multiple clubbed objects with different class labels for improving classification efficiency.

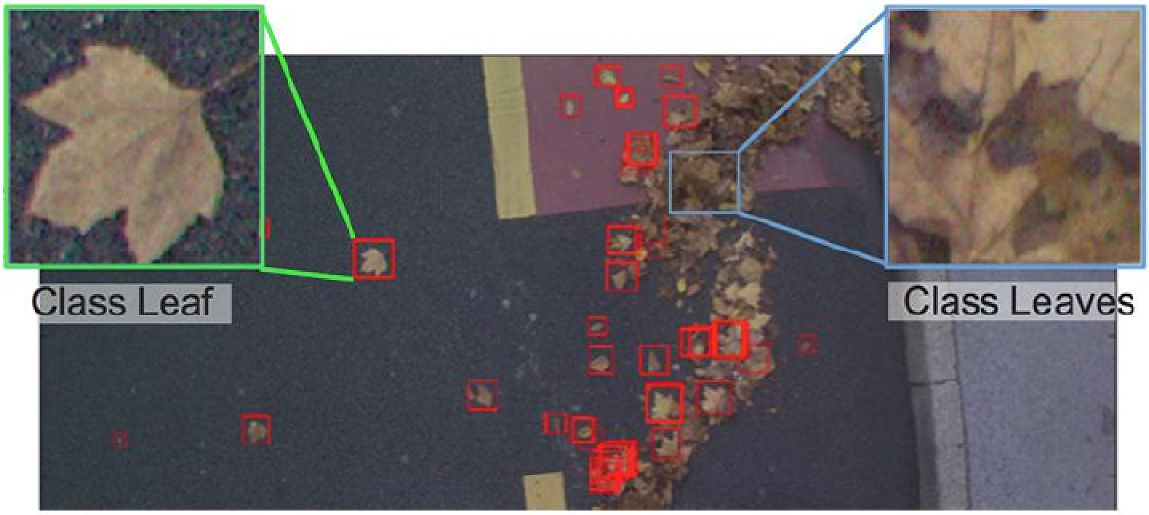


Figure 3.1: Representative of waste images from [10]

In [21], [19], authors proposed bin level segmentation using smart bins. They also employed a mixed approach of Deep Learning framework with Machine Learning Algorithms for the classification task and achieved some remarkable results. But as bin level detection and separation is not current infrastructure compatible from our country perspective, we can't adopt this approach.

In [8] authors adopted a deep learning based object localizer and classifier to figure out waste objects of 2 categories namely - bio degradable and non bio degradable. In [13], authors adopted state of the earth mobilenet, densenet and inception networks for waste object classification. But, the waste object is not represented contextually in the data set as they adopted the TrashNet data set and they have not taken the detection granularity to the object level as well.

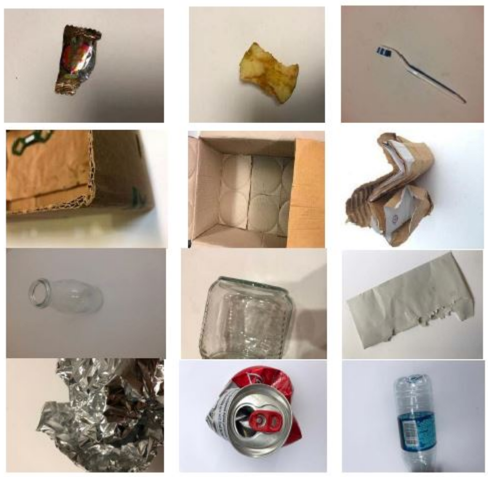


Figure 3.2: Representative of waste images from [13]

In [14], authors trained a multi-layer hybrid deep learning network. This system involves both high resolution camera and sensors to extract useful features and finally process the image features using a CNN based algorithm and deploys a Fully Connected MLP to blend image features with other information to classify waste as recyclable or others. This system tried to mimic human sensory organs like eyes with a pretrained AlexNet model and ears/noise with a MLP and achieved a better classification accuracy than other systems at that time. But 2 problems still persist - the model fails to pinpoint the waste object rather classifies the whole image frame which can contain multiple waste objects and although they employed image augmentation to present different form of the image to the model, the background and natural diversity of the real world is still absent here as seen from Figure 3.1. In [9], the authors tried with more linear classifiers like SVM and AlexNet based Neural network models with applied data augmentation. But the context is missing here as well which we can see from the Figure 3.2.

Figure 3.3: Representative of waste images from [14]

Figure 3.4: Representative of waste images from [9]

In [7] & [18], authors used more realistic images to deal with context problems and both contributed their own datasets GINI and TACO. But [7] offers only binary detection – garbage is present or not which takes out the recycling application and [18] detects only litters leaving the bio degradable wastes out of equation. Moreover, both of them used, a region proposal based 2-stage object detector which is slower to make inference on real time data.

Chapter 4

Data Set Preparation

Building the data set is one of the more challenging tasks of any machine learning project. The struggle becomes scarier when we don't have any benchmark data set for our problems - which actually happened in our case. Now, building data set can be done in a few different ways as follows. I shall briefly discuss the ins and outs of each strategies and our adoptions among them -

4.1 Publicly available open labelled data sets

There are popular open source data sets like ImageNet, COCO Data Set, Google Open Images Data Set etc. which contains hundreds and thousands of labelled images of hundreds of predefined categories. But in our case the required categories haven't been found in those large data sets, that's why we are unable to use this strategy for our project.

4.2 Web Scraping

We can also perform an image search on the web, and hand-pick to download manually or write an automated script to download the images. Now, images found on the internet are of lots of different sizes and some of them are very high resolution and some are of very low resolution. AS we'll convert all the images as per network requirements before passing them to a forward pass of training, varying sizes won't impose a technical problem. Having said that, some images are of pretty low resolution and our target task can contain 50-100 objects er frame which can make the job difficult for the detector on very low resolution images.
So, after downloading, we also manually inspected each image and took decision as per the merit of the image to portray the real world waste scenario. I have demonstrated a few sub categories of waste images that we've downloaded in fig-8. We have tried to cover the common categories of waste that are seen in the garbage dump in Bangladesh.

4.3 Capturing Images

When we are unable to find pictures of object that we need, we also can collect them by taking photographs. This can be achievable by (a) manually clicking each

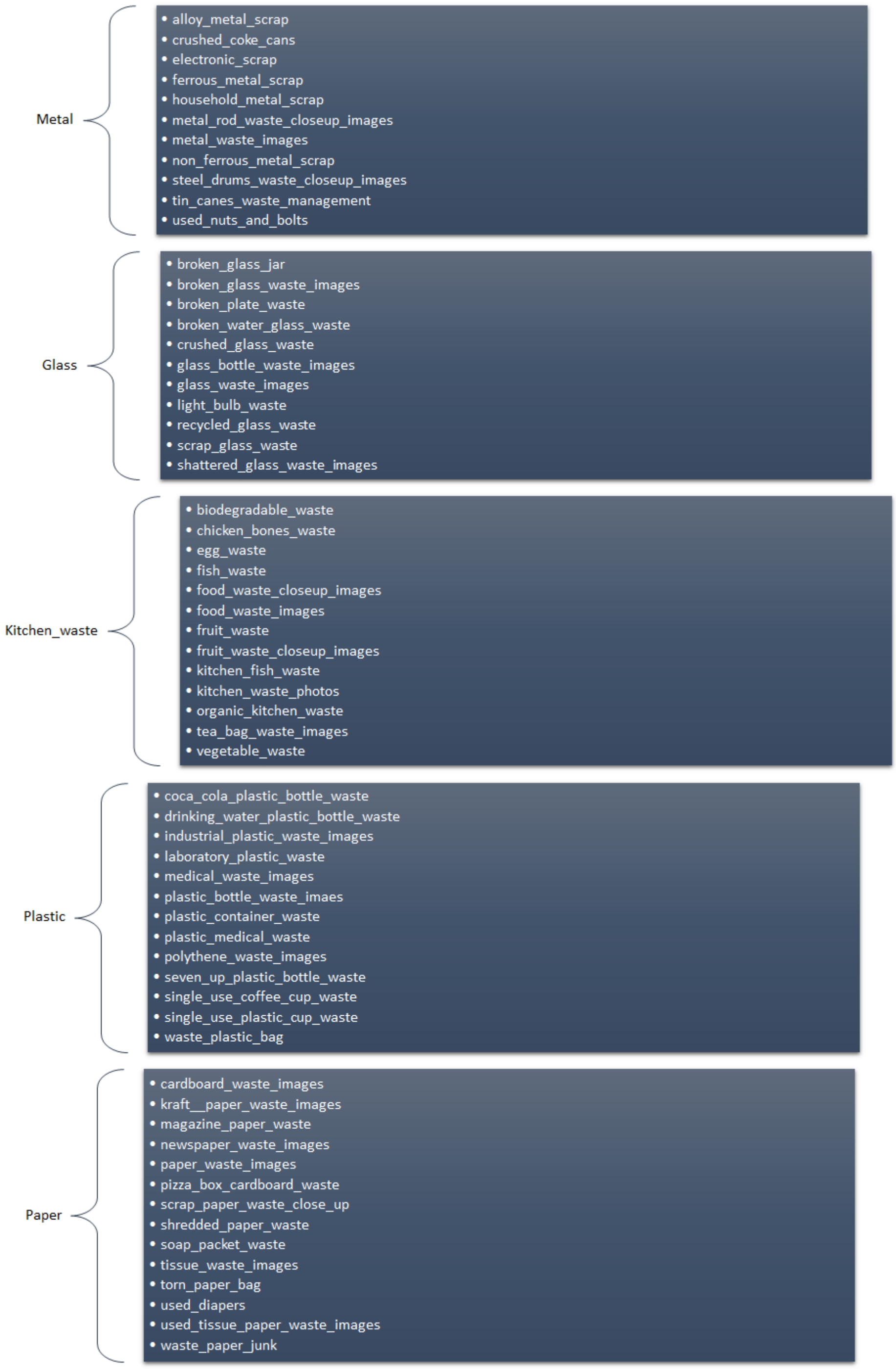


Figure 4.1: Search Category granularity to accumulate garbage photos through web scrapping

image by ourselves or (b) crowd sourcing, which is employing people to take images of the subject matter for us. (c) Installing programmed camera in our environment of choice. Due to lack of financial support and government endorsements, we were unable to take (b) and (c) as an option. So, we had to proceed with manually clicked photographs by ourselves.

4.4 Using Data Augmentation

As deep learning algorithms are data hungry ones, when we have a relatively small data set, it becomes difficult to train a good model. In these cases, different kinds of data augmentation can be adopted to extrapolate more training data. Transformations like flipping, cropping, rotation and translation are the most common data augmentation techniques. Using data augmentation in training images not only enriches our data set by creating variation, but also reduces model over fitting problem. But in our case, as we annotated the bounding box co ordinates around the objects in images, these transformations cause the co ordinates to be changed. There are techniques available to transform the labels as well but we have left it outside the scope of this work for time limitations and the complexity of this task. We'll adopt these techniques to enrich our data set more in future.

4.5 Synthetic Data Generation

Sometimes in the absence of abundant real data, some fake data can be developed for training our customized model. As it incur very low cost, usage of synthetic data generation is on the rise in Artificial intelligence domain. GAN is one of many techniques which is heavily used for artificial data generation. GAN is a modelling technique, where analytical instances are created from a data set to mimic the characteristics of the original data set. But due to complexity of training another GANs, this is out of our current scope as well.
As we can see from Figure- 4.2 and 4.3, we have made our efforts clear to leave minimal distribution difference between the two sources of image collection, in easy words, we have tried to carefully pick up photos form the scrapped images which can better mimic the real world scenario.

4.6 Data Set Preparation Steps

Finally, if we summarise the steps that we had taken for building the data set -

1. First, We've roamed a 10 KM radius around a suburb near my home and captured the spotted trash using Redmi 4X device and accumulated more than 1200 images. We have taken images from different angles and verticals trying to mimic a moving agent which will automate the process. All the images are captured in broad daylight at an optimum resolution.

2. Then, We have carefully investigated the images and identified the varieties and different orientations with which waste appears in open environment.

Figure 4.2: Annotated Images of Garbage Waste - Self Taken Photographs

Figure 4.3: Annotated Images of Garbage Waste - Web Scrapped Images

3. After that, we have wrote a web scrapper in python to download photos of given category (refer to Figure - 4.1) from google images automatically. We have accumulated photos of more than 70 different sub-categories waste, which belong to our main five categories.

4. From there, We have carefully investigated and picked 1000 photos from the collection that matches the real scenarios by hand.

5. We have annotated the final images using the labelImg tool [20] in PASCAL VOC format.

6. The annotated images are then further verified by two another human individual and finally used for model development purpose.

We have also tried to make our data set evenly balanced so that no data driven bias exists in the trained model.

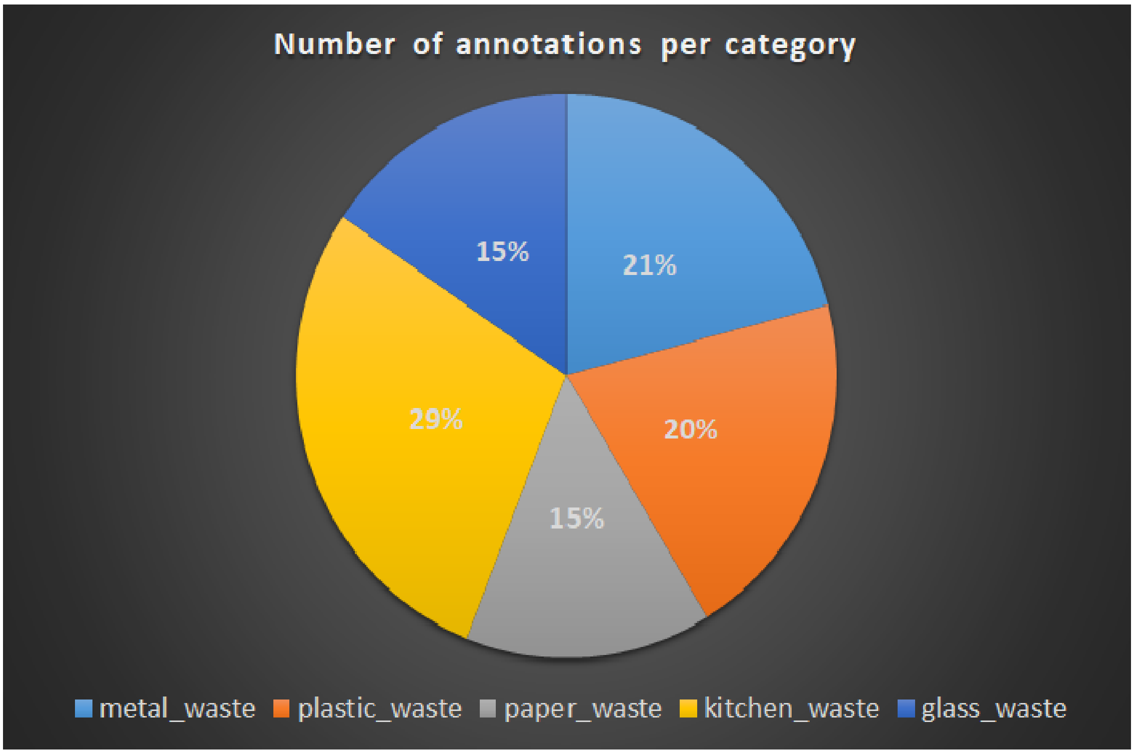


Figure 4.4: Annotation percentage per category

Chapter 5

Methodology

5.1 Model Architecture

The whole pipeline can be divided into two prime constituents : Feature Extraction layers, Object detection layers. During a forward pass, the pixel data passes through the feature extraction layers first, so that the inherent features at multiple different scales can be obtained. These feature values are then fed into at least three branches of the detectors (called Yolo Heads) to get the bounding box dimensions and class information.

5.1.1 Input of the network

Although the algorithm is invariant of input size of an image, in practise, we prefer to stick to a constant input size due to various problems - a big one among them is that training is batches demands the images to be of fixed height, weight and color channels. This is required for clubbing multiple images into large batches. In modern deep learning based computer vision research, the input dimension is usually chosen from 32x32 to 608x608 or even further based on the complexity of task and the computational trade off. In our case, we've sticked to an input dimension of 416*416*3 which is pretty reasonable for the task in hand.

5.1.2 Feature Extractor backbone

As stated above feature extractor backbone is responsible for extracting the rich features out the input pixel data and pass to the detection heads for object recognition and classification. In our case, we have used transfer learning for the feature extraction backbone. We have run several experiments with some of the most popular feature extractors these days and tweaked the activation function for some layers as well.

5.1.3 Darknet-53

This is a fully connected neural network having 53 convolutional layers, each followed by batch normalisation layer with stride 2 which is used for down sampling the feature maps. This particularly helps to prevent loss of low level features often associated with applying the pooling operation. The network down samples the

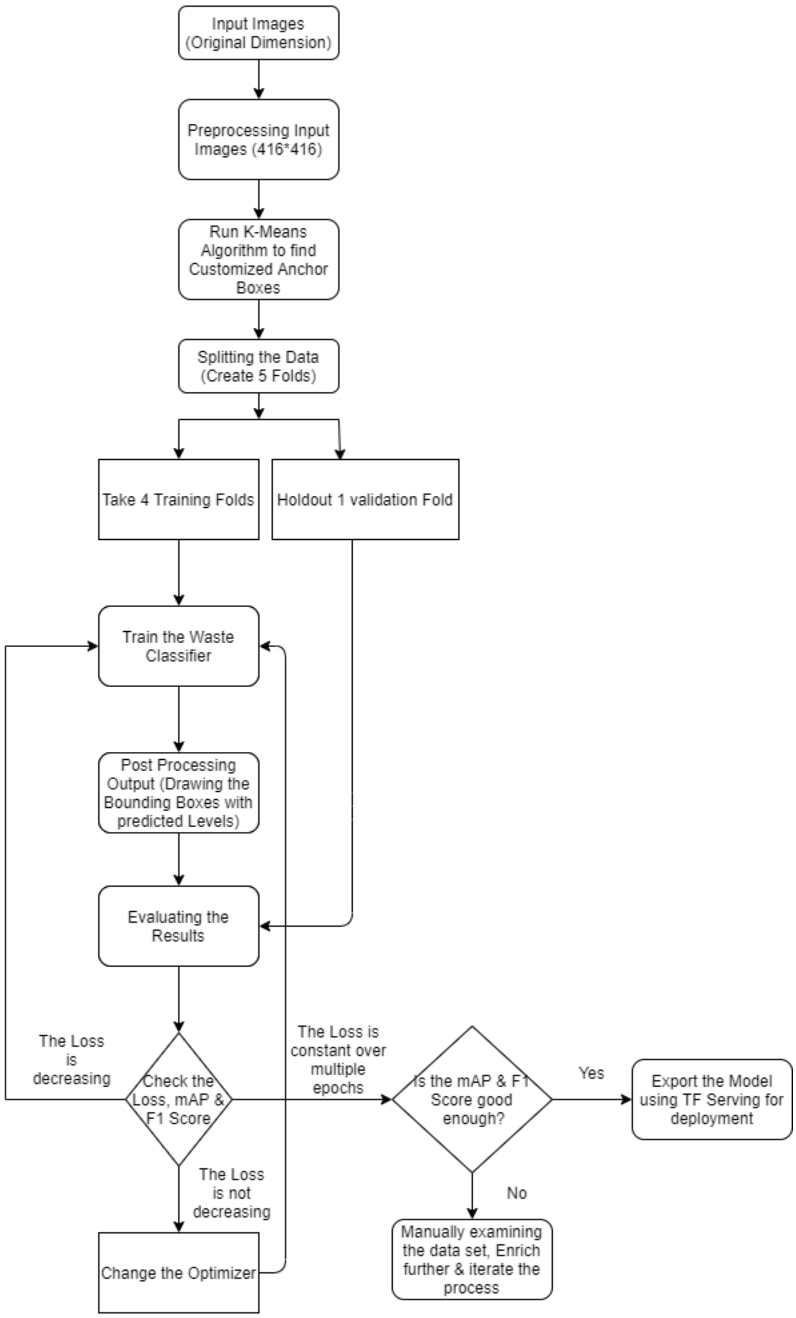


Figure 5.1: The End to End Model Development Process

image by a factor, called the stride of the network. For example, if the network stride is 32, then an input image of 416x416 will result in an output size of 13x13. Basically, Darknet-53 borrowed the idea of skip connections from the famous ResNet paper and successfully extends it's layers from it's predecessor.
If we think of each rectangular box as a residual block then the whole network can be thought of as an array of multiple consecutive blocks with stride 2 convolutional layers in between to step down the dimension. Within the block, there's a bottleneck shape, that is - 1*1 conv followed by a 3*3 conv, plus a skip connection. Now as illustrated in [17], our detector head demands a higher input network size for detecting multiple small-sized objects (which we have already chosen as an optimum resolution of 416*416), More layers for a higher receptive field (we have stretched the layers from 53 to 107) and finally more parameters for increasing the capacity of the model for detecting multiple objects with varying sizes. The receptive field here is very important when it comes to object recognition task. The influence of different sizes of receptive fields can be organised like - (a) If the size of the receptive field matches the object size, it allows the model to view the whole object, (b) if it matches the network size, it enables the model to view the context around the object as well (c) if it exceeds the network size, it widens the connection amounts between the image pixels and the final activation.

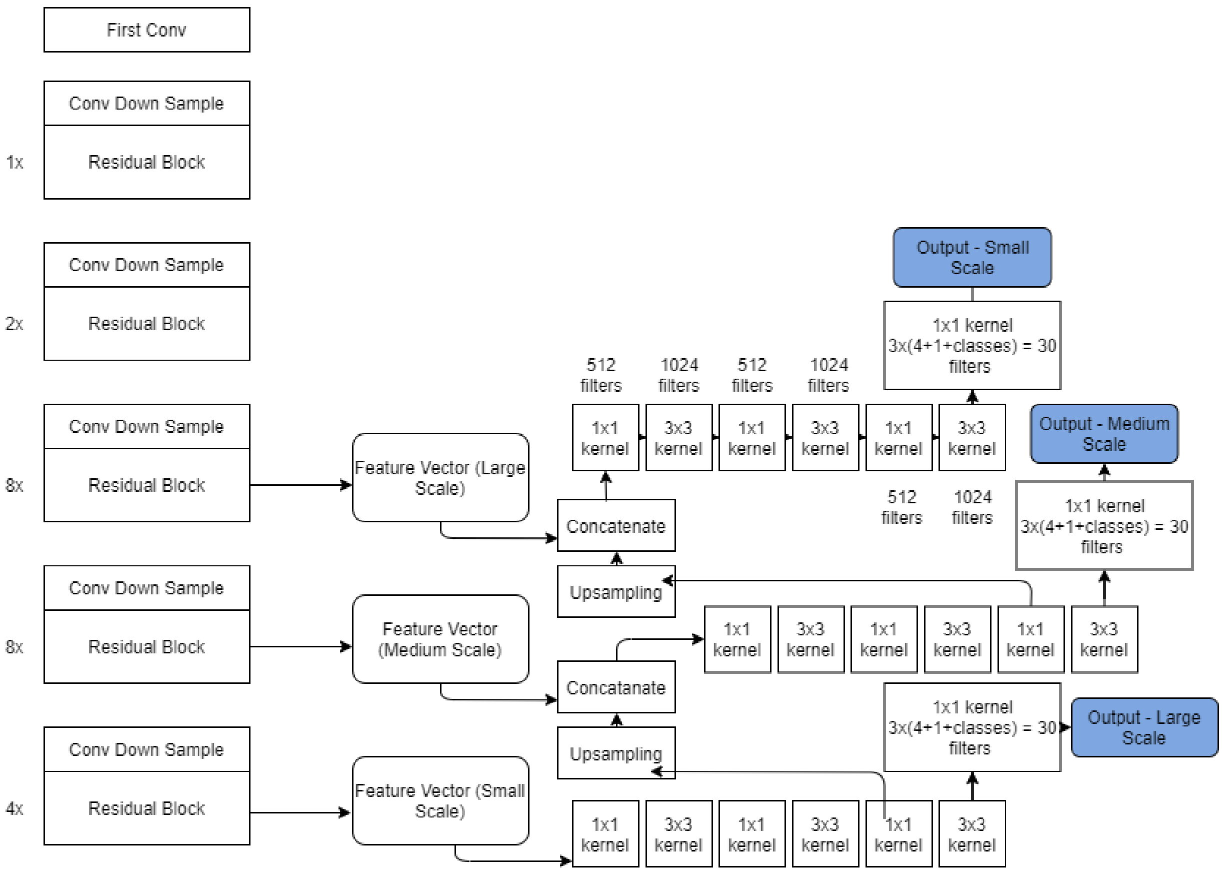


Figure 5.2: The High Level Architecture

$$n_{out} = \frac{n_{in} + 2p - k}{s} + 1 \tag{5.1}$$

$$j_{out} = j_{in} * s \tag{5.2}$$

$$r_{out} = r_{in} + (k-1) * j_{in} \tag{5.3}$$

$$start_{out} = start_{in} + (\frac{k-1}{2} - p) * j_{in} \tag{5.4}$$

These are the equations to calculate the receptive field arithmetic.

- The number of output features based on the number of input features and the convolution properties is estimated by the equation 5.1.
- The equation 5.2 estimates the deviation in the output feature map, which is identical to the change in the input map times the number of input features that we skip over when applying the convolution operation (the size of the stride).
- The equation 5.3 estimates the size of the receptive field of the output feature map, which can be derived from the area that covered by 'k' input features

  $$(k-1) * j_{in} \tag{5.5}$$

  adding the extra area which is covered by the receptive field of the input feature map.
- The equation 5.4 estimates the center position of the receptive field of the first feature output, which is equal to the middle position of the first input feature adding to the euclidean distance from the point of the first input feature to the center of the first convolution

  $$(k-1)/2 * j_{in} \tag{5.6}$$

  subtracting the padding space

  $$p * j_{in} \tag{5.7}$$

  Point to be noted that multiplication with the jump of the input feature map is required to get the actual distance.

So, we have done experiments with both darknet-53 and an extended and a bit modified version CSPdarknet-53 and chosen CSPdarknet-53 as the backbone feature extractor as it reduced the number of false negatives significantly.

5.2 Multi-scale detection

Now, a detection head is needed with this feature extraction part. Since our model here is intended to be a multi-scaled object predictor, we additionally need highlights from various scales. Accordingly, highlights from last three lingering blocks are totally utilized in the object recognition task. As our image s scaled to a fixed size of 416x416, so three scale vectors would be 52x52, 26x26, and 13x13.
Now, comes to the final stage of the network that illustrates - how the feature vectors are wired to the final outputs. Here, some more bottleneck layers (multiple 1x1 and 3x3 Conv layers) are used in front of the final 1x1 Conv layer to constitute the final output. For medium and little scope, it likewise links highlights from the past scale. Consequently, little scope location can likewise profit by the consequence of huge scope discovery.

5.3 Output of the Network

The final output of the network would be [(52, 52, 3, (4 + 1 + 5)), (26, 26, 3, (4 + 1 + 5)), (13, 13, 3, (4 + 1 + 5))] where the 3 items in the list represent detection in three scales. In each tuples within the list, the first three values corresponds to the anchor box dimensions [6], the last one (4 + 1 + 5) represents - 4 (The location offset against the anchor box: x, y, w, h); 1 (The objectness score that indicates the probability of the box containing an object); 5 (The number of classes, which is, in our case -5)

5.4 Loss Function

With the final detection result at hand, we will be able to calculate the loss with respect to the true labels now. The loss function consists of four parts: (1) centroid (xy) loss, (2) width and height (wh) loss, (3) objectness (object and no object) loss and (4) classification loss. Putting everything together in perspective, the final equation looks like below :

$$\begin{aligned} TotalLoss = lambda * Sum(MSE((x,y),(x',y') * object\_mask) \\ +Lambda * Sum(MSE((w,h),(w',h') * object\_mask) \\ +Sum(BCE(object, object') * object\_mask) \\ +Lambda\_Noobj * Sum(BCE(object, object') \\ *(1 - object\_mask) * ignore\_mask) \\ +Sum(BCE(class, class') * object\_mask) \end{aligned} \tag{5.8}$$

Now if we break down the loss function into pieces -

$$xy\_loss = Lambda * Sum(MSE((x,y),(x',y') * object\_mask) \tag{5.9}$$

here x & y is the relative centroid location from the true values. x' and y' is the centroid expectation from the indicator straightforwardly. The more modest this loss is, the closer the centroids of expectation and true values are. Since this is a relapse issue, we utilize ordinary MSE here. Additionally, if there's no item starting from the earliest stage for specific cells, we don't have to incorporate the loss of that cell into the last loss. Subsequently, we likewise multiply with object mask here. object mask is either 1 or 0, which demonstrates if there's an item or not. Finally, Lambda is a hyper parameter that was presented in YOLO v1 paper. This is to put more accentuation on localization rather than characterization. The worth recommended is 5.

$$wh\_loss = Lambda * Sum(MSE((w,h),(w',h') * object\_mask) \tag{5.10}$$

Same way as above, this is the height and width loss of the predicted bounding boxes.

$$objectness\_loss = Sum(BCE(obj, obj') * object\_mask) \tag{5.11}$$

$$\begin{aligned} noobject\_loss = Lambda\_Noobject * Sum(BCE(obj, obj') \\ *(1 - object\_mask) * ignore\_mask) \end{aligned} \tag{5.12}$$

The 3rd & 4th items are the objectness and non-objectness loss. Objectness shows how probably is there an item in the current cell. Here, twofold cross-entropy loss has been utilized. In the ground truth, objectness is consistently 1 for the cell that holds an object, and 0 for the cell that doesn't have any object. By estimating this object_loss, we can steadily show the model to distinguish an area of intrigue. Meanwhile, we don't need the algorithm to cheat by proposing objects all over the place. Henceforth, we need noobject_loss to punish those bogus positive proposition. We get false positives by veiling forecast with 1-object_mask. The ignore_mask is utilized to ensure we possibly punish when the current box doesn't have a lot of cover with the ground truth box. In the event that there is, we will in general be milder on the grounds that it's quite near the appropriate response. We likewise need this Lambda_Noobject = 0.5 to ensure the model won't be overwhelmed by cells that don't have objects.

$$class\_loss = Sum(BCE(class, class') * object\_mask) \tag{5.13}$$

Finally the classification loss. As we have 5 classes altogether, the class and class' can be visualised as the one-hot encoding vector that has 5 qualities. We additionally apply twofold cross-entropy for each class individually and summarize them since they are not totally unrelated. Furthermore, as we did to different misfortunes, we likewise increase by this object_mask so we just check those cells that have a ground truth object.

5.5 Some Additional Tweaks

5.5.1 Yolov4 Neck

Finally based on the YOLOv4, also a detection neck has been used which basically works as a feature aggregator - it mixes and combines the features formed in the ConvNet backbone to prepare for the detection step.The components of the neck typically flow up and down among layers and connect only the few layers at the end of the convolutional network.

5.5.2 Yolov4 Bag of Specials

YOLOv4 introduces a strategy called a "Bag of Specials", so termed because they add marginal increases to inference time but significantly increase performance, so they are considered worth it.
The authors [17] experiment with various activation functions. Activation functions transform features as they flow through the network. With traditional activation functions like ReLU, it can be difficult to get the network to push feature creations towards their optimal point. So the research has been done to produce functions that marginally improve this process. Mish is an activation function designed to push signals to the left and right.[11]

Finally, we have chosen the best backbone feature extractor, detector head, feature aggregator neck, improved loss function and tested techniques like a few of "Bag of

Freebees" and "Bag of Specials" to form the best Object Detector Architecture as of today.

Chapter 6

Implementation & Result Analysis

In this section, we basically will demonstrate the performance of our Object Detector system to effectively detect and classify our predefined category of wastes. Experiments carried out on our develop data set (discussed in chapter 4).

6.1 Implementation Details

Below are the key highlights of our training set up -

- We have used pre-trained weights to initialize our network. The weights are from hour of training the network on COCO data set. AS our data set is small in size, we have used this efficient and established trick to converge faster on our problem. We have also freeze the earlier layers of our network and train only the later layers as the earlier layer already worked great in detecting low level features on the source task and we assume it will do the same at our transferred task as well.

- The pre defined anchor box sizes mentioned my the YOLO authors in his implementation is - (10,13), (16,30), (33,23), (30,61), (62,45), (59,119), (116,90), (156,198), (373,326). In our case, we have run a K-Means algorithm on our whole data set anchor sizes, to figure out the correct values specific to our problem. The yielded values are - (53,103), (166,326), (260,503), (397,621), (445,641), (531,774), (555,886), (758,1023), (841,1169)

- We have adopted a 5-fold cross validation method to iteratively search the best model as our data set is small. For each fold, the data set is split randomly into 80% training set and 20% development set. From our experiments, we haven't found any significant deviation in the validation loss/accuracy among these 5 folds. So, we have chosen the model which yields the maximum validation accuracy among them.

- We have used Tensorflow 2.0 for model building and training purpose. The training logs like training and validation loss, accuracy, learning rate decay etc. have been monitored by Tensorboard tool.

- The polynomial decay learning rate scheduling technique has been adopted with a learning rate of 0.01 initially; Both Adam and RMSprop is used as

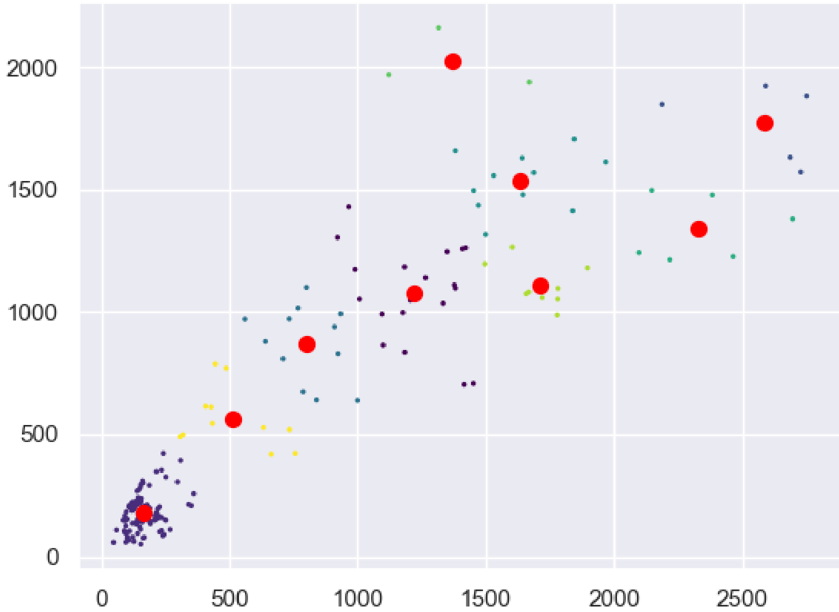


Figure 6.1: Customised Anchor Box coordinates determined by running K-Means Algorithm over the whole data set images

optimizers during experiments - finally sticked with RMSprop; the momentum has been set to 0.9 and weight decay to 0.005.

- Different experiments have been carried out with varying IoU thresholds while the objectness score is set to 0.40 for reducing the number of false positives. We'll show the results in a tabular format later. I have adopted popular Mean Average Precision (mAP) [1] as my accuracy metrics which is obtaining Average Precision over the number of classes and averaging them.

- All the training and experiments are run on on a Intel(R) Core(TM) i5-6500 CPU @3.2GHz, 12GB RAM, 4GB GeForce GTX 1050 Ti GPU.

6.2 Training History

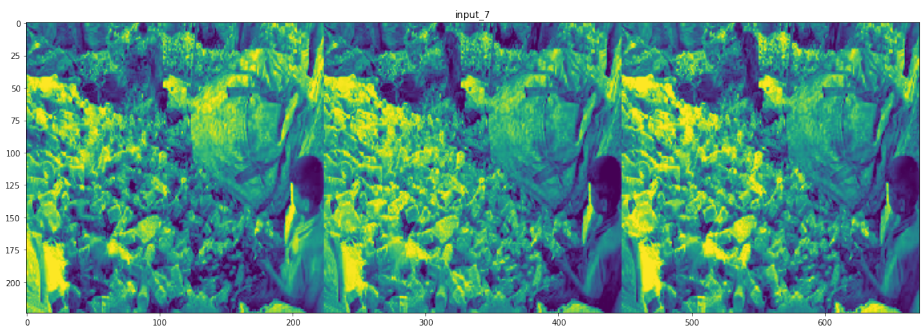


Figure 6.2: Feature Visualisation from Feature Extractor Output of glass images

Figure 6.3: Feature Visualisation from Feature Extractor Output of plastic images

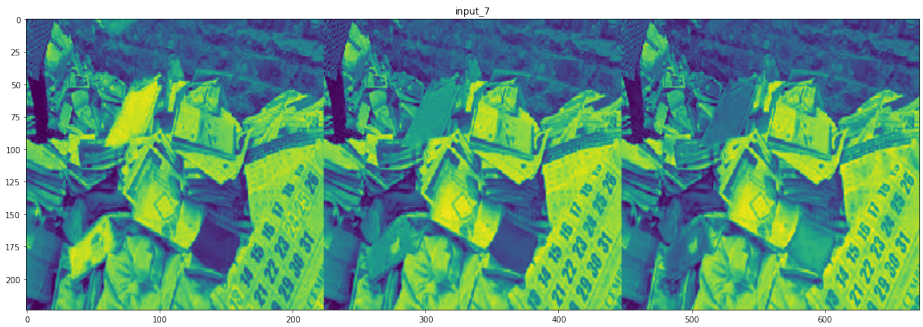


Figure 6.4: Feature Visualisation from Feature Extractor Output of paper images

6.3 Results Analysis

6.3.1 Metrics at I0U- 0.50

Calculation mAP (mean average precision)...

detections count = 26480, unique truth count = 2956
class id = 0, name = metal, AP = 64.09% (TP = 368, FP = 219)
class id = 1, name = plastic, AP = 61.72% (TP = 316, FP = 214)
class id = 2, name = paper, AP = 43.07% (TP = 225, FP = 195)
class id = 3, name = kitchen_waste, AP = 60.73% (TP = 485, FP = 232)
class id = 4, name = glass, AP = 53.86% (TP = 167, FP = 77)

for confidence threshold = 0.25, precision = 0.62, recall = 0.53, F1-score = 0.57

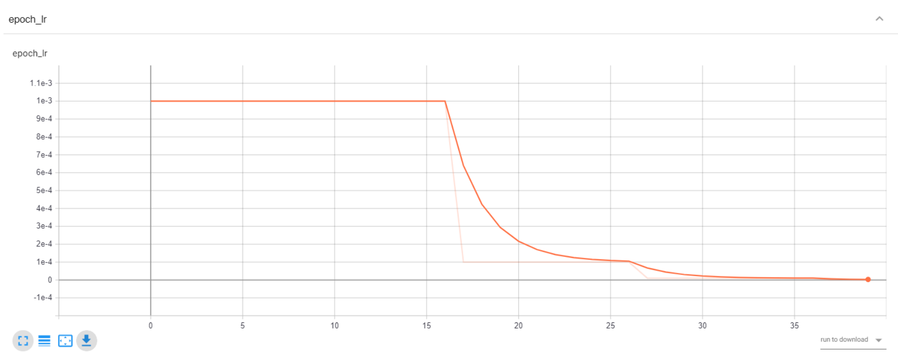


Figure 6.5: Learning rate decay during training

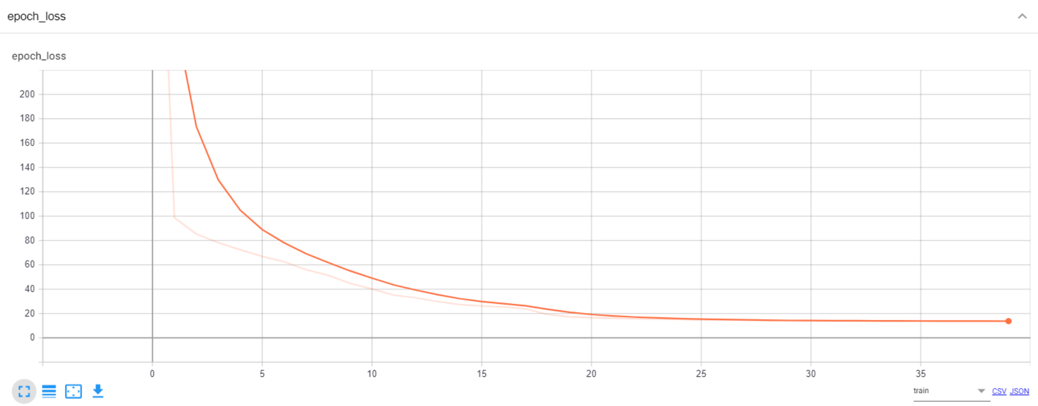


Figure 6.6: Training Loss deceasing over epochs during training

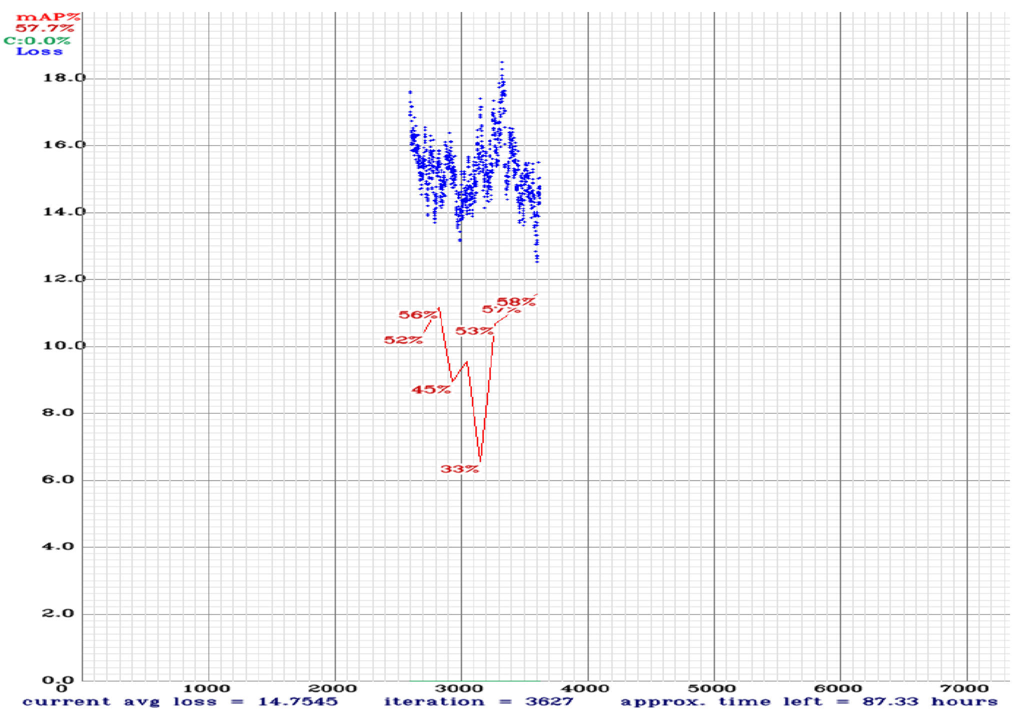


Figure 6.7: Validation Loss and mAP Monitoring during Training

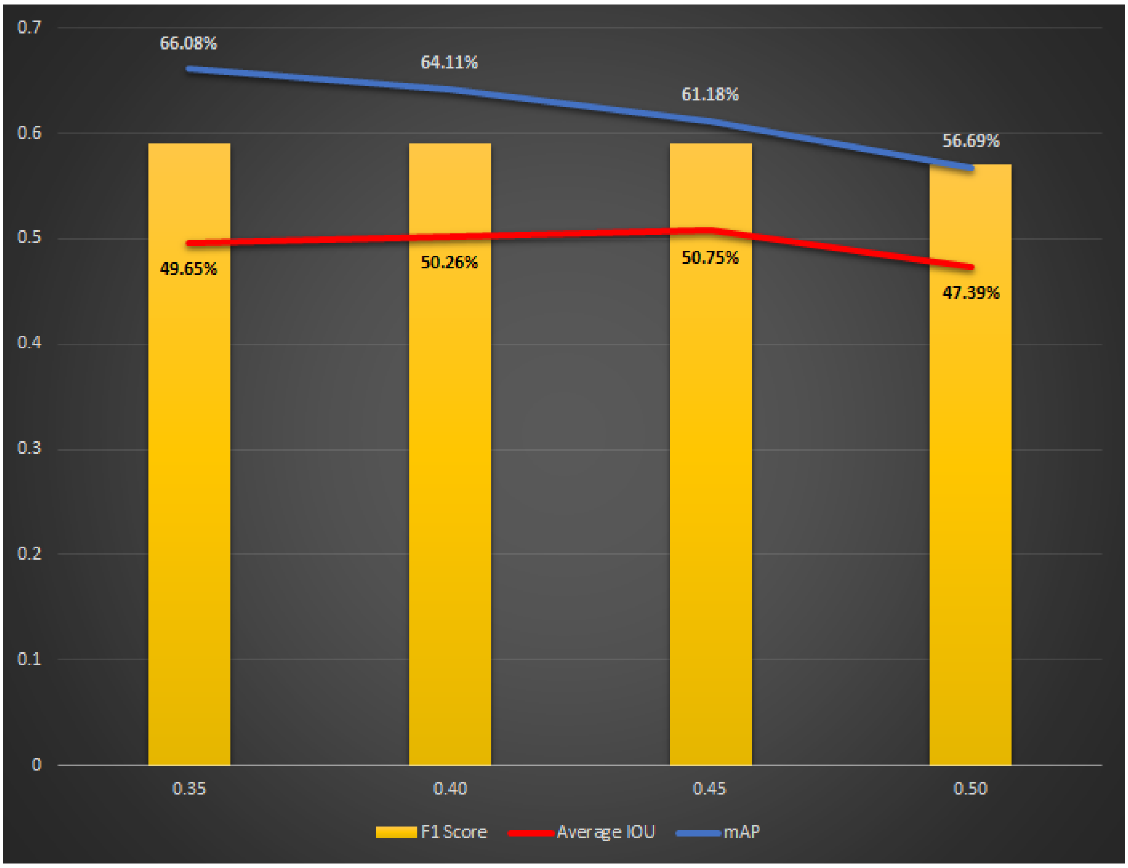


Figure 6.8: mAP, Average IoU & F1-Score Over different IoU Thresholds

for confidence threshold = 0.25, TP = 1561, FP = 937, FN = 1669, average IoU = 47.39 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.566944, or 56.69 %

6.3.2 Metrics at I0U- 0.45

detections count = 22374, unique truth count = 2956
class id = 0, name = metal, AP = 69.67% (TP = 376, FP = 185)
class id = 1, name = plastic, AP = 62.95% (TP = 243, FP = 78)
class id = 2, name = paper, AP = 50.79% (TP = 241, FP = 164)
class id = 3, name = kitchen_waste, AP = 69.11% (TP = 534, FP = 241)
class id = 4, name = glass, AP = 53.37% (TP = 143, FP = 54)

for confidence threshold = 0.25, precision = 0.68, recall = 0.52, F1-score = 0.59
for confidence threshold = 0.25, TP = 1537, FP = 722, FN = 1419, average IoU = 50.75 %

IoU threshold = 45 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.45) = 0.611760, or 61.18 %

6.3.3 Metrics at I0U- 0.40

detections count = 22374, unique truth count = 2956
class id = 0, name = metal, AP = 69.67% (TP = 376, FP = 185)
class id = 1, name = plastic, AP = 62.95% (TP = 243, FP = 78)
class id = 2, name = paper, AP = 50.79% (TP = 241, FP = 164)
class id = 3, name = kitchen_waste, ap = 69.11% (TP = 534, FP = 241)
class id = 4, name = glass, AP = 53.37% (TP = 143, FP = 54)

for confidence threshold = 0.25, precision = 0.68, recall = 0.52, F1-score = 0.59
for confidence threshold = 0.25, TP = 1537, FP = 722, FN = 1419, average IoU = 50.75 %

IoU threshold = 40 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.40) = 0.641065, or 64.11 %

6.3.4 Metrics at I0U- 0.35

detections count = 22374, unique truth count = 2956
class id = 0, name = metal, AP = 76.80% (TP = 385, FP = 176)
class id = 1, name = plastic, AP = 65.67% (TP = 227, FP = 94)
class id = 2, name = paper, AP = 57.55% (TP = 244, FP = 161)
class id = 3, name = kitchen_waste, AP = 73.06% (TP = 535, FP = 240)
class id = 4, name = glass, AP = 57.32% (TP = 146, FP = 51)

for confidence threshold = 0.25, precision = 0.68, recall = 0.52, F1-score = 0.59
for confidence threshold = 0.25, TP = 1537, FP = 722, FN = 1419, average IoU = 49.65 %

IoU threshold = 35 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.35) = 0.660795, or 66.08 %

As IoU threshold 0.35 yielded decent mAP of 66.08% and F1 Score of 0.59 with good enough average IoU of 49.65%, we recommend to use this setting in production environment.

Attributes	Results
Network Used	CSPDarknet53 YOLOv4
mAP	66.08%
F1 Score	0.59
Sensitivity	52.55%
Total BFLOPs	59.592
Infer time	70 milliseconds
FPS	32

Table 6.1: The highlights of accuracy metrics, inference efficiency of our model

6.4 Checkout the model on some test data

Figure 6.9: Sample inference on real world data (1)

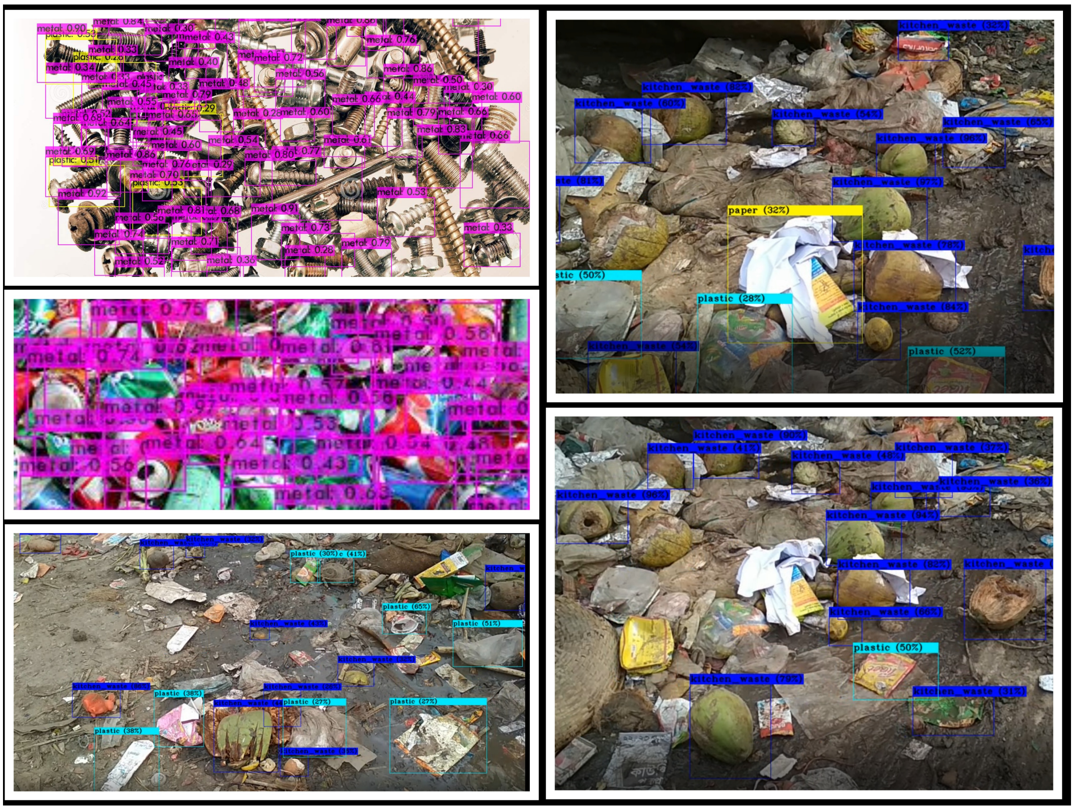


Figure 6.10: Sample inference on real world data (2)

Figure 6.11: Sample inference on real world data (3)

Chapter 7

Conclusion & Future Research

So, here we have proposed a deep learning based single stage waste detector and classifier which can make real time predictions of garbage objects in contextual images both in images and videos. There is still a huge room for improvement in this work. Our model is doing a bit poorly when it comes to classifying metal objects and glass objects as the lack of real world waste photos regarding those categories. The data set can be further enriched with a lot of training images - which is not feasible single handedly - so, financial support and collaborative effort is required for collecting and annotating huge amount of images from garbage dump using the techniques that has been elaborated in chapter 4. The network architecture can be further improved using Reinforcement learning based automatic network search leading to some manual tweaking as well, as described in [16]. However, we have proposed a novel deep learning based approach to fight against one of the most prominent problems prevails in our country at the moment. So, we believe it's s solid baseline to take the work forward and enrich further.

Bibliography

[1] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge”, *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[2] K. Bahauddin and M. Uddin, “Prospect of solid waste situation and an approach of environmental management measure (emm) model for sustainable solid waste management: Case study of dhaka city”, *Journal of Environmental Science and Natural Resources*, vol. 5, no. 1, pp. 99–111, 2012.

[3] M. M. Russel, M. Chowdhury, M. Shekh, N. Uddin, A. Newaz, M. Mehdi, and M. M. M. Talukder, “Development of automatic smart waste sorter machine”, Nov. 2013.

[4] W. Concern, “Bangladesh waste database 2014”, *Dhaka: Waste Concern. Retrieved June*, vol. 29, p. 2019, 2014.

[5] 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.

[6] R. Girshick, “Fast r-cnn”, in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[7] 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.

[8] 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.

[9] M. Yang and G. Thung, “Classification of trash for recyclability status”, *CS229 Project Report*, vol. 2016, 2016.

[10] M. S. Rad, A. von Kaenel, A. Droux, F. Tieche, N. Ouerhani, H. K. Ekenel, and J.-P. Thiran, “A computer vision system to localize and classify wastes on the streets”, in *International Conference on computer vision systems*, Springer, 2017, pp. 195–204.

[11] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for activation functions”, *arXiv preprint arXiv:1710.05941*, 2017.

[12] A. Alfarrarjeh, S. H. Kim, S. Agrawal, M. Ashok, S. Y. Kim, and C. Shahabi, “Image classification to determine the level of street cleanliness: A case study”, in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, IEEE, 2018, pp. 1–5.

[13] 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.

[14] 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.

[15] EPA, “Recycling basics”, 2019.

[16] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, *et al.*, “Searching for mobilenetv3”, in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1314–1324.

[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection”, *arXiv preprint arXiv:2004.10934*, 2020.

[18] P. F. Proença and P. Simões, “Taco: Trash annotations in context for litter detection”, *arXiv preprint arXiv:2003.06975*, 2020.

[19] G. White, C. Cabrera, A. Palade, F. Li, and S. Clarke, “Wastenet: Waste classification at the edge for smart bins”, *arXiv preprint arXiv:2006.05873*, 2020.

[20] Tzutalin, “Labelimg”, [Online]. Available: https://github.com/tzutalin/labelImg.

[21] D. Vinodha, J. Sangeetha, B. C. Sherin, and M. Renukadevi, “Smart garbage system with garbage separation using object detection”,